ITHEA

International Journal INFORMATION THEORIES & APPLICATIONS

2003 Volume 10 Number 2

International Journal
INFORMATION THEORIES & APPLICATIONS
ISSN 1310-0513
Volume 10 / 2003, Number 2

IJ ITA is official publisher of the scientific papers of the members of the Association of Developers and Users of Intellectualized Systems (ADUIS).

IJ ITA welcomes scientific papers connected with any information theory or its application.
Original and non-standard ideas will be published with preferences.

Papers must be written in English.

Responsibility for papers published in IJ ITA belongs to authors.
Please get permission to reprint any copyrighted material before you send it to IJ ITA.

IJ ITA rules for preparing the manuscripts are compulsory.
*The **rules for the papers** for IJ ITA as well as the **subscription fees** are given on* **www.foibg.com/ijita** .

The camera-ready copy of the paper should be received by e-mail: foi@nlcv.net

International Journal "INFORMATION THEORIES & APPLICATIONS"
Vol.10, Number 2, 2003
Printed in Bulgaria
Edited by the Institute of Information Theories and Applications FOI ITHEA, Bulgaria
Publisher: FOI-COMMERCE - Sofia, 1000, P.O.B. 775, Bulgaria
www.foibg.com e-mail: foi@nlcv.net

ISSN 1310-0513

SELECTION OF THEMATIC NL-KNOWLEDGE FROM THE INTERNET

V.Gladun, A.Tkachev, V.Velichko, N.Vashchenko

***Abstract**: The paper deals with methods of choice in the INTERNET of natural-language textual fragments relevant to a given theme. Relevancy is estimated on the basis of semantic analysis of sentences. Recognition of syntactic and semantic connections between words of the text is carried out by the analysis of combinations of inflections and prepositions, without use of categories and rules of traditional grammar. Choice in the INTERNET of the thematic information is organized cyclically with automatic forming of the new key at every cycle when addressing to the INTERNET.*

***Keywords**: semantic analysis, information search, INTERNET.*

1. The purposes and base ideas

Among various variants of practical use of storehouses for the textual information the necessity to find the information having thematic unity prevails. These are needs of a scientist, a journalist, a politician, an official, a writer, a student. Usually the theme arises as one or several concepts, some initial situation having a number of blank valences and situational roles which serve as reference points for search of the new relevant information. The new information gives rise to new directions of search. This complex, sometimes psychologically painful, creative process requires the automated support. Thematic search needs laborious work with the texts stored in libraries, archives, the INTERNET, textual databases. Difficulty of this work consists, in particular, in necessity to select not the whole texts, but relevant to the theme fragments of texts. The contents of many texts are an interlacing of a number of themes. Thus, a problem arises of search inside textual documents of fragments, relevant to the given theme.

In the paper methods, software and results of selection of the thematic textual information are considered. The researches submitted in the paper continue the works published in [1-3].

The solving of the problem unites the following actions:

1) selection of texts or fragments of texts relevant to an investigated theme;
2) selection from the relevant information of the most important, first of all, such which defines and connects the most essential terminology of a theme;
3) representation of the chosen information in the user-friendly form.

Implementation of the specified actions is based on the following ideas:

1) to focus a technique of selection of the thematic textual information on the INTERNET, as on the most full storehouse of the textual data;
2) to combine search by key words with the semantic analysis of NL-texts;
3) to use semantic criteria for selection of the most important thematic information;
4) to organize automatic cyclic process of key words formation to investigate the theme as complete as possible.

2. A technique

The initial stage of the thematic information selection consists in search of textual documents in the INTERNET using the given key. Existing methods of information search in the INTERNET give out a lot of unnecessary for user “garbage” information which filtration takes too much time. The way out consists in use of the semantic criteria providing selection of the most essential characteristics of concepts concerning which the information is gathered.

The offered method is based on the assumption, that the most important user information is contained in *kernel constructions* of sentences. The term “kernel constructions” is used in transformation grammar for designation of simple base judgment by which transformation the sentence as a whole is formed. In our case the kernel construction consists of a subject, a predicate and a link.

The method represents cyclically repeating sequence of the following operations:

1. Selection of the given quantity (parameter) of texts using a key. A set of used search systems is unlimited. Now the program can use the following search systems: Yandex, Rambler, Meta-Ukraine, Aport, Google.
2. Selection in the found texts of the sentences containing a given key.
3. Selection in set of the sentences that were chosen in item 2, the sentences containing kernel constructions. For item 3 performing the natural-language semantic analyzer is used.
4. Formation *of n-step expansions of the kernel* of the selected sentences. *n*-step expansion of the kernel is a part of the sentence containing its kernel, and also the words connected in a tree of dependencies with elements of the kernel by paths which length does not exceed *n. n* is a user-given parameter.

The item 4 is performed on the basis of the semantic analysis of the sentence.

5. Selection in the set of the sentences chosen in item 3, such sentences in which *n*-step expansions of the kernel contain the given key.
6. Formation of a new key on the basis of the analysis of semantic representations of before selected sentences. Transfer to the item 1.

The initial key word is given by a user. New keys on the subsequent cycles of the algorithm are chosen among *terms* that are significant words used only within the limits of investigated domains. The terms are marked in the dictionary.

When choosing a new key, the degree of its relevance to the given theme is taken into account. The relevance is defined on the basis of results of semantic analysis of sentences. At the following cycle of the algorithm the term, that was not used earlier and has the greatest relevancy coefficient, is chosen as a key.

After a choice of a new key, actions 1 - 6 are repeated.

3. The semantic analysis

The basic operation of the semantic analysis of natural-language texts is recognition of the syntactic and semantic relations connecting words of the text. Recognition of relations is carried out on the basis of their descriptions (models). Such models are necessarily present at all methods of the analysis though it is not always obvious. In the majority of the analysis methods the process of recognition of relations is preceded with translation of initial natural-language representation of relations to be recognized in the language of categories of traditional grammar (gender, case, time, etc.). Rules of recognition of syntactic and semantic relations operate with grammatical descriptions of words. Binding to grammatical descriptions of elements of the text results in the following imperfections: heterogeneity of ways of processing separate words and word combinations; bulkiness of processing; complexity of adaptation to changes of lexicon and a user's domain; laboriousness of the research. Meanwhile, transition to grammatical descriptions is not an obligatory condition for performance of the semantic analysis of natural-language texts. The information necessary for recognition of syntactic and semantic relations is contained directly in the text. As a proof to that, there are "human" processes of the analysis of the natural-language texts, which are not connected with grammatical categories and rules. Therefore, it is competent another approach based on use of conformity between relations and means of their expression in natural-language texts. Recognition of syntactic and semantic connections between words is carried out by the analysis of combinations of inflections and prepositions, without using categories and rules of traditional grammar. By virtue of its basic features, such approach allows to exclude the imperfections named above.

Models of relations in which elements of natural-language texts are used for recognition of syntactic and semantic relations, we shall refer to as *lexical models of relations*. The algorithm of the semantic analysis of natural-language sentences on the basis of lexical models of relations is described in [1-3].

4. Implementation and results

The structure of the program complex realizing processes of thematic knowledge formation consists of the programs which are carrying out the following actions:

1. Selection in the INTERNET of the textual fragments containing a given key.
2. Formation of semantic representations of sentences (the linguistic processor).

3. Selection of sentences, relevant to a theme, on the basis of the analysis of semantic representations of sentences.
4. Choice of a new key.

At the present time lexical data and knowledge bases of the complex are created for Russian language.

As a result of working a program complex the text is formed which consists of separate sentences that are relevant to a theme designated by an initial key which is given by a user. For each sentence, the address of corresponding document is indicated. The set of sentences selected from one document allows generating a conception about its thematic relevance as a whole. The high level of relevance of the document may induce a user to choose this document for detailed studying. The set of all selected sentences throws light on an investigated theme as a whole. The degree of completeness of the selected information on a theme depends on efficiency of the used search machine and quantity of the texts chosen in the INTERNET. Experience of the complex exploitation shows that the set of sentences selected by a program on the basis of the thematic analysis, well correlates with result of “manual” selection of “useful” sentences by an end user. The complex provides the high degree of elimination of the information that is unnecessary for a user.

Conclusion

Above described method of thematic selection of information can be used for the information search not only in the INTERNET, but in any textual databases. We also consider it as the instrument for creation of ontology’s. The merit of the method is effective filtration of the information on the basis of criteria of relevancy to the given theme that is obtained at the cost of semantic analysis of sentences and a cyclic process of automatic selection of a new key at every cycle. The method allows comparatively simple adaptation to changes of a text language.

The literature

1. Gladun V.P. Processes of formation of new knowledge. - Sofia: СД "Педагог". 1994. - 192p. (in Russian).
2. Gladun V.P. Planning of decisions. Kiev: Наукова думка, 1987.-168p. (in Russian).
3. Gladun V.P. Natural language in purposeful systems.//DIALOG-2000. Applied problems. 2000, p.99-102. (in Russian).

Author information

Victor Gladun - V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: glad@aduis.kiev.ua

Alexander Tkachev - V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: glad@aduis.kiev.ua

Vitaly Velichko - V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: glad@aduis.kiev.ua

Neonila Vashchenko - V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine; e-mail: glad@aduis.kiev.ua

PROCESSING OF KNOWLEDGE ABOUT OPTIMIZATION OF CLASSICAL OPTIMIZING TRANSFORMATIONS

Irene L. Artemjeva, Margarita A. Knyazeva, Oleg A. Kupnevich

Abstract: *The article describes the structure of an ontology model for Optimization of a sequential program. The components of an intellectual modeling system for program optimization are described. The functions of the intellectual modeling system are defined.*
Keywords: *Knowledge based system; Program optimization; Domain ontology*

Developing knowledge-based systems for any domain needs constructing its ontology [Kleshchev, 2002]. Ontology is an explicit description of domain notions and contains terms for describing reality and knowledge and agreements restricting the interpretations of these terms. The ontology of a domain defines the structure of knowledge and the structure of domain reality.

The problems in the "Program optimization" domain are mainly grouping around the unification of the notion system for describing program schemes in terms of which one could describe optimizing transformations and standardization of notion system for describing optimizing transformations. The other set of problems has to do with how to effectively use the accumulated knowledge in the problematic area, i.e. is connected with the special training in program optimization, the development of skills of putting theoretical knowledge about program optimization into practice.

The ontology model of the knowledge domain "Sequential program optimization" and its using when developing computer knowledge banks on program optimization is presented in this work.

The ontology model of the "sequential program optimization" domain

The formal description of the terminology of a knowledge domain together with definition of meanings of terms is called "ontology model" [Kleshchev, 2001]. The terms of the knowledge domain "Program optimization (classical optimizing transformations)" can be divided into two groups: (i) the terms for describing programs (the terms of this group will be called the terms for describing the optimization objects), and (ii) the terms for describing the optimization process. Therefore the ontology model of this knowledge also consists of two parts.

The optimization object is a program. The characteristics of a program are always formulated in terms of a mathematical model of this program. The characteristic of a program is the language (a set of programs) this program belongs to. The characteristics of the language are also formulated in terms of a mathematical model of this language. Thus, a number of terms for describing the object of optimization can be divided into two groups: (i) the terms for describing the language model, (ii) the terms for describing the program model. Before optimizing a program, the language, this program belongs to, must be determined. Consequently, the terms for describing the language model are parameters of the ontology model, and the terms for describing the program model are the unknowns of the ontology model. Then the language model is represented by the values of the parameters, and the program model – by the values of the unknowns.

It is evident that the program model describes not one program but a set of programs that have the same characteristics; the language model describes a set of languages that have the same characteristics. Therefore the following requirements are set on the language model: it must allow to present basic constructs of imperative programming languages essential for describing sequential OTs; it must be flexible and extensible in order to expand the class of modeled programs, if necessary; the form of presenting program models must be convenient for analyzing both information flows and control flows in the program.

The ontology model of the knowledge domain "Sequential program optimization" consists of two modules. The first module contains the terms for describing the optimization object, the second one – the terms for describing the optimization process.

The first module is an unenriched system of logical relationship with parameters written in sentences of a many-sorted language of the applied logic. Any program consists of fragments that in their turn consist of other fragments [Kasyanov, 1988] [Knyazeva, 1999], i.e. any program has its syntactic structure that is reflected by a mathematical model of this program.

Each fragment – as an element of the program – has a number of characteristics. First of all, it has *the address of the fragment* – the unique characteristic unambiguously defining each fragment in the program.

All the fragments can be divided into three groups: declarative statements, imperative statements, and entries of statements. Its class characterizes each fragment, e.g. the fragment can have the class of assignment statement, iteration statement, declarative statements of functions, etc. The function *FragClass* returns the class of this fragment for the indicated address of the fragment. A set of names of fragment classes (of each group) in the program assigns the values of the parameters D*eclarative statements, Imperative statements, Entries of statements.*

Control arcs assign the syntactic structure of the program. Each control arc connects two fragments of the program. Each control arc has its label identifying a type of connection between the fragments. The function the name of which coincides with the arc label is used to assign the control arc. The value of the parameter *Names of control arcs* assigns what labels can be owned by control arcs in the program. The control arc area is defined for each control arc. This area is assigned by the value of the functional parameter *Control arc area* – a function that matches each arc label with a pair consisting of two sets of names of fragment classes: the first set determines what fragments classes can be arguments, the second one determines what fragments classes can be results of the control arc with this label.

There are always a number of various *Identifiers* in the program. Identifiers can be of different types, e.g. identifiers of variable, functions, constants, and data types. In the program identifiers of each type make a set the name of which coincides with the name of the type. The value of the parameter *Types of identifiers* assigns what types of identifiers can be present in the program.

Functions and relationships identifying the structure of the program and some of its characteristics are defined on a set of fragments and identifiers of the program.

Functions with one argument (a fragment or an identifier) are called attributes. Each attribute has its name. The value of the parameter *Names of attributes* assign what attributes can be used for describing the characteristics of identifiers or fragments in the program. The definitional domain and the range of attribute values are established for each attribute. They are assigned by the values of the functional parameters *Definitional domain* and *Range of attribute value.*

The value of the parameter *Names of functions* assigns what functions with two or more arguments can be used for describing the characteristics of fragments or identifiers of the program. The definitional domain and the range of values are established for each function. They are assigned by the values of the parameters *Definitional function domain* and *Range of function values.*

The value of the parameter *Names of relationships* assigns what relationships can be defined on a set of fragments and identifiers of the program. The values of the parameters *Determination of Relationship* for each relationship name assign a formula of truth determination of a relationship between fragments and identifiers of the program.

Correctness is another characteristic of the fragment. The value of correctness is assigned by the predicate *Correctness* that matches the address of the fragment with the truth if it has all the control arcs and attributes necessary for this fragment. The value of the parameter *Determination of correctness* assigns control arcs and attributes for each class of fragments.

The value of the parameter *Elementary types* assigns for the language a set of identifiers of data types that are basic for all the rest data types of this language.

The value of the parameter *Modes of generation* assigns a set of names of constructors of data types in the language. The values of such characteristics as a base type and the method of constructing a type from the base one are set for each identifier of data type in the program. The value of the first characteristic is assigned by the function *BaseType* that matches each identifier of the type with a chain of identifiers of types that are used when constructing this data type. The value of the second characteristic is assigned by the function *ConstructMethod* that matches each identifier of the type with the name of the constructor. The value of the parameter *Compatibility of types* is a predicate defining the possibility of implicit transformation of one type to another.

The value of the parameter *Reserved names* assigns a set of names of roles that can be played by fragments or identifiers of the program. The value of the parameter *Area of reserved name value* matches each name of a role with predicate defining proprieties for a fragment or an identifier playing this role in the program.

The value of the parameter *Operators* assigns a set of symbols used in the program for identifying operations in expressions (arithmetic, logical, of a transformation, etc.) of the program. The functional parameters *Arity* and *Priority* match each symbol of the operation with a number of arguments and priority.

Optimization is understood as a chain of steps at each of them one transformation is applied to an optimized program. Each step will be called a step of the optimization history. The program model written in terms of the language model is the optimization object. The program model changes at each step of optimization: the rule of transformation that is established by the optimizing transformation that is used at the current step of optimization is applied to it. Thus, at each step of optimization there is its own version of the program model with its set of fragment addresses, its set of identifiers, etc. Therefore all the terms defined in the work [Artemjeva, 2002] are functions the first argument of which unambiguously identifies the current program model (i.e. the current set of fragment addresses, the current set of identifiers, etc.). A number of a step of the optimization history plays a role of this argument.

Let us define the terms for describing optimization process. Each optimizing transformation has the following characteristics: the saving block, the context condition, the indicative function, and the application strategy.

Normally an optimizing transformation (OT) is not applied to the whole program but to one of its blocks not necessarily continuous. The set of program fragments necessary for making a decision about optimization will be called candidate for saving blocks and the set of fragments where the context condition is true will be called saving block. Thus the saving block is a candidate for which the context condition of an optimizing transformation is true.

An ordinary saving block is a fixed set of program fragments for which the number of fragments, their types and their mutual location in the program are known. However the saving block consists of two parts. The first part is a tuple of fragments where each element's type and location in the saving block are known.

The second part is a set of tuples of fragments. The number of tuples belonging to the second part of the saving block is variable for different saving blocks of one program but the number of elements of each tuple, types and mutual location of fragments included in the chain are fixed, i.e. these tuples of fragments own certain identical characteristics. When applying an optimizing transformation, all the tuples of fragments of the second part of the saving block change the same way. For example, the saving block contains both declarative statement of a procedure and all its invocations for certain optimizing transformations of procedures (functions). The declarative statement of a procedure is always single but invocation operators can be numerous. All the elements of the set of invocations are assigned by declarative statement of a procedure. Thus, after applying an optimizing transformation, all operators of invocation of a procedure (function) change the same way.

The saving block in the model is represented as a pair the first element of which will be called the simple part of the saving block; the second one will be called the multiple part of the saving block. The first element of the pair is a tuple of addresses of fragments. The second element of the pair is a set of tuples of addresses of fragments. For the example given above, fragments of declarative statement of a function make the simple part of the saving block and a set of tuples of fragments of invocation operators makes the multiple part of the saving block.

Each optimizing transformation has simple and multiple parameters. Each fragment of the simple part of the saving block will be called the value of the simple parameter of the saving block corresponding to it.

The set of fragments included in the set of tuples of fragments of the multiple part of the saving block will be called the value of multiple parameter of the saving block. The function *Number of simple parameters of OT* assigns the number of elements of the tuple of the simple part of the saving block. The function *Number of multiple parameters of OT* assigns the number of elements of each chain of the multiple part of the saving block. The functions *Classes of simple parameters of OT* and *Classes of multiple parameters of OT* assign chains of classes of fragments forming the simple part of the saving block and chains of classes of fragments forming each tuple of the multiple part of the saving block.

The context condition of the optimizing transformation describes the characteristics of the saving block this optimizing transformation is applied to. In this ontology model the context condition is presented as a predicate the arguments of which are the number of a step of optimization history and a candidate to saving blocks.

There can be several saving blocks in each step of the optimization process. Therefore it is necessary to have a criterion to choose one block form many. *Indicative function* (IF) is a formula the arguments of which are fragments from the saving block. It matches each saving block with its estimate – the rational number.

Optimization strategy is a formula helping to choose one number from the set of rational numbers – results of indicative function for various saving blocks. Correspondingly the saving block with the chosen characteristic undergoes optimization on this step of history.

Parameters with the identical context are a set of pairs of numbers of parameters of the saving block before and after optimization contexts of which must coincide.

The chain of application of OT assigns the order of OT application. Each OT is applied until there are valid saving blocks left for it.

All the terms defined above are considered as terms for describing knowledge of program optimization, whereas optimization strategy, indicative function and chain of application of OT allow assigning parameters of optimization; context conditions and transformation formulas assign statistic knowledge about optimizing transformations.

Let us further define terms for describing situations of the knowledge domain. These terms are mainly meant for complete recording of the modeling process of application of optimizing transformations to the program.

The situation consists of a chain of optimization steps (history steps). Sets of fragment addresses, identifiers, and values of all attributes, functions and relations in the program are defined for each step. Some fragments, attributes, arcs and identifiers are defined while constructing the program model and known on the first step of optimization but some are recomputed before the beginning of each next step with the help of a special function *Enrichment of SMP*. The work of this function leads to that all the DSCH class fragments are added to the program model, Begin, End, Parent, SCH arcs are defined for them, all relations and functions are defined, values of all the attributes from a set of *Computable* attributes are also computed.

At each history step there is *a chain of current OT*. At the beginning of optimization a chain of *Current OTs* coincides with a chain of OT application.

The first OT from a chain of current ones for which a set of candidates to the SB is not empty is called *applied OT* and its number in the application chain is written in *Number of Applied OT*. A set of estimates is formed for all candidates to SB for applied OT with the help of indicative function and one estimate is selected from this set and with the help of optimization strategy and becomes the *characteristic of chosen SB*. The saving block corresponding to this characteristic becomes *chosen SB*. Apart from this, *optimized saving block* is defined on each step – it is a combination of fragments from next history step that becomes true when permutated to the transformation formula together with chosen SB. Optimized block appears as a result of OT application to the chosen saving block.

As a result of an applied optimizing transformation a number of fragments of the source saving block can be deleted, a number of fragments can be changed, new fragments (with new addresses)can be added, existing identifiers can be deleted or new ones can be added. For each optimizing transformation it is known how many elements will be included in the simple part of optimized saving block and how many elements will be included in each element of the multiple part of the saving block. This information is assigned by functions *Number of elements of the simple part of optimized SB* and *Number of elements of multiple part of optimized SB.*

Transformation formula is a predicate the arguments of which are two saving blocks from two consequent history steps; this predicate is true if the second SB is the result of OT application to the first SB.

Tasks given in terms of ontology model

As it follows from the previous section, in the ontology model there are groups of parameters defining Programming Language (PL), Optimizing Transformation Description Language (OTDL), Optimizing Transformations (OT); Optimization Strategy and groups of unknowns defining program characteristics before optimization, complete protocol of optimization process and characteristics of optimized program. Besides it is necessary to enter parameter Estimating function. This function will allow comparing various programs and, thus, to estimate optimization results.

Let us mention main classes of tasks that can be specified in terms of ontology model:

1. given PL, OTDL, an optimizing transformation and a program, required to get an optimized program and check the correctness of OT application;
2. given a program, PL and OTDL, strategy, estimating function, required to get an optimized program, estimate its optimality, check the correctness of OT, Strategy, analyze protocol;

3. given a set of programs, PL and OTDL, strategy and estimating function, required to get a set of optimized programs and estimates of their optimality, study the dependence of program optimality estimate on its characteristics;
4. given a set of programs, PL and OTDL, a set of strategies, estimating function, required to get a set of optimized programs and their optimality estimates, study the dependence of program optimality estimate on its characteristics and applied strategy;
5. given a set of programs, PL and OTDL, a set of strategies and estimating functions, required to get a set of optimized programs and their optimality estimates, study the dependence of program optimality estimate on its characteristics and applied strategy for various estimating functions;
6. given a set of programs, PL and OTDL, a set of strategies, an optimization criterion, required to get a set of optimized programs and their optimality estimates, study the dependence of program optimality estimate on its characteristics, applied strategy and programming language.

From the list given above one can see that all tasks are special cases of one task: given a set of PL, OTDL, input programs on these PL, and also a set of optimizing transformations, strategies of their application and functions – optimality estimates, required to build a set of optimized programs, protocols of their optimization and a set of optimality estimates for all programs on each step of optimization.

Method of solving the task of modelling optimization process

It is obvious that solving any task from the given above comes to solving the following task: with PL, OTDL fixed, the only input program, a set of OT, strategy of their application and function – optimality estimate are defined, required to build an optimized program, get the protocol of the optimization process and optimality estimate.

The algorithm to solve the task is given below.

```
BEGIN
Step of history=1;
Current OT (Step of history)= Chain of OT application;
Number of applied OT(Step of history)=1
Last step=False
REPEAT
   First OT application=True
   To analyze SMP(Step of history)
REPEAT
   IF Not First OT application
      THEN Number of applied OT(Step of history)= Number of applied OT(Step of history)+1
   Applied OT= Chain of OT application[Number of applied OT (Step of history)]
   Candidates to SB(Step of history)=To find saving blocks(Step of history, Applied OT)
   First application =False;
UNTIL (Candidates to SB(Step of history)≠∅) or
       (Number of applied OT (Step history)=Length(Chain of OT application))
IF  Candidates to SB(Step of history)≠∅
   THEN
      FOR SB in Candidates to SB(Step of history)
         DO SB characteristic(Step of history, SB) =
            To build SB characteristics (Indicative function(Step of history, SB))
            Characteristic of chosen SB(Step of history)=To realize Strategy (Strategy of OT(characteristic
            of SB(Step of history)))
            Chosen SB=To choose SB (Step of history, Characteristic of chosen SB(Step of history))
            Optimized SB(Step of history)=To build optimized SB(Step of history, Chosen SB, Applied
            OT)
```

To build new SMP(Step of history+1), where exists the only Optimized SB (Step of history) where Transformation Formula (Step of history, chosen SB(Step of history), Step of history+1, Optimized SB(Step of history)), and the rest context coincides.
Step of history= Step of history+1

ELSE Last step=True
UNTIL Last step=True
Number of Optimization Steps= Step of history
END.

This algorithm is simple and obvious enough to serve as a kernel of instrumental system for program optimization. However, the functions applied in it: **To analyze SMP, To find saving blocks, to build characteristics of SB, To realize strategy, To build optimized SB and To build new SMP** are not that obvious and can be considered as separate subtasks.

The structure of intelligent system on program optimization

The developed ontology, the tasks given in it and proposed methods of solving them provide the basis for the instrumental system for program optimization. Instrumental modeling expert system of program optimization (I_MESPO) is intended to support teaching of classical optimizing transformations.

This system allows the user to describe optimizing transformations, to set their application conditions and transformation rules, to form various sets of optimizing transformations, to assign chains, to trace the program optimization history.

The input data of the system are optimizing transformations knowledge, testing program on an algorithmic high-level language.

The result of the work of the system is the protocol of the optimization history protocol where for each optimization step it is shown what transformation has been applied, what saving blocks have been found on this step, which block has been chosen and what it has been replaced by.

Since this task is connected with the complicated processing of the knowledge given and generation of new one, the given subsystem was done as an expert system that models program optimization process.

The expert system includes a subsystem of visual input of knowledge about program optimization, knowledge base description language translator (Synthesizer), high-level language translator in the model of structured programs, estimating and result visualizing subsystem, integrated shell providing interface among these subsystems.

The work with the system I_MESPO begins with that the researcher defines the system of optimizing transformations, i.e.: assigns a list of OTs, a chain of their applications, and defines context conditions for each OT, transformation formulas, indicative functions and optimization strategies. After assigning the values of these parameters, Synthesizer executes translating knowledge base about program optimization into implemented module, thus creating application expert system (AES).

The functioning of the created application expert system begins with the work of a translator included in the system that transforms the structured program written in a high-level language into the structured program model (SPM). SPM is an internal form of relational presentation of programs that is convenient for analyzing and optimizing. According to this model, the application expert system makes an inference and builds up the optimization history protocol of this program. After receiving this protocol, the estimate and visualizing subsystem allows to estimate the program optimality before and after optimizing with the help of assigned estimating function and to compare the texts of the programs on each optimization step on high-level language and to analyze the implemented changes.

Conclusion and Acknowledgements:

Knowledge processing in the field of program optimization makes it possible to use this knowledge in industry, science and education.

Describing optimizing transformations within the terms of one model must facilitate the unification of different transformations within the framework of one system of OT. Thus, specialists can spend much less effort to study and use optimizing transformations to solve program optimization problems.

The use of the knowledge gives an opportunity to train highly qualified specialists to solve tasks on program optimization.

The access to knowledge through the Internet will attract all specialists interested in knowledge exchange on this problem.

References:

[Artemjeva, 2002] Artemjeva I.L., Knyazeva M.A., Kupnevich O.A. Domain ontology model for the domain "Sequential program optimization". Part 1. The terms for optimization object description. In The Scientific and Technical Information, 2002. (In Russian).

[Kasyanov, 1988] Kasyanov V. N. Optimizing transformations of the programs. Moskow: Nauka, 1988.

[Kleschev, 2001] Kleschev A.S., Orlov V.A. Requirements on a computer bank of knowledge. In Proceedings of the Pacific Asian Conference on Intelligent Systems 2001, Seoul, Korea, 2001.

[Kleshchev, 2002] Kleshchev A.S., Chernyakhovskaya M. Yu. The present state of computer knowledge processing. http://www.dvo.ru/iacp/es/publ/kpe.htm

[Knyazeva, 1999] Knyazeva M.A., Kupnevich O.A. Expert system for simulation of program optimization. Joint NCC&IIS Bull., Comp. Science,12 (1999), 24-28.

Author information:

Irene L. Artemjeva: artemeva@iacp.dvo.ru
Margarita A. Knyazeva: mak@nt.pin.dvgu.ru
Oleg A. Kupnevich: garfield1@yandex.ru
Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences
5 Radio Street, Vladivostok, Russia

A KNOWLEDGE-ORIENTED TECHNOLOGY OF SYSTEM-OBJECTIVE ANALYSIS AND MODELLING OF BUSINESS-SYSTEMS

M. Bondarenko, V. Matorin, S. Matorin, N. Slipchenko, E. Solovyova

***Abstract**: A new original method and CASE-tool of system analysis and modelling are represented. They are for the first time consistent with the requirements of object-oriented technology of informational systems design. They essentially facilitate the construction of organisational systems models and increase the quality of the organisational designing and basic technological processes of object application developing.*

***Keywords**: Knowledge, systemology, natural classification, objects modelling, conceptual knowledge.*

The civilization sustainable development is based on formation of the informational society as a first stage of the noosphere. At the same time, as the transition to the informational society, as economic activity in it become based on knowledge. This knowledge represents the "**informational resource**". It directly influences at the material factors of progress and ensures the "phase transition of knowledge into a power", i.e. efficiency of business, production and any administrative solutions.

The submission about the tendency of knowledge-oriented development of an alive nature is entered into scientific practice by V.I. Vernadskiy under a title "*the Dan's principle*". The knowledge-oriented development should be considered as the universal tendency enveloping not only biological, but also all other complicated systems. The social (organizational) and information systems also develop ***in a direction of increasing of a knowledge role for their sustainable functioning***.

This tendency is exhibited in the unprecedented growth of knowledge and scientific information; increasing of a role of inclusive, depth knowledge; rapid development of methods and means of knowledge processing, analytical activity, acute need of the appropriate experts, influence of informational resources to all sides of the human activity. The technologies and methods of purchase, extraction, submission, processing of knowledge (knowledge management, knowledge engineering) in substantial aspect also develop in the knowledge-oriented direction (data mining, text mining, knowledge discovery, knowledge mining, object modelling, ontological engineering). In the foreign expert's opinion, the development of these directions is broken by absence of the effective methodologies by the availability of developed technologies.

Accumulated in the given spheres experience and potential even more acutely shows the necessity of the account of depth knowledge, objective factors, system and simultaneously object approach to modelling of complicated systems. Grew the role of the veritable human's resource – "conceptual knowledge", which becomes the core of the informational resources, knowledge bases, and ontology models. In such new spheres of the scientific-practical activity as, for example, business process reengineering, decision support making, object paradigm has appeared the similar necessity, which was already expressed in expediency of the organization's mission definition, context account, systems analysis first of all from the point of view of their functionality, correspondence to requests more high level.

The systemology [1] can become the unique scientific basis of such researches. Systemology is a system approach of the new noospheric stage of science development, which comes to change the differentiation of sciences in analytical paradigm - second in the whole history of science after antique stage.

Systemology allows to work successfully with the complicated systems of the first nature i.e. not human created, and with open systems. At the same time, in difference from other system approaches, is ensured the possibility to consider as a system not only objects, but also classes of objects (systems-classes). The development of the systemology of systems-classes has allowed us to synthesize the system and classification analysis for a solution of problems of conceptual modelling of the low formalized problem areas [2, 3].

Systemology most objectively allows getting the next things for the complicated systems of any nature and with any minuteness:

- to understand the reasons of origin, dynamics of becoming and development;
- to define the influence to other systems;
- to explain the outcomes of adaptation and interaction;
- to predict development in various conditions;
- to make conclusions about necessary measures of stable development;
- to prevent crisis situations and to reduce risks;
- to take into account the main properties and priorities.

Systemology really takes into account system effect, i.e. for the first time considers the system as a qualitatively new essence, instead of reduces it to the sum of component parts. It is ensured owing to the consideration of the system for the first time as:

- integral object, instead of as a set;
- the main properties of a system are explained proceeding from properties of a super system;
- the system is considered as functional object;
- "substation" of a system is taken into account, i.e. "material" from which it is made;
- the shaping and operation of a system of any level is considered from "above" and is determined by the "request" of its super system.

Systemology represents an exposition, oriented on methodological use, of concepts and principles of dialectics, which can be interpreted in terms of any concrete science. Besides, it is a unique system approach, which is agreed at a conceptual level not only with formal logic, object-oriented ideology, but also with a complex of modern scientific-practical disciplines engaging the problems of studying and perfecting of organizational systems (the theory of organization, logistics, and business engineering).

Systemological methods can be applied in cognitive direction of researches, which is major component of knowledge-oriented technologies. It is connected first with the orientation on human is now most necessary for maintenance of harmonic interaction of computer systems with the human.

The development and application of systemology in scientific-educational Knowledge acquisition laboratory (NUL PZ) with the cognitive methods has allowed to decide the fundamental, delivered more than 150 years back, problem of a "natural classification" (NK), to reveal and to formalize its regularities and criterions [3]. NK (systematization) as the ideal of a classification is considered as a privileged system chosen by nature, takes into account the essential properties and relations of objects, the maximum amount of the purposes and can form the basis of the most objective and reality adequate models of knowledge. The features of such classification were studied by many scientists, because it has the greatest value, cognitive and prognostic force and makes a basis of a scientific picture of the universe, but only with the help of systemological approach we succeed in opening its laws. The rules of NK can be taken into account in any problem area and allows creating the effective methods of knowledge systematization and conceptual classification modelling (systemological classification analysis).

The methods of system analysis and instrumental program CASE-tools of their supporting are widely used at the present time for decision of business, administrative and production problems. However, methods and means of traditional system-structural analysis (SADT, DFD, BPwin, etc.), that are used for business-processes modelling, are historically based on procedure-oriented programming paradigm. Therefore the results of their application can't be immediately used during the developing of object-oriented software.

The most of modern program systems, especially large, at present time are created namely within the frameworks of object-oriented approach. However, the object-oriented analysis (OOA) and language UML are primordial used for software developing. Therefore, they are badly adapted for solution of the problems of business analysis and modelling. At the same time, such problems obligatorily arise, especially during the creation of complex program applications. And what is more, a standard process of object-oriented software developing (Rational Unified Process - RUP) begins with the technological process of business modelling.

A given discrepant situation stipulates the actuality of system and object-oriented methodologies integration. The researches in this direction, carried in NUL PZ, allowed to work up a new original **system-object (systemological) approach and** object-oriented **systemological methodology of analysis and designing (OMSAD)** [4, 5], permitting the marked contradiction. Analytic methods and instrumental means of such approach allow automating the considerable part of analytic work and essentially raising its effectiveness.

Let us consider the basic peculiarities of system-object approach and systemological methodology, and also procedures and possibilities of the new method of system analysis, for the first time consistent with the requirements of object-oriented design.

The traditional system approach (analysis) is peculiar to the *procedure* (functional) system decomposition, and object approach – is peculiar to the *object* system decomposition. At the same time all specialists, as of system analysis, as of object approach consider them as orthogonal. In it's turn, system-object (systemological) approach allows to combine exposure processes of the functional and object structure of analysed system. Thus, the basic peculiarity of the given approach is providing the unity of the decomposition of analysed system, as on functional, as on objective (substantive) sign. This reaches due to the consideration of any system not as a set, but as a *functional «flowing» object [1, 2].* Acknowledgement the status of such an object after the system provides a simultaneous calculation of structural, functional and substantial system existence aspects.

To begin with, any system is a component part of the system structure of higher level (super system), because any system is connected and co-operates with other systems. Herewith any link between systems is the process of mutual *exchange* of elements of definite deep layers of connected systems. Thus, a feature of system is understood as manifestation of it's activity to be included into links, into *exchange flows* with other systems in the super system structure. Consequently, from structural point of view a system is a crossroad of incoming and outgoing links (streams), i.e. unit (node).

Secondly, the functioning (activity, work, behaviour) of any system provides or supports the functioning of the super system, to which this support is necessary. At the same time functioning of the system as a support of the functional ability of the super system consists in the providing of the balance of "influx" and "outflow" on the incoming and outgoing links. Consequently, from the functional point of view the system is a function, which provides a balance of incoming into the system and outgoing from the system streams in accordance to that unit, where this system is in the present moment.

Thirdly, any system is not only a unit and function, but also a substance, which plays a role of definite unit in the structure of the super system and provides its functional balance. Consequently, from the substantial point of view a system is an object, realising a function, set by a unit in the structure of the super system.

Given reasoning allows the representing of any system in appearance of the three elements construction – UFO-element (figure 1) [6], i.e. at the same time:

- as the structural element of the super system – unit, as a crossroad of the relations with the other systems;
- as the functional element, doing a definite role for supporting super system by balancing the given unit – function;
- as the substantial element – object, realising the given function in the appearance of some material formation, having constructive, operational and other characteristics.

The basic peculiarity of the OMSAD methodology is the **formal-semantic adaptive alphabet** of the UFO-elements, and also **categorical principle** that is used during the analysis and designing of systems. Alphabet is a collection of units (crossroads of system links), collection of functions, balancing these units, and collection of objects, realising these functions. At the same time, we use facet classification for units collection, defined by the taxonomic categorical classification of the kinds of system's links (figure 2).

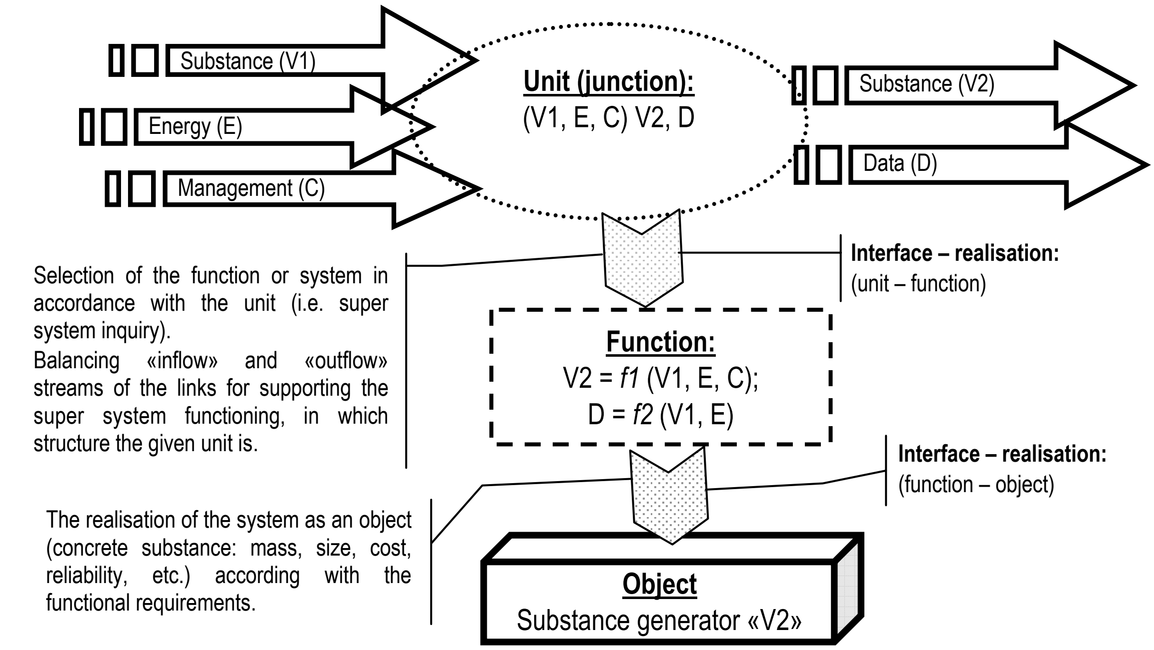


Figure 1. «Unit – Function – Object» approach.

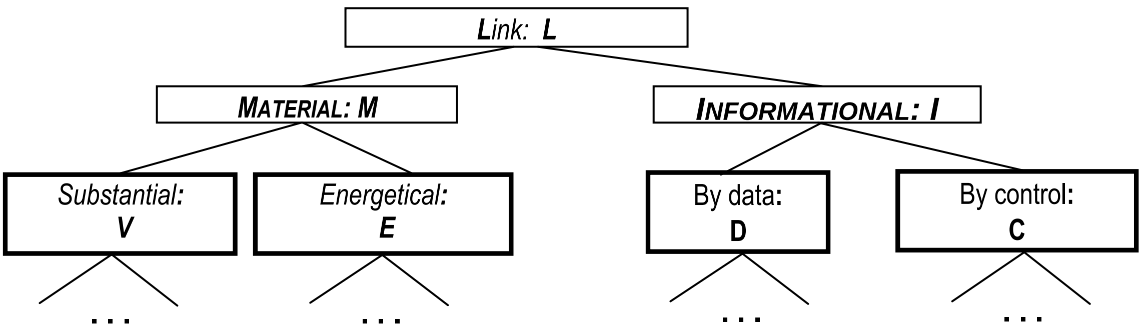


Figure 2. Basic taxonomic classification of system links.

The links classification provides the parametric units classification and constructive determination of symbol semantics of these units. Naturally, the links and units classifications (functions and objects) can be specialised with any degree of accuracy for any concrete domain. The use of classifications for forming the alphabetical collection of the UFO-elements and the possibility of their specialisation turns this collection into the formal-semantic adaptive alphabet.

Parametrical taxonomic classification of UFO-elements represents a classification, in which the objects are systematized depending on functions, which they are realizing, function - depending on what units they are balancing, and the units are determined by that, what crossroad of link they are. This is a conceptual model of application domain in the terms of knots, functions and objects witch plays a role of a " categorical" grid, through which the analyst looks at the domain. The specialization of such a categorical classification model should be carried out in the correspondence with the recommendations of the *systemological classification analysis* offered in the work [3] and directed on the construction of the classifications, witch takes into account the properties and regularities of the natural classification. In the correspondence with these recommendations during the construction and specialization of the classification the good, natural classification will be obtained, if the definite sequence of operations mentioned below is observed.

Any units got by combining of the links from the classification can be considered as alphabetical elements. However with practical point of view it's expediently to consider not all of the possible combinations, but only such ones, which corresponds to the actual physical laws (for example, to preservation laws). Point is that energy does not exist without any material bearer, information does not exist without any material bearer and administration does not exist without any data transmission. This leads to the relatively small number of variants on the level of the links of the base classification (figure 2 In the given tables we use brief markings for data on the material (VD = D) and power (VED = G) bearers, and for control data on the material (VDC = C) and power (VEDC = Q) bearers. This determines that, at present time, only paper (D and C) and electronic (G and Q) information bearers have the wide diffusion.

The use of alphabet (libraries) of UFO-elements allows formulating the combining rules of these elements naturally following from the systemological approach, for constructing UFO-configurations. We offer to call these rules the **rules of system decomposition:**

1. The rule of association: elements should be linked together according to the qualitative and quantitative characteristics of links inherent in them;
2. The rule of balance: during the connection of elements to each other (according rule 1) the qualitative end quantitative balance of the inflow and outflow of input and output functional links must be observed at units of the system structure;
3. The rule of realisation: during the connection of elements to each other (according to the rules 1 and 2) the interfaces accordance and the accordance of the objective and functional characteristics must be observed;
4. The rule of closeness: internal (supporting) links (streams) of elements in system must be reserved.

Offered alphabet and named rules forms **a formal-semantic normative system** of systemological analysis and modelling, formalising by the pattern theory of Grenander funds.

Table 1

		Entries:					Exits:		
		Produc-tion	Providing			**Adminis-trator**	**Product**	**Informa-tional**	**Wastes**
			Substan-tial	**Energetic**	**Informa-tional**				
Business system		V, E, D(G), C(Q)	V	E	D(G)	C(Q)	V, E, D(G), C(Q)	D(G)	V, E
Production	Substance	V_{in}	V_{pr}	E_{pr}	$D(G)_{pr}$	$C(Q)_{pr}$	V_{out}	$D(G)_{out}$	V_{wst}, E_{wst}
	Energy	V_{in}, E_{in}	–"–	–"–	–"–	–"–	E_{out}	–"–	–"–
	Information	$D(G)_{in}$	–"–	–"–	–"–	–"–	$D(G)_{out}$ $C(Q)_{out}$	–"–	V_{wst}
Transport	Substance	V	–"–	–"–	–"–	–"–	V	–"–	–"–
	Energy	E	–"–	–"–	–"–	–"–	E	–"–	V_{wst}, E_{wst}
	Information	D(G), C(Q)	–"–	–"–	–"–	–"–	D(G), C(Q)	–"–	V_{wst}
Allocation	Substance	V	–"–	–"–	–"–	–"–	V	–"–	–"–
	Information	D(G)	–"–	–"–	–"–	–"–	D(G)	–"–	–"–

Besides, OMSAD methodology is using a categorical principle during the construction of models. This principle postulates the necessity of prior assignment (definition) of the synthesised (designed) systems from categorical classification of such systems. Named principle, in fact, in the obvious form fixes the common sense used in the practical analytic work. The point is that decomposition (analysis) and aggregation (synthesis) procedures can be successfully realised only in that case, if they are directed by the final result. During the realisation of synthesis operation, it's necessary to know something at least about the kind of synthesised system, and during the realisation of analysis operation it's necessary to know something about the types of the parts, on which analysed system can be decomposed. Thus, mentioned above alphabet is a realisation of the categorical principle from the point of view of system analysis procedure. For solving the modelling and designing problems of organisational systems OMSAD methodology is using the systems categories, represented in the table 1.

The experience of the practical using of systemological methodology showed, that context model of any organisation (business-systems), and also of any of it's subdivision, can be represented as unit from the table 1. For example, workshop, model and tool shops, naturally, are represented as the systems of the material production. Department of main constructor, office of the production technical training, economic planning department, accountancy, labour and salary department, marketing department, etc. is represented as the systems of information production. Department of technical control, department of main mechanic, department of main technologist, provision department, sales department, department of technical documentation, etc. is represented as distributive systems.

Formal-semantic normative analysis and modelling system and business-system categories can be considered, in particular, as the development and addition, for example, of the popular technology SADT. As is well known, this technology grants the formal universal possibilities on constructing of the functional business-processes structures. However it doesn't take into account the semantics of the domain and does not give to the analytic the information about the concrete interactions between the analysed systems and their possible filling. So, a context modelling and systems decomposition with the SADT funds are heuristic procedures and don't have any support with the proper CASE-tools (for example BPwin) on the substantial level.

Systemological approach «Unit – Function – Object» and OMSAD methodology allowed to work up a new method of business-systems analysis and modelling (**UFO-analysis**), which allows to adapt it's funds to the concrete data domain, i.e. to take into account it's semantics [6, 7]. Besides, the systems representation with this method as configurations of UFO-elements provides the concordance of the derivable models with the requirements of object-oriented design.

Briefly, the following main steps can represent UFO-analysis procedures:

- Revealing units links in the structure of the modelling (designing) system based on functional links of the system as a whole, defining by the customer or solving problem;
- Revealing of functionality supporting (providing, balancing) found units;
- Determining objects, corresponding to the revealed functionality, i.e. those realising it.

The specific peculiarity of this analysis method is providing automation possibility of these steps. Automation reaches due to using of formally semantic adaptive alphabet. At the same time it's necessary to take into account prepared beforehand classification of UFO-elements (UFO-library), which contains suitable elements for the given problem (data domain). In this case the first step may be identified with the system analysis stage, the second - with it's design, and the third - with it's implementation.

For automated application of UFO-analysis method we developed a program complex «UFO-toolkit», which is the CASE-tool, using knowledge base of the special configuration for providing a component approach to modelling, using semantics of domain and intellectualisation of interaction with user [7]. The tool is intended for object and simulation models construction of complex dynamic (organisational) systems. It has the following features:

- noticeably reduces the designing labour-output ratio owing to intensified automation of analytic activity;
- increases the objectivity of the analysis and the adequacy of modelling;
- automates a models creation process, through the use of ready (alphabetical, library) functional objects, presented in the knowledge base of the Tool in the form of UFO-elements;
- provides «intelligent» interaction with user, making familiar the ready component (UFO-elements).

At the same time if alphabetical elements appear to be program objects, realised as ready classes, then we can talk about UFO-analysis as a part of component technologies and business-objects technology CORBA (Business Object Facility – BOF). In the last situation the program CASE-tool, automating UFO-analysis procedures, can function within the frameworks of component business-objects architecture (Business Object Component Architecture – BOCA). At the same time, it will carry out an organiser (Framework) role, which, integrating business-objects into the functioning system, gives them the working places for realising their tasks. If we consider the alphabetical elements as the engineering elements, then UFO-analysis will be confirmed with the CALS-technology.

Thus, UFO-analysis method represents the development and concrete definition of the OMSAD methodology. It allows to use the formalised rules of revealing classes and objects of application domain during the OOA process and to realise the system analysis of the events of different nature, considering them as functional flowing objects. Consequently, UFO-analysis can be considered as the method of system-objective analysis and modelling.

On the whole the considered method and Tool (UFO-technology) provide to user:

- objectivity of analysis and synthesis procedures of organisational systems;
- economy of the man-hours of analysis and modelling, because these procedures both comes to the construction of only one model;
- simplicity and availability of the business-processes analysis and modelling by specialists without special training;
- uniform presentation of external and internal models of the business-system, described by the same modelling language;
- facilitate of models adaptation to the concrete domain (taking into account the semantics domain);
- the possibility of creation and use the libraries (repository) of the model components for different application fields.

Besides, they have the following merits:

- provide the concordance of the system analysis results with requirements of object-oriented design, previously considering as orthogonal;
- provide a possibility of immediate use of the system analysis results during the creation of object-oriented software;
- raises the level of formality and automations of the modelling and analysis procedures;
- guarantee a concordance of all system characteristics due to unification of the different system consideration aspects in one model;
- provide facilitate of the construction of visual models of different abstraction level, representing at the same time a functional and objective structure of system;
- provide a possibility of the modelling of functional system characters, not having a mathematical interpretation or interpreted by any mathematical means, and also simulation of system functioning without any special modelling algorithm.

Represented analysis and modelling technology is used for correcting information-analytic business-systems accompaniment and provides an essential rise of the effectiveness of their activity. Developed method and program tool essentially facilitate the construction of organisational systems models and increase the quality of the organisational design and initial technological processes of developing object applications.

Bibliography

Melnikov G.P. Systemology and linguistic aspects of cybernetics. New York, Paris, Monreal, Tokyo, Melbourne. Gordon and Breach.- 1988.- 440 p.

Bondarenko M.F., Matorin S. I., Solovyova E. A. Analysis of systemological tools for conceptual modelling of application fields // automatic document and mathematical linguistics. New York: Allerton Press Inc. 1997. V. 30, No. 2. P. 33-45.

Соловьёва Е.А. Естественная классификация: системологические основания. Харьков: ХТУРЕ, 1999.222 С.

Matorin S.I. A new technology of system-object analysis and its application for business-systems modelling // EEJET № 1(1) 2003 P 15-20.

Matorin S.I. A New Method of Systemological Analysis Co-ordinated with the Object-Oriented Design Procedure. I // Cybernetics and Systems Analysis. Plenum Publishing Corporation, 2001. V. 37, No. 4. P. 562-572.

Matorin S.I. A New Method of Systemological Analysis Co-ordinated with the Object-Oriented Design Procedure. II // Cybernetics and Systems Analysis. Plenum Publishing Corporation, 2002. V. 38, No. 1. P. 100-109.

