ITHEA

International Journal INFORMATION MODELS & ANALYSES

2012 Volume 1 Number 2

International Journal
INFORMATION MODELS & ANALYSES
Volume 1 / 2012, Number 2

IJ IMA is official publisher of the scientific papers of the members of the ITHEA® International Scientific Society

IJ IMA rules for preparing the manuscripts are compulsory.
The **rules for the papers** for ITHEA International Journals as well as the **subscription fees** are given on www.ithea.org .
The camera-ready copy of the paper should be received by ITHEA® Submission system http://ij.ithea.org .
Responsibility for papers published in IJ IMA belongs to authors.

General Sponsor of IJ IMA is the **Consortium FOI Bulgaria** (www.foibg.com).

International Journal "INFORMATION MODELS AND ANALYSES" Vol.1, Number 2, 2012

Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with
Institute of Mathematics and Informatics, BAS, Bulgaria,
V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
Universidad Politechnika de Madrid, Spain,
Hasselt University, Belgium
Institute of Informatics Problems of the RAS, Russia,
St. Petersburg Institute of Informatics, RAS, Russia
Institute for Informatics and Automation Problems, NAS of the Republic of Armenia,
and Federation of the Scientific - Engineering Unions /FNTS/ (Bulgaria).

Publisher: **ITHEA®**
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com
Technical editor: Ina Markova

Printed in Bulgaria

ISSN 1314-6416 (printed) **ISSN 1314-6424 (CD)** **ISSN 1314-6432 (Online)**

NON SMOOTH OPTIMIZATION METHODS IN THE PROBLEMS OF CONSTRUCTING A LINEAR CLASSIFIER

Yurii I. Zhuravlev, Yuryi Laptin, Alexander Vinogradov, Nikolay Zhurbenko, Aleksey Likhovid

***Abstract**: We consider the technique using nonsmooth optimization methods for pattern recognition problems. The results of numerical experiments of comparison of the proposed approach with support vector machines are presented.*

***Keywords**: cluster, decision rule, discriminant function, linear and nonlinear programming, nonsmooth optimization*

***ACM Classification Keywords**: G.1.6 Optimization - Gradient methods, I.5 Pattern Recognition; I.5.2 Design Methodology - Classifier design and evaluation*

***Acknowledgement**: This work was done in the framework of Joint project of the National Academy of Sciences of Ukraine and the Russian Foundation for Fundamental Research № 10-01-90419.*

Introduction

Mathematical models of the problems of constructing linear and nonlinear discriminant functions (classifiers) have been considered in many papers (see, e.g., [1, 2]). An approach proposed in [3–6] has certain advantages over the method of support vector machines (SVM). Mathematical model proposed in these papers can be conveniently represented in the form of convex optimization problems. The technique using efficient nonsmooth optimization methods [7] to solve these problems will be considered below. We present the results of computational experiments for specific test problems of large dimension.

The first section presents new results [8-11], allowing efficiently generate the equivalent unconstrained optimization problem for problems with constraints. In the second section we describe the mathematical models of constructing classifiers. In the third section characteristics of test problems and results of computational experiments are given.

1. A brief description of methods for solving nonsmooth convex constrained optimization problems

The scheme for solving optimization problems with constraints consists in construction of an equivalent unconstrained optimization problem, and in its solution by efficient subgradient algorithms (for example, r-algorithm by N.Z.Shor [7]).

To construct an equivalent unconstrained optimization problem we use the following approaches:

- exact penalty functions method;

- conical extensions of functions [10, 11].

Exact penalty functions are considered in a large number of publications, for example in [7,12,13]. The practical usage of penalty functions and various generalizations are considered in these papers.

Convex programming problem with constraints has the form: find

$$f^* = \min\{f(x) : x \in C\} \quad (1)$$

where $C = \left\{x \in R^n : h_i(x) \le 0, i = 1,..., m\right\}$, $f, h_i : R^n \to R$ – convex functions.

Let f, h_i take finite values at all x. We will consider the penalty function of the form

$$S(x,s) = f(x) + s \cdot h^+(x), \; s \in R, \; s \ge 0, \quad (2)$$

where $h(x) = \max\{h_i(x), \; i = 1,..., m\}$, $x^+ = \max\{0, \; x\}$. Consider the problem: find

$$S^*(s) = \min\left\{S(x,s) : x \in R^n\right\} \quad (3)$$

Penalty function $S(x,s)$ is exact for given values of penalty coefficients s, if the solutions of (1) and (3) coincide.

To select the values of the penalty coefficients it is usually necessary to solve the auxiliary dual problems, or this selection is put on the user, which leads either to an overestimation of the values used, or to the necessity of multiple solving the same problem for the satisfactory selection of the penalty coefficients. In [8, 9] an approach to build an automatic procedure for determining the values of penalty coefficients during the optimization process is proposed. Consider a brief description of this approach.

We assume that C is a bounded closed set. Denote by $S'(x,s,p)$, $f'(x,p)$, $h'(x,p)$ derivatives of functions S, f, h at a point $x \in R^n$ in the direction p for the fixed value s, $p(x,y) = (y-x)/\|y-x\|, \; y \ne x$.

Let $\tilde{x}$ be a solution of (3), and convergent sequences $x^k \in R^n$, $y^k \in C$, $k = 0,1, ...$ are given, $x^k \to \tilde{x}$ when $k \to \infty$. The sequence x^k is generated during the solution of problem (3) by algorithm for unconstrained optimization, the point y^k is determined by an auxiliary rule for the current point x^k. Such rules can be defined in various ways, for example, we may assign $y^k = y^0$, where y^0 is an initial feasible point such that $h(y^0) < 0$, or may choose y^k among the feasible points generated in previous iterations.

Let $x^k \notin C$. We denote by $\pi_C(x^k, y^k)$ the intersection point of the segment $\left[x^k, y^k\right]$ with the boundary of the set C, $\overline{x}^k = \pi_C(x^k, y^k)$.

Theorem **1** [9]. Let $\varepsilon > 0$, $s > 0$ such that for each x^k, $x^k \notin C$, $k = 0,1, ...$ *the following constraint* is satisfied

$$S'(\overline{x}^k, s, p(\overline{x}^k, x^k)) \ge \varepsilon \quad (4)$$

Then $\tilde{x}$ *is* a solution of *(1), i.e.* $S(x,s)$ *is* the exact penalty function.

Thus, for using this approach to determine the value of penalty coefficient s it is necessary to check the condition (4) at each step of the optimization algorithm, which requires the solution of one-dimensional problem of finding the intersection point $\overline{x}^k = \pi_C(x^k, y^k)$ of the segment $\left[x^k, y^k \right]$ with the boundary of the set C. This search procedure can be implemented effectively.

In the case when inequality (4) at some iteration of the algorithm is violated, we will increase the penalty s, so that inequality (4) is satisfied. This increase must be not less than B, where $B > 0$ is a given parameter. It is easy to see that if there exists such finite $\overline{s}$ that for $s > \overline{s}$ the inequality (4) holds on all iterations of the algorithm, then the amount of such penalty increases is finite throughout the optimization process.

Theorem **2** [9]. Given a sequence $x^k \in R^n$, $k = 0,1, \ldots$ converging to a solution $\tilde{x}$ of problem *(3),* $y^k = y^0$, $k = 1, 2, \ldots,$ $h(y^0) < 0$. Then there exists $\overline{s} < \infty$ such that the conditions of Theorem *1* are *satisfied* for $s > \overline{s}$.

In [9] a special rule of the choice of y^k was considered. It was showed that $\overline{s} = \sum_{i=1}^{m} u_i^*$ where u_i^*, $i = 1, \ldots, m$ are optimal values of dual variables.

Theorem 2 allows us to construct the automatic determination of penalty coefficients during the optimization process in the case when the starting point y^0, $h(y^0) < 0$, is known. If this point is unknown, then the solving process of the problem is divided into two phases - the first phase is to find the point y^0, $h(y^0) < 0$, on the second phase the original problem is solved.

Conical extensions of functions [10, 11] is another approach to generate an equivalent unconstrained optimization problem. The objective function of this problem coincides with the objective function of the original problem on the feasible set. Outside of the feasible set the formed function is defined by the behavior of the objective function of the original problem on the boundary of the feasible set. The original objective function can not be defined outside the feasible set. As above consider a brief description of this approach.

As before, the problem (1) is considered. It is assumed that C is a closed bounded set, a feasible point $x^0 \in C$ such that $h_i(x^0) < 0$, $i = 1, \ldots, m$ and a number E, $E < f(x^0)$ are given.

For $x \notin C$ we denote $\pi_C(x^0, x)$ the intersection point of the segment $[x^0, x]$ with boundary of the set C. Let

$$R_C(x^0, x) = \frac{\left\| x - x^0 \right\|}{\left\| \pi_C(x^0, x) - x^0 \right\|}, \tag{5}$$

$$\chi^E(x) = E + (f(\pi_C(x^0, x)) - E) \cdot R_C(x^0, x), \tag{6}$$

$$\psi^E(x) = \begin{cases} f(x), & \text{if } x \in C \\ \chi^E(x), & \text{if } x \notin C \end{cases}. \tag{7}$$

It is easy to see that $\psi^E(x)$ is a continuous function. Consider the problem of finding

$$\psi^{*E} = \inf\left\{\psi^E(x) : x \in R^n\right\}. \quad (8)$$

Lemma **1** [11]. Let $E < f^*$ *, then* $\psi^{*E} = f^*$.

Theorem 3 [11]. Let C is a closed bounded set, $C \subset \operatorname{int} \operatorname{dom} f$. Then there exists a finite number E^* such that $\psi^E(x)$ is a convex function for all $E \le E^*$.

We denote $g_f(x)$, $g_h(x)$ subgradients of functions f , h at the point x .

Theorem 4 [11]. Let $\overline{x} = \pi_C(x^0, x)$. *Then the vector*

$$g = g_f(\overline{x}) + \frac{E - f(\overline{x}) - \left\langle g_f(\overline{x}), x^0 - \overline{x}\right\rangle}{\left\langle g_h(\overline{x}), x^0 - \overline{x}\right\rangle} g_h(\overline{x}) \quad (9)$$

Is a subgradient of function $\chi^E(x)$ at the point x (subgradient of function $\psi^E(x)$ if $x \notin C$).

Thus, if f^* , E^* are known, and conditions of the Lemma 1 and Theorem 3 are satisfied, then any algorithm for minimizing convex functions can be used for solving the problem (8). The solution of the problem (8) is a solution of the original problem (1).

Consider the case where the values of f^* and E^* are unknown. Denote as $f'(x, p)$ the derivative of function f at the point x in direction p , $p(x^0, x) = (x - x^0)\Big/\left\|x - x^0\right\|$. Suppose that we use a convergent algorithm A for unconstrained minimization of convex functions, at each iteration of which the value of objective function and its subgradient are computed.

Theorem 5 [11]. Let numbers E and $\delta > 0$ are given, algorithm A is used to solve problem (8), and the following condition is satisfied at each iteration k of the algorithm: if $x^k \notin C$ then

$$E < f(\overline{x}^k) - \delta, \quad (10)$$

$$E < f(\overline{x}^k) - f'(\overline{x}^k, p(\overline{x}^k)) \cdot \left\|\overline{x}^k - x^0\right\| \quad (11)$$

where $\overline{x}^k = \pi_C(x^0 x^k)$, x^k *is* the current point at the iteration k . Then the sequence of points generated by the algorithm A converges to the solution *of problem (1).*

If at some iteration k inequalities (10), (11) are violated, then it's necessary to change the value E iteratively: $E = \Delta - B$, where $\Delta = \min\left\{f(\overline{x}^k), \quad f(\overline{x}^k) + f'(\overline{x}^k, -p(\overline{x}^k)) \cdot \left\|\overline{x}^k - x^0\right\|\right\}$, $B > 0$ is a given parameter.

In view of finiteness f^* and E^* there will be just finite number of changes of the value E , after which the algorithm converges to the optimal solution of problem (1).

Thus, the approach under consideration allows to construct an equivalent unconstrained optimization problem, and to solve the original problem using unconstrained optimization algorithm.

2. Description of the mathematical models for constructing classifiers

The problems of constructing linear classifiers are considered in [3-5].

Given a set of linear functions $f_i(x,W^i)=\left\langle w^i,x\right\rangle+w_0^i,\ i=1,...,\ m$, where $x\in R^n$ is an attribute vector, $W^i=(w_0^i,w^i)\in R^{n+1}$ is a parameter vector, $i=1,...,m$.

Let introduce the notations $W=(W^1,...,W^m)$, $W\in R^L,\ L=m(n+1)$. When $m>2$ we consider the linear classification algorithms (linear classifiers) in the form:

$$a(x,W)=\arg\max_i\left\{f_i(x,W^i):\ i=1,...,\ m\right\},\ x\in R^n,\ W\in R^L. \tag{12}$$

When $m=2$,then the linear classifiers are defined by linear functions $f(x,W)=(w,x)+w_0,\ W=(w,w_0)\in R^{n+1}$, and are presented in the form

$$a(x,W)=\begin{cases}1,\ \text{if}\ f(x,W)\ >0,\\ 2,\ \text{if}\ f(x,W)\le 0,\end{cases} \tag{13}$$

We consider a given finite family of disjoint sets (training set) of points: $\Omega_i=\left\{x^t:x^t\in R^n,\ t\in T_i\right\}$, $i=1,...,\ m$, $T=\bigcup_{i=1}^{m}T_i$.

It is said that the classifier $a(x,W)$ correctly separates the points of $\Omega_i,\ i=1,...,\ m$, if $a(x,W)=i$ for all $x\in\Omega_i,\ i=1,...,\ m$. We define function $i(t)$ returning index of the set which contains the point $x^t\in\Omega_{i(t)}$, $t\in T$.

If $m>2$, then the value

$$g^t(W)=\min\left\{f_i(x^t,W^i)-f_j(x^t,W^j):\ j\in\{1,...,\ m\}\setminus i,\ i=i(t)\right\}=$$

$$=\min\left\{\left\langle w^i-w^j,x^t\right\rangle+w_0^i-w_0^j:\ j\in\{1,...,\ m\}\setminus i,\ i=i(t)\right\} \tag{14}$$

is called a gap of classifier $a(x,W)$ at the point x^t, $t\in T$.

In the case of $m=2$ a classifier gap at the point x^t is the value

$$g^t(W)=\begin{cases}\ f(x^t,W),\text{if}\ t\in T_1,\\ -f(x^t,W),\ \text{if}\ t\in T_2.\end{cases} \tag{15}$$

The value $g(W)=\min\left\{g^t(W):t\in T\right\}$ is called a gap of classifier $a(x,W)$ on the family of sets $\Omega_i,\ i=1,...,\ m$. The classifier $a(x,W)$ correctly separates the points of the sets $\Omega_i,\ i=1,...,\ m$, if $g(W)>0$. The sets $\Omega_i,\ i=1,...,\ m$ are called linearly separable, if there exist a linear classifier that correctly separates the points of these sets.

If the sets Ω_i, $i=1,...,\ m$ are linearly separable, then the problem of constructing an optimal classifier (determination of the parameters W) has the following form: find

$$g^* = \max_W \left\{ g(W) : \eta(W) \le 1,\ W \in R^L \right\}. \tag{16}$$

Here $\eta(W)$ is the norm of vector W, $\eta(W) = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(w_j^i)^2}$.

This problem can be written in the equivalent forms

$$\eta^* = \min_V \left\{ \eta(V) : g(V) \ge 1,\ V \in R^L \right\}, \tag{17}$$

$$\eta^* = \min_V \left\{ \eta(V) : g^t(V) \ge 1, t \in T, V \in R^L \right\}. \tag{18}$$

The equivalence is understood in the sense that if W^* is the optimal solution of problem (16) then equalities $V^* = W^* \Big/ g^*$, $\eta^* = 1 \Big/ g^*$ hold for the optimal solution V^* of problem (17) or (18).

A continuous relaxation of the problem of minimizing the empirical risk is proposed in [4-5] to build a classifier in the case of linearly inseparable sets. This relaxation has the form: find

$$q^* = \min \sum_{t \in T} d^t(W) \tag{19}$$

under constraints

$$\eta(W)^2 \le 1, \tag{20}$$

$$\sum_{t \in T_i} d^t(W) \le |T_i| - 1,\ i = 1,...,\ m \tag{21}$$

$$d^t(W) \le 1,\ t \in T, \tag{22}$$

where $d^t(W) = \max\left(0,\ \frac{1}{B}\left(\bar{\delta} - g^t(W)\right)\right)$, $\bar{\delta} > 0$ is a parameter of the reliability required for separating points of the training set, B is a parameter (a sufficiently large positive number).

Consider the problem: find

$$\eta^* = \min\left\{ \frac{1}{2}\eta(V)^2 + C\sum_{t \in T} \xi^t : g^t(V) \ge 1 - \xi^t, \xi^t \ge 0, t \in T, V \in R^{n+1} \right\}. \tag{23}$$

This problem is solved by the method of support vector machines (SVM) for the case $m = 2$. The SVM method is used to build an optimal classifier for linearly separable classes, and for linearly nonseparable classes.

Note that for linearly separable classes the problems (18) and (23) have the same solutions if coefficient C is sufficiently large. This follows from the theorem on non-smooth penalties [7,12].

In [5] it was shown that in the case of linearly nonseparable classes the problem (23) can be obtained from (19) - (22) by Lagrangian relaxation with a special selection of the values of the Lagrange multipliers.

A choice of coefficient C is a significant problem when the problem (23) is used in the case of linearly nonseparable classes. It should be noted that this problem does not arise when problem (19) - (22) is used.

Consider the characteristics of problems (16) and (19)-(22) which are useful when the approaches described in the previous section are used:

The point $W^0 = 0$ is an interior point of the feasible set.

1. The optimal values of these problems are always greater than or equal to zero.
2. Implementation of one-dimensional search for a point on the boundary of the feasible set is simple: let $\eta^k = \eta(W^k)^2$ is a squared norm of the points W^k, $\eta^k > 1$, then the point $\bar{W} = \frac{W^k}{\sqrt{\eta^k}}$ is the required point on the boundary of the feasible set.
3. Functions $g^t(W)$ have the property – $g^t(\alpha W) = \alpha g^t(W)$.

3. Software implementation and results of computational experiments

Software implementation for the following approaches to the problems under consideration was developed:

- for problems (16) and (19)-(22) - a method of exact penalty functions with automatic adjustment of the penalty factor, the method of convex conical extensions;
- for problems (18) and (23) - a method of exact penalty functions without the automatic adjustment of the penalty coefficient.

Unconstrained optimization problems, to which the original problems with the constraints are reduced, were solved using r-algorithm by N.Z.Shor [7].

The problems of constructing linear classifiers for two classes were generated randomly for computational experiments. The parameters of problems varied in the range:

- the dimension n of attribute space R^n – from 5 to 100;
- the number of points in the training set – from 40 to 100 000.

The points in the training set for each class were generated on the basis of a uniform distribution within the unit cube. These cubes are shifted relative to each other along the first coordinate, so that the distance between them is equal to unity. For each problem P_0, constructed in such way, a family of problems P_i, $i = 1,..., 10$ was generated by reducing the distance between the classes (cubes). The distance between classes of the problem P_i is equal to 2^{-i}. All problems from the generated families are linearly separable.

To construct linearly nonseparable problems (sets) a membership to a class of some points of training set is changed.

According to the results of computational experiments we can do the following conclusions:

– the method of exact penalty functions with automatic adjustment of the penalty coefficient and the method of convex conical extensions showed approximately the same efficiency for the problem (1 1), all problems from the generated families were solved successfully (the accuracy of the objective function $\sim 10^{-6}$), the iteration

number of the r-algorithm changed from ~ 100 for the dimension $n = 5$ to ~ 1500 for the dimension $n = 100$;

– the choice of coefficient C is essential when using the model SVM (problem (23)), in the computational experiments we used the value $C = 1000$, and the problems P_i, $i \le 5$ from the generated families were solved successfully (separating hyperplanes were found), but the problems P_i, $i \ge 7$ were not solved (separating hyperplanes were not found).

The developed software tools were compared with existing software (LIBSVM – http://www.csie.ntu.edu.tw/~cjlin/libsvm/). In the table the elapsed time for solving problems of constructing a linear classifier in the space of dimension $n = 100$, depending on the number of points in the training set, is shown. The standard settings for LIBSVM were used.

Table.

Number of points	The solution time, in seconds	
	LIBSVM	Automatic adjustment of the penalty coefficient
5000	9.421	20.8
10000	24.234	24.3
25000	83.468	43
40000	186.484	51,1
50000	266.203	84,8

Thus, the methods of nonsmooth optimization provide greater opportunities in the construction of linear classifiers in comparison with traditional approaches. At the same time the performance of the developed programs is comparable with existing software.

References

1. Vladimir Vapnik. Estimation of Dependences Based on Empirical Data. – Springer Verlag, 2006, 2nd edition.
2. Thorsten Joachims. Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms. Kluwer, 2002
3. Laptin Yu., Likhovid A. P., Vinogradov A.P. Approaches to Construction of Linear Classifiers in the Case of Many Classes // Pattern Recognition and Image Analysis, Vol. 20, No. 2, 2010, p. 137-145.
4. Zhuravlev Yu., Laptin Yu., A.Vinogradov Minimization of empirical risk in linear classifier problem // New Trends in Classification and Data Mining, ITHEA, Sofia, Bulgaria, 2010. – Pages 9-15
5. Журавлев Ю.И., Лаптин Ю.П., Виноградов А.П. Минимизация эмпирического риска и задачи построения линейных классификаторов // Кибернетика и системный анализ. 2011, № 4.- С. 155 – 164.
6. Журавлев Ю.И., Лаптин Ю.П., Виноградов А.П. Построение нелинейных классификаторов в случае многих классов // Applicable Information vodels. ITHEA, Sofia, 2011. – P. 7 – 13
7. Shor N. Z. Nondifferentiable Optimization and Polynomial Problems. – Amsterdam / Dordrecht / London: Kluwer Academic Publishers, 1998. – 381 p.
8. Лаптин Ю.П. Некоторые вопросы использования негладких штрафных функций // Теорія оптимальних рішень. 2011, № 10. с. 127 – 135.

9. Лаптин Ю.П. Некоторые вопросы определения коэффициентов негладких штрафов // Теорія оптимальних рішень. 2012, № 11. (В печати).
10. Лаптин Ю.П. Один подход к решению нелинейных задач оптимизации с ограничениями // Кибернетика и системный анализ. 2009, № 3. С. 182 – 187.
11. Лаптин Ю.П., Лиховид А.П. Использование выпуклых продолжений функций для решения нелинейных задач оптимизации // Управляющие машины и системы. 2010, № 6. – С. 25–31.
12. Пшеничный Б.Н. Метод линеаризации. – М.: Наука. – 1983. – 136 с.
13. Evtushenko Y.G., Rubinov A.M., and Zhadan V.G. General Lagrange-type functions in constrained global optimization. Optimization Methods and Software, 2001, vol. 16, Part I: pp.179-217, Part II: pp. 231-256.

Information about authors

Yurii I. Zhuravlev *– Academician of the RAS, Deputy Director, Dorodnicyn Computing Centre of the RAS, Vavilova 40, 119333 Moscow, Russian Federation; e-mail: zhuravlev@ccas.ru*

Yuryi Laptin *– Senior Researcher, V.M.Glushkov Institute of Cybernetics of the NASU, Prospekt Academika Glushkova, 40, 03650 Kyiv, Ukraine; e-mail: laptin_yu_p@mail.ru*

Alexander Vinogradov *– Senior Researcher, Dorodnicyn Computing Centre of the RAS, Vavilova 40, 119333 Moscow, Russian Federation; e-mail: vngrccas@mail.ru*

Nikolay Zhurbenko *– Senior Researcher, V.M.Glushkov Institute of Cybernetics of the NASU, Prospekt Academika Glushkova, 40, 03650 Kyiv, Ukraine; e-mail: zhurbnick@yandex.ru*

Aleksey Likhovid *– Researcher, V.M.Glushkov Institute of Cybernetics of the NASU, Prospekt Academika Glushkova, 40, 03650 Kyiv, Ukraine; e-mail: o.lykhovyd@gmail.com*

ON SOME PROPERTIES OF REGRESSION MODELS BASED ON CORRELATION MAXIMIZATION OF CONVEX COMBINATIONS

Oleg Senko, Alexander Dokukin

***Abstract**: The article is devoted to thorough study of a new regression method performance. The proposed method based on convex correcting procedures over sets of predictors is subject to modifications and tested in comparison with the acknowledged regression utility. The modifications touch both resource consumption and quality aspects of the method and tests are performed with sets of generated samples.*

***Keywords**: forecasting, bias-variance decomposition, convex combinations, variables selection.*

***ACM Classification Keywords**: G.3 Probability and Statistics - Correlation and regression analysis, Statistical computing.*

Introduction

Present article continues a series of works devoted to an approach in which optimal forecasting models are built by large ensembles of preliminary trained predictors that in turn can be simple univariate egressions. Several statistical methods were developed last years that allow improving significantly prognostic ability of regression modeling in tasks of high dimension. Efficiency of these methods is associated with effective selecting of prognostic variables. Least angle regression or Lasso [Tibshirani, 1996], [Efron et al., 2004] methods may be mentioned thereupon. However we believe that a problem of low generalization ability of empirical models in high-dimensional tasks cannot be considered completely solved. Thus, a number of convex correcting procedures optimization method has been proposed [Senko, 2009], [Senko et al., 2010], [Senko et al., 2011].

Suppose that we have set of L predictors $z_1,...,z_L$ that forecast some variable Y. Let $c=(c_1,...,c_L)$ be a vector of nonnegative coefficients satisfying condition $\sum_{i=1}^{L} c_i = 1$. Convex correcting procedure (CCP) calculates forecasted value as a weighted sum of prognoses that are calculated by single predictors:

$$z_{ccp}(c) = \sum_{i=1}^{L} c_i z_i .$$

Convex combinations are widely used in pattern recognition. The bagging and boosting techniques [Breiman, 1999], [Kuncheva, 2004] may be mentioned as an example, as well as methods based on collective solutions by sets of regularities [Zhuravlev et al., 2008], [Zhuravlev et al., 2006], [Kuznetsov et al., 1996]. Convex correction is used in regression tasks also. Thus, neural networks ensembles are discussed in [Brown et al., 2005] that are based on optimal balance between individual forecasting ability of predictors and divergence between them. Efficiency of convex combinations of repressors' pairs was shown in [Senko, 2004]. Earlier it was shown that error of predictors' convex combination in any case is not greater than the same convex combination of single predictors' generalized errors [Krogh et al., 1995].

A method for CCP optimization that is based on minimization of general error estimates was studied in [Senko, 2009], [Senko et al., 2010]. Experiments with simulated data demonstrated that CCP error optimization also implements effective selection of informative prognostic variables.

In [Senko et al., 2011], however, it was shown that CCP variance is decreased comparing to the same combination of single predictors' variances and such a decrease deteriorates the CCP's prognostic ability. An additional adjustment to be made to CCP predictions leads to the necessity of maximizing Z_{ccp} and Y correlation. Such a technique based on the same concept of irreducible ensembles searching that was used in [Senko et al., 2010] was proposed in the article.

Again, experiments with simulated data demonstrated that CCP correlation optimization shows great results comparing to LARS method, the only drawback of the result being that LARS was implemented by the authors and thus may be not the optimal one. So, in present article the method is compared to widely acknowledged Glmnet for Matlab written by Jerome Friedman and Hui Jiang [Friedman et al., 2007], [Friedman et al., 2010].

In the next few sections we afford repeating some definitions and theorems concerning irreducible ensembles searching and convex correctors' correlation optimization. Then, some modifications to the correlation maximization method (CCPCMM) will be described. And finally, the results of experiments will be shown.

Irreducible ensembles relative to correlation coefficients

It is supposed further that predictors from initial set are additionally transformed with the help of optimal univariate regression models to achieve best forecasting ability. Such predictors will be further called reduced. In other words predictor z will be called reduced if for all α, β the inequality

$$E_\Omega\left(Y-\alpha z-\beta\right)^2 \le E_\Omega\left(Y-z\right)^2$$

is correct. Here $E_\Omega\left(X\right)$ is mathematical mean of X by space of admissible objects with defined σ-algebra and probability measure. It will be further denoted as $\hat{X}$. It is known that following inequalities are true for a reduced predictor z:

$$\operatorname{cov}\left(Y,z\right)=E_\Omega\left[\left(Y-\hat{Y}\right)\left(z-\hat{z}\right)\right]=E_\Omega\left(z-\hat{z}\right)^2.$$

The use of the described conditions allows effectively searching ensembles with maximal prognostic ability, but the approach has its drawbacks. First of all, there are many ensembles with the prognostic ability close to the optimal one and it would be rational using them all. Secondly, CCP always decrease prognoses' variation and univariate correcting transformation becomes inevitable. Of all predictors the maximal quality is provided by the one most correlated with Y.

Standard Pearson correlation coefficient is defined as the ratio:

$$K\left(Y,Z_{ccp}\right)=\frac{\operatorname{cov}\left(Y,Z_{ccp}\right)}{\sqrt{V\left(Y\right)V\left(Z_{ccp}\right)}}.$$

On the other hand $\operatorname{cov}\left(Y,Z_{ccp}\right)=\sum_{i=1}^{L}c_i\operatorname{cov}\left(Y,z_i\right)$. But z_i is a reduced predictor. So, $\operatorname{cov}\left(Y,z_i\right)=V\left(z_i\right)$, $i=1,\ldots,L$ and therefore

$$K\left[Y,Z_{ccp}\left(c\right)\right]=\frac{\sum_{i=1}^{L}c_iV\left(z_i\right)}{\sqrt{V\left(Y\right)}\sqrt{\sum_{i=1}^{L}c_iV\left(z_i\right)-\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}c_ic_j\rho_{ij}}},$$

where ρ_{ij} denotes discrepancy between i-th and j-th predictors.

Further discussions are based on irreducible ensemble concept. A set of predictors $\tilde{z}$ is called irreducible ensemble if removing of at least one predictor from it does not allow constructing CCP with the same prognostic ability as of $\tilde{z}$. The following is a strict definition of ensemble's irreducibility.

Definition 1. Sets $\overline{D_L}$, D_L from $\mathbb{R}^L$ are defined as

$$\overline{D_L} = \left\{ c \left| \sum_{i=1}^{L} c_i = 1; c_i \geq 0, i = 1,\ldots,L \right. \right\},$$

$$D_L = \left\{ c \left| \sum_{i=1}^{L} c_i = 1; c_i > 0, i = 1,\ldots,L \right. \right\}.$$

Definition 2. Set of predictors $z_1,\ldots,z_L$ is called irreducible ensemble relative to some functional $F(c)$, that characterize forecasting ability, if there is such vector $c^* \in D_L$, that $\forall c' \in \overline{D_L}$, $F(c^*) > F(c')$.

A set of points from $\mathbb{R}^L$ simultaneously satisfying constraints: $\sum_{i=1}^{L} c_i = 1$ and $\sum_{i=1}^{L} c_i V(z_i) = \theta$ will be further referred to as $W(\theta)$.

Theorem 1. A necessary condition of irreducibility of predictors set $z_1,\ldots,z_L$ relative to $K(Y, Z_{ccp})$ is existence of such real θ that quadratic functional

$$P_f(c) = \sum_{i=1}^{L}\sum_{j=1}^{L} c_i c_j \rho_{ij}^{v}.$$

achieves strict maximum at $W(\theta)$ in $c_1^*,\ldots,c_L^*$ that satisfies conditions $c_i^* > 0$, $i = 1,\ldots,L$.

The maximum necessary condition is existing of positive $\theta > 0$, such that the following equation holds

$$\sum_{i=1}^{L}\sum_{j=1}^{L} c_i c_j \rho(z_i, z_j) \to \max \qquad (1)$$

with the next contingencies:

$$\sum_{i=1}^{L} c_i E(z_i^2) = \theta,$$

$$\sum_{i=1}^{L} c_i = 1, \qquad (2)$$

$$c_i \geq 0,\ i = 1,\ldots,L.$$

Lets write down a Lagrange functional for the task (1)

$$L = \sum_{i=1}^{L}\sum_{j=1}^{L} c_i c_j \rho(z_i, z_j) + \lambda\left(\sum_{i=1}^{L} c_i E(z_i^2) - \theta\right) + \mu\left(\sum_{i=1}^{L} c_i - 1\right),$$

and equal its partial derivatives to zero

$$\frac{\partial L}{\partial c_k} = 2\sum_{i=1}^{L} c_i \rho(z_i, z_k) + \lambda E(z_k^2) + \mu = 0,$$

$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{L} c_i E\left(z_i^2\right) - \theta = 0,$$

$$\frac{\partial L}{\partial \mu} = \sum_{i=1}^{L} c_i - 1 = 0.$$

Moving to a vectorial form we get

$$2DC + \lambda E + \mu I = O,$$

$$E^T C = \theta,$$

$$I^T C = 1.$$

where $D = \left\| \rho\left(z_i, z_j\right) \right\|_{n \times n}$, $E = \left\| E\left(z_i^2\right) \right\|_{1 \times n}$, $C = \left\| c_i \right\|_{1 \times n}$, $I = \left\| 1 \right\|_{1 \times n}$, $O = \left\| 0 \right\|_{1 \times n}$.

Lets denote $\alpha = E^T D^{-1} E$, $\beta = I^T D^{-1} E$, $\gamma = I^T D^{-1} I$ for short. The received equation system gets the following form

$$2\theta + \lambda\alpha + \mu\beta = 0,$$

$$2 + \lambda\beta + \mu\gamma = 0.$$

From these equations a dependence between c and θ can be derived

$$c_k = \frac{\theta\gamma - \beta}{\alpha\gamma - \beta^2} \sum_{i=1}^{L} d_{ki} E\left(z_i^2\right) + \frac{\theta\beta - \alpha}{\beta^2 - \alpha\gamma} \sum_{i=1}^{L} d_{ki} > 0,\ k = 1,\ldots,L, \tag{3}$$

where d_{ij} is an element of the D^{-1} matrix.

It must be noted also that the point c^* can be a point of strict maximum of P_f only if

$$\sum_{i=1}^{L} \sum_{j=1}^{L} \rho_{ij} \varepsilon_i \varepsilon_j > 0 \tag{4}$$

for any $\left(\varepsilon_0, \ldots, \varepsilon_L\right)$ satisfying conditions $\sum_{i=1}^{L} \varepsilon_i = 0$. Let $\theta_{\min}$ is minimal and $\theta_{\max}$ is maximal value of θ for which one of inequalities (3) becomes equality. Let $R_k = \sum_{i=1}^{L} d_{ki} E\left(z_i^2\right)$, $P_k = \sum_{i=1}^{L} \rho_{ki}$,

$$c_k = \theta\Gamma_k^1 + \Gamma_k^0$$

$$\Gamma_k^1 = \frac{\gamma R_k - \beta P_k}{\alpha\gamma - \beta^2},$$

$$\Gamma_k^0 = \frac{\alpha P_k - \beta R_k}{\alpha\gamma - \beta^2}.$$

then $P_f = \beta_0 + \beta_1\theta + \beta_2\theta^2$, where

$$\beta_0 = \sum_{i=1}^{L} \sum_{j=1}^{L} \Gamma_i^0 \Gamma_j^0 \rho_{ij},$$

$$\beta_1 = \sum_{i=1}^{L}\sum_{j=1}^{L}\left(\Gamma_i^0\Gamma_j^1 + \Gamma_i^1\Gamma_j^0\right)\rho_{ij},$$

$$\beta_2 = \sum_{i=1}^{L}\sum_{j=1}^{L}\Gamma_i^1\Gamma_j^1\rho_{ij}.$$

It is easy to show that

$$K\left(Y, Z_{ccp}\right) = \kappa(\theta) = \frac{1}{\sqrt{V(Y)}}\frac{\theta}{\sqrt{\beta_0 + \beta_1\theta + \beta_2\theta^2}}.$$

Theorem 2. Simultaneous correctness of inequalities $\theta_{\min} < \frac{2\beta_0}{1-\beta_1} < \theta_{\max}$, $\kappa\left(\frac{2\beta_0}{1-\beta_1}\right) > \kappa(\theta_{\min})$ and negativity of the condition (4) is necessary condition of irreducibility of predictors set $z_1,\ldots,z_L$.