Маторин С.И. Анализ и моделирование бизнес-систем: системологическая объектно-ориентированная технология / Под ред. М.Ф. Бондаренко; Предисловие Э.В. Попова. (ISBN 966-659-049-2) Харьков: ХТУРЭ, 2002. 322 с.

Author information

Michael Bondarenko – Rector of Kharkov National University of Radioelectronics, Professor; **Vasiliy Matorin** - Kharkov National University of Radioelectronics, student ; **Sergei Matorin** - Kharkov National University of Radioelectronics, doctorant ; **Nikolay Slipchenko** - KHNURE, chief of scientific department, professor ; **Ekatherina Solovyova** – KHNURE, head of Knowledge Acquisition Laboratory, professor, 14 Lenin Avenew, Kharkov, Ukraine 61166, nulpz@kture.kharkov.ua

ANALOGICAL REASONING TECHNIQUES IN INTELLIGENT COUNTERTERRORISM SYSTEMS

A.B. Markman, D.A. Rachkovskij, I.S. Misuno, E.G. Revunova

***Abstract**: The paper develops a set of ideas and techniques supporting analogical reasoning throughout the life-cycle of terrorist acts. Implementation of these ideas and techniques can enhance the intellectual level of computer-based systems for a wide range of personnel dealing with various aspects of the problem of terrorism and its effects. The method combines techniques of structure-sensitive distributed representations in the framework of Associative-Projective Neural Networks, and knowledge obtained through the progress in analogical reasoning, in particular the Structure Mapping Theory. The impact of these analogical reasoning tools on the efforts to minimize the effects of terrorist acts on civilian population is expected by facilitating knowledge acquisition and formation of terrorism-related knowledge bases, as well as supporting the processes of analysis, decision making, and reasoning with those knowledge bases for users at various levels of expertise before, during, and after terrorist acts.*

***Keywords**: analogical reasoning, structure-mapping theory, associative-projective neural networks, knowledge bases, terrorism, terrorist acts, antiterrorism, counterterrorism, SMT, SME, APNN*

Introduction

Since September 2001, the world has awakened to a new danger - the threat of international terrorism. Combating terrorism has become a high priority in international cooperation. A key component of counterterrorism measures is going to be the creation of tools that can supplement human reasoning to handle the vast amount of data that is being generated by counterterrorism measures. Technologies from computer science will play an important role in endeavors ranging from intelligence, prevention, preparedness planning, response training - to crisis management, reaction, mitigation, and recovery (e.g. [SAIC]). Computer-based systems to assist expert and nonexpert personnel to reason about terrorist activities (henceforth referred to as terracts) will benefit from further development of such systems. The key to these enhancements, as we propose, is the integration of existing tools and the creation of new computer-based counterterrorism systems using analogical reasoning techniques. The resulting analogical processing tools working with data and knowledge bases (KB) of experience accumulated from known terracts will provide new capabilities to counterterrorism systems. This approach can be used to support reasoning, prediction, assessment, analysis, decision-making, problem solving, planning, response actions throughout life-cycle of terrorist incidents.

Ultimately, such systems should be able to reason and learn from examples in the area of terrorism and other complex real-wold domains in the same manner as humans, who are capable of accumulating knowledge and experience by assimilating examples, working through problems, and reusing examples to solve new problems. However, humans have well-known limitations of their abilities. In particular, human memory tends to preserve the gist of prior situations, without necessarily providing access to detailed descriptions of past situations. In addition, humans often do not work efficiently under stress and time pressure. Therefore, it would be useful to have a system that supplements human reasoning and addresses people's shortcomings. Furthermore, such a system must be compatible with human reasoning processes in order to facilitate integration of a computer system's recommendations with ideas of a human user.

Analogical reasoning is an excellent candidate for serving such a function. In analogy, one situation is viewed as similar to another on the basis of relationships among the objects and actors in the situation. Once two situations are seen to be similar, predictions can be generated by carrying over information from one domain to another. In this paper, we discuss Structure-Mapping Theory (SMT; [Gentner, 1983]), which is the most prominent theory of analogical reasoning in the cognitive science literature. This theory has been implemented in a number of computational models. We discuss two implementations of SMT, the Structure Mapping Engine (SME; [Falkenhainer et.al., 1989]) and Associative-Projective Neural Networks [Rachkovskij, 2001]. We explore how these implementations can be used to supplement human reasoning in the domain of counterterrorism.

Analogical vs case-based reasoning

Past experience provides an important source of information for reasoning about terracts. In particular, it is important to recognize when a new situation is sufficiently like some previous situation that some action ought to be taken.

One approach to the use of prior knowledge involves expert systems in which domain experts are asked about the rules they use to reason. Unfortunately, experts are rarely able to articulate complete and correct general-purpose rules that they use to reason. Also, expert systems that use the rules obtained from experts often have difficulty with new cases that are not an exact match to those for which the rules were constructed.

Two other approaches apply information about specific known episodes to new situations. The first is case-based reasoning (CBR). On this approach, a KB is developed that consists of descriptions of cases that are indexed by key aspects of the environment such as the goals that the case is designed to solve. New situations with the same index call up prior cases. The prior cases are then examined for their appropriateness by determining whether the differences between the new situation and the old one are critical. If the differences are not important, then the old case is tweaked to allow it to be applied to the new situation. CBR has been applied in a number of domains [Kolodner, 1993; Schank, Kass, & Reisbeck, 1994], but it has a number of limitations including:

- it cannot deal with complex structured knowledge. Representations of cases are usually unstructured sets of attributes, without relational information;
- it is not so flexible as human reasoning. The indexing scheme that allows new cases to be accessed based on prior cases needs hand-tailoring;
- it lacks robust, domain-independent algorithms for finding similarity and applying information from a previous case to generate advice specific to the current case.

Analogical reasoning is one of the most commonly encountered and vivid cognitive processes. That is why a lot of work is devoted to the development of theories and computational models of analogy. (For an introduction, see, e.g., [Gentner & Markman, 1995, 1997; Holyoak & Thagard, 1989; Hummel & Holyoak, 1997; Thagard et. al, 1990; Eliasmith & Thagard 2001] and references therein).

Analogical reasoning provides the flexibility to reason on-the-fly given a knowledge base of historical cases by supporting comparisons between known episodes and current conditions. Counterterrorism requires the ability to flexibly extend existing ideas to new situations, and so analogical reasoning is a good match for it.

The analogical reasoning account supports a number of important sub-processes:

- finding relevant episodes in the KB and comparing them taking structure into account, as in human reasoning [Gentner, Rattermann, & Forbus, 1993];
- inferring new information in order to generate predictions [Clement & Gentner, 1991; Markman, 1997];
- retrieving and adjusting associated information in order to create action plans [Keane, 1996];
- reasoning with, learning and generalizing from examples and applying knowledge appropriately to the case situation [Forbus et al., 1999];
- providing the ability to understand analogies and metaphors in natural language texts [Fauconnier, 1997].

The ways to overcome the limitations of CBR using analogical reasoning are investigated in large-scale DARPA projects, High Performance Knowledge Bases [HPKB] and Rapid Knowledge Formation [RKF], that deal with construction of and reasoning over a large KB. As evidenced by those programs, analogical reasoning tools can serve as a basis for new AI technologies, with such applications as, e.g., crisis management or getting expertise in weapons of mass destruction.

Thus, introducing analogical reasoning to KB and expert system enables:

- a more advanced, human-like reasoning about the world (as in the HPKB project);
- support for creation of KBs (as in the RKF project);
- help in human interaction with KB, using natural language, examples and generalization;
- overcoming some limitations of human reasoning (such as the limited span of working memory, inability to extract precise description of past situations, limitations of input and exchange information channels, "standard" human thinking that limit the possibilities of creativity),
- overcoming other human limitations (such as the inability to work efficiently and to make decisions under pressure, the need to discuss with somebody, the difficulty to take responsibility, vigilance loss, etc.).

The Structure Mapping Theory and its symbolic implementation

The most advanced and elaborated theory of analogical reasoning is Gentner's Structure-Mapping Theory (SMT) of analogy and similarity [Gentner, 1983], further developed in [Gentner & Markman 1995, 1997; Markman 1997, Markman & Gentner, 2000]. The theory explains analogy and similarity in terms of comparisons involving structured representations, not just lists of features.

In an analogy, a base domain, which is the one people typically know more about, is compared to a target domain, which is typically the new situation to be reasoned about. The base and target are represented using structured hierarchical representations consisting of entities (the objects in the domain), attributes (one-place predicates that describe objects) and relations (two- or more-place predicates that relate entities, attributes, and other relations. Finding an analogy between two domains involves finding overlapping relational structure between the domains.

Two domains have overlapping relational structure if their representations contain some identical relations (i.e., they share semantic similarity), if matching relations have matching arguments (i.e., they display parallel connectivity), and if each element in one domain matches to at most one element in the other domain (i.e., there is a one-to-one mapping). Considerable psychological evidence is consistent with the operation of these constraints in human analogical reasoning [Gentner & Markman, 1997].

Unlike CBR, analogical reasoning does not require that cases be indexed for later retrieval. Instead, access is mediated by retrieval algorithms that find base domains in memory that share similarity with a target probe. Retrieval involves two stages. First, the contents of memory are filtered to include only those domains that share substantial semantic similarity with the target. This stage does not require attention to the structure of the domain (i.e., to parallel connectivity or one-to-one mapping) and hence can be done in a computationally efficient manner. Furthermore, recent work suggests that the semantic similarity component of retrieval can be done using high-dimensional semantic space models of the lexicon, which eases the process of developing domain representations (Ramscar & Yarlett, 2003). Those domains that pass through the initial stage of retrieval are given a structural comparison to the target domain, and those domains with a good structural match are retrieved from memory and made available for further reasoning.

There is quite a bit of psychological support for the basic principles of SMT. For example, [Clement and Gentner, 1991] observed that people show a preference for systematicity in that the prefer analogies that have deeply connected relational structures to analogies that preserve only a limited set of disconnected matches. Furthermore, Markman (1997) demonstrated that one-to-one mapping is critical for the formation of analogies. In other work, Markman & Gentner [1997] have extended structure-mapping theory from analogies to comparisons in general, and have created a number of experimental methods for gathering evidence about the way people make comparisons. This theory provides the scientific basis for developing systems that approach the flexibility of human analogical reasoning, for analyzing and predicting of human behavior, for natural interaction with humans.

This psychological theory has been implemented in a number of computational models (e.g., Falkenhainer et al., 1989; Hummel & Holyoak, 1997; Keane, et. al., 1994). The most prominent model is the Structure Mapping Engine (SME) [Falkenhainer et. al.,1989], that has been used successfully in a number of state-of-the-art large scale AI (knowledge-related) projects, such as [HPKB; RKF]. SME is implemented as a symbolic model. The representations given to SME consist of predicate-argument representation structures that can be represented as directed acyclic graphs. Thus, the problem of finding an analogy between domains is computationally intractable in general, and so heuristics and limitations imposed by the SMT must be used to find analogical matches efficiently.

SME makes the analogy process computationally tractable by using a local-to-global match algorithm in which predicates in one domain are first matched to those in the other domain when they have identical semantics. After this initial sequence, matching predicates are assessed to determine whether they have matching arguments, and each set of matching predicates is checked to ensure that matches are one-to-one. This model requires manual construction of representations of items and structures based on them.

SME is also incremental. An initial correspondence between the base and target domains can be extended when more information is obtained about the domains. If analogy were only able to find correspondences between domains, it would be limited in its usefulness to counterterrorism situations. However, SME also generates candidate inferences by carrying over information from the base domain to the target when that information is connected to the correspondence between the domains and is structurally consistent with it.

The central mechanisms embodied in SME are all consistent with what is known about the way people process analogies.

SME is one tool that can be used for the development of counterterrorism tools. The symbolic algorithm can be set up to operate with a KB of prior terracts and other relevant information. A second technique for analogical mapping is the APNN architecture. We describe it in the next section.

Analogical reasoning with APNNs

Associative-Projective Neural Networks [Kussul,1992; Kussul et.al. 1991] are based on a scheme for sparse binary distributed representations of information. These "structure-sensitive" distributed representations take into account both semantic and structural aspects of similarity. This approach provides an opportunity to combine the advantages of connectionist networks (semantic sensitivity, parallelism) and symbolic representations (compositionality, systematicity).

Traditional distributed representations allow a natural representation and computation of gradual similarity and make an efficient use of representational resources. Similar items are represented by correlated codevectors where similarity can be estimated using the dot product. A large information capacity is provided by the possibility to represent exponentially many items by different codevectors of the same dimensionality. Distributed representations are robust and neurobiologically plausible. Also, they allow unsupervised learning of similar representations for similar items using such methods as Learned Vector-Space Models (Latent Semantic Analysis, Context Vectors, Random Indexing, etc. [Caid et. al., 1995; Kanerva et. al., 2000]).

However, it was thought that distributed representations cannot represent nested (recursive) structures because of superposing pairs of vectors would lead to the loss of information about the relationship between elements and their arguments, see [Rachkovskij & Kussul, 2001] for discussion and references). In APNNs, Holographic Reduced Representations (HRRs) [Plate 1995; 2000], Binary Spatter Codes (BSCs) [Kanerva 1996, 1998], the scheme of [Gayler & Wales, 1998], it has been possible to create on-the-fly (without any training) distributed representations of recursive compositional structures with codevectors of the same dimensionality for arbitrary (even novel, non-similar) items.

Complex structures are chunks of a small number of component (sub)structures. The codevectors for more complex structures are built from the set of component codevectors. Because each component may itself be a complex compositional (sub)structure, structures of arbitrary complexity can be represented. To preserve grouping of components in chunks of various compositional levels, binding by Context Dependent Thinning [Rachkovskij & Kussul, 2001] is used in APNNs for component codevectors of each chunk. This allows an on-the-fly construction of a composite bound codevector from its component codevectors.

APNNs encode items of any nesting level, elementary or compositional, are represented by large codevectors of the same dimensionality. The codevectors are binary (with 0 or 1 elements) and sparse (with small fraction of 1s), e.g., with N=100,000 elements representing neurons and M=1,000 of 1s. To build APNN representations of the episodes, it is necessary to encode the entities and relations using base-level codevectors of the lowest composition level. Random independently generated codevectors were used for base-level codevectors in [Rachkovskij, 2001], though correlated codevectors for similar items are possible. Similar items are represented by codevectors with a more-than-random overlap of 1s.

Thus, distributed representations of structures can be constructed that carry immediate information on both the set of structural components of various hierarchical levels and their structural organization. Similarity of resulting bound representations is influenced both by the set of components and their arrangements. So, similar structures are encoded by similar codevectors. Therefore, it is not necessary to search for the match between the elements of two structures in order to estimate their overall similarity by dot product of their codevectors. Similarity found by one-shot dot product of APNN codevectors takes into account both semantic and structural similarity of episodes. Such representations exemplify “reduced descriptions” [Hinton, 1990] or "meaningful symbols" [Kanerva, 2000] that are central to analogical reasoning.

A mapping process has been defined for APNNs that uses alternating sequential and parallel steps to find structural correspondences between pairs of representations [Rachkovskij, under revision]. Processing is based on similarity preservation in reduced representations and includes finding similarity between elements of the same hierarchical level for mapping, and between elements of different levels for structure traversing.

In order to map two analogs, first their elements (chunks of all hierarchical levels) must be encoded by codevectors. Then, the simplest mapping technique involves placing in correspondence the analogical elements of the same hierarchical level having the largest overlap of codevectors. For interpretation of more formal analogies, a technique with synchronous traversal of hierarchical representations may be required, such as finding the corresponding roles and putting into correspondence their fillers. The consistency of mapping can be verified by checking if both techniques produce the same results. Using various attribute structures and representation schemes, and even changing them in the process of mapping may be required for mapping of more difficult analogies.

Usage of distributed associated memories with fast or even one-shot storage and fast retrieval of the most similar episodes [Frolov et.al, 2002] is also facilitated by sparse and binary character of the APNN representations. These features provide APNNs with a scaling potential and flexibility necessary for analogy-processing for large-scale real-world problems emerging at various stages of counterterrorism activity.

Application of analogical reasoning to antiterrorism tasks

Analogical reasoning can facilitate efforts to minimize the effects of terrorist acts by allowing the use of past episodes to influence the interpretation of current events. There are three stages at which analogical reasoning can be used: prevention of new acts, reasoning about the course of ongoing terrorist activity, and reasoning about the consequences of a new act.

Before terrorist acts, the task is to *prevent* them and *prepare* countermeasures. These include: revealing terrorist groups and individuals; prediction of terracts; preparation of countermeasures (general and specific); full-spectrum assessment of and preparation for threats, effects, and consequences; specialized training of personnel and people (from preventing to responding to an incident); action planning and executing in view of a potential terract. *Within* and *after* terrorist acts, the task is to *interdict*, *mitigate*, *learn*, and *prevent new* terracts, including: optimal response to acts of terrorism; evacuation and emergency medical aid; assess the affected population and damage; emergency and consequence management; prevention of follow-up terracts; study of the terract and its effects.

Applications of analogical reasoning in computer-based systems dealing with various aspects of the life-cycle of terrorist acts for minimizing their effects on civilian populations may include, but are not exhausted, by the following:

- constructing case libraries or KB of life-cycles of known terrorist attacks in order to develop countermeasures to them, to analyze what novel terracts may be and prepare to them;
- inclusion in the KB of information about previous cases that can be used for predicting terracts;
- inclusion of information about what was done after the previous terracts, including errors and correct actions, and giving action planning proposals for the current terract;
- generation of new terract scenarios by analogy to prior attacks, given an initial set of conditions, e.g., a terrorist group, its assets, information about potential targets, etc. These scenarios could then be used to test current anti-terrorist measures as well as emergency response preparations;
- supporting analysis and expert decision making for various scenarios of terracts, from prediction through the course to response;
- inventing new schemes or some aspects of terracts by relaxing some constraints on human analogical reasoning or otherwise changing real cases thus simulating reasoning of terrorists learning from previous attempts of terracts and using this information for analysis and developing countermeasures;
- a better understanding of the way people behave after a terrorist attack, in particular how people make plans by drawing analogies to that attack, such as avoiding doing things that can lead to getting into situation similar to that of a recent terract.

When performing analytic tasks, analysts use analogies in at least four ways

-for organizing and understanding information arriving about a new, emerging or otherwise unfamiliar situations;

-to sharpen understanding of a current situation by comparing it to past situations;

-to help test assumptions about new situations;

-to test for projection of our own biases, goals, values and thought systems into uncertain situations.

Consider the case of generating a prediction for a new terract. There is good evidence that when people generate new ideas, they do so by analogy to known ideas (e.g. [Gentner, et al., 1997; Moreau, Markman, & Lehmann, 2001; Ward, 1994]. Using analogical reasoning, a known incident can be combined with a new set of conditions to predict a new incident. For example, the 1972 hostage crisis at the Munich Olympic games can be combined with the conditions predicted to exist at the 2004 Summer Olympic games in Athens to create a novel scenario for these games. Scenarios generated in this way can be used to test readiness of Civil Defense personnel and to suggest potential suspicious activities that might alert authorities of possible terracts. Different patterns of terracts such as homicide bombers, hijacking planes, and the attempted bombing of the WTC at the beginning of 1990 could provide analogy to the Sept. 11 attack.

Implementation issues

We propose the following scheme to implement and test this approach. The analogical reasoning tools for analogical access, mapping, and inference operating with complex structured episodes should be implemented based the techniques and algorithms of the APNN system. Then the system should be tested on terrorism-related episodes from a KB developed for this purpose. These require:

(1) Finding, constructing, and adapting benchmark episodes that describe the life-cycle of terrorist acts for elaborating and testing analogical reasoning techniques in counterterrorism-related tasks. These benchmark episodes can be constructed by the researchers, with the help of experts where appropriate, based on information that is readily available from public-domain sources (e.g., newspapers, historical records, Internet [Terrorism-related resources]). Obviously, more detailed scenarios can be constructed using information that is not typically publicly available (e.g., military intelligence). The principles of APNNs described here will provide more detailed predictions for new situations when the base domains where they are given are similarly detailed.

(2) Implementing reasoning tools using APNNs:

- parsing of input symbolic representations and its transformation into internal XML-based format of episodes;
- setting the schemes for relational representations; distributed encoding of analogical episodes; analogical access, mapping, inference.

These tools will enable retrieval of previously supplied episodes of terracts and their usage for solving problems concerning new terracts. Then the system can be tested with the constructed complex episodes and situations describing the life-cycle of terrorist acts, such as preparations, execution, consequences. These sample terrorism-related benchmark episodes are meant to be demonstrations of the utility of this approach. Of particular interest is the degree to which descriptions of prior terracts can be used to generate predictions of new scenarios.

Conclusion

Thus, our approach consists in using techniques of Computer Science, specifically Artificial Intelligence and Neural Networks, and combining them with knowledge obtained through the progress of cognitive science in order to create analogical processing tools that provide new quality to counterterrorism systems.

The main goal is twofold:

- to develop the set of ideas and techniques supporting terrorism-related analogical reasoning using structure-sensitive distributed representations within the architecture of Associative-Projective Neural Networks;
- to implement software components and a prototype of an analogical processing toolkit that can potentially enhance the intellectual level of computer-based systems for a wide range of personnel dealing with various aspects of the pressing problem of terrorism and its effects.

The new methods and techniques of analogical reasoning with meaning- and structure-sensitive distributed representations should be developed, investigated, and applied to the problem of counterterrorism. They combine the advantages of distributed representations (possibility to be learnt, natural representation and estimation of similarity, an efficient use of representational resources, generalization potential, etc.) with the necessity of complex structured representations to describe adequately the real-world situations, such as terrorist incidents and associated scenarios, plans, analyses, predictions, countermeasures, etc.

This new knowledge will have a substantial impact on the efforts to minimize the effects of terrorist acts on civilian populations by facilitating knowledge acquisition and formation of terrorism-related knowledge bases, as well as supporting the processes of analysis, decision making, and reasoning with those KBs for users at various levels, from expert to usual personal, before, in the process, and after terracts.

This technology has a potential commercial value at the market place by enhancing intelligence level of existing computer systems, as well as providing a basis for development of a new generation of such systems, that help to provide technological solutions to the threat of terrorist acts, as well as other KB systems. This technique can also provide a basis for developing a psychological theory of human analogical reasoning that is consistent with behavioral data and is also neurally plausible.

Bibliography

[Caid, 1995] W.R.Caid, S.T.Dumais, & S.I.Gallant (1995) Learned vector-space models for document retrieval. Information Processing and Management, Vol. 31, No. 3, pp. 419-429.

[Clement & Gentner, 1991] C.A.Clement & D.Gentner (1991). Systematicity as a selection constraint in analogical mapping. *Cognitive Science, 15*, 89-132.

[Eliasmith & Thagard, 2001] C.Eliasmith & P.Thagard (2001). Integrating Structure and Meaning: A Distributed Model of Analogical Mapping. Cognitive Science, 25(2), 245-286.

[Fauconnier, 1997] G.Fauconnier (1997). *Mappings in thought and language*. New York: Cambridge University Press.

[Falkenhainer et.al., B. Falkenhainer, K.D.Forbus, & D.Gentner (1989) The Structure-Mapping Engine: Algorithm and Examples. Artificial Intelligence, 41, 1-63.

[Forbus et.al., 1999] K.D. Forbus, P.B.Whalley, J.O.Everett, L.Ureel, M.Brokowski, J.Baher, & S.Kuehne (1999). CyclePad: An articulate virtual laboratory for engineering thermodynamics. *Artificial Intelligence, 114*, 297-347.

[Frolov et.at., 2002] A.A.Frolov, D.A.Rachkovskij, D.Husek (2002). Informational efficiency of sparsely encoded Hopfield-like autoassociative memory.

[Gayler & Wales, 1998] R. W.Gayler, R.Wales (1998).Connections, binding, unification and analogical promiscuity. In: K. Holyoak, D. Gentner, & B. Kokinov (Eds.): Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences.(Proc. Analogy'98 workshop, Sofia). NBU Series in Cognitive Science, Sofia, Bulgaria: New Bulgarian University, pp. 181-190.

[Gentner, 1983] D.Gentner (1983). Structure-mapping: A theoretical framework for analogy. Cognitive Science, 7, pp. 155-170.

[Gentner et al., 1997] D.Gentner, S.Brem, R.Ferguson, A.B.Markman, B.B.Levidow, P.Wolff, & K.D.Forbus (1997). Conceptual change via analogical reasoning: A case study of Johannes Kepler. *Journal of the Learning Sciences*, *6,* 3-40.

[Gentner & Markman, 1995] D.Gentner & A.B.Markman (1995). Analogy-Based Reasoning. In M. A. Arbib (Ed.), Handbook of brain theory and neural networks (pp. 91-93). Cambridge, MA: MIT Press.

[Gentner & Markman, 1997] D.Gentner & A.B.Markman (1997). Structure Mapping in Analogy and Similarity. American Psychologist, 52(1), 45-56.

[Gentner et.al. 1993] D.Gentner, M.J.Rattermann, & K.D.Forbus (1993). The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology, 25*(4), 524-575.

[Hinton,1990] G.Hinton(1990).Mapping part-whole hierarchies into connectionist networks. Artificial Intelligence 46,47-75.

[Holyoak & Thagard, 1989] K.J.Holyoak, & P.Thagard (1989). Analogical mapping by constraint satisfaction. Cognitive Science, 13, 295-355.

[HPKB] DARPA's High Performance Knowledge Base Initiative http://projects.teknowledge.com/HPKB/

[Hummel & Holyoak, 1997] J.E.Hummel & K.J Holyoak (1997). Distributed representations of structure: A theory of analogical access and mapping. Psychological Review, 104, 427-466.

[Kanerva et.al., 2000] P.Kanerva, J.Kristoferson, and A.Holst (2000) "Random indexing of text samples for Latent Semantic Analysis." In L.R. Gleitman and A.K. Josh (eds.), Proc. 22nd Annual Conference of the Cognitive Science Society (U Pennsylvania), p. 1036. Mahwah, New Jersey: Erlbaum.

[Kanerva, 1996] P.Kanerva (1996). Binary Spatter-Coding of Ordered K-tuples. In C. von der Malsburg, W. von Seelen, J. C. Vorbruggen, & B. Sendhoff (Eds.). Proceedings of the International Conference on Artificial Neural Networks - ICANN'96, Bochum, Germany. Lecture Notes in Computer Science, 1112, 869-873. Berlin: Springer.

[Kanerva, 1998] P.Kanerva (1998) "Dual role of analogy in the design of a cognitive computer." In: K. Holyoak, D. Gentner, and B. Kokinov (Eds.), Advances in Analogy Research: Integration of Theory and Data from the Cognitive, Computational, and Neural Sciences (Proc. Analogy'98 workshop). NBU Series in Cognitive Science, Sofia, Bulgaria: New Bulgarian University, pp. 164-170.

[Kanerva, 2000] P.Kanerva (2000) Large patterns make great symbols: An example of learning from example. In S. Wermter and R. Sun (eds.), HYBRID NEURAL SYSTEMS (pp. 194-203). Heidelberg: Springer.

[Keane, 1996] M.T.Keane (1996). On adaptation in analogy: Tests of pragmatic-importance and adaptability in analogical problem solving. *Quarterly Journal of Experimental Psychology, 49A*(4), 1062-1085.

[Keane et.al., 1994] M.T.Keane, T.Ledgeway, & S.Duff (1994). Constraints on Analogical Mapping: A comparison of Three Models. Cognitive Science 18, pp. 387-438.

[Kolodner, 1993] J. Kolodner (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

[Kussul, 1992] E.M.Kussul (1992) Associative neuron-like structures. Kiev: Naukova Dumka. (In Russian).

[Kussul et.al, 1991] E.M.Kussul, D.A.Rachkovskij & T.N.Baidyk (1991) Associative-Projective Neural Networks: architecture, implementation, applications. In Proceedings of the Fourth International Conference "Neural Networks & their Applications", Nimes, France, Nov. 4-8, 1991 (pp. 463-476).

[Markman, 1997] A.B.Markman (1997). Constraints on Analogical Inference. Cognitive Science, 21(4), pp.373-418.

[Markman & Gentner, 2000] , A.B.Markman & D.Gentner (2000) Structure mapping in the comparison process. American Journal of Psychology, 113, 501-538.

[Moreau, Markman, & Lehmann, 2001] C.P. Moreau, A.B. Markman, & D.R. Lehmann (2001). 'What is it?' Categorization flexibility and consumers' responses to really new products. *Journal of Consumer Research*, *27*, 489-498.

[Plate, 1995] T.A.Plate (1995). Holographic Reduced Representations. IEEE Transactions on Neural Networks, 6, 623-641.

[Plate, 2000] T.A.Plate (2000). Analogical Retrieval and Processing with Distributed Vector Representations. Expert Systems: The International Journal of Knowledge Engineering and Neural Networks, 17(1), pp. 29-40.

[Rachkovskij, 2001] D.A.Rachkovskij, (2001) Representation and processing of structures with binary sparse distributed codes. IEEE TKDE 13(2), pp. 261-276.

[Rachkovskij, under revision] D.A. Rachkovskij (2002) Some approaches to analogical mapping with structure-sensitive distributed representations (under revision).

[Rachkovskij & Kussul, 2001] D.A.Rachkovskij & E.M.Kussul (2001) Binding and Normalization of Binary Sparse Distributed Representations by Context-Dependent Thinning. Neural Computation 13(2), pp. 411-452 (paper draft available at http://cogprints.soton.ac.uk/abs/comp/199904008).

[Rachkovskij & Kussul, unpublished] D.A.Rachkovskij & E.M.Kussul. Building large-scale hierarchical models of the world with binary sparse distributed representations. www.bbsonline.org/Preprints/Rachkovskij/Referees/Rachkovskij.pdf

[Ramscar & Yarlett, 2003] M.Ramscar and D.Yarlett - Semantic grounding in models of analogy: An environmental approach. 27(1), 2003

[RKF] DARPA's Rapid Knowledge Formation Program http://reliant.teknowledge.com/RKF/

[SAIC] SAIC's Counterterrorism Webpage http://www.saic.com/natsec/counterterrorism

[Schank, Kass, & Riesbeck, 1994]. R.C. Schank, A. Kass, & C.K. Riesbeck (1994). *Inside case-based explanation*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[Terrorism-related resources] Terrorism-related resources of Message Understanding Conference (MUC 3-4). Resources at the International Association For Counterterrorism & Security Professionals http://www.iacsp.com. UN terrorism-related resources http://www.un.org/Depts/dhl/resources/terrorism/elinks.htm. U.S. Department of State counterterrorism resources http://www.state.gov/s/ct.

[Thagard, 1990] P. Thagard, K.J.Holyoak, G.Nelson, D.Gochfeld Analog Retrieval by Constraint Satisfaction Artificial Intelligence 46(1990) 259-310

[Ward, 1994] Ward, T.B. (1994). Structured imagination: The role of category structure in exemplar generation, Cognitive Psychology, 27, 1-40.

Author information

Arthur B. Markman - Department of Psychology, University of Texas, Austin, TX 78712, USA. email: markman@psy.utexas.edu

Dmitri A. Rachkovskij, Ivan S. Misuno, Elena G. Revunova - International Research and Training Center of Information Technologies and Systems, Pr. Acad.Glushkova 40, Kiev 03680, Ukraine. email: dar@infrm.kiev.ua

A PROPOSED STRUCTURE OF KNOWLEDGE BASED HYBRID INTELLIGENT SYSTEMS FOR SOPHISTICATED ENVIRNOMENTS

Agris Nikitenko

Abstract*: The paper deals with a problem of intelligent system's design for complex environments. There is discussed a possibility to integrate several technologies into one basic structure. One possible structure is proposed in order to form a basis for intelligent system that would be able to operate in complex environments. The basic elements of the proposed structure have found their implemented in software system. This software system is shortly presented in the paper. The most important results of experiments are outlined and discussed at the end of the paper. Some possible directions of further research are sketched.*

Keywords*: Artificial intelligence, knowledge based intelligent systems, hybrid intelligent systems, autonomous intelligent systems, inductive reasoning, deductive reasoning, case based reasoning, associative reasoning.*

Introduction

The Artificial intelligence is one of the youngest branches of the modern science.

During a short period of time (lasting only several decades) there have been developed a lot of different technologies and approaches to solve various types of problems existing in the field of artificial intelligence. A complexity of those tasks that can be performed by intelligent systems is growing from year to year. In this paper I would like to keep a closer watch on those intelligent systems that would be able to operate autonomously in complex environments that are close to real world. Obviously, if an intelligent system operates autonomously in a complex environment it needs some kind of environment's model. In spite of model's less complexity it is still quite sophisticated. Though, the basic question is what kind of components is necessary for such an intelligent system in order to maintain and use such a complex environment's model.

Before trying to build a structure of an intelligent system, it is necessary to define the environment in which the system will operate. The basis of such a definition can be found in the assumption that every object (in this case the environment of intelligent system) can be described as a system [Lit.1.] Obviously, a complex environment can be described as a complex system. There are several features that define a complex system [Lit. 2.]:

- uniqueness – usually complex systems are unique or number of similar systems is unweighted.
- hardly predictable – complex systems are very hard to predict. It means that it is hard to calculate the next state of a complex system if the previous states are known.
- an ability to maintain some progress resisting against some outer influence (including influence of the intelligent system).

Of course, any complex system has every general feature of a system such as a set of elements, a set of relations etc [Lit. 3.]

To built a complete model of an environment (complex system) that corresponds to the listed features is either impossible or very expensive. In this case the intelligent system will have incomplete model of environment or will not have it at all. In complex environments usually it is impossible to describe the environment completely. This is caused by a huge space of possible states of environment (or even infinite), expanses or other reasons. It means that an intelligent system in the great part of cases will be using only an incomplete model of environment during its existence.

A very promising way to deal with a complexity and incompleteness is to use some kind of learning mechanisms in order to adjust the intelligent system to new conditions.

A closer watch even on the early methods used in the system theory shows that an analysis of a complex system requires three basic types of reasoning: Deductive, Inductive and Associative [Lit.1,2,3.].

It means that an intelligent system also needs to be able to use all of those reasoning techniques in order to be effective enough.

Basic Features of an Intelligent System

In this section the basic features are outlined and explained according to the previous research activities [Lit.4.].

Summarizing the basic features is:

- an ability to generate a new knowledge from the already existing. This ability can be achieved by means of deductive reasoning. In order to increase an effectiveness a case based reasoning can be used [Lit.5.] Under this feature lies ability to reason logically.
- an ability to learn. This feature can be implemented by means of inductive reasoning. During operation the intelligent system can collect a set of facts through sensing an environment which forms an input for learning.
- an ability to reason associatively. This feature is necessary due to a huge set of possible different situations that the intelligent system can face with. For example, there may be two different situations which can be described by n parameters (n is big enough number) where only k parameters are different (k is small enough number). It is obvious that it is possible to reason about these situations as about similar situations.
- an ability to sense environment. This feature is absolutely necessary for any intelligent system that is built to be more or less autonomous.
- an ability to act. This feature also is necessary for any intelligent system that is designed to do something. If the system (autonomous) is unable to act, it won't be able to achieve its goals.

These features form the basis for an intelligent system that operates in sophisticated environments. According to the features of complex systems that are listed before, any of them can be implemented as it is needed for particular task. The question is how to bind all listed features in one intelligent system.

Obviously, there is a necessity for some kind of integration. There are many good examples of different kinds of integration. For example so called soft computing which combines fuzzy logic with artificial neuron nets [Lit.6.] or Case based reasoning combined with deductive reasoning [Lit.7.].

In order to adjust an intelligent system for some particular tasks different structures can be used [Lit 14].

In this paper I would like to present one of the alternative structures that could be used in order to implement all of the listed features and can form a kernel of an autonomous intelligent system.

Structure of the Intelligent System

According to the list of very basic features there can be outlined the basic modules that correspond to the related reasoning techniques:

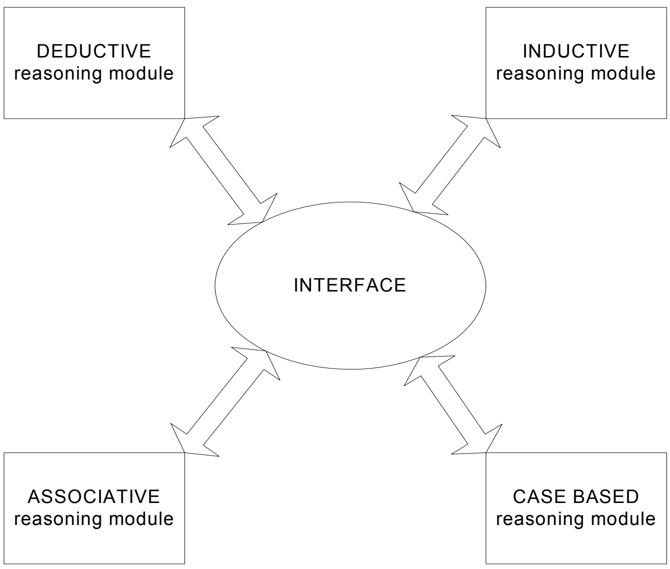


Figure 1. Basic modules

As it is shown in figure 1, there are four basic modules.. As it is said before in complex environments there may be a while of unique situations. To extract (or to learn) any rule an intelligent system needs at lest two equal (or similar – the most part of feature (attributes) are equal) situations. It means that in complex environments a while of situations experienced by an intelligent system may remain unused. Obviously, these unique situations (or cases) may be extremely valuable not only for the intelligent system but also for the researcher that uses the system like it is in medicine. The case based reasoning module is involved to process and use these unique situations.

As it is depicted in figure 1, all of the four modules need some interface to communicate with each other.

Of course, the intelligent system needs additional modules that would supply it with necessary information about the environment and mechanisms to perform some actions. During this research there is developed an alternative structure of the interface that allows combination of four mentioned reasoning techniques. This structure is depicted in the figure 2.

The structure consists of several elements. The fundamental element of whole structure is *object*.

Object. Objects are key elements in the interface structure. They correspond to some kind of entities in the environment (or in the intelligent system). Every object is described with a set of features (attributes). Each feature has some value. As it is depicted in figure 2. objects are linked to each other by associative links. These links form basis for associative reasoning. Two objects are linked if there is any common feature between them. These links can be weighted. The greater weight becomes the more common features are between objects.

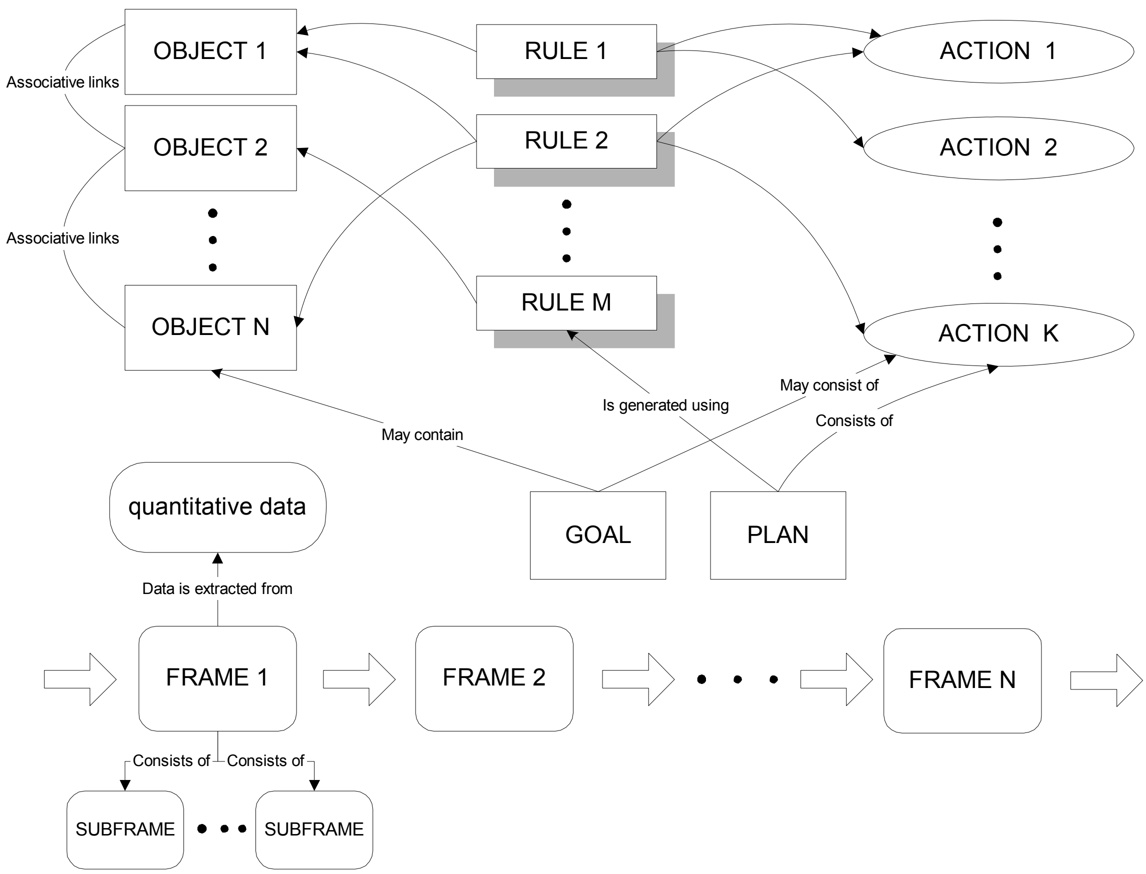


Figure 2. Structure of the interface.

When an intelligent system runs into new situation some subset of objects is activated. These objects map to those entities that the intelligent system currently senses. If there is no rule that can be activated (see below), then the intelligent system may try to activate associated objects where links have some threshold weight. Thus the system can try to reason about objects by using associated rules. The result may be less feasible, but using association between objects the system can run out of the dead end situations. Associative links can spread the “activation” through the whole net of objects if the links among them are strong enough.

A mechanism of associative memory is very useful when the system works with noisy data. This mechanism allows to correct faults of the sensing mechanism [Lit.8.]. For example, if the input vector of the sense which corresponds to some entity has some uncertain or incorrect elements (attributes of object) then the system would not be able to activate any of the objects. In this case associative memory mechanism will activate the closest object [Lit.8.] thus the sensing error will not have significant effect on the reasoning process.

Rules. Rules are any kind of notation that represents causalities. In the practical experimentations were used a well known *if..then* notation. As it is depicted in the figure 2 rules are linked to objects and actions. Each rule may contain a reference to some objects (or its attributes). Therefore if the rule contains such a reference then it is linked to the object. These links help to reduce a searching space and forms a basis for associative reasoning. Each time when the intelligent system receives a new situation, there are activated those objects that are sensed by the system. Linked rules are also activated. The system searches for suitable rules only in the set of activated rules. When system activates objects by using associative links linked rules also are activated thus system can scan also a set of "associated" rules. This ability can significantly improve system's ability to adapt. As it is depicted in figure 2 rules are linked to actions. Rules (for example, those of type *If..Then*) may reference not only to facts but also to actions.

Actions. Actions are some kind of symbolic representation that can be translated by the intelligent system and cause the system to do something. For example "*turn to the right*" causes the system to turn to the right by 90^0.

Actions may be structured in hierarchies. Thus a system can built high level actions that consist of a set of basic actions. For example an action "*open the door*" may consist of two lover level actions: "*unlock the door* " and "*push the door*". High level actions may be formed like scenarios. A scenario can consist of a sequence of lover level actions. It means that rules referencing to the high level actions do not need to reference to each of the basic actions.

The scenarios of actions can be formed as sequence of actions that drive the system to the goal. It means that actions of high level in some way are representation of the positive experience of the system.

Each action consists of three parts: precondition, body and postcondition. Precondition is every factor that should be true before the action is executed. For example, before opening the door it has to be unlocked. Body is a sequence of basic (or lover level) actions. Post conditions are factors that will be true after the execution of action. For example after opening the door, the door will be opened.

2 actions may be sequenced one after another if the following is true:

$$c \in C, C = c' \cup Z ,$$

where c ' – a set of postconditions of the first action
Z – a set of current conditions (known facts)
C – a set of precondition of the second action

Frames. Frames are some kind of data structures that contain the sense array from environment and from the system. It means that frames contain snapshots of the environment's and the system's states.

As it is depicted in figure 2 frames are chained one after another thus forming a historical sequence of the environment's and the system's states. Frames can be structured in hierarchies. Hierarchies help to see values of features that can not be seen in a single snapshot. For example motion trajectories of some object e.c.

Frames are the input for learning (induction module) algorithms. It is obvious that queue of frames can not be infinite due to bounds of hardware equipment. It means that there should be used some "forgetting" mechanism that determines the length of the frame sequence.

Goal. A goal is some kind of a task that has to be done by the system. It can be defined in three ways: as a sequence of actions that should be done, as some particular state that should be achieved or as a combination of actions and states.

Goals can also be structured in hierarchies that determine the priorities of different goals.

Plan. A plan is a sequence of actions that the system is trying to accomplish. It can be formed using both basic and complex actions. After the plan is accomplished it is evaluated depending on whether the goal is achieved or not.

Quantitative data. This element is used to maintain any kind of quantitative data that is needed for the system. For example it can contain certainties about facts or rules, possibilities e.c. A source of quantitative data is the chain of frames.

All of those components together form the interface for the basic modules: Inductive, Deductive, Case based and Associative reasoning.
Fundamental elements of the structure are implemented in experimental software.

Practical Implementation

As it is mentioned above, fundamental elements of the proposed structure are implemented into experimental software.
The implemented elements are: Case based reasoning, Inductive reasoning and Deductive reasoning. Deductive reasoning is implemented as a statement logic module based on rules designed in *if…then* manner. The induction module is implemented using well known algorithm ID3 [Lit.9.] It has its more effective successor C4.5 [Lit.10]. The case based reasoning module is implemented using pairs {situation, action}. Each of such pair has its value which determines how effective it is in particular case. During the planning this value determines which actions are selected if more then one action may be selected.
The environment is implemented as world of rabbits and wolf (domain of pray and hunter). There are defined additional objects “obstacles”. The number of rabbits and obstacles is not specified. The intelligent system is implemented as wolf. Rabbits may be moving or standing at one place. Wolf can catch rabbits. The wolf is moving according to its plan. User can freely change number and place of obstacles and rabbits. The goal can defined and changed at any time.
The intelligent system demonstrates flexibility of the proposed structure. The results of experiments and experience accumulated during the implementation shows that new types of objects can be introduced without changing the proposed structure. The goal can be changed freely.
It means that even being incomplete this structure demonstrates good ability to adapt and to operate
I believe that implementation of the whole structure can give very flexible system that would be able to operate in more complex environments.

Possible Advances and Future Research

Obviously there may be such tasks that can not be done using a single intelligent systems. In other words there may be a task that could be done only by a set of intelligent systems. For example, simulation of real world (battlefields, transport systems e.c.) It means that intelligent system has to be ready to negotiate with other intelligent systems that may be built using different kind of technology. It does not mean that always it is necessary to communicate.
This question is under discussion among researchers working in the field of multiagent systems.
There are different ways to design multiagent system [Lit.11., Lit 12]. In different domains there can be different solutions. Before designing a system for operating in a multiagent environment several questions should be answered.
Some of the basic questions are:

- Will the agents be able to communicate?
- How the agents will communicate to each other.
- What resources they will shared and what resources will stay unshared
- How the conflicts will be solved.
- Of what type the agents will be (competitive, cooperative e.c.)

Only after answering on those questions the intelligent system can be adjusted according to the collected answers.
Referencing to the said above, there may be outlined one of the directions of farther research and experiments – adjustments of the proposed structure in order to allow the intelligent system operate in a heterogeneous communicating multiagent environment. This is the most sophisticated type of multiagent

systems [Lit.12.] and the most interesting form the point of view of research. The structure adjusted for such an environment should be able to operate in less complex environments.

One of the most sophisticated problem in such a multiagent environment is communication because every of communication parameter may be variable. It is easy to imagine that two intelligent systems may try to communicate using different knowledge representation schemas, different knowledge, different communication protocols, different type of "conversation" (for example: questioning, answering, argumentation e.c.) or even different physical communication channels (radio frequency, verbal communication e.c.). All said means that it is almost impossible to design an intelligent system that would be able to adjust its communication mechanism to all possible variations. It means that the system should be design for one or few of the possible communication standards. Which one to choose? May be some of the existing standards should be used [Lit.13.]. This question is to be answered only after a deeper analysis. This is the second possible direction of farther research.

In the field of practical research and experimentation the next step is to design an experimental intelligent system that would be an implementation of the whole proposed structure in order to carry out more complex experiments using different environments and different goals.

The other practical experiments that are of my special interest are experiments with some kind of robotic system in order to try the proposed structure in the field of motion control.

Conclusions

Practical experiments show that the proposed structure may be very flexible even in very changing environments with variable goals. It means that it is reasonable to carry out farther research and experiments in order to advance this structure.

Some problems are related with amount of processed data during the reasoning that significantly slows down the whole system.

It means that an optimization of the processes needs to be a part of future activities.

In spite of the quite promising results that are collected during the practical experiments there are still a lot of open questions that should be answered in farther research activities.

References

1. В.Н. Волкова «Теория систем и методы системного анализа вуправлении и связи» Радио и Связь 1983.
2. В.В. Дружинин, Д.С. Конторов «Системотехника» Радио и Связь 1985.
3. А.Д.Цвиркун «Структура сложных систем» Советское радио 1975.
4. A.Nikitenko, J.Grundspenkis "The kernel of hybrid intelligent system based on inductive, deductive and case based reasoning" KDS2001 conference proceedings, St. Petersburg 2001.
5. Bradley J. Rhodes "Margin Notes Building a Contextually Aware Associative Memory", *The Proceedings of the International Conference on Intelligent User Interfaces* (IUI '00), New Orleans, LA, January 9-12, 2000.
6. M. Pickering "The Soft Option" 1999.A.R.Golding,
7. P.S.Rosenblum "Improving Accuracy by Combining Rule-based and Case based Reasoning" 1996.
8. A.M.Wichert "Associative Computation" der Fakultät der Informatik der Univesität Ulm Wrotslav 2000.
9. J.R. Quinlan "Comparing Connectionist and Symbolic Learning Methods" Basser Department of Computer Science University of Sydney, 1990.
10. J.R. Quinlan "Improved Use of Continuous Attributes in C4.5" Journal of Artificial Intelligence Research 1996.
11. P.Scerri, D.Pynadath, M.Tambe "Adjustable Autonomy in Real-world Multi-Agent Environments" Information Institute and Computer Science Department University of Southern California, 2000
12. P.Stone, M.Veloso "Multiagent systems: A Survey from a Machine Learning Perspective." IEEE Transactions on Knowledge and Data Engineering. 1996.
13. J.Dale, E.Mamdeni "Open Standards for Interoperating Agent-Based Systems" 2000 (last changes)
14. S. Goonatilake, S. Khebbal "Intelligent Hybrid systems" 2002 john wiley & sons

Author information

Agris Nikitenko - Riga Technical University, Division of Systems Theory, Meza street 1/, Riga, LV-1048, Latvia , phone: (+371) 7901045, cell.: (+371) 9424825; E-mail: agris.nikitenko@rembox.lv

KNOWLEDGE LEARNING TECHNOLOGY FOR INTELLIGENT TUTORING SYSTEMS

Taran T.A., Sirota S.V.

***Abstract**: In this work we suggest the technology of creation of intelligent tutoring systems which are oriented to teach knowledge. It is supposed the acquisition of expert's knowledge by using of the Formal Concept Analysis method, then construction the test questions which are used for verification of the pupil's knowledge with the expert's knowledge. Then the further tutoring strategy is generated by the results of this verification.*

***Keywords**: Computer tutoring, Formal Concept Analysis.*

Introduction

The conception of the up today's computer tutoring systems is based on teaching knowledge which represented like a pack of the facts and rules. In [3] the learning type's systematization was suggested accordant to cognitive levels used in studying. There are four cognitive levels marked: (1) creative, (2) analogy-generalization, (3) explanation and (4) programming. The 4-th level means the training to solve certain type of tasks, in the 3-d level the studying goes by explanation, the aidless work with exercises and problems needs the analogy-generalization, the highest, creative, level is supposed the pupil to ideate himself the concepts and relations between them. The interaction between the pupil and the tutoring system can hit different cognitive levels. It is clear that using of all four levels is optimal strategy of teaching. But up to day tutoring systems are not able to interact on creative level [3].

In [9] two basic directions of the studying process are marked: the concept learning and training. The concept learning has next stages:

- The description of objects attributes, quality and quantity characteristics of the objects and processes in the domain;
- The eduction of concepts;
- The definition of the relations between concepts;
- The definition of dependences on the sets of attributes and characteristics of objects and processes;

In concept learning tutoring systems the cardinal problems are representation of the facts, description of the attributes and relations and rigorous definition of the concepts to be memorized. The memorizing is the main procedure pupil to do. The further action of the system is directed to monitoring of knowing the main concepts of domain. The training started when the pupil have got certain set of knowledge. Then we create some problems to make pupil use his knowledge for solving. Often the pupil is suggested the gradually more difficult tasks. During the solving the pupil can return to the material studied and get some help refilling his knowledge.

In present tutoring systems the most attention is paid to the second stage. The training subsystems are most active component while during concept learning the developers are sated with passive presentation. As researches showed [8], the pupil's knowledge must to reach certain level to avoid antagonism for further tutoring.

In this paper we suggest the tutoring systems technology of interactive conceptual learning including the test of pupil's knowledge. This technology is embodied in software tools. The main features of this software are: the extraction of expert knowledge; the formalization of the domain's concepts; the construction of the knowledge base; the semi automatic production of the test questions; the construction of the pupil's cognitive model; the estimation of the pupils' knowledge; the creation of the tutoring strategy.

The main task of the primary learning is habituation with the domain concepts. For the formalization of these concepts we use the Formal Concept Analysis.

Such approach represents the formal concept like as a pair <intent; extent> where the extent represents the set of objects and the intent is the set of their attributes. For the concept extraction we use the formal context for the corresponding part of the domain. The formal context is represented by the table <object; attribute> in witch every attribute is marked if it is the proper of certain object. For every formal concept maximal nested full submatrix corresponds. The set of concepts is ordered by the relation superconcept - subconcept and forms a full lattice. For testing of the pupil we use the set of questions witch is generated automatically using this lattice. The purpose of testing is to estimate how the pupil grasped the concepts and relations between them. The test questions are mainly of the closed type. The main problem of their recomposing is a choice of destructors. For this we suggest to use the nearest to the concept testing conceptual lattice elements. The new conceptual lattice is creating using the pupils' answers. This lattice represents the domain in the pupils mind. Then we compare it with the master lattice. The differences between them are used to generate a tutoring strategy.

The Concept Presentation in Tutoring System

The method of Formal Concept Analysis is used for knowledge representation of problem domain. It was suggested by R. Wille [4, 5] and at the present time is successfully applying in the problems of data-mining and computer tutoring [1, 2, 10]. The backbone is follows. Let us consider the set of objects V and the set of attributes A with an arbitrary relation $I \subseteq V \times A$, such that pIa, where $p \in V$, $a \in A$, if and only if a is the attribute of object p. Then $K = (V, A, I)$ is called the *formal context.* The binary matrix defines the correspondence of objects and attributes. Let define the correspondence [6]:

$$P' := \{y \in A \mid xIy \text{ for all } x \in P\}, \text{ for } P \subseteq V,$$

$$G' := \{x \in V \mid xIy \text{ for all } y \in G\}, \text{ for } G \subseteq A.$$

Then the pairs (P, G) satisfying $P \subseteq V$, $G \subseteq A$, $P' = G$, $G' = P$ are called *the formal concepts* of the formal context $K = (V, A, I)$. The set of objects P amounts the *extent* of concept and the set of all their attributes amounts its *intent.* Every object $p \in P$ have all attributes from subset G. So, the formal concept is the set of objects from domain such that every one of them has all attributes from certain subset of attributes of that objects.

The set of formal concepts (P, G), where $P \subseteq V$, $G \subseteq A$, is partially ordered by the relation: $(P_1, G_1) \le (P_2, G_2)$, if $P_1 \subseteq P_2$ and $G_2 \subseteq G_1$, and form a complete lattice $L(K)$, called *the concept lattice* of the context K [4]. The pair (P_1, G_1) is called the subconcept of the concept (P_2, G_2), and the pair (P_2, G_2) is called the superconcept of the concept (P_1, G_1).

The concept lattice can be represented by the line diagram (Hasse diagram) in which every node of the concept lattice is corresponded by the concept from context. The dual isomorphism on the concept lattice reflects the inverse between the intent and extent of the concepts: the bigger is extent the less is intent

Example. The formal context "Geometry Figures" is represented in the Table 1. This context is based on the set of geometry figures and the set of their attributes. Maximal nested full submatrix corresponds for every concept. For example, marked submatrix in Table 1 corresponds to formal concept {<*Triangle*, *Tetragon* (*quadrangle*), *Pentagon*, *Hexagon*>, <*Vertexes*, *Area*, *Sides*, *Angle*>} witch can be defined like a "*polygon*". Hereby formal concept is the set of objects from domain such that every one of them has all attributes from some subset of domain attributes.

Table 1. Context «Geometry Figures»

	Has Vertexes	Has Length	Is Line	Has Area	Has Sides	Has Angle
Point						
Straight Line			x			
Half Line	x		x			
Straight Line Segment	x	x	x		x	
Angle	x				x	x
Circle				x		
Circumference		x	x			
Curve Line		x	x			
Poly Line	x	x	x		x	x
Triangle	x			x	x	x
Tetragon (quadrangle)	x			x	x	x
Pentagon	x			x	x	x
Hexagon	x			x	x	x

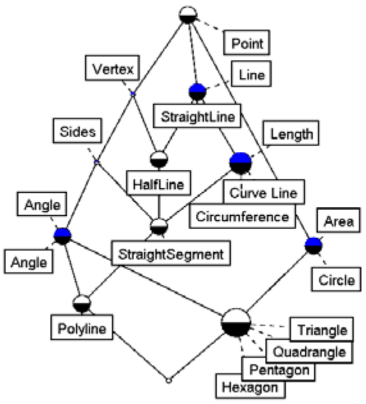


Fig. 1. Conceptual lattice of the context «Geometry figures»

The diagram of conceptual lattice is represented in a Figure 1. By this lattice we can easy find which attributes corresponds to proper object. One concept *(P, G)* corresponds to each node on the diagram. *P* includes the labels of all objects directly underlying, and *G* includes labels of all attributes lying directly above of the given node. The node on the diagram is marked by the label *(p, a)*, where *p* includes the labels of objects, and *a* – labels of attributes directly appropriate to the given node. For example, in Fig. 1 the node with a label «*angle*» and the attribute «*has angle*» corresponds to the concept {<*Angle, Poly line, Triangle, Tetragon (quadrangle), Pentagon, Hexagon*>, <*Vertexes, Sides, Angle*>}. In this context the concept with the empty set of attributes {<*Point*>,∅} corresponds to the lattice unit. The point is a primary concept which can not be defined by the set of properties (attributes).

On a concept lattice it is easy to find the *common* attributes for several objects and the *unique* attributes corresponding only to one of objects. For example objects «*triangle*» and «*circle*» has one common attribute "*area*" Likewise easy to find all objects that have certain attribute. For example we can see that all of "Straight Line", "Half Line", "Straight Line Segment", "Circumference", "Curve Line", "Poly Line" are "lines"

One more important result is that the construction of a concept lattice defines the partial order on a set of objects and attributes. Partial ordering of objects and attributes allows us to reveal the dependence between them. Let $K = (V, A, I)$ is a formal context, $X \subseteq A$, $Y \subseteq A$. Then $X \rightarrow Y$, т.е. *X implies Y,* or the set of attributes *Y depends* on the set of of attributes *X,* if all objects from $P \subseteq V$, that have attributes from *X,* also have all attributes from *Y,* i.e. $X' \subseteq Y'$ (or $Y \supseteq X'$). In this case the concept containing *X* stands lower of the concept containing *Y* on the diagram of a concept lattice.

For example we can see on the diagram that the property "*to have a angle*" is followed by the properties "*to have sides*" and "*to have a vertex*", and property «*to have a length*» is followed by the property «*to be a line*».

The described above features of the concept lattices allow to arrange the pupil's knowledge about subject domain and from other side allow to semi automate the process of the test questions production for the checkout of a pupil's knowledge.

Representation of the Subject Domain Knowledge

The suggested tools environment uses material is represented by traditional means of hypermedia such as texts, pictures, animation. For the automation of the tutoring and knowledge checkup process we need the structures reflected the connection between the parts of material studied and connections between the main concepts inside every part. The common structure of the subject domain is represented by the ontology. The main structure witch corresponds the knowledge by the fragments is the semantic networks. The construction of the semantic network is process to be difficult formalized. Mainly supposed, that the expert is to do it. He describes the connections between the concepts and objects of the subject domain by the means of graphics visualization [7]. Using the formal contexts allows automating partially the semantic network construction. It is easier to preset the objects and their properties and then using concept lattice to define the main concepts and relations between them. The same process can be used for the checkout of the pupil's knowledge. Constructing his conceptual lattice the pupil reflects the concepts system like he ideates it himself. The main criteria of the full retention of the material are the isomorphism between the conceptual model of the pupil and the master model created by expert.

Transformation of the concept lattice into the semantic network is under next rules. We define the context objects like primary concepts of the semantic network. Then single-place predicates-attributes like *PredicateName*(*X*) *we* transform into two-place predicates like *Function*(*PredicateName, X*), where the *Function* means belonging *X* to the class, type, set or describes other connection between object and attribute (the sets of the objects and attributes can intersect i. e. the same essence can be an object and an attribute. (In our example such essence is a "*angle*"). Than we specify the predicates: every variable is replaced by the object from the predicate truth set according the formal context matrix.

For illustration let us see example above. In context «Geometry Figures» every attribute is a single-place predicate defined on preset set of objects – Geometry figures. Case in point *"X is a Line", "X has a Area"* etc. The predicates truth sets marked in a Table 1. We transform all this predicates into two-place predicates *Is(X, Y)* and *Has(X, Y)*. There are only two in our case. Decomposition of the lattice connections allows to construct hierarchy semantic network (fig. 2).

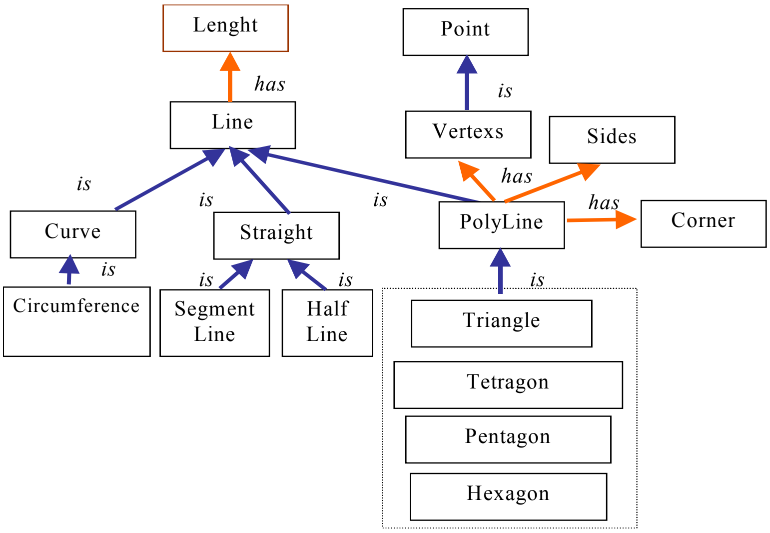


Fig. 2 Hierarchical semantic network for the Context «Geometry Figures»

The data input is realized by creating and filling the context tables. It is possible to organize the row and column titles like hyperlinks to pages that describe the definitions of objects, backbones of their properties and explanations the attributes.

Some objects can have attributes only if the attributes of higher level exist. This allows extracting the attributes of lower level into separate context linked with the main context by the attribute of higher level. Only objects with this attribute are included in it. For example the figures attributed "*Has area*" can be separated to the new context. From the other side some objects can have additional attributes not immanent to other objects. In this case nested context can be created. For example nested context for the object "*Triangle*" will has attribute "*Equilateral*", "*Isosceles*", "*Right*" etc. As the result the context hierarchy (Hypercontext) can be constructed. This structure has a lot of advances. Hypercontext can be processed whole and partially with free choice of processing deepness levels, separate branches, or single contexts. Thus exclude dismissing of the dependent attributes, if the object does not have determinative attribute of higher level. Often necessity of using quantitative attributes appears. Than we use many-valued contexts [4].

Context processing includes next steps: generating of the concept set, creating the line diagram of the concept lattice, extracting of the implications basis. Resulted knowledge system is used by the next way:

– Concepts are raw data for the test generation;
– Implications are used by expert for checkout of context fullness and for test generation;
– Concept lattice is used by test constructor and choice of the tutoring strategy;
– Line diagram is used for visualization of the domain structure fragment.

The implications proved from context can be differed into three groups.

A) Non common sensitive implications. These are ones having false premise on the set of all context objects. Such implications are excluded from further processing.

B) Right (True) implications witch reflects the domain relations correctly. These are relations where the intent of premise and consequence coincides. The accuracy of such implications is 100%.

C) Plausible implications witch reflects the domain relations correctly, but not for all objects, i.e. the intent of premise is bigger than intent of consequence.

For example in context «Geometry Figures» the implication basis includes the right implications:

1. If the Object «*has sides*», that it «*has vertex*» (7 objects);
2. If the Object «*has angle*», «*has vertex and sides*» and (6 objects);
3. If the Object «*has length*», that it «*is line*» (4 objects);
4. If the Object «*has vertex and area*», that it «*has sides and angle*» (4 objects;
5. If the Object «*is line and has vertex and has sides*», that it «*has length*» (2 objects;

6. If the Object «*is line and has vertex and length*», that it «*has sides*» (2 objects;
7. If the Object «*is line and has area* », that it «*has vertex, length, sides, and angle*» (0 objects).

These implications are apparent, because their accuracy in this context is 100%. Besides this there are plausible implications.

8. If the Object «*has vertex*», that it «*has sides*» (accuracy – 86%).
9. If the Object «*has vertex and sides*», that it «*has angle*» (accuracy – 83%).

All this implications from B and C are suggested for experts checkout as the questions like: «Is it true that IF <common name objects> <attribute A_{11}> & <attribute A_{12}> & … & <attribute A_{1n}>, THAN <attribute A_{21}> & <attribute A_{22}> & … & <attribute A_{2n}>?». Expert can adopt implication or do not. If the expert can not answer "yes" for this question he must to suggest counter-example from the domain and include it into context, or to correct given context if it has a mistake and reprocess it. The process will be done, when expert accepted all implications.

In the context above implication 7 which is false for any objects is excluded. The implications 1-6 which were apparent are accepted as a rule. Implication 8 is false for one object "*segment*" and implication 9 is false for the object "*half-line*". These implications can be cased as a rule with elimination because the concept «*segment*» is basic for such concepts like «*poly-lines*» and "*Polygon*". "*Half-line*" is basic for such concepts like "*angle*". It is possible to add the predicate - «*x consist of* ». Thus our conceptualization will be more complete.

Testing of the pupils knowledge

Interactive tutoring environment supposes closed iterative cycle of learning which includes such components like: presentation of new material, testing of the pupils' knowledge, constructing of his cognitive model, its comparison with a master one, generation of the tutoring strategy.

This software can automatically generate the tests witch can be used for estimation of pupil's knowledge about main concepts and relations between them.

Any question consists of premise and subject. The subject determines the set of possible answers (explicatively and implicatively). The premise adjects the subject with instruction using witch it is necessary to choose the right answer. The formal concepts let us to test the knowing of their attributes. By the concept (*P, G*) we can generate two types of questions:

- For given object *P* to define the set of their attributes *G*;
- For given set of attributes *G* to define the set of objects *P* witch have all properties from *G.*

Beside this it is possible to recompose tests by intersection of concepts intents and/or extents.

- What common properties have objects $P_i \cup P_j$;
- What objects have properties $G_i \cap G_j$.

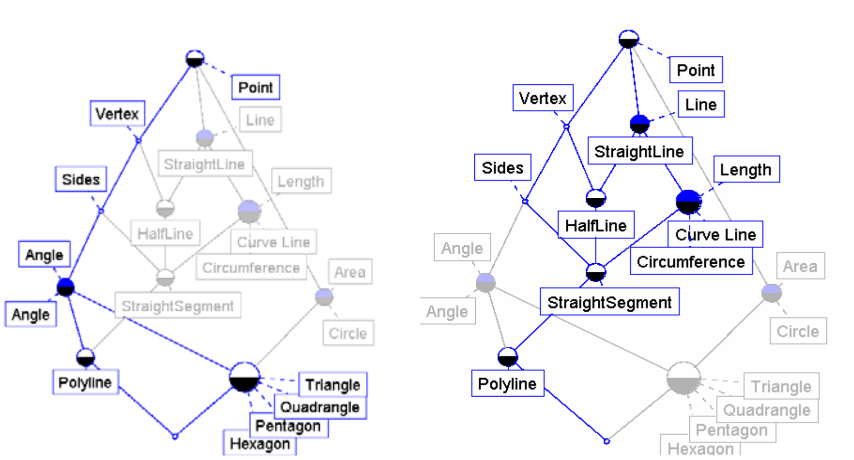
For example, the question can look like: « *What attributes has circumference*? »; « *What objects has an area*? ». The tests like that let us to compose the cognitive model of pupil as the formal context (lattice).

Using implications we can test how the pupil understood the logic and rules of given subject. In the questions composed using implications the rules of the conjunctions elimination are used. For example the question can be like "What *attributes have figures having sides*?" Such questions may be difficult for the pupil. That is why we use not only open form questions but closed form, meaning the questions where one answer from a proposed list must be chosen ('Yes" or "Not"). The exampled question in closed form can be like: "*Is it true that if the object has sides, than it has vertex*?" (answer: «Yes» or «Not»).

The main problem of automatic generation of questions is the choice of destructors i.e. most believable alternatives of answer. Traditionally this problem is solved by the open testing (without the predefined alternatives) and most resent wrong answers become destructors. This solution is quite expensive. That's why we suggest follows.

Let $J(a)$ – ideal, generated by the concept a, $a \in L$, where L – initial lattice; $D(a)$ – dual ideal, generated by the concept a. It contains all upper elements having path to a. For selection of destructors it is necessary to take sequentially concepts $x \in D(a) \backslash \{a\}$ under criteria of minimal distance from a, and generate new ideals $J'(x)$. The destructors will be objects from all concepts y, where $y \in J'(x) \backslash J(a)$. For attributes testing we generate new dual ideals $D'(x)$ and choose destructors from the set $D(x) \backslash D'(x)$.

For example, we can see on fig. 3,a ideal and dual ideal, generated by the concept (*{Angle, Poly-line, Triangle, Tetragon, Pentagon, Hexagon*}, {*has angle, has sides, has vertex*}).



a) *b)*

Fig. 3. Ideals of lattice "Geometry Figures"

The question may be like: «What common properties have *Angle, Poly-line, Triangle, Tetragon, Pentagon, Hexagon?".* To the answer list beside right answers we must add some destructors. Let us construct the ideal generated by the nearest upper concept node (labeled "*sides*"). It includes the object *"segment".* For this node we construct the dual ideal (fig. 3, b). We see that the object *"segment"* has attributes «*is line*» and «*has length*» which belongs to difference between this dual ideal and previous one. This attributes will be taken as destructors.

The questions in which it is necessary to decide if the sentence is correct suppose the answers: "Yes", "No". Questions of this type can be constructed by implications and concepts as well. The questions of this type contains always true sentence. For the construction of the incorrect sentences we can replace some premises between two implications or some attribute sets between concepts. More complete questions suggest arranging the elements by given criteria, combining elements into the groups or to specify the value of … (volume, weight, etc.) of the listed objects". For this type we supposed to use the many-valued contexts. The quantity attributes are contained directly in the concept. That is why this type is not difficult.

Conclusions

Nowadays computer technologies let us to create intelligent tutoring systems where the knowledge about subject studied meta-knowledge about tutoring control and pupil's knowledge estimation are represented distinctively. Developed technology and software tools using FCA let us not only to visualize concept system of domain but also automate the generating of exercises for acquisition of pupil's knowledge for its testing.

Bibliography

[1] Burmeister P. Formal Concept Analysis with ConImp: Introduction to basic features // G.Stumme, R.Wille (eds.) Begriffiche Wissensverarbeitung: Methoden und Anwendungen. Springen-Verlag. – 1999.
[2] Kuznetsov S.O., Ob'edkov S.A. Algorithms for the construction of the Set of All Concepts and Their Line Diagram, Dresden, 2000.
[3] Stefanuk V. L. Leaning Level Analysis in Intelligent Tutoring Systems // Japan-CIS Symposium on Knowledge-Based Software Engineering, 1994 (JCKBSE'94). Pereslavl-Zalesski, May 10 - 13, 1994, P. 9 -13.
[4] Wille R. Ganter D. Formal Concept Analysis. Springer –Verlag. Berlin. - 1999.
[5] Wille R. Knowledge Acquisition by methods of Formal Concept Analysis // Preprint № 1238. Technische Hochschule Darmstadt. 1989.
[6] Birkgoff G. Lattice theory. Amer. Math. Soc. Providence / R.I. 1967.
[7] Gavrilova T.A., Choroshevski V.F. Knowledge bases of intelligent systems. Sankt-Petersburg. Piter. 2000 (in Russian).
[8] Golitsin G.A., Petrov V.M. Information – behavior – creation. - M.: Nauka, 1991.
[9] Petrushin V.A. Expert – tutoring systems. – Kiev: Naukova dumka, 1992.
[10] Taran T.A., Sirota S.V. Concept tutoring in intelligent tutoring systems based on formal concept analysis. // J. Artificial Intelligence. Donetsk. №3. 2000. (in Russian).

Author information

Tatyana A. Taran - National Technical University of Ukraine "KPI" Applied Mathematic Department Av. Peremogy, 37, 03056, Kiev, Ukraine e-mail: taran@pma.ntu-kpi.kiev.ua ; **Sergiy V.Sirota -** Prosvita publishing Ltd. 46-Shevchenka blwd. Kiev Ukraine Tel/Fax: + 38044 2349523 e-mail: sirota@prosvita.kiev.ua

KNOWLEDGE PRESENTATION AND REASONING WITH LOGLINEAR MODELS

Veska Noncheva, Nuno Marques

***Abstract**: Our approach for knowledge presentation is based on the idea of expert system shell. At first we will build a graph shell of both possible dependencies and possible actions. Then, reasoning by means of Loglinear models, we will activate some nodes and some directed links. In this way a Bayesian network and networks presenting loglinear models are generated.*

***Keywords**: computer oriented statistics, knowledge discovery, learning Bayesian networks, automatic analysis of multivariate categorical data sets.*

Introduction

Our main aim is to link statistical theory to some networks in order to enrich computer's reasoning capability. In this paper we will define a new data structure called LLN and offer an algorithm for learning a Bayesian network by using knowledge obtained from categorical data set. This algorithm finds out the loglinear model, describing data, a presentation of this model by a LLN and a Bayesian network describing the relationship among the variables of interest.

Categorical data is most often modelled using loglinear models. In this paper we provide a principled foundation for reasoning with Loglinear models. We will discuss loglinear models that describe association patterns among two and three categorical variables.

Graphs are natural data structures for digital computers. We will provide a framework presentable by a direct graph for modelling categorical data and interpreting the results. Different directed graphs can represent the same dependence structure for the set of associated variables. Consequently, if the links have no causal interpretations, we will obtain a set of equivalent graphical structures.

Basic concepts and definitions

Suppose we have a set of *n*, $n>1$ possibly related categorical variables $V=\{X_1, X_2, \ldots, X_n\}$. This set can be represented pictorially by a set of nodes – one node for each variable of *V*. These nodes can be connected by arcs. The dependency structure could be presented by a Bayesian network. The language of Bayesian networks is described in [Castillo,Gutierrez,Hadi,1997].

Suppose also that we have a set of *k* actions $A=\{A_1, A_2, \ldots, A_k\}$, that could be applied on some of these *n* nodes. The objective of an action is building a loglinear model describing categorical data available. These actions can be applied to some variables from *X*. The application of an action to objects is visualised by directed arcs.

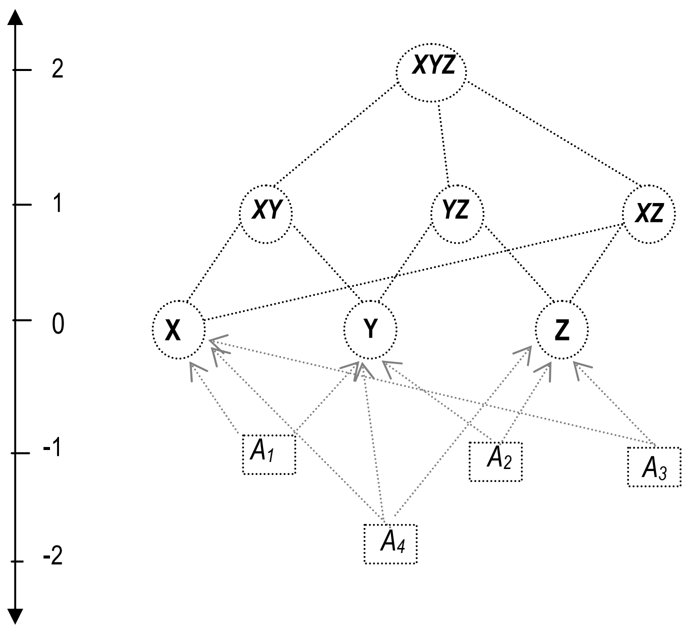
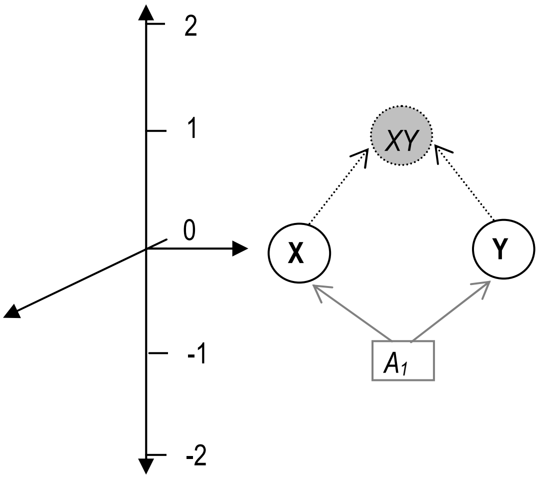


Figure 1. Graph shell.

Let *V*={*X*,*Y*,*Z*}. Actions and nodes are placed on different levels in the graph shell (see *Figure 1*). On level 0 the nodes, presenting the associated categorical variables *X*, *Y*, and *Z* are placed. On level 1 – nodes, presenting new variables *XY*, *YZ*, and *XZ*. The variable *XZ* could be composed of the *IK* combinations of levels of *X* and *Z* or it could present partial association only. On level 2 – nodes, presenting a new variable *XYZ*. The variable *XYZ* could be composed of the *IJK* combinations of levels of *X*, *Y* and *Z* or it could present three-factor interaction only. On level -l – actions applicable to nodes from level 0 and giving results in nodes from level *l*, *l*=1,2. On a sub-network in the level interval [-1, 1] we can reason about the relationship between two variables. On a sub-network in the level interval [-2, 2] we can reason about the relationship between three variables. In the general case on a sub-network in the level interval [-*l*, *l*] we could reason about the relationship between *l+1*, *l=1,2*,… variables.

Loglinear models and their interpretations

In two dimensions only two distinct loglinear models can occur, in general. Either the two variables are independent, or they are associated. The loglinear models, presented bellow, present these two cases.
Given a two-dimensional contingency table the models assume a sample of size *n* distributed over *IJ* cells. Under multinomial sampling, the probability that an observation will fall into cell ij is π_{ij} for all *i=1*, … , *I*, *j=1*, …, *J*. The expected value m_{ij} is $n\pi_{ij}$. The expected value for the observed counts in a contingency table could be estimated by independence loglinear model $\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y$, where the parameters $\{\lambda_i^X\}$ and $\{\lambda_j^Y\}$ satisfy $\sum \lambda_i^X = \sum \lambda_j^Y = 0$, or by dependence loglinear model $\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_{ij}^{XY}$, where the parameters $\{\lambda_i^X\}$, $\{\lambda_j^Y\}$ and $\{\lambda_k^{XY}\}$ satisfy $\sum \lambda_i^X = \sum \lambda_j^Y = \sum \lambda_K^{XY} = 0.$
These two mathematical models are graphically presented in *Figure 2* and *Figure 3*. The case of independence is shown in Figure 2. This properly indicates the presence of mutual independence. The case of interaction is depicted in Figure 3.
Given a three-dimensional contingency table the model assumes a sample of size n distributed over *IJK*=*N* cells. Under multinomial sampling, the probability that an observation will fall into cell *ijk* is π_{ijk} for all *i=1*,… ,*I*, *j=1*,…,*J*, *k=1*,…,*K*. The expected value m_{ijk} is $n\pi_{ijk}$. The model of mutual independence for a three-dimensional contingency table is $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z$. The parameters $\{\lambda_i^X\}$, $\{\lambda_j^Y\}$ and $\{\lambda_k^Z\}$ satisfy $\sum \lambda_i^X = \sum \lambda_j^Y = \sum \lambda_K^Z = 0.$



$$\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y$$

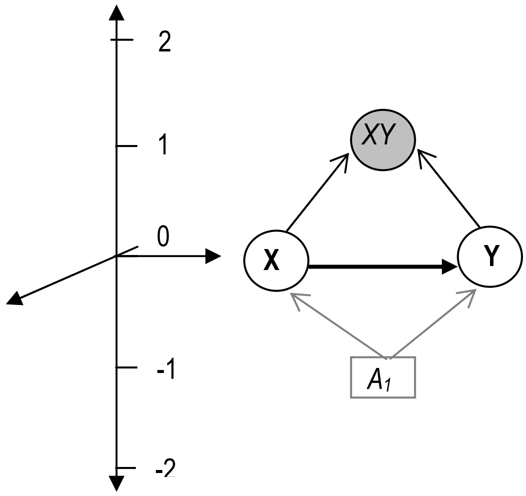
$$p(x, y) = p(x)p(y)$$

Figure 2. Graphical presentation of the independence loglinear model

Interactions between two or all three variables can be modelled by including the additional terms $\{\lambda_{ij}^{XY}\}$, $\{\lambda_{jk}^{YZ}\}$, $\{\lambda_{jk}^{YZ}\}$ and $\{\lambda_{ijk}^{XYZ}\}$ with zero sums over the parameters. The interaction structures are following:

mutual independence, partial independence, conditional independence, no three-way interaction, and three-way interaction.

Mutual independence of the three variables is equivalent to $\pi_{ijk} = \pi_{i++}\pi_{+j+}\pi_{++k}$. The loglinear model of mutual independence is $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z$. All variables are pair-wise (mutually) independent. Thus, only the main effects λ_i^X, λ_j^Y, and λ_k^Z appear in the model. Each pair of variables is also conditionally independent and marginally independent.



$$\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_{ij}^{XY}$$

$$p(x, y) = p(x)p(y \mid x)$$

Figure 3. Graphical presentation of dependency loglinear model for two-dimensional contingency table

Partial independence means presence of λ_i^X, λ_j^Y, λ_k^Z, and additional presence of one $\lambda^{AB}, A, B \in \{X, Y, Z\}, A \neq B$. Suppose that the variable X is partially independent of *Y* and Z. In accordance with the definition the variable *X* is partially independent of *Y* and Z, if $\pi_{ijk} = \pi_{i++}\pi_{+jk}$ for all *i,j,k*. The composite variable *YZ*, which has *JK* different levels combinations of *Y* and *Z*, is mutually independent of X. The corresponding loglinear model is $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{jk}^{YZ}$. The variable X is jointly independent of Y and Z. The variables *X* and *Y* are conditionally independent given *Z*, and *X* and *Z* are conditionally independent given Y. The variables *Y* and *Z* are conditionally dependent. The variable *X* is also independent of *Y* and *Z* in the *X-Y* and *X-Z* marginal tables.There are three models of partial independence.

Conditional independence of *X* and *Y* given *Z* means $\pi_{ij|k} = \pi_{i+|k}\pi_{+j|k}$ for all *i,j,k.* The loglinear model is $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$. The variable *Z* is conditionally dependent with both *X* and *Y*. The variables *X* and *Y* may be marginally dependent, even though they are conditionally independent. There are three models of conditional independence.

The loglinear model of no three-way interaction is $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$. All three pairs of variables are conditionally dependent. Although every variable interacts with each other variable, there is no interaction between all three variables. No pair of variables is conditionally independent. When there is an absence of three-factor interaction, the association between two variables is identical at each level of the third variable. The cell probabilities have form $\pi_{ijk} = \psi_{ij}\phi_{jk}\varphi_{ik}$.

All parameters are included in the three-way interactions model. The loglinear model of three-way interaction is $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ} + \lambda_{ijk}^{XYZ}$. This is the model where every possible interaction is included. The only interpretation of this model is the fact that apparently all other models failed to represent the data in a suitable way.

The most common problem in loglinear modelling is to find the most suitable model. The better model describing data includes as few interaction terms as possible and declares as much of the deviation from

mutual independence as possible. We usually use χ^2 and G^2 statistics to judge the adequacy of a loglinear model. Roy and Mitra gave Person-type statistics for large-sample tests.
We have illustrated ideas using the two and three-variable case. Loglinear models for four-way tables are more complex than for three-way tables, because of the variety of potential partial association, three-factor interaction patterns and four-factor interaction pattern.
We can readily extend the framework to arbitrary multi-way tables.
When the number of dimensions increases, both the number of possible interaction patterns and the number of cells dramatically increase. Frameworks for multy-way tables will be much more complex.
The question whether this interaction is statistically significant or not remains unrevealed, as long as the number of underlying observations is unknown. A way for presenting our belief in a loglinear model is presented in [Noncheva,Marques,2002].
Each of these models are visualised below. The topological shape of each model-type is not invariant against a permutation of the variables.
A visual method based on mosaic plots for interpreting and modelling categorical data is considered in [Theus,Lauer,1999]. Paik suggested circle diagrams for presenting results from three-way tables [Paik,1985].

Types of independence in a three-way cross-classification of X, Y, and Z

A relationship is defined by the join distribution of the associated random variables. The joint distribution determines the marginal and conditional distributions. Simplification occurs in a joint distribution when the component random variables are statistically independent. We will discuss four types of independence for categorical variables.
The three variables are mutually independent when $\pi_{ijk} = \pi_{i++}\pi_{+j+}\pi_{++k}$ for all i, j, and k. On a log scale, mutual independence is the loglinear model $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z$.
Variable Y is jointly independent of X and Z when $\pi_{ijk} = \pi_{i+k}\pi_{+j+}$ for all i, j, and k. This is ordinary two-way independence for X and the new variable YZ composed on the JK combinations of levels of Y and Z. The loglinear model is $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{jk}^{YZ}$. Mutual independence implies joint independence of any one variable from the others.
If X and Y are independent in the partial table for the kth category of Z, then X and Y are said to be conditionally independent at level k of Z. If $\{\pi_{ij|k} = \pi_{ijk} / \pi_{++k}, i = 1,...,I, j = 1,...,J\}$ denotes the joint distribution of X and Y at level k of Z, then conditional independence at level k of Z is $\pi_{ij|k} = \pi_{i+|k}\pi_{+j|k}$ for all i and j. X and Y are conditionally independent given Z when they are conditionally independent at every level of Z, or equivalently, when $\pi_{ijk} = \pi_{i+k}\pi_{+jk} / \pi_{++k}$ for all i, j, and k. Conditional independence of X and Y is the loglinear model $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$. If *Y* is jointly independent of *X* and *Z*, then *X* and *Y* are conditional independent.
We say that X and Y exhibit marginal independence if $\pi_{ij+} = \pi_{i++}\pi_{+j+}$. Joint independence of Y from *X* and *Z* (or of *X* from *Y* and *Z*) implies *X* and *Y* are marginally independent.
The relationships among the four types of independence are summarized in Figure 5. The basic question is how these types of independence could be presented graphically within the framework we are building. A solution of this task is given in Figure 6 and Figure 7.

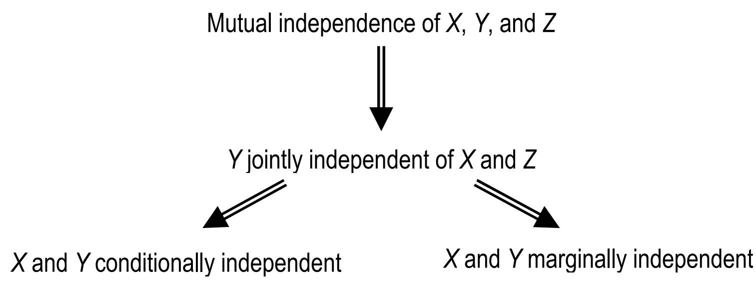


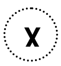
Figure 5. Relationships among types of X-Y independence.

Decision network for reasoning with Loglinear Models

Decision networks are graphic structures, that represent probability relations and information flows ([Shachter,86], [Shachter,88]). We introduce a kind of decision network for reasoning with loglinear models called loglinear network (LLN). A variable from the loglinear model is presented as a node in the graph of the loglinear network.

Definition: A loglinear network (LLN) comprises of the following set of items: (X,(A,I), (X',P), A*, u, δ). It is within this particular set where they are shown as below:
X is the directed graph of the (in)dependency among all variables X_i, *i=1,…,n*, in the model. It is called LLN graph.
(A,I) is recognized as a directed graph of basic operations, where A={A_i, *i=1,…,k*} is the set of these basic operations, I={(A_i,X'_m), i=1,…,k; m=1,…,*n*} is the set of directed information arcs and X' $\subset$ X is the set of the associate variables that we are interested in. The objective of a basic operation is building a loglinear model describing categorical data available.
(X',P), is a Bayesian network, built for. X' $\subset$ X is the set of the associate variables that we are interested in.
P is the set of conditional probabilities.
A* is the decision set. A* is the set of adequate loglinear models.
u: X'$\rightarrow R$ is the utility function, where R represents the real numbers set. Usually χ^2 and G^2 statistics are used as utility functions.
$\delta:$ A$\rightarrow$ A* is the decision rule. Usually the decision rule is based on χ^2 and G^2 statistics.
We distinguish between explanatory and independent variables. Independent variables are classified as explanatory if they are involved in the model under study otherwise they remain independent variables that could be involved in a model. Both the explanatory variables and the response variable from the loglinear model are presented as nodes in the LLP graph. The alphabet of the LLP graph language is as follows:

Z is *the response variable*



X is an *independent variable*. It is equivalent to setting the model parameter equal to zero in the general model.

Y is an *explanatory variable*.

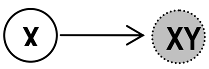
For example,

Response variable of the three-way interaction model

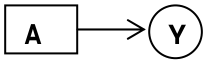
$$\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ} + \lambda_{ijk}^{XYZ}$$

Response variable of the no three-way interaction model

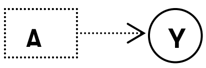
$$\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$$



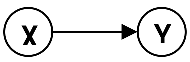
The variable *X* is included in the loglinear model with response *XY* or *X* is an explanatory variable in the loglinear model with response *XY*.



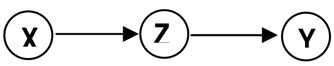
The action *A* is applied to the variable *Y*.



The action *A* could be applied to the variable *Y*.



Variable *Y* depends on variable *X*.



Variables X and *Y* are independent given variable *Z*.

Inference Algorithm

Once the initial knowledge has been presented, one of the most important tasks of a system is to draw conclusions when new information is observed. An algorithm of drawing conclusions about both dependency and probabilistic structure of a Bayessian network is roughed out below.

Inference algorithm

Input: A set of *n* random variables and its graph shell and an *n*-way contingency table (*n*=2,3,…).

Output: A Bayesian network over the set of variables.

Steps: Generate different tasks for building loglinear models starting with most restricted ones (from mutual independence model to saturated model).

1. Build a restricted loglinear model. Go to step 2.

2. Check for adequacy. If the loglinear model is adequate then activate the appropriate arcs in the graph shell and update the probability distributions of the variables of interests according to the newly available information. Go to step 3. If the loglinear model is not adequate then go to step 1.

3.End.

For example, possible results could be the graphs of Bayesian networks presented in Figure 6 and Figure 7.

Model 1: $\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y$

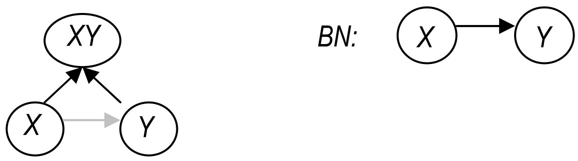
LLN graph: *BN:*

p(*x*,*y*)=*p*(*x*)*p*(*y*)

Assertion: I (X,Y| ϕ)

Model 2: $\log(m_{ij}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_{ij}^{XY}$

LLN graph: *BN:*



p(*x*,*y*)=*p*(*x*)*p*(*y*|*x*)

Assertions: D (X,Y| ϕ)

Figure 6. LLN frameworks for two-way contingency tables.

Model 1: Mutual Independence $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z$

Interpretation: All variables in the model are independent of all other variables in the model.

LLN graph: BN:

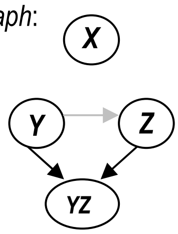
p(*x*,*y*,*z*)=*p*(*x*)*p*(*y*)*p*(*z*)

Assertions: I (X,{Y,Z} | ϕ), I ({X,Y},Z | ϕ), I (Y,{X, Z} | ϕ)

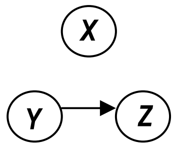
Model 2: Partial Independence $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{jk}^{YZ}$

Interpretation: One factor is independent of the other factors.

LLN graph:



BN:



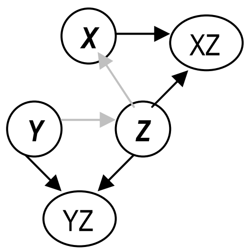
p(*x*,*y*,*z*)=*p*(*x*)*p*(*y*,*z*)=*p*(*x*)*p*(*y*)*p*(*z*|*y*)

Assertions: D (Y,Z| ϕ), I (X,{Y,Z}| ϕ)

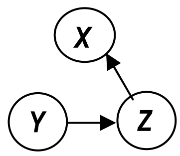
Model 3: Conditional Independence $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$

Interpretation: We make decision for independence of two factors and there is a relationship between both of those factors and the third factor.

LLN graph:



BN:



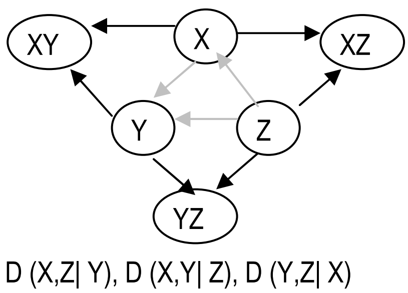
p(*x*,*y*,*z*)=*p*(y)*p*(*z*|y)*p*(*x*|*z*)

Assertions: D (X,Z| ϕ), D (Y,Z| ϕ), I (X, Y | Z)

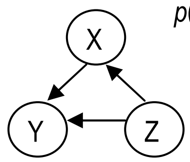
Model 4: No Three-Way Interaction $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$

Interpretation: There is an association between X and Y that is the same for each level of Z; Y and Z have an association that is the same for each level of X, and X and Z have a relationship that is the same for each level of Y.

LLN graph:



BN:

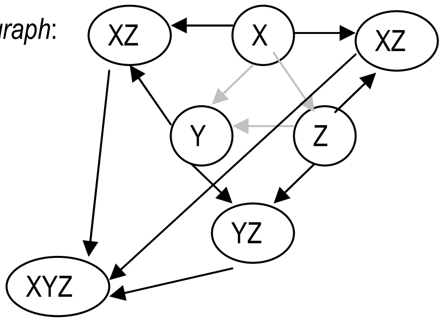


p(*x*,*y*,*z*)=*p*(z)*p*(*y*|x,z)*p*(*x*|*z*)

Assertions: D (X,Z| Y), D (X,Y| Z), D (Y,Z| X)

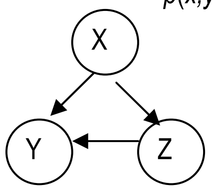
Model 5: Three-Way Interaction $\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ} + \lambda_{ijk}^{XYZ}$

LLN graph:



BN:

$p(x,y,z)=p(x)p(y|x,z)p(z|x)$



Assertions: D (X,Z| ϕ), D (X,Y| ϕ), D (Y,Z| ϕ), D (X,{Y,Z}| ϕ), D (Y, {X,Z}| ϕ), D ({X,Y},Z | ϕ)

Figure 7. LLN frameworks for three-way contingency tables

Conclusion

We have introduced a new graphical representation of loglinear models. We have presented a framework for reasoning with loglinear models. In our framework both dependences between nodes and actions on nodes enjoy a graphical representation. Loglinear networks are graphs with three types of nodes and two types of arcs, representing dependencies and actions. Using this framework we construct a Bayesian network of associate random variables.

Bibliography

[Castillo,Gutierrez,Hadi,1997] Castillo E., Gutierrez J.M., and A.S. Hadi, Expert Systems and Probabilistic Network models, Springer, 1997.

[Noncheva, Marques,2002] Noncheva V., N. Marques. Agent's Belief: A Stochastic Approach, Proc. of the 14th Belgian-Dutch Conference on Artificial Intelligence BNAIC 2002, pp.227-234.

[Paik,1985] Paik, M. A graphic representation of a three-way contingency table: Simpson's paradox and correlation. Amer. Statist., 1985, 39: 53-54.

[Shachter,86] Shachter R.D. (1986). Evaluating influence diagrams, Operations Research, No.34, pp. 871-882.

[Shachter,88] Shachter R.D. (1988). Probabilistic Inference and Influence Diagrams, Operations Research, 36, 591-604.

[Theus, Lauer,1999] Theus Martin, Stephan R. W. Lauer Visualizing Loglinear Models, Journal of Computational and Graphical Statistics, 1999, Volume 8, Number 3, pp. 396-412.

Author information

Veska Noncheva – Faculty of Mathematics and Informatics, University of Plovdiv, 24 Tzar Assen St. 4000 Plovdiv, Bulgaria; e-mail: wesnon@pu.acad.bg

Nuno Marques Departamento de Informática, FCT, Universidade Nova De Lisboa, Quinta da Torre, 2829 - 516 Caparica, Portugal, e-mail: nmm@di.fct.unl.pt

NEURAL APPROACH IN MULTI-AGENT ROUTING FOR STATIC TELECOMMUNICATION NETWORKS

Timofeev A.V., Syrtsev A.V.

Abstract: *The problem of multi-agent routing in static telecommunication networks with fixed configuration is considered. The problem is formulated in two ways: for centralized routing schema with the coordinator-agent (global routing) and for distributed routing schema with independent agents (local routing). For both schemas appropriate Hopfield neural networks (HNN) are constructed.*

Keywords: *centralized and distributed multi-agent routing schemas, Hopfield neural network, data streams distribution*

Introduction

It is usual to consider only one "source-destination" pair of nodes, while stating and solving the routing problem in the local and global telecommunications networks (TCN). In this case of searching for an optimal route, different sourse-nodes (information resourses) and destination-nodes (TCN-clients) parallel work capability is not taking for granted. Therefore, in case of collective (multy-agent) employment of TCN for searching information resourses it is possible that network conflicts and TCN congestions could take place, and casts the TCN efficiency to decrease until the operability is lost. This problem was described in works [Timofeev, 2002] and [*Newton,* 2002].

In this paper we consider a model of multi-agent allocation of information streams in TCN with all participant-agents, involved into the searching and necessary data transfer process, took into account. In statement of the problem there two optimized allocation of data streams criteria are extracted: global, in case of which common TCN capacity is optimized, and local, when data stream allocation is being optimized for each source-destination pair.

By way of efficient computing models for solving such kind of routing problems it is convenient to use Hopfield neural networks (HNN) as it was shown in works [Timofeev, Syrtsev, 2002] and [Timofeev, Syrtsev, 2002a].

For both of optimization criteria an employment capability has been investigated and HNN models were constructed in this work.

Firstly multi-agent data stream routing problems are being considered in two cases:

- for centralized scheme with coordinator ("global optimization"), when decision acceptance is effected by special driving coordinator agent.;

- for distributed scheme ("local optimization"), when every TCN agent is accepting decision independently;

In this paper the possibility of solving stated problems for TCN with limited traffic capacity of communication channels is being considered and proved. Suggested solutions of multi-agent data stream routing in centralized and distributed schemas problems are basing on using HNN adapted to the statement conditions.

In the conclusion main work results are resumed and expanded directions of following investigation are considered.

1. Problem Formulation

As mathematical model of static network, so-called TCN, that doesn't change with the course of time, we will consider a graph

$$G(V, E(V)) \qquad (1)$$

where V – set of nodes, E – ordered set of directed arcs. For given graph, modeling TCN, let's consider a set of its' nodes pairs D_0.

$$D_0 = \{(s,d) \mid s,d \in V, s \neq d\} \qquad (2)$$

Where the first element of paired nodes is necessary data source-node and second one is the destination-node of demanded data. Thus, every multi-agent data stream routing problem is capable to be described as eventual set of such TCN node pairs from D_0. Let's denote this set as D ($D \subset D_0$) and consider it to be fixed, i.e. independent from time.

Let us introduce the following notions:

Π_d – set of paths for pair $d \in D$,

Π – ordered set of all paths for all pairs from D,

Φ_d – rate of information stream between nodes of pair $d \in D$,

$\Phi = (\Phi_1, \Phi_2, \ldots)$ – information streams rate distribution vector,

$\Phi(D) = \sum_{d \in D} \phi_d$ (or Φ_D) – total information streams rate D,

x_p – total information stream rate at path $p \in \Pi$,

$x = (x_1, x_2, \ldots)$ – information streams distribution vector,

δ_{lp} – percentage rate of information stream rate x_p, which pass through arc *l*,

ρ_l – utilization of arc $l \in E(V)$, where $\rho_l = \sum_{p \in \Pi} \delta_{lp} x_p$,

$\rho = (\rho_l, \rho_2, \ldots)$ – utilizations distribution vector,

$T_l(p_l)$ – weighted cost of arc *l*,

T_p – average cost of path $p \in \Pi$,

$T = (T_1, T_2, \ldots)$ – paths costs distribution vector.

In possible data transfer routers Π_d let us specify an incidence matrix

$$\Gamma = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots \\ \gamma_{21} & \gamma_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}\text{-}, \quad \text{где } \gamma_{pd} = \begin{cases} 1, \mathrm{p} \in \Pi_d \\ 0, \mathrm{p} \notin \Pi_d \end{cases} \tag{3}$$

To formalize and solve the problem of optimal routing let's introduce new conditions:

I. Average value of route is determined as sum of its' arcs loading values $\mathrm{T_p} = \sum_{\mathrm{l \in E(V)}} \delta_{lp} T_l(p_l)$;

II. For each arc $l \in E(V)$ is fair that $T_l : [0, \infty) \to [0, \infty]$ and $T_l(0) < \infty$;

III. For each arc $l \in E(V)$, $T_l(p_l)$ is convex and either strongly monotonically increasing on interval, where $T_l(p_l)<\infty$, or $T_l(p_l)$ = const;

IV. $T_l(p_l)$ function is continuous on all definitional domain, moreover on interval, where $T_l(p_l) <\infty$, it is continuously differentiable.