Necessary conditions allows effectively evaluate irreducibility of predictors set. It is sufficient to calculate $\theta_{\min}$ and $\theta_{\max}$ to evaluate negativity conditions (4) and to evaluate inequalities $\theta_{\min} < \frac{2\beta_0}{1-\beta_1} < \theta_{\max}$. It is evident that in case when necessary conditions are satisfied and $\kappa\left(\frac{2\beta_0}{1-\beta_1}\right)$ for the evaluated ensemble is greater than maximal correlation coefficient for any irreducible ensemble with less predictors than the evaluated ensemble is irreducible.

Regression models based on sets of unexpandable irreducible ensembles

At the first stage initial set of reduced predictors is formed with the help of standard univariate least squares technique. Let $\tilde{Z} = \left(z_1,\ldots,z_L\right)$ is initial set of L predictors. An irreducible ensemble $\tilde{z}'$ consisting of l' predictors will be called unexpandable irreducible ensemble (UIE) if there are no irreducible ensembles in $\tilde{Z}$ with number of predictors greater l' that contain all predictors from $\tilde{z}'$. Two ways of regression model construction by sets of UIE were considered that are based on enumerating of all possible UIE. The first method chooses single best UIE where correlation coefficient of optimal Z_{ccp} with Y is maximal. This optimal Z_{ccp} ($Z_{ccp}^{\max}$) is the final regression model of the first method. The second method selects set of UIE where correlation coefficient of optimal Z_{ccp} with Y is greater than $(1-Tr)K\left(Y, Z_{ccp}^{\max}\right)$, $Tr \in (0,1)$. Thus threshold parameter Tr allows to select UIE with correlation coefficient of optimal Z_{ccp} with Y close to maximal value $K\left(Y, Z_{ccp}^{\max}\right)$. In the second method parameters of final regression models are calculated as average by all UIE with $K\left(Y, Z_{ccp}\right) > Tr * K\left(Y, Z_{ccp}^{\max}\right)$.

Method of UIE enumerating is based on gradual raising of predicates set meeting irreducibility condition.

Procedure 1. Process subset of predictors $Z = \left(z_{i_1},\ldots,z_{i_t}\right)$.

Step 1. Using Theorem 2 check whether Z is irreducible.

Step 2. Calculate $\left(c_{i_1},\ldots,c_{i_t}\right)$ and $K(Y,Z)$.

Step 3. If $K(Y,Z) > K^*$, where K^* is the previous best result, replace best subset $Z_{ccp}^{\max}$ with Z and set $K^* = K(Y,Z)$.

Step 4 (second method only). Store Z in historic list for voting purposes.

Procedure 2. Main algorithm.

Step 1. Enumerate all pairs of predictors (z_i, z_j), apply Procedure 1. If (z_i, z_j) is irreducible, store it in pairs dictionary and in list of candidates.

Step 2. Enumerate all current candidates $Z = (z_{i_1}, \ldots, z_{i_t})$, enumerate all pairs from dictionary, beginning with $z_{i_t} : (z_{i_t}, z_k)$. Apply Procedure 1 to the subset $Z' = (z_{i_1}, \ldots, z_{i_t}, z_k)$. If it is irreducible, store it in next level candidates.

Step 3. If there are any next level candidates, go to Step 2. Otherwise stop and return current $Z_{ccp}^{\max}$ (and historic list).

Step 4 (second method only). Filter historic list based on K^* and Tr and average coefficients over all remaining combinations. Let's consider a set of combinations produced by the algorithm: $\{Z_1, \ldots, Z_p\}$, where $Z_t = \sum_{i=1}^{L} c_i^t z_i$. The final predictor $Z = \sum_{i=1}^{L} \frac{1}{p} \left(\sum_{j=1}^{p} c_i^j \right) z_i$.

CPPCMM modifications

First of all, lets state that only second method, i.e. voting over some set of best combinations, is considered as proved to be better in experiments.

A new set of experiments performed for the sake of this article has revealed a major drawback of the described method. Significant time consuming in cases of larger dimensions was accompanied by memory exhaustion. Thus, strict UIE enumerating demanded additional branch reducing:

Procedure 3. Reduced main algorithm.

Step 1. Enumerate all pairs of predictors (z_i, z_j), apply Procedure 1. If (z_i, z_j) is irreducible, store it in pairs dictionary and in list of candidates.

Step 2. Consider level l. Enumerate all current candidates $Z = (z_{i_1}, \ldots, z_{i_t})$, enumerate all pairs from dictionary, beginning with $z_{i_t} : (z_{i_t}, z_k)$. Apply Procedure 1 to the subset $Z' = (z_{i_1}, \ldots, z_{i_t}, z_k)$. If $K(Y,Z') > K_{l+1}^*$, where K_{l+1}^* is the previous best result of $l+1$ level, set $K_{l+1}^* = K(Y,Z')$. If Z' is irreducible and $K(Y,Z') > K_l^*$, store it in $l+1$ level candidates.

Step 3. If there are any next level candidates, go to Step 2. Otherwise stop and return current $Z_{ccp}^{\max}$ (and historic list).

Step 4. Filter historic list based on K^* and Tr and average coefficients over all remaining combinations.

The proposed correction although provided giant boost in time and memory saving, slightly dropped overall forecasting quality. The next two modifications are aimed to its correction.

Definition 2. A predictor z_i is dominating z_j if $K(Y,z_i) \geq aK(Y,z_i) + bK(Y,z_j)$ for all $a,b>0$, $a+b=1$.

Theorem 3. A predictor z_i is dominating z_j if $\dfrac{V(z_j)^2 - V(z_i)V(z_j) - V(z_j)\rho_{ij}}{\left(V(z_i)-V(z_j)\right)^2 - \left(V(z_i)+V(z_j)\right)\rho_{ij}} > 1$.

The second modification consists in removing all dominated predictors from voting according to Theorem 3.

Third modification is weighting votes of different predictors in Step 4: $Z = \sum_{i=1}^{L}\left(\sum_{j=1}^{p} w_j c_i^j\right) z_i$, where $w_k \geq 0$ and $\sum_{k=1}^{p} w_k = 1$.

In case of no domination filter applied the weights are calculated simply in proportion to $\dfrac{1}{\left(1-K\left(Y,Z_i\right)^2\right)}$.

Otherwise they are more complicated. Let $w_{ij} = \dfrac{\left(\rho_{ij} + e_j - e_i\right)}{2\rho_{ij}p}$ and consequently $w_i = \sum_{j=1}^{p} w_{ij}$. Again, normalization is applied to satisfy $\sum_{k=1}^{p} w_k = 1$ condition.

With that last modification, the parameter Tr (threshold) described in previous section, although planned as close to zero, proved to be more efficient when close to 1 (see experiments).

Experiments

In all studies dependent variable Y and regression variables X are stochastic functions of 3 latent variables U_1, U_2, U_3. The vector levels of variables U are independently distributed multivariate normal with mean 0 and standard deviation 1. The value of dependent variable Y in j-th case is generated by formula $y_j = \sum_{k=1}^{3} u_{jk} + e_y^j$ where u_{jk} is a value of the latent variable U_k, e_y^j is a random error term distributed $N\left(0, d_y\right)$. At that 85% of cases were generated with $d_y = 1$, 15% of cases were generated with $d_y = 2$ or $d_y = 2.5$. That is how main and noisy components of data were formed. The values of relevant variable X_i were generated by binary vector $\beta^i = \left\{\beta_1^i, \beta_2^i, \beta_3^i\right\}$. In j-th case $x_{jk} = \sum_{k=1}^{3} u_{jk}\beta_k^i + e_{xi}^j$, where u_{jk} is a value of the latent variable U_k, e_{xi}^j is a random error term distributed $N\left(0, d_{xi}\right)$. In the following experiments relevant variables were generated according $d_{xi} = 0.5$. The levels of irrelevant variable X_i in j-th case are generated by formula $x_{jk} = e_{xi}^j$.

In each experiment 100 pairs of data sets were calculated by the random numbers generator according to the same scenario. Each pair includes training set that was used for optimal regression model construction and control data set that was used to evaluate prognostic ability of this model. In all experiments relevant variables were generated at $\beta = \{1,1,0\}$, $\beta = \{1,0,1\}$, $\beta = \{0,1,1\}$. In Table 1 there are other parameters of the test samples described.

Table 1. Experiment sample series.

Task	Number of objects	Number of features	...of them irrelevant	Noize coefficient d_y
data1	30	120	70	2.0
data2	30	120	70	2.5
data3	30	100	50	2.5
data4	30	140	90	2.0
data6	20	160	85	2.0
data7	20	160	85	2.5
data8	15	150	85	2.1
data9	40	150	81	2.5

First, the described data was used for the threshold parameter Tr impact study. The following two graphs show the dependency between resulting forecast correlation and the parameter. Here and further on an average values over 100 independent control tasks are shown.

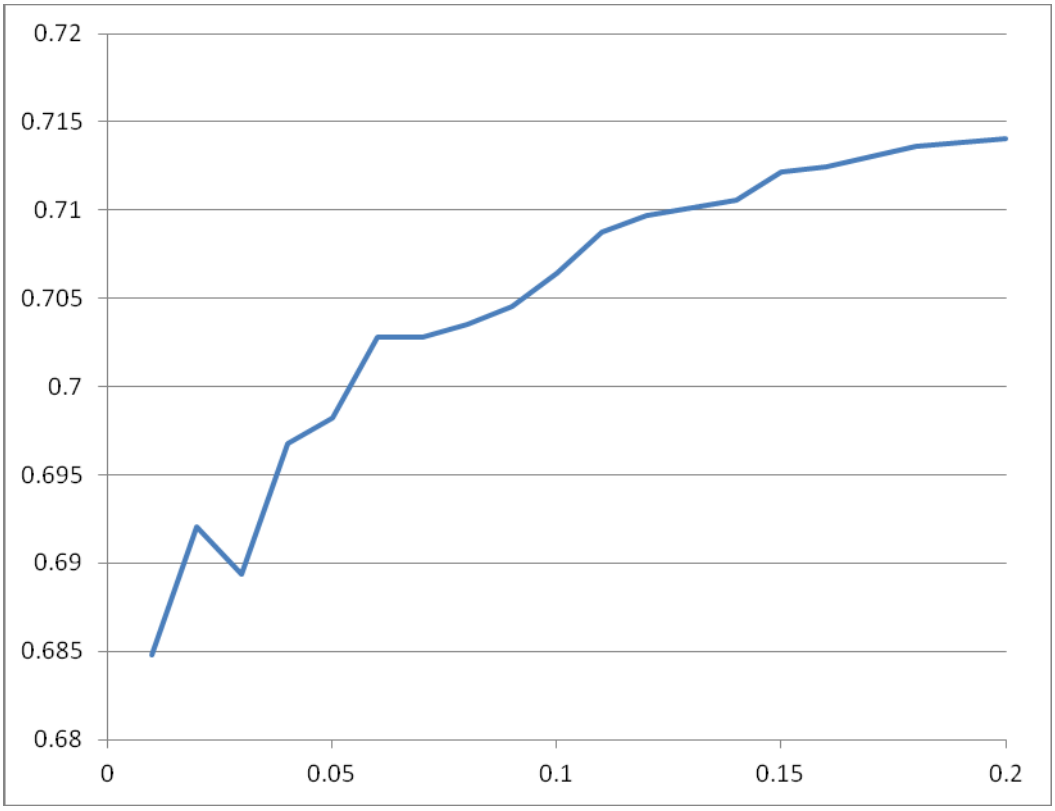


Fig. 1. Data7 test sample, threshold range 0.1–0.2, step 0.01.

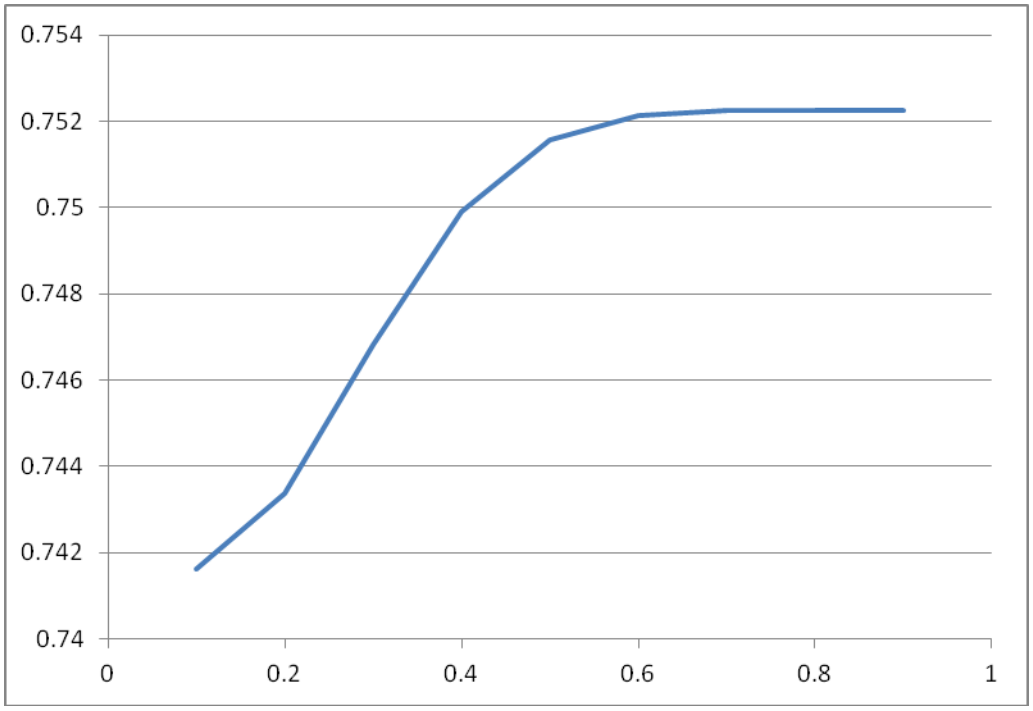


Fig. 2. Data8 test sample, threshold range 0.1–0.9, step 0.1.

The clear and most unexpected result of this and other similar experiments is that better result are achieved at threshold values close to 1. It means that every tested irreducible combination is important for the resulting weighted sum. Furthermore, the dependency is quite monotonic and thus threshold in every comparative experiment can be set to 1.

Finally, the second set of experiments shows comparison of the proposed method to Glmnet. It need to be mentioned that Glmnet for Matlab also has some parameters. Thus, to make results more undoubted its optimization was performed, so all tables and graphs contain its best result over parameters grid.

Table 2. Results of expiriments. Prognostic ability.

Task	CCPCMM's correlation	Glmnet for Matlab	
		Correlation	Optimal parameter λ
data1	0.776	0.763	0.55
data2	0.746	0.726	0.5
data3	0.741	0.722	0.55
data4	0.752	0.739	0.55
data6	0.768	0.736	0.55
data7	0.728	0.691	0.75
data8	0.752	0.7135	0.75
data9	0.732	0.711	0.57

The same results are shown on the following graph.

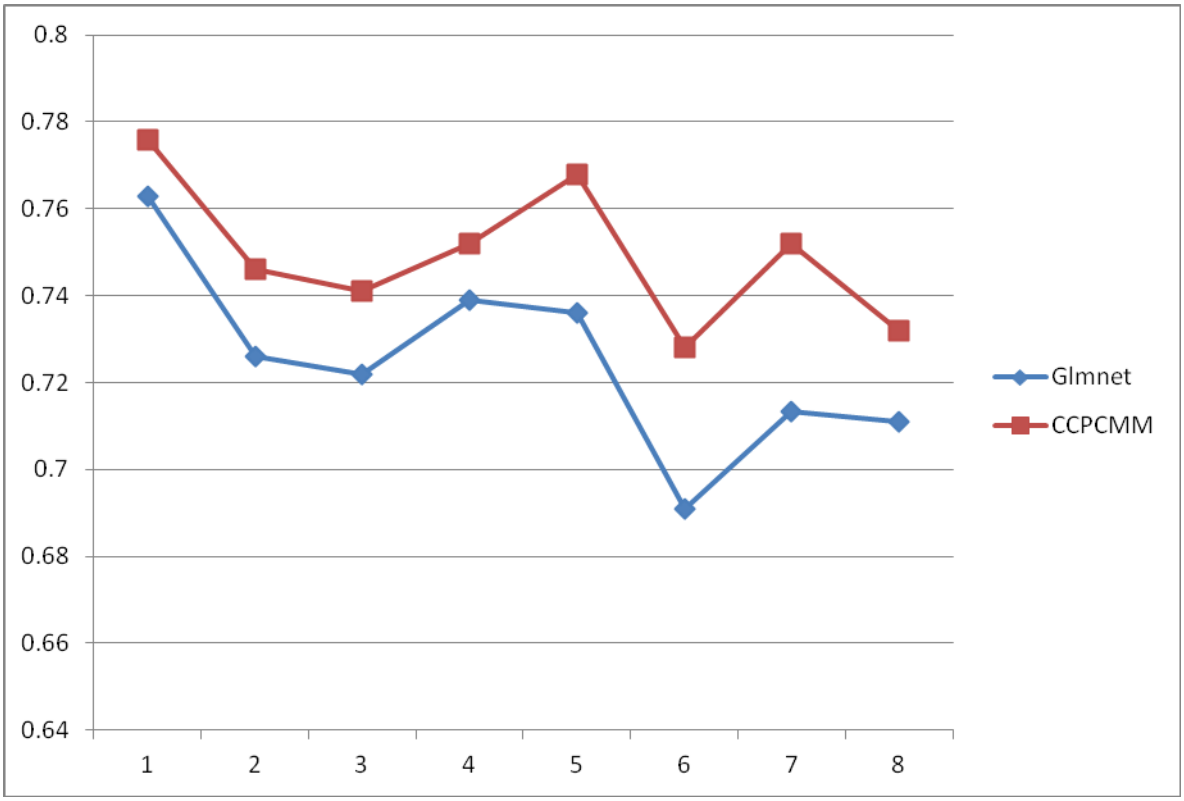


Fig. 3. Correlation in test tasks.

Conclusion

Some modifications of the novel regression method are described, which correct its time and memory consuming as well as forecasting quality. The results shown in figures 1 and 2 exclude any parameters from the training process, which made it suitable for unsupervised use. Moreover, the results shown in table 2 and figure 3 clearly show its superiority comparing to well known and widely acknowledged regression tool.

Thus, the modified method can be recommended for a wide range of forecasting applications, especially in automatic unsupervised applications.

Bibliography

[Efron et al., 2004] B. Efron, T. Hastie, I. Jonnstone and R. Tibshirani. Least Angle Regression. Annals of Statistics. 2004, Vol. 32, No. 2, 407–499.

[Tibshirani, 1996] Tibshirani R., Regression shrinkage and selection via the lasso // J. Roy. Stat. Soc. 1996. Vol. 58, p. 267–288.

[Breiman, 1999] L. Breiman, Random forests - random features. Technical report 567. Statistics department. University of California, Berkley, September 1999 // www.boosting.org.

[Kuncheva, 2004] L.I. Kuncheva, Combining Pattern Classifiers. Methods and Algorithms. Wiley Interscience, New Jersey, 2004.

[Zhuravlev et al., 2008] Zhuravlev Yu.I., Kuznetsova A.V., Ryazanov V.V., Senko O.V., Botvin M.A., The Use of Pattern Recognition Methods in Tasks of Biomedical Diagnostics and Forecasting // Pattern Recognition and Image Analysis, MAIK Nauka/Interperiodica. 2008, Vol. 18, No. 2, pp. 195–200.

[Zhuravlev et al., 2006] Zhuravlev Yi.I., Ryazanov V.V., Senko O.V., RECOGNITION. Mathematical methods. Program System. Applications. —Moscow: Phasiz, 2006, (in Russian).

[Kuznetsov et al., 1996] Kuznetsov V.A., Senko O.V. et all., Recognition of fuzzy systems by method of statistically weighed syndromes and its using for immunological and hematological norm and chronic pathology // Chemical Physics, 1996, v. 15, N 1, p. 81–100.

[Brown et al., 2005] Gavin Brown, Jeremy L. Wyatt, Peter Tino, Managing Diversity in Regression Ensembles. Journal of Machine Learning Research 6: 1621-1650. 2005.

[Krogh et al., 1995] A. Krogh and J. Vedelsby, Neural network ensembles, cross validation, and active learning. NIPS, 7:231–238, 1995.

[Senko, 2004] Senko Oleg V., The Use of Collective Method for Improvement of Regression Modeling Stability // InterStat. Statistics on the Internet http://statjournals.net/, June, 2004.

[Senko, 2009] O.V. Senko,. An Optimal Ensemble of Predictors in Convex Correcting Procedures // Pattern Recognition and Image Analysis, MAIK Nauka/Interperiodica. 2009, Vol. 19, No. 3, pp. 465–468.

[Senko et al., 2010] Senko O., Dokukin A. Optimal Forecasting Based on Convex Correcting Procedures.// New Trends in Classification and Data Mining -ITHEA, Sofia, Bulgaria, 2010, p. 62-72.

[Senko et al., 2011] Senko O., Dokukin A. Correlation Maximization in Regression Models Based on Convex Combinations // International Journal “Information Theories and Applications”, Vol. 18, Number 3, 2011, P. 224-231.

[Friedman et al., 2007] Jerome Friedman, Trevor Hastie, Holger Hofling, Robert Tibshirani. Pathwise coordinate optimization, The Annals of Applied Statistics. Volume 1, Number 2 (2007), 302-332.

[Friedman et al., 2010] Jerome Friedman, Trevor Hastie, Rob Tibshirani. Regularized Paths for Generalized Linear Models via Coordinate Descent // Journal of Statistical Software 33(1), 2010.

Acknowledgements

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Authors' Information

Oleg Senko *– CCAS, chief researcher, 119333, Vavilova Str. 40, Moscow, Russian Federation; e-mail: senkoov@mail.ru*

Major Fields of Scientific Research: Pattern Recognition, Data Mining

Alexander Dokukin *– CCAS, researcher, 119333, Vavilova Str. 40, Moscow, Russian Federation; e-mail: dalex@ccas.ru*

Major Fields of Scientific Research: Pattern Recognition, Data Mining

CORRELATION-BASED PASSWORD GENERATION FROM FINGERPRINTS

Gurgen Khachatrian, Hovik Khasikyan

***Abstract**: In this paper, a methodology for reliable password generation from fingerprints is developed. In contrast to traditional biometric systems, proposed algorithm does not authenticate user by matching his or her biometrics. Reference data gives no information about the password and fingerprint. In hand with cryptography, this method can provide highly secure protection for cryptographic keys used in Digital Signatures and Digital Rights Management systems.*

***Keywords**: Password Generation, Confidentiality, Authentication, Privacy, Security, Fingerprints, Image Processing, Pattern Recognition, Template Matching.*

***ACM Classification Keywords**: D.4.6 Security and Protection (K.6.5)*

Introduction

In most cryptographic applications, secret keys such as private keys in public key systems or symmetric keys are protected by passwords. Secret keys are stored on the computer and encrypted by hashed password. They are released for user authentication by entering a proper password by the user.

However, conventional passwords are relatively simple and sometimes easy to guess or to break. People remember only short passwords. What is more, they tend to choose passwords, which are easily cracked by dictionary attacks [1, 2, 3].

Biometric technologies provide alternative solution to this problem. In traditional biometric systems user is authenticated by matching his or her biometrics. In these systems, there is no secret key, which can be guessed or lost. Instead of decrypting the secret key with a password, system performs matching of input biometrics with corresponding biometrics stored in the database. If matching is successful, the secret key is released.

Nonetheless, biometric systems have their vulnerabilities [4]. The main weakness of these systems is in their design. Since biometric data is stored locally (on smart card or another computer), possession of the smart card or access to the database gives access to the biometric template of the users. This gives attacker an opportunity to break current and all the other authentication systems using the same biometric identifier. What is more, authentication process in this design is completely decoupled from the key release and outputs only one bit accept/reject decision. This makes the system vulnerable against Trojan horse attacks (which can simply overwrite the decision bit).

Based on the above observations, in this paper a password generation method is proposed from fingerprints. The generated passwords are not stored on any device. Only a small reference data should be kept in database or on smart card with all the other credentials of the user. Reference data gives no information about the password. Thus, the passwords cannot be guessed, lost, shared or stolen.

The accuracy of the system is evaluated on a large database of fingerprints and gives promising results for the use in commercial authentication, where very low False Rejection Rate (FRR) is required for a given False Acceptance Rate (FAR).

Previous Works

The proposed approach is inspired in particular by the "Generation of Secret Key from Fingerprint Image" of Maslennikov [5]. In order to generate keys from fingerprint images in [5] a correlation based method was used, which is described in detail in "A Correlation-Based Fingerprint Verification System" [6].

In correlation based verification, system selects appropriate template images from the primary fingerprint and then uses template-matching techniques to locate them on the secondary image. If their locations are the same as they were on the primary image then the owner is recognized as genuine (Figure 1).

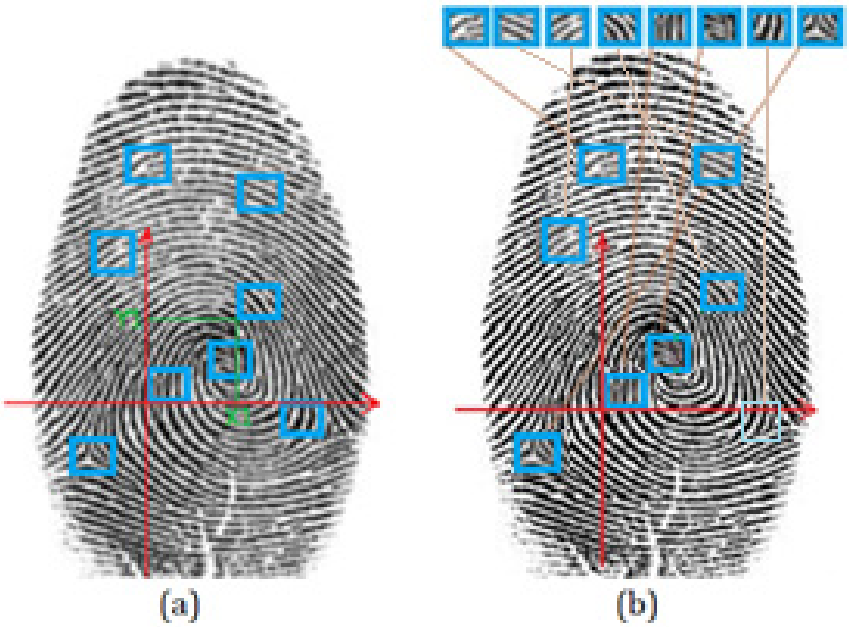


Figure 1: Correlation-based template matching. (a) primary fingerprint and chosen templates; (b) secondary fingerprint and located templates.

In [5], these locations were used to generate a cryptographic key. Having all the templates located on the secondary image, system determined coordinates of the aligned templates, considering the first template as a starting point in the coordinate system. Coordinates are in the form: X1, Y1, X2, Y2, X3, Y3 …, where X1 shows the horizontal distance between the first template and the second one, Y1 is the vertical distance between the first template and the second one. X2 is the horizontal distance between the first and third templates and so on. Using these coordinates and some additional data (e.g. passwords), in [5] system generated cryptographic key.

The weaknesses of the system were high FRR and long execution time. To be authenticated user had to scan his/her finger 5 to 7 times [5]. These factors made the system uncomfortable and not practical.

Fingerprint Processing

Fingerprint is one of the noisiest image types. In order to reduce the noise and enhance useful information, first Gaussian Smoothing [7] algorithm is applied to mitigate noisy points on fingerprint image, and then the image is binarized using adaptive image binarization techniques [8].

Using traditional threshold based methods to convert gray scale fingerprint image into black white shows low accuracy. The reason is that different points of finger are pressed with different strength on the screen of scanner. Results of binarization with different threshold values are illustrated in Figure 2.

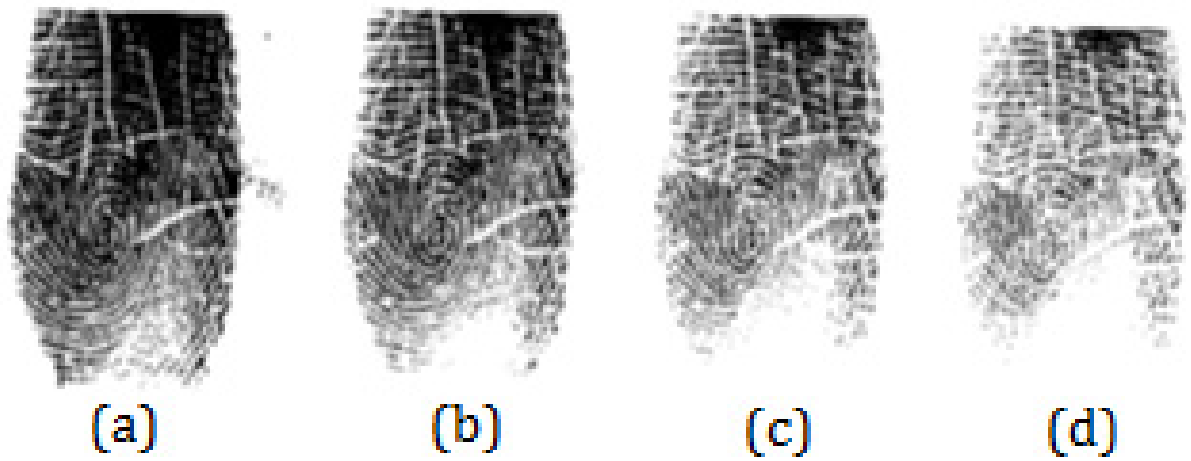


Figure 2: Results of binarization with different threshold values (threshold value increases from left to right).

From Figure 2 it can be seen, that for a small threshold values valleys at the upper part become too dark. For bigger threshold values, ridges at the bottom disappear. This problem can be solved using adaptive binarization algorithm. In this case there is no universal threshold value for whole image, but for each pixel its own threshold value is calculated separately. For each pixel a square with specific sizes is chosen, where the total sum of pixel intensity is calculated. If the gray value of the pixel is greater than threshold, then it is set to white, otherwise it is black. This method is much more accurate, if the size of the square for threshold calculation is selected appropriately. Analyses have shown that the best size of the square is different for different fingers. The best results are obtained when block size is a little bigger than twice of the thickness of the ridge. This can be explained with the fact, that the square of this size is highly probable to contain equal quantities of black and white pixels. In such case, it works better than histogram correction algorithms. The result of adaptive binarization with correct block size is illustrated in Figure 3.

Figure 3: Results of adaptive

Shapes of the Reference Images

The shape of reference images is very important for the accuracy of the system. In the previous works square templates were used in template matching, but because of physical features of fingers these templates do not show very accurate results. Analyses on the local database have shown, that skin of the fingers is deformed mainly in horizontal axis, whereas on the vertical axis the distortion is minimal. Because of these concerns, the local database was analyses for all possible rectangular patterns.

The most optimal shape of the templates was found to be a wide rectangle with its length equal two to four times height. This ratio is justified with the previously stated explanation of shape distortions. All the further calculations in this work are implemented for the best rectangular shape for the whole database. However, the best shape of templates is individual for each finger and can be found by distinct analysis.

Template Selection

The template selection process is the least attractive part of correlation-based verification. The number of operations is very high and makes the system very slow.

The quality of a template is considered high, if it is located on the secondary image easily and precisely. In other words the template should fit as well as possible at the same location, but as badly as possible at the other locations. This feature is the same as the uniqueness of the template. To count the uniqueness of the template, template is compared with all the other templates on image pixel wise.

In this work a novel template selection methodology is proposed, based on specific features of the templates with high uniqueness. Analyses show that for the unique templates there is specific distribution of similar patterns. The most similar templates, which are deterministic for counting the final uniqueness of the template, are located close to the original. Figure 4 illustrates this distribution.

The yellow square shows the location of the original template. Red circles show locations of the first four closest templates (by Hamming distance) to the original one. The green triangles are next 5 to 10 templates, blue points 11 to 20. This was the general way of distribution for the majority of fingerprints of the local database. Nevertheless, this is not a rule. For some fingerprints very unique templates are found, for which very similar patterns were located far from it (Figure 5).

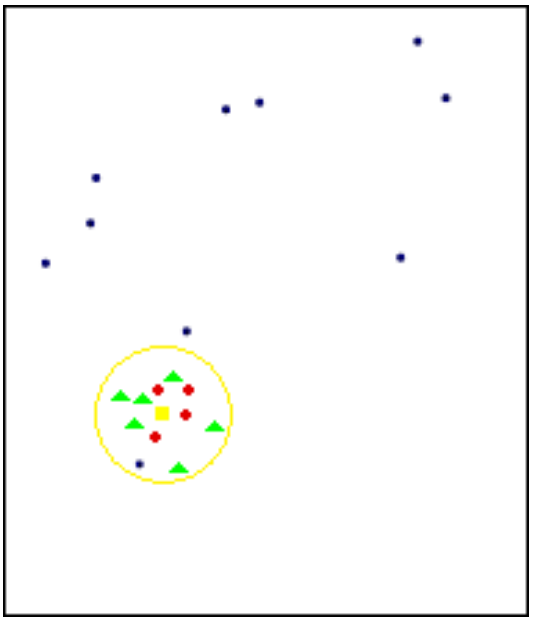


Figure 4: Example of similar pattern distribution.

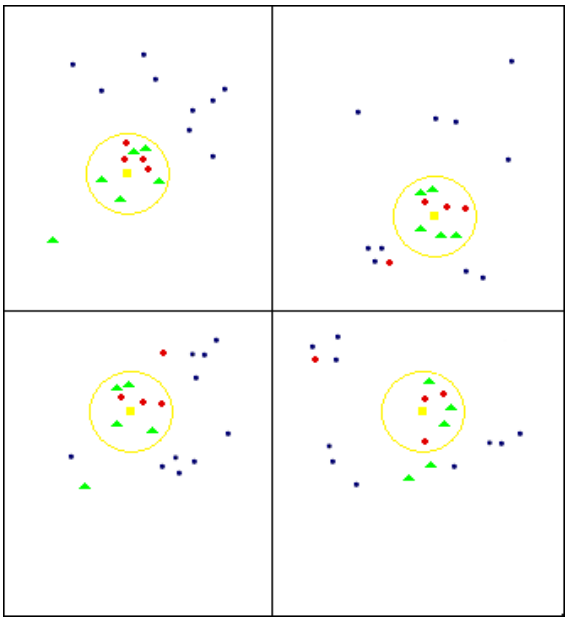


Figure 5: Distribution of the similar patterns for the most unique templates.

To use this novel characteristic some special measures are required in our algorithm. First, for each template its uniqueness is calculated by comparing it with its neighbors. Then these templates are sorted by uniqueness value. After sorting, the most unique templates are chosen and verified for validity. The decision is reached by aligning new templates on the image and locating the closest templates. If algorithm finds very close template far from the original, the template is treated as invalid. It is removed to another place in the queue, according to its new uniqueness value. The next template by uniqueness is considered as candidate and the same procedures are implemented until all the required templates are found.

One more observation was carried out, that when counting the uniqueness of a template, the neighbor templates (in two pixel distance) should not be considered. These templates are very similar to the original and have misleading effect, when counting the uniqueness of the template. Furthermore, since approximations should be made with the coordinates of the templates, there is no difference between the original template and the closest neighbors.

Template Matching

For faster template matching it is important to exclude comparisons with obviously dissimilar templates. In this work it is proposed to use the first lines of the reference image and the candidate template to make a decision. The first lines are used to find out whether two templates have anything in common or the system should omit this candidate and continue pixel wise search. The further comparisons are permeated or cancelled based on the correspondence of these lines and on the predefined threshold value. The threshold value should be about 70%; not more, otherwise it affects accuracy. For higher confidence, the threshold value can be decreased down to 55%. For the threshold value less than 50% there is no acceleration at all, because practically all candidate templates satisfy this requirement.