For centralized scheme with coordinator-agent, multi-agent routing problem decision is introducing itself as certain data streams allocation, by which common costs for them are minimal. Let F – common cost of data streams allocation. Then

$$F = \sum_{p \in \Pi} \frac{x_p}{\Phi(D)} T_p = \frac{1}{\Phi(D)} \sum_{l \in E(V)} \rho_l T_l(\rho_l). \tag{4}$$

For every route $p \in \Pi$ and every pair of nodes $d \in D$ following correlations a fair:

$$x_p \geq 0, \qquad \sum_{p \in \Pi_d} \gamma_{pd} x_p = \varphi_d . \tag{5}$$

Thus, the problem statement for centralized multi-agent routing will be represented as follows:

$$\mathrm{F} \to \min \tag{6}$$

with the following constraints:

$$\Gamma^T x = \phi, \qquad x \geq 0 . \tag{7}$$

In distributed scheme of multi-agent routing for each pair $d \in D$ problem is stated separately. In this case an optimal decision is the data stream allocation, at which its' cost for every pair by itself is minimal. Such decision is locally optimal.

Let us introduce concept of minimal stream between pair of nodes d as function: $A_d(x) = \min_{p \in \Pi d} T_p(x)$, $d \in D$. Then, as it was shown in work [*Altman, Kameda*], the decision of problem will be streams distribution vector x, suiting following correlations:

$$\left(T(x) - \Gamma \mathrm{A}(x)\right) \cdot x = 0, T(x) - \Gamma \mathrm{A}(x) \geq 0, \qquad \Gamma^T x - \varphi = 0, x \geq 0\,, \tag{8}$$

where A(x) = (A_1(x), A_2(x), …) – minimal streams vector.

3. Modification of capacity cost on communication channel for TCN with limited traffic capacity

For TCN, where channel traffic capacity is limited, function $T_l(p_l)$ will suit following constraints.

$$T_l(p_l) < \infty, 0 \leq p_l \leq p_{\max}, \quad T_l(p_l) = \infty, p_l > p_{\max}\,, \tag{9}$$

Where p_{max} – maximal traffic capacity of communication channel *l* .

In this case IV condition will be violated. It could be avoided in the event that auxiliary function $T_l^{(\varepsilon)}(p_l)$ is involved :

$$T_l^{(\varepsilon)}(p_l) = \begin{cases} T_l(p_l), p_l \notin [p_{\max} - \varepsilon, p_{\max}) \\ T_l(p_l)\left(1 - \dfrac{\pi}{2\varepsilon}(p_l - (p_{\max} - \varepsilon)) + tg\left(\dfrac{\pi}{2\varepsilon}(p_l - (p_{\max} - \varepsilon))\right)\right), p_l \in [p_{\max} - \varepsilon, p_{\max}) \end{cases} \tag{10}$$

Where amount ε>0 and arbitrarily small.

Lemma: For $T_l^{(\varepsilon)}(p_l)$ of aspect (10) I-IV conditions are held.

Proof: It is evident, that I and II conditions will be held. Let us prove, that $T_l^{(\varepsilon)}(p_l)$ is continuously differentiable on [0, p_{max}).

Let us consider function $z(p_l) = \dfrac{\pi}{2\varepsilon}(p_l - (p_{\max} - \varepsilon))$. It is continuously differentiable on whole definitional domain and strongly monotonically increases, moreover

$$z(p_{\max} - \varepsilon) = 0, \qquad z'(p_{\max} - \varepsilon) = z'(p_l) = \frac{\pi}{2\varepsilon}\,. \tag{11}$$

Let us consider $T_l^{(\varepsilon)}(p_l)$ in point {p_{max}-ε}. Allowing (11) we derive:

$$\begin{aligned} T_l^{(\varepsilon)}(p_{\max} - \varepsilon) &= T_l(p_{\max} - \varepsilon)\left(1 - z(p_{\max} - \varepsilon) + tg(z(p_{\max} - \varepsilon))\right) = \\ &= T_l(p_{\max} - \varepsilon)\left(1 - 0 + tg(0)\right) = T_l(p_{\max} - \varepsilon) \end{aligned} \tag{12}$$

From (10) follows, that in point {p_{max}} function $T_l^{(\varepsilon)}(p_l)$ is tending to ∞, i.e. $T_l^{(\varepsilon)}(p_l)$ is continuous on whole definitional domain. Subject to (10) and (11) we derive following phrase for auxiliary function derivate.

$$\left(T_l^{(\varepsilon)}(p_l)\right)' = \begin{cases} T'_l(p_l), p_l \in (0, p_{\max} - \varepsilon x) \\ T'_l(p_l)\left(1 - z(p_l) + tg(z(p_l))\right) + T_l(p_l) z'(p_l)\left(\dfrac{1}{\cos^2 z(p_l)} - 1\right), p_l \in (p_{\max} - \varepsilon, p_{\max}) \end{cases} \tag{13}$$

This implies, that function $T_l^{(\varepsilon)}(p_l)$ is continuously differentiable on (0, p_{max}))\{p_{max} -ε}. Let us consider limits of its' derivate from the right and left of the point {p_{max}-ε}.

$$\lim_{p_l \to p_{\max} - \varepsilon - 0} \left(T_l^{(\varepsilon)}(p_l)\right)' = (T_l(p_{\max} - \varepsilon))'$$

$$\lim_{p_l \to p_{\max} - \varepsilon + 0} \left(T_l^{(\varepsilon)}(p_l)\right)' = \lim_{p_l \to p_{\max} - \varepsilon + 0} \left(T_l(p_l)'\left(1 - z(p_l) + tg(z(p_l))\right) + T_l(p_l) z'(p_l)\left(\frac{1}{\cos^2 z(p_l)} - 1\right) \right) =$$

$$= (T_l(p_{\max} - \varepsilon))'(1 - 0 + tg(0)) + \frac{\pi}{2\varepsilon} T_l(p_{\max} - \varepsilon) tg^2(0) = (T_l(p_{\max} - \varepsilon))'.$$

As from this follows, that $\lim_{p_l \to p_{\max} - \varepsilon - 0} \left(T_l^{(\varepsilon)}(p_l)\right)' = \lim_{p_l \to p_{\max} - \varepsilon + 0} \left(T_l^{(\varepsilon)}(p_l)\right)'$, then function $T_l^{(\varepsilon)}(p_l)$ is continuously differentiable on $(0,p_{max})$. Hence the condition IV applies for auxiliary function (10).

For condition III to be applied it is enough to prove its' applying on interval (p_{max}-ε, p_{max}). On this interval the function $T_l^{(\varepsilon)}(p_l)$ will be convex and strongly monotonically increasing as the composition of two convex and affirmative functions.

Thus lemma is proved.

This implies, that multi-agent routing optimization methods being considered are applicable towards TCN with communication channels limited traffic capacity. Here optimal routers are constructed with some fallibility.

4. Centralized Multi-Agent Routing Scheme

Let us consider optimization problem (6), (7). As it was shown in work [*Altman, Kameda*], under conditions I-IV this problem has at least one decision. Here for all of the decisions the valuation of traffic distribution vector ρ will be the same.

In work [*Cichocki, Bargiela*] the schema of solving similar systems using neural network was considered, and in work [Timofeev, Syrtsev, 2002] the consideration if realizing this scheme using HNN took place. A sufficient condition for solving the optimization problem (6), (7) with linear constraints using HNN, is strongly monotonically increasing of minimized function ([*Cichocki, Bargiela*]). Let us consider system (4). As for particular set $D \subset D_0$ the value of function Φ_D is positive constant which is possible to be moved to the left part while not changing tasks conditions, i.e.:

$$F\Phi_D = \sum_{l \in E(V)} \rho_l T_l(\rho_l)\,. \qquad (14)$$

Let us introduce matrix Δ:

$$\Delta = \begin{pmatrix} \delta_{11} & \delta_{12} & \cdots \\ \delta_{21} & \delta_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}, \qquad (15)$$

where δ_{ij} – is part of data stream x_j intensity, which falls on arc *i.*

As $\rho_l = \rho_l(x)$, then task (6)-(7) could be reformulated in the following way:

$$(F\Phi_D) \to \min \qquad (16)$$

at following constraints:

$$\Gamma^T x = \Phi, \qquad \Delta x = \rho, \qquad x \ge 0\,. \qquad (17)$$

As possible problem (16) , (17) decisions we will search (ρ, x) vectors. Without generalization loss it is possible to consider, that $0 \le \rho \le 1$ and $0 \le x \le 1$ (similar method of adduction problem to this view is described in [*Cichocki, Bargiela*]).

Let us construct energy function E = E(ρ, x) for HNN, that solves an optimization problem (16), (17). As well we demand function E to be a quadratic form of (ρ, x).

Firstly let us consider function E_0:

$$E_0 = \frac{\alpha_{11}}{2}(\sum_{l\in E(V)} \rho_l T_l(\rho_l))^2 + \sum_{d\in D}\frac{\alpha_{2d}}{2}(\sum_{p\in\Pi}\gamma_{pd}x_p - \varphi_d)^2 + \sum_{l\in E(V)}\frac{\alpha_{2l}}{2}(\sum_{p\in\Pi}\delta_{lp}x_p - \rho_l)^2, \quad (18)$$

where α_{ij} - are certain positive constants, where α_{11} – is sufficiently small value ([Timofeev, Syrtsev, 2002]). However E_0 is not a quadratic form, as the first sum includes non-linear components $T_l(\rho_l)$. Let us replace first sum in (18) with squared linear combination ρ. Then we get following energy function for HNN:

$$E_0 = \frac{\alpha_{11}}{2}(\sum_{l\in E(V)} c_l\rho_l)^2 + \sum_{d\in D}\frac{\alpha_{2d}}{2}(\sum_{p\in\Pi}\gamma_{pd}x_p - \varphi_d)^2 + \sum_{l\in E(V)}\frac{\alpha_{2l}}{2}(\sum_{p\in\Pi}\delta_{lp}x_p - \rho_l)^2 \quad (19)$$

where $c_l \geq 0$ and sufficiently small.

It is important to notice, that the main demand, while constructing energy function for solving similar systems with linear constraints, is sufficiently small value of summand appropriate to the function being minimized ([*Cichocki, Bargiela*]) , in order to approximate solution would not heavily differ from the exact one. Therefore coefficients of linear combination are to be taken sufficiently small. At the same time they shouldn't be assigned to small, as far as it will slow down convergence of searching decision process.

Substituting (18) by (19) is possible, as far as for two strongly increasing functions with same definitional domains (given by system (17)) extremums are reachable in the same points.

Thus HNN model consisting of |E|+|Π| neurons ([Timofeev, Syrtsev, 2002]) is synthesized and adapted to multi-agent centralized schema conditions.

5. Distributed Multi-Agent Routing Scheme

Let us consider a problem of local optimization (8). It was shown in work [*Altman, Kameda*], that if conditions I-IV are applied, the solution of this problem exists. Let us consider systems (8) first equation. By virtue of second and fourth inequations of the system, we get, that for any x following inequation is fair:

$$\left(T(x) - \Gamma A(x)\right)\cdot x \geq 0 \quad (20)$$

This implies, that non-negative function (T(x)-Γy)x accepts zero values (subject to other constraints) in points of possible solutions of equations (20). It implies, that points of minimum for given function and solutions of optimization task (8) are coinciding. Subject to (20) and

$$T(x)x - \Gamma A(x)x = (F\Phi_D) - \Gamma A(x)x \quad (21)$$

Let us reformulate task (8) in the following way:

$$(F\Phi_D) - \Gamma A(x)x \to \min, \qquad T(x) - \Gamma A(x) \geq 0, \quad \Gamma^T x - \varphi = 0, \quad x \geq 0 \quad (22)$$

according to works [*Cichocki, Bargiela*] and [Timofeev, Syrtsev, 2002] task (22) is reducable to the following task:

$$(F\Phi_D) - \Gamma A(x)x \to \min, \qquad T(x) - \Gamma A(x) - z = 0, \quad \Gamma^T x - \varphi = 0, \quad x \geq 0, \quad z \geq 0 \quad (23)$$

We remark, that z_p=0, if $T_p(x) = (\Gamma A(x))_p$. Hence, subject to inequations $x_p \geq 0$ and $z_p>0$ we get, that $T_p(x)>(\Gamma A(x))_p$ and x_p=0.

Let us construct HNN, that solves an optimizing problem (23). As possible solutions we will search vectors (x, z). The same way as in previous paragraph we will assume, that $0 \leq x \leq 1$ and $0 \leq z \leq 1$. From the first parity in system of constraints follows, that

$$\left(F\Phi_D\right) - \Gamma A(x)x = zx \quad (24)$$

Thus, energy function E for HNN will have following presentaion:

$$E = \alpha_{11}(\sum_{p\in\Pi} z_p x_p) + \sum_{p\in\Pi} \frac{\alpha_{2p}}{2}\left(T_p(x) - \sum_{d\in D}\gamma_{pd} A_d - z_d\right)^2 + \sum_{d\in D}\frac{\alpha_{3d}}{2}\left(\sum_{p\in\Pi}\gamma_{pd} x_p - \varphi_d\right)^2 \qquad (25)$$

Appropriate (25) model of HNN consists of 2|П| neurons [Timofeev, Syrtsev, 2002] and adapted to conditions of the local optimization problem for multi-agent routing distributed schema.

Conclusion

Conducted investigations showed adaptability of NN for solving multi-agent data stream routing problem in static TCN, which configurations doesn't change in the course of time. Two models of HNN for global and local data stream distributing optimization were constructed. On demand of TCN agent-users. Besides this, capability of obtained NN solutions adapting to TCN with data stream limited traffic capacity has been investigated.

Common default of NN-routing for distributed and centralized schemas is large amount of beforehand calculations (filler of all possible routes, weights calculations, depending on large amount of parameters, etc.). Thus, it is being planned to avoid incipient hardships and lacks in following investigations.

The work has been supported by the state contract №37.029.11.0027 with RF Ministry of Industry and Science and by grant of RFBR №03-01-0024 and by grant of RGSF №03-06-120196.

Bibliography

[*Altman, Kameda*] *E. Altman, H. Kameda,* Equilibria for Multiclass Routing in Multi-Agent Networks

[*Cichocki, Bargiela*] *A. Cichocki, A. Bargiela,* Neural networks for solving linear inequality systems

[*Pornavalai et al,* 1995] *Ch. Pornavalai, G. Chakraborty, N. Shiratiri,* A neural network approach to multicast routing in real-time communication networks, IEEE, 1995

[*Newton,* 2002] *W. Newton,* A neural network algorithm for internetwork routing, Technical report, 2002

[Timofeev, Syrtsev, 2002] Timofeev A.V., Syrtsev A.V. Neural networks approach in dynamical data stream routing problem. – Proceedings of the "Regional Informatics-2002" Conference (St. Petersburg, 26-28 Nov 2002), ch.1, pp.87-88

[Timofeev, Syrtsev, 2002a] Timofeev A.V., Syrtsev A.V. Adaptive data stream management models based on neural routers Proceedings of the X-th "Neuroinformatics and its applications" seminar (4-6 Oct 2002, Krasnoyarsk)

[Timofeev, 2002] Timofeev A.V. Adaptive data stream management problems and approaches in telecommunicational systems. – Proceedings of the "Regional Informatics-2002" Conference (St. Petersburg, 26-28 Nov 2002), ch.1, p.87

Authors Information

Adil Vasilievich Timofeev – Saint-Petersburg Institute for Informanics and Automation of RAS, 199178, Russia, Saint-Petersburg, 14-th line, 39; E-mail: tav@iias.spb.su

Alexey Vladimirovich Syrtsev – Saint-Petersburg State University, the faculty of mathematics and mechanics; E-mail: airleks@inbox.ru

INTELLECT SENSING OF NEURAL NETWORK THAT TRAINED TO CLASSIFY COMPLEX SIGNALS

Reznik A. Galinskaya A.

Abstract: *An experimental comparison of information features used by neural network is performed. The sensing method was used. Suboptimal classifier agreeable to the gaussian model of the training data was used as a probe. Neural nets with architectures of perceptron and feedforward net with one hidden layer were used. The experiments were carried out with spatial ultrasonic data, which are used for car's passenger safety system neural controller learning. In this paper we show that a neural network doesn't fully make use of gaussian components, which are first two moment coefficients of probability distribution. On the contrary, the network can find more complicated regularities inside data vectors and thus shows better results than suboptimal classifier. The parallel connection of suboptimal classifier improves work of modular neural network whereas its connection to the network input improves the specialization effect during training.*

1. Task description

Experience shows that learning speed and decision's accuracy of complex tasks can be improved using different methods of neural networks combination in the multimodular systems [Sharkey. 1996], [Sharkey et al, 1997]. Great amount of papers is dedicated to understanding the potential capabilities and practical application of modular neural networks. Most of these papers describe homogeneous multimodular structures based on feedforward neural networks. Methods of input data preprocessing and different types decision fusion modules are discussed [Giacinto et al, 2001], [Crepet et al, 2000]. Multimodular architectures based on different neural network types are investigated [Crepet et al, 2000], [Tang et al], [Happel et al, 1994]. Most of researches incline to common conclusion that multimodular neural networks act like a number of experts, which consider task from different positions. Due to such organization decisions based on local reactions of individual neural ensembles are more infallible.

Unfortunately, simple mechanical transfer of collective human expert behavior to the neural ensembles, level of complexity (and intellect) of which could be compared may be only with worm's neural system, hardly appropriate. Of course, neural network is able to learn and take self-dependent decisions, and so we can consider that it has artificial intellect that characterized by outer world models presence. But such interpretation cannot be used for understanding inner procedure of forming and co-ordination of neural modules' decisions.

In our investigation we used sensing method for clarifying factors that define behavior of trained neural network. For this purpose statistically optimal receiver solving the same problem that neural network was used. We proceed from that neural network during training tends to statistically optimal behavior for defined training set and errors criteria. Having statistical distribution for training data it's possible to create statistically optimal device that satisfies defined criteria of work quality and estimates the characteristics of trained network's behavior. Combining such optimal device with neural net in modular structure it's possible to investigate their interaction during training and decision-making. Nature of information features of initial data, which neural network uses for making decision, can be understood by varying different combination methods.

Unfortunately, it's necessarily to have consistent estimates for every combination of training set elements' values to create statistically optimal device. It's almost impossible for big data sets. But if data vectors' elements have weak statistical dependence we can assume that the conditions of central limit theorem are met. In this case the gaussian model of distribution for training set can be a good approximation. This model considers first two moments of distribution: average and covariation. It's easy to obtain the estimations for their values. Having such estimations we can create suboptimal receiver that reflect the main features of training data set.

The goal of this work is experimental investigation of described approach to the analysis of real neural net behaviour. Comparison of behaviour of suboptimal classifier and neural net was carried out with spatial ultrasound signals used in car's passenger safety system. Analysing these signals neural net should estimate level of safety for passenger position and block air-bag deployment if passenger can be damaged. All experiments were carried out with a help of MNN CAD software [Kussul et al, 2002] using data provided by Automotive Technologies International (ATI Inc. Danville, New-Jersey, USA).

2. Description of investigated neural architectures

Five base classifier models were used during experiments:

1. Feed-forward neural network with one hidden layer;
2. Simple perceptron;
3. Linear suboptimal classifier;
4. Quadratic suboptimal classifier;
5. Combination of linear and quadratic suboptimal classifiers.

Experiments were carried out with a help of MNN CAD [Kussul et al, 2002] software that allows creating multimodular structures using different types of neural networks and additional modules. Figure 1 shows the architecture of one of the used methods of modules combination. It's composed of feedforward neural net (module B3) and suboptimal classifier (B4). Fusion module (B5) consists of one neuron. Module B1 is used for input vector normalization, module B2 forms target vectors of all modules during training.

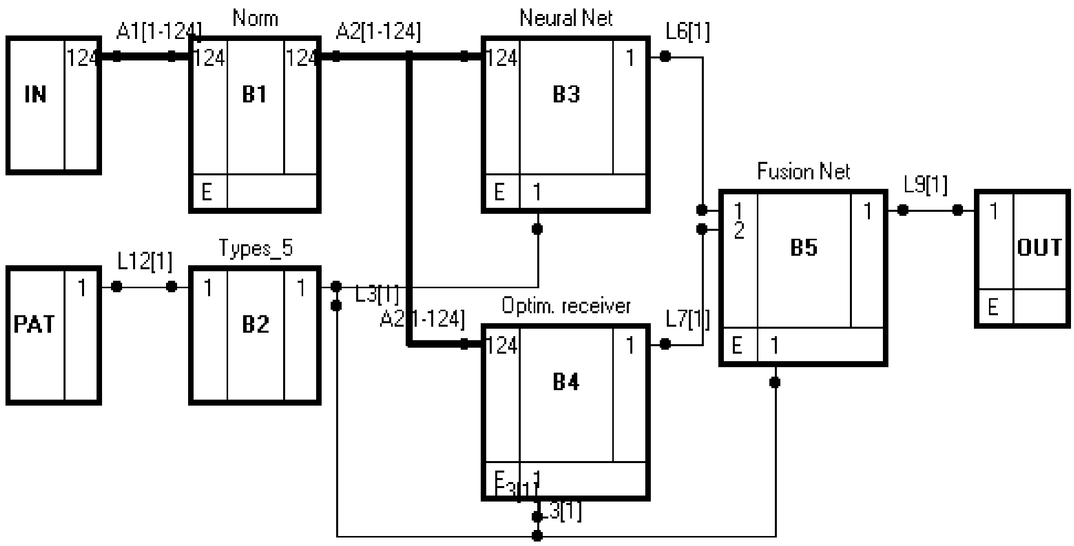


Fig. 1. Multimodular classifier architecture (Var2)

We examined 12 different variants of classifier including 5 base models, combination of feedforward network with perceptron and 6 types of hybrid modular network that combined feedforward net with suboptimal classifier. Two ways of construction of such hybrid network were used. In the first net's variant (Var1) suboptimal classifiers were connected to additional input of neural network that is additional component of input vector was created. In the second structure type (Var2) we used additional fusion neuron with output of neural network and normalized suboptimal classifier's output as its inputs. In both variants neural network was trained after connection of suboptimal classifier that is its reaction took part in the neural network's decision forming.

3. Structure of suboptimal classifier

In our task training data set consists of vectors *X* for two classes "0" and "1". We consider that data is distributed normally. Frequency distribution of probability for *i –th* class is:

$$W_i(X) = \frac{1}{\sqrt{(2\pi)^N |\Psi_i|}} \exp\left[-\frac{1}{2} (X - A_i) \Psi_i^{-1} (X - A_i)^T \right]. \quad (1)$$

Here Ψ_i is the covariance matrix for data of *i*-th class; $|\Psi_i|$ is the determinant of the matrix Ψ_i;

A_i is the average of distribution of *X*; X_i^T is the transposed vector; N is the dimension of the vectors A_i, X_i.

A statistically optimal solution that provides a minimum risk of error is based on a threshold estimate of the value *T(X)* that defines structure of optimal classifier [Middleton, 1960] :

$$T(X) = \ln W_1(X) - \ln W_0(X)\ , \qquad (2)$$

Decision of the situation either "0" or "1" is made after results of comparison of value *T(X)* with the threshold that depends on the relationship between the costs of losses caused by errors to one or the other side. After substitution of expression (1) to (2) obtain:

$$T(X) = \underline{const} + X\left(A_1\Psi_1^{-1} - A_0\Psi_0^{-1}\right)^T - \frac{1}{2}X\left(\Psi_1^{-1} - \Psi_0^{-1}\right)X^T \qquad (3)$$

$$\underline{const} = \frac{1}{2}\left(\ln\left|\Psi_0\right| - \ln\left|\Psi_1\right| + A_0\Psi_0^{-1}A_0^T - A_1\Psi_1^{-1}A_1^T\right).$$

The second and the third terms of the expression that depend on *X* define the linear and the quadratic components of the classifier. The function of the former can be represented by the following sum:

$$S_l = \sum_{i=1}^{N} u_i x_i\ , \qquad (4)$$

where u_i are components of the optimum filter vector:

$$U = A_1\Psi_1^{-1} - A_0\Psi_0^{-1}\ . \qquad (5)$$

The quadratic component function of the classifier is:

$$S_s = \frac{1}{2}X\left(\Psi_0^{-1} - \Psi_1^{-1}\right)X^T\ . \qquad (6)$$

Using expressions (4)-(6) we can create suboptimal classifier with sample estimates of distribution average and covariation matrices for training set. If the covariation matrices values for both classes are similar then optimal classifier consists of only linear part. Also it is possible that linear component (5) is zero so optimal is the quadratic classifier (6).

4. Statistical characteristics of data set

The data vectors used are sequences of 124 short integer values of output signals of four ultrasound sensors. Data was divided into three groups: Train – 128000 vectors; Test – 38400 vectors and Valid – 16800 vectors, which were used for training, testing during training and network validation correspondingly. Each array consists of "0" class vectors (safe passenger position) and "1" class vectors (air-bag deploy should be blocked) with approximately equal amount.

Statistical characteristics of vectors obtained on Train data set are shown on Figure 2. On this figure diagonal elements of the matrices and the boundaries of the sensor areas can be clearly seen. Differences of covariation matrices (on the right window) are the most pronounced classification features of spatial signals.

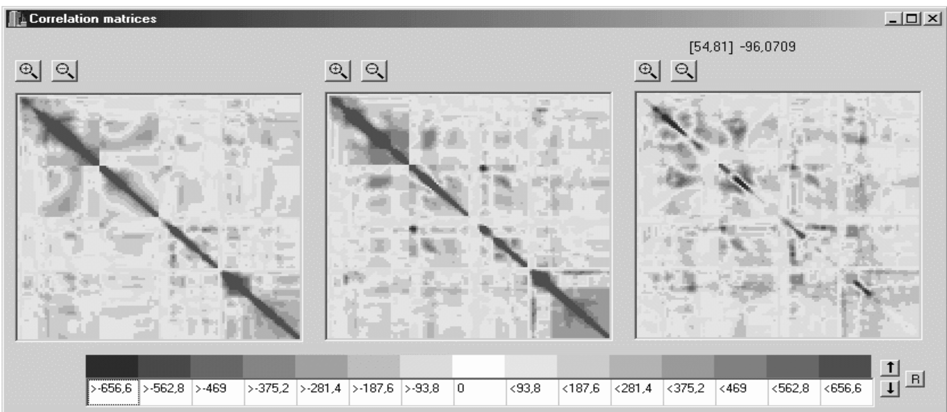


Fig. 2. Covariance matrices for classes "0" and "1" and their difference (on the right).

The results of data analysis shown on Fig. 2 were used to suboptimal classifier's components calculation (4-6). Figure 3 shows values of linear part *U* of filter.

5. Neural modules

Before investigation of modular network with suboptimal classifier in it we chose the best architecture of neural network. Best results were obtained using three-layer neural network with 15 neurons in hidden layer. Sigmoid activation function and EDBD learning algorithm [Reed et al, 1999] were used. Input data was normalized in [0,1] diapason. Normalization was made independently for each input using training data set. Initial values for neural network module were defined by random values. Each experiment was carried out 5 times and results were averaged.

Perceptron module consisted of one neuron with sign activation function and was trained using Hebb learning rule. Initial weights for this module were zero. Weight coefficients obtained during training are shown on Fig. 3. The same figure shows weights of suboptimal linear filter (5). Position of extreme values is similar for both charts. Suboptimal filter has strong extremes only in the range of 64-67 inputs. In contrast to this values of input's weights for perceptron distributed more equally.

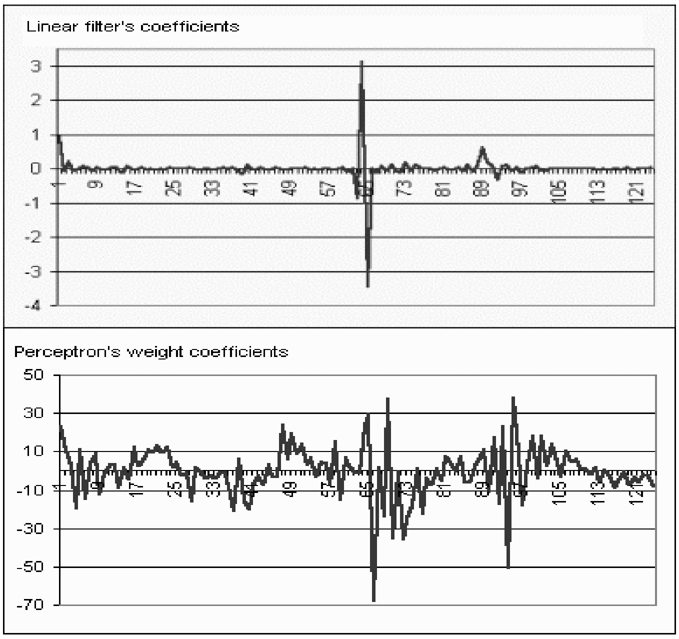


Fig. 3. Weight coefficients of perceptron and suboptimal linear filter

6. Results of independent modules' testing

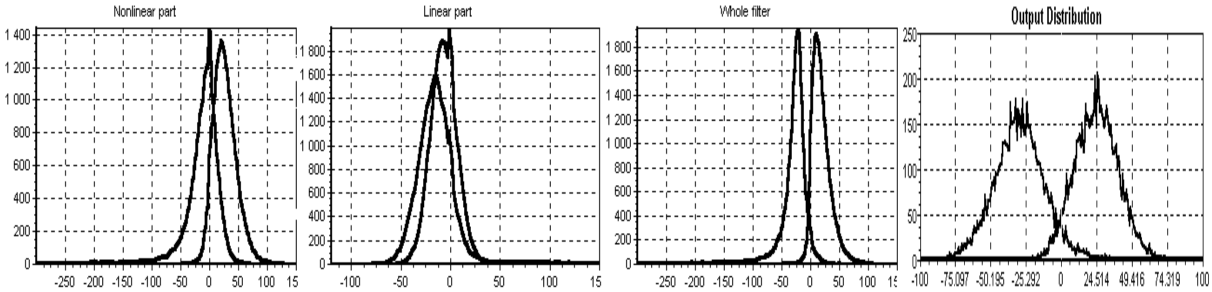
Table 1 shows comparative testing results for base classifier's models on Train and Valid data sets.

Table 1. Error rate for base modules.

Type	Train %	Valid %
SO/Linear	42.15	37.92
SO/Square	18.25	26.85
SO/Lin. + Sq.	4.81	5.92
Perceptron	6.02	4.83
Neural network	1.98	3.17

Table shows that feedforward neural network gives best results. Perceptron and suboptimal classifier (even with both linear and quadratic components) show much worse success rates. Suboptimal classifier shows better results than perceptron on training data set but much worse on validation set. So we can conclude that neural network during training finds more complex associative relations than contained in average values and mutual correlation functions. To some extent the same suggestions can be applied to perceptron, decisions of which are based on linear transformation of input data. This transformation is similar to that linear suboptimal classifier uses.

Figure 4 shows histograms of reactions of suboptimal classifier and perceptron for two data classes. Diagrams correspond (left to right) to the nonlinear component, linear component, full suboptimal classifier and postsynaps values of perceptron. Under each diagram there is the threshold value and the average number of correct decisions for each situation class. It can be seen that results for separate linear and quadratic classifiers are much worse than for full classifier. Distribution of potential at the perceptron's input is similar to the distribution of reaction of full suboptimal classifier.



0 / 1	19.29/ 17.81	25.55 / 58.75	5.48 / 4.14	6.23 / 5.82
Total %	18.55	42.15	4.81	6.02

Fig. 4. Histograms of reactions for suboptimal classifier and perceptron. Left to right: linear, quadratic, full suboptimal classifier, perceptron (Train data set).

7. Experiments with hybrid networks.

The goal of experiments was to understand the importance of classification features used by suboptimal classifier. We tested hybrid networks Var1 and Var2 with linear, quadratic and full suboptimal classifiers. Thus neural network before training had all useful information that suboptimal classifier extracts from input data. Also we investigated variant of hybrid network Var2 where already trained perceptron was used instead of suboptimal classifier.

Training was carried out using Save Best method until 1.5M vectors training depth. Best result was saved. Each experiment was carried out 5 times with different weight initialization. In each series of experiments average and minimum error rates were calculated.

Experimental results are shown in Table 2. Data in the last row of table for Var2 were obtained while autonomous testing of neural network within hybrid network Var2.

Table 2. Experimental results for two types of hybrid network

Mode	Test mode	Var1		Var2	
		Avg.%	Min%	Avg.%	Min%
SO/ Linear	Train	2.02	1.78	1.88	1.75
	Test	4.39	4.10	4.34	4.29
	Valid	3.59	3.32	3.53	3.33
SO/ Square	Train	2.13	1.74	1.92	1.79
	Test	4.35	4.17	4.39	4.18
	Valid	3.49	3.18	3.46	3.20
SO/ Lin. + Sq.	Train	1.87	1.64	1.74	1.63
	Test	4.31	4.23	3.98	3.77
	Valid	3.56	3.38	2.97	2.67
Perceptron	Train			1.89	1.76
	Test			4.28	4.22
	Valid			3.26	3.13
Neural network	Train	2.23	1.98	2.09	1.94
	Test	4.31	4.29	4.49	4.15
	Valid	3.48	3.17	3.67	3.28

Testing results on Train data set show that connection of suboptimal classifier leads to decreasing of classification errors for both types of hybrid network. Best results were obtained using both linear and quadratic components of suboptimal classifier. But for Test array there is no such improvement and error rate on Valid data set even increased for Var2 network. Testing of Var2 network on this array gives opposite result – error rate decreases almost on one third.

Connection of perceptron in Var2 network also decreases error rate for all data arrays. But success rate is worse than with connection of suboptimal classifier.

Quantitative estimation of information level of suboptimal classifier's decision can be obtained by comparison of weights values for the inputs of fusion neuron in Var2 network. Table 3 shows these estimations. We used such symbols: W_{NN} – weight value for neural network's output; W_{SO} – weight value for suboptimal classifier's output; $k=W_{SO}/(W_{NN}+W_{SO})$ – information level for corresponding component. Information level is about 15% for linear component of suboptimal classifier, for quadratic it's much higher – more than 25%. Naturally highest information level was obtained for sum of components – more than 37%. Estimation of information level for perceptron appears enough unexpected – less than 3%.

Table 3. Estimation of information level for gaussian features in hybrid neural network

Mode	W_{NN}	W_{SO}	$k = W_{SO}/ (W_{NN}+ W_{SO})$
SO/ Lin.	4.92	0.91	0.156
SO/ Sq.	4.74	1.63	0.256
SO/ Lin. + Sq.	4.15	2.49	0.376
Perceptron	3.95	0.11	0.027

8. Conclusion

Distribution of used experimental material is close enough to multidimensional gaussian distribution. So we could expect that neural network will use mainly the same classification features that suboptimal classifier does. But neural network behavior shown is far from that. Neural network finds complex associative relations between data elements during training. But it does not fully make use of more simple correlation dependencies used by suboptimal classifier. As it can be seen in Table 2, it's obvious that connection of suboptimal classifier always decreases error rate of neural network. Success rate improvement is the most when suboptimal classifier is connected to fusion neuron (Var2). Improvement effect in this case is the strongest for Valid data set. This is a very important result. It shows that generalization ability was improved. Also we can conclude that decision-making criteria used by neural network and suboptimal classifier are relatively independent.

If suboptimal classifier is connected to neural network input (Var1) success rate increases only for Train data set. It shows that specialization effect becomes stronger due to suboptimal classifier's connection. We found such effect for the first time and have no explanation for it yet. We can only suppose that neural network's hidden layer blocks information flow from suboptimal classifier to network's output. The nature of this phenomenon is unclear but we can make practical suggestion that the most informational features should be connected closer to output of hybrid neural network.

Low information level of perceptron connected to fusion neuron appeared quite unexpected. Its weight is less than 3% that is much less than for linear and quadratic suboptimal classifiers (15% and 25%). This is opposite to results shown in Table 2. We can suppose that in this case information features formed by neural network have the same character that for perceptron but they are much more powerful. So perceptron contribution to the final decision is insignificant.

Acknowledgments

The work is supported by INTAS grant 2001-0257 "Smart Sensors for Field Screening of Air Pollutants". The authors wish to thank David Breed and ATI Inc. for the giving data sets.

Bibliography

[Sharkey, 1996] Amanda J.C. Sharkey, On combining artificial neural nets. - Connection Science. - 1996. - V. 8. - N. 3/4. - P. 299-314.

[Sharkey et al, 1997] Amanda J.C. Sharkey, Noel J. Sharkey, Combining diverse neural networks. - The Knowledge Engineering Review. - 1997. - V. 12: 3. -P. 1-17.
[Giacinto et al, 2001] Giorgio Giacinto, Fabio Roli, Design of effective neural network ensembles for image classification purposes. - Image and Vision Computing. -Elsevier Science. - 2001. - V. 19. - P. 699-707.
[Crepet et al, 2000] Agnes Crepet, Helen Paugam-Moisy, Emanuelle Reynaud, Didier Puzenat, A modular neural model for binding several modalities/ In Proc. of IC-AI'2000, Int. Conf. of Artificial Intelligence. - Las Vegas, USA. - 2000. - P. 921 – 928.
[Tang et al] Bin Tang, Malcolm I. Heywood, Michael Shepherd, Input Partitioning to Mixture of Experts
[Happel et al, 1994] Bart L.M. Happel, Jakob M.J. Murre, The design and evolution of modular neural network architecture. - Neural Networks. – 1994. – V. 7. – P. 985-1004.
[Kussul et al, 2002] Kussul M., Riznyk A., Sadovaya E., Sitchov A., Chen T.Q. A visual solution to modular neural network system development // Proc. of the 2002 International Joint Conference on Neural Networks IJCNN'02. - Honolulu, HI, USA. - 2002. - 12-17 May. - Vol.1. - N749.
[Middleton, 1960] D. Middleton, An Introduction to Statistical Communication Theory, V. 2., 1960.
[Reed et al, 1999] R.D. Reed and R.J. Marks, Neural Smithing. MIT Press. - 1999. – P. 346.

Author information

Reznik A., Galinskaya A. - Department of neural technologies, Institute of Mathematical Machines and systems NAS Ukraine. neuro@immsp.kiev.ua

APPLICATION OF THE SUFFICIENCY PRINCIPLE IN ACCELERATION OF NEURAL NETWORKS TRAINING

Krissilov V.A., Krissilov A.D., Oleshko D.N.

Abstract: *One of the problems in AI tasks solving by neurocomputing methods is a considerable training time. This problem especially appears when it is needed to reach high quality in forecast reliability or pattern recognition. Some formalised ways for increasing of networks' training speed without loosing of precision are proposed here. The offered approaches are based on the Sufficiency Principle, which is formal representation of the aim of a concrete task and conditions (limitations) of their solving [1]. This is development of the concept that includes the formal aims' description to the context of such AI tasks as classification, pattern recognition, estimation etc.*

Keywords: *neural networks*

Introduction

Nowadays developers have a lot of different models of neural networks and algorithms of their training [2, 3] for disposal. Though the scientific researches are permanently carried on in this field, the theory of neural networks is still feebly formalised. However, even now two stages of creation of artificial neural systems could be defined: structural and parametric synthesis. At the first stage, developer has to do the following: choose the model for the network, define its structure and choose the algorithm for its training. The parametric synthesis includes training processes of the created network and verification of the obtained results. Then, depending on verification results, there can be a necessity of return to one of the stages of structural or parametric synthesis. Thus, becomes obvious that creation of the neural system is an iterative process.

Feeble formalisation of these stages results in necessity for the developer of the neural system to solve a number of problems. E.g., at the structural synthesis stage, in case of solving a non-standard task, it is necessary to spend a lot of time for choosing the corresponding model for the network, choosing its structure and training method. The problem of the parametrical synthesis is a considerable training time. If real tasks are being solved without any simplification, then duration of training process for created network could be too long. However, some tasks require spending as less training time as it is possible, e.g., real-time tasks.

The aim of the given article is to offer possible methods to reduce the training time for neural networks with back propagation training algorithm. As such methods are offered: control of procedures of modification and evaluation of weight coefficients, reorganisation of objects in recognition classes. Two possible ways for solving this problem were offered in [4]. The first one was based on choosing the particular functional base for the network. The second method controlled the value of the step of weights modification, considering it from the point of view of a centrifugal force and, adjusting it so that its vector was always directed on an optimum of the set of weights.

In this paper the given problem is considered from the point of view of overtraining the network. In most cases a neural network is trained, while its error will not become equal to zero. It can result in inadmissible spending of time. Though, for most tasks it is *enough* for this error not to exceed some defined value.

Sometimes the level of sufficiency is determined by conditions of the task and required result. However, in most cases this process flows past at an intuitive level and the guided principle is not sufficiently fixed by us. Actually this moment is one of most important in solving similar problems, and optimal value of the varied parameter can depend on many basic values and limitations of the task. Thus, there is a necessity for formalising the given principle, in further – the Sufficiency Principle (SP).

Using SP for training neural networks

Let's consider training of the multilayer back propagation neural network within the frames of solving the classification problem.

Three kinds of errors can be picked out in the training process. Let's name them Elementary Error, Local Error and Global Error. The Elementary Error is the error of a single neuron of the network, for neurons from the output layer it can be evaluated as follows:

$$e_i = Y_i - A_i \qquad (1)$$

where Y_i – standard value, A_i – neuron activation level.

The Local Error is the common average error for all neurons of output layer on a single iteration.

$$E_{Li} = \sqrt{\frac{\sum_{k=1}^{m} e_k^2}{m}} \qquad (2)$$

where m – number of neurons in the output layer of the network, i – number of training process iteration.

The Global Error is obtained as it is shown below:

$$E = \sqrt{\frac{\sum_{i=1}^{n} E_{Li}^2}{n}} \qquad (3)$$

where n – number of training sets in the training sample.

The neural network is considered to be ideally trained if its Global Error is equal to zero [5]. However, usually it is difficult to train the net to such level and sometimes it is even impossible. These hardships are connected with presence in the training sample of similar training sets. Thus, the more of such sets are in the sample, the harder will be to train the net.

The essence of the SP is the rejection from attempt to reach the Ideal in solving the concrete task. Considering the training process from the point of view of SP and Local Error, it is possible to say, that complete recognition ($E_L = 0$) is not always necessary. Usually, in order to refer some object to a specific class, the Local Error just shouldn't exceed some defined δ.

Thus, in the frames of errors considered above, three kinds of applying the SP are represented below. The first one offers to accept the error of a single neuron equal to zero if its Elementary Error lies within some boundaries ($e_i \leq \delta_e$; δ_e – elementary sufficiency parameter). The second considers the Local Error of the neural network. If E_{Li} is less or equal to δ_{EL} (δ_{EL} – local sufficiency parameter), then procedure of recounting the weights won't be applied for this training iteration. And the last one offers to stop the training process after the Global Error of the network will reach value of some δ_E (global sufficiency parameter).

The minimal value of each δ depends on kind of the training sample. Let's consider the following characteristics of the sample: its completeness heterogeneity, and contradictoriness. The completeness is characterised by provision of classes with training sets. The number of training sets for each class should be in 3 – 5 times more, than number of its *features* used in the set [6]. Let's evaluate the value of completeness as follows:

$$F_{TS} = \frac{N_F}{N} * 100\ \% \qquad (4)$$

where N_F – number of classes satisfying to the condition mentioned above; N – number of all classes.
The heterogeneity shows how uniformly the sets are distributed among classes. In order to obtain its value let's take the number of training sets for the i-th class $[C_i]$. Then the mean deviation of this value on sample for the given class is:

$$\overline{\Delta}_{C_i} = \sqrt{\frac{\sum_{k=1}^{N}\left([C_i]-[C_k]\right)^2}{N-1}}; \quad k \neq i \qquad (5)$$

Let's evaluate the average of distribution for $\overline{\Delta}_{C_i}$ and $[C_i]$, on condition that values are equiprobable:

$$R_\Delta = \frac{\sum_{k=1}^{N}\overline{\Delta}_{C_i}}{N}; \qquad R_C = \frac{\sum_{k=1}^{N}[C_k]}{N} \qquad (6)$$

Then heterogeneity can be evaluated as following:

$$H_{TS} = \frac{R_\Delta}{R_C} \qquad (7)$$

Contradictoriness is a rate of conflicting sets in the training sample. Conflicting sets have the same features, but distributed to different classes. Thus, contradictoriness can be obtained as following:

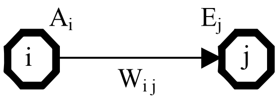
$$I_{TS} = \frac{N_I}{N} \qquad (8)$$

where N_I – number of conflicting sets.
It is obvious, that the lower contradictoriness and heterogeneity, the more narrow can be intervals δ.
Proposed procedures allow to reduce the number of idle changes of weight coefficients. Thus they speed up an approximation of the weights' set to its optimum.

Adjusting the step of weights modification

In original the expression for changing weights between neurons i and j is as following [7]:

$$W_{ij}^{t+1} = W_{ij}^{t} + \alpha * E_j * A_i^t \qquad (9)$$



Pic. 1

where E_j – the error of the j-th neuron;
A_i – the activation level of the i-th neuron;
α – the step of weights modification.

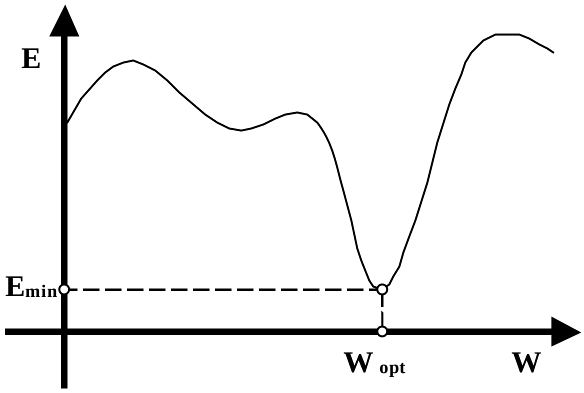
In (9) α is a constant value. However, it is obviously, that if α will be too small, then training will last too long. On the other hand, if α is big, then when the network comes near the minimum point of the error function $E = f(W)$ (E – the Global Error; W – the set of weights) (Pic. 2), it won't be able to reach it. The network will continuously oscillate around this point re-counting its weights and only making worse its characteristics.

Thus it is necessary to manage the value of α. It is obvious, that if W^{opt} should be reached for the minimal number of iterations, then some average value of α is not acceptable.
Then, at the beginning of the training process some maximum value for α should be set. It will provide a quick approximation to the area of W^{opt}. During the approximation the value of α should be gradually decreased.

$$\alpha_0 = \alpha_{\max}; \quad \alpha_{t+1} = \alpha_t - \partial\alpha \qquad (10)$$

where $\partial\alpha$ – is decrement of the α.

The offered method of dynamical adjusting the step of weights modification allows keeping the speed of error's decreasing on a sufficient and satisfactory level.



Pic. 2

Reorganization of recognition classes

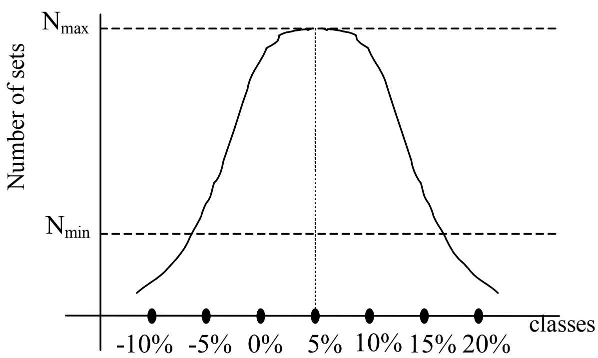
There are number of AI tasks which suppose of possibility of reorganising objects between classes and classes themselves, e.g. creating the forecast based on analysis of time series. This provides two ways for acceleration of the training process in this case. The aim for these ways is to perfect the training sample's characteristics.

There are number of AI tasks which suppose of possibility of reorganizing objects between classes and classes themselves, e.g. creating the forecast based on analysis of time series. This provides two ways for acceleration of the training process in this case. The aim for these ways is to perfect the training sample's characteristics.

Let's consider reducing the number of recognition classes. It is known that the smaller a neural network is, the quicker will be its training. For back propagation neural network its structure is defined by created training sample: the number of recognition classes uniquely defines the number of neurons in the output layer. Thus, reducing the number of classes results in decreasing the size of the network.

However, there is a big number of real tasks, where such losses in precision of classification are inadmissible. Thus, this method can be applied only for tasks without tight restrictions on precision.

It is offered to reduce the number of classes by their combining. In order to find classes for combining, it is necessary to analyse the completeness and heterogeneity of the training sample. If the number of training sets for some class doesn't satisfy the completeness condition, or it is greatly less than in other classes, then recognition of this class by the network will be difficult. For example, results obtained after analysis of the training sample can be the classical normal distribution looking as it is shown on Pic. 3. In order to decrease the heterogeneity of the training sample, classes with number of sets lower than some $N_{\min}$ should be taken and then neighbouring classes should be combined. Then the number of training sets will get over the barrier of $N_{\min}$ and network will be able to train qualitatively and quickly. However, it will also results in reducing the precision in solving the given task. Thus, it is necessary to adjust, using SP, the number of classes recognised by the network with its size.



Pic. 3

Further, let's consider the contradictory training sample. In such a sample classes have both:
objects with low dispersion and located close to the standard of this class – Rules, and objects remote from the standard and located somewhere near the class' boundaries –Eliminations.

Also in the sample there can be classes with high dispersion inside, for which it is impossible to find the standard. Eliminations and Fuzzy classes increase the contradictoriness of the training sample, essentially slow down the training and sometime make it ever impossible. Presence of such elements in the sample can indicates that subsetting of the objects' space on classes was wrong. The solving of this problem is moving Eliminations to other classes and/or forming new classes with lower dispersion.

Thus, the training speed of the network can be increased either by reducing the number of recognition classes, or by moving objects among classes and by forming new classes. The second way increases the training speed by perfecting the training sample, and the first one also by reducing the size of the network.

Conclusions

Thus, three ways of accelerating of the training process for back propagation neural network were considered in this paper.

The first way is based on the analysis of networks' errors. Three levels of errors were described: Elementary Error, Local Error and Global Error. Depending on the kind of the analysed error, different algorithms and software procedures of their implementation were created for obtaining values of the network's weights.

The second way consists in dynamic adjusting the step for changing values of the network's weights. The aim of this method is a minimisation of number of training iterations by reducing the inconsistent adjustments of weights.

The third way considers the reorganization of objects in recognition classes as the way of perfecting characteristics of the training sample: completeness, contradictoriness and heterogeneity.

All proposed ways were applied in forecast and pattern recognition tasks and have brought positive results. They have shown ability to decrease the number of iterations of the training process.

As the test case the task of forecasting the residuals on the bank accounts was solved. The training time was about 30 – 40 hours that was two times less in comparison with original methods.

Applying of them has allowed creating the forecast (for two weeks horizon) with mean-root-square error not greater than 4%.

Bibliography

[1] Krissilov, V.A., Krissilov, A.D. “High-Quality Decision Making by Aim-Oriented Modeling”, Proc. of 19-th International Conference of NAFIPS, Atlanta, GA, 2000, pp.241-245

[2] Krissilov, V.A., Oleshko, D.N., Trootnev, A.V. “Applying of neural networks in tasks of intellectual analysis of information”, Review of Odessa State Polytechnic University, Vol.2 (8), 1999, pp. 134-139

[3] Dayhoff J. Neural network architectures, New-York: Van Nostrand reinhold, 1991.

[4] Patrick, P. “Minimisation methods for training feed forward Neural Networks”, NEUARAL NETWORKS, 1994, Volume 7, Number 1, pp. 1-11

[5] Fausett, L. Fundamentals of Neural Networks. New York: Prentice Hall, 1994.

[6] Bishop C. Neural Networks for Pattern Recognition, Oxford: University Press, 1995.

[7] Patterson D. Artificial Neural Networks, Singapore: Prentice Hall, 1996.

Author information

Victor A. Krissilov – Ph. D., Head of Chair "System Software Design" of Odessa Polytechnic University, App.36., 20 Deribasovskaya Str, Odessa-26, 65026, Ukraine; E-mail: VictorK@OL405.paco.net

Anatoly D. Krissilov – Ph. D., Senior Researcher, Institute for Market Problems and Econo-ecological Research of Ukrainian National Academy of Science 29 Francuzskij Blvrd., Odessa-44, 65044, Ukraine.

Dmitry N. Oleshko – post graduate student of Odessa Polytechnic University; E-mail: boss@ic.ospu.odessa.ua

MULTI-AGENT SECURITY SYSTEM BASED ON NEURAL NETWORK MODEL OF USER'S BEHAVIOR

N. Kussul, A. Shelestov, A. Sidorenko, V. Pasechnik, S. Skakun, Y. Veremeyenko, N. Levchenko

***Abstract**: It is proposed an agent approach for creation of intelligent intrusion detection system. The system allows detecting known type of attacks and anomalies in user activity and computer system behavior. The system includes different types of intelligent agents. The most important one is user agent based on neural network model of user behavior. Proposed approach is verified by experiments in real Intranet of Institute of Physics and Technologies of National Technical University of Ukraine "Kiev Polytechnic Institute".*

***Keywords**: neural network, multi-agent system, network security system, user behavior model, intrusion detection system.*

Introduction

During last decades information technologies based on the computer networks play an important role in various spheres of human activity. Problems of great importance are entrusted on them, such as keeping, transmission and automation of information processing. The security level of processed information can vary from private and commercial to military and state secret. Herewith the violation of the information confidentiality, integrity and accessibility may cause the damage to its owner and have significant undesirable consequences. Thus the problem of information security is concerned. Many organisations and companies develop security facilities that require significant contributions. On the other hand, the impossibility of creating completely protected system is a well-known fact – it will always contain mistakes and «holes» in its realization.

To protect computer systems such accustomed mechanisms as identification and authentication, mechanisms of the delimitation and restriction of the access to information and cryptographic methods are applied. However they possess following drawbacks:

- exposure from internal users with malicious purposes;
- difficulties in access differentiation caused by information resources globalisation, which washes away differences between "own" and "foreign" subjects of the system;
- reduction of productivity and communication difficulties due to mechanisms for access control to the resources, for instance, in e-commerce;
- simplicity of passwords definition by making combinations of simple users' associations.

Therefore logging and audit systems are used along with these mechanisms. Among them are Intrusion Detection Systems (IDS).

The Intrusion Detection Systems

IDS are usually divided to systems detecting already known attacks (misuse detection systems) and anomaly detection systems registering the life cycle deviations of the computer system from its normal (typical) activity. Besides, IDS are subdivided to network-based and host-based types by information source. Herewith they can be as real-time (online), so offline.

Network-based IDS analyse network dataflow, protecting its participants, practically not affecting the productivity of their work. Network-based systems do not use information about processes from separate workstation. In turn, the host-based systems are installed on the separate computers and analyse information from their logging mechanisms.

If IDS is real-time an attack can be registered on the stage of its preparation and warned on the stage of its generation (that is more preferable). In this case there is no need to store large amounts of logged data. However the real-time host-based IDS may vastly influence upon the system productivity.

In contrast there are offline systems, which, as a rule, are activated at night or at any other time, when workstation load is low. Thereby, they do not use system resources, necessary for other tasks. Their drawbacks: to analyse information it is necessary to save sufficient amount of audit-data logged during observation, and reaction on attacks is greatly remitted.

At present a lot of IDS are developed. Among them are: Haystack, GrIDS, NIDES, ASAX, DARPA, EPIC2, snort and others. They have made the significant contribution to development of IDS. These systems are based on different algorithms. The main trends are:

- Building activity graphs (Graph-based Intrusion Detection System – GrIDS) in which nodes represent hosts and edges represent network activity among them. The detection technique is to compare graph to a known pattern of intrusive activity.
- Statistical deviation detection methods (Next Generation Intrusion Detection Expert System – NIDES). These systems are the prime examples of anomaly detection systems.
- Employing expert evaluations. In this approach more scalability is achieved by hierarchical arrangement of the expert systems (Extensible Prototype for Information Command and Control – EPIC2).

Main drawbacks of the described IDS are:

- high probability of the false positive and false negative warnings;
- primitive mechanisms of determining new, unknown in advance intrusions;
- unstable reaction to distributed attacks;
- need of human expertise during all the working time.

To eliminate such defects new approaches were developed. They allow to build completely or highly automated IDS [1]. These approaches are mainly directed on “intellectualisation” of IDS. Among them:

1. Use of neural networks [2,3], genetic algorithms, utilising variable-sized Markov chains [4] etc.
2. Systems based on agent approach [1,5].

It is known, that approximately 70% of attacks are initiated from the inside of network. It might be as password stealing, so using vulnerabilities of information security and the software. So, modern approaches actively use the user behavior model.

Developing IDS it is also necessary to take into account distributed nature of attacks on computer network. All these factors show agents approach to be more preferred for creating the security systems.

Agent paradigm

The agents system is meant to be the system of interacting agents. They are coordinated by general global purpose (the strategy) but autonomous enough to realize their own tasks within the framework of the general strategy (the own tactics).

Importance of transition to agent paradigm is compared with importance of using object-oriented approach. Agent technology can be effectively applied in different areas of information technology, e.g. computer networks, software development, object-oriented programming, artificial intelligence, human-machine interaction etc.

Main advantages of intelligent agent systems are as follows.

- **Distribution**. Functional independence of system parts, ability of solving heterogeneous tasks from all domains.
- **Intelligence**. The ability to adapt to the changing environment.
- **Scalability**. Property that makes possible solving new tasks without bringing significant changes to the system architecture.

System structure and functionality

Integrated network IDS should detect different attack types (known and unknown) and anomaly activities. To meet these requirements it should contain various (rather autonomous) interactive modules. Such architecture can be implemented on the base of agent approach. An important role is played by User Agent that should monitor the user behavior and detect anomalies in its activity. Other types of agents are responsible for other aspects of security.

- **User Agent.** This agent allows detecting anomalies in user activity on base of neural network user model. It predicts user actions on the base of the model and compares them to real activity. But we should take into account that behavior of the same user differs for various operating systems. Consequently, User Agent is developed for each type of operating system available in the network (e.g. Win2000, 98, XP, Free BSD).
- **Host Agent.** Performs system calls processing and detects anomalies and known types of attacks. For example, it allows detecting “Trojan horse” attacks.
- **Network Agent.** Operates at the firewall and analyses the network traffic. The information extracted from packets is used to detect known attacks and anomalies in the network. It may be done utilising neural network and probability approaches (e.g. Bayesian networks and variable-sized Markov chains).
- **Server Agents.** Group of agents is responsible for the server security.
- **Controller Agent**. Responsible for anomalies analysis and detection of distributed attacks in scale of whole system, initialising agents, interaction with database and between various parts of the system.

The structure of proposed system is shown in Fig. 1.

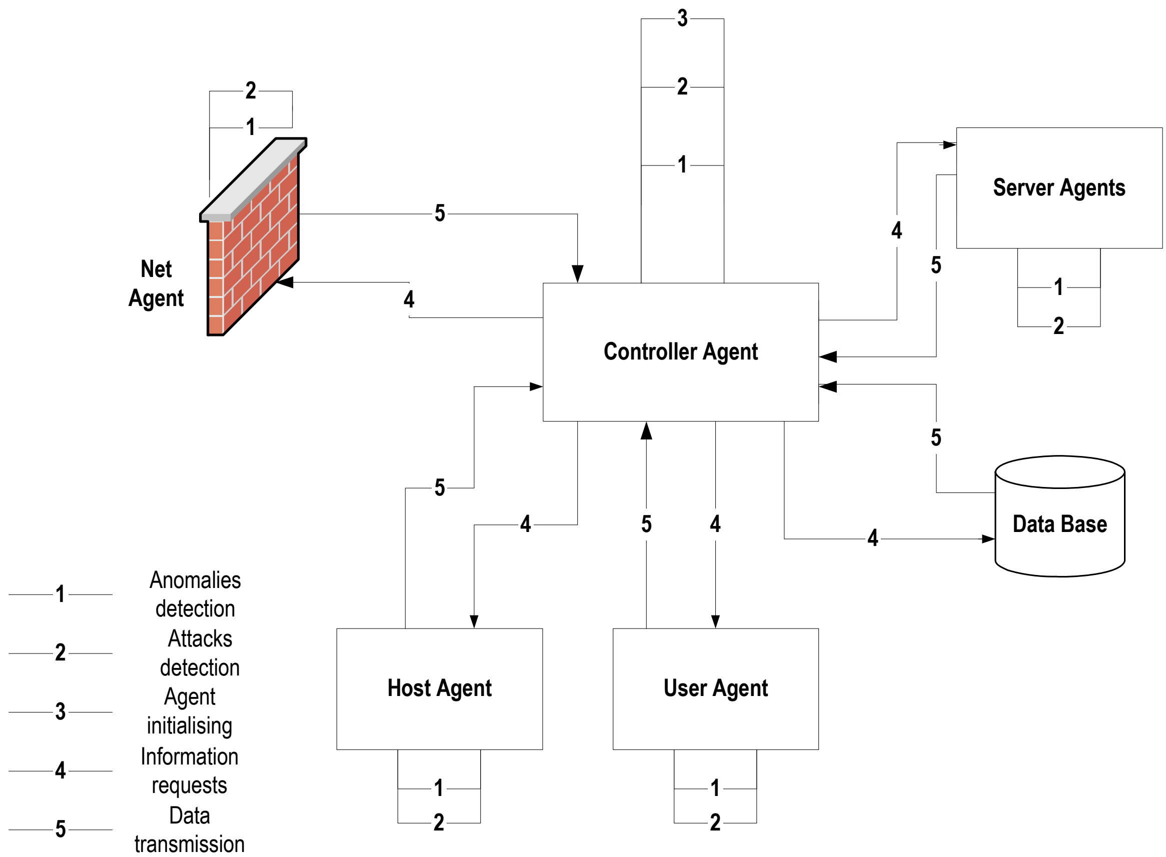


Fig. 1. System structure.

As the user logs on, Controller Agent creates correspondent User Agent and initialises it. During the user session agent controls the user's activity on the base of neural network behavior model. At the same time it picks data for behavior model correction. When the session is finished it sends data for database update. In the case of anomaly detection User Agent informs Controller Agent about suspicious activity. Host Agents and Server Agents detect system anomalies and known attacks.

Experimental results

Efficiency of suggested approach is confirmed by experimental results. We have built neural network user behavior model for operating system FreeBSD [3]. For this purpose we applied the feed forward neural network that was trained to predict a command by given number of previous ones. The experiments were carried out Intranet of Institute of Physics and Technologies of National Technical University of Ukraine "Kiev Polytechnic Institute". About 4000 user behavior models were analysed. Experimental results confirmed such models to be capable to detect anomalous user activity. Taking into account this experience we propose to spread given approach on other operating systems.
Neural network user behavior model was applied for operating system Windows 98. Initial information for neural network was process sequences run in the system. Neural network was to predict running process by the previous ones. The criterion for the optimal neural network prediction is to distinguish appropriate user from others (Fig. 2).

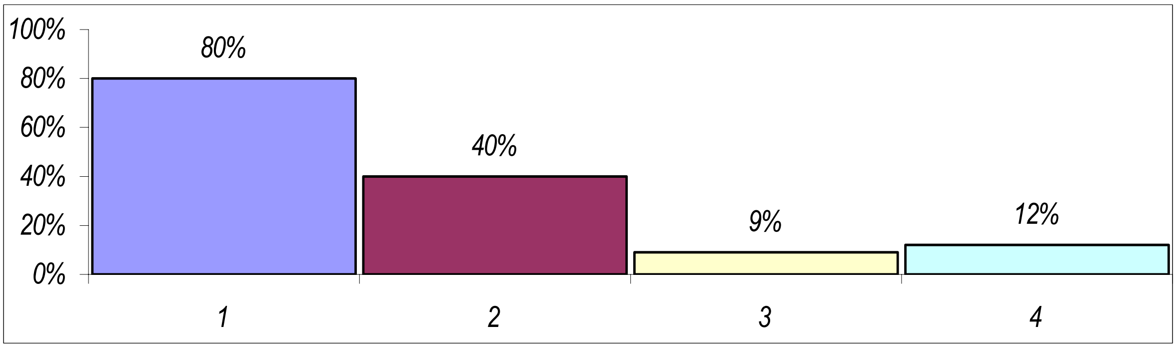


Fig. 2. Indexes of predicted processes for different users.

1 - Index of predicted processes for legal user on training set.
2 - Index of predicted processes for legal user on testing set.
3 - Index of predicted processes for illegal user #1.
4 - Index of predicted processes for illegal user #2.
Prediction errors for process for one session are shown in Fig. 3.

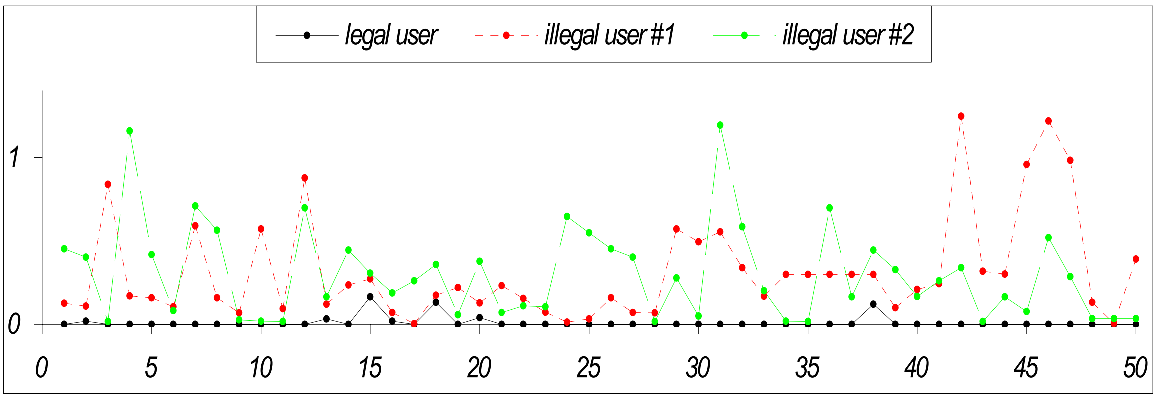


Fig. 3. Prediction errors for processes.

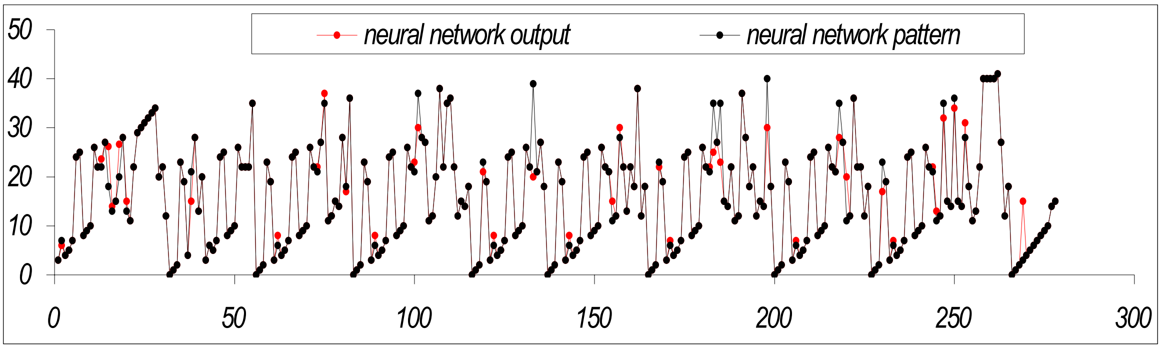


Fig. 3. Predicted process.

These results show the possibility of neural network to distinguish different users' behavior. Also other experiments were carried out in order to find optimal number of processes (premises) for correct prediction. Best results were achieved by predicting every 6-th command in sequence.

Conclusion

The above approach takes advantage of both intellectual methods of intrusion and anomaly detection and multi-agent architecture. The use of neural networks enables detection previously unknown attack types, while agent-based architecture provides features of intelligence and scalability as well as possibility to work in a heterogeneous environment. Currently, research of user behavior model demonstrates effectiveness of such approach.

Bibliography

1. V.Gorodetski, O.Karsaev, A.Khabalov, I.Kotenko, L.Popyack, V.Skormin. Agent-based model of Computer Network Security System: A Case Study. Proceedings of the International Workshop "Mathematical Methods, Models and Architectures for Computer Network Security". Lecture Notes in Computer Science, vol. 2052, Springer Verlag, 2001, pp.39-50.
2. James Cannady, James Mahaffey. The Application of Artificial Neural Networks to Misuse Detection: Initial Results.
3. A.M. Reznik, N.N. Kussul, A.M. Sokolov. Neural network identification of the behavior of the users of computer systems. Cybernetics and computational techniques, 1999, vol.123, pages 70-79.
4. A.M. Sokolov Computer System Intrusion Detection utilizing second-order Markov chain. Artificial Intelligence. Vol. 1, pp. 376-380. (in russian)
5. Jai Sundar Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, Eugene Spafford, Diego Zamboni. An Architecture for Intrusion Detection using Autonomous Agents http://citeseer.nj.nec.com/balasubramaniyan98architecture.html.

Author information

Natalia Kussul - PhD, Head of Space Information Systems Department, Space Research Institute NASU-NSAU; 40 Glushkov Ave, 03187 Kiev, Ukraine; e-mail: inform@space.is.kiev.ua, nkussul@dialektika.kiev.ua

Andrey Shelestov - PhD, Senior Scientist, Space Research Institute NASU-NSAU; 40 Glushkov Ave, 03187 Kiev, Ukraine; e-mail: inform@space.is.kiev.ua

Anton Sidorenko – Bachelor in Applied Mathematics; System Developer, Space Research Institute NASU-NSAU; 40 Glushkov Ave, 03187 Kiev, Ukraine; e-mail: inform@space.is.kiev.ua

Vladimir Pasechnik - Bachelor in Applied Mathematics; System Developer, Space Research Institute NASU-NSAU; 40 Glushkov Ave, 03187 Kiev, Ukraine; e-mail: inform@space.is.kiev.ua

Sergey Skakun - Bachelor in Applied Mathematics; System Developer, Space Research Institute NASU-NSAU; 40 Glushkov Ave, 03187 Kiev, Ukraine; e-mail: inform@space.is.kiev.ua

Natalia Levchenko - Bachelor in Applied Mathematics; System Developer, Space Research Institute NASU-NSAU; 40 Glushkov Ave, 03187 Kiev, Ukraine; e-mail: inform@space.is.kiev.ua

Yuri Veremeyenko - Bachelor in Applied Mathematics, Physics and Technology Institute, National Technical University of Ukraine "Kiev Polytechnic Institute"; 37 Peremogy Ave, 03057 Kiev, Ukraine; e-mail: yur@pth.ntu-kpi.kiev.ua

ROC CURVES WITHIN THE FRAMEWORK OF NEURAL NETWORK ASSEMBLY MEMORY MODEL: SOME ANALYTIC RESULTS

P.M. Gopych

***Abstract**: On the basis of convolutional (Hamming) version of recent Neural Network Assembly Memory Model (NNAMM) for intact two-layer autoassociative Hopfield network optimal receiver operating characteristics (ROCs) have been derived analytically. A method of taking into account explicitly a priori probabilities of alternative hypotheses on the structure of information initiating memory trace retrieval and modified ROCs (mROCs, a posteriori probabilities of correct recall vs. false alarm probability) are introduced. The comparison of empirical and calculated ROCs (or mROCs) demonstrates that they coincide quantitatively and in this way intensities of cues used in appropriate experiments may be estimated. It has been found that basic ROC properties which are one of experimental findings underpinning dual-process models of recognition memory can be explained within our one-factor NNAMM.*

***Keywords**: ROC, mROC, memory, neural network, cue index, recall, recognition, signal detection theory.*

1. Introduction

Receiver operating characteristics (ROCs or ROC curves) are widely used in classic signal detection theory to provide the performance of linear Fisher or Euclidian classifiers for different values of their thresholds; ROCs plot the probability of correct detection of a noisy signal as a function of the probability of its false detection or false alarm [1]. Usually, it is assumed that distributions of initial patterns (vectors), conditioned on the presence or absence of the sought-after signal (prior probabilities of the both hypotheses are chosen to be ½), are Gaussians with the same (or similar) variances and a specific distance between them. In neurosciences, ROCs are used, for example, in data analysis where in single or multiple neuronal spike trains the encoding and processing of sensory information are studied, e.g., [2]. Lately, a method for deriving ROCs by means of human memory testing has been developed but up to present there exists no computer memory model which was able to reproduce empirical ROCs neither qualitatively or quantitatively [3]. For this reason in the field of computer memory modeling understanding observed ROC curves is recognized as one of the most important unsolved problems [4].

In contrast to abstract computer models, neurobiology models directly address the problem of functional nature and neuroanatomical substrates of different kinds of memory. For example, now recognition memory is hotly debated within dual-process models (DPMs) which consider recognition as consisting of two components, recollection and familiarity, e.g., [3,5,6]. Recollection is thought of as an event where a person recalls both particular stimulus (a human face, for example) and episode where it was encountered earlier and familiarity represents the person's experience (or feeling) that particular stimulus was encountered before but without specific memory about where, when, or why it happened. It is claimed [3] that DPMs are supported by many results of cognitive, neuropsychological, and neuroimaging memory studies but in spite of long history of research even basic properties of DPMs are ambiguously defined and rather often even their basic terms are used by different authors in different ways [3]. Additionally, DPMs are not specified on computational level because most computer models consider recognition as one- not as two-factor process (although see [6]). On the other hand, none of computer models describes the whole body of recognition memory traits (in particular, ROCs) and, for this reason, their separate inferences which are not consistent with predictions of DPMs cannot be viewed as convincing arguments against them.

In present work analytical formulae for optimal ROC calculations are derived and, using convolutional (Hamming) version of Neural Network Assembly Memory Model (NNAMM) [7,8], we show that ROCs, as one of experimental findings underpinning DPMs, can be explained within our NNAMM without assuming that recognition memory is a dual process. A method of taking into account explicitly prior probabilities of alternative hypotheses on the structure of information initiating memory retrieval is proposed; on this basis modified ROC (mROC, unconditional probability of correct recall vs. false alarm probability) and overall probabilities of memory trace recall/recognition were introduced. It has been found that comparison of calculated and empirical ROCs (or mROCs) provides a method for extraction of those cues which were used actually in appropriate memory experiments.

2. Some NNAMM Backgrounds

According to NNAMM (see ref. 8 for details), components of initial ternary vectors take their values from the triple set –1,0,1 but most of these values are 0s (that is so called sparse coding). After data preprocessing, initial ternary vectors are transformed into binary feature vectors with components –1 or 1 (that is so called dense coding). In fact, feature vectors are quasibinary ones because their spinlike (–1,1) components cannot be shifted to other (0,1) binary representation and they could manifest (although do not manifest) their third, zero, components. Below only quasibinary vectors are considered but, for short, the preposition "quasi" will be omitted.

Neural network (NN) assembly memory is constructed from interconnected (associated) and equal in rights assembly memory units (AMUs) and the basic properties of assembly memory as a whole depend on the properties of its components, AMUs. AMU has original architecture and involves regular Hopfield two-layer autoassociative NN (that is the AMU's central element), *N*-channel time-gate, additional reference memory, and two nested feedback loops [8].

NN related to particular AMU is subserved by binary vectors mentioned. We refer to such an *N*-dimensional arbitrary vector as x. If x represents information stored or that should be stored in AMU then we term it x_0. We define random vector or binary noise x_r as x with components –1 or 1 randomly chosen with uniform probability, ½. Damaged reference vector, $x(d)$, is defined as x_0 with its damage degree d. The components, $x_i(d)$, of $x(d)$ are defined as

$$x_i(d) = \begin{cases} x_0^i, & if \quad u_i = 0, \\ x_r^i, & if \quad u_i = 1 \end{cases} \quad i = 1,..,N \tag{1}$$

where u_i are marks whose magnitudes 0 or 1 are chosen randomly with uniform probability and fixed d:

$$d = \sum u_i / N, \quad i = 1,..,N. \tag{2}$$

If the number of marks $u_i = 1$ is m then $d = m/N$; $0 \le d \le 1$; $x(0) = x_0$ and $x(1) = x_r$. Damage degree d is a fraction of noise in vector $x(d)$ while intensity of cue or cue index $q = 1 - d$ is a fraction of correct, undamaged information about x_0 in $x(d)$ [7,8]. The data coded in such a way naturally arise when to solve a very important problem of local feature discrimination across smooth background and additive noise, line or half-tone images are binarized using a convolutional NN recognition algorithm [9]. Expressions 1 and 2 define an original data coding procedure [7]. To design appropriate data decoding rules we explore two-layer auto-associative NN with N cells in its entrance (or exit) layer. Entrance and exit layer cells are connected by "all-to-all" rule, they are McCalloch-Pitts model neurons with rectangular response and triggering threshold θ.

Following ref. 10 for perfectly learned intact Hopfield NN, the elements w_{ij} of *synapse matrix* w are defined as

$$w_{ij} = \eta x_0^i x_0^j \tag{3}$$

where $i,j = 1,..,N$; $\eta > 0$ is a *learning parameter* (below $\eta = 1$); x_0^i, x_0^j are the components of reference vector x_0 (all w_{ij} may differ from each other in sign only). It is crucially important to stress that NN with synapse matrix w is learned to remember *only one* memory trace x_0 and we *deliberately* reject the available possibility of storing other traces in the same NN. Also we posit that an input vector x_{in} is decoded (recognized as reference vector x_0) successfully if learned NN transforms x_{in} into output vector $x_{out} = x_0$ [7,8,9].

The transformation algorithm is the following. For the *j*th neuron of the NN exit layer an *input signal* h_j is given by

$$h_j = \sum w_{ij} v_i + s_j \tag{4}$$

where v_i is an *output signal* of the *i*th neuron of the NN entrance layer; $s_j = 0$.

The signal v_j of the jth neuron of the NN exit layer (the jth component of x_{out}) is calculated according to the model neuron's rectangular response function (signum function or 1 bit quantifier) with triggering threshold θ as

$$v_j = \begin{cases} +1, & if \quad h_j > \theta \\ -1, & if \quad h_j \le \theta \end{cases} \qquad (5)$$

where for $h_j = \theta$ the value $v_j = -1$ was *arbitrary* assigned.

3. Convolutional and Hamming Versions of NNAMM

If $h_i = x_{in}^i$ then from Expression 5 follows that $v_i = x_{in}^i$. Of this fact and Equations 3 and 4 for the jth exit layer neuron we have: $h_j = \sum w_{ij} x_{in}^i = \eta x_0^j \sum x_0^i x_{in}^i = \eta x_0^j Q$ where $Q = \sum x_0^i x_{in}^i$ is a convolution of vectors x_0 and x_{in} ($-N \le Q \le N$). The substitution of $h_j = \eta x_0^j Q$ into Expression 5 gives that $x_{out} = x_0$ and vector x_{in} is successfully decoded (recognized as x_0) if $Q > \theta$ (if $\eta \ne 1$ then $Q > \theta/\eta$). Hence, NN algorithm given in Section 2 and the convolutional algorithm just now introduced are equivalent although in present form the latter is valid only for perfectly learned intact NNs (see details in ref. 8). Since for each x_{in} exists such a vector $x(d)$ that $x_{in} = x(d)$, inequality $Q > \theta$ can be written as a function of $d = m/N$ and as a result

$$Q(d) = \sum\nolimits_{i=1}^{N} x_0^i x_i(d) = \sum\nolimits_{i=1}^{N-m} (x_0^i)^2 + \sum\nolimits_{i=1}^{m} x_0^i x_r^i = N - m + (m - 2k) = N - 2k > \theta \qquad (6)$$

where the dimension of all vectors x, the number of noise components of $x(d)$, the number of corresponding bits of $x(d)$ and x_0 which always coincide, and the number of corresponding bits of particular x_r and x_0 which currently differ are N, m, $N - m$, and k, respectively; θ is threshold value of Q or model neuron's triggering threshold.

It is easy to obtain directly that $Q = N - 2D$ and $D = (N - Q)/2$ where D is a Hamming distance between x_0 and $x(d)$ (Hamming distance is a number of corresponding bits of x_0 and $x(d)$ which are different, $0 \le D \le N$). Since between D and Q there is an univocal correspondence, along with inequality $Q > \theta$ the inequality $D < (N - \theta)/2$ is also valid (cf. Inequality 6 where $k = D$). Moreover, $Q(d)$ can merely be interpreted as an expression for computation of Hamming distance D. That means that the above convolutional (Hamming) decoding algorithm or Hamming classifier directly discriminates the patterns $x_{in} = x(d)$ which are more close to x_0 than a given Hamming distance between them [8]. Hence, for data coding described in Section 2, NN, convolutional, and Hamming distance algorithms mentioned are equivalent. As Hamming classifier/recognition/decoding algorithm is the best (optimal) in the sense of statistical pattern recognition quality (that is no other algorithm cannot outperform it) [11], above NN and convolutional algorithms are optimal (the best) in that sense too.

4. Conditional Recall/Recognition Probabilities and ROCs

The basic idea of NNAMM is to build a NN memory model from simple objects defined within coding/decoding approach (optimal binary signal detection theory) introduced. For this purpose in Sections 2 and 3 it is simple enough instead of coding and decoding to speak about encoding and retrieval, respectively [8]. In this way NNAMM was formulated and fundamental recall/recognition properties of its assembly memory unit, containing corresponding Hopfield NN as its central element, were found optimally by multiple computations [7,8]. But convolutional (Hamming) version of NNAMM gives also a chance to obtain optimal (the best) formulae for this aim analytically.

Below we derive a formula for the probability $P(m,N,\theta)$ of correct recall/recognition of memory trace x_0 stored in perfectly learned intact NN with the model neurons' triggering threshold θ under condition that data patterns $x(d)$ initiating many-step memory trace retrieval [8] are actually x_0 with damage degree $d = m/N$ (earlier the same probability was calculated by multiple computations, examples for $\theta = 0$ see in ref. 7,8). Now we need to find the number $T(m,N,\theta)$ of vectors $x(d)$ for which Inequality 6 is valid and the total number of all possible different vectors $x(d)$. Since $x(d)$ contains m randomly combined noise components with randomly chosen magnitudes –1 or 1 (the probability of their choice is ½), the latter equals $2^m C^N_m$. To find $T(m,N,\theta)$ we use the fact that for each set of k, m, and N values the number of vectors $x(d)$ satisfying Inequality 6 is $C^N_m C^m_k$ where C^N_m is the number of ways arranging m noise components in N components of $x(d)$ and C^m_k is the same for k

components which have the sign opposite to the sign of corresponding components of x_0 in m noise components of $x(d)$. Consequently, $T(m,N,\theta) = C^N_m \sum C^m_k$ where the summation is made over $k = 0,1,..,kmax$ (k is Hamming distance between particular $x(d)$ and x_0). The probability $P(m,N,\theta)$ is computed by dividing $T(m,N,\theta)$ by $2^m C^N_m$, i.e.

$$P(m,N,\theta) = \sum_{k=0}^{k\max} C_k^m / 2^m \tag{7}$$

where if $kmax \le kmax_0$ then $kmax = m$ else $kmax = kmax_0$ and

$$k\max_0 = \begin{cases} (N-\theta-1)/2, & if \quad N \quad is \quad odd \\ (N-\theta)/2-1, & if \quad N \quad is \quad even \end{cases} \tag{8}$$

is defined by Inequality 6 and the signum function specified by Equation 5. Since $0 \le kmax \le m \le N$, if N is odd then $-(N+1) \le \theta \le N-1$ and if N is even then $-(N+2) \le \theta \le N-2$.
Let us consider two important special cases, $P(m,N,\theta) = 1$ and $m = N$, $\theta = 0$:

- Since $\sum C^{kmax}_k = 2^{kmax}$ ($k = 0,1,..,kmax$), from Equation 7 follows that for any N $P(m,N,\theta) = 1$ while $m \le kmax_0$.
- Since $\sum C^N_k = 2^N$ ($k = 0,1,..,N$), if N is odd then $P(N,N,0) = (2^N/2)/2^N = ½$ ($m = N$ and $\theta = 0$). Since $C^m_k = C^m_{m-k}$, if N is even then the sum $S = \sum C^N_k$ ($k = 0,1,..,N/2 - 1$) is defined by equation $2S + C^N_{N/2} = 2^N$ ($C^N_{N/2}$ is the number of events $Q = 0$). Thus, in this case $P(N,N,0) = ½ - \Delta P(N)$, $\Delta P(N) = C^N_{N/2}/2^{N+1} \sim 0.4/\sqrt{N}$ (here for large N Stirling's formula was used). The facts that $\Delta P(N) < 0$ and the minus sign was assigned to 1s in Expression 8 are caused by the choice of signum function form. If in Equation 5 for $h_j = \theta$ the value $v_j = +1$ is assigned then $\Delta P(N) > 0$ and in Expression 8 the plus sign before 1s should be chosen.

For odd and even N and for different choice of signum function, probabilities $P(m,N,\theta = 0)$ are shown in Figure 1 (as in ref. 7,8 to underline discrete character of NNAMM results, small values of N are taken, for example).

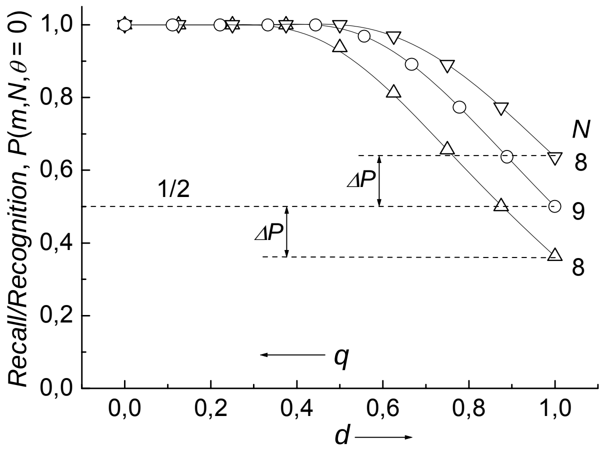


Figure 1. Conditional probability $P(m,N,\theta)$ of free recall ($d = 1$), cued recall ($0 < d < 1$), and recognition ($d = 0$) calculated according to Equations 7 and 8 for perfectly learned intact NNs with $\theta = 0$ and $N = 9$ (open circles) and $N = 8$ (triangles) vs. damage degree $d = m/N$ of memory trace x_0 or intensity of cue $q = 1 - m/N$. If $N = 9$ (i.e., N is odd) then free recall (false alarm) probability equals ½; if $N = 8$ (i.e., N is even) then free recall probability is $P(8,8,0) = ½ - \Delta P(8)$, $\Delta P(8) = C^8_4/2^9 = 70/512$. If $N = 9$ and $m \le 4$, if $N = 8$ and $m \le 3$ then $P(m,N,0) = 1$. If in Equation 5 for $h_j = \theta$ the value $v_j = 1$ is assigned then $P(8,8,0) = ½ + \Delta P(8)$ (upper curve).

Figure 2 shows two families of curves calculated according to Equations 7 and 8. In different form they represent the same probabilities $P(m,N,\theta)$ for perfectly learned intact NN memory unit with odd N and all possible values of $d = m/N$ and θ (if N is even then the curves can be splitted by the choice of signum function form).

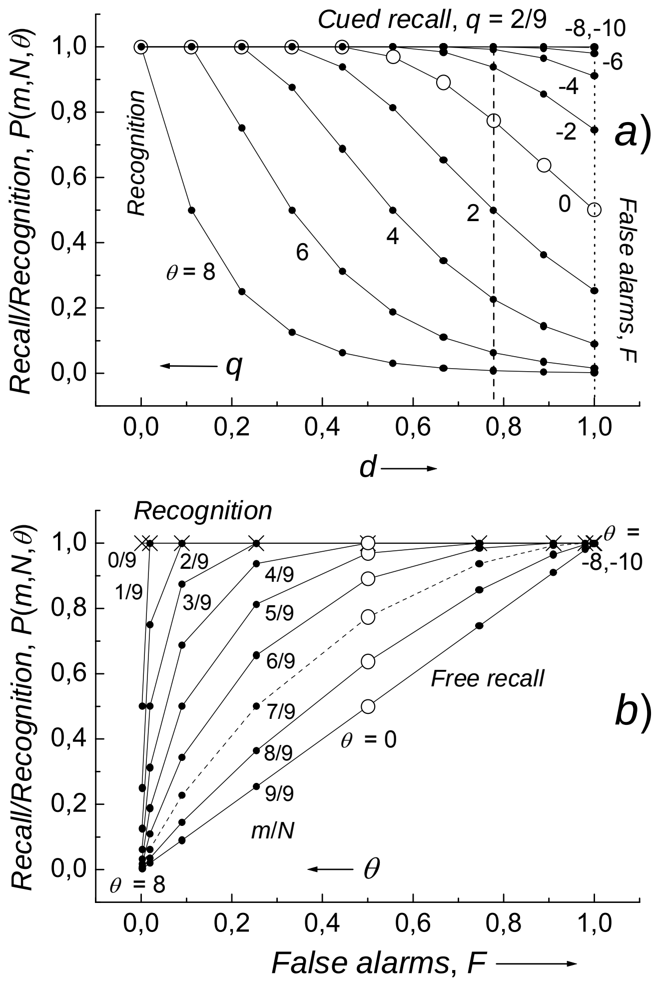


Figure 2. Data preparation for (**a**) ROC plot (**b**) and ROCs for the perfectly learned NN memory unit with $N = 9$.

a) Probabilities $P(m,N,\theta)$ vs. $d = m/N$, $q = 1 - m/N$, and θ. Open circles denote probabilities for $\theta = 0$, $P(m,9,0)$ (i.e., here and in all other Figures open-circle points are the same); dashed line connects cued recall probabilities for different θ and cue index $q = 2/9$, $P(7,9,\theta)$; free recall ($q = 0$) or false alarm probabilities, $F = P(9,9,\theta)$, are situated along the dotted line; for all curves their left-most points are the same, $P(0,9,\theta) = 1$ (that is recognition probability). The number of curves is $N + 1 = 10$. Since $0 < F \le 1$, the value $F = 0$ is impossible. For each θ right-most point of each curve represents appropriate value of false alarm F needed to plot ROCs.

b) Probabilities $P(m,N,\theta)$ vs. F, θ, and m/N. The values of F used for the ROC plot lie in the panel a) along the dotted line. For each ROC curve the value of m/N (q or d) is the same, the number of ROC points is $N + 1 = 10$. The more the value of cue, q, the more the curvature of respective ROC and the more the value of probability $P(m,9,\theta = 8)$, ROC's left-most point. Linear ROCs correspond to free recall ($q = 0$, $d = 1$) and recognition ($q = 1$, $d = 0$). Crosses denote recognition probabilities, $P(0,9,\theta)$.

5. Unconditional Recall/Recognition Probabilities, mROCs, and Overall Probabilities

In Section 4 conditional recall/recognition probabilities were discussed. But it is *a priori* unknown whether initial pattern $x(d)$ is a sample of pure noise (hypothesis H_0) or memory trace x_0 damaged by noise (hypothesis H_1). To obtain unconditional (*a posteriori*) probabilities of false and correct recall/recognition of the trace x_0 stored in NN memory unit, we used famous Bayes' formula and have as a result:

$$p_{FR}(m/N,F) = 1/(1+\kappa\frac{P(m/N,F)}{F}),\quad p_{CR}(m/N,F) = 1/(1+\kappa^{-1}\frac{F}{P(m/N,F)}) \qquad (9)$$

where $p_{FR}(m/N,F)$ and $p_{CR}(m/N,F)$ reflect unconditional false recall/recognition (FR) and correct recall/recognition (CR) probabilities; $p_{FR} + p_{CR} = 1$; $\kappa = P(H_1)/P(H_0)$; $P(H_0)$ and $P(H_1)$ are prior probabilities of hypotheses H_0 and H_1, respectively. Since $P(H_0)$ and $P(H_1)$ are usually unknown, in most cases $\kappa = 1$ is postulated. Here, we pay also attention to the fact of changing designations. As there is an univocal correspondence between F and θ (see Figure 2a) in Equation 9 and below, instead of θ, we write F; as all probabilities depend on m and N as on m/N (see Figures 1 and 2), we write these two parameters in the form of their ratio, m/N; thus, $P(m/N,F) = P(m,N,\theta)$.

Our data coding approach introduced in ref. 7 allows to find κ in explicit form directly. Indeed, by definition, a pattern $x(d)$ contains a fraction $d = m/N$ of noise components and a fraction $q = 1 - m/N$ of undamaged components of x_0 (see Section 2). Hence, d and q may be interpreted as the probabilities $P(H_0)$ and $P(H_1)$, respectively. That means that in Equation 9, within our NNAMM (or data coding/decoding) approach, κ is given by

$$\kappa = P(H_1)/P(H_0) = q/d = (N-m)/m. \qquad (10)$$

If $m = 0$ then, according to Equation 10, κ does not exist and in this special case we posit that $p_{FR} = 0$ (at the same time $p_{CR} = 1$); if $m = N$ then κ^{-1} does not exist and in this special case we posit that $p_{CR} = 0$ (at the same time $p_{FR} = 1$). Both propositions are in full concordance with the fact that the former is the case of undamaged memory trace x_0 and the latter is a case of pure noise. Taking into account that $0 < F \le P(m/N,F) \le 1$ (see Figure 2), that Equation 10 and propositions $p_{FR} = 1$ (if $m = 0$), $p_{CR} = 0$ (if $N = m$) are valid, we have: $0 \le p_{FR} \le 1$, $0 \le p_{CR} \le 1$ (instead of $0 < p_{FR} \le ½$, $½ \le p_{CR} < 1$ if it is as usual supposed that $\kappa = 1$ and $0 < F \le P(m/N,F) < 1$).

Equations 9,10 provide unconditional probability $p_{CR}(m/N,F)$ as a function of false alarm F and for this reason for the fixed m/N we refer to particular $p_{CR}(m/N,F)$ as modified ROC or mROC. Figure 3 illustrates this claim.

Let us define

$$P_{FR}(m/N) = \sum p_{FR}(m/N,F)/(N+1), \quad P_{CR}(m/N) = \sum p_{CR}(m/N,F)/(N+1) \qquad (11)$$

where $P_{FR}(m/N)$ and $P_{CR}(m/N)$ provide overall, not depending on F, unconditional FR and CR probabilities of recall/recognition of the memory trace x_0 stored in perfectly learned NN; summations are made over all $0 < F \le 1$; $p_{FR}(m/N,F)$, $p_{CR}(m/N,F)$ are calculated according to Equations 9,10; as it was expected, $P_{FR} + P_{CR} = 1$.

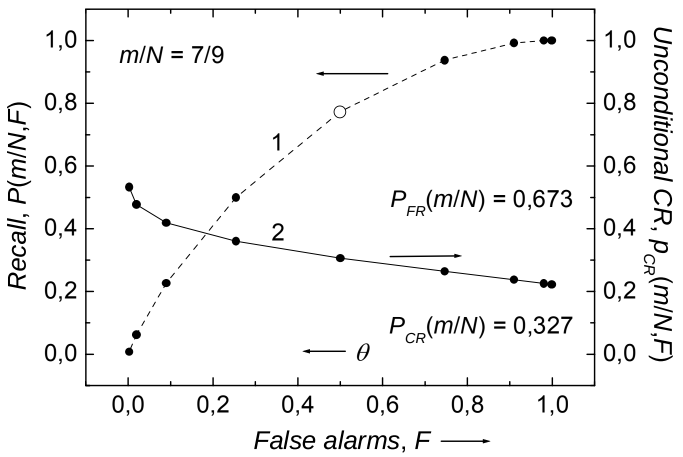


Figure 3. ROC curve (curve 1, left-hand scale) and mROC curve (curve 2, right-hand scale) for $d = m/N = 7/9$, $q = 2/9$. ROCs along the dashed line are as in Figure 2. The mROC curve is a plot of unconditional correct recall probability $p_{CR}(m/N,F)$ vs. false alarm F; mROC points according to Equations 9,10 were calculated; the special case $p_{CR}(m/N,F) = 1$ is not shown and not considered. Average values of $p_{CR}(m/N,F)$ and $p_{FR}(m/N,F)$ reflect overall probabilities $P_{FR}(m/N)$ and $P_{CR}(m/N)$, respectively, they were estimated by Equations 11.

6. Comparison with Experiments

In Figure 4 NNAMM numerical predictions (calculated ROC curves) are compared with ROCs observed in item recognition or similar tests. In different panels typical examples of empirical many-point and two-point ROCs are examined, estimated empirical data were taken from ref. 3. As one can see, even for illustrative model example $N = 9$ where only cue index q was as a fit parameter (the change of N does not change the form of ROCs), a good quantitative agreement between theory and experiment is achieved. Thus, the comparison of empirical and model ROCs may be viewed as a method for estimation of the value of the intensity of cue available in the process of the recall or recognition for specific memory system under specific conditions of specific experiment.

As Figure 4 demonstrates, there is no problem of reproducing available empirical ROCs within NNAMM both qualitatively and quantitatively and comparison of calculated and empirical ROCs may be successfully used for the value of intensity of cue, q, estimation. Since for empirical many-point-confidence-scale ROCs the value of cue changes along the curves (see Figure 4a), both the model's predictions and the details of experimental protocols demand scrutiny.

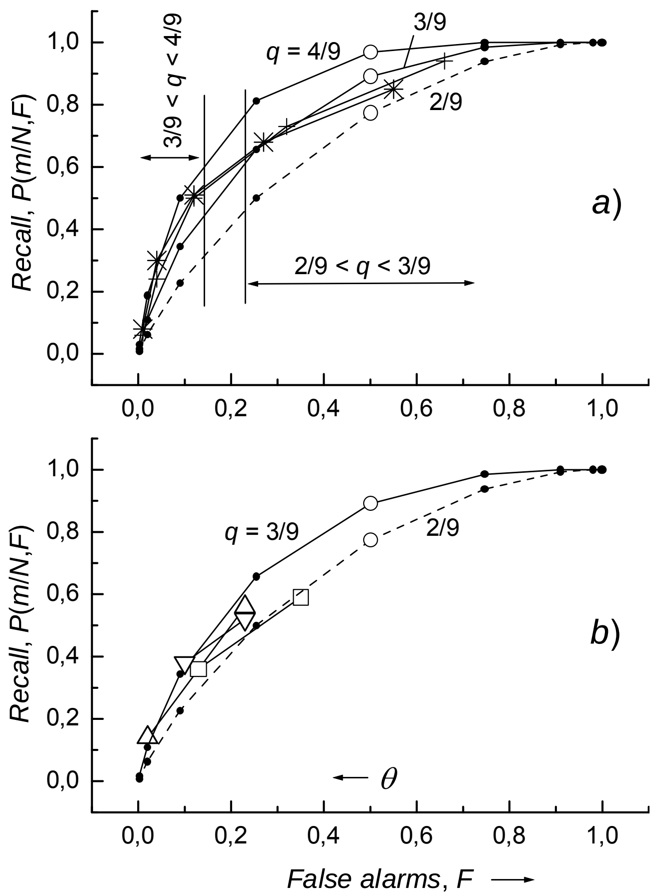


Figure 4. Theoretical (N = 9) and empirical ROCs. For each calculated ROC respective value of q is shown. Here and in Figures 2 and 3 the dashed-line curve is the same. **a)** Comparison of theoretical and empirical ROCs derived using 5-point-confidence-scale experiments. Original results are from ref. 12 and 13, the first 3 and the last 2 points of empirical ROCs are consistent with the assumption that $3/9 < q < 4/9$ and $2/9 < q < 3/9$, respectively. **b)** The same for empirical ROCs derived using 2-point-confidence-scale experiments. Original results are from ref. 14 and 15, they are consistent with $2/9 < q < 3/9$.

In many experiments (e.g., associative recognition tests, remember/know or process-dissociation procedures) subjects are required to recall both an item itself and other information related to it [3]. That means that in such experiments those memory events could be selected and investigated where subjects are able a target item to retrieve and "to assess" its *a posteriori* probability taking into account *a priori* probabilities of prior hypotheses on the structure of information initiating retrieval (i.e., taking into account $P(H_1)$, the probability of the fact that vector $x(d)$ reflects damaged target item, and $P(H_0)$, the probability of the fact that $x(d)$ is a lure item). Hence, empirical results obtained using such an experimental paradigm could provide unconditional (*a posteriori*) recall/recognition probabilities $p_{CR}(m/N,F)$ introduced in Section 5. This assumption is examined in Figure 5.

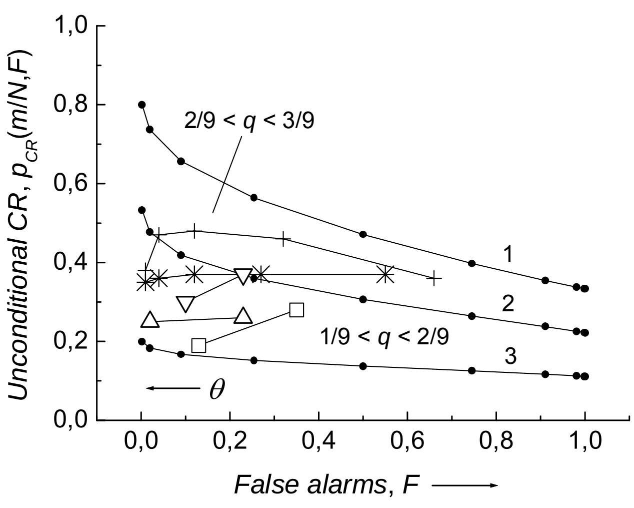


Figure 5. Theoretical (N = 9) and empirical mROCs. Curves 1, 2, and 3 reflect $p_{CR}(m/N,F)$ calculated according to Equations 9,10 with cue indices $q = 1 - m/N$ = 3/9, 2/9, and 1/9, respectively. Empirical mROC curve in the same signs as ROC curve in Figure 4 was taken from the same reference [3,12-15].

As Figure 5 demonstrates, theoretical mROCs provide good quantitative description of observed mROCs [3,12-15] and their comparison may also be viewed as a method for estimation of specific values of the intensity of cue for specific memory experiments. For example, empirical 2-point mROCs [14,15] and 5-point mROCs [12,13] are consistent with the assumption that $1/9 < q < 2/9$ and $1/9 < q < 2/9$ to $2/9 < q < 3/9$, respectively (in the latter case q changes along the curves).

Comparison between the values of q estimated using ROCs and mROCs shows that they are similar but not always coincide. Indeed, an analysis of ROCs and mROCs observed in experiments [14,15] gives inconsistent results (2/9 < q < 3/9 and 1/9 < q < 2/9, respectively) while the analysis of experiments [12,13] gives consistent results if only 2 right-most ROC and mROC points are considered (2/9 < q < 3/9) and inconsistent results if some left-most ROC and mROC points are taken into account (3/9 < q < 4/9 and 1/9 < q < 2/9, respectively). To explain these features, additional analysis of the model's predictions and experimental details is needed.

7. Discussion

The properties of empirical ROC curves have been used as one of four basic arguments in favour of DPMs of recognition memory. For example, as Figures 4 and 5 demonstrate empirical ROCs derived in item and associative recognition tests are essentially different [3]. ROCs related to item recognition tests are curvilinear with changing shape across measurement conditions; they can be approximated by a two-factor formula related to traditional signal detection theory and containing recollection and familiarity as stochastically independent fit parameters. For this reason, it is claimed that "at least two separate memory components are needed to account for recognition performance" [3, p.442]. This idea was realized as a two-factor parameterization of empirical ROCs: $P_i = R + (1 - R)\Phi(d'/2 - c_i) + F_i - \Phi(d'/2 - c_i)$ where P_i, F_i, R, d', c_i, and Φ reflect correct recall probability (a counterpart to probability $P(m/N,F)$ defined by Equation 7), false alarm, recollection, familiarity, response criterion, and item distribution (Gaussian), respectively. The fitting of this equation to observed ROC curves provides estimations of recollection (R) and familiarity (d'). Since ROCs observed in item recognition tests (Figure 4) are well fitted by this formula and ROCs observed in associative recognition tests (Figure 5) are not, it is suggested that the former can be described by a signal detection theory while the latter can not [3].