Password Generation

In this paper an approximation method is proposed for reliable password generation from the extracted coordinates. Analyses show that for the fingerprint image with sizes 240x280 these coordinates waive 0 to 5 pixels from their original locations (Table 1).

	X1	Y1	X2	Y2	X3	Y3	X4	Y4	X5	Y5	X6	Y6	X7	Y7
Original	72	-27	-25	47	-36	110	-73	46	-58	-113	-29	77	-12	-70
Imprint1:	72	-27	-24	48	-36	111	-72	50	-58	-112	-28	79	-12	-71
Imprint2:	73	-27	-27	46	-37	109	-75	45	-59	-114	-32	76	-12	-70
Imprint3:	72	-27	-26	47	-37	110	-75	47	-60	-113	-32	77	-13	-70
Imprint4:	72	-27	-25	47	-36	110	-74	46	-59	-113	-30	77	-12	-70
Imprint5:	72	-27	-25	47	-36	110	-73	46	-59	-113	-29	76	-13	-70
Imprint6:	72	-27	-23	47	-35	110	-71	46	-58	-113	-26	78	-12	-70
Imprint7:	72	-27	-26	48	-36	110	-73	49	-59	-112	-30	78	-12	-69

Table 1: Coordinates of the localized templates

Thus, with precision of five pixels, the final result will be exact. Therefore, ([X0- Xn] mod 10) is kept in the database for each of the reference images. In the example at the table 1

$\Delta X1 = ([X0- X1] \bmod 10) = 2$, $\Delta Y1 = ([Y0- Y1] \bmod 10) = 3$, $\Delta X2 = ([X0- X2] \bmod 10) = 5$ etcetera.

Testing and Analysis

All the tastings and analyzes are performed on the local database of fingerprints. The database consists totally of flat (dab) impressions. There are 320 fingerprints of 40 different persons (8 imprints of index fingers from each person). The images have been captured using U.are.U 4500 optical fingerprint reader [9]. All fingerprint images are recorded at 512 dpi and as 256 gray tone images (8-bit grayscale). After scanning images are resized to 240x280 pixels and stored in the database. The fingers have been pre-scanned to insure a representative mix of varying quality impressions, ranging from those of poor quality to those of excellent quality (Figure 7).

Figure 7: Examples of fingerprints from the local database

The experimental demonstrate that the proposed algorithm has acceptable accuracy for the use in commercial authentication (where very low False Rejection Rate is required for a given False Acceptance Rate). The FRR of the algorithm equals 3.35%, while FAR is less than 0.1%.

Entropy

Entropy of the system is important to avoid generating the same passwords for different users. To demonstrate the distribution of the coordinate values, for each coordinate X1, Y1, X2… the quantity of accepted values is counted. Figure 8 illustrates this distribution for the first position.

Analyses have shown that there is a dangerous peak between -3 to 1. The reason for accumulations is that the second template by its uniqueness, was mostly located very close to the first one. The third and fourth templates are also presupposed to be close, so their values were between -5 to 5.

These accumulations made the system vulnerable, because attacker could use the most favorable values and break the system after some iteration of the most favorable values.

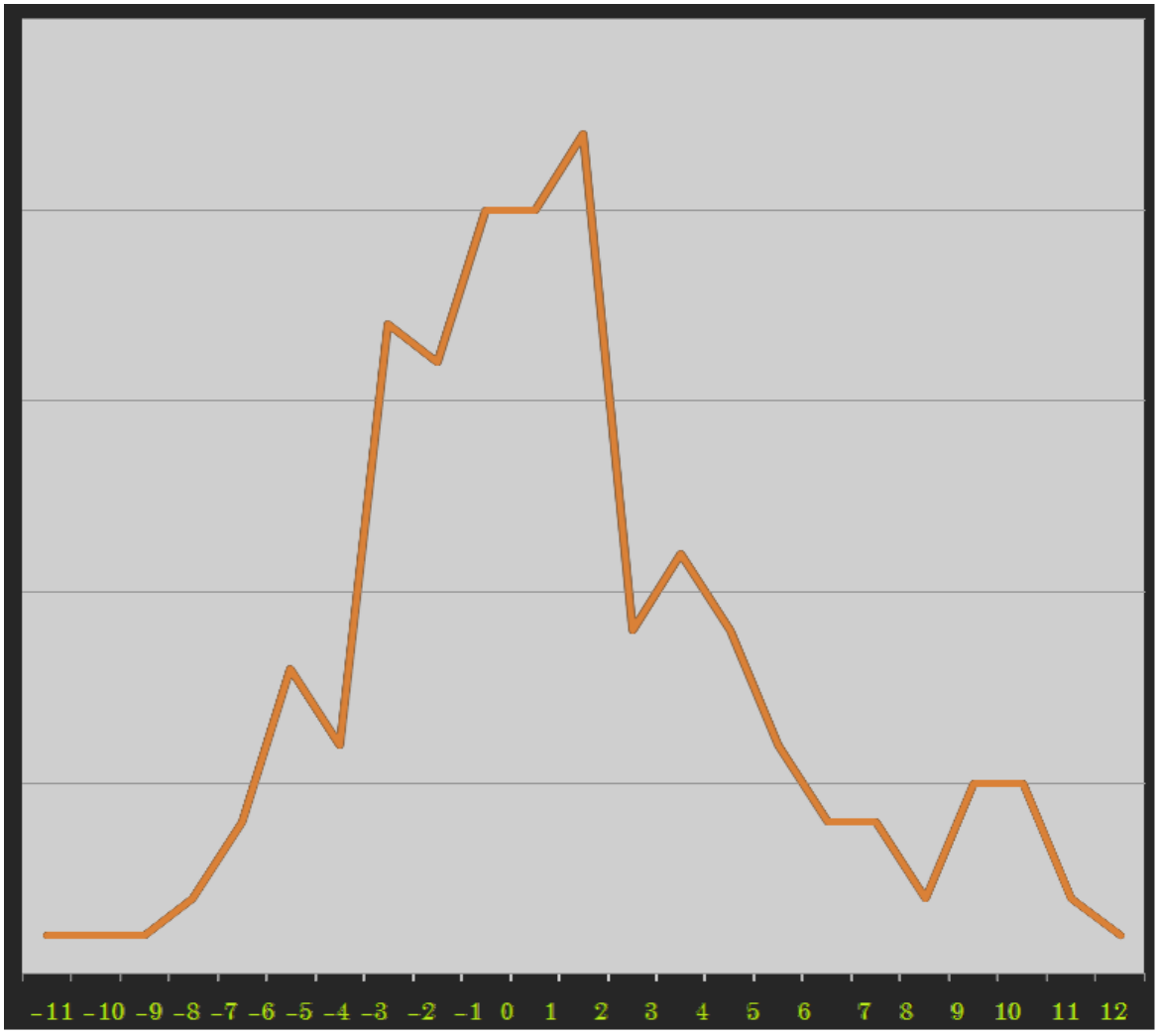


Figure 8: Accumulations of the coordinate values.

In order to handle this problem permutations of the reference templates have been introduced at the enrollment phase. After locating all required templates, their indexes are randomly shuffled, so the most unique template is stored from the first to the last positions randomly. The result of this shuffles on the distribution are depicted in the Figure 9.

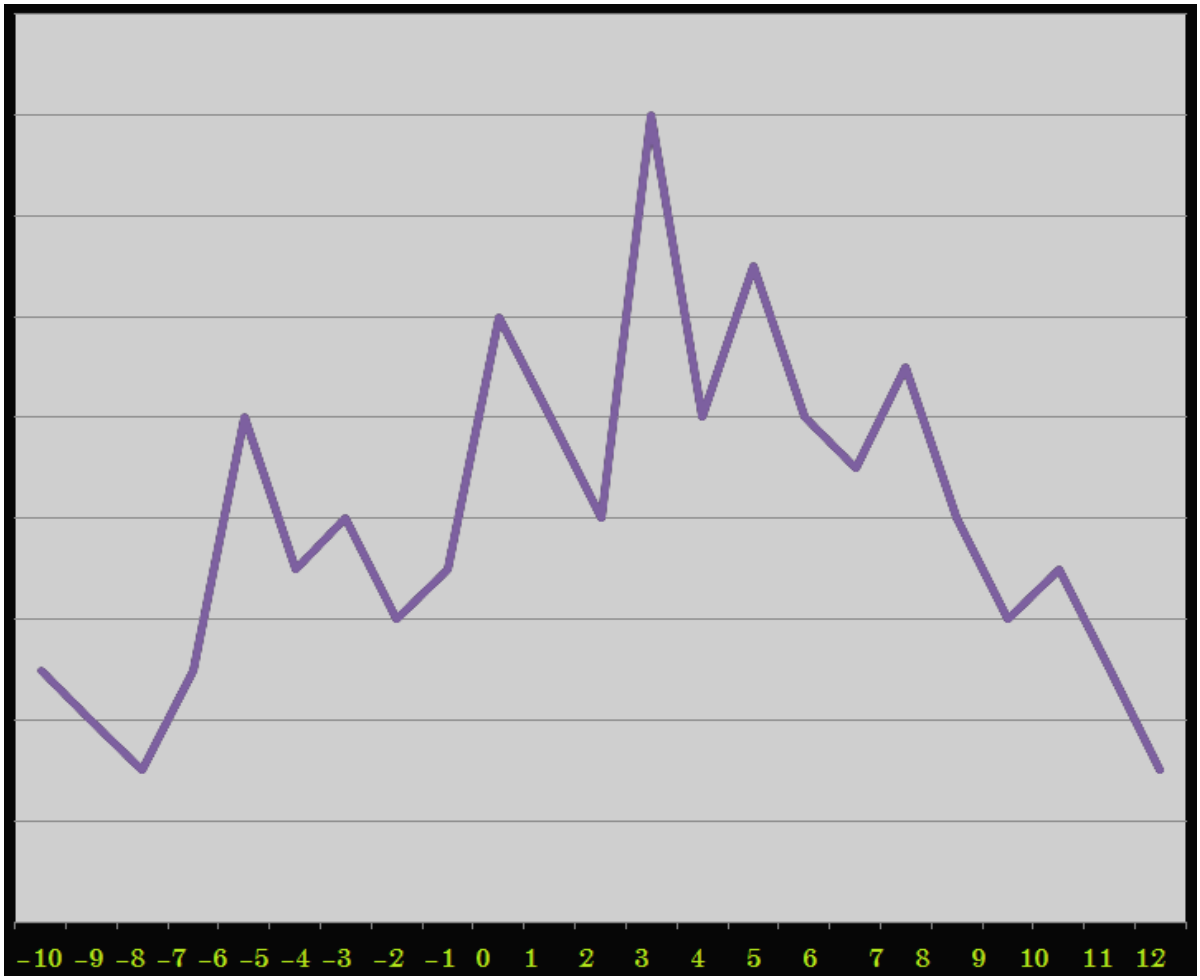


Figure 9: Accumulations after permuting the templates.

Security Analyses

The security of the system is analyzed from two points of view. At first, it was found that the proposed system has fairly low FAR (which can be made as low as 0.01%). This is due to two-stepped verification; first, system aligns templates on the secondary image and counts the accuracy of the fitting. If this value is less than predefined threshold value, this person is not authenticated. In the second phase algorithm generates a password based on the template locations. These passwords should be exact; otherwise, the user is again rejected.

Another important security measure is the size of the generated password. In this work, the size of the password is 84 bits. After locating all templates (in this work it is eight templates), system generates seven X and seven Y coordinates. The first template is the starting point in the coordinate system. Each coordinate can waive between

-24 to 24, thus each requires 6 bit memory. The size of the password equals 14*6 = 84 bit, which is the smallest general purpose-level key size [10].

The size of the password can be increased by keeping more templates, but this affects the accuracy of the system. The dependency of the error from the number of the templates is illustrated in Figure 10 (the best number of templates is found to be 8).

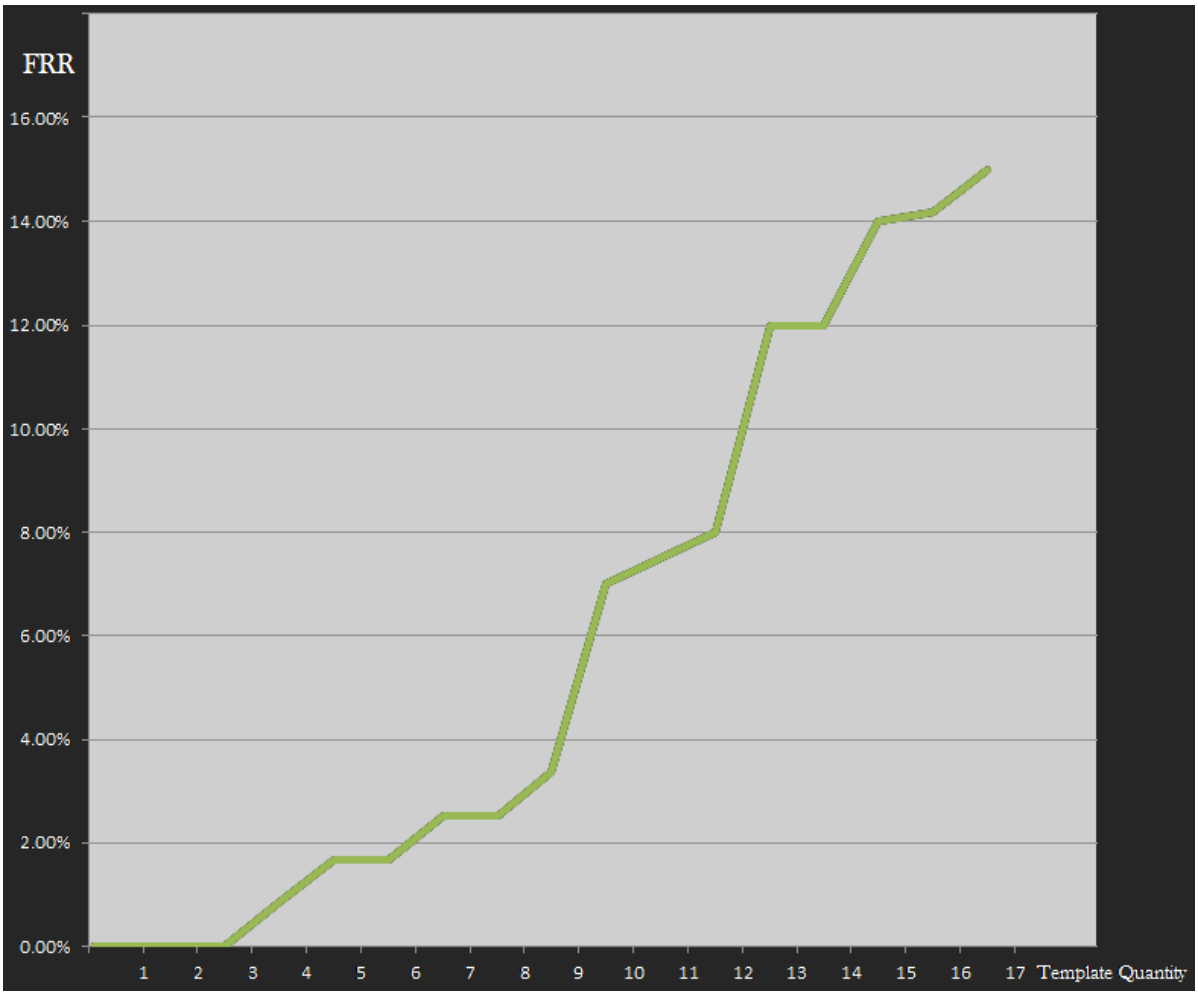


Figure 10: Error dependency from the template quantity.

For generating longer passwords template matching and coordinate approximation techniques should be made much more sensitive. To further enhance the security of the system, biometric passwords can be combined with other additional information (for example passwords or passphrases).

The problem of the fingerprint faking is also considered. In order to take over these issues only registered scanners should be used in the system, which can ensure if the fingerprint is covered with artificial layer or not (e.g. performing skin distortion [11] and odor analyses [12]).

Comparisons with Other Reference Systems

The advantage of the proposed method is that biometric data is not stored locally. Stealing the reference data (or smart card) becomes meaningless as it does not give access to the secret key and password without the finger of legitimate user. Trojan-horse attacks are also excluded, since algorithm makes decryption of the encrypted secret key.

In contrast to minutiae-based systems, this method demonstrates higher reliability in the field of password generation. In the minutiae-based systems, despite the fact, that only reliable minutia can be chosen as a source for password generation [13, 14] the probability that some of minutiae will not be observed or wrong minutiae will be extracted is still high. Unlike minutiae-based techniques, this method is able to handle low-quality images with missing and spurious minutiae.

The main disadvantage of the previous works was the demand for high computational power and low accuracy. For an image with sizes 240x280 and template's size 30x30 pixels, (240-30)*(280-30)*30*30=47*106 XOR operations were required to locate a template on the secondary image. As such, the enrollment phase required 210*250*(47*106) = 2.5*1012 XOR operations, and the verification phase 8*(47*106) = 378*106 XOR operations.

In the current work, enrollment requires 2.5 to 6*109 XOR processes for the same fingerprint (500 times faster). Verification is also improved and takes 18 to 23*106 XOR operations (16 times faster).

The accuracy and sensitivity of the system are also improved; FRR equals 3.35%, FAR is 0.01% (the EER of previous system was 7.98% [5, 6]). These improvements are the results of better template shape selection and more accurate enrollment and verification procedures.

Conclusion

The main target of this work was to develop a reliable password generation algorithm. Passwords should have been generated exactly the same each time user was verified. Nevertheless, the consistency of the passwords was one of the primary challenges.

The entropy of the generated passwords is also analyzed and is enough to resist brute force attacks. The result of analyses can be different for a very large database of fingerprints, but as these biometric passwords are used for encryption of a randomly generated secret key, a little change in entropy cannot affect the security of the system.

The security of the passwords' size is considered to be the smallest general-purpose level (84 bit key provides long-term protection against small organizations). However, it can provide only short-term protection against agencies [10]. The size of the password can be increased by processing another biometrics (e.g. palm prints) and by developing more sensitive averaging and alignment algorithms.

Acknowledgement

This research has been carried within the project *Application of Security to Biometrics and Communications*, sponsored by the Volkswagen Foundation.

Bibliography

[1] E. Spafford. Observations on reusable password choices. In Proceedings of the3rd USENIX Security Symposium, September 1992.

[2] T. Wu. A real-world analysis of Kerberos password security. In Proceedings of the 1999 Network and Distributed System Security Symposium, February 1999.

[3] R. Morris and K. Thompson. Password security: A case history. Communications of the ACM, 22(11):594–597, November 1979.

[4] Stavroulakis, P., Stamp, M.: Handbook of Information and Communication Security. Springer, Heidelberg (2010).

[5] M. Maslennikov, Practical Cryptography, Saint Petersburg, (2003).

[6] A.M. Bazen, G.T.B. Verwaaijen, S.H. Gerez, L.P.J. Veelenturf, B.J. van der Zwaag: A correlation-based fingerprint verification system, 11th Annual Workshop on Circuits Systems and Signal Processing (2000).

[7] R. Deriche, Recursively implementing the Gaussian and its derivatives, V. Srinivasan, Ong S.H., Ang Y.H. (Eds.), Proc. Second Int. Singapore Conf. on Image Proc. (Singapore, Sept.7–11, 1992), 263–267, 1992.

[8] T.Romen Singh, Sudipta Roy, O.Imocha Singh, Tejmani Sinam and Kh.Manglem Singh," A New local Adaptive Thresholding Technique in Binarisation", IJCSI-Vol 8, issue 6 No. 2 pp. 271-277 (Nov, 2011).

[9] U.are.U 4500 Fingerprint Reader

http://www.digitalpersona.com/Biometrics/Hardware-Products/U-are-U-4500-Reader/4500-Reader/

[10] Cryptographic Key Length Recommendation, ECRYPT II Recommendations (2011)

http://www.keylength.com/en/3/

[11] Antonelli, A., Cappelli, R., Maio, D., and Maltoni, D. (2006), "Fake Finger Detection by Skin Distortion Analysis," IEEE Transactions on Information Forensics and Security 1(3), 360–373 (2006).

[12] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni, "Fake fingerprint detection by odor analysis," in Proc. Int. Conf. on Biometric Authentication (ICBA06) (2006).

[13] N. J. Short, A. L. Abbott, M. S. Hsiao, and E. A. Fox, "A Bayesian Approach to Fingerprint Minutia Localization and Quality Assessment using Adaptable Templates". In Proceedings of the International Joint Conference on Biometrics, 2011.

[14] Min Wu, A. Yong, Tong Zhao and Tiande Guo, "A Systematic Algorithm for Fingerprint Image Quality Assessment". In International Joint Conference on Biometrics, 2011.

Authors' Information

Gurgen Khachatrian *– Professor, American University of Armenia, Full member of Armenian National Academy of Sciences*

e-mail: gurgenkh@aua.am

Major Fields of Scientific Research: Cryptography, Error-control coding

Hovik Khasikyan *– Researcher, American University of Armenia*

e-mail: hovik_khasikyan@edu.aua.am

Major Fields of Scientific Research: Computer Vision, Image Processing, Biometrics, Security, Privacy

SEGMENTATION BASED FINGERPRINT PORE EXTRACTION METHOD

David Asatryan, Grigor Sazhumyan

Abstract*: In this paper, an algorithm for a fingerprint closed pore extraction is proposed. A closed pore is considered as a segment of binarized fingerprint image. Segment contains maximal information about a pore shape, orientation or other significant features. The proposed algorithm is based on the consecutive performance of some simple and well known image processing procedures, namely image binarization, segmentation, inversion, whitening etc. Segmentation is a process of splitting an image into non-overlapping partitions with connected pixels of the same intensity interval. After segmentation a pore is presented as a white segment in a black background. Inversion transforms the white pore segment into a black segment. Whitening is an operation to change pixels of the segment of certain size to pixels of intensity 255. This operation deletes black pores from the inverted image. Thus we can extract all the pores by comparing the intermediate images. The proposed algorithm consists of mentioned operations applied by appropriate choosing of thresholds. An example of application of described algorithm to show the effectiveness of our approach to the pore extraction problem is given.*

Keywords*: fingerprint, closed pores, segmentation, binarization, inversion.*

ACM Classification Keywords*: Image Processing and Computer Vision*

Introduction

Fingerprint recognition is widely popular but a complex pattern recognition problem. The information contained in a fingerprint can be categorized into three different levels, namely, Level 1 (pattern), Level 2 (minutia points), and Level 3 (pores and ridge contours) [Jain, 2007]. Most existing automated fingerprint recognition systems (AFRS) utilize only level one and level two fingerprint features for personal identification [Maltoni, 2003]. Level-three fingerprint features like pores, though seldom used, are also very distinctive [Stosz, 1994]. During last decades more and more researchers are exploring how to extract and use level-three features in AFRS.

Several methods have been proposed for pore detection, extraction and matching. The pore extraction algorithm can be broadly classified into two classes: the first class of algorithms extract pores by tracing fingerprint skeletons (Stosz, 2004, Kryszczuk, 2004]), the second class of algorithms extracted pores directly from gray scale image (Jain, 2007). The review of pore extraction methods are considered in [Zhao, 2010].

In this paper, a pore extraction algorithm from the second class is proposed.

The algorithms of second type use some standard operations, namely binarization, segmentation, filtration etc. It is known that the procedures of pore extraction usually are computationally expensive. The concrete algorithm depends on the problem to be solved. In many AFRS it is enough to determine only the coordinates of detected pores to use them for fingerprint recognition purposes. However the literature analysis shows that in certain cases it can be important to extract a pore as a segment to have maximal information about a pore size, shape, orientation, mutual disposition or other features.

In this work we consider the pore extraction procedure as a special segmentation method using relatively simple operations. The procedure is based on the repeatedly application of the hierarchical segmentation algorithm created by authors earlier [Asatryan, 2007], combining it with other simple processing algorithms.

The rest of paper is organized as follows. At first we describe the hierarchical segmentation method and software system proposed in [Asatryan, 2007]. Then we introduce some operations for fingerprint image processing and closed pores extraction algorithm based on that operations. In final part of the paper we describe an experiment for pore detection in a fingerprint image.

Hierarchical Segmentation Method

We consider an image of format Gray Scale (8 bit). So image S of size $N \times M$ has pixels of intensity $w(m,n) \in \{0,1,..,255\}$, where $m = 0,1,..,M-1; \quad n = 0,1,..,N-1$.

Let $S' \in S$ be a set of pixels. A path $P(A,B)$ between two pixels $A,B \in S'$ is a sequence of $n > 1$ pixels $A, A_1, A_2,..., A_n = B$ such that any two successive pixels of the sequence are adjacent. A set of pixels $S' \in S$ is called *connected*, if for any pair of pixels $A,B \in S'$ there exists a path $P(A,B)$ such that the sequence of pixels $A, A_1, A_2,..., A_n = B$ belongs to $S' \in S$.

Let $0 < L < 255$ be an integer, and the intervals $I_1 = [0,\theta_1]$, $I_2 = [\theta_1 + 1, \theta_2]$,..., $I_{L+1} = [\theta_L + 1, 255]$ are formed by thresholds $\theta_1, \theta_2,..., \theta_L$. We say that a pixel A belongs to interval I_ℓ , if $w(A) \in I_\ell$, $\ell = 1,2,..,L+1$. We say that a set $S' \in S$ belongs to interval I_ℓ , $\ell = 1,2,..,L+1$, if all pixels of $S' \in S$ belong to the same interval I_ℓ . Then the connected set $S' \in S$ is called *segment* of intensity interval I_ℓ , if it belongs to the interval I_ℓ . So the *coherent segmentation* of image S results in the set of segments $S_1, S_2,..., S_K$, which satisfy the following properties

$$S = \sum_{i=1}^{K} S_i \,,\; S_i \cap S_j = 0 \text{, when } i \neq j \,,\; i,j = 1,2,..,K \,.$$

The thresholds $\theta_1, \theta_2,..., \theta_L$ can be determined in various ways coming from the image histogram properties. This problem is not considered in this paper. We use, as a rule, the thresholds $\theta_1, \theta_2,..., \theta_L$, which are determined by dividing of interval $[0,255]$ into $L+1$ approximately equal intervals by the following formula

$$\theta_\ell = \left[\frac{255}{L+1} \right] \times \ell, \; \ell = 1,2,..,L \,.$$

This segmentation procedure and corresponding software tool *Image Repair* are described in [Asatryan, 2007]. The name of that software shows that it can be used for damaged image repair purposes. The repairing process is based on the possibility to change the content of any obtained segment. Some operations can be done automatically. In this paper we apply the segmentation procedure to a binary image at $L = 1$ and use the whitening operation described below. All segments and the number of pixels of corresponding segments are available as output parameters of the software tool.

Pore extraction procedure

The procedure for a fingerprint pore extraction includes the following operations:

1. Binarization of a fingerprint image.

2. Segmentation. We use the procedure of segmentation described above for $L = 1$. Let $S_1, S_2, .., S_K$ be the resulted segments. Denote by n_k the number of pixels of segment S_k, $k = 1,2,\ldots,K$.

It is necessary to note that the hierarchical segmentation process can be applied to the initial fingerprint twice. The first application can be performed at $L > 1$ then the segmented image is simplified by averaging of the pixels of every segment and changing the intensities of all pixels by the average value. This process can be interpreted as a special preliminary filtration method. After simplification the resulted image can be considered instead of the initial fingerprint.

3. Segment whitening procedure changes the intensities of all "black" pixels of a segment S_k to 255 if $n_k \le t_1$, where t_1 is a prior chosen integer. As a result we get "white" segment S_k^1 with the same number n_k of pixels of intensity 255.

4. Inverting of an image is a well known procedure, which substitutes a pixel of intensity I by a pixel of intensity $255 - I$.

5. Subtraction of two images is an operation which obtains an image from two another images by subtracting of intensities of that images. In general, this operation must be used correctly to provide the visualization of resulted image. In this paper the correctness of the procedure is provided automatically.

Pore extraction algorithm. Let F be an initial fingerprint image with pores. The pore extracting algorithm consists of the following steps.

Step 1. **Binarization** of the fingerprint image F. Binarization is performed using a threshold by Otsu method [Otsu, 1974] or other method for well distinguishing the pores in the image. Denote by F_B the binarized image F.

Step 2. **Segmentation** of the image F_B by using the software tool *Image Repair* at $L = 1$. Let K be the number of segments, and $S_1, S_2, .., S_K$ be the segments. The program *Image Repair allows* the visualization of the segmented image and any segment S_k by choosing a point within the segment. Thus, we can estimate the maximal size of pores which we want to extract from the image F. Denote by t_1 the maximal size of extracting pores.

Step 3. **Inversion** of the segmented image F_B. Denote it by F_B^I.

Step 4. **Whitening** the image F_B^I at threshold value of $t_0 - 1$. Let's denote the whitened image by $F_B^I(t_0)$.

Step 5. **Whitening** the image $F_B^I(t_0)$ at threshold value of $t_1 + 1$. Let's note that after this operation all the black segments of size between t_0 and t_1 of the image F_B^I will be whitened. Let's denote the whitened image by $F_B^I(t_0, t_1)$.

Step 6. **Subtraction** of the image $F_B^I(t_0)$ from the image $F_B^I(t_0, t_1)$. The closed pores will be presented in the resulted image as white segments on the black background. Denote it by $\Delta F(t_0, t_1)$.

Results of an experiment

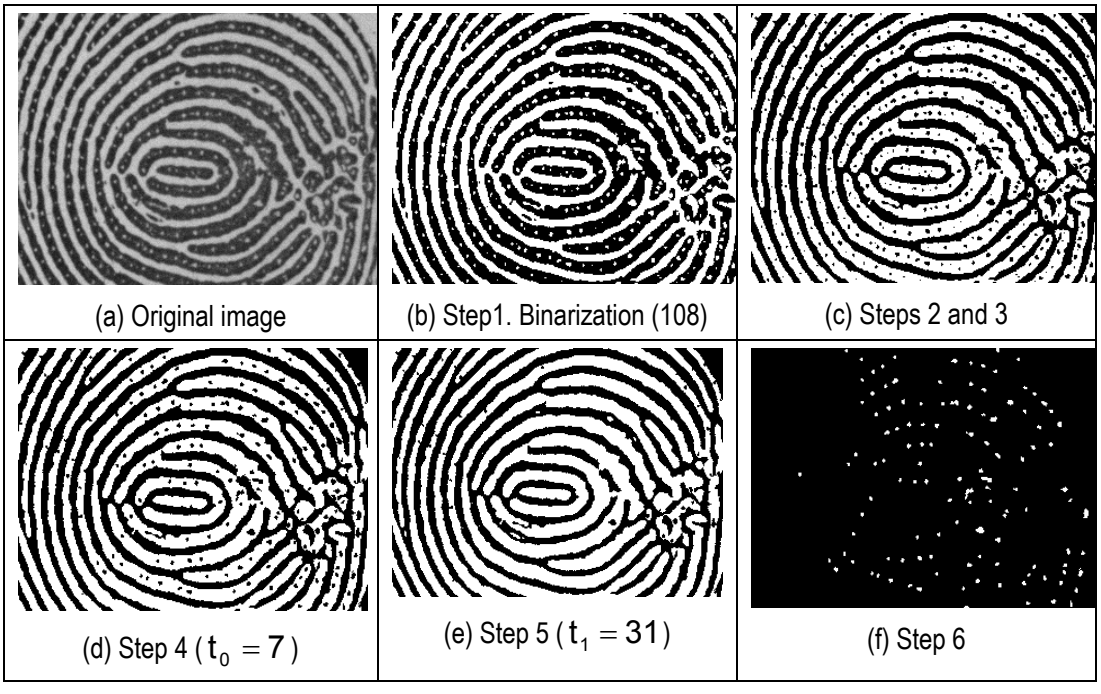
Before application of above described algorithm it is necessary to determine the interval of pixel number for pores which corresponds to a fingerprint scanner resolution. Starting from the literature information [Busselaar, 2010], we can be oriented to pores size of 60–220 micron, which corresponds to 8-30 pixels per pore for 1000 ppi scanned fingerprint image. Therefore we can choose the segments which include correctly determined 8-30 pixels.

The pore extraction algorithm can be illustrated by the following experiment.

Experiment. Original fingerprint image fragment and results of the experiment are shown in Table. Binarization of the image F (a) is performed at threshold value of 108 (b). Whitening of segments of binarized and inverted image is performed at $t_0 - 1 = 7$. The image $F_B^I(t_0)$ is shown in (d). Then the image $F_B^I(t_0)$ was whitened at threshold $t_1 + 1 = 31$ (e). The resulted image $\Delta F(t_0, t_1)$ after subtraction is shown in (f).

All extracted pores are clearly seen in the black background.

Table. Fingerprint pore extraction results at each step of the algorithm



(a) Original image | (b) Step1. Binarization (108) | (c) Steps 2 and 3

(d) Step 4 ($t_0 = 7$) | (e) Step 5 ($t_1 = 31$) | (f) Step 6

The accuracy of proposed procedure can be checked by comparing the locations and the number of pores detected by visual analysis in the original fingerprint image and in the image of right bottom cell of Table 1.

Conclusion

The proposed algorithm for closed pore extraction from a fingerprint image is based on the consecutive performance of above specified simple image processing procedures, namely image binarization, segmentation, inversion, whitening etc. The algorithm consists of mentioned operations applied by above described steps. The described example of pores detection and extraction results shows the effectiveness of our approach to the pore extraction problem. The proposed algorithm can be easily included into automated fingerprint recognition systems.

Bibliography

A. K. Jain, Y. Chen, M. Demirkus, "Pores and Ridges: High-Resolution Fingerprint Matching Using Level 3 Features". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 1, pp. 15-27, Jan. 2007.

D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. Handbook of Fingerprint Recognition. Springer, New York, 2003.

J. Stosz and L. Alyea. Automated system for fingerprint authentication using pores and ridge structure. Proc. SPIE Conference on Automatic Systems for the Identification and Inspection of Humans, 2277:210–223, 1994.

K. Kryszczuk, A. Drygajlo, P. Morier, Extraction of level 2 and level 3 features for fragmentary fingerprints, in: Proceedings of the 2nd COST Action 275 Workshop, 2004, pp. 83–88.

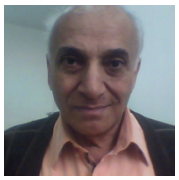
Q. Zhao, D. Zhang, L. Zhang, N. Luo. Adaptive fingerprint pore modeling and extraction. Pattern Recognition, 43, pp. 2833–2844, 2010.

N. Otsu (1979). A threshold selection method from gray-level histograms. IEEE Trans. Syst. Manage. Cybern. (SMC) 9: 62; pp. 377–393.

D.G. Asatryan, G.S. Sazhumyan, H.S. Shahverdyan. Technique for Coherent Segmentation of Image and Applications. Mathematical Problems of Computer Science, IIAP, Yerevan, Armenia, Vol. 28, 2007, pp. 88-93.

E.J. Busselaar. Improved pores detection in fingerprints by applying ring led's (525 nm). Optica Applicata, Vol. XL, No. 4, pp. 843-861, 2010

Authors' Information



David Asatryan *– Professor, Head of group of the Institute for Informatics and Automation Problems of NAS Armenia, 1, P.Sevaki Str., 0014, Yerevan, Armenia, e-mail: dasat@ipia.sci.am.*

Major Fields of Scientific Research: Digital signal and image processing.

Grigor Sazhumyan *– Candidate of Technical Sciences, Software Engineer, Institute for Informatics and Automation Problems of NAS Armenia, 1, P.Sevaki Str., 0014, Yerevan, Armenia, e-mail: grigorsazhumyan@gmail.com .*

Major Fields of Scientific Research: Digital signal and image processing, Software developing.

ON A MODIFICATION OF THE FREQUENCY SELECTIVE EXTRAPOLATION METHOD

Gevorg Karapetyan and Hakob Sarukhanyan

Abstract: *In this paper is described a method for automatic analysis of missing block neighbor area. The analysis, which is based on Canny edge detection and calculation of homogeneity coefficient in missing block neighbor area, provides suboptimal rectangular support area for each block. The suboptimal support area for each block is used in Selective Extrapolation algorithm. Paper includes experiment results of proposed method which are compared with results of selective extrapolation where the support area size is fixed for all blocks.*

Keywords: *Image processing, selective extrapolation, Canny edge detection, missing blocks concealment.*

Introduction

Image and video compression standards such as JPEG, MPEG, H.263 [Stockhammer, Hannuksela, 2005] are used for efficient data transmission and representation over the internet. In these standards the data is coded by block based techniques. In case of JPEG the image is segmented into non-overlapping blocks. After segmentation each block is compressed, then stored or transmitted over communication channels. The transmission over the fading channels may lead to transmission errors. If the block is well compressed the error of one bit may cause loss of a whole block. After such erroneous transmission the image may contain block losses. Concealing of block loses is important problem in the image processing.

A powerful algorithm, based on signal extrapolation, is Frequency Selective Extrapolation [Kaup, et all, 2005], which operates in the Fourier domain and has high extrapolation quality. The Frequency Selective Extrapolation algorithm conceals block losses by estimating lost areas from correctly received adjacent areas.

In algorithm of selective extrapolation the support area size for missing blocks is predefined. Missing block surrounded by fixed number of known pixels in each direction forms the support area. Therefore, the missing block is located in the center of support area. But if the support area is not homogeneous, some parts of it may negative affect on the results of the extrapolation. To overcome that problem we analyze the support are in each direction and choose suboptimal size of it. Then the suboptimal sizes of support areas are used in Selective extrapolation algorithm.

The paper organized as follows: In the Section 2 the theory of frequency selective extrapolation is shortly described. Then, in Section 3 the proposed algorithm of support area analysis is described and step by step algorithm of selective extrapolation. In Section 4 the results of the experiments are shown and in Section 5 we conclude the work and summarize our future plans.

Frequency Selective Extrapolation Overview

Fig. 1 schematically shows example of area Λ which is a part of digital image, where missed block has occurred. Λ area is composed of dark grey area **B** (missing block area), which has to be estimated by extrapolation of elements in light grey area **A** (support area).The part of image, which is contained in the area Λ we denote F = {f[m,n]}, where $m = \overline{0, M-1}$, $n = \overline{0, N-1}$. In order to extrapolate the observed area Λ elements of the known area are approximated by the parametric model $G = \{g[m,n]\}$, which is also defined in Λ

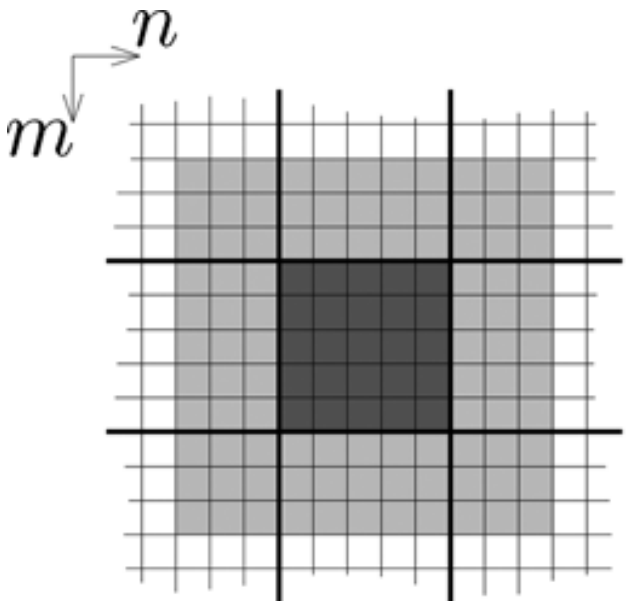


Fig.1: Part of digital image with missing area

The parametric model is a weighted liner combination of basis functions. In this algorithm as basis functions 2D DFT functions are used. The parametric model is generated successively, in each iteration we calculate $g^{(v)}[m,n] = \sum_{(k,l)\in K^v} c_{k,l}^{(v)} \varphi_{k,l}[m,n]$, where K^vrepresents a set of basis functions used for the expansion of the parametric model $g[m,n]$, before iteration v. $c_{k,l}^{(v)}$is expansion coefficient which is calculated in iteration v.$\varphi_{k,l}[m,n]$ is DFT basis function which is defined in entire area Λ.

$$\varphi\_(k,l)\,[m,n] = e\text{^}(j\,2\pi/M\,mk)\;e\text{^}(j\,2\pi/N\,nl), \qquad j = \sqrt{(-1.)}$$

For each iteration a residual error is calculated

$$"r\text{^}((v))\,[m,n] = (f[m,n] - g\text{^}((v))\,[m,n]\,)b[m,n]",$$

where $b[m,n]$ is the window function

$$b[m,n] = \begin{cases} 1, (m,n) \epsilon A \\ 0, (m,n) \epsilon B \end{cases}.$$

In each iteration the residual error of support area is decreased by $\Delta g[m,n]$, which shows the change of parametric model between iteration steps $v$and $v+1$,$\Delta g[m,n]$ is updated in each iteration. Then, the residual error is calculated in the following way: $r^{(v+1)}[m,n] = r^{(v)}[m,n] - \Delta g[m,n] b[m,n]$.

The $c_{k,l}^{(v)}$ expansion coefficient is calculated for minimizing weighted instantaneous residual error energy E_A (1). For iteration step $v+1$ the residual error energy is calculated in following way:

$$E_A^{(v+1)} = \sum_{(m,n)\epsilon\Lambda} w[m,n]\left(r^{(v)}[m,n] - \Delta g[m,n]\right)^2, \quad (1)$$

where

$$w[m,n] = \begin{cases} 0, when\ (m,n)\epsilon A \\ positive\ value, when\ (m,n) \in B \end{cases}$$

$\Delta g[m,n]$ is updated in the following way:$\Delta g[m,n] = \Delta c \varphi_{u,v}[m,n]$,where$\Delta c$ is optimal update of expansion coefficient in each iteration, (see, [Kaup, et all, 2005]). The expansion coefficient $c_{u,v}$ is updated as:

$$c_{u,v}^{(v+1)} = c_{u,v}^{(v)} + \Delta c \quad (2)$$

The pair (u,v) is included in the set of basis functions K^{v+1}, if a function with such coefficients was not included before: $K^{(v+1)} = K^{(v)} \cup (u,v), \text{If } (u,v) \notin K^{(v)}$

The error energy is updated in following way, see [Kaup, et all, 2005]

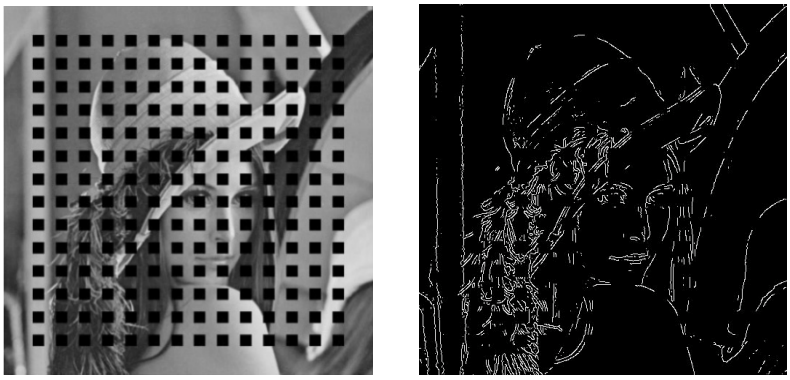
$$E_A^{(v+1)} = E_A^{(v)} - \Delta E_A^{(v+1)} \quad (3)$$

From equation (3) we see that $E_A^{(v+1)}$ has minimum value, when $\Delta E_A^{(v+1)}$ has maximum value.

$(u,v) = \text{argmax}\ \Delta E_A^{(v+1)}$

Modification of selective extrapolation algorithm

We need to define support areas for each missing block. The definition of support area sizes is made by analysis of missing block neighbor area. We analyze the copy of input image, which is shown in Fig.2 (a), processed by Canny edge detection algorithm [B.Jahne, 2005] Fig.2 (b). The processed image has only white and black pixels. White pixels are equal to 1 and show the edges of input image. The black pixels are equal to 0 and show regions of image that have high homogeneity.



(a) (b)

Fig. 2 (a) Input Image (b) Copy of input image processed by Canny algorithm

For estimation of support area we calculate coefficient of homogeneity (COH) by following formula

$$COH = \frac{(number of white pixels)}{(\text{number of all pixels})} * 100.$$

We analyze the neighbor area of missing block in each direction and define number of surrounding supporting pixels for it. Fig. 3 shows missing block (black area) which has $(x, y), (x_1, y_1)$.For analysis of support area we use 4 types of windows for each direction. The coordinates of those windows are given below:

Left Side:$(x - (k+1)m, y - m),\ (x - km, y_1 + m)$;

Right: $(x_1 + km, y - m),\ (x_1 + (k+1)m,\ y_1 + m)$; (4)

Top: $(x - m, y - (k+1)m),\ (x_1 + m,\ y_1 - km)$;

Bottom:$(x - m,\ y_1 + km),\ (x_1 + m, y_1 + (k+1)m),\ 0 \le k \le 4.$

Example of left side window is shown in Fig. 3, where A_0 is window for k=0

and A_1 for k=1.

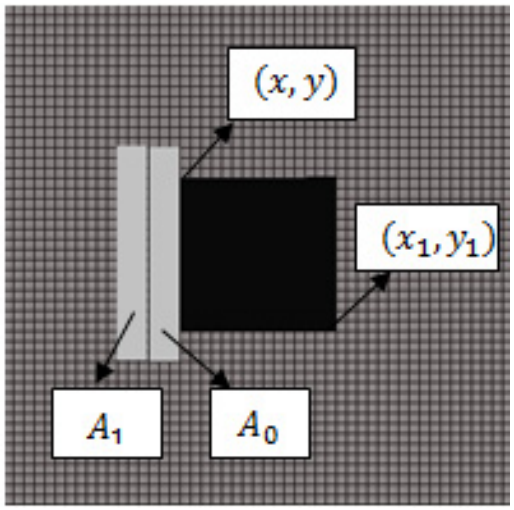


Fig. 3: Missed block (black), analyzed areas (light grey)

For each side of support area we calculate the absolute value of difference of COH for neighbor windows A_k and A_{k+1}:

$$\Delta = |COH(A_{k+1}) - COH(A_k)|.$$

If $\Delta \leq 5$, we calculate Δ for A_{k+1} and A_{k+2}. We continue this until k=4 or $\Delta > 5$ .After termination for current side of support area, we obtain the number of support pixels. The number of pixels is equal to (k+1)m, where m is fixed number. Then, we implement the same algorithm for all sides of missing block support area. The number of surrounding supporting pixels for all sides we record in array. After implementation of the algorithm for all missing blocks we pass the array of support area parameters to modified selective extrapolation algorithm.

The step by step algorithm is given below:

Implementation of Canny edge detection algorithm on copy of input image

Record support area sizes for all blocks in array, pass the array to selective extrapolation algorithm

Selective extrapolation algorithm is initialized with $g[m,n]^{(0)} = 0$ and the residual error equals to weighted original signal in the first iteration: $r_w^{(0)}[m,n] = w[m,n]f[m,n]$. Where,

$$w[m,n] = \begin{cases} 0.74\sqrt{\left(m - \frac{M-1}{2}\right)^2 + \left(n - \frac{N-1}{2}\right)^2}, (m,n) \epsilon A \\ 0, (m,n) \epsilon B \end{cases}.$$

Then, we transform $w[m,n]$ and $r_w^{(0)}[m,n]$ into frequency domain by Fast Fourier Transform algorithm [Brigham, Morrow,1967].

$\Delta E_A^{(v+1)}$ energy decrease computation.

Selection of basis functions with indexes (u,v).

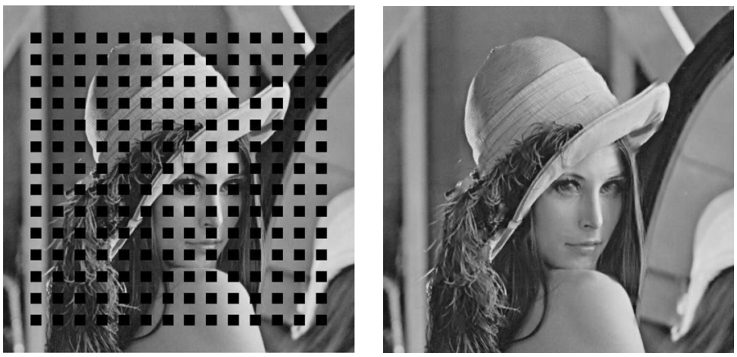
$c_{u,v}^{(v+1)}$ coefficient update for Δc, see equation (2).

Check if the $\Delta E_A^{(v+1)}$ is less then predefined E_{min}=15 threshold or if the number of iterations is more then 11. If no go to step 2 else go to step 6.

The algorithms terminates and we get parametric model by Inverse Discrete Transform of $G^{(v)}[k,l]$: $g^{(v)}[m,n] = IDFT_{M,N}\{G^{(v)}[k,l]\}$.

In the end elements of the missed block of input $f[m,n]$ image are replaced with corresponding elements of $g^{(v)}[m,n]$ parametric model.

The algorithm is implemented in Matlab. Fig. 4 shows the result of the selective extrapolation algorithm.

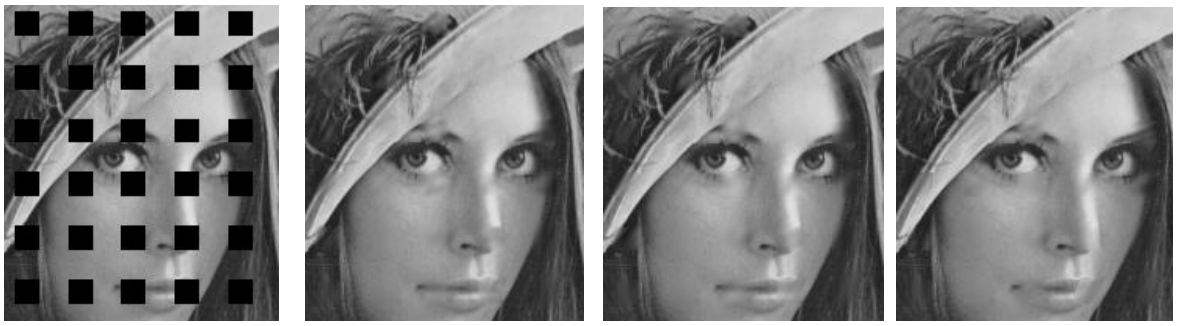


(a)……(b)

Fig.4 (a) Input image with missing blocks; (b)Concealed Image.

Experiment results

We have implanted the proposed algorithm and compared it with result of selective extrapolation algorithm. The size of missing block is 16x16. The value of m in (4) is equal to 3px. The PSNR value is calculated only for extrapolated regions.



(a)……(b) (c) (d)

Fig. 5: (a) Input image with missing blocks; (b) concealed image by Selective Extrapolation (SE) with fixed 8px support area PSNR: 23.13dB; (c) concealed image by SE with fixed 16px support area. PSNR: 23.50dB; (d) concealed image by proposed method. PSNR: 23.61dB.

Fig.5 shows the result of proposed method picture (d) which is compared with results of Selective Extrapolation method (b), (c). The Selective Extrapolation algorithm was implemented for different sizes of support areas.

We have paid attention on not homogeneous regions such as region of eyes and compared the results of proposed method with results of SE method. We can see that in not homogeneous regions by proposed algorithm is obtained significantly higher PSNR value in comparison to methods we the support area size is fixed.

(a) (b) (c)

Fig. 6: (a) Input image with missing blocks; (b) concealed image by SE with fixed 16px support area. PSNR: 21.53dB; (c) concealed image by proposed method. PSNR: 22.28dB;

The experiment results show that the proposed method is effective and for not homogeneous regions provides high quality of extrapolation.

Conclusion and future work

In this paper was described modification method of frequency selective extrapolation. The modification based on support area analysis is targeted to choose suboptimal size of support area. The usage of Canny algorithm allow us easily detect the edges of input image, thus to calculate coefficient of homogeneity.

The results of the experiments have shown that the method is effective. We have shown that the size of support area affects the quality of extrapolation. In our future work we are going to improve the support area analysis technique. Moreover, in the frequency selective extrapolation method discrete Fourier transform (DFT) is used, we are going to use other transformations such as Haar, discrete cosine transform (DCT) and Hadamard transform [S.Agaian, et all, 2011].

Bibliography

[Stockhammer, Hannuksela, 2005] T. Stockhammer and M. M. Hannuksela, "H.264/AVC video for wireless transmission," IEEE Wireless Communications, vol. 12, no. 4, pp. 6–13, 2005.

[Kaup, et all, 2005] Kaup, K. Meisinger, and T. Aach, "Frequency selective signal extrapolation with applications to error concealment in image communication," International Journal of Electronics and Communications, vol. 59, no. 3, pp. 147–156, 2005.

[Brigham, Morrow, 1967] E. O. Brigham; R. E. Morrow; "The fast Fourier transform" , IEEE, vol. 4,No 12, Dec. 1967

[B.Jahne, 2005] B. Jahne. Digital Image Processing, Springer–Verlag Berlin Heidelberg,2005, 580p.

[S.Agaian, et all, 2011] S.Agaian, H.Sarukhanyan, K.Egiazarian, J.Astola. Hadamard Transforms, SPIE Press, 2011, 520p.

Authors' Information

Gevorg A. Karapetyan– *PhD Student of Institute for Informatics and Automation Problems of NAS of RA;e-mail: gevorgka@gmail.com*

Major Fields of Scientific Research: Digital Signal and Image Processing, Concealment of block losses, Signal extrapolation

Hakob G. Sarukhanyan– *Head of Digital Signal and Image Processing Laboratory, Institute for Informatics and Automation Problems of NAS of RA;e-mail: hakop@ipia.sci.am*

Major Fields of Scientific Research: Digital Signal and Image Processing, Fast Orthogonal Transforms, Parallel Computing

ACTIVITY RECOGNITION USING K-NEAREST NEIGHBOR ALGORITHM ON SMARTPHONE WITH TRI-AXIAL ACCELEROMETER

Sahak Kaghyan, Hakob Sarukhanyan

Abstract*: Mobile devices are becoming increasingly sophisticated. These devices are inherently sensors for collection and communication of textual and voice signals. In a broader sense, the latest generation of smart cell phones incorporates many diverse and powerful sensors such as GPS (Global Positioning Systems) sensors, vision sensors (i.e., cameras), audio sensors (i.e., microphones), light sensors, temperature sensors, direction sensors (i.e., magnetic compasses), and acceleration sensors (i.e., accelerometers). The availability of these sensors in mass-marketed communication devices creates exciting new opportunities for data mining and data mining applications. So, it is not surprising that modern mobile devices, particularly cell phones of last generations that work on different mobile operating systems, got equipped with quite sensitive sensors. This paper is devoted to one approach that solves human activity classification problem with help of a mobile device carried by user. Current method is based on K-Nearest Neighbor algorithm (K-NN). Using the magnitude of the accelerometer data and K-NN algorithm we could identify general activities performed by user.*

Keywords: *human activity classification; K-NN algorithm; mobile devices; accelerometer; Android platform*

Introduction

The data to recognize human's activity is from the physical hardware sensors, and the combination of the accelerometer, the compass sensors and GPS are the most commonly used sensor devices. This project's objective is to explore how effective is in general the K-NN algorithm and, in future works, its modifications in user activity classification problem solving. For our current research we have taken the base algorithm without any serious modifications, although in our future works we shall use different combinations of this one and other methods such as, for example, decision trees. In order to know the accuracy of this algorithm we also created two applications. One of them is a smartphone application that works on the Android platform and is able to get and store data concerning user's physical activity using incoming signals from tri-axial accelerometer that comes with mobile device that he or she cares. It stores raw data on security disk card or just SD card of given device. After the data saved, it will be transferred on server for further work. Second application is a desktop application that does the rest – activity classification using K-nearest neighbor algorithm. It analyzes incoming data in order to classify transferred activity.

In sections below we shall briefly describe what Android operating system is and what an accelerometer is and it works on a mobile device. Then we shall give a description of K-NN algorithm itself. Finally, our approach to recognize activity from accelerometer data using this algorithm will be introduced and them it will be followed by results.

Android operating system

First step of activity classification problem, discussed in this article, is to create an application that will be able to retrieve acceleration values from smartphone. So, it will be logical to start from choosing a platform that will be used. There are several platforms for mobile phones. Most popular of them are Android, IOS and Windows

Mobile platforms. And here we used Android as a target platform for our experiments. Thus, here we shall focus on explaining how this operating system is organized and what advantages it has. Whether you're an experienced mobile engineer, a desktop or web developer, or a complete programming novice, Android represents an exciting new opportunity to write innovative applications for mobile devices. So, the main question, from which we shall start, will be the following one:

What is Android?

Google describes Android as: The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation.

Generally, Android is a combination of three components:

- ✓ A free, open-source operating system for mobile devices.

An open-source development platform for creating mobile applications.

Devices, particularly mobile phones, that run the Android operating system and the applications created for it.

More specifically, Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK (Software Development Kit) provides the tools and APIs (Application Programming Interfaces) necessary to begin developing applications on the Android platform using the Java programming language [Meier, 2010].

So we decided to use the Android-based cell phones as the platform for our experiments because the Android operating system is free, open-source, relatively easy to program, and according to October of 2011 statistical data, it claims quite impressive position among other mobile operating system manufacturers. Our project currently tested on the following type of Android phone: HTC Desire HD. Data that was collected from accelerometer of this cell phone was stored on cell phone's SD card, although we expect to change this way of data storing because modern cell phones have all necessary interfaces and means to send data directly to server, for example via wireless networks, such as Wi-Fi, or via the Internet. However, data in this work was transferred to server via a USB (Universal Serial Bus) connection, but will make it at least less common or it just will be fully replaced by wireless data transfer mode in our future works. We also expect to modify software application so that later it will be able to do the activity recognition process right on user's cell phone.

At the same time Android platform is widely used not only because of it is free or open-source, but also because of the mechanisms, designed to protect the privacy and security of Android users, as well as the operating system. These methods include the Android security architecture, application certificates and application permissions. The purpose of the Android security architecture is to prevent applications from being able to automatically perform operations that could jeopardize the security of other applications, the operating system or the user. Certificates are used to identify the author of a specific application and to prevent users from installing fraudulent software on their devices. Android will not install an application that has not been signed with a certificate. Therefore, the origin of all published applications is traceable. Android security permissions are handled by the *AndroidManifest.xml* file present within all application files. When a user downloads an application onto their device, they are automatically notified of the permissions the application has access to. This informs the user of what type of information an application is able to collect from the device as well as the hardware the application can use. The *AndroidManifest.xml* file takes care of both software and hardware permissions. But while Android does require permissions for the use of hardware devices such as the camera and vibrator, it does not require permissions to be set in place for the use of any available sensors, including the accelerometer, orientation, and gyroscope sensors. Therefore, alongside with other tools, such as the internet and GPS, can also pose as security threat to the user. And it is possible for an application to collect user information from these sensors without the user's knowledge. Android's application-neutral APIs provide low-level access to the

increasingly diverse hardware commonly available on mobile devices. The ability to monitor and control these hardware features provides a great incentive for application development on this platform.

Accelerometers

The data to recognize human's activity is from the physical hardware sensors, and the combination of the accelerometer, the compass sensors and GPS are the most commonly used sensor devices. The accelerometer is another component that is becoming a standard item in new devices. Several types of accelerometers, the most currently used are based on electro-mechanical devices (micro electro-mechanical systems or MEMS) that include a series of needle-like structures that detect motion, generating the readings are then transmitted to the main circuit.

A tri-axial accelerometer is a sensor that returns a real valued estimate of acceleration along the x, y and z axes from which velocity and displacement can also be estimated. Accelerometers can be used as motion detectors. It measures proper acceleration, which has an experience relative to gravity and is the acceleration felt by people and objects. Accelerometers have been proposed by previous studies as a tool to monitor and assess physical activities of subjects in a free-living environment. The acceleration signals recorded through accelerometers have been used to classify daily living activities. There are extensive researches on using accelerometers to classify activities such as walking, running, falling, sitting, cycling, etc. [Lee, 2010], [Nishkam, et al.,2005], [Fomby, 2008], [Kwapisz et al., 2010]. This sensor can be used to detect movement and the rate of change of the speed of movement. One of benefits of Android platform is that the using of accelerometers in Android applications does not require the application to have permission to use it. So, it is possible to collect accelerometer data from user without his or her knowledge. This will make process of application executing easier to user, because he will not be prompted to give agreement to use built-in accelerometer each time the program runs. The most obvious application for the accelerometer is to change the screen orientation when we rotate the device (Figure 1). The big question is almost all devices use screens with vertical orientation, but activities such as surfing the web or watch videos require a screen with horizontal orientation. Normally you would need to activate an option to change the orientation, but with an accelerometer that can be done automatically. It's something simple, but ends up having a great effect with respect to usability. But there are also other situations when this sensor is also used.

Figure 1: Smartphone changes display of screen from vertical to horizontal when user rotates it.

The accelerometer can also be used to shortcut functions, such as changing the track on MP3 player, answer or reject a call, open or close applications and so on. A good shake-up band, two put the phone in silent mode and so on. Two other important areas are the games and the applications of GPS. In the case of games, accelerometer lets you deploy controls in the style of the Nintendo Wii [Nintendo Official Site], which opens a whole new range of possibilities. For GPS applications, the accelerometer enables the software to detect ripped brakes, cornering and so on. Variables, which can then be used to make software more intelligent, detecting when you missed a turn, or keeping a rough estimate of the location when it loses the satellite signal for a few seconds, for example.

These are improvements that alone does not say much, but together they end up making a big difference, more than enough to pay the small cost increase resulting from the additional component [Hall et al., 2008].

Human activity classification

Human physical activity recognition has been receiving increasing attention in recent years. Human behavior and its classification are significant for the disciplines such as medicine, behavioral sciences, physiotherapy, etc. An accelerometer is an inexpensive, effective and feasible body-worn sensor which has been frequently used in daily physical activity classification. Use of accelerometer in patient's mobile device will help to know whether what kind of moves he does and do that without disturbing him.

Many research groups have studied activity recognition as part of context awareness research [Parkka et al., 2006]. Context sensing and use of context information is an important part of the ubiquitous computing scenario. Context sensing aims at giving a computing device (e.g., cellular phone, wrist-top computer, or a device integrated into clothes) senses, with which it becomes aware of its surroundings. With the senses the device is capable of measuring its user and environment and it becomes context aware. The context describes the situation or status of the user or device. Different devices can use the context information in different ways, e.g., for adapting its user interface, for offering relevant services and information, for annotating digital diary (e.g., energy expenditure), etc. Location and time belong to the group of the most important contexts and the use of these contexts has been studied extensively. However, to recognize the physical activities of a person, a sensor-based approach is needed.

Activity recognition is formulated as a classification problem. In this study we consider following activities performed by user: standing; walking; running; sitting; climbing up stairs; climbing down stairs. The reason of selecting these activities was quite simple. They were selected because they are performed regularly by many people in their everyday life. These activities also involve motions that repeat in time and this in ideal way the data that comes from accelerometer will be periodic and will we think that it can make the recognition process easier. When we record the data from for each of these activities, we record acceleration in three axes. Process of our human activity classification project can be represented by diagram below (Figure 2).

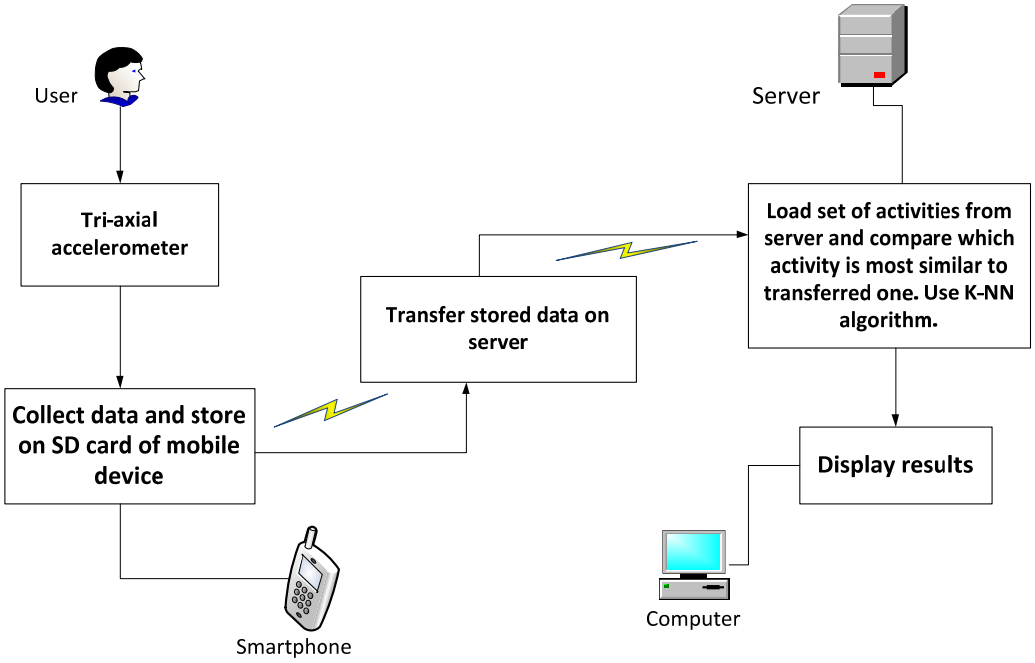


Figure 2: Process of collecting and storing data from smartphone, data transfer and analyzing (logic flow)

First step is to collect data from smartphone user carries in his pocket and store it in memory of mobile device. Second step is to transfer data on server and save it in proper way. Third step is the classification process itself

where unknown activity template will be compared in a loop with every activity template from predefined training set. After calculating distances between each type of activity and target activity, algorithm will display as result that activity which will have minimal distance from the unknown one. Thus, the accuracy of results mostly depends on the templates that were chosen for selected activities.

K-Nearest Neighbor

K-Nearest Neighbor is a supervised learning algorithm where the result of new instance query is classified based on majority of K-Nearest Neighbor category. It is one of the most popular algorithms for pattern recognition. The purpose of this algorithm is to classify a new object based on attributes and training samples. The classifiers do not use any model to fit and only based on memory. K-Nearest Neighbor algorithm used neighborhood classification as the prediction value of the new query instance. Many researchers have found that the K-NN algorithm accomplishes very good performance in their experiments on different data sets. The traditional K-NN text classification algorithm has three limitations: (a) calculation complexity due to the usage of all the training samples for classification, (b) the performance is solely dependent on the training set, and (c) there is no weight difference between samples. The best choice of *k* depends upon the data; generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good *k* can be selected by various heuristic techniques, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when *k*=1) is called the nearest neighbor algorithm. In pattern recognition field, K-NN is one of the most important non-parameter algorithms and it is a supervised learning algorithm. The classification rules are generated by the training samples themselves without any additional data. The K-NN classification algorithm predicts the test sample's category according to the *k* training samples which are the nearest neighbors to the test sample, and judge it to that category which has the largest category probability.

The accuracy of the K-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification. A particularly popular approach is the use of evolutionary algorithms to optimize feature scaling. Another popular approach is to scale features by the mutual information of the training data with the training classes. In binary (two class) classification problems, it is helpful to choose k to be an odd number as this avoids tied votes. One popular way of choosing the empirically optimal k in this setting is via bootstrap method [Suguna et al., 2010].

In similar activity recognition works this algorithm came in combination with other helping methods. For example in [Das et al., 2010] there were also used decision tables and decision trees. They were able to increase the accuracy after the device was properly calibrated for given user.

In pattern recognition, the k-nearest neighbor algorithm (K-NN) is a method for classifying objects based on closest training examples in the feature space. K-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The K-Nearest Neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If *k*=1, then the object is simply assigned to the class of its nearest neighbor. Figure 3 illustrates situation when k is taken 5. Unknown point will be compared with 5 closest neighbors and depending on results will be marked as belonging to proper set. The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, *k* is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most

frequent among the *k* training samples nearest to that query point. Usually Euclidean distance is used as the distance metric; however this is only applicable to continuous variables. In cases such as text classification, another metric such as the overlap metric or Hamming distance [Math32031, 2007], for example, can be used.

A drawback to the basic "majority voting" classification is that the classes with the more frequent examples tend to dominate the prediction of the new vector, as they tend to come up in the *k* nearest neighbors when the neighbors are computed due to their large number.

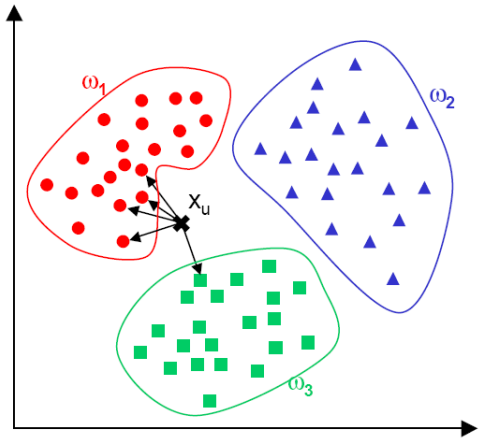


Figure 3: X_u is point unknown template. In this example k=5. Euclidean distance between this point and its 5 closest neighbors is calculated. 4 of them belong to ω_1 and 1 belongs to ω_3, so X_u assigned to ω_1 set.

One way to overcome this problem is to weight the classification taking into account the distance from the test point to each of its *k* nearest neighbors.

There are several ways to calculate the distance between two points in multidimensional space. Suppose we have two points x, y where each point is an n-dimensional vector, i.e. $x = \{x_1, x_2, \ldots, x_n\}$, $y = \{y_1, y_2, \ldots, y_n\}$.

Distance measuring functions can be taken the following ways. We can define distance function $d_E(x, y)$ between two points by measuring their distance according to Euclidean formula or $d_A(x, y)$ distance function that measures absolute distance between them using formulas below:

$$d_E(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2},\ d_A(x,y) = \sum_{i=1}^{n}|x_i - y_i|.$$

Our approach uses Euclidean formula to calculate distance between any given 3-dimensional points.

However, this method it has limitations such as: great calculation complexity, fully dependent on training set, and no weight difference between each class. This once again confirms that one of important means of improving the accuracy of result data is a good choice of training set. In the prediction problem the K-NN “tuning” parameter is the neighborhood size *k*. In the binary classification problem, the K-NN model requires the tuning of two parameters, the neighborhood size and, for each neighborhood size, the cutoff probability for choice. In each of these cases the tuning parameter(s) is (are) chosen to optimize the scoring performance in the K-NN model in the validation data set. Finally, scoring an optimal K-NN model on the test data set provides the opportunity to obtain unconditional performance measures of the optimal K-NN model.

Data Collection, Analyzing and Results

Data from the accelerometer has the following attributes: time, acceleration along x axis, acceleration along y axis and acceleration along z axis.

There are cases, when program must be preliminarily calibrated for given user in order to make the accuracy more effective like it was done in [Das et al., 2010].

We used K-NN algorithm to analyze raw data that we got from mobile device. We used sequential comparing of our activity pattern with training sets, each time calculating the distances between incoming acceleration data points and points of our training sets.

Standard classification algorithms cannot be directly applied to raw time-series accelerometer data. Instead, we first must transform the raw time series data into examples. To accomplish this we divided the data into 10-second segments and then generated features that were based on the 200 readings contained within each 10-second segment. We refer to the duration of each segment as the example duration (ED). We chose a 10-second ED because we thought that it provided sufficient time to capture several repetitions of the (repetitive) motions involved in some of activities. Although we did not perform experiments to determine the optimal example duration value, but we shall explore that in our future works and will find out how a longer period of ED will effect on accuracy.