Our NNAMM is based on our optimal binary signal detection theory (Sections 2-5, ref. 7-9] and for intact perfectly learned memory unit it is actually a one-factor computer model; this factor (intensity of cue or cue index, q) is the amount of undamaged information about the memory trace x_0 containing in vectors $x(d)$ which initiate many-step memory retrieval [8]. It is essential that such an one-factor approach on a common ground successfully describes different types of memory including free recall (q = 0), cued recall (0 < q < 1) , and recognition (q = 1) [7,8] and for this reason there is no need to introduce any new type of memory, like recollection or familiarity, for example ("recollection" and "familiarity" of DPMs are loosely equivalent to recall and recognition of NNAMM, respectively). By definition, all acts of the particular item's recall and recognition are different in time processes and, consequently, they are stochastically independent and do not run in parallel. According to NNAMM, recognition ("familiarity" of DPMs) is an one-step process of testing selected assembly memory unit (AMU) *without* using the cues stored in other related AMUs [8]. In general, such a process may correspond to an item recognition test of so called *semantic* memory. Recall ("recollection" of DPMs) is a many-step process of testing selected AMU *with* using the cues stored in other (one or more) AMUs [8]. In general, such a process may correspond to an associative recognition test of so called *episodic* memory (for relations between semantic and episodic memories see ref. 16, for example). As one can see from Section 6, ROCs observed in item recognition tests and mROCs observed in associative recognition tests are successfully described within our NNAMM based on our optimal binary signal detection theory.

Since all basic properties of empirical ROCs (and mROCs) have been qualitatively and even quantitatively reproduced within one-factor NNAMM, ROCs might be excluded from the list of findings underpinning DPMs of recognition memory. On the ground of our previous [7,8,17,18] and present results it is natural to anticipate other items of this list (different speeds of response for recollection and familiarity, their different electrophysiological correlates, and different extents of their disruption by certain brain injuries [3]) are also consistent with NNAMM.

8. Conclusion

For the first time a method for theoretical description of empirical ROC curves has been proposed within a computer memory model. For this purpose a convolutional (Hamming) version of our NNAMM based on our optimal binary signal detection theory was used. Analytical formulae for optimal (the best) calculation of conditional and unconditional probabilities of false/correct recall/recognition of memory trace stored in intact perfectly learned NN memory unit have been found. In particular, a method of taking into account explicitly *a priori* probabilities of alternative hypotheses on the structure of information containing in vectors $x(d)$ and initiating memory retrieval and a method for estimation of overall recall/recognition probabilities are proposed. Using the derived optimal analytical formulae, empirical ROCs obtained in item recognition tests and empirical mROCs obtained in associative recognition tests were described and the values of intensity of cue, *q*, for some specific experiments were quantitatively estimated; thus, the comparison of theoretical and empirical ROCs is a method proposed here to estimate cue indices for specific experiments. It has been shown that ROCs might be excluded from the list of empirical findings underpinning popular DPMs of recognition memory.

I am grateful to HINARI (Health Internetwork Access to Research Initiative) for free on-line access to recent full-text journals, participants of the KDS-2003 Conference, Varna, Bulgaria, June 16-26, 2003 for helpful discussion, and my family and my friends for their help and support.

Bibliography

[1] D.Green & J.Swets. Signal Detection Theory and Psychophysics. New York, Wiley, 1966.

[2] W. Metzner, C. Koch, R.Wessel, & F.Gabbiani. Feature Extraction by Burst-Like Spike Pattern in Multiple Sensory Maps. Journal of Neuroscience, 1998, **18**(6), 2283-2300.

[3] A.P.Yonelinas. The Nature of Recollection and Familiarity: A Review of 30 Years of Research. Journal of Memory and Language, 2002, **46**, 441-517.

[4] B.B.Murdock. Classical Learning Theory and Neural Networks. The Handbook of Brain Theory and Neural Networks. Cambridge, MIT Press, 1995, 189-192.

[5] J.R.Manns, R.O.Hopkins, J.M.Reed, & E.G.Kitchener, L.R.Squire. Recognition Memory and the Human Hippocampus. Neuron, 2003, **37**(1), 171-180.

[6] K.A.Norman & R.C.O'Reilly. Modeling Hippocampal and Neocortical Contributions to Recognition Memory: A Complementary Learning Systems Approach. Psychological Review, 2003, **110**(4).

[7] P.M.Gopych. Determination of Memory Performance. JINR Rapid Communications, 1999, **4**[96]-99, 61-68 (in Russian).

[8] P.M.Gopych. A Neural Network Assembly Memory Model with Maximum-Likelihood Recall and Recognition Properties. Vth International Congress on Mathematical Modeling, Dubna, Russia, September 30 – October 6, 2002, Book of Abstracts, vol.2, p.91. See also http://arXiv.org/abs/cs.AI/0303017.

[9] P.M.Gopych. Identification of Peaks in Line Spectra Using the Algorithm Imitating the Neural Network Operation. Instruments & Experimental Techniques, 1998, **41**(3), 341-346.

[10] J.J.Hopfield & D.W.Tank. Computing with Neural Circuits: a Model. Science, 1986, **233**, 625-633.

[11] R.Hamming. Coding and Information Theory. Englewood Cliffs, NJ, Prentice Hall, 1986.

[12] A.P.Yonelinas. Receiver-Operating Characteristics in Recognition Memory: Evidence for a Dual-Process Model. Journal of Experimental Psychology: Learning, Memory, and Cognition, 1994, **20**(6), 1341-1354.

[13] A.P.Yonelinas. Consciousness, Control, and Confidence: The 3 Cs of Recognition Memory. Journal of Experimental Psychology: General, 2001, **130**(3), 361-379.

[14] F. Strack & J.Foerster. Reporting Recollective Experiences: Direct Access to Memory Systems? Psychological Science, 1995, **6**(6), 352-358.

[15] E.Hirshman & A.Henzler. The Role of Decision Processes in Conscious Recollection. Psychological Science, 1998, **9**(1), 61-65.

[16] E.Tulving, H.J.Markovitsch. Episodic and Declarative Memory: Role of the Hyppocampus. Hyppocampus, 1998, **8**, 198-204.

[17] P.M.Gopych. Three-Stage Quantitative Neural Network Model of the Tip-Of-the-Tongue Phenomenon. Proceedings of the IXth Internetional Conference KDS-2001, St-Petersburg, Russia, June 19-22, 2001, p.158-165 (in Russian). See also http://arXiv.org/abs/cs.CL/0103002, http://arXiv.org/abs/cs.CL/0107012, http://arXiv.org/abs/cs.CL/0204008.

[18] P.M.Gopych. Computer Modeling of Feelings and Emotions: A Quantitative Neural Network Model of the Feeling-Of-Knowing. Kharkiv University Bulletin, Series Psychology, 2002, no.550(1), 54-58 (in Russian). See also http://arXiv.org/abs/cs.AI/0206008.

Author information

Petro Mykhaylovych Gopych – V.N.Karazin Kharkiv National University; Svoboda Sq., 4, Kharkiv, 61077, Ukraine; e-mail: pmg@kharkov.com.

PARONYMS FOR ACCELERATED CORRECTION OF SEMANTIC ERRORS *

I. A. Bolshakov, A. Gelbukh

Abstract: *The errors usually made by authors during text preparation are classified. The notion of semantic errors is elaborated, and malapropisms are pointed among them as "similar" to the intended word but essentially distorting the meaning of the text. For whatever method of malapropism correction, we propose to beforehand compile dictionaries of paronyms, i.e. of words similar to each other in letters, sounds or morphs. The proposed classification of errors and paronyms is illustrated by English and Russian examples being valid for many languages. Specific dictionaries of literal and morphemic paronyms are compiled for Russian. It is shown that literal paronyms drastically cut down (up to 360 times) the search of correction candidates, while morphemic paronyms permit to correct errors not studied so far and characteristic for foreigners.*

Keywords: *error correction, correction candidates, semantic errors, malapropisms, paronyms, literal paronyms, morphemic paronyms, paronymy dictionaries.*

Introduction

Various errors made by authors in theirs natural language texts can be categorized as follows:

- Orthographic errors transform a correct word into a senseless letter string, e.g., *interesting vook* (instead of *book*);
- Syntactic errors transform one real word to another, thus violating syntactic correctness of the texts concerning agreement of adjectives with their ruling nouns in gender and/or number in Slavic or Romance languages, e.g. Rus. *маленький мальчики* 'little$_{SG}$ boys$_{PL}$' instead of *маленькие*$_{PL}$; grammatical cases of the valence dependent noun in Slavic languages (Rus. *довольный правительству*$_{DAT}$ lit. 'content to the government' instead of *правительством*$_{INS}$), personal verb forms (*he go* for *goes*) (SG, PL are singular and plural; DAT, INS are dative and instrumental case), etc.
- Semantic errors leave the text orthographically and syntactically faultless, but make it senseless or absurd (*inculpation period* for *incubation period, massy migration* for *massive migration,* etc.).

All modern text editors have tools for error detection. Purely orthographic errors are detected always, and lists of potential correction candidates are given out similar to the suspicious string in letters and/or sounds. Grammatical errors are not always detectable because of deficiencies of modern syntactic analyzers, and variants of syntax corrections are rare so far. Semantic errors are not detected at all.

Meanwhile, methods are already proposed of how to correct one type of semantic errors. For this type, one real word is replaces by another "similar" to the intended one in literal or sound content. If such errors violate semantic correctness of texts, they are referred to as malapropisms.

In [Hirst & St-Onge, 1998; Hirst & Budanitsky, 1998] detection and correction of malapropisms use paradigmatic semantic links between words occurring in adjacent paragraphs and sentences. These are links between direct word repetitions, a word and its hyperonym (*appliance* Vs. *vacuum cleaner*), a part and the whole (*steering wheel* Vs. *car*), etc. For several languages, the links are recorded in thesauri, among which EuroWordNet is well known [Vossen, 2000]. For example, the replacement of *wheel* by *weal* semantically isolates *weal* from words *car*, *brakes* or *gas* within a text.

In [Bolshakov, 2002] malapropism processing uses syntagmatic links between words in a sentence. Malapropisms destroy stable syntactically linked and semantically admissible combinations of content words (=collocations). E.g., *massive migration* is collocation whereas the syntactically correct *massy migration* is not; cf. [Bolshakov & Gelbukh, 2001, 2002]. Thus malapropisms make some content word(s) in a sentence semantically isolated concerning collocations.

For any method of malapropism detection, a generator of correction candidates is necessary. They should be somehow "similar" to the intended words. Such generation is analogous to candidate search for orthographical errors but it differs in the rational search strategy.

* Work done under partial support of Mexican Government (CONACyT, SNI), IPN (CGPI, COFAA) and Korean Government (KIPA Professorship for Visiting Faculty Positions). The second author is currently on Sabbatical leave at Chung-Ang University.

Indeed, word forms of natural language are rare interspersions in the space of literal strings. For approximate evaluation of this rarefaction, let us take into account that in such highly inflectional language as Russian there exist ca 1.2 million of different word forms, whereas in low inflectional English, say, four times less. To calculate the number of all possible strings above a given alphabet of *A* letters, suppose their length equal to the mean length *L* of real words in a corresponding dictionary. Then the total string number equals A^L, i.e. $32^9 \approx 3.5*10^{13}$ in Russian and $26^8 \approx 2.1*10^{11}$ in English. This means that in Russian a real word form contrasts 29 millions of senseless strings, while in English, contrasts 700,000. The change of the mean length of word form in a dictionary to the mean textual value decreases this contrast, still leaving it striking.

If word forms as letter strings were absolutely stochastic in structure, the probability to meet two forms at a short distance would be inconsiderable. In fact, words are built of few thousands of radixes and even fewer prefix and suffix morphs (they are few hundreds in the whole functional morphemarium). Some semantic and morphonological restrictions are imposed on the sets of radixes, prefixes, and suffixes, since not all combinations are reasonable and not all reasonable ones are pronounceable.

Just this circumstance facilitates the candidate search for replacement of one real word by another. Whereas for an orthographical error a wrong string can be arbitrary and the task to gather beforehand, for each string, literally close real words seems impractical, the environments of a real word form, as our evaluations show, contains only few other real words. Hence, the close words can be gathered for each word that has them. Being put in a special dictionary, they could be used for malapropism correction, to cut down the search of candidates. Indeed, for correction of one-letter error in a string with the length *L*, it is necessary $A(2L+1)+L–1$ tries, that for a word of nine letters equals 616. For two-letter errors already ca. 360,000 tries are necessary. In the same time, forehand gathered one-letter-apart candidates are numbered few units, for two-letter-apart ones, numbered few tens. For words that are not in the dictionary of substitutes, the candidate search is unneeded, and this also cuts the search.

This work has the objective to classify semantic errors in some detail and to propose for malapropism correction dictionaries of paronyms, i.e. of words similar to each other in some specific sense. Paronyms can be introduced of the following intersecting types:

- ***Literal paronyms*** [Гусев & Саломатина, 2000, 2001] differ in few letters, so they are within easy distance in the space of letter strings, e.g., Rus. *ожижать* ‘to liquidize’ Vs. *ожидать* ‘to wait,’ *рок* ‘doom/rock’ Vs. *срок* ‘period.’ They are intended for correcting errors characteristic for careless and/or poorly literate persons.
- ***Sound paronyms*** differ in few sounds, so they are within easy distance in the space of phonological records of speech, e.g., Rus. *проектировать* ‘to design’ Vs. *проецировать* ‘to project’). They are indispensable for poorly literate persons.
- ***Morphemic paronyms***, known in Russian lexicography as paronyms proper [Бельчиков & Панюшева, 1994], have the same radix, pertain to the same part of speech (POS), and differ only in prefixes and/or suffixes. E.g., *sens-ible* Vs. *sens-itive* differ in one suffix; *re-volu-tion Vs. in-volu-tion,* in one prefix; *sens-ation-al* Vs. *sens-itive,* in two suffixes. Such paronyms can be close in the space of strings of morphemic symbols. They are important for poorly educated native speakers and for foreigners.

This work reports on compiling Russian dictionaries of one-letter and morphemic paronyms. The dictionaries’ fragments and general statistical parameters are given. Literal paronyms cut the search trials by approximately 360 times, while morphemic paronyms permits to quickly detect the errors not yet discussed anywhere but really occurring in texts and speech.

Sources of semantic errors and their effect

Let us classify semantic errors against their sources, giving minimal contexts.

1. Random error directly giving a real word. This could occur by the following reasons:
   - A writing slip immediately gives another real word, e.g. Rus. *испытательный рок* ‘trial rock’ instead of *срок* ‘period.’
   - A slip gives senseless string that is falsely “corrected” based on a spellchecker menu, since the author took incorrect candidate among those proposed by text editor. If we enter Rus. *испытательный* ***мрок***(?), the menu of spellchecker will contain for the highlighted string the items *мирок, мрак, прок, рок, срок*, *урок,* along with some non-nouns, and the careless author can select a wrong item.
   - A correct but very rare word is entered, for which the spellchecker contains one or more alternatives. For example, in the sentence *Ethology of these animals is not studied* spellchecker will propose to

replace *ethology* by the more known *etiology* or *ethnology,* and the author can hastily accept such corrections.

- An entered rare word is automatically corrected by a special utility of automatic correction embedded in the text editor. In this case the user transfers a power to the software to make some amendments without any consultations.

2. Ignorance or imprecise knowledge of the intended word, so that instead of it a different word is entered similar to the intended one in sound, e.g. *scientific hypotenuse* instead of *hypothesis*.
3. Imprecise knowledge of meaning for words with the same radix (which really can have the same semantic components), e.g. *sensual news* instead of *sensational news.*
4. Wrong facts or incorrect logic of reasoning transferred in the text. This rarely implies an error in one word, and if so, the resulting word frequently differs from the correct one: *His mother died in **infancy*** (for *youth*?); *Hendel was **half*** (for *partially*?) *German, **half** Italian, and **half** English*. Every human (but not a computer) knows that if a female died in infancy she had no children; that no dividable entity can have three halves, etc.

Hereafter, we deal with the errors of the types 1 to 3. In contrast to errors of the type 4, they violate purely linguistic knowledge on how to commonly use words within the same text. The textual word proved to be:

- similar to the intended one in letters, sounds or morphs,
- preserving syntactic correctness of the utterance, and
- essentially deforming its meaning.

Just such errors are called malapropisms [Encyclopædia, 1998]. Linguistic knowledge is violated by them in the aspects of :

- Syntagmatic semantic links in the texts. The resulted word combinations are not collocations but are syntactically correct. The examples were given above (except of p. 4). More examples are: *polling **company*** (for *campaign*); ***hysterical*** (for *historical*) *center;* ***dielectric*** (for *dialectic*) *materialism*; *travel about the **word*** (for *world*); *equal **excess*** (for *access*) *to school.*
- Paradigmatic semantic links in the texts. Here is an example fit for a single sentence: *Total **garniture*** (for *furniture*) *was ruined*: *tables, chairs, armchairs*. Tables, chairs, and armchairs really are related to furniture (not of garniture!), and this is also linguistic knowledge: interrelation of parts and the whole. However, furniture never form collocations with tables, chairs, and armchairs.

The task of candidate search is the same for both type of violation of linguistic knowledge.

Literal paronyms

One literal string of the length *L* can be formed from any other with the series of editing operations [Kashyap & Oomen, 1981; Mays *et al.*, 1992; Wagner & Fisher, 1974]. Let us take strings under an alphabet of *A* letters. Elementary editing operations are: replacement of a letter with any other letter in any place within source string [giving $(A–1)L$ options]; omission of a letter [*L* options]; insertion of a letter [$A(L+1)$ options]; permutation of two adjacent letters [$L–1$ options].

The string obtained with any of $A(2L+1)+L–1$ operations mentioned, is at the distance 1 from the source string, i.e. on the sphere of radius 1 in the string space. Making another elementary step off, we form a string on the sphere with radius 2—with regard to the source one, etc. Points obtained with minimum *R* steps are on *R*-sphere, points of *r*-spheres with $r < R$ and the source point are not here. Among previous examples,

- *word* Vs. *world, ethology* Vs. *etiology, ethology* Vs. *ethnology* are at the distance 1,
- *hysterical* Vs. *historical, dielectric* Vs. *dialectic, excess* Vs. *access, garniture* Vs. *furniture* are at the distance 2,
- *company* Vs. *campaign*, *massy* Vs. *massive, sensible* Vs. *sensitive, hypotenuse* Vs. *hypothesis* are at the distance 3 or more.

Though the mean distance between word forms is large in any language, they proved to be disposed in clusters. Firstly, such clusters contain elements of morphological paradigms of various lexemes, word forms within them being usually distanced 0 to 3 from each other. Just such a cluster is lexeme, and one of the composing forms is its dictionary name. Secondly, paradigms of various lexemes with similar morphs can be close to each other, sometimes even with intersection.

For our purposes, the paradigm pairs with the same number of elements and correlative elements at the same distance are of interest. E.g., all four elements of paradigms of Eng. verbs *bake* and *cake* differ in the first letter only. Let us call such paradigms parallel. If the distance equals 1, let us call them close parallel.

Thus, any element $\lambda(\chi)$ of the paradigm of λ (χ is a set of intra-lexeme coordinates, i.e. morphological characteristics selecting a specific word form) can be obtained from the correlated element of the parallel paradigm by use of the same editing operator $R_i()$, where *i* is cardinal number of the operator in an effective enumeration of such operators. Then the relation between dictionary names (they correspond to $\chi = \chi_0$) and specific word forms of parallel lexemes can be represented by the proportion

$$\lambda(\chi) : \lambda(\chi_0) = R_i(\lambda(\chi)) : R_i(\lambda(\chi_0)). \quad (1)$$

The formula (1) means that, for any suspicious form $\lambda(\chi)$ in text, it is necessary to find its dictionary form $\lambda(\chi_0)$, and, if a close parallel $R_i(\lambda(\chi_0))$ for it exists, $R_i(\lambda(\chi))$ should be tried as a correction candidate. For such try, the syntactic correctness pertains as a rule, and some try can correct the error.

The parallelism permits to unite sets of word forms, storing in the dictionary only one their representative, i.e. dictionary name of lexeme. However, strictly parallel paradigms are not so frequent in highly inflectional languages. More usually the parallelism between subparadigms can be found. As such subparadigms, it is reasonable to take grammemes corresponding to fixed combinations of characteristics χ.

For example, noun lexemes of European languages have grammemes of singular and plural. They play the same role in a sentence but differ in the sets of collocations they can be in. The division by grammatical number permits to describe easier Slavic declension and well serves for our purposes. E.g., the subparadigms of singular for Russian *метель* 'blizzard' and *мотель* 'motel' are not parallel, whereas they do—in plural.

Russian verbs have grammemes of personal forms (we join the infinitive to them), of active and passive participles in all grammatical cases, and of gerund. These grammemes differ in their role in a sentence, so that their separate use keeps syntactic correctness of text after the substitution. It is also reasonable to divide each Slavic verb grammeme to its perfect and imperfect aspects, morphonologically rather different.

Each grammeme has its own dictionary name, e.g., a participle is represented by the singular form of nominative case. For the dictionary names and specific forms, the formula (1) pertains. Note that it is not obligatory to require strict parallelism within whole grammemes. E.g., formula (1) applied to Rus. *метры* 'meters' и *меры* 'measures' fails in genitive case. However such failed tries are not too burdensome.

The idea to divide morpho-paradigms into grammemes is not taken at random. The CrossLexica system elaborated by authors [Bolshakov & Gelbukh, 2001] operates just with grammemes, and paronyms dictionaries under questions are oriented primarily to systems of this kind.

Let us call ***literal paronyms*** any two grammemes that:

- are of the same part of speech;
- concern to the same grammeme type, e.g., both are participles;
- have (close) parallel forms; and, only for nouns,
- have the same gender in singular or are both plural.

With such definition, we have searched close parallel literal paronyms among rather frequent content Russian words. The pairs with at least one member being functional word (pronoun, preposition, conjunctions, etc.) were omitted. A large preliminary version of dictionary was compiled first, and then a special utility proofreads this version for repetitions, omission of inverted pairs, larger distances, wrong orders, etc. Note that in [Гусев & Саломатина, 2000, 2001] the same task has been performed for lexeme names, thus giving less information (see above).

In the current version, there are more than 6,000 paronym groups each having a item-head grammeme to be replaced and the rest grammemes as substitute candidates. The mean number of candidates stably equals 2.25, while the mean name length is 6.75.

Functional words, the shortest in any language, were excluded. Nevertheless, the mean word length in our dictionary proved to be two letters shorter than the mean dictionary length value. So grammemes in our dictionary are seemingly the shortest among the content words, and probably the most frequent among them.

Below, we give a fragment of our dictionary. Note that homonyms like *болеть*1 'to be ill' Vs. *болеть*2 'to ache' or *белки*1 'squirrels' Vs. *белки*2 'proteins' enter separately, but the group for one of them does not include the others. The number of candidates varies from 1 to 12. The maximum number is for the shortest words, i.e. of three letters.

```
бездомный
  бездонный
  бездумный
бездумный
  бездомный
  безумный
безумный
  бездумный
бекон
  бетон
белеть
  белить
  болеть1
  болеть2
  велеть
  мелеть
белея
  болея
  мелея
белить
  белить
  делить
белка
  булка
  елка
  челка
  щелка
белки1
  балки
  бели
  булки
  челки
  щелки
белки2
  балки
  бели
  булки
  челки
  щелки
```

The main gain in candidate search is reached thanks to looking up only the candidates given in our dictionary. Using the total number of tries for a 9-letter Russian word, we get the gain coefficient G_1 = 616/2.25 = 274.

ClossLexica contains ca. 100,000 one-word grammemes. Even if after further replenishments the total number of groups would reach 6500, this will be only 6.5% of the whole systemic dictionary. Nevertheless, the revealed paronyms are supposedly the most frequent among content words. With the reasonable assumption that the rank distribution of all words in systemic dictionary conforms to Zipf law, these paronyms cover approximately 80% of all word occurrences in texts, and we have the additional gain coefficient $G_2 = \ln 100000 / \ln 6500 = 1.31$ owing to that all other 93,500 word are ignored in the candidate search. The global gain is $G_1 \times G_2 \approx 360$.

Morphemic paronyms

Several errors of a different nature were demonstrated above: *massy* Vs. *massive, sensible* Vs. *sensitive, revolution* Vs. *involution*. They are of the same POS and have the same radix (*mass-, sens-, -volu-*). In Russian linguistics, only this similarity is called paronymy. Confusions of morphemic paronyms are usual errors, especially for foreigners. For example, it is rather difficult to explain to them how to use Rus. paronyms *вислый* 'slouching', *висящий* 'hanging,' *висячий* 'bangled,' and *повисший* 'flagging' that differ only in one suffix and one prefix.

We have gathered morphemic paronyms into groups with the following additional requisites:

- Grammemes are taken as units of the dictionary, so that, e.g., *бок* 'side' and *бока* 'sides' are put into the same group;
- Grammemes of participles are considered as adjectives;
- All grammemes with homonymous radixes are pu to the same groups, e.g., adjectives *бур-ный* 'roaring,' *бур-овой* 'boring,' and *бур-ый* 'brown';
- Homonymous lexemes are given in the groups separately, however none of them can replace another;
- Two-root words are involved, one radix considered as the radix proper and another as the so-called suffixoid or a prefixoid. The negation *не* is a common prefix, the inseparable reflexive particle *–ся* is considered as suffix after the ending.

All in all, a morphemic paronym can be represented as a string $P_1...P_mRS_1...S_nE$, where $P_1,...,P_m$, m = 0, 1..., are symbols of prefixes; R is radix: $S_1,...,S_n$, n = 0, 1..., are suffixes; E is ending. The distance between paronyms within a group is measured by the number of elementary editing operations in the space of morphemic symbol strings. For example, *+отеч-еств*о* 'homeland' Vs. *отч-еств*о* 'patronym' and *+бед*а* Vs. *+бед*ы* are at the distance 0, *+волос-ат*ый* Vs. *волос-ист*ый*; *вы-нос* Vs. *из-нос*; *эффект-ив-н-ост*ь* Vs *эффект-н-ост*ь* are at the distance 1, *юнош-еск*ий* Vs. *юн*ый*, *гриб-н*ой* Vs. *гриб-к-ов*ый* are at the distance 2. Here the sign '+' initiates a radix; '–' a prefix or a suffix, '*' an ending. The differences in endings are ignored, since inflexional class is implied by POS and the previous suffix, and specific ending is different for each element of a grammeme.

Our dictionary of morphemic paronyms contains now 1120 paronymy groups with the mean length 5.65. A group element has on an average 1.4 paronyms at the distance 0 or 1. Summarize 'all-to-all' links in all groups at any distances, the total link number is up to 55,000, i.e. approximately 49 links within each group. Following is a fragment of the morphemic dictionary:

```
+бег*
  +бег*а
  +бег-л-ост*ь
  +бег-ств*о
  +бег-ун*
  +бег-ун-ок*
  +бег-ун*ья
  -на+бег*
  -при+беж-щ*е
  -про+бег*
  -про+беж-к*а
  -раз+бег*
  -у+беж-ищ*е
+бег-ающ*ий
  +бег-л*ый
  +бег-ов*ой
  +бег-ущ*ий
  -при+бег+ающ*ий
  -при+бег+ну-вш*ий
  -раз+беж-авш*ий-
ся
  -с+бег-ающ*ий
  -с+беж-авш*ий
  -у+бег-ающ*ий
  -у+беж-авш*ий
+бед*а
  +бед-н-ост*ь
  +бед-н-от*а
  +бед-ств-енн-
ост*ь
  +бед-ств-и*е
  +бед-ств-и*я
  +бед-ств-ован-и*е
  +бед*ы
  -о+бед-н-ени*е
+бед-н*еть
  +бед-овать
  +бед-ств*овать
  -о+бед-н*еть
+бед-н-еющ*ий
  +бед-н*ый
  +бед-ов*ый
  +бед-ств-енн*ый
  +бед-ств-ующ*ий
  -о+бед-н-евш*ый
  -о+бед-н-енн*ый
```

The search of morphemic errors is cut down by the same ways as for literal errors. If the suspicious word is in the dictionary, only its co-members are taken at the distant 0 or 1 to match. If the textual word is not available in the dictionary, no candidate of morphemic type is searched. We cannot compare our method with others quantitatively, since the latter do not exist. Indeed, the letter distance between morphemic paronyms is usually so high that their direct search in the literal space is absolutely impractical.

Conclusion

It is argued that correction of some semantic errors (namely, malapropisms) is possible by the use of paronyms, i.e. of words similar to each other in letters, sounds or morphs. It is proposed to compile paronymy dictionaries of three types beforehand. Literal paronyms essentially cut the search of correction candidates. Morphemic paronyms permit to quickly correct errors not studied so far and specific for foreigners. Russian dictionaries are already created—for literal and morphemic paronyms. The compiling of sound paronyms is the task for the future.

Bibliography

[Бельчиков & Панюшева, 1994] Бельчиков, Ю. А., М. С. Панюшева. Словарь паронимов современного русского языка. М.: Русский Язык, 1994.

[Bolshakov & Gelbukh, 2001] Bolshakov I. A., A. F. Gelbukh. A Very Large Database of Collocations and Semantic Links. In: Bouzeghoub et al. (eds.) Natural Language Processing and Information Systems. Natural Language Applications to Information Systems. Lecture Notes in Computer Science No. 1959, Springer, 2001, p. 103-114.

[Bolshakov & Gelbukh, 2002] Bolshakov, I. A., A. Gelbukh. Word Combinations as an Important Part of Modern Electronic Dictionaries. Procesamiento del Lenguaje Natural (Spain), No. 29, Sept. 2002, p. 47-54.

[Bolshakov, 2002] Bolshakov, I. A. Detección y Corrección de Malapropismos en Español mediante un Sistema Bietapa para Comprobar Colocaciones. Memorias del XI Congreso Internacional de Computación "Avances en Ciencias de la Computación e Ingeniería de Cómputo" CIC'2002, noviembre 2002, CIC-IPN, México, v. II, p. 303-313.

[Encyclopædia, 1998] The New Encyclopædia Britannica. Micropædia Vol. 7. Encyclopædia Britannica, Inc., 1998.

[Гусев & Саломатина, 2000] Гусев, В. Д., Н. В. Саломатина. Электронный словарь паронимов: версия 1. Научно-Техническая Информация (НТИ), Сер. 2, № 6, 2000, с. 34-41.

[Гусев & Саломатина, 2001] Гусев, В. Д., Н. В. Саломатина. Электронный словарь паронимов: версия 2. Научно-Техническая Информация (НТИ), Сер. 2, № 7, 2001, с. 26-33.

[Hirst & St-Onge, 1998] Hirst, G., D. St-Onge. Lexical Chains as Representation of Context for Detection and Corrections of Malapropisms. In: C. Fellbaum (ed.) WordNet: An Electronic Lexical Database. The MIT Press, 1998, p. 305-332.

[Hirst & Budanitsky, 1998] Hirst, G., A. Budanitsky. Correcting Real-Word Spelling Errors by Restoring Lexical Cohesion. Computational Linguistics (to be published).

[Kashyap & Oomen, 1981] Kashyap, R.L., B.I. Oomen. An effective algorithm for string correction using generalized edit distances. I. Description of the algorithm and its optimality. Information Science, 1981, Vol. 23, No. 2, p. 123-142.

[Mays et al., 1992] Mays, E., F.J. Damerau, R.L. Mercer. Context-based spelling correction. Information Processing and Management. 1992, Vol. 27, No. 5, p. 517-522.

[Vossen, 2000] Vossen, P. (ed.). 2000. EuroWordNet General Document. Vers. 3 final. www.hum.uva.nl/~ewn.

[Wagner & Fisher, 1974] Wagner, R.A., M. J. Fisher. The string-to-string correction problem. J. ACM, Vol. 21, No. 1, 1974, p. 168-173.

Author information

Igor A. Bolshakov – CIC-IPN, Research Professor; Center for Computing Research (CIC), National Polytechnic Institute (IPN), Av. Juan Dios Bátiz s/n esq. Av. Miguel Othón Mendizábal, Unidad Profesional "Adolfo Lopez Mateos", Col. Zacatenco, C.P. 07738, D.F., Mexico; e-mail: igor@cic.ipn.mx

Alexander Gelbukh – CIC-IPN, Research Professor and Chung-Ang University, Visiting Professor; Center for Computing Research (CIC), National Polytechnic Institute (IPN), Av. Juan Dios Bátiz s/n esq. Av. Miguel Othon Mendizabal, Unidad Profesional "Adolfo Lopez Mateos", Col. Zacatenco, C.P. 07738, D.F., Mexico; e-mail: gelbukh@cic.ipn.mx, Internet: www.gelbukh.com

TOWARDS COMPUTER-AIDED EDITING OF SCIENTIFIC AND TECHNICAL TEXTS

E. I. Bolshakova

***Abstract**: The paper discusses facilities of computer systems for editing scientific and technical texts, which partially automate functions of human editor and thus help the writer to improve text quality. Two experimental systems LINAR and CONUT developed in 90s to control the quality of Russian scientific and technical texts are briefly described; and general principles for designing more powerful editing systems are pointed out. Features of an editing system being now under development are outlined, primarily the underlying linguistic knowledge base and procedures controlling the text.*

***Keywords:** scientific and technical texts, automatic editing, linguistic knowledge base.*

Introduction

Scientific and technical writing is by no means easy, even for skilled and experienced authors. Usually, the elaboration of a good scientific or technical (sci-tech) text is iterative and time-consuming process, with several persons taking part in it. Besides an author of the document, colleagues, reviewers, and an editor participate in the process, helping the author to improve the text.

Scientific papers and technical documents are essential means of communication between scientists and engineers; therefore the efficacy of the communication depends on the quality of texts. A professional editor of sci-tech texts not only looks for grammar and spelling mistakes, but also accomplishes editing specific for functional style of scientific and technical prose: controlling word usage, revealing drawbacks in logic of reasoning, judging text organization, etc. [10]. The editor explains revealed defects and drawbacks, as well as proposes possible ways of how to overcome them, thereby helping the author to improve the text and to enhance its stylistic uniformity. Almost all sci-tech writers need some aid of professional editor, and without it they lack computer systems automating certain editor functions.

Of course, well-known universal computer text editors and spellers (e.g., MS Word) are widely used for preparing texts. These systems reveal many mistakes, including spelling and simple syntactic mistakes, and their facilities are permanently extended. But the universality of these systems means that they do not account for specificity of the particular text style and genre, in particular, sci-tech prose with its intensive usage of terms and the other highly standard units. Therefore, additional computer tools are needed for checking scientific and technical texts.

As a whole, sci-tech editing involves a wide spectrum of checks concerning different text levels, so that a deep syntactic, semantic, and logic analysis of the text are required. For this reason, it cannot be fully automated in the nearest future. Nevertheless, several computer systems were built for improving the quality of sci-tech documents, e.g., [1, 5, 7], demonstrating useful features. The systems can do many checks to provide initial editing that earlier has been done by editors or by the writer personally. Most systems are special-purpose editing systems, such as CRESS [5] designed to simplify texts in a narrow problem domain, namely, navy technologies.

Two experimental editing systems named correspondingly LINAR and CONUT were developed in 90s at Moscow State University (MSU) [1, 7]. While the former system was intended to control the quality of technical documents in the narrow subdomain of computer science, the latter was built to support students' practice in writing theses, primarily, to check students' texts with respect to the formal rules of text design, e.g., regularity of abbreviations, bibliography list, references, etc. Having encouraging results in the development of these systems, we aimed at development of a system with advanced facilities, regarding it as a further step towards automation of intellectual functions of sci-tech editor.

The paper shortly describes the systems LINAR and CONUT and summarizes experience of their development, in order to propose designing principles for future editing systems. Then, the research effort going on at the MSU to design more powerful system for checking the quality of sci-tech texts is discussed, including the incorporated linguistic knowledge base and procedures controlling the text. For the sake of clarity, specific features of sci-tech prose are outlined first, along with frequent defects of sci-tech texts.

Sci-Tech Prose: Norms and Defects

Functional style of scientific and technical prose comprises texts of various genres and particular types – research papers and monographs, theses and manuals, reviews and abstracts, technical reports and instructions, patents, etc. This style is admittedly the most distinctive one; its specialty stems from the necessity to express ideas in precise and simultaneously concise manner. The specialty concerns various language levels: lexis and phraseology, syntax, discourse, and composition. Norms and rules, as well as standard language devices and units optimizing sci-tech communication were formed on each level [8, 10].

Sci-tech lexis and phraseology comprises both terms of the particular terminology (e.g., *compiler*, *square root*) and common scientific words and expressions (e.g., *to test the hypothesis, for this reason, summing up*). Whereas specific terms denote concepts, objects, and processes of the particular domain, domain-independent common expressions are used to organize sci-tech text narrative, namely, to express the logic of reasoning, to connect text fragments devoted to different topics, and to structure the text. Among collocations taken from common scientific lexicon, there are clichés, which are relatively stable standard expressions exploited as ready-for-use colloquial formulas (e.g., *to outline directions of further research, the paper reports on*).

As regards the other language levels and corresponding devices and units, we should point out discourse devices. Sci-tech text narrative is organized in accordance with a number of typical discourse-composition frames, including specific ones used in texts of particular genres.

The commonly accepted norms and rules governing sci-tech texts are more of less completely explained in the books devoted to sci-tech writing, e.g. [6, 11]. The writer should follow these rules in order to obtain an accurate, informative, easy readable and understandable text. In particular, on the lexis level, terminological consistency and stability within the given document are required, which implies usage of a standard terminology, unambiguous nominating of concepts within a paper, as well as correct introducing of new terms into the text. We should note, by the way, that newly introduced terms, we call them author's terms, are inevitable in sci-tech prose. Indeed, in scientific papers they denote new concepts and ideas, while in technical documents author's terms designate certain processes and devices. Thereby, author's terms should be properly defined or explained as they are introduced.

To follow all commonly accepted norms and rules is rather difficult task for writers, so sci-tech texts often have various faults. Especially, many students' works (theses and abstracts) are full of defects, since students usually acquire writing skill mainly through their writing practice. Our experience in reading scientific papers in the domains of computer science and computational linguistics, as well as in reading and editing texts written by students shows that observed text faults vary on their nature. We have compiled and classified a list of frequent defects, which is presented below along with some illustrative examples (given in English and Russian languages) and short explanations of violated rules or requirements.

Inaccuracy in usage of special terms, both author's and generally accepted ones (the latter are usually referred to as dictionary terms): usage of new terms without their definition or explanation; unjustified usage of multiple term variants (i.g., grammar synonyms *swing smoothing method* and *method for smoothing of swings*, Rus. *сцепление и сцепка*). It is worth noting that terminological synonymy is not advisable, only a few term variants is acceptable.

Stylistic and grammar mistakes in combining words of common scientific lexicon and in standard clichés: wrong collocations (e.g., *to examine a problem* instead of *to study a problem*), mistakes in syntactic agreement and government, omission of clichés elements (e.g., Rus. *обращает внимание сходство* instead of *обращает на себя внимание сходство*).

Awkward phrases and sentences with multiple complex constituents, such as subordinate clauses, homogeneous parts of the sentence, and nested constructs in parenthesis; as well as several similar syntactic dependencies with the same preposition (e.g., *process of revision of complex fragments of expository texts*).

Syntactic ambiguity entailing semantic ambiguity: ambiguous anaphoric elements (mainly pronouns), ambiguous grammar structure of a word combination or of a sentences in the whole (e.g., Rus. *недостаток машин* – a lack of machines or a defect of machines?)

Syntactic heterogeneity of list items (e.g. *1) to consider material 2) to process data 3) analysis of results*); semantic incompatibility of homogeneous parts of the sentence (e.g., *compilation, interpretation, and translator*)

Drawbacks of discourse-composition structure: weak coherence of the text, lack of logic relations between text fragments (sentences and paragraphs). It is worth noting that such relations are normally expressed by connectors, such as *nevertheless, since, to this end*.

Violations of commonly accepted rules of sci-tech texts design, such as rules of citation, referring, numeration of text units, abbreviation of words and word combinations, etc.

Clearly, many of these defects and drawbacks are specific to sci-tech prose and are not controlled by universal commercial text editors. Meanwhile, special-purpose editing systems, such as LINAR and CONUT described below check for some presented defects.

Two Systems for Sci-Tech Text Editing

LINAR [7] seems the first system developed for editing Russian sci-tech text. It was intended to control the quality of Russian sci-tech documentation, primarily technical reports and texts of technical tasks on the theme "Architecture of multiprocessor systems".

Besides spelling control, LINAR provides a sufficiently wide set of specific checks. It reveals some style defects (such as presence in the phrase of several words with the same root: e.g., *functional, function*), defects of sentence structure (e.g., violations of neutral order of words), particular semantic defects (e.g., inconsistencies like *compiler, interpreter, and processor*), defects in text composition (such as absence of obligatory parts of document or improper order of parts). These facilities are based on relatively deep morphologic and syntactic parsing of words and phrases; elements of semantic analysis are used as well. Controlling procedures exploit several computer dictionaries, among them a large dictionary of word stems and a semantic dictionary (thesaurus) representing relations between terms of the given problem domain.

As compared with universal text editors, LINAR presents, besides diagnostics indicating revealed text defects and variants to correct them (if any), short explanations of the violated rules. To initiate checking process, the user specifies what checks are to be applied and to what text fragments. This feature of LINAR is connected with its module organization: its program kernel consists of procedures, and each of them is intended to control the particular rule or property (aspect) of text.

The system CONUT [1] was implemented in the late 90s for controlling and editing students' texts (theses and abstracts). It checks texts with respect to formal rules of text design, which comprise:

- ✓ consistency of abbreviations (their introduction and usage);
- ✓ correspondence of bibliography references in the text to the bibliography list;
- ✓ presence of obligatory parts of document (e.g., introductory part and table of contents);
- ✓ correctness of table of contents (its correspondence to headings and pages in the text);

- ✓ regularity of numeration of text units and pictures.

Such rules are considered formal because they do not concern the meaning of the text and its units.

CONUT can also estimate some aspects of style of the text, in particular, its simplicity (readability). A heuristic formula was proposed for this purpose, which accounts for various elements indicating sentence complexity (the number of words, punctuation marks, conjunctions, and pronouns in the sentence).

It is considered important that CONUT not only reveals defects in texts and estimates text style, but also explains the essence of formal rules and style estimation methods being applied. CONUT provides a reference guide accumulating information about formal rules of sci-tech text design. The guide is usable in checking process: when any defect is identified, a corresponding page of the guide (wherein the violated rule is explained) is given to the student. The reference guide is flexibly organized: its text material is represented as a hypertext, enabling both free navigation and learning the material in the recommended (predefined) order.

Comparing this system with LINAR, it should be noted that CONUT does not analyze text properties for which natural language parser is needed, most formal rules can be applied for the check without large dictionaries and syntactical-semantic analysis of phrases. Meanwhile, CONUT provides wider range of formal checks oriented just to revising students' texts and has advanced tutoring function.

Two described systems demonstrate some significant features, which ought to be considered while developing more powerful and helpful editing systems.

Designing Principles for Sci-Tech Text Editing Systems

Starting at MSU the development of new experimental editing system and having in mind the list of frequent defects described earlier, we intended to concentrate upon checks that are not fully implemented in LINAR and CONUT or are absent in them. At the same time, we take into account following crucial principles derived from our previous experience.

First, future sci-tech text editing systems will inevitably be based on semi (not fully) automatic editing procedures. One reason is obvious: the reliability of all automatic text processing programs is ranging approximately from 70 to 97%. Thus, the result of automatic checks of the text might be wrong, and proposed variants of how to revise text defects (if any) might be improper. So the author of the text should control all results and should ultimately choose ways of text revising. Hence, revising process implies a dialog between the user and the editing system, the latter indicates problems areas to be corrected and proposes variants of revision (as a human editor usually does) while the user makes appropriate decisions.

Second, it is not convenient to straightway apply all possible checks at user's text, since the spectrum of checks and estimations is sufficiently wide even in LINAR and CONUT, and their accomplishment might lead to a time-consuming process with vast diagnostics. It seems more reasonable when the user sequentially chooses desirable checks, initiates checking, and then analyzes obtained results, making necessary decisions. This feature of user interface determines the principle of system organization – the program kernel performing various text checks and estimations should be implemented as a set of procedures, each one controlling the particular rule or norm (for example, correctness of abbreviations) or estimating the particular aspect of the text. This principle was successfully tested within LINAR and CONUT; it enables to easily increment the editing power.

Third, although tutoring function of editing systems is clearly an auxiliary one, it is of no little significance. Comprehensive explanations of particular text defects, and also the explanatory information about formal and informal requirements to sci-tech documents can facilitate writing. Thus, it makes sense to reinforce editing systems with special dictionaries (such as systematic dictionary of common scientific words and expressions) and a reference guide explicating various norms and rules of sci-tech writing – even if some rules can not be checked so far in the particular system. For example, the guide can present explanations and typical examples of how to properly introduce new terms into scientific and technical texts. For students, such a guide can serve as a tool for systematic learning of sci-tech writing.

Forth and finally, in order to implement checks and estimations specific to sci-tech prose, an editing system should include vast linguistic knowledge base comprising both domain specific and domain independent components reflecting features of the prose.

According to discussed principles, main components of our novel system are the following: module implementing user interface, procedures controlling and estimating text properties, linguistic knowledge base, and reference guide that provides (besides explanation material) browsing data from the knowledge base.

Linguistic Knowledge Base

In order to ensure special-purpose control and estimation of Russian sci-tech texts, several linguistic components are being built into the system:

- Terminological dictionary [3] accumulating units, i.e., single and multi-word terms of commonly accepted terminology of computer science, gathered from several available text dictionaries. The dictionary includes known synonymous variants of terms, in particular, acronyms and other abbreviated forms (e.g. *central processing unit, CPU*). Most terms are nouns and noun combinations, but verbs are included as well. The dictionary represents relations between terminological units, primarily, class-subclass and part-whole relations, thus the dictionary can be regarded as thesaurus.
- List of definition templates [2] describing typical single-sentence definitions of new terms. Such definitions were compiled through manual scanning of sci-tech texts, they contain standard lexical units, such as nouns *term*, *name*, etc., verbs *call, refer, define*, etc. Definition template specifies both immanent lexical components and empty slots (places), their syntactic and semantic properties. An example of definition template is <*Ph*> *we will be call* <*N*>, where *N* denotes an author's term, and *Ph* is a noun phrase explaining its meaning.
- Dictionary of words and word expressions of common scientific lexicon [4]. It comprises both autosemantic and auxiliary words, noun and verb-noun combinations, adverb and participle expressions, compound prepositions and conjunctions. The dictionary unit represents adequate information: syntactic properties of word combinations (interrupted / uninterrupted, stable / free, semantic and syntactic valences, etc.), and semantic class and group of the unit within the proposed semantic classification. The classification comprises 5 main classes: text structuring and composing (e.g., *in addition, next*); expressing logical relations (e.g., *provided that, hence*); indicating sources of information (e.g., *in their opinion*), author's estimates (e.g., *essentially, it is quite likely*); structuring scientific knowledge via common scientific variables (generic nouns), such as *analysis*, *result.* For the latter class, dictionary units describe also syntactic combinability of the noun (e.g., *significant result, to derive result, to question result*).
- Dictionary of standard clichés (stable colloquial expressions) comprising both phrasal formulas (e.g., *the paper describes main features, argument can be made against*) and predicative constructs (e.g., *to outline directions of further research, to take as starting point for*). Some clichés are common for sci-tech prose, the others are specific for particular genres. Clichés are described by templates similar to definition templates: empty slots are indicated, and their syntactic and semantic properties are specified.
- Morphological dictionary of word stems, which covers all words encountered in the other dictionaries (separately or within any multi-word combination). The dictionary unit represents adequate morpho-syntactic information, e.g., part of speech and flexional class (if any), as well as pointers to units of the other dictionaries describing available combinations with the given word (stem). Thus, morphological dictionary connects units of different dictionaries, facilitating their recognition in texts.
- Inventory of prototype discourse frames specifying discourse-composition structures of sci-tech texts. Some frames are domain-specific, e.g., a frame specifying composition of texts that describe particular technical devices. Each slot of prototype frame corresponds to typical subtopic (e.g., functionality of device) and contains pointers to dictionary clichés and common scientific expressions that signal the subtopic in the text.

Text Analysis on Different Levels

We outline a few methods and procedures proposed to implement specific checks of sci-tech texts, which concern different text levels and exploit surface syntactical analysis.

Analyzing Terms and Common Scientific Expressions

For checking regularity of term usage and usage of common scientific expressions, terms and expressions should be recognized in texts. Identification of author's terms presents the major difficulty [2]. Indeed, they are free and often unstable multi-word nominal combination (e.g., *coefficient adjustment learning*) matching several possible syntactic patterns. Moreover, author's terms might be used in texts without any definition or explication, and in order to properly recognize them, local syntactic analysis should be complemented with elements of lexical semantics and term occurrence statistics.

An automatic recognition procedure is proposed [2] based on surface syntactic analysis and dictionary information. The procedure makes use of particular syntactic patterns of terms (e.g., coordinated combination of adjective and noun) and takes into account possible syntactical-semantic variants of new terms (such as *candidate elimination algorithm* and *algorithm for elimination of candidates*). The procedure exploits morphological analyzer converting words to their normalized forms and computes frequencies of term occurrences.

According to this procedure, occurrences of dictionary terms and collocations of common sci-tech lexicon are extracted first. Then, certain author's terms are identified by looking for sentences that match dictionary definition templates and by extracting lexical units from the proper places of the encountered sentences. And finally, the procedure attempts to recognize undefined author's terms: it detects word combinations of the given syntactic patterns, identifies among detected combinations different variants of the same term, and gathers them into groups of related term variants along with computed frequencies.

Units recognized at the last step of the procedure are regarded as term candidates, i.e., as potential new terms. Compiled list of term candidates is to be presented to the author of the text for validating and further revising, for example, selecting the most appropriate term in each group of related variants.

Identifying Syntax Ambiguity and Heterogeneity

We consider simple kinds of syntactic ambiguity, mainly ambiguous pronouns and syntactically ambiguous noun combinations. To check pronouns, local analysis of previous context is applied. For noun combinations we propose rather simple procedure of local syntactical analysis that checks dependencies of words within the noun combination.

To check syntactic homogeneity of items in lists, as well as homogeneous parts of the sentence, an automatic procedure is used, which differentiates noun and verb phrases from phrasal constructs and additionally distinguishes several types of noun and verb phrases.

Analyzing Discourse Structures

For recognition discourse structures of texts we propose a heuristic multi-step procedure, which exploits data from the dictionary of standard clichés and the dictionary of common scientific lexicon.

The recognition procedure first searches in the text for all sentences and composing clauses that match dictionary clichés. Occurrences of common scientific expressions signaling discourse relations are looking for as well. For this purpose, local context techniques similar to those described in [9] are used. When an instance of dictionary cliché or common scientific expression is recognized in the text, the procedure extracts lexical units from proper places of the instance and makes an attempt to fill slots of discourse frames related with this cliché or this expression. In general case, after recognition in the text of all cliché instances and common scientific expressions, slots of several discourse frames might be filled, therefore the frame with the maximum number of filled slots is selected as appropriate one. Information associated with this frame is used to estimate drawbacks of the recognized text structure, in particular, improper order of text parts devoted to particular subtopics.

Estimating Composition

Among parameters indicating quality of text composition, we consider proportionality of units of the same level, for example, proportionality of chapters or proportionality of items in itemized list. The proportionality means that units have comparable sizes, and the size can be computed as the number of words in the unit (another option is the number of units of the level underneath). It seems reasonable to estimate the proportionality of sentences within each paragraph, items within each itemized list, paragraphs within each chapter or text section, and chapters within the whole text.