After user runs the mobile application, he will be able to save incoming data from device into text file where each line contains date and time of accelerations along each axis and the acceleration values themselves.

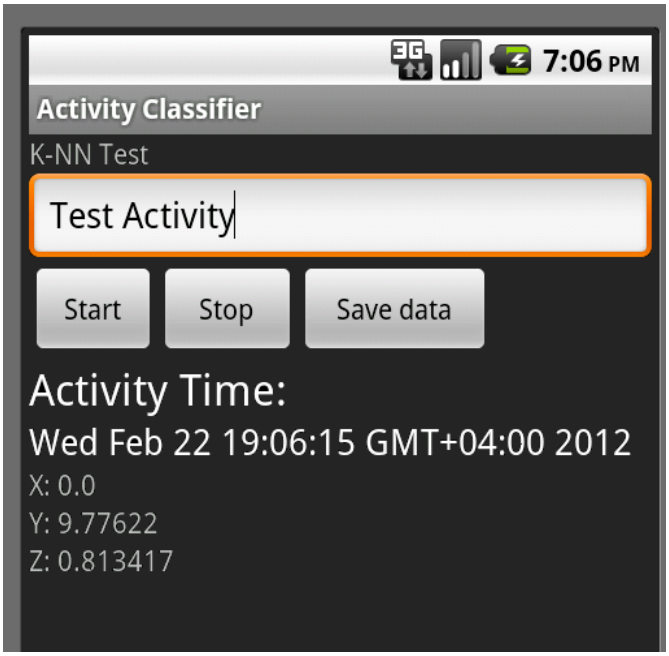


Figure 4: User interface to collect and store data from smartphone sensor

Here is part of code that is responsible for retrieving sensor values from smartphone:

```
1.  // check sensor type
2.  if(event.sensor.getType()==Sensor.TYPE_ACCELEROMETER){
3.      // assign directions
4.      float x=event.values[0];
5.      float y=event.values[1];
6.      float z=event.values[2];
7.      ...
8.      Calendar now = Calendar.getInstance();
9.      ...
10.     s1.add(String.valueOf(x));
11.     s1.add(String.valueOf(y));
12.     s1.add(String.valueOf(z));
13.     currentDT.add(now.getTime());
14. }
15. ...
```

As it was mentioned before, modern mobile devices, especially smartphones, can be equipped with various sensors. So, when we program function that will listen to changes that will occur when sensor data changes, first we must be sure that these changes come from accelerometer and not from any other sensor. Thus, we preliminary must enclose all commands of proper acceleration values retrieving in a block that checks whether incoming data is from accelerometer or other sensor.

The user interface and acceleration values of our Android based acceleration acquiring application is shown on following figures (Figures 5-7):

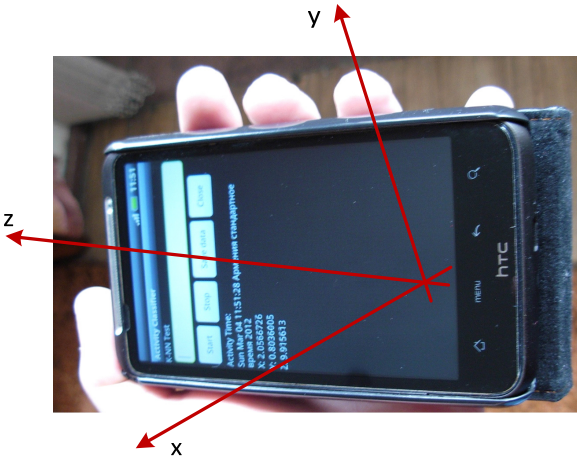


Figure 5: Acceleration along z-axis.
x: 2.056672, y: 0.8036005, z: 9.91561≈ 1g

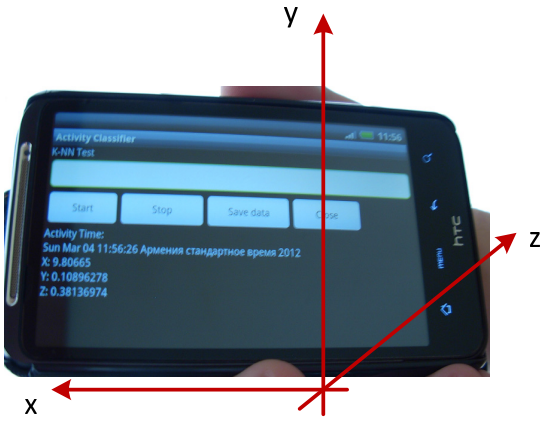


Figure 6: Acceleration along x-axis.
x: 9.80665 ≈ 1g, y: 0.10896278, z: 0.38136974

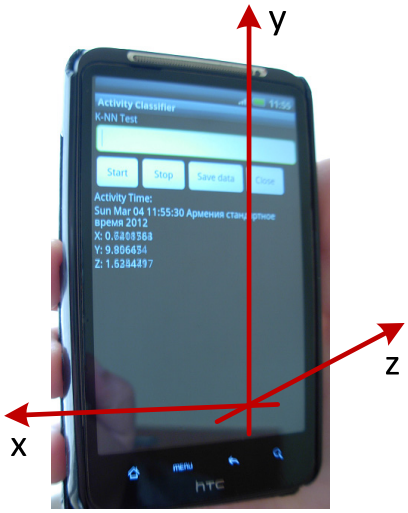


Figure 7: Smartphone is in vertical position.
Acceleration along y-axis.
x: 0.6403563, y: 9.806434 ≈ 1g, z: 1.6334297

Three different positions of this smartphone are shown on this pictures: first one illustrates the situation when smartphone is laying in user's hand (figure 5), or it can just lay on table; figure 6 displays horizontal position of device; finally, last picture shows acceleration values when device stands vertically.

As we can see from these pictures when smartphone is in position shown on figure 5, acceleration along z-axis approaches to 9.8 m/s^2, i.e. to 1g. The same behavior is noticed when mobile device is in position (horizontal or vertical) in which acceleration along proper axis also approximates to 1g.

Also, when device is in horizontal position the display of smartphone automatically turns on 90 degrees.

When inner timer of application stops, data retrieving step overs and the next one, i.e. collected data saving step starts. After data is saved in mobile phone's memory, it will be manually transferred via USB cable on server. Then, after it is on server, the algorithm implementation does the program that runs on server. Modern personal computers have several cores and in order to decrease the time that application spends on calculation process, we used multithreading. To compare incoming data with data representing each activity (each training set) we gave a single thread to it. Although for these experiments k was taken equal to 1 and training sets had less than

thousand points representing each activity, so it gave us only a little bit of efficiency, but if training set will increase and k will increase, this computing method will be very helpful.

Application finds out which template is closer to the current activity vector by measuring distance of each point of our target pattern with all points of each template.

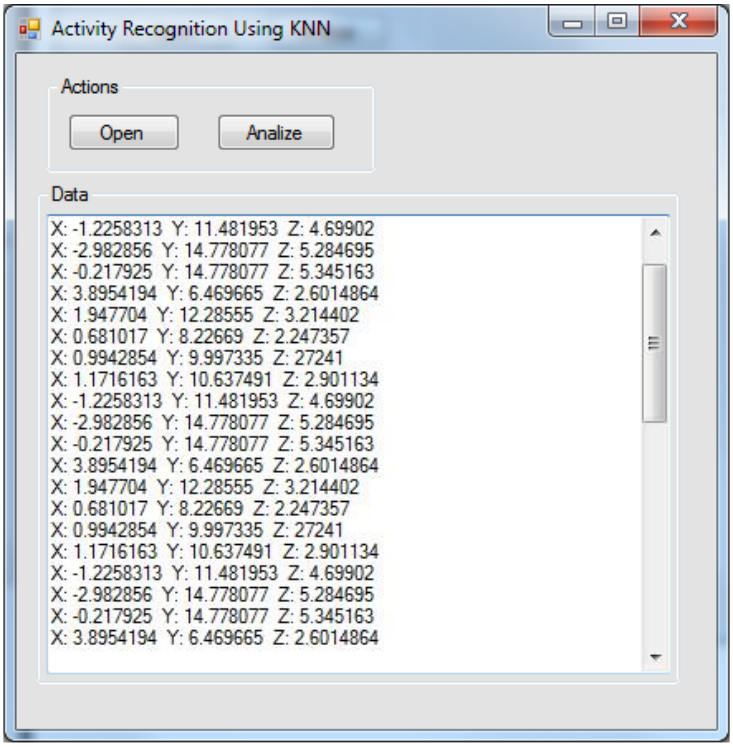


Figure 4: Windows desktop application. User selects data and program does the classification using K-Nearest Neighbor algorithm.

The algorithm that does the classification can be given as follows:

1. TS = { Set of templates describing each activity; Each template is represented as a 3-dimentional array} ;
2. TT = { Target template };
3. MD = { Minimal distance between element of training set and target template };
4. MD = Calculate_Distance(TS[0], TT);
5. **for each** ts $\in$ TS \ TS[0]
    a. LD = { Local value of distance between element of training set and target element };

LD = Calculate_Distance(ts, TT);

    b. **if** (MD $\geq$ LD)
        i. MD = LD;
        ii. Remember_Activity_Class();
    c. **else continue;**

Output = Get_Proper_Activity_Class_Name();

Here *Calculate_Distance(), Remember_Activity_Class()* and *Get_Proper_Activity_Class_Name()* are functions that help to do distance calculation, given activity saving operation and final results display respectively.

It is not hard to notice that every template is a multidimensional vector and, thus, in real program there will be one *for* loop nested inside another, so the complexity of this method is $O(n \cdot p^2)$, where *p* is the length of vector representing single element from training set and *n* is total number of activity vectors. Software application, implementing this algorithm and doing the classification process, was written on C# programming language [Mackey, 2010], [Freeman, 2010] and used multithreading concept in order to increase speed of calculation.

As it was mentioned above, *k* was taken equal to 1. When *k* increases, calculation process becomes more complicated. Despite of that is not so big problem for modern personal computers (because data can be processed on personal computers with multiple cores or even clusters), but processing time increases anyway. This can be a serious problem for data analysis on mobile phone where battery power is limited and power saving problem always plays one of major roles.

For *sitting* and *standing* activities method gives 100% accuracy. For all other activities accuracy can be increased by increasing training set of each template.

Conclusion and future work

The accelerometer proved that it is a useful tool in identifying activities based on user's phone's movements and that the activities can be recognized with fairly high accuracy using a single tri-axial accelerometer. Despite that, activities that are limited to the movement of just hands or mouth are comparatively harder to recognize using a single accelerometer worn near the pelvic region.

We expect in our future work to increase the accuracy of collected data analyzing with K-NN algorithm by studying its modifications. Also using the means of Android platform programming we can more effectively collect data from sensor by choosing other delay mode. At the same time we shall study other methods that help to classify user activity (for example fast Fourier transformations or Hidden Markov Models).

It is also possible to combine data that will be retrieved from accelerometer with the data that will come from GPS sensor. As a result, this combination can make activity recognition process more efficient.

Bibliography

[Lee, 2010] Jungoo Lee, Mobile phone based training application for kayaking, http://cs.anu.edu.au/student/projects/10S2/Reports/Jungoo%20Lee.pdf

[Meier, 2010] Meier, Rito. Professional AndroidTM 2 Application Development, Wiley Publishing, Inc., 2010

[Царьков] Сергей Царьков. Алгоритм ближайшего соседа. http://www.basegroup.ru/library/analysis/regression/knn/

[Das, Green, Perez, Perring, 2010] Sauvik Das, LaToya Green, Beatrice Perez, Michael Murphy, Adrian Perring, Detecting user activities using the accelerometer on Android smartphones, 2010

[Nishkam, Dandekar, Mysore, Littman, 2005] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, Michael L. Littman, Activity recognition from accelerometer data, 2005

[Fomby, 2008] T. Fomby, K-Nearest Neighbors Algorithm: Prediction and Classification, 2008

[Kwapisz, Weiss, Moore, 2010] J.R. Kwapisz, G.M. Weiss, Samuel A.Moore, Activity Recognition using Cell Phone Accelerometers, 2010

[Suguna, Thanushkodi, 2010] N. Sugunal, and Dr. K. Thanushkodi, An Improved k-Nearest Neighbor Classification Using Genetic Algorithm, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 2, 2010

[Tenkomo, 2006] K. Tenkomo. K-Nearest Neighbor Tutorial. 2006. http://people.revoledu.com/kardi/tutorial/KNN/index.html

[Mackey, 2010] Alex Mackey, Introducing .NET 4.0, 2010

[Freeman, 2010] Adam Freeman, Pro .NET 4 Parallel Programming in C#, 2010

[Ancillotti] Ancillotti. Smartphones Accelerometers. http://ancillotti.hubpages.com/hub/SmartphonesAccelerometers

[Hall, Park, 2008] P. Hall; B. U. Park; R. J. Samworth (2008). Choice of neighbor order in nearest-neighbor classification. Annals of Statistics 36: 2135–2152. doi: 10.1214/07-AOS537.

[Parkka, Ermes, Korpipaa] Juha Parkka, Miikka Ermes, Panu Korpipaa. Activity Classification Using Realistic Data from Wearable Sensors. IEEE Transactions on information technology in biomedicine, vol. 10, No. 1, 2006.

[Math32031, 2007] Coding Theory. Part 2 – Hamming Distance, 2007, http://www.maths.manchester.ac.uk/~pas/code/notes/part2.pdf

[Nintendo Official Site] http://www.nintendo.com/wii

Authors' Information

Sahak Kaghyan *– PHD student, Department of System Programming, Russian-Armenian (Slavonic) University, e-mail:* sahak.kaghyan@gmail.com

Major Fields of Scientific Research: Digital Signal and Image Processing, Data Mining, Object Oriented Systems, Soft Computing

Hakob Sarukhanyan *– Head of Digital Signal and Image Processing Laboratory, Institute for Informatics and Automation Problems of NAS of RA; e-mail: hakop@ipia.sci.am*

Major Fields of Scientific Research: Digital Signal and Image Processing, Fast Orthogonal Transforms, Parallel Computing

PARETO-OPTIMUM APPROACH TO MATHEMATICAL MODELING OF ODOURS IDENTIFICATION SYSTEM

Andriy Zavorotnyy, Veda Kasyanyuk

***Abstract**: Mathematical model of vapor identification system is developed. Calibrating signals from vapor sensors are used to identify unknown input to vapor sensors and approximate output from eventual sensor system. Approximation formulas are resulted from pareto-optimum solution of multi-criterion problem. The developed method can be used to create new measuring-calculating systems within "device + PC = device with added benefits" framework.*

***Keywords**: identification, an odorant, impacted data, measuring system, pareto-optimization*

***ACM Classification Keywords**: I.6 Simulation and Modeling*

Introduction

Modeling of awareness is one of the most important tasks scientists have faced in the space of human modeling. This problem is complex and comprises lots of subtasks that can be considered as pretty much independent problems though. The most outstanding problems are investigation of decision-making process, modeling of human memory, pattern identification, etc. Furthermore the last one is deemed as a complex problem and consists of identification of visual images («an electronic eye»), identification of liquids («electronic tongue»), recognition of odours («an electronic nose»), etc. In this article one of possible approaches to modeling of "an electronic nose» is presented.

Measurements Methodology

Structure of gas media can be placed on record by means of quartz microbalance. This approach is founded on the fact of proportion between weight of substance adsorbed on a surface of a quartz plate (lets denote it as Δm) and frequency of fluctuations of the quartz (lets denote it as Δf) [Eichelbaum, 1999]:

$$\Delta f \approx -\frac{2f_0^2}{A\sqrt{\rho_q \mu_q}} \Delta m \ ,$$

Here f_0 is a fundamental frequency of the quartz resonator, A is an effective area of a surface of quartz, μ_q is a “piezoelectric hardness” and ρ_q is a density of the quartz. Selectivity of sensor system is achieved due to coating of quartz surface with layers that are sensitive to given chemical elements.

Measurements process with sensor-based system results into output signals data set. This data then can be interpreted and unknown odor can be identified within reasonable timeframe. For that mathematical method has been developed. This method take into account that output signals can be inaccurate. It is also assumed that sensor system is complex enough therefore its model is deemed to be unknown. Instead the method uses sensor

system's measurements of template odors to assess unknown odor and simulate output of eventual measuring system for this odor.

Mathematical Model of Measurements Results Processing

The proposed method of mathematical processing of sensors' signals is based on pareto-optimum approach to calibration of unknown model with etalon measurements results [Belov, 2001].

According to the approach the unknown model of measuring process can be presented in terms of linear operator that is defined on a Hilbert space. A series of template measurements should be conducted. Then the results of measurements are used to approximate just unknown input to measuring process, i.e. without interim approximation of unknown model of the process. In this way ultimate approximation results are less dependent on interim errors that probably were present should unknown model would be approximated.

Approximation is posed as an optimization problem with two criteria. The first criterion is minimization of noise energy, i.e. dispersion of noise in optimization result. The second criterion is minimization of population mean of squared subtraction of expected output signal from approximated output signal across all template inputs.

The optimization problem is set according to pareto-optimum principle, i.e. these two criteria are directed towards minimum simultaneously.

So, considering the primary assumption that communication between an input and an exit from sensors system is described by the unknown linear operator and considering presence of noise in results of measurements, we can write down

The following mathematical model describes measuring process of input odor by sensor-based system

$$\vec{y} = G\vec{u} + \vec{\nu}\,. \tag{1}$$

The model (1) takes into account that measuring process is linear since G is unknown linear operator that is mathematical abstraction of sensor system. Furthermore the model consider known measurement results $\vec{y} = (y_1 \dots y_n)^*$ are impacted by environment noise $\vec{\nu} = (\nu_1 \dots \nu_n)^*$. Unknown input $\vec{u} = (u_1 \dots u_m)^*$ is mathematical abstraction of a multi-component odor that should be identified. It's a set of content levels of primary odorants. Asterisk * denotes an adjunction of an element, i.e. either vector or matrix.

Let's assume that $M(\vec{\nu}) = 0$, where M denotes an average of distribution. Then let's $\Re$ denotes an operator of covariance of noise [Pitiev, 1989]. The operator $\Re$ is deemed to be known and nonsingular. Finally let's consider q template measurements that have been conducted according to the scheme (1) on known odorants $\vec{u}_j$, $j = \overline{1,q}$. As a result we have q measurement results

$$\vec{y}_j = G\vec{u}_j + \vec{\nu}_j,\ j = \overline{1,q}\,. \tag{1}$$

For all measurements (2) nose summands $\vec{\nu}_j$, $j = \overline{1,q}$ are deemed to satisfy to $M(\vec{\nu}_j) = 0$ condition and result into the same covariance operator $\Re$.

Once it's assumed that sensor system's parameters are unknown let's use measurement results (2) to resolve optimization problem that has been set with criteria described above. Should P denotes eventual sensor-based

system that provide desirable processing of the same unknown odor then the optimization objective is to approximate both unknown content levels of primary odorants $\hat{\vec{u}}$ and $\hat{P\vec{u}}$. P is deemed to be set beforehand, e.g. by odor identification domain expert.

Approximation of $P\vec{u}$ is done by means of processing with a linear operator B of known signal $\vec{y} = (y_1 \dots y_n)^*$ [Pitiev, 1989]. The operator B should be calculated in the way to satisfy pareto-optimization problem stated above. In formal terms the problem is set as follows

$$\begin{cases} h(B) = M\|B\vec{\nu}\|^2 \to \min\limits_B \\ \varphi(B) = M\sum\limits_{j=1}^{q}\|B\vec{y}_j - P\vec{u}_j\|^2 \to \min\limits_B \end{cases} \quad (3)$$

The first criterion in (3) poses minimization of noise energy. Respectively the second criterion is a minimization of mean of squared subtraction of expected output signal from approximated output signal.

This problem has been put and successfully resolved in the general case for linear operators G, P, B that are defined on a Hilbert space [Zavorotnyy, 2004]. Additional restriction of being bounded has been imposed to the operator G.

To resolve Pareto problem (3) let us minimize convex convolution

$$M\left(\lambda\|B\vec{\nu}\|^2 + (1-\lambda)\sum_{j=1}^{q}\|B\vec{y}_j - P\vec{u}_j\|^2\right) \to \min_B,\ \lambda \in (0;1). \quad (4)$$

To find operator B that would be optimum in reference to (4) Frechet derivative from convex convolution (4) should be set equal to zero. The constructed equation can be solved and result into continuum set of solutions in the form

$$B(\alpha) = \sum_{j=1}^{q} P\vec{u}_j f_j^*\left(\sum_{j=1}^{q} f_j f_j^* + (\alpha + q)\Re\right)^{-1},\ \alpha \in (0;1). \quad (5)$$

In formula (5) conventional signs mean $f_j = M\vec{y}_j$ and $\alpha = \dfrac{\lambda}{1-\lambda}$ is a pareto-optimisation parameter. Any operator B in (5) is an effective solution of Pareto problem (3).

The developed method has been used for interpretation of measurements results to identify odorants from spirit group. Therefore the equation (2) and the problem (3) have been instantiated in terms of Euclidean spaces. Specifically operator $\Re$ has been defined as covariant matrix R. Similarly to the assumption that has been stated for $\Re$ the matrix R has been deemed non-singular too, i.e. $\det R \neq 0$. In turn operator P has been instantiated as identity operator. Therefore content levels of primary odorants have been approximated.

It is easy to see from (5) that in this case formulas for input and output of sensor-based system (i.e. $\hat{\vec{u}}$ and $\hat{\vec{y}}$) are in the form:

$$\hat{\vec{u}} = \sum_{j=1}^{q} \vec{u}_j f_j^* \left(\sum_{j=1}^{q} f_j f_j^* + (\alpha + q)\Re \right)^{-1} \vec{y}, \ \alpha \in (0;1), \tag{6}$$

$$\hat{\vec{y}} = \aleph \hat{\vec{u}},$$

In the expression (6) $\aleph$ denotes an operator that is destined to model desirable processing of an odor. Usually such operator is defined by domain knowledge holder.

For approximate calculation of $\hat{f}_j = M\vec{y}_j$ in (6) statistical average can be used. It's quite common practical approach.

Pareto-optimization parameter α should be used for parity regulation between criteria of pareto-optimisation problem (3). This two criteria have opposite trends, i.e. while α increases the first criterion is reducing whereas the second criterion is increasing to $\sum_{j=1}^{q} \left\| Pu_j \right\|^2$. It's shown in [Zavorotnyy, 2004] that the criteria stick to conservation low $\varphi'(B(\alpha)) + \alpha h'(B(\alpha)) = 0$.

Indeed in terms of pareto-optimum problem (3) every single solution (5) has no benefits or disadvantages setting it against to any other solution (5). However in practice specific value of parameter α should be used. Let's present some common approaches to select α based on general principls of multi-criteria optimization [5].

For instance α can be found from minimization of sum of criteria, i.e. $\alpha = \arg\min_{\alpha} (\varphi(B(\alpha)) + h(B(\alpha)))$. From derivative of function $z(\alpha) = \varphi(B(\alpha)) + h(B(\alpha))$ it's easy to get stationary point. Taking into account conservation low presented above and the fact that $h'(B(\alpha)) < 0$ [Zavorotnyy, 2004] it's easy to see that $z'(\alpha) = h'(B(\alpha))(1-\alpha) = 0$ and the only possible stationary point is $\alpha = 1$. This is minimal value of $z(\alpha)$ since $z''(\alpha) = h''(B(\alpha))(1-\alpha) - h'(B(\alpha))$ and therefore $z''(1) = -h'(1) > 0$.

"Eldorado" principle can be used to select α too. According to the principle pareto-optimum parameter is resulted from minimization of sum of squared parameters of optimization, i.e. $\alpha = \arg\min_{\alpha} (\varphi^2(B(\alpha)) + h^2(B(\alpha)))$. In other words criteria α is selected as close as possible to coordinate origin in the space formed with all possible values of criteria $\varphi(B(\alpha))$ and $h(B(\alpha))$.

Conclusion

Let's highlight that identification of spirits has been chosen for instance. In the same way it is possible to qualify other odorants either mixes of odorants, also to identify whether an odor belongs to given group. Of course for each case appropriate odor samples are required while calibration measurements.

Therefore the developed approach once applied to output of sensor-based system brings identification to qualitatively new level. It is resistant to data errors. As a result it can be used in real identification systems and increase accuracy of identification. Moreover duy to low amount of computation this approach can be a valued addition to existent sensor-based systems with low impact to their response time. The proposed mathematical

processing can be implemented with and ran by PCs. Therefore it allows saving time and money since there is no need to create specific device with high identification accuracy.

To sum up the quality of volatile compounds identification can increased significantly due to the developed approach of mathematical processing of signals fed from sensor system given it has got optimum set of sensitive coating layers.

Acknowledgements

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Bibliography

[Eichelbaum, 1999] F. Eichelbaum, R. Borngraber, J. Schroder, R. Lucklum and P. Hauptmann, Interface circuits for QCM sensors, Review of Scientific Instruments, 70, may, 1999.

[Belov, 2001] Belov Yu. A., Zavorotnyy A. L., Kasyanyuk V. S. "On one approach to problem of processing of measurements using calibration signals based on multicriteria optimization", Journal of Automation and Information Sciences, 2001, #5 pp. 257-267.

[Pitiev, 1989] Pitiev Yu.P Mathematical methods of experiment interpretation, High school, Moscow, 1989.

[Zavorotnyy, 2004] Zavorotnyy, A. Decision of super-resolution measuring-calculating system modelling problem based on multicriterion optimization. – Bulletin of the University of Kiev, Series: Physics &Mathematics, 2004, #3, pp. 198-205.

Authors' Information

Andriy Zavorotnyy *– PhD, Researcher; TC science sector, theoretical cybernetics department, faculty of cybernetics, Kyiv National Taras Shevchenko University, Glushkov av. 2, build. 6, Kyiv-03127, Ukraine; e-mail: zalbxod@mail.ru*

Major Fields of Scientific Research: pareto-optimization, operator model of measuring-calculating system, fuzzy theory

Veda Kasyanyuk *– PhD, Head of Science Sector, TC science sector, theoretical cybernetics department, faculty of cybernetics, Kyiv National Taras Shevchenko University, Glushkov av. 2, build. 6, Kyiv-03127, Ukraine; e-mail: veda.sia@mail.ru*

Major Fields of Scientific Research: measurements' reduction to calculations, calibration of unknown measurements' model, spline approximation

ABOUT POSSIBILITY-THEORETICAL METHOD OF PIECEWISE-LINEAR APPROXIMATION OF FUNCTIONAL DEPENDENCIES IN PROBLEM OF ODOURS' RECOGNITION

Veda Kasyanyuk, Iryna Volchyna

***Abstract**: This paper considers the problem of recognizing and classifying the odorants to preset classes of volatile matters. It is assumed that the data registered by sensory elements and been liable to processing has been distorted by errors – fuzzy values. The possibility-theoretical method of piecewise-linear approximation of functional dependencies is proposed to solve the problem.*

***Keywords**: possibility-theoretical method, odorants, fuzzy errors.*

***ACM Classification Keywords**: I.6 Simulation and Modeling.*

Introduction

The problem of determining the composition of volatile matter and the quantitative characteristics of its elements, as well as the problem of classifying the tested odorants to preset classes of volatile matters are the important tasks arising under the problem of recognizing the odors. Very often the task of clarifying the composition of the volatile matter should be regarded as a sub-task of modeling the olfactory system.

One can choose several ways to create a "good" gas-analysis sensory system: optimization of gas-dynamic characteristics, the use of layers with better selective properties, the improvement of mathematical tools for processing the experiment data. Among the traditional methods used for detecting the gas mixtures, we should like to mention the method of principal component analysis (PCA), the discriminant analysis (DA) and the neural networks [Zieger, 1998, Jurs, 2000]. Unfortunately, these methods either do not provide sufficiently reliable identification or require the significant computing power. Furthermore, under the real conditions the data at the output of olfactory receptors are distorted by noise, and to analyze this data it is necessary to minimize their impact as much as possible. In authors' opinion, the best way out of this situation is to build a gas-analysis system within the concept of "device + PC = new possibilities", according to which the desired effect should be achieved by mathematical processing of received experimental data. The use of such processing algorithms provides a whole number of advantages: these algorithms are robust to errors in the data, they provide more accurate results, and it is quite easily to implement them by usual personal computers, i.e. there is no need to invest the substantial funds in creating a new device with improved characteristics. Furthermore, such algorithms do not require a large number of computations, i.e. it is possible to process the results of the measurements under real time.

Method of piecewise-linear approximation of functional dependencies by fuzzy data

The mathematical formulation of the problem is the following. We assume that the odor is represented by the vector of the concentrations of elementary odorants. The result of the device (gas-analyzer) measurement of some volatile compound is a certain number of numerical sequences which reflects the changes in time of sensors' responses on the tested matter; the number of sequences is equal to the number of sensors. The

problem is the following: it is necessary to find the vector of elementary odorants, i.e. the odor by sequences of sensors' responses on this matter. The method proposed in this paper requires a stage of training when known compounds (they may or may not be the same as those which later are submitted at the input for recognizing) are associated with corresponding sensors' response on it.

To solve such problem of recognizing the odors we should use the method of recovering the functional dependencies proposed in [Pitiev, 2000]. It will be about the problem of approximating the function $y(\cdot): T \to Y$ belonging to the known class of functions (linear in this case) by the results of observing the values of its argument $(t_1,...,t_N)$ and corresponding values of the function $y_1,...,y_N$, and the data of observation isn't precisely known.

We should assume that the number of sensors of the gas analyzer (the basic odorants) is equal to *l*, and concentration of the odorant fixed by sensors of the gas analyzer is a linear function of time. Then, at the output of the device which measures the concentration of some volatile compound we observe *l* vectors: $y_n^{(s)}$, $s = \overline{1,l}$ at moments of time t_n, $n = \overline{1,N}$.

We divide the received data into *k* groups by intervals Δ_p, $p = \overline{1,k}$ and there are m_p, $p = \overline{1,k}$ observations in each interval. We denote them as $y_{p,i}^{(s)}, i = \overline{1,m_p}$. For each interval we choose the following linear model of relationship between *y* and *t*:

$$y_p^{(s)}(t) = a_{p,1}^{(s)} t + a_{p,2}^{(s)}, \quad p = \overline{1,k}, \; s = \overline{1,l}$$

and assume that the values of the argument $t_{p,i}^{(s)}$, $i = \overline{1,m_p}$ are precisely known, and corresponding values $y_{p,i}^{(s)} = y_p^{(s)}(t_{p,i}^{(s)})$, $i = \overline{1,m_p}$ are modeled as the values of coordinates of the fuzzy vector:

$$\eta_{p,i}^{(s)} = a_{p,1}^{(s)} t_{p,i}^{(s)} + a_{p,2}^{(s)} + \nu_{p,i}^{(s)}, \quad i = \overline{1,m_p}, \; p = \overline{1,k}, \; s = \overline{1,l},$$

where $\eta_p^{(s)} = (\eta_{p,1}^{(s)},...,\eta_{p,m_p}^{(s)})$ - the fuzzy output vector ($y_{p,i}^{(s)}$ - its observed values), $\nu_{p,i}^{(s)}$ - the values of the fuzzy vector of errors $\bar{\nu}_p^{(s)} = (\nu_{p,1}^{(s)},...\nu_{p,m_p}^{(s)})$, which distribution is defined in the following way:

$$\pi^{\bar{\nu}}(\bar{z}) = \rho\left(\max_{1 \le j \le N} \frac{|z_i|}{\varepsilon_i}\right), \quad \varepsilon_j > 0, \; j = \overline{1,N},$$

$\rho: \Re^+ \to [0,1]$ is the continuous function strictly monotonically decreasing on $[0,1]$, which is equal to zero on $[1,\infty]$, $\rho(0) = 1$.

The values ε_i determine how much the error deviates from zero at $j-$th measurement. According to the possibility-theoretical method developed in [Pitiev, 2000] we should find such $a_{p,1}^{(s)}$ and $a_{p,2}^{(s)}$ which deliver a maximum of distribution

$$\pi^{\bar{\eta}_p^{(s)}}(\bar{y}_p^{(s)}, a_{p,1}^{(s)}, a_{p,2}^{(s)}) = \pi^{\bar{\nu}_p^{(s)}}(\bar{y}_p^{(s)} - a_{p,1}^{(s)}, \bar{t}_p^{(s)} - a_{p,2}^{(s)})$$

for each interval of the partition.

Since the input data can be considered as the values of the continuous function, we should require so that the approximating function satisfies the continuity condition:

$$a_{p-1,1}^{(s)} t_{p-1,m_{p-1}}^{(s)} + a_{p-1,2}^{(s)} = a_{p,1}^{(s)} t_{p,m_p}^{(s)} + a_{p,2}^{(s)}, \quad p = \overline{2,k}, \; s = \overline{1,l}.$$

So, to find the estimates $\hat{a}_{p,1}^{(s)}$ and $\hat{a}_{p,2}^{(s)}$ of maximum possibility [Pitiev, 2000] for each $s = \overline{1,l}$ at each interval of the partition Δ_p, $p = \overline{1,k}$ we have the following problem

$$\rho\left(\max_{\substack{1 \le i \le m_p \\ 1 \le p \le k}} \frac{| y_{p,i}^{(s)} - a_{p,1}^{(s)} t_{p,i}^{(s)} - a_{p,2}^{(s)} |}{\varepsilon_{p,i}^{(s)}} \right) \to \max_{a_{p,1}^{(s)}, a_{p,2}^{(s)}, p=\overline{1,k}}$$

$$\left\{ a_{p-1,1}^{(s)} t_{p-1,m_{p-1}}^{(s)} + a_{p-1,2}^{(s)} = a_{p,1}^{(s)} t_{p,m_p}^{(s)} + a_{p,2}^{(s)}, \quad p = \overline{2,k}, \right.$$

$$s = \overline{1,l}$$

Using the definition of the function $\rho(\cdot)$, we have

$$r = \max_{\substack{1 \le i \le m_p \\ 1 \le p \le k}} \frac{| y_{p,i}^{(s)} - a_{p,1}^{(s)} t_{p,i}^{(s)} - a_{p,2}^{(s)} |}{\varepsilon_{p,i}^{(s)}} \to \min_{a_{p,1}^{(s)}, a_{p,2}^{(s)}, p=\overline{1,k}}$$

$$\left\{ a_{p-1,1}^{(s)} t_{p-1,m_{p-1}}^{(s)} + a_{p-1,2}^{(s)} = a_{p,1}^{(s)} t_{p,m_p}^{(s)} + a_{p,2}^{(s)}, \quad p = \overline{2,k} \right.$$

$$s = \overline{1,l}$$

Let us transform the resulting problem to a standard linear programming problem:

$$r \to \min_{a_{p,1}^{(s)}, a_{p,2}^{(s)}, p=\overline{1,k}}$$

$$\begin{cases} a_{p-1,1}^{(s)} t_{p-1,m_{p-1}}^{(s)} + a_{p-1,2}^{(s)} = a_{p,1}^{(s)} t_{p,m_p}^{(s)} + a_{p,2}^{(s)}; p = \overline{2,k} \\ \dfrac{| y_{p,i}^{(s)} - a_{p,1}^{(s)} t_{p,i}^{(s)} - a_{p,2}^{(s)} |}{\varepsilon_{p,i}^{(s)}} \le r; p = \overline{1,k}, p = \overline{1,m_p} \end{cases}$$

$$s = \overline{1,l},$$

and to solve it we can use the simplex-method (or one of its modifications). The following vector is the solution of this problem:

$$\overline{a}^{(s)} = (a_{1,1}^{(s)}, a_{1,2}^{(s)}, \ldots, a_{k,1}^{(s)}, a_{k,2}^{(s)}), \quad s = \overline{1,l}$$

of the dimension $2 \cdot k$.

Constructing the solutions for linear programming problems for all $s = \overline{1,l}$ we obtain the vector:

$$\overline{a} = (a_{1,1}^{(1)}, a_{1,2}^{(1)}, \ldots, a_{k,1}^{(1)}, a_{k,2}^{(1)}, \ldots, a_{1,1}^{(l)}, a_{1,2}^{(l)}, \ldots, a_{k,1}^{(l)}, a_{k,2}^{(l)})$$

of the dimension $l \cdot 2 \cdot k$ which can be considered as a characteristic (characteristic vector) of the tested matter.

Computational experiment

Let us present one example of using the method of piece-linear approximation for recognizing the volatile matters by their odors. As mentioned above, at the beginning we have input the known matters and mixtures into the gas-analysys system. For each of them we should find the above-mentioned characteristic vector and enter it into the database. For the characteristic vectors we introduce the metric $\rho(A,B) = sqrt([A-B, A-B])$, where $[.,.]$ - scalar product, and then we choose the threshold of closeness δ which is constrained in the following way: if A and B are the characteristics of the known matters then $\rho(A,B) > 2\delta$.

Then, we input the unknown matter into the gas-analysis system and calculate its characteristic A . Among all characteristics from the database we should find such characteristic B that $\rho(A,B) < \delta$. If B exists then we conclude that the tested matter coincides with the matter which corresponds to B , otherwise, we assume that the composition of the tested matter differs very much from all matters from the database.

Let us consider the application of this method giving the example of recognizing the odor of the matter chlorine. During performing the computational experiment we used the data obtained by the gas-analysis sensory system developed at the Institute of Semiconductor Physics of the National Academy of Sciences of Ukraine [Shirshov, 2002, Kalchenko, 2002]. According to the experiment, the air under pressure with a high concentration of chlorine was input into gas-analysis system. Measurements were made using eight sensors at the time $t \in [0,72]$ as shown in Fig.1.

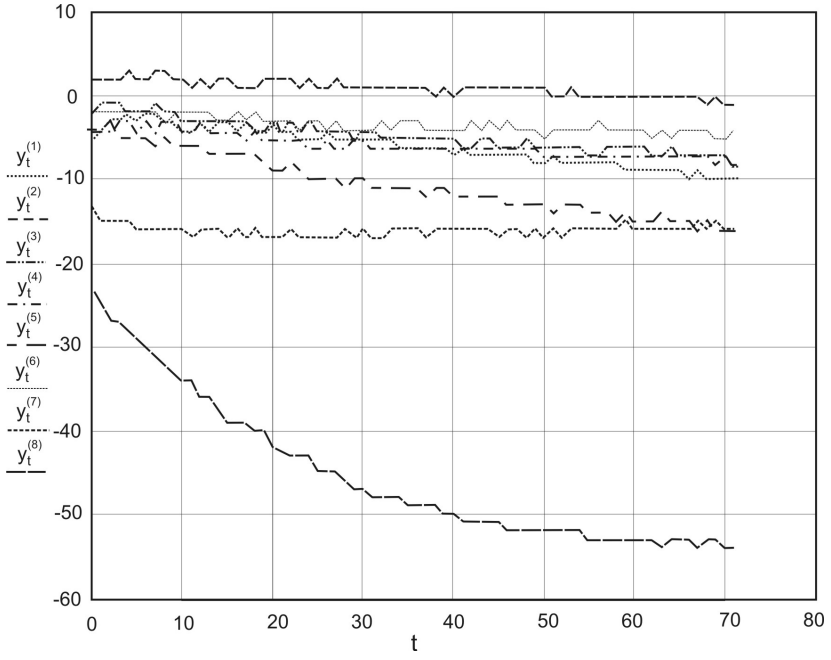


Figure 1. Measurement results for matter chlorine (first measurement)

After portioning the data into intervals, we considered 12 intervals with 12 measurements for each sensor and solved the canonical linear programming problem using the modified simplex-method. The resulting solution $\bar{a}^{CHLOR}$ were considered as a characteristic of the matter chlorine and consisted from 192 elements. Then, the back actions were conducted: the approximant of measurements' results were constructed in accordance with the vector $\bar{a}^{CHLOR}$ (see Fig. 2). Fig. 3-5 represents the results of comparing the real and modeled data $y_t^{(s)}$ for $s = 2,5,8$, i.e. for 2nd, 5th and 8th sensors, respectively.

The Figurers show that modeled data slightly deviates from measured ones.

We can improve the model by reducing the value of group intervals and reducing ε for each sensor, respectively, as well as by removing some of the time intervals at which measurements are questionable. Usually such decisions are made by the decision maker – the person, who knows much about physics of the process and can take into account the various factors and measurement conditions.

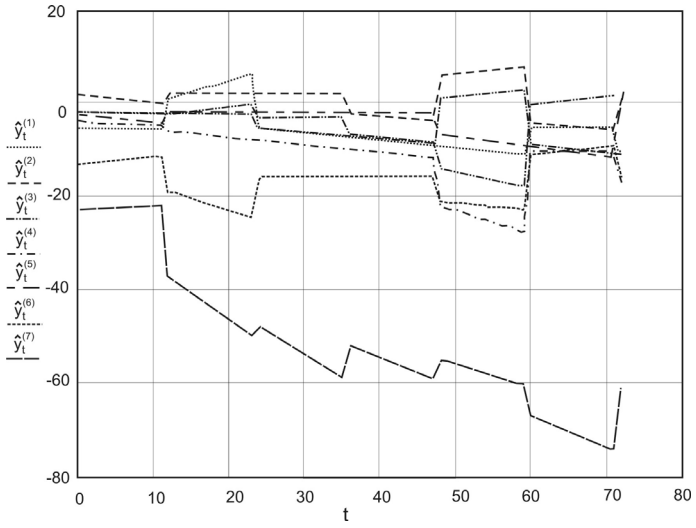


Figure 2. Approximation of measurement of chlorine

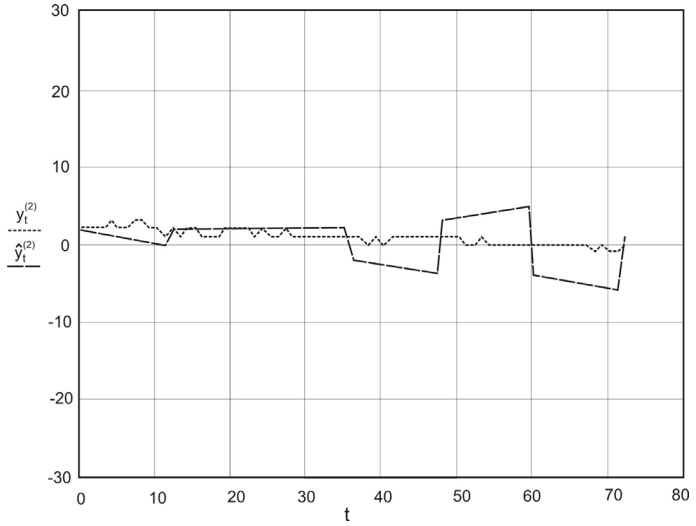


Figure 3. Data approximation by 2nd sensor

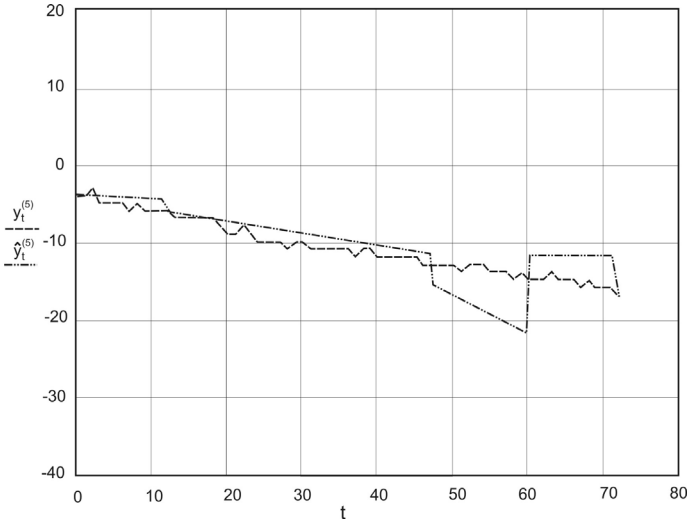


Figure 4. Data approximation by 5th sensor

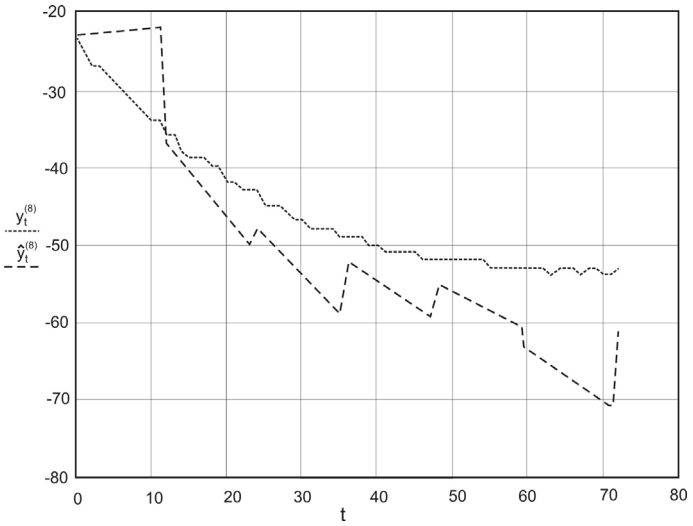


Figure 5. Data approximation by 8th sensor

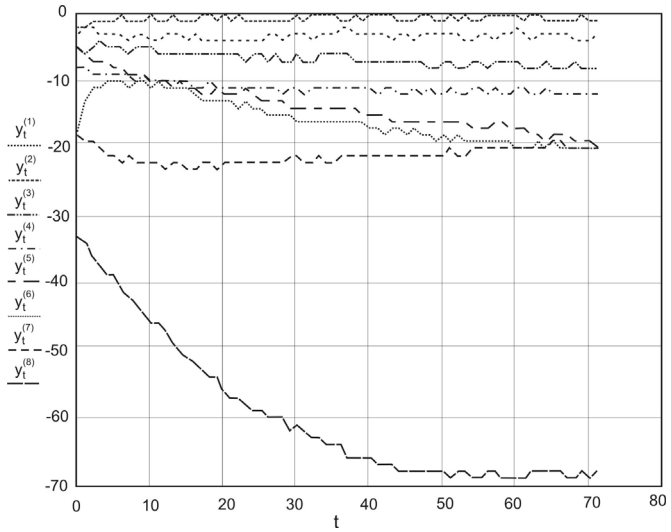


Figure 6. Measurement results for matter chlorine (second measurement)

After that we consider the problem of recognition of two matters. We should measure the matter chlorine once more and obtain other experimental data at the output of the gas analyzer (see Fig. 6). This data do not differs very much from the data presented in Fig. 1, and also the expected model should not differ very much from the previous one.

Let us construct the vector $\bar{a}^{CHLOR\_2}$ by the second measurement of chlorine and find the deviation of one measurement from another by their characteristic vectors $\bar{a}^{CHLOR}$ and $\bar{a}^{CHLOR\_2}$: $|| \bar{a}^{CHLOR} - \bar{a}^{CHLOR\_2} ||= 28.833$.

Then we input the matter from another class which differs very much from chlorine, for example, brandy, into the gas-analyser. The measurement results are shown in Fig. 7.

After constructing the vector $\bar{a}^{TAW}$ by measurements of the matter brandy we should find the deviation $\bar{a}^{CHLOR}$ from $\bar{a}^{TAW}$: $|| \bar{a}^{CHLOR} - \bar{a}^{TAW} ||= 290.927$. One can see that difference between the characteristics of chlorine and brandy is by an order greater than the difference between the characteristics of the different measurements of chlorine, i.e. at the low δ chlorine and brandy differ by the algorithm, which confirms the efficiency of its use.

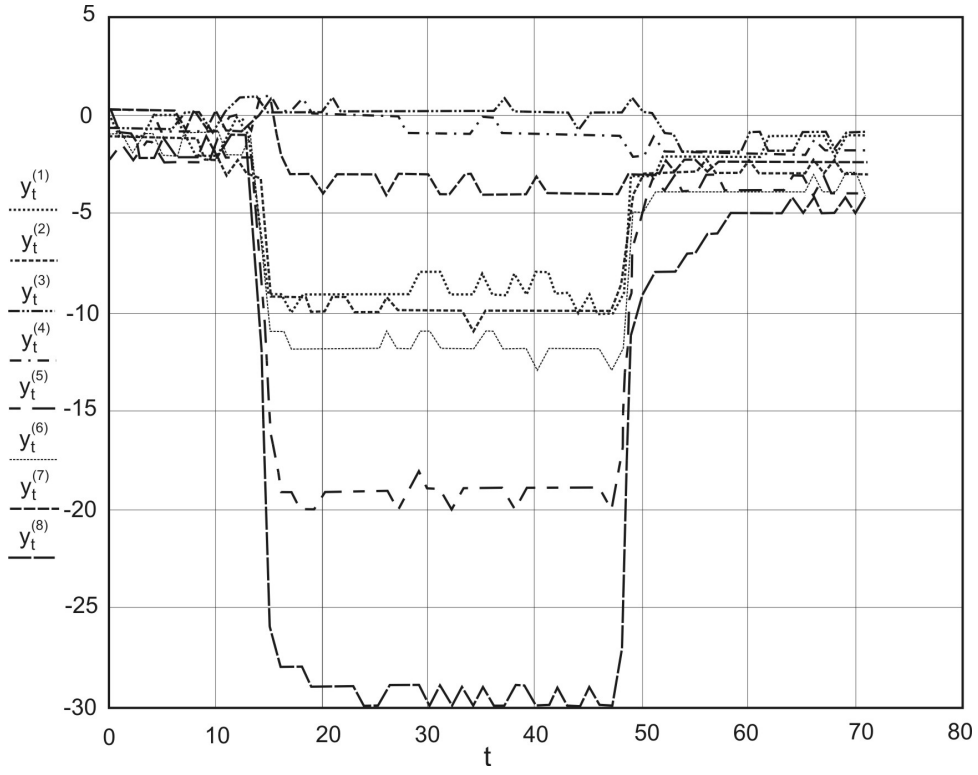


Figure 7. Measurement results for matter brandy

We would like to mention that the recognition of brandy has been chosen only as an example. In much the same way, one can recognize other odorants and mixtures of odorants, classify the odorants to classes, naturally, after the corresponding measurements, and under certain conditions it is also possible to recognize volatile matters, previously conducted a stage of training on the components of these compounds.

Conclusion

Thus, the method of piece-linear approximation by fuzzy data proposed in the paper allows recognizing the odors regardless of the concrete sensory systems. It has become possible thanks to the stage of training, during which we use the information only on the results of the test measurements without taking into account the complex internal self-structure of the gas-analysis sensory system – the data provider. Another advantage of this method is the fact that from the very beginning the method has supposed an occurrence of errors in the data, and the recognition is conceptually focused on minimizing the need of errors (the estimates of maximum possibility should be constructed).

The method proposed in this paper is quite simple to implement and it allows recognizing in real time. Thus, the use of this method together with optimally selected sensitive coatings of sensors can improve very much the process of recognizing the volatile matters and molecules by gas-analysers.

Acknowledgements

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Bibliography

[Zieger, 1998] Zieger Ch. et al. Bioelectronic noses: a status report. Part II // Biosensors & Bioelectronics. – 1998. – №13. – P. 539-571.

[Jurs, 2000] Jurs P.C., Bakken G.A., McClelland H.E. Computational Methods for the Analysis of Chemical Sensor Array Data from Volatile Analytes // Chemical Reviews. – 2000. – Vol. 100, – P. 2649-2678.

[Pitiev, 2000] Pitiev Yu.P The possibility. The elements of theory and applicationsб, – Moscow. Editotial URSS 2000. – (in Russian).

[Shirshov, 2002] Shirshov Yu.M., Koshets I.A., Kopylov O.N. Gas-dynamic parameters effect on response of gas-analysis sensory system. // Optoelectronics and semiconductor engineering. 2002. – № 37. – С. 153-168 (in Russian)

[Kalchenko, 2002] Kalchenko V.I., Koshets I.A., Matsas E.P., Kopylov O.N., Solovyov A.V., Kazantseva Z.I., Shirshov Yu. M. Calixarene-based Acoustical Sensors Array And Its Response to Volatile Organic Vapours // Material Science. – 2002. – Vol. 20, № 3. – P. 71-88.

Authors' Information

Veda Kasyanyuk *– PhD, Head of Scientific Sector, theoretical cybernetics sector, theoretical cybernetics department, faculty of cybernetics, Kyiv National Taras Shevchenko University, Glushkov av. 4-D, Kyiv-03127, Ukraine.*

Iryna Volchyna *– Engineer; theoretical cybernetics sector, theoretical cybernetics department, faculty of cybernetics, Kyiv National Taras Shevchenko University, Glushkov av. 4-D, Kyiv-03127, Ukraine, e-mail: volchyn@voliacable.com*

SOFTWARE FOR THE RECOGNITION OF POLYHEDRON CONTOUR IMAGES IN THE FRAMEWORK OF LOGIC-OBJECTIVE RECOGNITION SYSTEM

Natalya Bondar, Tatiana Kosovskaya

Abstract. The paper is devoted to the implementation of logic-objective approach to the solving of a polyhedron contour images (in particular partially covered images) recognition problem in a complex scene represented on the display screen. A way of predicate value calculation for representation the display screen is described in the paper. Examples of a program run constructing descriptions of both separate pictures and classes of objects are presented. For recognition of partially covered objects on the complex scene the concept of partial deducibility is used. Additionally the certainty level of the correct recognition is calculated.

***Keywords**: artificial intelligence, pattern recognition, predicate calculus.*

***ACM Classification Keywords**: I.2.4 ARTIFICIAL INTELLIGENCE Knowledge Representation Formalisms and Methods – Predicate logic.*

Introduction

The problem of recognition and analysis of a situation is one of the central problems of artificial intelligence. To formalize such a problem a logic-objective approach is proposed in [1]. This approach uses a language of predicate calculus and logical deduction in it.

Difficulties that appear during solving the problem of an object recognition in a complex scene are connected with the presence of a "vicious circle": to distinguish an object on the scene it is necessary first of all to single it out. And to single it out it is necessary to recognize it. On the first approach it seems that the way out from the "vicious circle" is only one – the full exhaustion of all elements of the image situated on the screen. Although, a deeper analysis of the problem allows to formulate and solve it as a problem of logical deduction search [1].

Just such an approach is implemented by programs represented in this paper.

Logic-objective approach to the pattern recognition problems

Let Ω be a set of finite sets $\omega = \{\omega_1,...,\omega_n\}$, that will be called below recognizable objects (here $t = t(\omega)$, that is in different sets there may be a different number of its elements). Any subset (not necessarily proper) τ of the set ω will be called its part. Let also a collection of predicates $p_1,...,p_n$ that characterizes properties and relations between elements of object ω Is done.

Let Ω be a union of K (may be intersected) classes $\Omega = \bigcup_{k=1}^{K} \Omega_k$.

Below the notation $\bar{y}$ is used for a list of all the elements from the set y. In particular, $\exists \bar{y}_{\neq}(P(\bar{y}))$ means that there is the set of values $\bar{y} = (y_1,...y_m)$ that $y_1 \in \omega \,\&\, ... y_m \in \omega$ (m is the number of free variables in formula P) for which the formula P is valid and such a formula is used as a notation for the formula $\exists y_1,...\exists y_m(\&_{i=j}... \& \, y_1 \neq y_j \,\&\, P(y_1,...y_m))$.

A set of all the true constant formulas of the type $p_i(\bar{\tau})$ or $\neg p_i(\bar{\tau})$ written out for all possible parts τ of the object ω will be called logical description $S(\omega)$ of the object ω.

A formula $A_k(\bar{x})$ with free variables $\bar{x}$ is called a logical description of the class Ω_k if

1. $A_k(\bar{x})$ contains only formulas of the type $p_i(\bar{y})$ as atomic (where $y \in x$);
2. $A_k(\bar{x})$ does not contain quantifiers;
3. $A_k(\bar{x})$ is a disjunction of elementary conjunction;
4. if for a list (an ordered set) $\bar{\omega}$ of all the elements of the set ω the formula $A_k(\bar{\omega})$ is valid then $\omega \in \Omega_k$.

Using the created descriptions the following pattern recognition problems may be solved. **The identification problem:** to check if the object ω or its part belongs to the class Ω_k. **The problem of classification:** to find all the numbers k of classes Ω_k such that $\omega \in \Omega_k$. **The analysis of a complex object problem:** to find and to classify all parts τ of the object ω for which $\tau \in \Omega_k$. The solution of these problems is reduced to the proof of the formulas $S(\omega) \Rightarrow \exists x_{\neq} A_k(x)$, $S(\omega) \Rightarrow \vee_{k=1}^{K} A(x)$, $S(\omega) \Rightarrow \vee_{k=1}^{K} \exists x_{\neq} A_k(x)$ respectively.

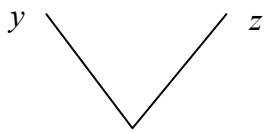
The solution of every of these problems is based on the proof of a logical sequence

$$S(\omega) \Rightarrow \exists x_{\neq} A(x) \qquad (1)$$

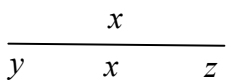
where $A(x)$ is an elementary conjunction.

Setting of a polyhedron contour images recognition problem

Consider a set of display screen images formed by segments of a straight lines defined by their ends. Two predicates V and L are done and defined in the following way.



$V(x,y,z) \leftrightarrow "\angle yxz < \pi"$



$L(x,y,z) \leftrightarrow "x$ lies on the segment with the ends y and $z"$

Two classes of objects are done: Ω_1 – a class of images of "boring machines", Ω_2 – a class of images of "turning machines". Examples of standard images are represented in the Fig. 1.

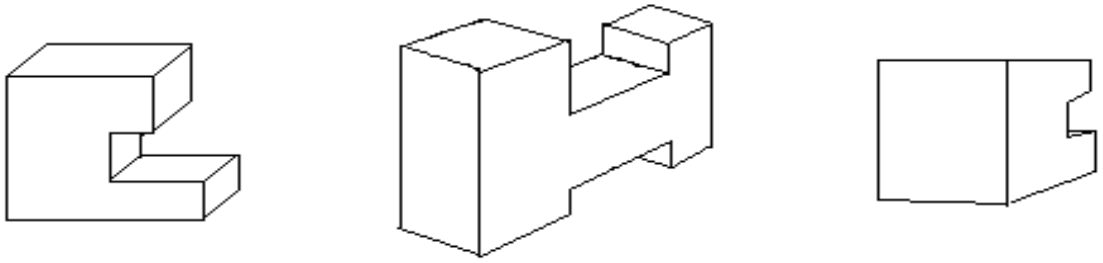


Fig. 1. Examples of standard images

Every image is represented on the display screen and defined by an intensity matrix.

Description of pictures and classes

According to the intensity matrix defining the image on the display screen we extract segments of straight lines forming a contour image. In such a case the straight line is represented on the screen of the display as a step figure. To resolve this problem a block of a program that implements smoothing was developed. In particular there were introduced two parameters: parameter *length* which permits to identify points the distance between which is less than the value of this parameter; and parameter *eps* that determines the minimal distance when the point belongs to the line.

If the real crosspoint of lines was displayed by three pixels then the gluing of these pixels was carried out.

In allocating the points of intersection that satisfy the predicate L it often occurred that the point of intersection of real lines are absent on the screen. To resolve the problem the gluing of the pixels was done.

After defining atomic formulas that are valid for the the image we receive the necessary set $S(\omega)$.

To obtain the description of the class according to the description of a standard object an elementary conjunction is build. In such a conjunction all the atomic formulas are obtained from the atomic formulas of object's description by replacement of different constants by different variables. The description of a class is a disjunction for all the standard objects of elementary conjunctions received in such a way.

Example of the program run creating descriptions of objects and classes

The lines forming the contour image on the display screen are allocated and the values of the initial predicates are calculated.

In the left upper corner in Fig. 2, 3 and 4 there are objects to be analyzed. In the lower part there is the result of allocation of the contour image vertxes by the program. In the right window there is the description of the picture.

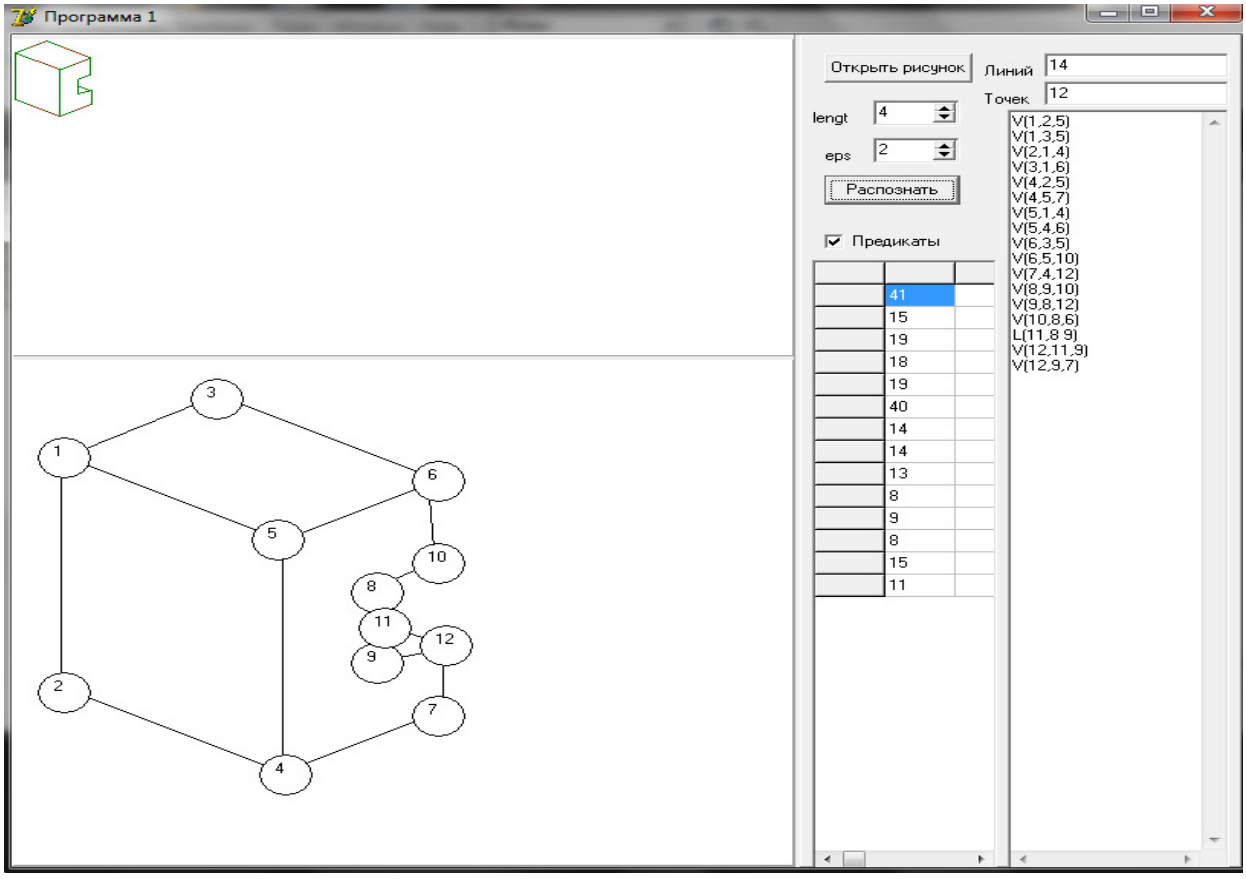


Fig. 2. The result of the program run creating the object description.

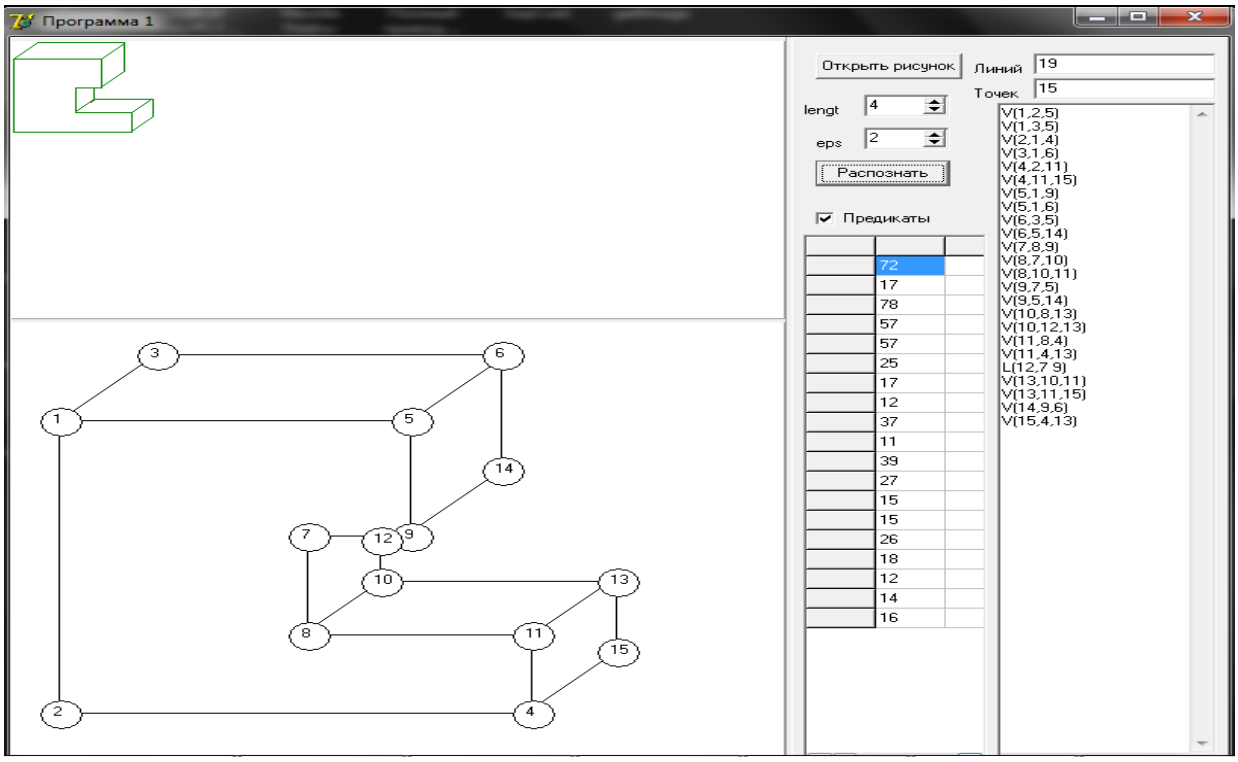


Fig. 3. The result of the program run creating the object description.

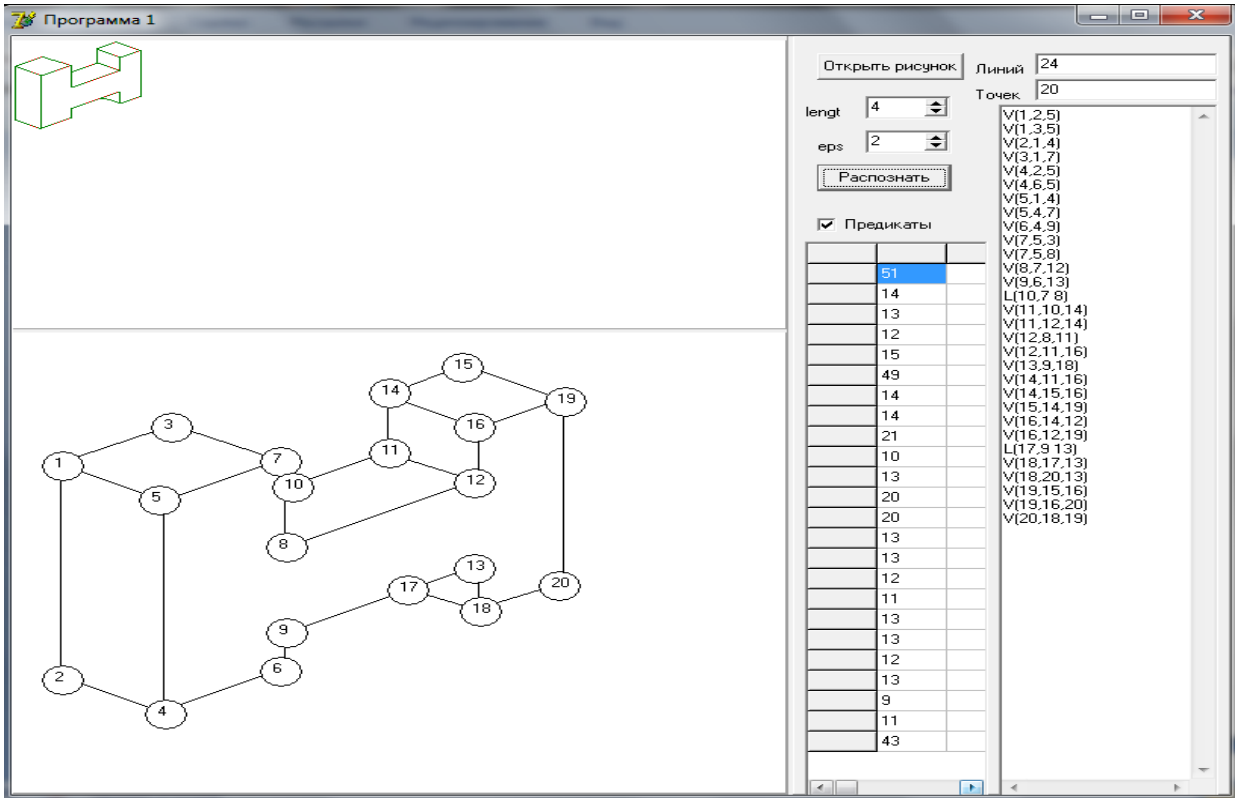


Fig. 4. The result of the program run creating the object description.

For the efficient run of the recognition program predicates shall be grouped in groups with the same names both in the set $S(\omega)$ and in the description of the class [2].

For the images in Fig. 2, 3 and 4 the following descriptions were respectively obtained:

$$S_1(\omega) = \{V(1,2,5), V(1,3,5), V(2,1,4), V(3,1,6), V(4,2,5), V(4,5,7), V(5,1,4), V(5,4,6), V(6,3,5), V(6,5,10), V(7,4,12), V(8,9,10), V(9,8,12), V(10,8,6), L(11,8,9), V(12,11,9), V(12,9,7)\}$$

$$S_2(\omega) = \{V(1,2,5), V(1,3,5), V(2,1,4), V(3,1,6), V(4,2,11), V(4,11,15), V(5,1,9), V(5,1,6), V(6,3,5), V(6,5,14), V(7,8,9), V(8,7,10), V(8,10,11), V(9,7,5), V(9,5,14), V(10,8,13), V(10,12,13), V(11,8,4), V(11,4,13), L(12,7,9), V(13,10,11), V(13,11,15), V(14,9,6), V(15,4,13)\}$$

$$S_3(\omega) = \{V(1,2,5), V(1,3,5), V(2,1,4), V(3,1,7), V(4,2,5), V(4,6,5), V(5,1,4), V(5,4,7), V(6,4,9), V(7,5,3), V(7,5,8), V(8,7,12), V(9,6,13), V(10,7,8), V(11,10,14), V(11,12,14), V(12,18,11), V(12,11,16), V(13,9,8), V(14,11,16), V(14,15,16), V(15,14,19), V(16,14,12), V(16,12,19), L(17,9,13), V(18,17,13), V(18,20,13), V(19,15,16), V(19,16,20), V(14,11,16), V(20,18,19)\}.$$

Recognition of the standard image

Check of $S(\omega) \Rightarrow \exists \bar{x}_{\neq} A_k(\bar{x})$, can be realized by a derivation in a predicate sequential calculus.

An example of the run of the program that recognizes standard images is represented in Fig. 5.