To evaluate the proportionality, dispersion D and mean square deviation σ are calculated:

$$\sigma = \sqrt{D}\,, \qquad D = \frac{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}{n^2}$$

where x_i is the number of words in i-th component of the text, and n is the number of components.

Function exp$(-0.1*\sigma)$ is proposed to obtain a measure within the interval *(0,1)* and with maximum values corresponding to texts with better composition.

So the formula for a proportionality value is $E_C = exp(-0.1*\sigma)$. The formula is used systematically to estimate the proportionality of units on different levels: sentences, itemized lists, paragraphs, sections, chapters, and also headings of sections, with x_i denoting the number of words in units being estimated, and n

denoting the number of them. The estimation value for the whole text can be computed as the arithmetic mean of proportional values already computed for all text units.

Conclusion

While discussing the problem of computer-aided editing of scientific and technical texts we outlined peculiarities of special-purpose systems built for editing Russian text and pointed out the principles applicable for designing editing system with advanced facilities. We also outlined some features of a novel editing system being under development, primarily its linguistic knowledge base and procedures controlling the text. We hope that the system will be useful for a wide community of sci-tech writers, in particular for inexperienced authors, such as postgraduate students.

By now, the linguistic knowledge base of the system is partially implemented, with the terminology dictionary covering terms in certain narrow subfield of computer science and the dictionaries of common scientific lexicon and standard clichés containing more than 700 units. The first versions of the controlling procedures are tested.

Bibliography

1. Bolshakova, E. *Computer Assistance in Writing Technical and Scientific Texts*. Proceedings of 2nd International Symposium "Las Humanidades en la Educación Técnica ante el Siglo XXI", México, 27-29 September, 2000, p. 59-63.
2. Bolshakova, E. *Recognition of Author's Scientific and Technical Terms*. In: Computational Linguistics and Intelligent Text Processing. Second International Conference CICLing 2001. A. Gelbukh (Ed.). Lecture Notes in Computer Science, N 2004, Springer-Verlag, 2001, p. 281-290.
3. Bolshakova, E., Vasilieva N., Yudin D. *Extraction of Dictionary Terminological Word Combinations in Scientific and Technical Texts*. Proceedings of International Workshop on Computational Linguistics and its Applications Dialogue'2001. Russia, 2001, V. 2, p. 48-51 (in Russian).
4. Bolshakova, E. *Desingning Principles for a Computer Dictionary of Commom Scientific Lexicon.* Proceedings of International Workshop on Computational Linguistics and Intellectual Technologies Dialogue'2002. Russia, 2002, V. 1, p.19-23 (in Russian).
5. Glenda, M. *Readability Formulas: Useful or Useless?* IEEE Transactions on Professional Communications. Vol. PC-30, No 1, March, 1987, p.12-15.
6. Emerson, F.B. Technical Writing. Houghton Mufflin, 1987.
7. Malkovsky, M.G., Bolshakova E.I. Intellectual System for Control of Text Quality. In: Intellectual Systems, V. 2, No. 1-4, Moscow, 1997, p. 149 –155 (in Russian).
8. Mitrofanova, O. *Language of Scientific and Technical Literature*. Moscow University Press, 1973 (in Russian).
9. Paice, C., Jones P. *The Identification of Important Concepts in Highly Structured Technical Papers*. Proc. of 16th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburg, 1993, p.69-78.
10. Senkevich, M. *The Style of Scientific Speech and Literary Editing of Scientific Works*. Moscow, Vysshaia Shkola, 1976 (in Russian).
11. Zobel, J. Writing for Computer Science. Springer, 1997.

Author information

Elena I. Bolshakova is Docent of Moscow State Lomonossov' University, Faculty of Computational Mathematics and Cybernetic, Algorithmic Language Department; Adress: Leninskie Gory, Moscow State University, VMK, Moscow 119899, Russia; E-mail: eibolsh@aha.ru ; bolsh@cs.msu.su

MANAGING INTERVAL RESOURCES IN AUTOMATED PLANNING

V.Poggioni, A.Milani, M.Baioletti

***Abstract**: In this paper RDPPLan, a model for planning with quantitative resources specified as numerical intervals, is presented. Nearly all existing models of planning with resources require to specify exact values for updating resources modified by actions execution. In other words these models cannot deal with more realistic situations in which the resources quantities are not completely known but are bounded by intervals. The RDPPlan model allow to manage domains more tailored to real world, where preconditions and effects over quantitative resources can be specified by intervals of values, in addition mixed logical/quantitative and pure numerical goals can be posed. RDPPlan is based on non directional search over a planning graph, like DPPlan, from which it derives, it uses propagation rules which have been appropriately extended to the management of resource intervals. The propagation rules extended with resources must verify invariant properties over the planning graph which have been proven by the authors and guarantee the correctness of the approach. An implementation of the RDPPlan model is described with search strategies specifically developed for interval resources.*

***Keywords**: AI, Automated Planning, Planning with resources, Propagation rule, Search strategies.*

Introduction

Various models have been proposed for extending the pure logical classical planning models in order to manage more real world features. A very promising issue toward this goal is the research line which aims to provide the planners with the ability of planning with resources. In this framework, in addition to the logical relationships among domain objects, operators and states, the planning models are able to cope with quantitative aspects of the world, such as actions which involves consumable/reusable resources, domain constraints on resources, goals involving quantities.

Several planning models for resources management have been proposed for extending virtually all the most successful planner approaches; among them it is worth noticing models UCPOP—like models [4], Graphplan-like [12], SAT—like [15,16], and also HTN based approaches [5].

The types and features of the modelled resources are also varying from unary and discrete resources [13] to reusable resources [4] and conjunctive constraints over resources [6,9]. A different approach is that of planners specialised in the management of time, as a quantitative resource; these planners allow the management of extension such as propositions which holds over time intervals , actions with durations and complex time numerical constraints [6,10]. The issue of a complex time management is beyond the scope of this work.

It is worth noticing that the introduction of quantitative resources in a planning framework has brought into planning some typical issues of scheduling and CSP, such as optimisation search and constraints management. Moreover having quantitative resources also change the typical view a planning problem can be regarded to. At the simplest level there are "pure logical" problem goals which can be specified as in the classical framework , nevertheless the plan generation phase will have to take into account of resources precondition/effects; the problem can otherwise specify "mixed" logical/quantitative goals, or even "pure quantitative goals", (e.g. consider the problem of finding a plan for the purely quantitative goals: Consume at least 100 calories, Produce 1 Billion profit etc.); finally pure/mixed "optimisation" goals can also be specified, where no logical or quantitative goals exist (e.g. consider the problem: "producing as much profit as you can"), this latter optimisation aspect has been incorporated in PDDL 2.1 [7] where it is possible to specify an object function to be optimised.

Models of planning with resources certainly represent an important step toward a more accurate model of the real world, but, on the other hand, most of the proposed models fail to give any account of the potential uncertainty which can affect the quantities related to resources. Many facts in a real world state can be described in a satisfactory way by a boolean proposition (e.g. (on A table) (open door) etc.), but it is not very realistic to assume that an exact number can model the continuous quantities describing a given resource. Most models of planning with resources allow to describe non exact quantities in preconditions, such as, for example, an interval of values that the resources can assume in order make the action to executable (e.g. the fuel must be between 10 and 30, the voltage must be between 210 and 230, this preconditions can be modeled both in [12] as well in [9]), surprisingly the same planning models do not allow to specify intervals of values in action effects. In fact models of planning with resources admit updates and assignment operations which allow functional quantities (for example consumed_fuel can be functionally computed by distance * fuel_consumption) but where the increment of the current resource level is a single well determined numerical value (e.g. *consumed_fuel = distance * fuel_consumption = 12 * 0.25 =3* that is a single value) [7,9,12]. Indeed, it seems to be an apparent contradiction that the semantics of preconditions can be also given in terms of non exact quantities, while the semantics of effects have to be given only in term of precise values.

In this work we show how this gap between non exact preconditions and fully specified effects can be bridged by specifying in both cases quantities varying over intervals. RDPPlan, the model of planning with resources which we will describe, is based on DPPlan [1], a planner which uses a non directional search algorithm on the planning graph. RDPPlan is compatible with the resources model as described in standard PDDL 2.1, and extends it by allowing updates and assignments of quantities specified by intervals.

In the following paragraphs, after recalling the main features of DPPlan, it is introduced RDPPlan, the planning model with resources, showing that the addition of resources management does not have a great impact on the overall DPPlan approach and on the planning graph structure. Resources management in RDPPlan is realized by the modification of propagation rules and the introduction of appropriate rules for failure detection caused by resources constraints violation. Moreover, the structure and the algorithms we provide for resources management can be easily extended for update operations which operate over intervals, as shown in the fourth paragraph.

It is worth noticing that RDPPlan architecture is not committed to any particular search strategy (i.e. the resources management and failure detection is embedded into the propagation rules), the consequence is that search strategies can be easily added to RDPPlan. Strategies specifically developed for resources are described in paragraph five.

Examples and experimental results show that this approach seems to be appropriate for modeling real world situations where the consumption/production of resources cannot be expressed by a single value (for example, assuming that f*uel_consumption per Km* is a non exact quantity which ranges between *[0.25, 0.33]*, then *consumed_fuel* can be calculated as an interval by computing distance * *fuel_consumption = 12 * [0.25,0.33] = [3, 4]*).

Finally we point out some possible topics which are worth to be investigated in the RDPPlan framework, such as further strategies and heuristics for problems with resources, and further extensions to the resource model, like managements of fuzzy quantities.

DPPlan and its propagation rules

DPPlan is mainly based on GraphPlan [2]. With this planner it shares the same representation of disjunctive states, obtained by connecting facts and operators to form a graph, called *planning graph*.

DPPlan has, with respect to GraphPlan and other related planners, like IPP [11] or STAN, a completely different method for searching a solution in the planning graph.

The fundamental feature of DPPlan is that to each node of the graph a boolean value is assigned. An operator is *true* if it is executed, *false* if it is not. A fact is *true* if it is achieved by some operator, *false* otherwise. During the search phase a fact *p* can be *true* also if it is required by some operator *o* (*p* is a precondition of *o*) and *false* if it is required to be false (*p* is a negative precondition of *o*). This fact makes possible to use the propagation rules before the fact is really achieved and to cause, in the case, a backtracking earlier than it would be else obtained.

However those two situations are very different: if a fact *f* is *true* because something requires it, then *f* must be seen as a (sub-)goal and the current plan cannot be correct if this fact is not reached by any action. Only when an action achieving *f* is added to the plan, then *f* is really "*true*". The same distinction should be made between a fact which is really "*false*" (it has been deleted by some operator, or all of its achiever is false) and a fact required to be *false* (some action has it as a negative precondition).

In order to distinguish these situations, a further value, called *state*, is assigned to each fact:

- the state is "*produced*", when the fact value is reached (either true or false),
- the state is "*consumed*", if the fact value is required (either true or false), and
- the state is "*produced-after-consumed*" if the fact value has been required and then achieved.

This last value of state is useful during the backtracking phase.

In the previous paper of DPPlan we have described the rules by which it is possible to propagate a choice on the value of a node in the graph. These rules show how to update the other nodes of graph because of that choice. The operations of changing state and value of a node are named according to the type of changing and to the type of node they are applied.

The operation **use** sets the value of an operator node to *true*, while the operation **exclude** sets it to *false*.

The operation **consume** sets the value of a fact node to *true* and the state to *consumed*, **produce** sets the value to *true* and the state to *produced* (or *produced-after-consumed*), **consume-not** sets the value to *false* and the state to *consumed*, **destroy** sets the value to *false* and the state to *produced* (or *produced-after-consumed*).

All of these operations can fail if they try to give a different value to an already valued node (e.g. **destroy** and **consume** on the same fact) or can be ignored if they try to give the same (or compatible) value or state (e.g. **produce** and then **consume** on the same fact). For further details see the original paper [1].

For an operator *o*, *In(o), NotIn(o), Out(o)* and *NotOut(o)* are respectively the list of its positive preconditions, negative preconditions, positive effects and negative effects. For a fact *f*, *In(f), NotIn(f), Out(f)* and *NotOut(f)* are respectively the list of the operators which have *f* as a positive effect, as a negative effect, as a positive precondition and as a negative precondition; moreover *npp(f)* and *npd(f)* are respectively the number of possible producer (destroyer), i.e. the number of elements of list *In* and *NotIn* whose value is still undefined. Finally, for any node *n*, *Mutex(n)* is the list of the nodes exclusive of *n*.

These propagation rules update the state and the value to the node they are applied, as well as the goal list. Note that without any particular additional procedure, DPPlan is able to solve problem with negative preconditions and goals, and with temporally qualified "*initial states*", like the fact *f* is true at time $t > 0$, and goals, like the goal *g* has to be achieved at time $t < t_{max}$.

The main algorithm operates in way resembling the celebrate Davis-Putnam algorithm for propositional logic. At the beginning all the variables receive the value *undefined*, then the procedure *produce* is performed for all the facts in the initial state, *consume* is performed for all the positive goals, and *consume-not* is performed for all the negative goals.

In its main loop, the algorithm chooses an undefined variable *v*, tries to set its value to one of the boolean values (e.g. *true*) until it finds a solution (no fact is in the state *consumed*), or some propagation fails, in this case a backtracking phase is performed by undoing all the propagations done after *v* and tries to set the value *v* to the opposite value (e.g. *false*). If a failure is obtained again, *v* can be neither *true* nor *false*, therefore the backtracking stops until the previously tried variable is unset: now the value of this variable is reversed and the search goes on. Note that when the algorithm has tried both the value for the first chosen variable, without reaching a solution, the search phase is ended, the graph is augmented with all the new applicable operators and all the new facts they produce, in a way similar to the expansion phase of GraphPlan, and a new search phase is performed.

What is completely free in our algorithm is how to choose what variable to try next and which value to try first. According to the method used, the planner can perform a forward search, a backward-chain search, a bidirectional search or simply a non directional search.

In the original paper [1] we have listed several ways of choosing a variable, essentially an operator to be tried to *use*, and then to *exclude*.

RDDPlan: Planning with numerical resources in DPPlan

RDPPlan is the extension of DPPlan to handle numerical resources. In the model together with the logical propositions, the use of numerical variables is allowed. Each numerical variable, called *resource*, can be cited as in preconditions and goals as well as in effects and initial states. Conforming to the PDDL 2.1 [7], resources are represented as a numerical functions whose parameters can be domain constants or action parameters.

Preconditions and goals

In preconditions and goals it is allowed to use conditions like `(compare resource value)` where `compare` is a comparison operator `(>, <, >=, <=, =)` and `value` can be an expression involving action parameters, numerical constants and arithmetic operators.

Since we do not allow for disjunctive preconditions and goals, several constraints on the same resource *r* reduce to a unique real interval, possibly empty, indicated with $P_{A,r}$ for the precondition of action *A* and with G_r for goals. These intervals are possibly unlimited in left and/or in right side.

Initial state and effects

In the initial state the resources are initialized by the proposition `(= resource value)` where now the value can only be a numerical constant.

In the effects a resource can be changed by proposition `(change resource value)` where `change` can be one of the operators `assign, increase, decrease` and `value` is an expression as in the preconditions. We indicate the value added of action *A* to resource *r* with $E_{A,r}$, intending $E_{A,r}=value$ for *increase* and $E_{A,r}= -value$ for *decrease*. For an action *A* which does not change a resource *r* we treat *A* as an increase operator of value 0, i.e. $E_{A,r}=0$.

As a syntactic sugar we allow to use, while expressing preconditions and effects, in the expressions called *value* some other resources, provided they are static, i.e. initialized in the initial state, but not changed by any action. A static resource is a sort of constant which remains unchanged during the plan, like weight function in domain *Depots*.

This restriction has the main effect that each resource can be changed independently from the others. Allowing to cite other (non static) resources in the preconditions would have generated more complex admissible resource domains, not reducible to the cartesian product of real intervals. On the other hand if an effect on resource *r* could depend on the value of some other (non static) resource, some interaction between resources would have arisen which are difficult to handle (e.g. increasing a resource can cause to another resource to decrease).

Realized value and desired interval

Associated to each resource *r* and each time level *t*, a numerical value R_{rt} and a real interval D_{rt} is computed during the planning phase. Every time an action is selected by the search procedure, R_{rt} and D_{rt} are updated for every resource *r* used by the action and for every time level *t* greater or equal to the time level at which the action is selected. The previous values of R_{rt} and D_{rt} are restored during the backtracking phase.

The number R_{rt} represents the current value of the resource *r* at time *t* realized by the actions since now inserted in the plan. At the start of search procedure, for each resource *r* and for every time *t*, R_{rt} is set to the value specified at the initial state.

The interval D_{rt}, called "*desired interval*", contains all the admissible values for the resource *r* that allow the execution of all the actions selected at time level *t*. At the start of search procedure for each resource *r* and for every time $t<T$ (*T* is the last time level), D_{rt} is set to $[-\infty,+\infty]$, while for each resource *r*, D_{rT} is set to G_r, the interval specified by the goals.

Solution plans and executability

Definition 1 *The obvious sufficient and necessary condition for a plan to be executable and to be a solution of the given planning problem is that for each resource r and for each time level t the condition* $R_{rt} \in D_{rt}$ *holds.*

Before describing the rules with which R_{rt} and D_{rt} are updated, we must define when two or more actions are executable at the same time level. We use the same concept of simultaneous executability as expressed in [7,15].

Definition 2 *A set of actions* $A_1, A_2,\ldots, A_m$ *is simultaneously executable if for every permutation* Π *of the actions in the set* $A_{\Pi(1)}, A_{\Pi(2)}, \ldots, A_{\Pi(m)}$

1) $A_{\Pi(1)}$ *is executable in the current state,* $A_{\Pi(2)}$ *is executable in the state after the execution of* $A_{\Pi(1)}$, $A_{\Pi(3)}$ *is executable in the state after the execution of* $A_{\Pi(1)}$ *and then* $A_{\Pi(2)}$, *and so on,*

2) the effect over the resources is always the same.

As a straightforward consequence of the second condition, an assignment on resource *r* is not simultaneously executable with any action changing *r* (additive operator).

The question remains open whether to allow an action changing *r* to be simultaneous with any action having a precondition with respect to *r*. Our approach is to allow simultaneity whenever the change does not affect the executability.

Proposition 1 *It is easy to prove that two additive actions* A_1 *and* A_2 *are simultaneously executable (with respect to r) if, let* $[\alpha_1,\beta_1]=P_{A1,r}$ *and* $[\alpha_2,\beta_2]=P_{A2,r}$ *be respectively the precondition intervals and let* $k_1=E_{A1,r}$ *and* $k_2=E_{A2,r}$ *their effect on r, we have* $\alpha \le \beta$ *where* $\alpha=\max\{\alpha_1,\alpha_1-k_2,\alpha_2,\alpha_2-k_1\}$ *and* $\beta=\min\{\beta_1, \beta_1-k_2, \beta_2, \beta_2-k_1\}$. *In* *the positive case we set $D_{rt}=[\alpha, \beta]$ if actions A_1 and A_2 are the only actions to be executed at time t. Otherwise A_1 and A_2 are marked to be mutually exclusive.*

The generalization to the case of many additive (over the same resource *r*) actions A_1, A_2, …, A_m is somehow straightforward.

<u>*Proposition 2*</u> *Called $[\alpha_i,\beta_i]=P_{Ai,r}$ the interval precondition and $k_i=E_{Ai,r}$ their effects, for i=1,…,m, we have that the action A_i's are simultaneously executable (with respect to r) if $\alpha \leq \beta$ where α = max $\{\alpha_i\text{-}km_i : i=1,\dots,m\}$, β = min$\{\beta_i\text{-}kp_i : i=1,\dots,m\}$ and* $kp_i = \sum_{j \neq i, k_j > 0} k_j$ *and* $km_i = \sum_{j \neq i, k_j < 0} k_j$ *for i=1,…,m.*

In the positive case $D_{rt}=[\alpha, \beta]$ if the actions A_i's are the only actions to be executed at time t..

α and β can be computed in linear time (in the number of resources and actions) by storing (and keeping updated) the values $kp_0 = \sum_{k_j > 0} k_j$ and $km_0 = \sum_{k_j < 0} k_j$.

The case of several additive actions includes the case of simultaneous execution of additive actions on resource *r* (possibly none) with other actions which do not change *r*. In the particular case, where all the actions do not change *r*, D_{rt} reduces to the intersection of all the precondition intervals.

A case not yet covered is the simultaneous execution of an assignment on *r*, say A_1 whose assigns to *r* the value *v*, with with actions $A_2,\dots,A_m$ which do not change *r*. It is easy to see that $A_1,A_2,\dots,A_m$ are simultaneous executable (with respect to *r*) if the intersection of all the precondition intervals is not empty and contains *v*.

After a new action *A* is selected to be *used* at time *t*, we must check if it is simultaneously executable with the already selected actions at time *t*, by computing for each resource *r* the quantities $\alpha(r)$ and $\beta(r)$. Only when each of these interval is not empty, then the desired interval for *r* is set to be $[\alpha(r), \beta(r)]$. Moreover an incremental way, which takes only a constant time to be computed, of updating D_{rt}, after the *use* of a new additive operator *A*, is the following.

<u>*Proposition 3*</u> *If $D_{rt} = [\alpha,\beta]$, $P_{A,r} = [a,b]$ and $E_{A,r} = k$, then the updated desired interval is $[\alpha',\beta']$ where $\alpha' = \max\{a - km_0, \alpha - \min\{k,0\}\}$ and $\beta' = \min\{b - kp_0, \beta - \max\{k,0\}\}$.*

Updating the values R_{rt} is done by recomputing them for every resource changed by *A*, starting from time *t+1* and ending at the first time where an assignment over *r* is selected. This computation can be efficiently done by storing and keeping updated the total amount to be added to *r*, computed considering all the actions selected since now.

Extension to interval resources

A very straightforward, yet significant, extension of the simple model above explained is to allow for non completely specified initial states and effects.

Interval on the initial states and effects

Instead of initializing a resource with a unique real value, we allow to specify a real interval I_r as a range for the initial value of the resource *r*. The planner operates in an under-specified domain in which the value of some resource is not exactly known, but it is bound to be in an interval. Suppose we do not know exactly how much gasoline is in the tank of our car: we just know that it surely the real amount is between 5 and 10 liters.

Similarly it is possible to have under-specified effects of any operator: the value which is added, subtracted or assigned to the current value of a resource is not exactly known, but only a lower and an upper bound is specified. Imagine that the car in the previous example, we do not know which is the exact consumption: all we know is that the car can travel from 10 to 15 kilometers per liter.

In this enhanced model, the real quantities $E_{A,r}$ and R_{rt} are therefore replaced by real intervals, which cannot be unlimited in the left or in the right side. The intervals R_{rt} are initialized with I_r, the intervals specified in the initial state description, and are updated according the following simple rules, where the current interval for R_{rt} is $[\gamma,\delta]$, the operator to be executed is *A* and $E_{A,r} = [e_{min}, e_{max}]$: if *A* is `ASSIGN` R_{rt} becomes $[e_{min}, e_{max}]$, if *A* is `INCREASE` R_{rt} becomes $[\gamma+e_{min}, \delta+e_{max}]$ and if *A* is `DECREASE` R_{rt} becomes $[\gamma - e_{max}, \delta - e_{min}]$.

Solution plans and executability

The definition of what solution plan is meant is similar to the definition described in the previous section.

<u>*Definition 3*</u> *For this model the necessary and sufficient condition for a plan to be a solution of a given planning problem is that for each resource r and time level t, $R_{rt} \subseteq D_{rt}$.*

The intended semantics of this meaning of the term *solution plan* Π is that if for every possible way of replacing each interval effect $E_{A,r}$ with a number $e_{A,r} \in E_{A,r}$ and of replacing each initial interval I_r with a number $i_r \in I_r$, the problem so obtained, which now is conform to the previous model, is solved by Π, according to the semantics expressed in the previous section. Expressed in other terms, a solution plan must solve every possible problem that is allowed by the constraints specified in the initial state and in the effects description.

The update rules for desired interval are similar to what we have seen in the previous section.

<u>Proposition 4</u> *Suppose that A_1 and A_2 be two additive actions with precondition intervals $P_{A1,r}=[\alpha_1,\beta_1]$ and $P_{A2,r}=[\alpha_2,\beta_2]$ and with effect intervals $E_{A1,r}=[e_{min,1},e_{max,1}]$ and $E_{A2,r}=[e_{min,2},e_{max,2}]$ respectively. If $\alpha \le \beta$, where $\alpha = max\{\alpha_1, \alpha_1\text{-}e_{min,2}, \alpha_2, \alpha_2\text{-}e_{min,1}\}$ and $\beta = min\{\beta_1, \beta_1\text{-}e_{max,2}, \beta_2, \beta_2\text{-}e_{max,1}\}$, then A_1,A_2 are simultaneously executable.*

The generalization to the cases with m additive or multiplicative actions are the following.

<u>Proposition 5</u> Called $P_{Ai,r}=[\alpha_i, \beta_i]$ the precondition intervals, for i=1,…,m and $E_{Ai,r}=[e_{min,i},e_{max,i}]$ the effect intervals over r for i=1,…,m, we have that the action A_i's are simultaneously executable if $\alpha \le \beta$, where $\alpha = max\{\alpha_i\text{-}km_i : i=1,\dots,m\}$, $\beta = min\{\beta_i\text{-}kp_i : i=1,\dots,m\}$ and $kp_i = \sum_{j \ne i, e_{max,j} > 0} e_{max,j}$ and $km_i = \sum_{j \ne i, e_{min,j} < 0} e_{min,j}$.

Strategies on Resources

In this section we present the strategies that we have defined for achieving the goals over resources. These strategies are necessary to solve "*pure numerical problems*", i.e. problems with goals only on resources. The methods that implement these strategies are combined with the ones for solving logical goals, by evaluating the difficulties of resources and logical goals and selecting the most difficult goal to solve.

Strategy for numerical resources

For each time step *t* and for each resource *r*, we check if the condition $R_{rt} \in D_{rt}$ holds: the negative cases are the goals on resources. An action that can help solving a goal on resource *r* at time *t* can be chosen according to the following rules.

First, the algorithm searches for an action that can achieve the goal in one only step, preferring, in the case of many available options, the action situated at the level nearest to *t*. If such an action does not exist, we choose an increaser (actions which make R_{rt} bigger) or a decreaser (actions which make R_{rt} smaller) according to the case. In the detail, a first search is performed for all those actions *A* present at any time $\tau < t$ such that the updated value (if *A* would be executed) of R_{rt}, say R'_{rt}, verifies the condition $R'_{rt} \in D_{rt}$. Among those, the action *A* with the highest τ is selected, by performing the search starting from the time-step *t–1* and going backward: the first action found is chosen.

If the first search fails, the algorithm chooses an action that can permit us to come closer to the goal. Called $[\alpha,\beta]$ the interval D_{rt}, the algorithm selects an increaser *A*, if $R_{rt} < \alpha$, or a decreaser *A*, if $R_{rt} > \beta$, that minimizes $\min\{|\alpha - R'_{rt}|, |R'_{rt} - \beta|\}$.

Strategy for interval resources

When we work with resources as intervals using the "interval algebra" explained in a previous section, we have to handle with real intervals whose width can in general only grow, except when an assignment is performed. In fact it is obvious that any numerical operation between intervals produce as a result an interval which has a width larger than the original widths.

The following example can show this characteristic. If we have in the tank an amount of fuel that we do not know

exactly, but that we know be between *10* and *15* liters (this is a case of incomplete knowledge in initial state) and we take from an another tank an unknown amount of fuel between *12* and *16* liters (example of non deterministic effects over resource), then the minimum amount of fuel in the first tank is *22* liters and the maximum is *31* liters. So, in the notations here used, from the rule $R_{rt} + E_{A,r} = R_{r,t+1}$, we will obtain *[10,15] + [12,16] = [22,31]*. This means that we have at time step *t*, before the action application, a realized interval with width *5* and then, at the next time-step *t +1*, an interval with width *9*.

If you think that the width of realized interval represents, in some sense, the indetermination on resource value, we have that the larger the interval width, the larger the indetermination. Moreover note that if the width of the realized interval is large, it is more difficult that the solution conditions $R_{rt} \subset D_{rt}$ will hold. Let $|R|$ denote the width of interval $R=[a,b]$, i.e. $|R|=b-a$. The first control to do is on the widths of realized and desired intervals.

1) If $|R_{rt}| > |D_{rt}|$ an assignment which assigns an interval with width less than $|D_{rt}|$ is the only possible choice. If there are many such assignments, the algorithm chooses that one which assigns the interval with the least width. If there are no assignments with this property, a backtracking is necessary.
2) If $|R_{rt}| \leq |D_{rt}|$ the algorithm tries to solve this goal using a procedure similar to that described in the previous section. Now the choice criteria takes into account the distance between D_{rt} and R'_{rt} (which is the realized interval updated after the execution of the action to be evaluated) and their widths.

The previously described criteria can be implemented by defining two preference functions, one for the interval widths, and one for the distance between intervals, and by searching for an action that maximizes a linear combination of the functions. The first function is $f_W(A, D_{rt}) = \frac{|D_{rt}| - |R'_{rt}|}{|D_{rt}|}$ and gives to each action A a numerical positive score between 0 and 1.

The second function is a decreasing function of the distance between the middle points of the two intervals $f_D(A, D_{rt}) = \exp(-\frac{1}{2}|\gamma + \delta - \alpha - \beta|)$ where $[\alpha,\beta] = D_{rt}$ and $[\gamma,\delta] = R'_{rt}$.

Also this function gives to each action A a numerical positive score between 0 and 1.

Experimental results show that good values for the ratio c_1/c_2 of the coefficients of the linear combination $f(A, D_{rt}) = c_1 f_W(A, D_{rt}) + c_2 f_D(A, D_{rt})$ are between 1 and 2.

A theoretical justification is that f_W is slightly more influent for driving the search algorithm, because it can be useless to get closer to the desired interval if the width of the realized interval is too large.

Conclusion

RDDPlan a model of planning with interval resources based on propagation rules has been described. This models seems to be more adequate than existing models of planning in order to describe real world operators which uses resources because it does not require a complete knowledge of the quantity to be updated, but an interval boundary.

The main contribution to RDPPlan comes from [1], whose propagation and failure rules have been extended with interval management. Other related works share with it a similar planning graph structure with a different semantics for resources and no management of intervals [12,15]. Although an actual planner and strategies for resources have been implemented on the basis of the proposed model, RDPPlan can be considered a platform on which strategies and heuristics for planning with resources can be experimented. Further investigations and experiments are planned in order to develop more accurate heuristics and strategies which takes into account of resources, moreover, in order to provide a meaningful evaluation it will be also required the development of a set of significant benchmarks for planning domains with interval resources.

Finally it is worth investigating further extensions to the resources model more accurate with respect to the uncertainty in the real world e.g. intervals with given probability distribution over resources values and fuzzy quantities.

Bibliography

[1] M.Baioletti,S.Marcugini,A.Milani. DPPlan: an algorithm for fast solution extraction from a planning graph. *In Proc. of AIPS-00*, 2000.

[2] A.Blum,M.Furst. Fast planning through planning graph analisys. *Artificial Intelligence (90):281-300*, 1997.

[3] B.Bonet,H.Geffner. Planning as heuristic search. *Artificial Intelligence 129*, 2001.

[4] S.Chien et al. ASPEN automated planning and scheduling for space mission operation. *In Proc of SpaceOps2000*, 2000.

[5] K.Currie,A.Tate. O-Plan: The open planning architecture. *Artificial Intelligence (52):49-86*, 1991.

[6] M.Do,S.Kambhampati. Sapa: A domain indipendent heuristic metric temporal planner. *In Proc. of ECP-01*, 2001

[7] M.Fox,D.Long. PDDL 2.1: An extension to PDDL for expressing temporal planning domains. Forthcoming in JAIR special issue on 3rd International Planning Competition

[8] J.Hoffmann,B.Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research (14): 253-302*, 2001.

[9] J.Hoffmann. Extending FF to numeric state variables. *In Proc. of ECAI-02*, 2002.

[10] A.Jonsson et al. Planning in the interplanetary space : theory and practice. *In Proc. of AIPS-00*, 2000.

[11] J.Koehler,B.Nebel,J.Hoffmann,Y.Dimopoulos. Extending planning graphs to an ADL subset. *In Proc. of ECP-97*, 1997.

[12] J.Koehler. Planning under resource constraints. *In Proc. of ECAI-98*, 1998.

[13] P.Laborie,M.Ghallab. Planning with sharable resource constraints. *In Proc of IJCAI-95*, 1995.

[14] X.Nguyen,S.Kambhampati,R.S.Nigenda. Planning graph as the basis for deriving heuristics for plan synthesis by state space and csp search. *ASU Technical Report*, 2002.

[15] J.Rintanen,H.Jungholt. Numeric state variables in constraint based planning. *In Proc. of ECP-99*, 1999.

[16] S.Wolfman,D.Weld. The LPSAT engine and its application to resource planning. *In Proc. of IJCAI-99*, 1999.

Authors information

Marco Baioletti – Dipartimento di Metodi Quantitativi, Univeristà degli Studi di Siena, P.zza S.Francesco 5, Siena, Italy

Alfredo Milani - Dipartmento di Matematica e Informatica, Università degli studi di Perugia, Via Vanvitelli, 06100 Perugia, Italy

Valentina Poggioni – Dipartimento di informatica e Automazione, Università degli Studi di Roma Tre, Via della Vasca Navale 79, 00146 Roma, Italy

A PLANNING MODEL WITH RESOURCES IN E-LEARNING

G. Totkov, E. Somova

Abstract: *This work proposes a model for planning of education based on resources and layers. Each learning material or concept is determined by certain characteristics: a layer and a list of resources and resource values. Models of studied subject domain, learner, information and verification unit, learning material, plan of education and education have been defined. The plan of education can be conventional, statical, author's and dynamic. Algorithms for course generation, dynamic plan generation and carrying out education are presented. The proposed model for planning of education based on resources and layers has been included in the system PeU.*

Keywords: *planning education, e-learning.*

Introduction

The e-learning action plan has been created by the Commission of the European Communities in 2001. The development of this plan shows the actuality of the present work. A lot of attempts for creation the e-learning standards with necessary requirement (SCORM, CMI, LOM, IMS, ARIADNE) and systems for creation of e-learning environments (Learning Space, WebCT, Top Class, First Class, Blackboard, Virtual-U, Web course in a Box, CourseInfo, Learning Landscapes, CoSE, CoMentor, ARIADNE, Asymetrix Librarian, Norton Connect, Allaire Forum, Team Wave, WebBoard, Asymetrix ToolBook, etc.) [Britain] have been done. In order to satisfy the requirements of the e-learning environments, the model of planning of education have to be done.

The known models for planning of education are classical [Koffman, 1975; Пасхин, 1985; Савельев, 1986; Зайцева, 1989; Grandbastien, 1994; Grant, 1997] and resource [Milani, 2000; Milani, 2001a; Milani, 2001b] respectively based on the classical and recourse models of problem planning.

The classical model for planning [Chien] is based on the initial and the final state of the problem and the operators for transformation of one state into another. The Chien model can not represent multiple used educational environment, where learners are interested in optimizing the path for passing over learning materials, not only from the point of view of the length of the path, but also depending on the time for passing over, price, level of difficulty, etc.

The classical model is expanded in [Koehler, 1998; Rintanen, 1999; Milani, 2001a] by adding resources (level, time, price, etc.). The disadvantage of the exiting models with resources is that they do not consider the structure and content of learning materials, the studied concepts and their representation (introduction, definition, example, classification, comparison, application, etc.). These representations we will call layers.

This work proposes a model for ***planning of education based on resources and layers***. Each learning material or concept is determined by certain characteristics: a layer and a list of resources and resource values [Somova, 2002].

Studied subject domain

Let $N = \{N_v\}_{v=1}^{h}$ be a set of learning mataterials' names from a different type, $L = \{L_j\}_{j=1}^{m}$ be a set of all layers, $R = \{r_k\}_{k=1}^{u}$ be a set of all resources.

Definition 1. Let S be a random finite set. The family of the sets $T = \{T_s : s \in S\}$ is called **S-set**. The S-set $T = \{T_s : s \in S\}$ is finite, if $|T_s| < \infty$ for each $s \in S$ and it is with one element, if $|T_s| = 1$, for each $s \in S$.

Let $\{S_i\}_{i=1}^{n}$ be the S-sorted sets, corresponding to the possible values of the resources. For example S_i ($i = \overline{1,n}$) can be the set of real numbers, an interval $\{[a,b] : a,b \in R\}$, a set (with full order) $\sum^*$ from all words above random alphabet $\sum$, or lexicograpf interval $\{[c,d] : c,d \in \sum^*\}$.

Definition 2. ***Model of studied subject domain*** (SSD) M_d is oriented graph $G_d = (V_d, E_d)$, where the set V_d consists of nodes, introducing concepts of SSD, and the set E_d consists of arcs, determining the relationships (of type *predecessor_of*) between the concepts.

Definition 3. Let SSD M_d be presented by graph $G_d = (V_d, E_d)$. ***Model of learning course*** M_c in M_d is orientented graph $G_c = (V_c, E_c)$, where: a) $V_c = V_c^{'} \cup B$, the set of nodes $V_c^{'}$ ($V_c^{'} \subseteq V_d$, $B \cap V_c^{'} = \varnothing$) introduces the concepts, studied in M_c, the set B consists of four type nodes – ***and***, ***or***, ***yes*** and ***not***; b) the set of arcs E_c introduces the relationship between the concepts, included in M_c and

$E_c = E_c^{'} \cup E_B$, $E_c^{'} \subseteq E_d$,

$E_B \subseteq \{(v,b) : v \in V_c^{'}, b \in B\} \cup \{(b_1,b_2) : b_1, b_2 \in B\} \cup \{(b,v) : b \in B, v \in V_c^{'}\}$.

Introducing nodes from B allows a model of different strategies for carrying out the education for learning concepts $V_c^{'}$, included in the learning course. Introduced type nodes – *and*, *or*, *yes* and *not* [Somova'2002] give the opportunity to set respectively parallel actions without order of passing over, variety of education, additional supporting materials, etc. during the learning process.

Definition 4. ***Model of the learner*** M_s^t at the moment t of learning on a given course is determined by the triple (V_b, V_t, R_t), where V_b is the initial set of known concepts, V_t is the current set of learned concepts (including V_b) in the process of learning ($V_b \subseteq V_t \subset V_c$), and R_t is the list of couples resource-value, specifying the current characteristics of the learner: $R_t = \{(r_{i_k}, s_{i_k}) \in R \times S\}_{k=1}^{n}, i_k \neq i_p, k \neq p$.

The models of initial and final (target) state of the learner M_s^b and M_s^e are determined respectively by the triples (V_b, V_b, R_b) and (V_b, V_e, R_e), where V_e is the final (target) set of concepts included in the course, and R_b и R_e – the initial and final set of characteristics of the learner.

Learning materials

The ***learning materials*** can be either ***learning units*** (text, multimedia, etc.) or ***verification units*** (tasks or tests for verification or self-assessment). Every learning material regards one or several concepts. The full representation and verification of a concept in the common case can be made in some learning and verification units.

Definition 5. ***Model of information unit*** M_l in SSD M_d is determined by the couple $\left(N_l, H_l\right)$, where $N_l \in N$ is the name of information unit, and H_l is a finite set of triples $\left(C, L_C, f_C\right)$, where $C \in V_d$ is a concept from the SSD M_d, $L_C \in L$ is a layer for the concept C, and $f_C : C \times L_C \rightarrow S_1^{\varepsilon} \times S_2^{\varepsilon} \times \ldots \times S_n^{\varepsilon}$ is a resource function. Here $S_i^{\varepsilon} = S_i \cup \varepsilon$, $i = \overline{1, n}$ and ε is used in cases when the value of the respective resource is not determined.

Note: In the realization of the model, each information unit M_l is determined by the triple $\left(N_l, H_l, Type(N_l)\right)$, where $Type(N_l)$ is the type of the physical file containing presentation of M_l. The interpretation of the information unit M_l during the learning process is different according to the file type (for example – a visualization of the file on the display in proper way). For each information unit the characteristics layer and list of the couples resource-value are determined.

Definishon 6. ***Model of verification unit*** **(test assignment)** M_v is determined by the order sextuple $\left(S, K, T, p, q, H_v\right)$, where K is a random with one element S-set (named c*ontext* or S-frame of the assignment); $T = \left\{T_s : s \in S\right\}$ is S-sorted set (presenting sorted sets of the particular slots of the S-frame); p – unary predicate (determining *potential answers* or limitations in filling in the slots at the same time); q – binary predicate (determining *decisions* of the assignment), and H_v is a finite set of couples $\left(C, f_C\right)$, where $C \in V_d$ is concept of SSD M_d and $f_C : C \rightarrow S_1^{\varepsilon} \times S_2^{\varepsilon} \times \ldots \times S_n^{\varepsilon}$ is a resource function.

Definition 7. ***Learning material*** is an information unit or a verification unit.

Planning of education

Definition 8. ***The Plan of education*** P on a given learning course $M_c = \left(V_c, E_c\right)$ is oriented graph $G_p = (V_p, E_p)$, where $V_p = \left[v_1, \ldots, v_k\right]$ is a finite set of lists of learning materials, and $E_p \subseteq E_c$. For each $i = \overline{1, k}$, if learning material $m \in v_i$, then: a) if m is an information unit (n, h), then the concept $c \in V_c$, the layer $l_c \in L$ and the resource function f exist, for which $(c, l_c, f) \in h$; b) if m is a verification unit (s, k, t, p, q, h), then the concept $c \in V_c$ and the resource function f exist, for which $(c, f) \in h$.

Definition 9. ***The Model of education*** on a given course M_c in SSD M_d is the sextuple $\left(R, S, L, \mu_l, \mu_v, P\right)$, where R is the set of all resources, S – the set of resource types, L – the set of all layers, μ_l – the set of information units, μ_v – the set of verification units, P – the plan of education.

The plan of education of a particular learner is determined individually from M_s^b and M_s^e.

Actions, which are caused by a learning and a verification unit, are respectively a ***learning*** and a ***verification action***. The ***learning action*** is realized when the material(s) is(are) offered to the learner and it is supposed that the knowledge of the learner is changed (increased). The ***verification action*** is realized when the material(s) are offered to the learner and they interact with the learner in order to evaluate learner's knowledge and skills.

After forming (modeling) the content of the learning course by pointing out the concepts and relationships from the SSD, which students have to learn, a respective learning course is automatically generated in the DB.

In the ***algorithm for course generation*** (Algorithm 1) the following procedures are used:

procedure Create_Course (**var** Course; Author; Subject_Domain; Course_Node; Course_Link) – creates a course of the given author in the given SSD with the given nodes and relationships between them;

procedure Domain_Concepts (Subject_Domain; **var** Domain_Concept[]) – finds and shows the concepts in the given SSD;

procedure Select_Concepts (Domain_Concept[]; **var** Selected_Concept[]) – chooses the concepts from the given SSD;
procedure Course_Nodes (**var** Course_Node[]) – finds and shows the nodes of the given course;
procedure Course_Links (**var** Course_Link[]) – finds and shows the relationships between the nodes of the griven course;
procedure Select_Nodes (Course_Node[]; **var** Selected_Node[]) – chooses the node/s of the course;
procedure Select_Links (Course_Link[]; **var** Selected_Link[]) – chooses the relationship/s between the course nodes;
procedure Add_Concept (Selected_ Concept[]) – adds concept/s in the course;
procedure Special_Node (Selected_Node[]; Node_Type) – links the selected nodes through special node of type Node_Type;
procedure Link (Selected_Node; Selected_Node[]) – links the node with its predecessor nodes;
procedure View_Node (Selected_Node) – shows the type of the node;
procedure Del_Node (Selected_Node[]) – deletes the node/s from the course, and
procedure Del_Link (Selected_Link[]) – delete relationship/s between two nodes from the course;

```
procedure Author_Course(Course, Author);
begin
  Domain_Concepts (Subject_Domain, Domain_Concept[]);
  Course_Node[]=empty; Course_Link[]=empty;
  Create_Course (Course, Author, Course_Node, Course_Link);
  repeat
    Course_Nodes (Course_Node[]);
    Course_Links (Course_Link[]);
    case Proc of
      View_Node: begin
        Select_Nodes (Course_Node, Selected_Node);
        View_Node (Selected_Node);
      end;
      Add_Node: begin
        Select_Concepts (Domain_Concept[], Selected_Concept[]);
        Add_Concept (Selected_Concept[]);
      end;
      Add_Special_Node: begin
        Select_Nodes (Course_Node, Selected_Node[]);
        Special_Node (Selected_Node[], Node_Type);
      end;
      Del_Node: begin
        Select_Nodes (Course_Node, Selected_Node[]);
        Del_Node (Selected_Node[]);
      end;
      Add_Link: begin
        Select_Links (Course_Link[], Selected_Link[])
        Link (Selected_Node, Selected_Node[]);
      end;
      Del_Link: begin
        Select_Links (Course_Link[], Selected_Link[])
        Del_Link (Selected_Link[]);
      end;
    end;
  until end;
end;
```

❖ **Algorithm 1. Course generation**

The plan of education is a sequence of learning and verification actions. The plan can be ***statical*** (it is represented with one learning material, containing learning information, responding to the initial state of the particular group of learners) and ***dynamical*** (for learners with different knowledge and skills, which are updated on the base of graph of learning materials and history of education). The education can be carried out ***statically*** (through using the static and conventional plan), ***in certain methodology*** (for example through passing over a given plan in depth, in width or in some other rules) and ***freely*** (through dynamic or author's plan).

The conventional plan of education is built only of one learning material, which responds to the requirements of the exact particular group of learners. This plan corresponds to the conventional learning courses and does not change according to the progress of the learner. In the common case, the material is in the form of hypermedia document and except following the content of the material, it can carry out navigation through hiperlinks realized in the document. In this case, it is not necessary to determine the relationships with other information units.

The static plan of education is generated on the base of other type plans of education. The plan is presented as a graph of learning materials (eventually with an indication of recommended relationships between them). However, the education is realized through following the learner's will, not following the indicated relationships between the learning materials.

Each author can propose his/her own plan of education (***author's plan***) by his/her own learning materials, i.e. a recommended path for learning the materials. The student is free to choose the order of learning (from actions unified with *and*) and to miss some actions in the presence of or-nodes. The author's plan is developed without using a graph of concepts. This path is accepted to be the best and is used by the dynamic plan for giving some recommendations about the choice between several nodes from the graph.

When choosing which learning materials to be included in the author's plan, resource and layer restrictions are not used. The process of automatic generation (procedures and algorithm) is analogical to the generation of the learning course with one difference – instead of building a graph of concepts a the graph of learning materials is built, and the initial set of concepts from SSD Domain_Concept[] is changed by the set of learning materials from a given author Author_Material[].

In order to generate the ***dynamic plan of education*** on the base of the graph of concepts, it is necessary to determine the initial and final state in the process of learning for each learner. During the generation of the plan about a given learner on the learning course, the teacher chooses the author (authors) from whom the learning materials should be found, and which should respond to the given characteristics (which resources, with what value and which layers). For each concept from the graph of the course suitable learning materials in the DB are searched. The suitable materials are shown, sorted by the number of characteristics, which have been satisfied; the number of the additional concepts (in and out of the thematic of the course), that are presented; etc. From the found learning materials (information and verification units), some materials are selected and ordered in a linear structure, which replaces the node-concept. The special nodes and relationships between all course nodes remain unchanged in the graph of the plan.

The found learning materials, explaining/verifying extra concepts (besides the searched concepts) which are not in the learning course, are recommended for additional learning/solving. It is recommended to search for a material in conventional bearers or a help from the teacher for the concepts, about which materials are not found in the DB. During the educational process some learning materials are proposed only for concepts, for which predecessor concepts are passed over with their respective materials. This way of generation can bring to a large number of learning materials, linked with one concept, if the resource and layer restrictions of the course are not defined well (for example, if in the DB, there is a lot of authors and materials for using in the particular case).

When selecting among the materials, created from different authors, in order to generate the plan of education, it is possible to assign coefficients for recalculation of authentic author valuations (levels). Therefore, the learning materials representing layers for one concept, created from different authors can be combined.

On the base of the author's plan of education one of the materials is always recommended as the best-fitting one [Somova, 2002]. However, the learner himself/ herself has the opportunity to choose the next material.

In the ***algorithm for generation of dynamic plan*** (Algorithm 2) the next procedures are used:

> **procedure** Initial_Concepts (Course; **var** Initial_Concept[]) – determines the initial concepts (which do not have predecessor concepts);

procedure Learn_Materials (Concept; Layer_Restrictions; Resource_Restrictions; **var** Learn_Material[]) – determines the information units for a given concept, which responds to the layer and resource restrictions;

procedure Assignments (Concept; Resource_Restrictions; **var** Assignment[]) – determines the assignments for a given concept, which responds to the layer and resource restrictions;

procedure Next_Concepts (Concept; **var** Next_Concept[]) – finds the next nonpassed concepts, which are direct predecessors of the given concept;

function count (Array): Number – returns the number of the array elements;

procedure Recursion (Concept_Array[], Prev_Material) – recursively passes over the course in order to generate the plan, finds the next level concepts and their materials and assignments, and makes the relationship between materials/assignments;

procedure Choose_Order (Material[]; **var** First_Material; **var** Last_Material) – the necessary materials and assignments from the array are chosen, they are ordered, then linked in the proper order and the first and last material/assignment are returned, and

procedure Link (Prev_Material, First_Material) – links one node with another with the relationship of type *predecessor_of*.

```
procedure Generation_Dynamic (Course; Layer_Restrictions; Resource_Restrictions);
begin
  Initial_Concepts (Course, Initial_Concept[]);
  Recursion (Initial_Concept [], “”);
end;
procedure Recursion (Concept_Array[], Prev_Material);
begin
  for i:=1 to count (Concept_Array) do
  Begin
    if (Concept_Array [i]<>”and”) and (Concept_Array [i]<>”or”) and (Concept_Array [i]<>”not”) and (Concept_Array [i]<>”yes”) then
    begin
      Learn_Materials (Concept_Array [i], Layer_Restrictions, Resource_Restrictions, Material[]);
      Assignments (Concept_Array [i], Resource_Restrictions; Material[]);
      if count (Material)=0 then
      begin
        write “Find materials/assignments on conventional bearers for the concept”, Concept_Array [i]);
        First_Material:= “”;
        Last_Material:= “”;
      end
      else
        Choose_Order (Material[], First_Material, Last_Material);
    end
    else
    begin
      First_Material:= Concept_Array [i];
      Last_Material:= Concept_Array [i];
    end;
    Link (Prev_Material, First_Material);
    Next_Concepts (Concept_Array [i], Next_ Concept[]);
    if count (Next_ Concept[])=0 then exit
    else
      Recursion (Next_Concept[], Last_Material);
  end;
end;
```

❖ **Algorithm 2. Generation of the dynamic plan**

Carrying out the education

The process of education consists of proposition of learning information units (for introduction) and verification units (for solving and testing) according to the respective plan. Because of the simplicity of the conventional plan (one unit, in which there can be hyperlinks) and the statical plan (characterized with free navigation among the materials without accounting the relationships among them), the algorithms for carrying out the education are not presented.