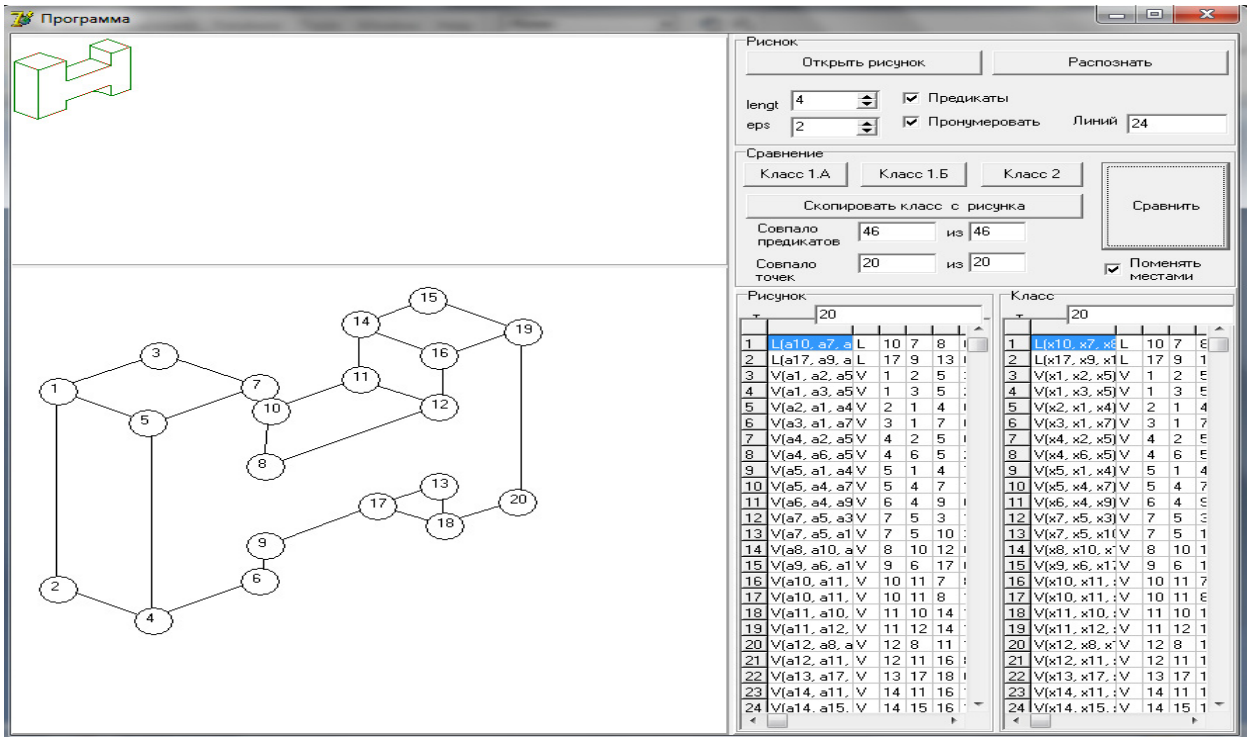


Fig. 5. Recognition of a standard image.

The object is recognized. In the left lower working window of the program result there is a description of the object. In the right lower working window there is a disjunctive member of the description of the class that is valid for this image. In the working windows above the descriptions the number of "coincided" predicates and arguments are given. In this example all 46 per 46 predicates and all 20 per 20 arguments coincide.

Recognition of partially covered image

Consider a solution of the above formulated recognition problems when not a full description $S(\omega)$ of the object ω is done but only a subset of it $\widetilde{S}(\omega) \subseteq S(\omega)$. Such a situation corresponds to the recognition of a partially covered object. To resolve the problem a concept of partial deduction was introduced in [3].

While solving the identification problem in spite of checking validity $S(\omega) \Rightarrow \exists \bar{x}_{\neq} A_k(\bar{x})$ one has a possibility to check only $\widetilde{S}(\omega) \Rightarrow \exists \bar{x}_{\neq} \widetilde{A}_k(\bar{x})$ where $\widetilde{A}(\bar{x})$ is a sub-formula of the formula $A_k(x)$. Besides that the "reminder" of the formula $A_k(x)$ must not be in the contradiction with $\widetilde{S}(\omega)$.

Let a and a' be the numbers of atomic formulas in $A(x)$ and $A'(x')$ respectively, m and m' be the numbers of objective variables in $A(x)$ and $A'(x')$ respectively. Then partial deduction means that the object ω is an r-th part ($r = m'/m$) of an object satisfying the description $A(x)$ with the certainty $q = a'/a$.

More precisely, the formula $S(\omega) \Rightarrow \exists x_{\neq} A_k(x)$ is partially (q,r)-deductive if there exists a maximal sub-formula $A'(x')$ of the formula $A(x)$ such that $S(\omega) \Rightarrow \exists x'_{\neq} A'(x')$ is deducible and τ is the string of values for the list of variables x' but the formula $S(\omega) \Rightarrow \exists x_{\neq} [DA'(x)]^{x'}_{\tau}$ is not deducible. Here $[DA'(x)]^{x'}_{\tau}$ is obtained from $A(x)$ by deleting from it all conjunctive members of $A'(x')$, substituting values of τ instead of the respective variables of x' and taking the negation of the received formula.

The result of solving the identification problem of "turning machine" (i. e. an object of the 1st class) is presented on the Fig. 6. The first object was identified almost completely because 20 points (points 1,...,20) taken for arguments of the 1st class description satisfy 26 atomic formulas of 30 in the class description. Hence, the object defined by points 1,...,20 with the certainty 26/30 belongs to the 1st class.

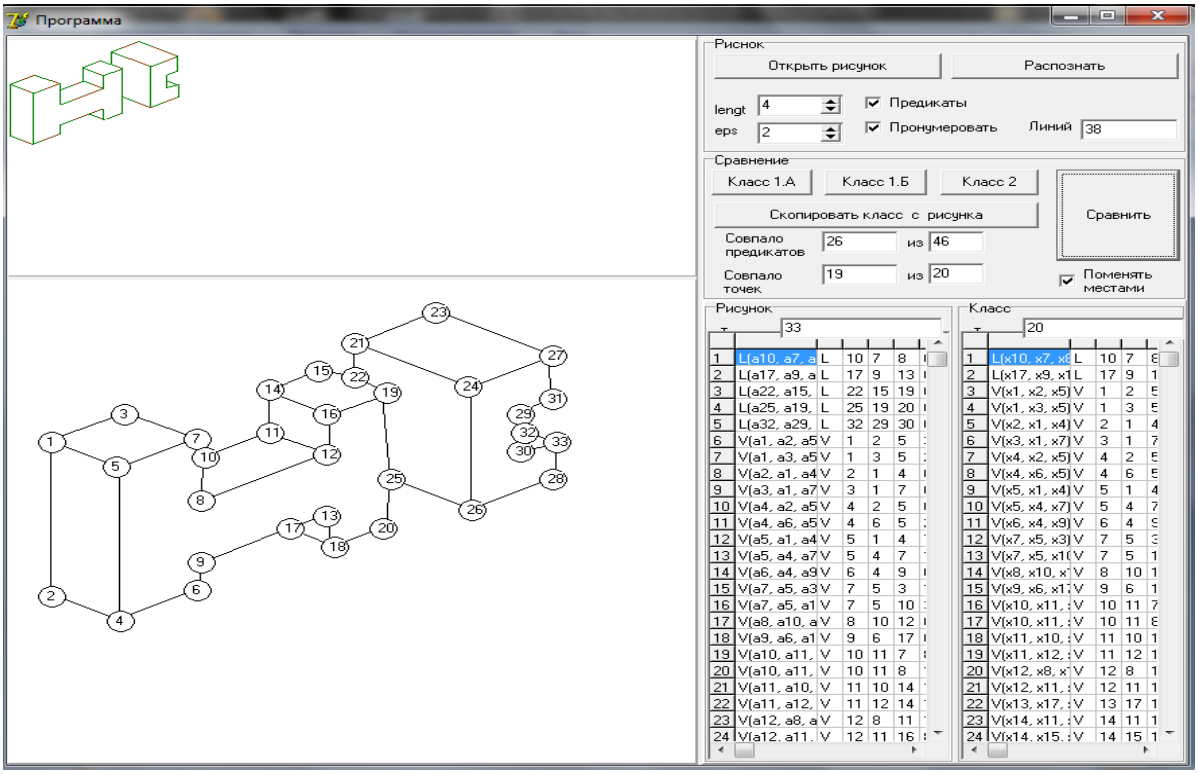


Fig. 6. Recognition of partially covered contour image

The result of solving the identification problem of "drilling machine" (i.e. an object from the 2-nd class) is presented on the Fig. 7. Points 21,23,24,26,...,33 and 22 or 25 taken for arguments of the 2nd class description satisfy 16 atomic formulas of 17 in the class description. Hence, the object defined by points 21,23,24,26,...,33 and 22 or 25 with the certainty 16/17 belongs to the 2ndclass.

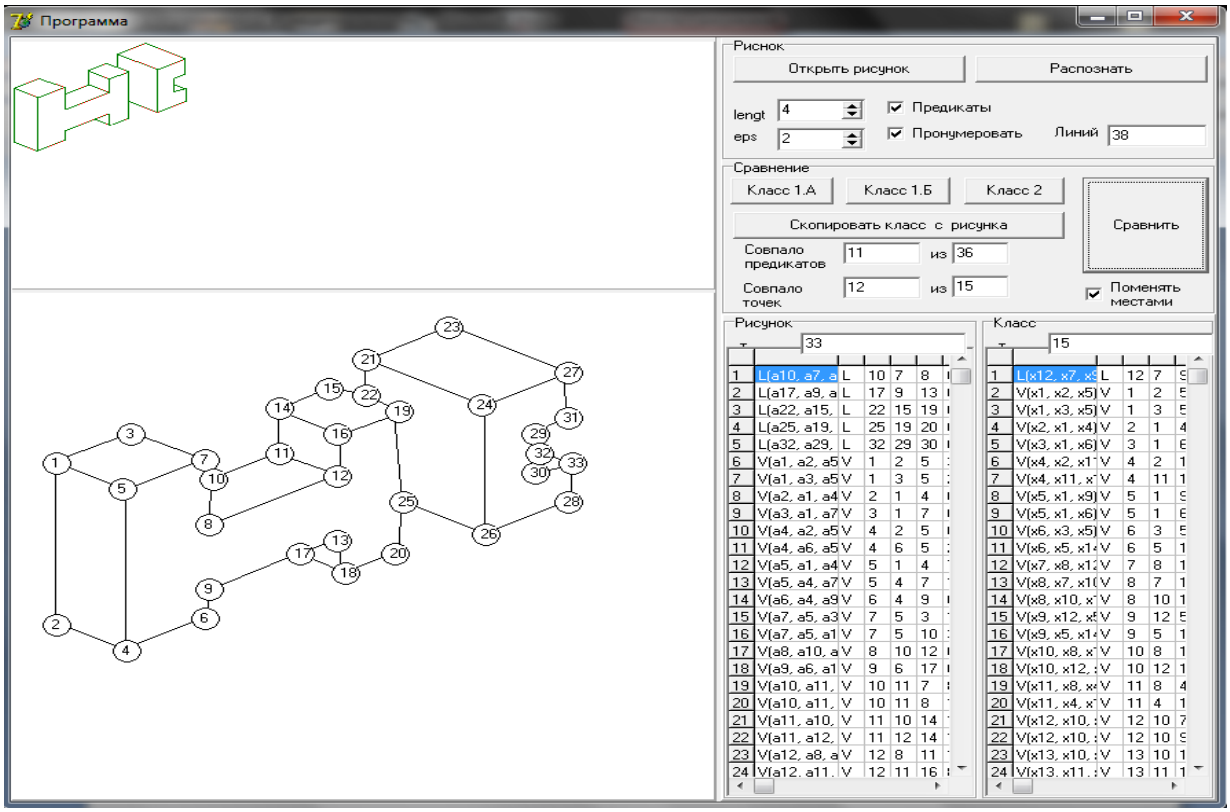


Fig. 7. Recognition of partially covered contour image

Conclusion

The proposed in [1] logic-objective approach permits not only to recognize separate contour objects on a display screen but also to allocate and to recognize in the same time objects in the complex scene. The use of the concept of partial deducibility [3] permits to recognize partially covered objects and to calculate the degree of certainty of their correct recognition.

Bibliography

[1] T.M. Kosovskaya, A.V. Timofeev. About one new approach to the formation of logical decision rules in pattern recognition problems. In: Vestnik LGU, 1985, No. 8. P. 22 – 29. (In Russian)

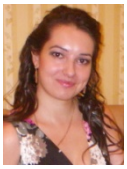
[2] T.M. Kosovskaya. Proofs of the number of steps bounds for solving of some pattern recognition problems with logical description. In: Vestnik of St.Petersburg University, Ser. 1, 2007. No. 4. P. 82 – 90. (In Russian)

[3] T.M. Kosovskaya. Partial deduction of a predicate formula as an instrument for recognition of an object with incomplete description. In: Vestnik of St.Petersburg University, Ser. 10, 2009. No. 1. P. 74 – 84. (In Russian)

Authors' Information

Tatiana Kosovskaya *– Dr., Senior researcher, St.Petersburg Institute of Informatics and Automation of Russian Academy of Science, 14 line, 39, St.Petersburg, 199178, Russia; Professor of St.Petersburg State Marine Technical University, Lotsmanskaya ul., 3, St.Petersburg, 190008, Russia; Professor of St.Petersburg State University, University av., 28, Stary Petergof, St.Petersburg, 198504, Russia, e-mail: kosov@NK1022.spb.edu*

Major Fields of Scientific Research: Logical approach to artificial intelligence problems, theory of complexity of algorithms.



Natalia Bondar – PHD student, St.Petersburg Institute of Informatics and Automation of Russian Academy of Science, 14 line, 39, St.Petersburg, 199178, Russia;, e-mail:

Major Fields of Scientific Research: Logical approach to pattern recognition problems.

The paper is published with financial support of the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

PRODUCT QUALITY ANALYSIS USING SUPPORT VECTOR MACHINES

A. Nachev, B. Stoyanov

***Abstract**: This paper presents an exploratory study of the effectiveness of support vector machines in the prediction of a product quality based on its characteristics. The study answers the following three questions: how does the choice of kernel and model parameters affect the predictive abilities of support vector machines; can an alternative subset of variables be unearthed that can be used in order to increase the predictive abilities of the data mining model; how will the removal of potential outliers affect the predictive abilities of the data mining model. We used a dataset of red and white wine samples presented by their physiochemical characteristics. Findings show that a correct selection of kernel and appropriate variable selection technique may have a significant impact on the prediction ability of the data mining model. Certain model settings can even make it to outperform the best technique reported thus far in the application area.*

***Keywords**: data mining, support vector machines, sensory preferences, variable selection, wine classification.*

***ACM Classification Keywords**: I.5.2- Computing Methodologies - Pattern Recognition – Design Methodology - Classifier design and evaluation.*

Introduction

Many studies have used data mining methods to predict quality outcomes of a range of products, based on data available for these products. A variety of domains and applications range from wine quality prediction [Cortez et al., 2009, Beltran et al., 2008, Bapna and Gangopadhyay, 2006, Fei et al., 2008, Li et al., 2010], manufacturing [Xiaoh, 2009, Deh, 2008], water quality prediction [Wang et al., 2010], textiles quality prediction [Selvanayaki et al., 2010], to image quality prediction and image steganography [Hsien-Chu et al., 2008, Narwaria and Lin, 2010]. Many of these studies have used support vector machines (SVM) for data analysis. Researchers report that SVM show superior predictive power to other data mining methods and techniques used in the domains.

This paper presents an exploratory study of the effectiveness of SVM in the prediction of wine quality, based on the physiochemical components thereof. Within the creation and marketing of wine, certification and quality assessment is of great importance, for both health considerations and quality assurance. Quality assessment, in effect is a contributing factor used in determining the price of wine. According to a study conducted by “Wine Business Monthly”, the salaries of wine tasters at vineyards accounts for a quantifiable proportion of expenditure [Tinney, 2006]. Yet human error can be a diminishing factor in the accuracy of this assessment. This opens up avenues for data mining as a good quality control process in the assessment of wine [Cortez et al., 2009]. It is based on these observations that the inherent business value of data mining physiochemical characteristics for predicting product quality becomes evident. Using data mining in the field would allow wine producers to migrate this expensive job function over to a technological platform.

The use of support vector machines in the prediction of wine quality is still in an early stage, yet initial studies within this domain have yielded promising results. Bapna and Gangopadhyay [2006] displayed that SVM exceed both Naive Bayes and Adaptive Bayes in the classification of wine with results based on performance estimation by the classification accuracy metric alone.

Beltran et al. [2008] utilise SVM in addition to, and in comparison with, radial basis function neural networks (RBFNN) and linear discriminant analysis (LDA), in the classification of Chilean wine. The analyses are carried out on data derived from wine aroma chromatograms of three different Chilean wine varieties. Two dimensionality reduction techniques were incorporated, namely principal component analysis (PCA), and wavelet transformation (WT). This work can also be extedned towards using various preformance metrics and different kernel types. Li et al. [2010] proposed use of star-graphs to study behaviour of variables in wine classification. These graphs provided a means of visualization of an instance, taking into account all variables simultaneously. Fei et al., [2008] utilized least squares support vector machines (LS-SVM) on physiochemical data of red wine samples obtained obtained through the use of visible and near infrared (Vis/NIR) transmittance spectroscopy. Cortez et al., [2009] discussed data mining techniques to be used in the prediction of wine taste preferences also. Utilizing a large dataset of Portuguese “vinho verde” samples, three regression techniques were used, namely, SVM, multiple regression (MR) and backpropagation neural networks (BPNN). In utilising the SVM technique, the authors adopted the Gaussian kernel, yet there is little descripton on how kernel type, as a hyperparameter, influences the model preformance. This work can also be extended towards study of different techniques for reduction of dimensionality and selection of optimal subset of variables.

Finally, whilst using the SVM model, many authors state that three issues play significant role in the model preformance: attaining the optimal input subset, correct kernel function, and the optimal parameters of the selected kernel [Fei et al., 2008]. This provides implications to future work, which is addressed in this study.

The structure of this paper is as follows: Section 2 describes SVM as data mining tools; Section 3 describes the dataset used in the study and outlines the variable selection techniques as part of the data pre-processing; Section 4 briefly outlines the role of outliers in data mining; Section 5 describes the experimental results and discusses their meaning.

Support Vector Machines

Support vector machines have grown in status over the past decade due to the satisfactory results returned over a diverse range of fields. SVM are data analysis techniques categorised within the domain of supervised machine learning [Dash and Singhania, 2009, Salfner et al., 2010], whereby the learning process results in a function being contingent on the supervised training data. Through this supervised machine learning process, the algorithm returns either a classification function, or a regression function. A support vector regression procedure suggests an optimal trade off between complexity and learning ability in order to achieve a strong generalization of accuracy [Xiaoh, 2009].

For a two-class, separable training data set, such as the one in Figure 1, there are lots of possible linear separators. Intuitively, a decision boundary drawn in the middle of the void between data items of the two classes seems better than one which approaches very close to examples of one or both classes. While some learning methods such as the perceptron algorithm find just any linear separator, others, like Naive Bayes, search for the best linear separator according to some criterion. The SVM in particular defines the criterion to be looking for a decision surface that is maximally far away from any data point. This distance from the decision surface to the closest data point determines the margin of the classifier. This method of construction necessarily means that the decision function for an SVM is fully specified by a subset of the data points, which defines the position of the separator. These points are referred to as the support vectors. Figure 2 shows the margin and support vectors for a sample problem. Other data points play no part in determining the decision surface that is chosen.

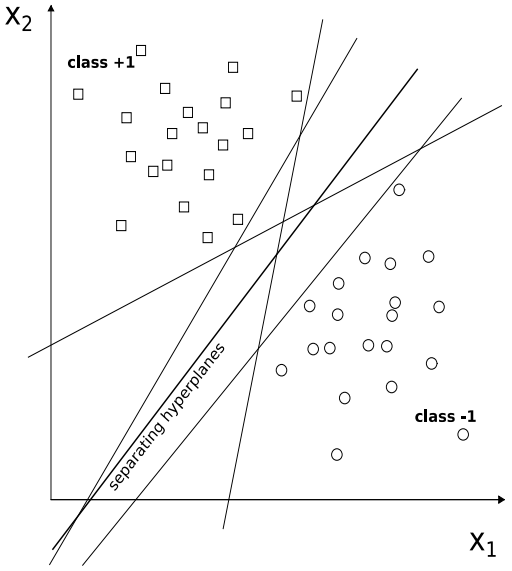


Figure 1. Separating lines for a two-class separable dataset

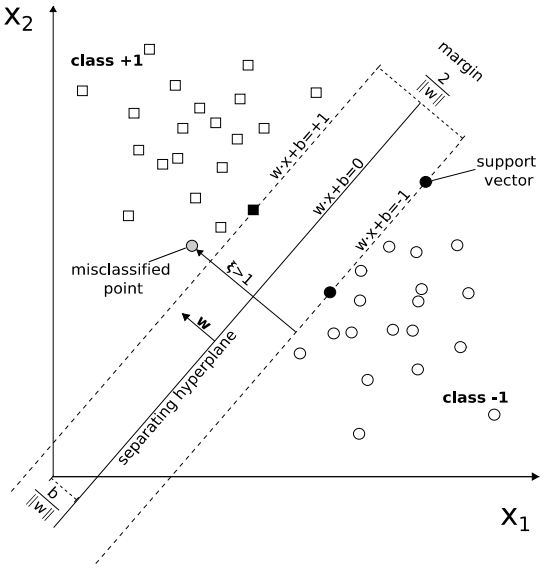


Figure 2. Geometry of support vector machines.

SVM can be formalized as follows. Training data of n samples is a set of pairs of data points $\vec{x}_i$ (p-dimensional vectors) and class labels y_i where -1 indicates one class; +1 the other class.

$$D = \{(\vec{x}_i, y_i) \mid \vec{x}_i \in \Re^p, y_i \in \{-1,+1\}\}_{i=1}^n \quad (1)$$

During training a SVM builds a decision boundary that separates the classes. The decision boundary is a p-1 – dimensional hyperplane (a line in the 2D case, a plane in the 3D case, etc.). A decision hyperplane can be defined by a normal vector $\vec{w}$ perpendicular to the hyperplane and a term b. The vector $\vec{w}$ is often called weight vector. The term b specifies the choice of hyperplane among all perpendicular to the normal vector. Because the hyperplane is perpendicular to the normal vector, all points x on the hyperplane satisfy

$$\vec{w}^T\vec{x} + b = 0. \quad (2)$$

Data points would fall into one or another side of the decision hyperplane turning the above equality into inequality, therefore the decision function of a linear SVM classifier can be defined as

$$f(\vec{x}) = sign(\vec{w}^T\vec{x} + b). \quad (3)$$

Class labels are +1, -1. The points closest to the separating hyperplane are called support vectors. The margin of a classifier is the maximum width of the band that can be drawn separating the support vectors of the two classes. It can be shown that maximizing the margin is the following minimization problem: find $\vec{w}$ and b such that

$$\frac{1}{2}\vec{w}^T\vec{w} \text{ is minimized and for all } \{(\vec{x}_i, y_i)\} \quad y_i(\vec{w}^T\vec{x}_i + b) \geq 1 \quad (4)$$

This task is optimization of a quadratic function subject to linear constraints. The solution of that problem involves constructing a dual form of the optimization problem where a Lagrange multiplier α_i is associated with each constraint $y_i(\vec{w}^T\vec{x} + b) \geq 1$ in the primal problem. The dual problem is: find $\alpha_i, \ldots, \alpha_N$ such that

$\sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j$ is maximized and $\sum_i \alpha_i y_i = 0$; $\alpha_i \geq 0$ for all $1 \leq i \leq N$ (5)

A solution of that problem allows building the decision hyperplane:

$$\vec{w} = \sum_i \alpha_i y_i \vec{x}_i \; ; \; b = y_k - \vec{w}^T \vec{x}_k \text{ for any } \vec{x}_k \text{ such that } \alpha_k \neq 0 \quad (6)$$

Most Lagrange multipliers found by the optimization problem are zero. Each non-zero indicates that it corresponds to a support vector. The classification function (2) can be presented in the form

$$f(\vec{x}) = sign(\sum_i \alpha_i y_i \vec{x}_i^T \vec{x} + b) \quad (7)$$

The above formulas that contain vectors also use dot product operation between them.

The simplest way to divide two classes is with a straight line in 2D, flat plane in 3D or an (N-1)–dimensional hyperplane in an N-dimensional attribute space. Sometimes, however, such a separation is impossible (as shown in Figure 3). Instead of fitting nonlinear curves (hyper-surfaces) to the data, an SVM can handle this using a kernel function that maps the data to a different higher dimensional space where a hyperplane can be used to do the separation. Indeed, if there are two data attributes (2D data points) and data set is not linearly separable by a line, the kernel function can add a third attribute in order to map the points into 3D, so that the data set could be linearly separable by a flat plane in 3D. It can be generalised that the kernel function transforms the data into a higher dimensional space to make separation by hyperplanes possible.

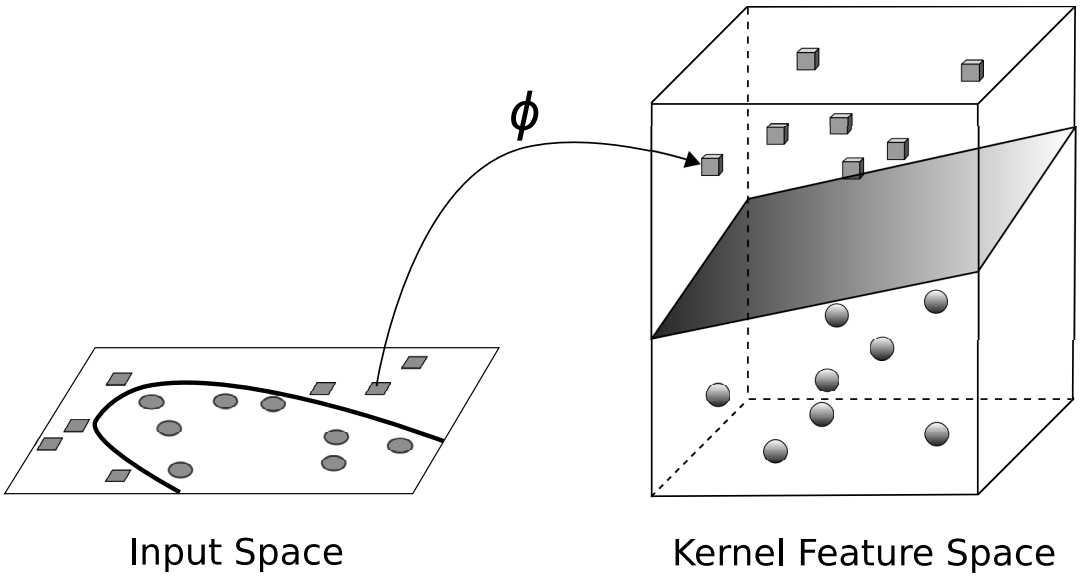


Figure 3. The kernel trick: a linearly inseparable input space can be mapped to a higher dimensional space, which is linearly separable.

The kernel function can be defined as

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i)\Phi(\vec{x}_j), \quad (8)$$

where $\Phi(\vec{x})$ maps the vector $\vec{x}$ to some other Euclidean space. The dot product $\vec{x}_i \cdot \vec{x}_j$ in the formulas above is replaced by $K(\vec{x}_i, \vec{x}_j)$ so that the SVM optimization problem in its dual form can be redefined as: maximize (in α_i)

$$\widetilde{L}(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j), \text{ subject to } \sum_i \alpha_i y_i = 0; \quad \alpha_i \geq 0 \text{ for all } 1 \leq i \leq N \quad (9)$$

Various kernel functions can be used with SVM and perhaps their number is infinite. But a few of them have been found to work well for a wide variety of applications. These are:

Linear: $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j$ *(10)*

Polynomial: $K(\vec{x}_i, \vec{x}_j) = (\gamma \vec{x}_i^T \vec{x}_j + r)^d, \gamma > 0$ (11)

Radial Basis Function (RBF) a.k.a. Gaussian kernel: $K(\vec{x}_i, \vec{x}_j) = \exp(\gamma \left\| \vec{x}_i - \vec{x}_j \right\|^2), \gamma > 0$ (12)

Sigmoid: $K(\vec{x}_i, \vec{x}_j) = \tanh(\gamma \vec{x}_i^T \vec{x}_j + r), \gamma > 0, r < 0$ (13)

Ideally, an SVM analysis should produce a hyperplane that completely separates the feature vectors into two non-overlapping groups. However, perfect separation may not be possible, or it may result in a model in so high dimensional space that the model does not generalize well. To allow some flexibility in separating the classes, the soft-margin SVM proposed by Cortes and Vapnik [1995] permit some misclassifications. The method chooses a hyperplane that splits data points as clean as possible while still maximizing the distance to the nearest cleanly split points. The method introduces slack variables ξ_i in $y_i(\vec{w}^T \vec{x}_i + b) \geq 1 - \xi_i$, $1 \leq i \leq n$, which measure the degree of misclassification of the points $\vec{x}_i$. If a training example lies on the 'wrong' side of the hyperplane, the corresponding ξ_i is greater than 1. Therefore, the primal form of the optimization problem is

$$\min_{w,\xi,b} \left\{ \frac{1}{2}\left\| w \right\|^2 + C\sum_{i=1}^{n} \xi_i \right\}, \text{ subject to } \forall_{i=1}^{n}: y_i(\vec{w}^T \vec{x}_i + b) \geq 1 - \xi_i; \forall_{i=1}^{n} \ \xi_i > 0 \quad (14)$$

The factor C in the formula is a parameter that represents the cost of misclassification. A small value of C will increase the number of training errors, while a large C will lead to a behavior similar to that of a hard-margin SVM. In that sense the cost parameter C that controls the trade-off between allowing training errors and forcing rigid margins.

The soft-margin optimization problem along with the constraint can be solved using Lagrange multipliers (as before) so that in a dual form it can be formulated as follows: minimize

$$\widetilde{L}(\alpha) = -\sum_{i=1}^{n} \alpha_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j), \text{ subject to: } \forall_{i=1}^{n}: \sum_{i=1}^{n} y_i \alpha_i = 0; \quad \forall_{i=1}^{n}: 0 \leq \alpha_i \leq C \quad (15)$$

The advantage of the dual form is that the slack variables vanish, with the parameter C appearing only as an additional constraint on the Lagrange multipliers.

The SVM method can also be applied to the case of regression. A version of SVM for regression, called support vector regression (SVR), was proposed by Drucker et al. [1997]. The basic idea of SVR is that a non-linear

function learns by a linear learning method in a kernel-induced higher dimensional space. Similarly to how SVM classification ignores data points that are not support vectors, the SVR depend on a small subset of training data points.

The SVM's major advantage lies with their ability to map variables onto an extremely high feature space. This, in essence facilitates a means for the exploration of nonlinear kernel-based classifiers [Oladunni and Singhal, 2009, Burges, 1998], however, they have been discovered to not favour large datasets, due to the demands it imposes on virtual memory, and the training complexity resultant from the use of such a scaled collection of data [Cortez et al., 2009, Horng et al., 2010].

Work from Fei et al. [2008] highlighted three "crucial problems" in the use of support vector machines. These are attaining the optimal input subset, correct kernel function, and the optimal parameters of the selected kernel, all of which are prime considerations within this study. Multiple authors also echoed sentiments of kernel selection problems [Wang et al., 2010, Selvanayaki et al., 2010, Petrujkic et al., 2008], which further indicated the importance of this factor for this research.

Dataset and Variable Selection

The data used in this study consists of two distinct sets, which represent the two most common variants of Vinho Verde wines, white and red. With regard to the red sample collection, data instances numbered 1599, while white instances totaled 4898. These instances held 12 variables respectively, relating to the physiochemical breakdown, namely: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and a quality rating. This quality rating was based on a sensory taste test carried out by at least three wine experts grading the wine quality on a scale between "0 (very bad) and 10 (very excellent).

The selection of a subset of variables, that return the best prediction rates in the SVM task, is another important consideration within an exercise in support vector machine prediction. Its importance lies in the need to develop predictive models from a reduced, minimal number of input variables which best summarize the overall input data resulting in maximal predictive power. Dimensionality reduction is the process undertaken in order to reduce the number of independent variables utilised within a data mining exercise. The variables within the dataset are examined to see if and how they relate to, and influence other variables. Fayyad et al., [1996] describe data reduction as the most tedious stage of data analysis. It is at this stage that the cataloguing, classification, segmentation and partitioning of data occur. However, in the aim of effectively discovering knowledge within datasets, reducing the dimensionality of the data is a required step and indeed, a fundamental tool for many data mining tasks. A synergetic benefit of dimensionality reduction is its tendency to reduce overfitting. Datasets possessing a high degree of dimensionality, i.e. a large quantity of variables, will be hindered by the choice of data mining method available to them. Effectively, the method used to reduce the dimensionality of a collection of data will influence the overall accuracy and effectiveness of the data mining exercise. Dimensionality reduction is considered application specific problem, which is not backed by a universal theory. It is as a major challenge in the data mining process as the 'best' variables in one data subset may not necessarily be the best in another; these best variables are, for the most part dependent on the model under employ.

As an indication of its importance within the realm of data analysis, there are many dimensionality reduction techniques, which have been proposed by researchers. These include, but are not limited to, Discrete Fourier Transformation (DFT), Singular Value Decomposition (SVD), Discrete Wavelet Transformation (DWT), Piecewise Aggregate Approximation (PAA), and Adaptive Piecewise Constant Approximation (APCA), etc. LDA and Principal Component Analysis (PCA) are widely popular methods of dimensionality reduction due to its simplicity

and effectiveness in comparison to others. Petrujkic et al., [2008] employ a particle swarm optimization (PSO) based cross-validation method for reducing dimensionality.

There are two distinct groupings of variable selection algorithms, specifically wrapper methods and filter methods. The wrapper methods employ the feature subset selection algorithm in unison with an induction algorithm. The selection algorithm proceeds to unearth a favorable subset of data whilst using this induction algorithm to evaluate proposed subsets. The filter methods use a preprocessing step and autonomously select variables independent of the induction algorithm. There are a number of algorithms that fall under the umbrella of the filter approach, such as the FOCUS algorithm, which inspects all subsets of features in a brute-force fashion in order to unearth a minimal subset of variables that adequately represent the whole; the relief algorithm, which assigns a weighting of relevance to each feature, that is, the relevance of the selected variable to the target output; and the decision tree algorithm, which is used to select feature subsets for the nearest neighbor algorithm [Kohavi and John, 1997].

Rueda et al., [2004] highlight a particular strength possessed by wrapper algorithms. The authors state that if variables are highly correlated with the response, the filter algorithm would typically include them, even if they diminished the overall algorithm performance. While in the wrapper approach, the induction algorithm may discover these diminishing effects, and exclude them.

Outliers

As previously mentioned, the effect of outliers (a.k.a. noisy data) can have diminishing effects on the accuracy of a data mining and analysis exercise. Many factors serve as the causes of these anomalies including human error/maliciousness, system faults, erroneous measurements or innate deviations [Hodge and Austin, 2004]. The exceptional behaviors of these datapoints go a long way in damaging the accuracy of a given experimentation if overlooked and included incorrectly in the mix. They contribute little or no relevant information to the overall model, and indeed, can be detrimental to the data mining process [Tang et al., 2007]. Detection and removal of detrimental outliers is a key component of this process. The method we use is commonly referred to as the Quartile or Fourth-Spread method [Devore, 2000]. Essentially, we identified the boundaries of each of the quartiles in your data set, measure the fourth-spread (fs), which is the distance between the lower and upper quartiles, and set the upper and lower outlier boundaries as a function of fs. A quartile is any of the three values that divide an ordered data set into four approximately equal parts. Quartiles are a particular type of quantiles, which divide the data into some given number of equal parts.

Experiments and Discussion

This study requires multiclass classification, as the training set consists of data points belonging to 10 different quality classes. From another hand, the SVM are inherently binary classifiers, which means that their usage is to discriminate between two classes. There are different strategies to make multiclass classification via SVM, such as one-vs.-all (OVA), one-vs.-one (OVO), or using SVM regression with error-tolerance mapping. The OVA technique presumes that binary classifiers are built for each class so that each of them distinguishes between one of the labels and the rest of labels, that is one-versus-all. The OVO technique presumes that a separate classifier is built for each pair of classes. By using a voting technique, the class with most votes is the winner. Due to the complexity of those techniques with regard to the nature of the task solved, the strategy applied in this study was using SVM for regression, which outputs wine quality as a real value. Values were then mapped to integer class labels by the error tolerance technique. The error tolerance τ, a positive real number, defines the interval $[X-\tau, X+\tau]$. A regression output is hit for a class X, if the value belongs to the interval, or miss

otherwise. This approach also preserves the order of preferences. For example, if the true quality class is 5, a model prediction 6 is better than prediction of class 8.

In order to build the SVM model, the dataset was divided into three separate subsets, namely training (50%), validation (25%) and testing (25%).

A number of metrics were used to estimate model performance. These include:

Prediction accuracy at certain error tolerance values were calculated. For the sake of consistency with previous studies [Cortez et al., 2009], the error tolerance thresholds used for experiments were 0.25, 0.5, 1, and 2.

- Mean Absolute Deviation (MAD) represented by (16) is a robust performance measure of the model variability

$$MAD = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \quad (16)$$

where $\hat{y}_i$ is the predicted value.

- Area over the regression error characteristic curve. The regression error characteristic (REC) curve plots the error tolerances along the horizontal axis versus the prediction accuracy on the vertical. The area over the REC curve (AOC) is a scalar value that estimates the overall model performance regardless of the error tolerance values applied to each model instance. The 'ideal' classifier is represented by the most north-west point indicating 100% accuracy and zero error tolerance. Therefore, the model with the least AOC is best performing. REC and AOC applied to SVM regression are analogous to the receiver operating characteristics (ROC) curve and the area above the ROC curve (AOC) metrics used to estimate binary classifiers.

The effectiveness and accuracy of an SVM model is largely dependent on the selection of kernel, the kernel's parameters and value of the cost parameter C. This is an empirical task as there is no theory that can suggest optimal parameter values and also those values strongly depend on features of the training data set and the nature of the task solved. There are a number of parameters that can control the learning and performance of SVM/SVR. The two most relevant are the insensitivity zone ε and the penalty parameter C, both selected by the user. The former parameter is a positive real number that controls the width of the insensitive zone used to fit the training set, controlling in that way the number of support vectors and model complexity. It also determines the level of accuracy of the approximated classification/regression function and finally the generalization capabilities of the model. An increase in ε means a reduction in requirements for the accuracy of approximation, but at the same time decreases the number of support vectors, which reduces the model complexity. If ε is larger than the range of the target values, results are poor. On the other extreme, if ε is zero, we can expect overfitting. Some studies report that a good empirical rule is that value for ε is one that leads to percentage of support vectors to be about 50% of the number of dataset samples. The latter parameter C is a penalty factor that can control the tradeoff between the training error and model complexity, which is the number of support vectors. If C is too large, we have high penalty for non-separable data points and many support vectors, which in fact which turns a soft-margin SVM into hard-margin SVM. This leads to overfitting. On the other extreme when C is zero, we have no penalty for misclassifications, few support vectors, and model underfitting. A reasonable proposal for value of C is to be close to the upper bound of the output values, i.e. if the model outputs in [0,B], a value close to B would be a robust choice.

In order to cast the broadest possible catchment area in search of the best performing SVM, four kernels were tested: linear; polynomial with parameter ranges d=[0,5], gamma=[0,5], and r=[0,5]; RBF with gamma=[-5,5]; and

sigmoid with parameters gamma=[-5,5], and r=[-5,5]. Also, the parameter C from (14) was explored in [0,10], and insensitivity zone epsilon=[0,1].

Popular techniques for finding optimal parameter values are grid search and a pattern search. A grid search tries values of each parameter across the specified search range using geometric progression, e.g. $C \in \{2^{-4}, 2^{-2}, 1, 2^{2}, \ldots, 2^{8}, 2^{10}\}$. Similarly, γ can take a range of values. The grid search tests the model with each pair of values. Obviously, the method can be computationally expensive in some cases, as it must be evaluated with many parameter values in the grid. The things can even get worse if cross-validation (CV) for each trail is applied. Another search technique called pattern search (also known as a compass search or a line search) can be applied. It starts at the center of the search range and makes trial steps in each direction for each parameter. If the model accuracy improves, the search center moves to the new point and the process is repeated. If no improvement is found, the step size is reduced and the search is tried again. If no step improves the model, the step size is reduced and the process is repeated. The search stops when the search step size is reduced to a specified tolerance. This method requires fewer evaluations but a weakness is that it may find a local rather than global optimal point for the parameters (local minimum problem). A combination of the two techniques is possible, e.g. grid search optimum is further refined by pattern search. Neither technique, however guaranties that the search will end up with a global optimum instead of a local one. For the purposes of this study we applied the pattern search technique. The results of the experiments displayed the polynomial kernel as performing best.

After SVM regression was carried out, k-fold cross-validation (CV) was applied to ensure the integrity of the experiments. This study uses k=5 as many authors, including Cortez et al. [2009], recommend five-fold CV as more robust than other validation techniques. The dataset was split into five subsets, each holding 20% of the instances. Each chunk was used for testing, while the 80% chunk was used in training. This process was then iterated 5 times, with each of the K subsamples used once as the testing data.

Two different attribute evaluation techniques were used, evaluation on either a subset or individual basis. Attribute subset evaluation techniques were classifier subset evaluation; consistency subset evaluation; and wrapper subset evaluation. The single attribute selection techniques used were: chi-squared evaluation, which is based on the chi-squared statistics; gain ratio attribute evaluation; information gain attribute evaluation; principal component analysis evaluation; relief attribute evaluation; and symmetric uncertainty attribute evaluation. A brief description of those techniques can be found in [Hall et al., 2009] and [Witten and Frank, 2005]. After a multitude of attribute evaluation runs and counting the AOC, it was found that the best performing attribute selection technique for red wine is chi-squared evaluation. Table 1 shows the pre-cross-validation top performers.

Table 1. Pre-CV red wine attribute selection techniques.

Attribute Selection	AOC
ChiSq+3+4+11+12	50.55938
ChiSq+2+3+4+6+8+9+11+12	50.73438
ClassifierSubsetEvaluatorRandomSearch+2+3+4+6+11+12	50.975
CfsSubsetEvalRandomSearch+3+4+8+9+11+12	51.0125
ChiSq+3+4+8+9+11+12	51.0125
OriginalPolynomial	52.375

The best post-cross-validation model, however, is the second in Table 1. It suggests using 8 attributes, namely: alcohol, volatile acidity, sulphates, citric acid, total sulfur dioxide, density, chlorides, and fixed acidity. Table 2 and Figure 4 show the worth value, i.e. the percentage of importance of each attribute as proposed by the chi-squared attribute evaluation.

It was also explored how removal of outliers affects the predictive capabilities of the model. For each individual attribute, boundaries were quantitatively set which excluded outliers that resided outside the assigned boundary. This boundary was set using the fourth-spread method. This method entailed identifying the boundaries of the quartiles of each attribute within the wine quality dataset, identifying the range between the upper and lower quartiles. Upper and lower outlier boundaries were set as a function of this fourth-spread. After outliers had been removed it was found that the model improved slightly its performance upon its predecessor, by 0.39%, which is insignificant. This shows empirically that the SVM technique is robust in the studied application area and works well with noisy data.

Table 2. Chi-square attribute importance, red wine.

Attribute	Chi-Squared Worth	Importance %
alcohol	497.7464	29.61
volatile acidity	354.4793	21.09
sulphates	252.0535	15.00
citric acid	169.8607	10.11
total sulphur dioxide	145.3958	8.65
density	130.73	7.78
chlorides	82.6207	4.92
Fixed acidity	48.0288	2.86
ph	0	0.00
residual sugar	0	0.00
free sulphur dioxide	0	0.00

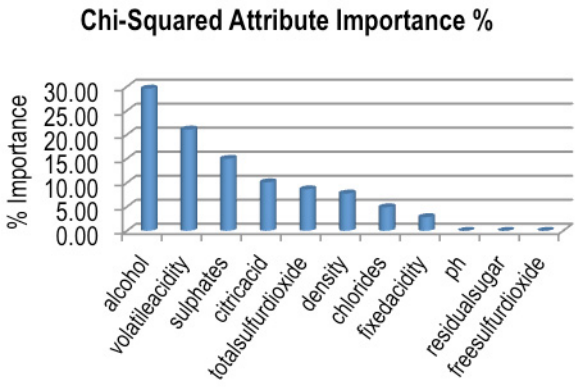


Figure 4. Chi-square attribute importance, red wine.

It was found that by using polynomial kernel, the model could be optimized so that in certain conditions it can outperform previously reported models. A combination of the attribute selection described above with SVM parameters C = 1.397998, epsilon = 0.745744, d = 1, gamma = 0.571688, and r = 0.529951 leads to mean squared error (MSE) reduced to 0.4369, which results in a confusion matrix presented by Table 3.

Table 3. Confusion matrix for SVM red wine prediction model. Bold writing denotes accurate predictions.

Actual Class	Red Wine Predictions				
	4	**5**	**6**	**7**	**8**
3	0	9	1	0	0
4	1	35	17	0	0
5	2	466	210	3	0
6	0	193	413	32	0
7	0	11	130	58	0
8	0	0	13	5	0

Results register that under those conditions and error tolerance τ=1, the model reaches prediction accuracy of 89.5%, outperforming the best model reported by Cortez et al. (2009).

Similar considerations were made regarding the white wine quality prediction task. In summary, the best attribute selection technique found was symmetrical uncertainty ranking [Hall et al., 2009]. It registered the lowest AOC upon cross-validation and suggests 7 attributes, presented and plotted in Table 4 and Figure 5. These are alcohol, density, chlorides, total sulfur dioxide, citric acid, free sulfur dioxide, and volatile acidity.

Table 4. Symmetrical uncertainty ranking of attributes, white wine.

Attribute	Symmetrical Uncertainty Ranking	% Importance
alcohol	0.08998	26.46626272
density	0.06524	19.18936408
chlorides	0.04878	14.34790282
total sulphur dioxide	0.03513	10.33296076
citric acid	0.03468	10.20060004
free sulphur dioxide	0.03376	9.929995882
volatile acidity	0.03241	9.532913701

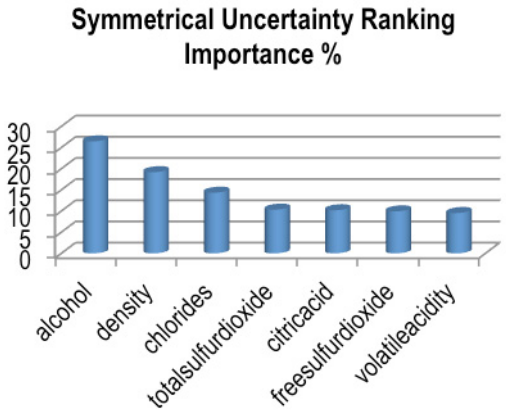


Figure 5. Symmetrical uncertainty importance of attributes, white wine

Similarly to the red wine task, part of the research was to combat the detrimental effects of outliers, which resided within the white wine dataset. This process, similar to the method used with the red wine dataset, was conducted through the use of the fourth-spread method. Once these outliers had been eradicated, a re-optimization procedure was conducted in order to attempt to find a better combination of parameters, which would improve the

overall performance of the SVM. This re-optimization failed to improve upon the previously attained MSE of 0.555 unearthed in the initial optimization runs.

It was found that by using polynomial kernel, the white wine model could be optimized so that a combination of the abovementioned attribute selection with certain SVM parameters leads to a confusion matrix presented by Table 5. The experiments clearly showed that the performance of the model built with a reduced attribute selection significantly outperforms its pre-reduced counterpart. Across all error tolerance levels there is a substantial prediction accuracy improvement held by the reduced model. With regards to the overall performance of these models, best depicted by the AOC metric, the reduced model holds a strong 9.61% improvement over its original complete state.

Table 5. Confusion matrix for SVM white wine prediction model. Bold writing denotes accurate predictions.

Actual Class	White Wine Predictions				
	4	5	6	7	8
3	0	2	17	0	0
4	19	55	88	1	0
5	7	833	598	19	0
6	0	235	1812	144	3
7	0	18	414	441	7
8	0	3	71	43	59
9	0	1	3	2	0

Conclusion

The goal of this study was to explore what factors affect the quality of the SVM model in the prediction of wine quality. I was found that the choice of kernel function greatly affects the model predictive abilities. The kernels explored were linear, radial basis function, polynomial, and sigmoid. It was only the polynomial kernel that returned workable results due to its abilities to transform the input space into a much higher dimensional one, thus improving the discriminatory power of the model. It was also found that an appropriate reduction of variables and finding an optimal subset greatly improves the predictive power of the model. An improvement of 9.61 % was found when comparing the pre-reduced model with the post-reduced model in the case of the white wine dataset. One of the primary contributions of this study is improvement of the model performance with regard to error tolerance 1 in the case of red wine dataset. By using variable selection technique based on the chi-squared attribute evaluation, the model outperforms that of Cortez et al. [2009]. Variables eliminated during the model constructions were residual sugar, free sulfur dioxide, and pH. It was also found that removal of outliers, which are anomalous in nature, can improve the overall performance, but marginally, which draws to the conclusion that SVM is a robust data mining technique in this application area and that eliminating outliers influences little the predictive abilities of the model.

Bibliography

[Agrawal et al., 1993] Agrawal, R., Faloutsos, C. & Swami, A. Efficient similarity search in sequence databases. Foundations of Data Organization and Algorithms, 69-84, 1993.

[Bapna and Gangopadhyay, 2006] Bapna, S. and Gangopadhyay, A. A Wavelet-Based Approach to Preserve Privacy for Classification Mining. Decision Sciences, 37, 623-642, 2006.

[Beltran et al., 2008] Beltran, N. H., Duarte-Mermoud, M. A., Soto Vicencio, V. A., Salah, S. A. & Bustos, M. A. Chilean Wine Classification Using Volatile Organic Compounds Data Obtained With a Fast GC Analyzer. IEEE Transactions on Instrumentation and Measurement, 57, 2421-2436, 2008.

[Burges, 1998] Burges, C. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2, 121-167, 1998.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. Support-vector networks. Machine Learning, 20(3): 273-297, 1995.

[Cortez et al., 2009] Cortez, P., Cerdeira, A., Almeida, F., Matos, T. & Reis, J. Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47, 547-553, 2009.

[Dash and Singhania, 2009] Dash, M. & Singhania, A. Mining in Large Noisy Domains. J. Data and Information Quality, 1, 1-30, 2009.

[Deh, 2008] Deh, W. 2008. Surface Hardness Intelligent Prediction in Milling Using Support Vector Regression. Fourth International Conference on Natural Computation, ICNC '08, 2008

[Devore , 2000] Devore, J.L. Probability and Statistics for Engineering and the Sciences. Pacific Grove, CA, 2000.

[Drucker et al., 1997] Drucker, H. Burges, C., Kaufman, L., Smola, A., and Vapnik, V., Support vector regression machines, Advances in Neural Information Processing Systems 9, pages 155-161, Cambridge, MA, MIT Press, 1997.

[Fayyad et al., 1996], Fayyad, U., Haussler, D. & Stolorz, P. Mining scientific data. Commun. ACM, 39, 51-57, 1996.

[Fei et al., 2008] Fei, L., Li, W. & Yong, H. Application of least squares support vector machines for discrimination of red wine using visible and near infrared spectroscopy. Intelligent System and Knowledge Engineering, ISKE' 08, 2008.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten. I., 2009. The WEKA data mining software: an update. SIGKDD Explor. Newsl. 11, 1, 10-18, 2009.

[Hodge and Austin, 2004] Hodge, V. & Austin, J. A survey of outlier detection methodologies. Artificial Intelligence Review, 22, 85-126, 2004.

[Horng et al., 2010] Horng, S., Su, M., Chen, Y., Kao, T., Chen, R., Lai, J. and Perkasa, C. A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications, 38, 306-313, 2010.

[Hsien-Chu et al., 2008] Hsien-Chu, W., Kuo-Ching, L., Jun-Dong, C. & Ching-Hui, H. An Image Steganographic Scheme Based on Support Vector Regression. Eighth International Conference on Intelligent Systems Design and Applications, 2008. ISDA '08, 2008.

[Kohavi and John, 1997] Kohavi, R. & John, G. H. 1997. Wrappers for feature subset selection. Artificial Intelligence, 97, 273-324, 1997.

[Li et al., 2010] Li, J., Wang, J.-J., Zhang, T., Ma, C.-X. & Hong, W.-X. The Graphical Feature Extraction of Star Plot for Wine Quality Classification. First International Conference on Pervasive Computing Signal Processing and Apps., 2010.

[Narwaria and Lin, 2010] Narwaria, M. & Lin, W. Objective image quality assessment based on support vector regression. IEEE Transactions on Neural Networks, 21, 515-519, 2010.

[Oladunni and Singhal, 2009] Oladunni, O. O. & Singhal, G. 2009. Piecewise multi-classification support vector machines. International Joint Conference on Neural Networks, IJCNN'09, 2009.

[Petrujkic et al., 2008] Petrujkic, M., Rapaic, M. R., Jakovljevic, B. & Dapic, V. Electric energy forecasting in crude oil processing using Support Vector Machines and Particle Swarm Optimization. 9th Symposium on Neural Network Applications in Electrical Engineering, NEUREL, 2008.

[Rueda et al., 2004] Rueda, I. E. A., Arciniegas, F. A. & Embrechts, M. J. SVM sensitivity analysis: an application to currency crises aftermaths. IEEE Transactions on Systems, Man and Cybernetics, 34, 387-398, 2004.

[Salfner et al., 2010] Salfner, F., Lenk, M. & Malek, M. A survey of online failure prediction methods. ACM Comput. Surv., 42, 1-42, 2010.

[Selvanayaki et al., 2010] Selvanayaki, M., Vijaya, M. S., Jamuna, K. S. & Karpagavalli, S. An Interactive Tool for Yarn Strength Prediction Using Support Vector Regression. Conference on Machine Learning and Computing (ICMLC), 2010.

[Tang et al., 2007] Tang, J., Chen, Z., Fu, A. & Cheung, D. Capabilities of outlier detection schemes in large datasets, framework and methodologies. Knowledge and Information Systems, 11, 45-84, 2007.

[Tinney, 2006] Tinney, M.-C. 2006. Wine Business Monthly Salary Survey Report [Online]. Wine Business Monthly. Available: http://www.winebusiness.com/wbm/?go=getArticle&dataId=45483.

[Wang et al., 2010] Wang, X., Lv, J. & Xie, D. A hybrid approach of support vector machine with particle swarm optimization for water quality prediction. 5th International Conference on Computer Science and Education (ICCSE), 2010.

[Witten and Frank, 2005] Witten, I. H. & Frank, E. Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann Pub, 2005.

[Xiaoh, 2009] Xiaoh, W. Intelligent Modeling and Predicting Surface Roughness in End Milling. Fifth International Conference on Natural Computation, ICNC '09, 2009.

Authors' Information

Anatoli Nachev *– Business Information Systems, Cairnes Business School, National University of Ireland, Galway, Ireland; e-mail: anatoli.nachev@nuigalway.ie*

Major Fields of Scientific Research: data mining, neural networks, support vector machines, adaptive resonance theory.

Borislav Stoyanov – Department of Computer Science, Shumen University, Shumen, Bulgaria; e-mail: *borislav.stoyanov@shu-bg.net*

Major Fields of Scientific Research: artificial intelligence, cryptography, data mining.

WEIGHTS OF TESTS

Vesela Angelova

Abstract*: Terminal test is subset of features in training table that is enough to distinguish objects in classes. Tests are used as supporting sets in classification algorithms of type estimate calculation or voting. We propose ideas for valuation of votes and for reasoning of different weights of votes. The modifications for weights are proposed that are based on representativeness of test, distribution of votes from objects, dependency of features and coding of tests.*

Keywords*: classification, voting, tests, combinatory, coding theory.*

1. Introduction

1.1. Classification as part of machine learning (or pattern recognition), attempts to assign each input record to one of given classes. One type of learning procedure uses a training set of pre-existing patterns or instances, which have been labeled by hand with the correct class. The data-set used for training consists of vectors of observed features (variables) and class-label. This training set produces a rule (or classifier) which can evaluate a class-label for new input records of variable measurements.

One of the classification methods with training table uses a subset of the variables such that the class-label of the records in the training data can be distinguished only by them. We call such subsets tests. If the classes are disjoint, the full set of variables is a test.

Irreducible or *terminal test* (TT) is a set of variables, such that it is a test and removing any variables from it would make the resulting set non-test [Zhuravlev,1980]. *Length* of TT is the number of variables in the set. TT is a fraction, sensitive to class-label, partial and minimal description of differentiation for all classes (2 or more). TT makes alone assumption for affiliation (membership) of record to class and this is a property learned by the training data.

If a new vector with the same variables is given by measurements of this variables, according to a number of coincidences for all TTs, we can establish the similarity to each class and choose join (recognize) to top rated class, with voting scheme. In the standard approach one coincidence between the pattern to be classified (new object) and the object of training table for TT is one vote for class labeled this object.

1.2. For this type of discrete heuristic algorithm one problem to overcome is to find the full set TTs. This is done only once and brute force approaches perform well even for tens of features in the training vectors. Doing this would be useful anyway when there are not enough records for statistical measurements.

A problem we discuss is the weight of the votes for short and long TTs. Long tests are intersecting and numerous. This creates excess in their votes over a test which consists of one variable, although the former distinguishes classes alone. We search in several directions for numerical expression of TT' weights as far as reasoning for different weights. Indeed one justification is proposed in [Angelova,1987], where one vote of test reflects in proportion on weights of relevant variables.

2. Tests according to its representativeness

The set of all subsets of variables may view as flower bud:

n number of subsets with length 1

$\frac{n(n-1)}{2}$ number of subsets with length 2

...

n number of subsets with length *(n-1)*

1 number of subsets with length *n*.

The votes from TTs should not be with equal weights, because each one has different base of reducible or non-terminal tests, which part is this TT. With standard voting TT with length one and all its extensions distinguish classes, but have negligible contribution for belonging to class. This type of pattern recognition independent of the length is unbalanced. One instrument for measure the weight of TT is to count all tests, representing by TT.

Let have training table with *n* variables for objects, labeled to classes.

For TT with length 1 (subset of one variable) we can find extended non-terminal tests representing by TT:

with length *2* -- *(n-1)* different choices for the second variable,

with length *3* -- $\frac{(n-1)(n-2)}{2}$ different choices for two variables,

....

with length *k* -- $\frac{(n-1)(n-2)\dots(n-(k-1))}{(k-1)\dots 2\cdot 1}$,

…

with length 1+*(n-2)* -- $\frac{(n-1)(n-2)\dots(n-(n-2))}{(n-2)\dots 2\cdot 1} = n-1$,

with length 1+*(n-1)* = *n* -- one choice or full set of variables.

TT with length *k* represents such pyramid of tests with truncated top of length *k* :

with length *κ+1* -- *(n-k)* different choices for adding another variable to set –TT,

with length *κ+2* -- $\frac{(n-k)(n-k-1)}{2}$,

...

with length *k+(n-k)* = *n* , one choice.

The picture is like this: the TT' variables set is extended by one until it reaches the base of full variable set. Thus every TT is peak (acute or truncated) of open leaf of bud flower. This point of view gives metric concerning TT representativeness with regard to reducible (non-terminal) tests.

Statement: The numerical expression of representativeness of TT with length *k* is 2^{n-k} .

Really, let count for each TT' length number of representing tests in whole leaf:

For TT with length 1: $1+(n-1)+\frac{(n-1)(n-2)}{2}+\dots+(n-1)+1 = 2^{n-1}$

For TT with length 2: $1+(n-2)+\frac{(n-2)(n-3)}{2}+\dots+(n-2)+1 = 2^{n-2}$

…

For TT with length *k: 1+(n-k)+* $\frac{(n-k)(n-k-1)}{2}$ *+…+(n-k)+1*= 2^{n-k} .□

Ratio is not depending on the number of TTs b_1, b_2, …, b_k respectively with length 1, 2,..., κ so we can normalize obtaining weights:

$2^{n-k}/x = b_1.2^{n-1} + b_2.2^{n-2}+... b_k.2^{n-k})/100$ gives the weight of the lightest and longest TT (with length 2^{n-k}) and the percentage when $2^{k-1} : 2^{k-2} : … : 2:1$.

At first glance it seems the weight of short TT is greater, but long ones are more and partly repeating due to intersecting of variables. Thus relative differences according to aggregate weights are more acceptable.

Example 1:

```
   class K1            (1,1,1,0,1,0,1,1,0,0,1),
                       (1,0,1,1,0,1,1,1,1,0,1),
                       (0,1,1,0,0,0,1,0,1,1,0),
   class K2            (0,1,0,0,1,0,0,1,0,1,1),
                       (1,0,0,1,0,1,0,0,0,1,1),
                       (0,0,0,1,1,0,0,1,0,0,0).
   New pattern         (0,0,1,1,1,0,1,0,0,0,1)
```

Number of all TTs is 16, sorted by length:

```
    1 2 3 4 5 6 7 8 9 10 11 length  weight
01  0 0 1 0 0 0 0 0 0 0 0 - 1        8
02  0 0 0 0 0 0 1 0 0 0 0 - 1        8
03  1 0 0 0 0 0 0 1 0 0 0 - 2        4
04  0 0 0 0 0 0 0 0 0 1 1 - 2        4
05  1 1 0 0 0 0 0 0 1 0 0 - 3        2
06  1 0 0 1 0 0 0 0 1 0 0 - 3        2
07  1 0 0 0 1 0 0 0 1 0 0 - 3        2
08  1 0 0 0 1 0 0 0 0 1 0 - 3        2
09  1 0 0 0 0 1 0 0 1 0 0 - 3        2
10  1 0 0 0 0 0 0 0 1 1 0 - 3        2
11  0 1 0 0 1 0 0 0 0 1 0 - 3        2
12  0 1 0 0 0 0 0 0 1 1 0 - 3        2
13  0 0 0 1 1 0 0 0 0 1 0 - 3        2
14  0 0 0 1 0 0 0 0 1 1 0 - 3        1
15  0 1 0 0 0 1 0 1 0 1 0 - 4        1
16  0 0 0 1 0 1 0 1 0 1 0 - 4        1
```

Total: 16 Sum of weights: 46

When each vote is regular (weight=1) the new pattern obtains 9 votes from K1 and 12 - from K2 (the possibility is from 3 objects to receive maximal 16 votes or total 48 votes from class), therefore is recognized as belonging to class K2.

When each vote is according representativeness of TT, the same new pattern obtains 60 votes from K1 and 24 votes - from K2 (maximum is 3.46=138 votes from class), therefore is recognized as belonging to K1 according to coincidences with measurements for TT consisting of one variable (variable 3 and variable 7).

3. Distribution of the objects in class

When a new object is joining to a class some short TTs may disappear and spread into few longer ones. This is another circumstance supporting our idea for alignment weight of short TT to weights of several derived from him, **as if they tend to keep total weight**.

How is the new object recognized before joining is essential question. If it is recognized with an advantage to a class by the votes of all objects in this class, then the number of TTs is changing less (respectively sum of weights), the over-fit is hold. If it is recognized to a class by average votes, but major differences between objects in this class, the increment of TT' number is larger. We can imagine in space model like adding to a group a new point, which is out of this group and near some participants. Adding this new point to a group expands differences (TTs) by number and lengths.

The recognition quality would be better when fewer and shorter TTs are used (therefore weighty), with uniform distribution of objects' votes. If these requirements are not fulfilled, the questions raise about more classes or distant characteristics of objects.

Example 2:

The new objects `(1,0,0,1,0,0,1,1,0,1,0)` is recognized according above mentioned training table, receiving:

- from class K1 - average 12,6 weighted votes (14,12,12), and non-weighted 2,3 (2,3, 2)

- from class K2 - average 14 weighted votes (8,26,8) , and non-weighted 3 (1,10,1).

If this new object is joining to K2, because of votes by only one participant, the TTs' number raises more than joining to K1. In both cases the new object breaks one short TT (length one).

We can conclude for this new object together with nearest object to make new class.

4. How to use dependency (intersecting) between TT

It is clear that only TTs with length one are independent, another ones form some groups of intersecting variables. Two approaches are possible for this problem – to represent depending feature like **common unit**, or to focus on the features out of this unit like **unique base** for weight of TT.

4.1. We can choose some subsets of variables, which are part of intersecting TTs, but are not TT itself. In these we find representative sets – i.e. different combinations, typical for the class, which are not found in other classes. The example shows the most frequent occurrence of a subset of features 1 and 9, and the measurement for these features (1, 1) and (0, 1) are typical of class K1, and (0, 0) is typical for K2. (But (1, 0) for the same features 1 and 9 occurs in both classes, this measurement is not representative sets, that's why the features 1 and 9 do not form TT). If there is not typical measurement for one class, anti-closeness is possible decision.

Quite often are found subsets {2, 10}, {4, 10}, {5, 10}, {9, 10}.

There are subsets that are not found in any test and may be excluded from consideration - for example {2, 4}, {5, 6} or (5, 8}.

4.2. Another approach is to determine the coverage of each TT upon reducible tests from level 4-features, but those that are covered in one way by TT. In this way we do slit of all subsets of some level, in the case of 4-

variables tests and give weight of TT on top of each of these sets, but without overlap. The task is solved by software.

№	3	7	11	8	6	2	4	5	**1**	**9**	10	#4-variables, uniquely covered	#3-variables
01	1	0	0	0	0	0	0	0	0	0	0	60	34
02	0	1	0	0	0	0	0	0	0	0	0	60	34
04	0	0	1	0	0	0	0	0	0	0	1	14	7
03	0	0	0	1	0	0	0	0	1	0	0	14	7
09	0	0	0	0	1	0	0	0	**1**	**1**	0	1	1
05	0	0	0	0	0	1	0	0	**1**	**1**	0	1	1
06	0	0	0	0	0	0	1	0	**1**	**1**	0	1	1
08	0	0	0	0	0	0	0	1	**1**	**1**	0	1	1
10	0	0	0	0	0	0	0	0	**1**	**1**	1	0	1
08	0	0	0	0	0	0	0	1	1	0	1	1	1
13	0	0	0	0	0	0	1	1	0	0	1	2	1
14	0	0	0	0	0	0	1	0	0	1	1	2	1
11	0	0	0	0	0	1	0	1	0	0	1	2	1
12	0	0	0	0	0	1	0	0	0	1	1	2	1
15	0	0	0	1	1	1	0	0	0	0	1	1	0
16	0	0	0	1	1	0	1	0	0	0	1	1	0

The dependencies between the features become apparent after length 2 of TT when trying to line-up of the triangular matrix and the found coating on level 4 and 3- variables indicate that TT with length 1 and 2 are crucial for classification (left table). On the other hand the unique base of TT (right columns) consists of very different number of reducible tests.

5. Code-words for tests

Another point of view is to represent TT like codeword. The *coding* is transformation to another alphabet; our goal is to compress the description of TTs (see the table in example 1). We have a table, each row is TT with 1 for involved variable and 0 for not-involved. This table is binary even for k-meaning variables. Thus the length of TT (number of variables) is *Hamming weights* of codeword and we look for big weight of TT with small Hamming weight of its code-word.

Assume that there is at least one 1 for variable, otherwise this variable is removed. No one description is part of other by definition of TT.

Statement: The *Hamming distance* between two code-words is at least 2.

Really, if some description t_i is part of (or involved in) another t_j, then t_j is not TT. Therefore t_i has at least one 1, for which t_j has in the same place 0. Because opposite statement is also true, they have at least two mismatches.□

Let sort the rows by its Hamming weights or number of ones. We can cut short each description to last 1, and the set of code-words is prefix-free binary code, by definition of TT no one contains in another. But we can truncate these descriptions or code-words more forward by rearrangement of variables (or columns of table). If we put first the variables with 1 for the shortest TT and cut the description after this 1, and if we continue in the same manner for the next lengths of TTs, we receive shortest code-words for TT with highest weight. Code-words for TT with

weight 1 (for example its number is k) have different length (from 1 for the first TT to k, k≤n for the last with length 1). We continue with length 2 taking into account to put forward the variables with more contributions or 1s in column. By increasing the length of TT more and more their 1s are pulled and the lengths of code-words like to be the same (restrict by n) but the code-words differ in 2 positions at least. Constructing a tree confirms the necessary of different weights for TTs, namely *weighted path length (depth)* of code is minimal when codeword length is short for maximum weight (usually proportional to probabilities).

Statement: If we take into account proposed weights, precisely $p_1 \geq p_2 \geq \ldots \geq p_s$. (Indeed $p_i = 2.p_j$ for test *i* with length by 1 shorter than length of test j), we can generate shortest description of set of all TTs.

Really, we can swap TT with equal length because of equal weight, although the code-word lengths are not equal. By finite number of attempts this swapping produces minimal total length of code-words in 3-dimensional space because of distance (unlike Huffman code in the plane where two code-words have distance one).□

Thus TTs minimal coding shows that the dividing hyper-surface between classes consists of units, low-dimensional are high-significant.

Example 3:

For our training table and table of TTs, one possible coding is:

	3	7	11	10	1	8	9	5	2	4	6
01	1										
02	0	1									
04	0	0	1	1							
03	0	0	0	0	1	1					
10	0	0	0	1	1	0	1				
07	0	0	0	0	1	0	1	1			
08	0	0	0	1	1	0	0	1			
05	0	0	0	0	1	0	1	0	1		
11	0	0	0	1	0	0	0	1	1		
12	0	0	0	1	0	0	1	0	1		
06	0	0	0	0	1	0	1	0	0	1	
13	0	0	0	1	0	0	0	1	0	1	
14	0	0	0	1	0	0	1	0	0	1	
15	0	0	0	1	0	1	0	0	1	0	1
16	0	0	0	1	0	1	0	0	0	1	1

This procedure provides minimal prefix-free coding of TTs, but not to be presented each other or to perform actions.

6. Conclusion

Updated software with proposed weight of TT in section 2 can be used for voting in such areas as recognition quality assessment software [Eskenazi,1990] or survey [Angelova,2008], where there are insufficient standards for statistics. The ideas from section 3, 4, 5 indicate directions for future work.

Acknowledgements

The paper is published with financial support by the project ITHEA XXI of the Institute of Information Theories and Applications FOI ITHEA (www.ithea.org) and the Association of Developers and Users of Intelligent Systems ADUIS Ukraine (www.aduis.com.ua).

Bibliography

[Zhuravlev,1980] Yu. I. Zhuravlev et al., "Recognition and classification problems with standard testing information", J. of Computing and mathematical Physics (Jurnal vichislitel'noi matematiki i matematicheskoi fisiki), 1294-1309, vol.20 No 5, 1980 (in Russian).

[Angelova,1987] V. Angelova, "Weighted estimation of characteristics", Mathematics and mathematical education, XVI UBM spring conference, 1987, pp.301-305 (in Bulgarian).

[Eskenazi,1990] A. Eskenazi, V. Angelova, "A New Method for Software Quality Evaluation", J. of New Generation Computing Systems, 3 (1990) 1, 47-53.

[Angelova,2008] V. Angelova, A. Eskenazi, "Application of classification methods to a problem related to specific groups of e-government users", Serdica J. Computing 2 (2008), 101-112.

Authors' Information

Vesela Angelova –*Software Engineering Department, Institute of Mathematics and Informatics*

Acad. Bonchev Str., Bl. 8, Sofia-1113, Bulgaria; e-mail: vaa@math.bas.bg

Major Fields of Scientific Research: Software technologies, Pattern recognition

TABLE OF CONTENT