The algorithm for carrying out the education through author's and dynamic plan (Algorithm 3) oblige the learners to respect given relationships among the learning materials. In the algorithm the following procedures are used:

procedure Initial_Nodes (Plan; **var** Initial_Node[]) – determines the initial nodes (these which do not have predecessor nodes);
procedure Recursion (**var** Pass_Node[]; **var** Potential_Node[]) – recursively passes over the education plan, gives and finds the passed and the next potential nodes;
function Show_Material (Material): Success – gives the access (as hyperlink) to the material and returns as a result: 0 – unsuccessfully passed node, 1 – successfully passed node, 3 – it is not known how the node has been passed);
procedure Next_Nodes (**var** Next_ Node[]); – finds the current potential nonpassed nodes;
procedure Show_Nodes (Potential_Node[]); – shows all the potential nodes;
procedure Select (Array_Node[]; **var** Node); – the user selects one of the nodes;
procedure Branch_Pass (Nodes[], Type); – the branches, beginning with nodes of the pointed type are marked as passed;
procedure Find_Successors (Node; **var** Successor[]); – finds the successors of the pointed node, and
procedure Choose_Node (Nodes[]; **var** Node); – selects one of the nodes.

```
procedure Dynamic_Learning (Plan);
begin
  Initial_Nodes (Plan, Initial_Node[]);
  Recursion ([],Initial_Node[]);
end;
procedure Recursion (Pass_Material[], Potential_Material[]);
begin
  write ("Choose one of learning materials");
  Show_Nodes(Potential_Material[]);
  Select (Potential_Material[], Material);
  if Material="and" or Material [i]="or" or Material [i]="not" or  Material [i]="yes" then
    begin
      if Material="or" then
        begin
          write ("Choose one of materials: ");
          Pass_Material[count(Pass_Material)+1]:= Material;
          Find_Successors (Material, Successor[]);
          Choose_Material (Successor[], Material);
          Success := Show_Material (Material);
          Branch_Pass (Successor[]-Material,"material");
          Branch_Pass (Successor[]-Material,"and");
          Branch_Pass (Successor[]-Material,"or");
          Branch_Pass (Successor[]-Material,"yes");
          Branch_Pass (Successor[]-Material,"not");
        end;
    end
```

```
    else
      Success := Show_Material (Material);
    Pass_Material[count(Pass_Material)+1]:=Material;
    Next_Nodes (Next_ Material[]);
    Find_Successors (Material, Successor[]);
    if Success=3 then
      begin
        Branch_Pass (Successor[],"yes");
        Branch_Pass (Successor[],"not");
      end
    else
      begin
        Branch_Pass (Successor[],"or");
        Branch_Pass (Successor[],"and");
        Branch_Pass (Successor[],"material");
        if Success=0 then Branch_Pass (Successor[],"yes");
        else if Success=1 then Branch_Pass (Successor[],"not");
      end;
    Recursion (Pass_Material[],Next_Material[]);
  end;
```

❖ **Algorithm 3. Education through author's and dynamic plan**

Conclusion

The proposed model for planning of education based on resources and layers has been included in the system PeU (http://peu.pu.acad.bg). The experiments with the system confirm that the model gives to the learner the opportunity – the generated plan for the course to vary dynamically on the base of learner's progress passing over each learning phase and to correspond to the learner's initial knowledge and way of learning.

Bibliography

[Зайцева, 1989] Л. Зайцева, Л. Новицкий, В. Грибкова, Разработка и применение автоматизированнњх обучающих систем на базе ЕВМ, Зинатне, Рига, 1989.

[Пасхин, 1985] Е. Пасхин, А. Митин, Автоматизированная система обучения ЕКСТЕРН, Издательство Московского университета, 1985.

[Савельев, 1986] А. Савельев, В. Новиков, Ю. Лобанов, Подготовка информации для автоматизированнњх обучающих систем, Вьсшая школа, Москва, 1986.

[Britain] S. Britain, O. Liber, A Framework for Pedagogical Evaluation of Virtual Learning Environments, http://www.jtap.ac.uk/reports/html/jtap-041.html

[Chien] S. Chien et al., The multimission VICAR planner: A knowledge-based system for automated image processing, NASA Science Information Systems, Issue 38.

[Grandbastien, 1994] M. Grandbastien, E. Gavignet, ECSA: An Environment to Design and Instantiate Learning Material (Handbook of Design and Production of Multimedia and Simulation-based Learning Material; de Jong T. and L. Sarti; eds.), Kluwer Academic, Publishers, Dordrecht, The Netherlands, 1994.

[Grant, 1997] T. J. Grant, M. Verhoef, L. P. Gale, AI planning for just-in-time training in space: ESA's integrated learning system, in Proceedings of 1st NASA Workshop on Planning and Scheduling for Space, Oxnard, CA, Oct 1997.

[Koehler, 1998] J. Koehler, 1998. Planning under resource constraints, in Proceeding of ECAI-98, 489-493, 1998.

[Koffman, 1975] E. B. Koffman, J. M. Perry, Description and evaluation of a model for generative CAI, Computers in education: Proc. II-d Conf. IFIP, Amsterdam, 1975.

[Milani, 2000] Alfredo Milani, Stefano Marcugini, Marco Baioletti, DPPLAN: An algorithm for fast solutions extraction from a planning graph, Proceedings of AIPS2000, 5th Int. Conf. on AI Planning and Scheduling, Chien & Kambhampati eds, AAAI Press, Menlo Park, USA, 2000.

[Milani, 2001a] Alfredo Milani, Valentina Poggioni, Automated planning of educational goals, Proceedings of the ICT&P 2001 Conference, Sofia, 49-60 p., 2001.

[Milani, 2001b] Alfredo Milani, Valentina Poggioni, RDDPlan: a system for planning with resources, Tech. Rep. 12-2001, Department of matematics and computer science, University of Perugia, Italy, 2001.

[Rintanen, 1999] J. Rintanen, H. Jungholt, 1999. Numeric state variables in constraint-based planning, in Proceedings of the 5th European Conference on Planning, ECP99, Durham, UK, 1999.
[Somova, 2002] E. Somova, G. Totkov, Modelling of Education in the Environments of Type "Virtual University", Conf. EDEN: Open and Distance Learning in Europe and Beyond Rethinking International Co-operation, Granada, Spain, 16-19 Jun, 2002, p. 275-280.

Author information

George Totkov - The University of Plovdiv, Dept. of Computer Science, 24 Tzar Assen st., 4000 Plovdiv, Bulgaria; e-mail: totkov@pu.acad.bg
Elena Somova - The University of Plovdiv, Dept. of Computer Science, 24 Tzar Assen st., 4000 Plovdiv, Bulgaria; e-mail: eledel@pu.acad.bg

PLANNING OF INTELLECTUAL ROBOT ACTIONS IN REAL TIME

N. Romanenko

Summary: *In article the mathematical model of the mobile robot actions planning at recognition of situations in extreme conditions of functioning is offered. The purpose of work is reduced to formation of a concrete plan of the robot actions by extrapolation of a situation and its concrete definition with the account a priori unpredictable features of current conditions.*
Key words*: the mobile robot, recognition of a situation.*

Introduction

Creation of the intellectual mobile robots, capable to adapt and plan the actions in conditions of aprioristic uncertainty of dynamically changing habitat, is one of the important strategic problems of modern techniques. Absence of preliminary environment formalization, and also presence of any way moving obstacles and purposes in it complicates the use of automatic control traditional methods. The given circumstance stimulates development of new control systems with presence on mobile robots (MR) board of situation recognition system on the basis of the multiprocessing computer with elements of artificial intelligence that provides adaptability of MR behavior in an environment.

Recognition of situations is the new area of cybernetics. The closest area is images recognition. But there is a basic distinction of these concepts: the image is "static", and the situation is dynamical, recognition of situations is always connected to a prediction (extrapolation) that usually does not happen in the theory of images recognition. At situation recognition there is no aprioristic classification as the number of possible situations is unlimited, but results are classified and have the final alphabet.

For MR control system we shall understand that the situation is a set of events, developing in time and space limited in radius of its action, and having the important consequences from the point of view of the chosen criterion function. The situation includes three basic components:

- the ground conditions fixed during the certain moment of time (presence of obstacles in a way);
- processes which can occur both with its condition, and with MR condition (dynamism of obstacles and the robot);
- result or possible consequences (planning of actions and forecasting).

To distinguish a situation by control system - means to develop decision about result of further MR movement on the basis of environment and proceeding process information.

Traditional mathematical models of MR management in extreme conditions of functioning becomes insufficiently as MR proper response to change of situations is not described, especially at occurrence of obstacles. The given problem was examined by many researchers, and there exist various ways of its decision [1-3].

Planning of actions is the major function of the mobile robots independently working in dynamic and uncertain environments. Scripts [4] are effective model of knowledge representation in such systems. Scripts represent the generalized description of sequence of MR actions in some stereotyped situation, allowing to achieve a required target condition. Formation of a concrete plan of action is carried out by a choice of one of possible situations and its concrete definition with the account a priori unpredictable features of current conditions. At functioning in the uncertain environment, time restrictions on decision-making are a priori unknown; therefore the robot should possess ability to adapt time of decisions to dynamics of processes occurring in the environment. Known models of representation and recognition of situations do not support such opportunity.

Formal representation of actions plan

For representation of robot actions plan we shall use hierarchical frame structure of the following kind. At the top-level plan FPm is set by the frame of a kind:

$$FP_m = (PN_m , S_0 , Act_1 , S_1 , Act_2 , S_2 ,..., S_{n-1} , Act_n , S_n),$$

where PN_m - a name of *m* plan; S_0 - the current condition of environment; Act_i - the frame of i action of the plan; S_i – the frame of environment condition after performance *of i* action of a plan; S_n - the target condition of environment. Frames of environment conditions and actions of the robot contain slots «Type of the frame», «Name of the frame» and set of frames of the bottom level serving for representation of parameters of conditions and actions.

Parameters of environment conditions can be divided on the following groups:

- external world condition parameters, which value do not depend on robot actions;
- environment condition parameters and the robot in the environment, changeable as a result of robot actions;
- robot inwardness parameters, describing internal robot resources.

Let MR, moving in a priori unknown to it environment, to find out on the way an obstacle and as one of possible variants of actions to consider a detour of an obstacle on the right. In this case it is possible to allocate two stages of obstacle overcoming: turn to the right and alignment of movement. Accordingly plan of MR action «Obstacle overcoming» contains two frames of action with names "To the right" and "Directly" and three frames of condition: «Before an obstacle», "Alignment" and «Behind an obstacle». It is obvious, that success of realization of the given plan is defined by ability of the robot to make a detour of an obstacle and entrance on a target trajectory. The successful detour depends on MR maneuverability and speed. In its turn speed depends on capacity of engines and density of a ground in a detour place, i.e. is defined as internal robot opportunities (ability to provide required speed at turn), and external conditions (character of a ground on a trajectory).

Statement of decision-making task in real time on the basis of actions planning

Under generalized problem situation (GPS) we shall understand the generalized description of environment condition in which the robot is required to make some decision. GPS is identified by the name; for example, GPS «Obstacle overcoming» corresponds to an above-mentioned example.

Generally in some problem situation the robot has not unique variant of possible actions and, accordingly, a plan of action subject to the analysis. So for an above-mentioned example the robot can, along with the script «Detour on the right», to consider also the script «Detour on the left». Decision-making with use of actions planning in real time assumes definition of the most effective in the given situation way of actions for limited time. Thus the stock of time T_d for decision-making should be defined dynamically, proceeding from the analysis of the current situation.

Proceeding from the aforesaid, process of decision-making by the intellectual robot in system of real time on the basis of actions planning includes the following steps:

1) preliminary estimation of a situation and definition of the general stock of time T_d for decision-making;
2) definition of actions variants set possible in the given situation;
3) distribution of the general budget of time for tasks of various actions variants estimation;
4) concretization and estimation of actions planning efficiency;
5) actions variants comparison and choice of the best of them.

Let's consider all listed steps of decision-making.

1. Definition of the general stock of time.

Time restrictions for decision-making by the robot are caused by possible approach of events undesirable to the robot if it in due time will not undertake corresponding actions. Time of approach of events is determined by dynamics of environment processes (in particular, actions of mobile obstacles) and a priori is not known. Thus, the robot should possess ability to define dynamically the time of critical event approach on the basis of forecasting of possible consequences of the current situation with use of knowledge of laws of various environment processes.

Set of possible in the future critical events $\{CE_i\}$ can be put in conformity with every GPS. For example, in a situation with obstacle overcoming (the detour on the right), critical event for MR moving is presence of a wall or a hole to the right as an obstacle. The robot stops before the obstacle if movement is impossible owing to the appeared dynamic obstacle. Function $f_d^i(P_j, ..., P_k)$ from predicates of the current situation, calculating stock of time T_d caused by possible approach of the given critical event can be put in conformity to each critical event.

The kind of this function is known at a stage of MR control system construction, and its description is stored in corresponding slot of action plan. The valid value of a stock of time T_d is calculated dynamically on the basis of the current values of situation parameters.

2. Definition of a set of possible variants of actions.

Every GPS at a stage of construction of base of MR knowledge puts in conformity set of tactical variants of actions submitted by plans of action $\{FP1, ..., FPq\}$. Each such plan has slot "Precondition" in which predicate *PC (precondition)* is written down, determining additional conditions of the given plan applicability. The predicate is determined on parameters of the current situation (both external, and inwardnesses of the robot) and allows to exclude some plans from the further consideration. For example, the detour of an obstacle at the left can be impossible because of a reservoir, taking place there, movement can be stopped owing to a steep slope of a line, etc.

Calculation of the given predicate is realized by function $g(P_j,...,P_k)$. Computing complexity of this function should be low and the top estimation of its calculation time should be known. Calculation of predicate *PC* can demand gathering of the additional information for reception of facts, which are not contained at present in a database. As a result of definition of the validity of preconditions of all plans contained in $\{FP_1,...,FP_q\}$ the reduced set $\{FP_m\}$ of possible plans for the further analysis is formed.

3. Distribution of the general stock of time between tasks of action variants estimation.

As the estimation of efficiency of all tactical plans from set $\{FP_m\}$, should be executed in time T_d , it is necessary to allocate the general stock of time between corresponding tasks of a concretization.

Various plans of actions, generally, demand various time of a concretization and this time depends on parameters of the current situation and it is not known at a stage of knowledge base construction. Besides time of action plan concretization can vary over a wide range not only depending on an external situation, but also on internal resources of the robot. For example, the concretization of the script «Detour of an obstacle» assumes scanning a surface of road for definition of roughnesses and density of a ground with the purpose of definition of an optimum trajectory of movement. It is obvious, that time of this task decision depends on the area of the scanning, the current condition of touch and MR processing resources, and other parameters which values are a priori unpredictable. At the same time, set of such parameters can be allocated beforehand and for each concrete plan FP_m the bottom estimation of decision time of task of concrete definition T_m^* can be expressed as function from these parameters:

$$T_m^* = h_m(P_l ,..., P_k)$$

These functions should have low computing complexity. Using the given functions, MR calculates the bottom estimation of total time of a concrete definition of all plans belonging to set $\{FP_m\}$ on the basis of the current situation:

$$T_\Sigma^* = \sum_m T_m^*$$

Then time allocated for concretization of *i* plan of action, will be determined as follows:

$$T_d^{(i)} = T_d * (T_i^* / T_\Sigma^*)$$

4. Concretization and estimation of actions planning efficiency.

Concretization of action plan Fp_i is reduced to a concretization of all of its steps - frames - actions. This task should be solved for limited time $T_d^{(i)}$ Tasks of search of optimum values of action plan parameters depend on the concrete mathematical models used for the description of corresponding steps of the plan, and can be the diversified. However, not narrowing the generality of consideration it is possible to count, that these tasks are under construction as a task of undefined time [5].

Thus the set $\{v_i^k\} = \{(T_i^k, Q_i^k)\}$ variants of its decision distinguished by time T_i and quality Q_i of received result should be put in conformity to each single task of step concretization. Use of a variant of undefined time creates a basis for formation of task decision for allocated time due to reduction in quality of the decision.

For MR working in the uncertain environment, variants of individual concretization tasks decision distinguished by time and quality of the decision should be formed dynamically in view of the current parameters of external conditions and internal resources of the robot. With this purpose in everyone frame - action of the plan slot is added, containing function $u_s(P_l ,..., P_k)$, forming set $\{v_i^k\} = \{(T_i^k, Q_i^k)\}$ for a concrete situation (index k - number of variant of tasks decision). Function $u_s(P_l ,..., P_k)$ should have low computing complexity.

The received sets $\{v_i^k\}$ serve as the initial data for distribution of time $T_d^{(i)}$ allocated for concretization of the given plan of actions, between individual tasks of separate steps concretization.

Conclusion

Use of suggested method of MR actions planning in uncertain environment allows to determine dynamically time of approach of critical event on the basis of forecasting of possible consequences of the current situation with use of various environment processes laws knowledge.

Tasks of search of optimum values of action plan parameters depend on the concrete mathematical models used for the description of corresponding steps of the plan, and can be the diversified.

At the same time, MR control systems with recognition of situations and planning of actions are characterized by great volume of the processed information and high complexity of used algorithms of processing of the information and decision-making. High reliability demands are also made for them. The specified characteristics can be achieved due to use of multiprocessing computing systems, for example as an artificial neural network. In hardware realization neural network is a network from set of simple processors, each of which has small local memory and communication connections with other processors.

Prototypes of such networks for MR control systems have been already in use now for forecasting of situations in financial sphere, images recognition, speech.

Literature

1. Popov E.V., Firdman G.R. Algorithmic Basics of Intellectual Robots and Artificial Intelligence. – M.: Nauka, 1976.
2. Homogeneous Managing Structures of Adaptive Robots / Kaljaev A.V., Chernuhin Yu.V., Noskov V.N., Kaljaev I.A. under A.V.Kaljaeva and J.V.Chernuhina edition.- Moscow: Nauka, 1990
3. Stenz A., Map-Based strategies for Robot Navigation in Unknown Environments/in proc. AAAI 96 Planning with incomplete information for Robot Problems
4. Artificial Intelligence. - In 3 books. B. 2. Models and Methods: Directory / Under D.A.Pospelov edition – M.: Radio I Svyaz, 1990.
5. Zilberstein S. Using anytime algorithms in intelligent systems, AI Magazine, 1996, v. 17, N 3, p.73-83.

Authors information:

Romanenko Nadezhda - Institute of Artificial Intelligence, B.Hmelnitsky avenue, 84, Donetsk - 83050, Ukraine e-mail: rni@iai.donetsk.ua

ON OBDD TRANSFORMATIONS REPRESENTING FINITE STATE AUTOMATA

S.Kryvyy, W.Grzywacz

Abstract: *We present OBDD transformation problem representing finite labeled transition systems corresponding to some congruence relation. Transformations are oriented toward obtaining the OBDD of a minimized transition system for this congruence relation.*

Keywords: *finite automata, OBDD, congruence relation, minimization, transformation.*

Introduction

Ordered Binary Decision Diagrams (OBDDs) are widely used in many domains of computer science. One of the most important applications of OBDDs is compact representation of finite transition systems (TSs), which allows one to extend verification possibilities of TSs (the Model Checking method). An important class of TSs is the class of finite TSs, which is the class of finite state automata in the case where the numbers of states and transitions of a TS is finite.

In this paper, we consider the following problem. Let a finite state automaton A, its representation as an OBDD, and a congruence relation R on the set of states of the automaton A be given. It is necessary to construct an OBDD that represents quotient automaton A/R. This problem can be solved in a traditional way as follows: first, to construct the quotient automaton A/R and, second, the OBDD that represents this quotient automaton. However, the relation R often can be formulated after the representation of the automaton A as an OBDD and, as a result, this OBDD can be easily transformed into the OBDD that represents the A/R automaton.

Preliminary notes

A Finite state X-automaton is understood to be a quadruple $A = (A, X, f, F)$, where A is a finite set of states, $X = \{0, 1\}$ is the alphabet of input symbols or the input alphabet of the automaton, $f : A \times X \rightarrow A$ is a transition function, and $F \subseteq A$ is a subset of terminal states.

Let $R \subseteq A \times A$ be some congruence relation with respect to the transition function, i.e., we have $a R b \Leftrightarrow (\forall x \in X) (f(a, x) R f(b, x))$. The automaton whose set of states consists of equivalent classes of relation R is called quotient automaton A and is denoted by A/R. An important congruence relation is the so-called automata equivalence relation R_A that, for X-automata, is defined in the following way:

$$a R_A b \Leftrightarrow (\forall p \in F(X)) (f(a, p) \in F \Leftrightarrow f(b, p) \in F),$$

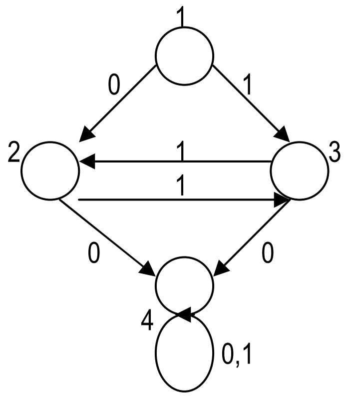
where $F(X)$ is the semigroup of all words of finite length in the alphabet X, $a, b \in A$, and F is the set of final states. An automaton is called **reduced** or **minimal** if all its states are pairwise nonequivalent. As is well known, the quotient automaton A/R_A is minimal in the class of all the automata that are equivalent to automaton A [4].

An OBDD is a graphical representation (an acyclic graph) of a Boolean function [1]. The use of OBDDs for representation of finite state automata is based on the representation its transition function f in the form of a ternary relation R_f defined on codes of states of an automaton. Let, for example, for states $a, b \in A$ and some $x \in X$, we have $f(a, x) = b$. If $\underline{x}$, $\underline{a}$, and $\underline{b}$ are the corresponding codes of a symbol $x \in X$ and states $a, b \in A$, then we have

$$R_f(\underline{x}, \underline{a}, \underline{b}) = \begin{cases} 1, & if \quad f(a,x) = b, \\ 0, & if \quad f(a,x) \neq b, \\ d, & in \quad other \quad cases \end{cases}$$

Now, it is obvious that this form of the relation R_f corresponds to a Boolean function $g_f(\underline{x}, \underline{a}, \underline{b})$ that equals 1 if and only if $R_f(\underline{x}, \underline{a}, \underline{b}) = 1$.

Consider an example. Let A = (A= {1,2,3,4}, X = {0,1}, f, F = {4}) be a finite state X-automaton whose transition function *f* is specified by the transition graph given below.



We assume that the symbols of the input alphabet *X* are coded in a natural way by an identical mapping and that the states of the automaton are coded as follows:

$$1 \rightarrow 00,\ 2 \rightarrow 01,\ 3 \rightarrow 10,\ 4 \rightarrow 11.$$

Then the relation $R_f(x, a_1, a_2, b_1, b_2)$ assumes the form:

$$R_f(0,0,0,0,1) = 1,\ R_f(1,0,0,1,0) = 1,\ R_f(0,0,1,1,1) = 1,\ R_f(1,0,1,1,0) = 1,$$
$$R_f(0,1,0,1,1) = 1,\ R_f(1,1,0,0,1) = 1,\ R_f(0,1,1,1,1) = 1,\ R_f(1,1,1,1,1) = 1.$$

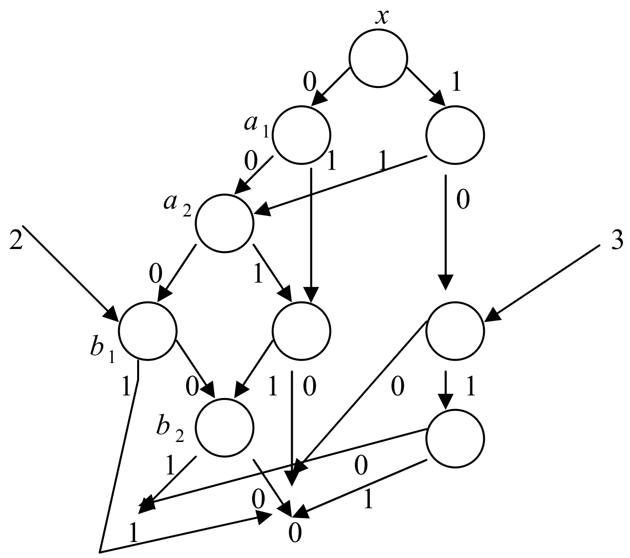
The above relation corresponds to the following Boolean function g_f:

$$g_f(x, a_1, a_2, b_1, b_2) = \bar{x}\bar{a}_1\bar{a}_2\bar{b}_1 b_2 \vee x\bar{a}_1\bar{a}_2 b_1\bar{b}_2 \vee \bar{x}\bar{a}_1 a_2 b_1 b_2 \vee x\bar{a}_1 a_2 b_1\bar{b}_2 \vee$$
$$\vee \bar{x} a_1\bar{a}_2 b_1 b_2 \vee x a_1\bar{a}_2\bar{b}_1 b_2 \vee \bar{x} a_1 a_2 b_1 b_2 \vee x a_1 a_2 b_1 b_2,$$

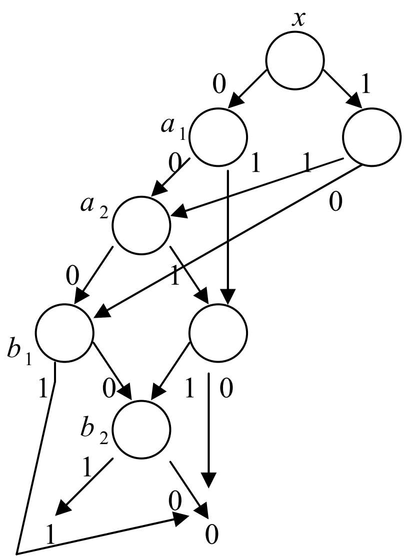
where x, a_1, a_2, b_1, and b_2 are codes that represent input symbols from the alphabet X and a and b are states from A, (we write a_i if $a_i = 1$ and $\bar{a}_i$ if $a_i = 0$, the same is true for b_i, i = 1,2).

Transformation of OBDD representing of non minimal automaton

We now describe transformations of OBDD that lead to the representation of the minimal A/R_A automaton. Let $K_1, K_2, \ldots, K_m$ be the equivalence classes of the set A in the sense of relation R_A. We can choose one representative element in every class K_i, i = 1,..., m. Let it be states $b_1, b_2, \ldots, b_m$, where $b_i \in K_i$. These states can be represented using the following codes: $b_{11} b_{12} \ldots b_{1k}, b_{21} b_{22} \ldots b_{2k}, \ldots, b_{m1} b_{m2} \ldots b_{mk}$, where $k = \log |A|$. The first step of the transformation is based on the reorganization of paths in the acyclic diagram representing OBDD. All the paths leading to a state from K_i must be replaced by ones that lead to their representative element. At the next step, we apply standard OBDD operations to remove unused nodes. At the third step, we introduce new nodes since it is necessary to correctly reorient the transitions so that unused nodes are eliminated. We do not have to introduce new nodes if there are transitions that make it possible to reach states from a class K_i, i = 1, …, m (that are equivalent to state b_i). In the minimal automaton, these states will be eliminated. The described method will be explained using an example given below. The figure below presents the initial OBDD graph.

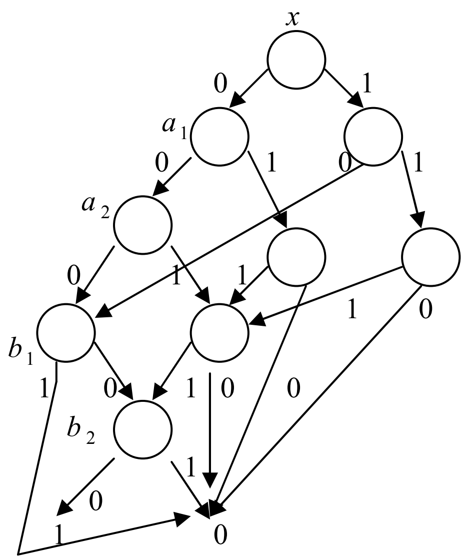


In this automaton, states 2 and 3 (see the above figure) are equivalent. Their codes are 01 and 10, respectively. We choose, for example, state 2 as a representative element in the equivalence class. We perform the first step and reorganize paths that lead to state 2. At the same time, we eliminate unused nodes. As a result, we obtain the OBDD presented below:

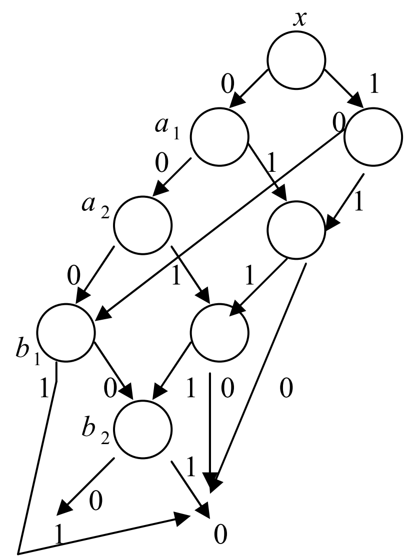


This graph does not correspond to the minimal automaton yet. This is because all the transitions to state 3 are eliminated, but the transitions outgoing from state 3 are not eliminated. To eliminate the latter transitions, it is necessary to introduce a new node (in this case, only one node). The explanation is as follows:

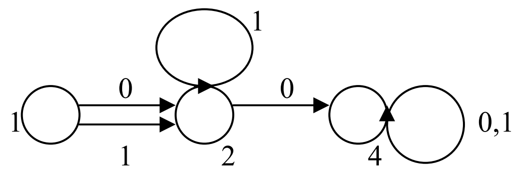
In the OBDD, 2 nodes exist that correspond to the unessential transition $xa_1 \overline{a}_2 \overline{b}_1 b_2$. These nodes are labeled by a_1. For these nodes, we introduce a new node labeled by a symbol a_2 and eliminate unessential transitions. All these transformations are showed below.



After an evident reduction, we obtain the final OBDD that represents the minimal automaton A/ R_A of the automaton A.



To show that this OBDD really represents A/ R_A automaton, we present the following automaton obtained from the above diagram:



It is obvious that the automaton obtained is a reduced automaton of the automaton A.

It follows from the above that the obtaining of the OBDD of the minimal automaton is reduced to the following transformations:

(a) elimination of transitions leading to equivalent states, which reduces to changing the paths leading to this state to lead to node, which is main representative state of equivalence class;

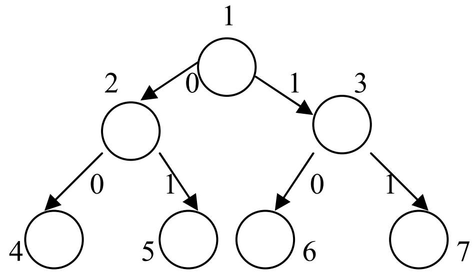
(b) elimination of transitions from equivalent states with eventual reproducing of necessary nodes in OBDD;

(c) elimination of unavailable nodes in OBDD graph.

Acyclic automata

The OBDD transformations described above can be applied to OBDD transformations representing an acyclic automaton. In this case, the relation R need not be a congruence relation, it must be an equivalent relation. To

obtain a reduced automaton, a congruent closure R^* of the relation R should be constructed. In this case, the closure R^* is defined as follows: $a\,R^*\,b \Leftrightarrow (a\,R\,b) \vee (\forall\ x \in X)\ (f(a, x)\ R^*\ f(b, x))$.
If an equivalence relation R is given, then we can obtain a reduced automaton from the OBDD.
In the general case, the relation R^* can lead to a cycle. To assure the acyclicity of R^*, we will consider the case where R relates the states on the same level. This constraint guarantees the acyclicity of relation R^* and quotient automaton A/R^*. All the above facts are illustrated by the following example of acyclic automaton A whose graph is presented below:



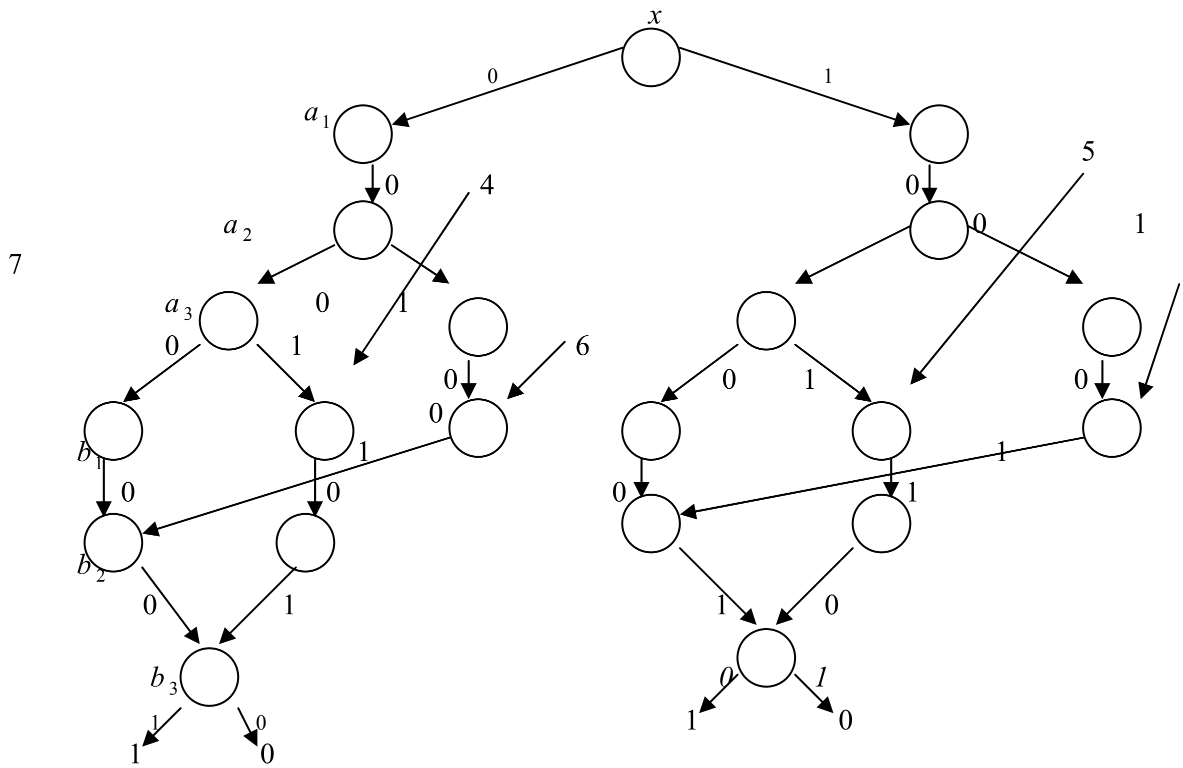
Let the states of this automaton are coded as follows:

$$1 \to 000,\ 2 \to 001,\ 3 \to 010,\ 4 \to 011,\ 5 \to 100,\ 6 \to 101,\ 7 \to 110.$$

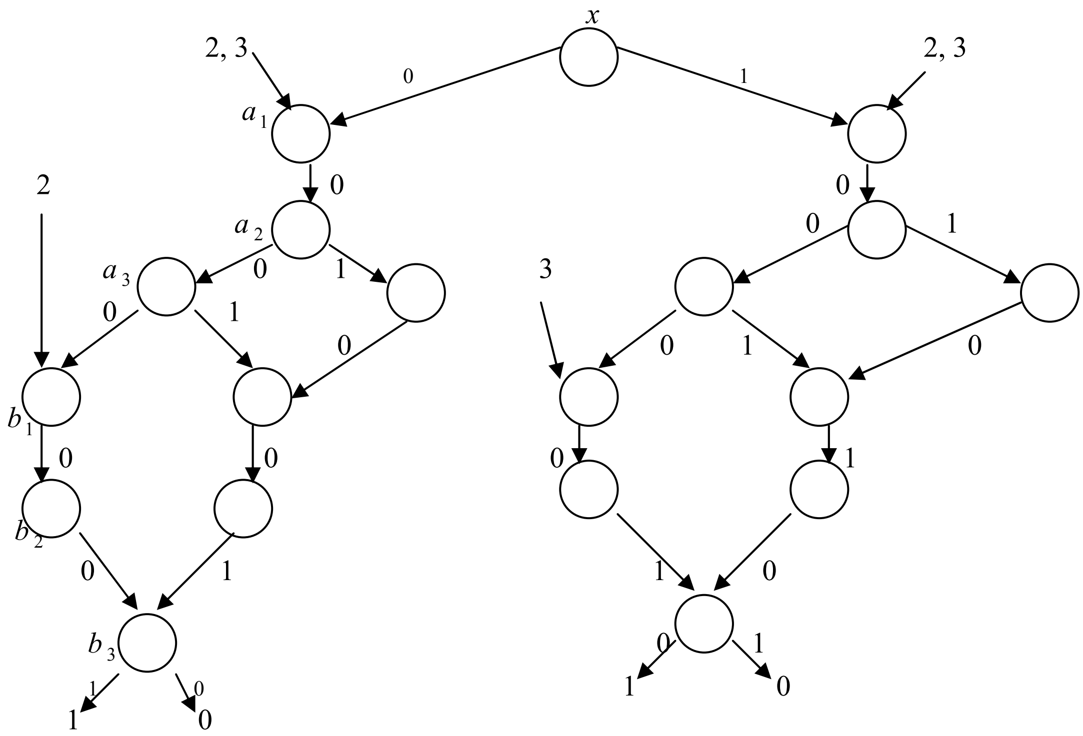
A Boolean function g_f corresponding to the transition relation R_f assumes the form

$$g_f\,(x, a_1, a_2, a_3, b_1, b_2, b_3) = \bar{x}\,\bar{a}_1\,\bar{a}_2\,\bar{a}_3\,\bar{b}_1\,\bar{b}_2\,b_3 \vee x\,\bar{a}_1\,\bar{a}_2\,\bar{a}_3\,\bar{b}_1\,b_2\,\bar{b}_3 \vee$$
$$\vee\ \bar{x}\,\bar{a}_1\,\bar{a}_2\,a_3\,\bar{b}_1\,b_2\,b_3 \vee x\,\bar{a}_1\,\bar{a}_2\,a_3\,b_1\,\bar{b}_2\,\bar{b}_3 \vee \bar{x}\,\bar{a}_1\,a_2\,\bar{a}_3\,b_1\,\bar{b}_2\,b_3 \vee$$
$$x\,\bar{a}_1\,a_2\,\bar{a}_3\,b_1\,b_2\,\bar{b}_3.$$

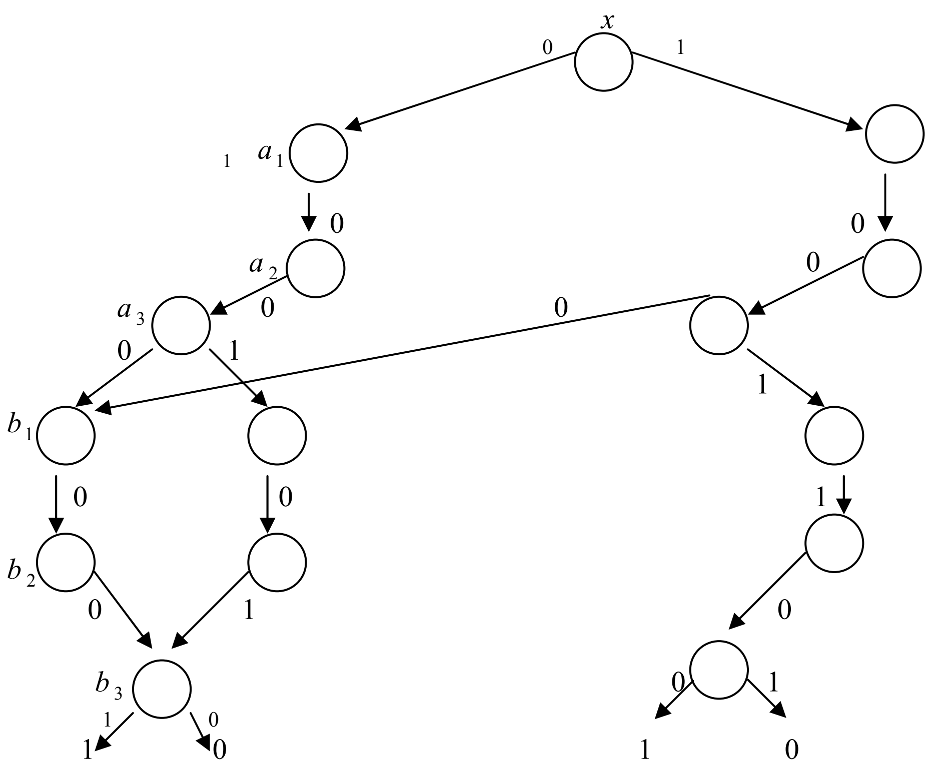
The OBDD that corresponds to this function assumes the form (to reduce the size of the diagram, we assume that all the missing paths lead to the node labeled by 0):



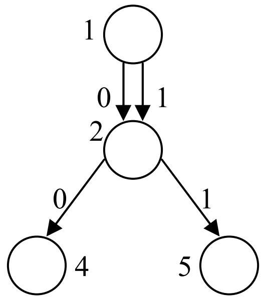
Let the relation R = {(4,6), (5,7)} be given. Then we choose the representatives of the equivalence classes, e.g., states 4 and 5, respectively, and reorient the transitions to these states. Thus, after this step of transformation, we obtain the following diagram:



As a result of this transformation, we obtain that the states 2 (001) and 3 (010) are equivalent. We choose state 2 as a representative of the equivalence class and eliminate the transitions from and to state 3. Repeating the first transformation step, we obtain the following OBDD:



The OBDD presented above corresponds to the following acyclic automaton:



Conclusions

The method presented in the paper allows one to reduce the number of nodes in OBDDs. The use of an equivalence relation instead of a congruence relation allows us to work with acyclic graphs. This makes it possible to simplify the solution of the common subexpressions problem.

This work on OBDDs deals with algorithms of minimization of finite state automata. This research shows the expediency of transformations of OBDDs.

Bibliography

[Bryant R.E., 1] Bryant R.E. Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams. Scool of Computer Science, Carnegi Mellon University, Pittsburg. 1992 (june). - 34 P.

[Clarke E.M., Schlingloff B.-H., 2] Clarke E.M., Schlingloff B.-H. Model Checking. In ed. A. Robinson and A. Voronkov Handbook of Automated Reasoning, Elsevier Science Publishers B.V. - 2001. - P. 1369 - 1522.

[McMillan K.I.,3] McMillan K.I. Symbolic Model Checking: an approach to state explosion problem. PhD thesis. Scool of Computer Science, Carnegi Mellon University. -1992. - 212 P.

[Glushkov V.M., Leticievskij A.A., Godlievskij A.B., 4] Glushkov V.M., Lieticievskij A.A., Godlievskij A.B. Mathematical biology methods. Book 6. Methods of synthesis of discret biological systems. Kiev: University. - 1983. - 264 pages.

Information about authors

Krivoi Sergey – Glushkov Institute of Cybernetics NSA Ukraine, Ukraine, Kiev, 03187, 40 Glushkova Street, Institute of Cybernetics, e-mail: krivoi@i.com.ua

Grzywacz Wioletta – Technical University of Czestochowa, Institute of Theoretical and Aplied Informatics, Czestochowa, Poland, e-mail: wiola@icis.pcz.czest.pl

About: …

INTELLECTUAL COMMUNICATIONS AND CONTEMPORARLY TECHNOLOGIES ALTERNATIVES OF THE SCIENCE LIBRARIES

Varbanova-Dencheva, K.

(Summary of: Varbanova-Dencheva, K. Intellectual communications and contemporarly technologies. Alternatives of the science libraries. Sofia, Marin Drinov academic publishing house. 2003, 114p**.**)

The new technologies and the globalization are the factors which have brought essential changes in human society and its environment. The unceasing dynamic changes imposed new strategies for survival and prosperity of institutions and people in the new conditions. The spheres with greatest potential for achieving competition priority are compatible to the fastness of research results implementation in each field of human activity.

The extended knowledge requires narrower specialization as well as interdisciplinarity to solve the arising problems. The new research fields and trends are a synthesis of science and high technologies determined by the new discoveries.

The present study aims at finding answers to the questions about the place of science library in the dynamic restructuring of research environment. The necessity of transformation of the scientific library's genetically set functions from a guardian of the achieved knowledge to an active participant in the creation of new knowledge is a natural consequence of the processes and tendencies of the social medium. The priorities of Europe and USA for intensive creation of knowledge economics are at the first place and this requires intensification of that research an integral part of which are the new communications realized at a new technological level.

Chapter One outlines the stages of one cycle of the development of research and achieving new knowledge. The evolution of scientific communications in the new information environment is determined by the new technological background of the communication channel. Their purposeful development is shown in the context of the research priority in the last ten years – creation of information society. The role of the scientific library as a mediator and a communication institute is surveyed in four basic projections: research communications, library and information support of research, infiltration of the scientific library in the infrastructure of information society and its transformation into a high technological component of infosphere.

Under the pressure of the changes in the information infrastructure which has begun with the computer revolution and the development of new communication generation, the traditional model of science library undergoes transformations oriented from static communication towards active, interactive, dynamic interaction with the users. In this way it has developed as a basic structural component of the information society being constructed.

The model of information society gains position in the accepted theory for the European development. The frame programmes are concrete terms of the concept, their priorities change in conformity with the development of technologies and the practical creation of information infrastructure.

On the other hand, the prior fields in knowledge development are a consequence of the intensification of research and the influence of the new forms of knowledge organization, including the self-organizing "invisible colleges", as an objective determinant of the new information technologies.

The general structure-determining components of the information infrastructure are represented in their interaction according to the given definition of the term infosphere. The concrete types of realization of infosphere expressed in transformation of the functions of science library from traditional to electronic and virtual, and the influence of the information superhighway Internet upon this process are regarded in their mutual determination and dynamic development.

The integration of science library into the educational process, which is an essential stage of the preparation of the intellectual potential for creation of new knowledge, requires the development of its immanent functions at a new level. The social needs regarded in their dynamic development predetermined by the socio-economic parameters of information society force the infiltration of science library into the new structure and contents of the educational process. Education and teaching are shown as a dynamic and changing constituent in the life of man. The science library has a decisive part also in the realization of the principles of democratization of society through development of new functions with which it participates of full value in the realization of the right of each citizen to information.

Chapter Two outlines the challenges of the changes in the system environment of science library. The problem situation of the science library is identified and the major factors of change with their dynamics and tendencies of impact are determined. The expected transformations of the functional model of science library in the fields of interaction with the dynamically changing system environment are traced.

The problem trends and the types of influences upon the structure determining and resource ensuring elements of science library are localized. The change of the system environment includes not only the new technological medium in which the science library functions, but also the corresponding changes in the users' interests and attitudes towards search of new information sources and resources (Internet, online data bases, electronic documents, etc.). This interdetemined series of changes leads to a change in the functions from rendering document sources towards rendering an access to them, i.e. the structure and the character of library services change – to direct use of library functions and to new structure of information resources.

The defined general problems require implementation of such strategies of science library development which ensure flexible and dynamic adaptation to the changes of the system environment and the conditions of its functioning. The models of its future development are determined by the concretization of the reasons for eliminating the library institution as an element of the new infosphere and respectively by the loss of staked out claims in the information field and of positions in the dynamically developing virtual markets.

The future possibilities contain the two mutually excluding extremes. At the first place among the destructive factors is the insoluble for the traditional libraries contradiction between the static of the implemented library classifications and the dynamics of modern knowledge. This contradiction is followed by others which are consequences of the considerable deviations from the initial conditions of functioning at which the traditional science library model is approved. Independently of the periodical improvements of the leading libraries in the world the unceasing dynamic changes are reflected by all elements of the science library models and it stops to function efficiently. The two shown models of the library information resource structuring – focal and fractal are promising from the point of view of the sustainable preservation and development of the basic function of science library. The implementation of modern methods and means for systematization and classification of information resources brings order in the increasing chaos, allowing maximum satisfaction of the search of information by the users. The advantage of the fractal model is determined by the context of the dynamics of changes in the system environment and by the necessity of corresponding to this dynamics flexibility, adaptability and maximum productivity. The fractal library is shown as a stage and a basis for passing to a virtual-fractal model of a library of the future.

The functional transformation of science library is shown in its dynamic interaction with the system environment. The reformation of the structure and type of library stocks is pointed at in conformity with the changes of infosphere and the necessity of achieving coordination between the information resources needed for research and the information ensured by science library. The evolution and the stages in the development of system environment – vertically and horizontally structured hierarchies, and the place of science library in them are shown as a natural consequence of the accepted concept "from possession to access" and in the light of the social and political and economic processes in the last thirty years in Europe and USA.

The specific and the general information functions of scientific library are re-defined in accordance with the pressing requirements of the system environment. The concrete reflections of the basic and super structural technological ware in the communication functions of library are expressed in the strategies for their transformation and achievement of integration of the user with the generated by them information resources.

A comparative analysis of the assessment criteria for the quality of the traditional and the new information resources is made. The classification of the new information resources is shown from the point of view of the new library functions. The recent achievements are in the field of creation of new and most valuable information resources – data metabases. This is an example for the technology of creation of such resources which includes an algorithm of procedures.

Chapter Three gives the basic elements of the library marketing-mix refracted by the realized at a new technological level traditional and new library functions. The elements of the marketing system of science library are marked as the different marketing concepts are shown in their evolution and are analyzed from the point of view of the new functionality of science library.

The main participants in the market of telecommunication and information services are presented, and the necessity of new moving force is substantiated which should give an impetus to the transformation of library functions. It is underlined that the bringing of competition to the field of library services does not necessarily require privatization of its stocks. The coming of new participants who get the right to carry out library services might look like a heretical perspective, but it is a reality in the developed countries. The variants of realization are multiple in the whole spectrum from competition to partnership including intermediate mobile units, etc.

The position of science library in the modern marketing system is presented as a consequence of the influence of the last upon its new functionality. The advantages of its new functional models are shown at a first place in Chapter Two. The described factors of impact on the marketing system are concretized in the priorities for development of science library – moving the stress from possession to access and to the activities ensuring it. On its part, the marketing system turns into a global factor for influence and dictates the changes in the future. The accelerating effect of new technologies and the expected reflection in the library functions are shown in the context of conversion of the library marketing system. The identification of the elements in the library marketing-mix is made by the offered marketing concept of the library. The integration and the adaptation of all elements of the library marketing-mix to dynamic changes of the users' interests require a precise monitoring for the achievement of which some of the most successful methods are shown. The impact of new technologies upon the heuristic potential of marketing finds most dynamic reflection in the evolution of electronic markets, especially in the tendencies of information markets' segmentation. The reaching of optimum mix is shown in dynamic interaction of the life cycle of the new information products of libraries with their position in the new infosphere. The offered strategies for management, organization and optimization of exchange rates are shown in the conditions of reorientation of science library towards active interactive communication with the identified user – individual and mass.

By concretization of the role of the library information broker in creating additional digital contents in the science library an attempt is made for identification of the transformation of the science library functions into services. The parameters of digital marketing and the market segmentation in Bulgaria are described. Among the mentioned different strategies and their basic elements the advantage of CRM is underlined for finding the place of the science library information services. The potential market niches are shown in the context of the new directives of the European Union and their expected impacts upon the constructed information infrastructure. The setting of Internet as a universal communication channel accelerates the new mobile communications, the share of intelligent nets increases as well as the use of computers as terminals. The stimulating influence of these factors upon Interconnect technologies leads to restructuring and expansion of markets, new services and participants, new level of production and new products.

The offered technology of information products and services creation by the science library is an attempt of synthesizing its mission and new functionality, having in mind the modern tendencies and the experience of the leading companies in this field.

A strategical perspective in the new conditions is the adequate use of the global net Internet through integration of functions and services, realized in the library. The growing of Internet and the integration of intelligent networks gives an opportunity for achieving surplus value from the services realized by them.

Having in mind these tendencies and moving the problem into the sphere of library services, it is necessary to determine the priorities and to develop such functionality of the library, by which not only products, but also services will be produced and rendered. In this way all elements along the channel of communication will be united and not only the high efficiency of information services and sustainable presence of library in the market niche, but also the real transformation of libraries from a passive, stationary communication institution to an active mediator and knot of research communication will be achieved.

The priorities of the Sixth Frame Programme are united into one major trend – the creation of integrated European research space which is achieved also by new institutionalization of the "invisible colleges", by new virtual structures and unions. The strategical trends are two – better life quality and a better life environment, which is a necessary condition for accumulation of a potential for expanding the borders of knowledge.

TABLE OF CONTENTS