NEAREST NEIGHBOR SEARCH AND SOME APPLICATIONS

Hayk Danoyan

***Abstract**: The problem of finding of nearest neighbors from the data set for a given query is considered. The important applications for different data types when the NN algorithms may be applied are enumerated namely for Textual data, Image data and Genome data. For each individual case the models under consideration (data representation type, similarity measure etc.) are those, which keep in view the specifications from practice.*

***Keywords**: Nearest neighbor, k-NN, text classification, parameter estimation, human pose estimation, sequence alignment, homology search, protein secondary function prediction*

***ACM Classification Keywords**: H.3.3, I.2.10, I.7.*

1. Introduction

Let we have some set V and some distance measure $\rho(u,v)$ between two elements u,v of V. For arbitrary subset A and vector x by $\rho(x,A)$ we denote the following $\rho(x,A) = min_{a\in A}\,\rho(x,a)$. The subset $F \subseteq V$ and a vector $x \in V$ are given. Consider the problem of finding the nearest elements (by means of distance $\rho(u,v)$) of F from x named as NN (nearest neighbor). In other words it is required to find the set$\mathcal{S} = \{y \in F/\rho(x,y) = \rho(x,F)\}$. The other variation of the problem also often considered named as "k-nearest neighbors" or short "k-NN" problem which consist in finding a set $\mathcal{S}$ such that:

I. $|\mathcal{S}| = k$;

II. $\forall y \in \mathcal{S}$ and $\forall z \in V\backslash\mathcal{S}$, $\rho(x,y) \leq \rho(x,z)$.

Keeping in mind the big volume of F which may considered in applications the main requirement is to exclude the linear scan which can made the problem practically unsolvable. The approximate variant of the problem is also considered named as approximate nearest neighbor problem [Andoni, 2009; Gionis, 1999].

There are a large amount of literature devoted to case when $V = \{0,1\}^n$, and $\forall x,y \in V$ by $\rho(x,y)$ is denoted the Hamming metric for example [Gionis, 1999; Rivest, 1974; Aslanyan, 2013; Aslanyan,

Danoyan, 2013; Aslanyan, 2014], etc. The R^n where R is the set of reals under metric of ρ_p is also under consideration, where $\rho_p(x,y) = \sqrt[p]{\sum_{i=1}^{n} |x_i^p - y_i^p|}$ for any $x, y \in R^n$. The aim of this paper is to bring survey on known applications where nearest neighbor method may be applied.

2. Orthography Correction with Dictionary

Let we have a dictionary of any language. Someone insert a word which may, in fact, contain an orthographic error. The aim is to detect the error. There are types of errors which can occur more likely: namely one can miss some character, type other character instead of right character or type an excessive character. Here we demonstrate the mentioned types of errors on word "character":

- caracter (missed the letter "h" after "c");
- charakter (instead of letter "k" mast be "c");
- charactere (letter "e" at the end is excessive).

The technic is the following: as a set F mentioned above we keep the set of orthographic forms of words of the language. The distance between words we consider the edit distance [Wagner, 1974]. So for the typed word we find the nearest neighbors from F by means of edit distance, which can allow correcting the error in the typed word.

3. Text Categorization

Let we have a set of documents $D = \{d_1, \dots, d_n\}$ and set of categories $C = \{c_1, \dots, c_k\}$. For each document $D_j, j = 1, \dots, n$ we have a label denoted as $y_j \in C$.

The problem is for unknown test instance d to predict the value of category vector [Sebastiani, 2002] or more formally it is required to approximate the unknown function $\Phi: D \to C$ where

$$\Phi(d) = c_i \Leftrightarrow \text{document } d \text{ belongs to class } c_i \text{ for some } i \in \{1, \dots, k\}. \quad (3.1)$$

The more general case when the classes may overlap can be considered [Sebastiani, 2002]. The text categorization problem is naturally arises in many applications. We mention the following [Sebastiani, 2002], [Aggarwal, 2012]:

Document Filtering. Let we have a collection of news articles. As the number of electronic articles written per day may be very big so it will become necessary to create a system which will automatically label each article to corresponding class (for example politics, art, science, business, sport, technics etc).

Document organization and retrieval. Let one has an electronic collection of text documents (scientific articles, news articles, web collections, etc.). It is required to organize document hierarchically such, that the browsing and retrieval will be efficient.

Spam Detection. When one receive the electronic email it may be spam. So in order to prevent the loss of user's time for reading it one need to categorize it spams or not spam.

For other possible applications we will refer to [Sebastiani, 2002; Aggarwal, 2012].

3.1. Text Representation

At first we have to represent the text document (which may be for example in format pdf, docx, etc.). The one of possible approaches is "bag-of-words" [Sebastiani, 2002] when each document is represented as $d_j = (w_{j1}, \dots, w_{j|T|})$, where T is a set of all possible terms, and w_{ji} is the weight of ith term in jth document. The set of all words occurring at least in one of the documents can be considered as T [Aggarwal, 2012, Salton, 1988, Sebastiani, 2002]. The articles (a/an/the etc), prepositions, conjunctions etc. and topic-neutral words are ignored. The second stage is stemming [Sebastiani, 2002] (grouping the words having the same morphological root).

There are different ways of defining weight [Sebastiani, 2002; Aggarwal, 2012; Salton, 1988]. One of the simplest ways is the following (also called the set-of-words representation): $w_{ji} = 1$ if ith term appear in jth document and $w_{ji} = 0$ otherwise. In general weights belong to range [0,1].

Probably one of the most used weighting method is:

$$w(d_i, t_j) = \phi(d_i, t_j) \log \frac{n}{\upsilon(t_j)}, \qquad (3.1.1)$$

where $\phi(d_i, t_j)$ denotes the number of times t_j occurs in d_i, and $\upsilon(t_j)$ denotes the document frequency of term t_j, that is, the number of documents in Tr in which t_j occurs.

In order for the weights to fall in the [0,1] interval and for the documents to be represented by vectors of equal length, the weights resulting from $w(d_i, t_j)$ are often normalized by cosine normalization, given by the following formula

$$w_{ij} = \frac{w(d_i,t_j)}{\sqrt{\sum_{k=1}^{|T|} w_{ik}}}. \quad (3.1.2)$$

Although normalized $w(d_i, t_j)$ is the most popular one, other indexing functions have also been used, including probabilistic techniques [Sebastiani, 2002; Salton, 1988].

The function representing the similarity of two document representing vectors may be useful, which may be defined as:

$$\rho(d_i, d_j) = \sum_{k=1}^{|T|} w_{ik} w_{jk}. \quad (3.1.3)$$

Mention that in case of binary text representation function defined by (3.1.3) coincides with hamming metric.

3.2. Text Classification Using NN Method

The dimensionality reduction technics may be applied to decrease the dimensionality of data type. For a survey for such technics we refer to [Sebastiani, 2002; Yang, 1997].

The nearest neighbor methods may be applied by the following way: Let the set of nearest neighbors of document d be the $(d_{i_1}, \dots, d_{i_k})$ and$(y_{i_1}, \dots, y_{i_p})$ are the corresponding labels. Now let $B_i = \{d \in F_x / \Phi(d) = c_i\}$. The label of document d to be classified will be assigned a label y where y defined as

$$y = \arg\max_i |B_i|. \quad (3.2.1)$$

As we already mentioned that this approach may easily be generalized for case with overlapping classes, i.e. the sample may belong to more than one class.

4. Human Pose Estimation

The articulated pose estimation problem is formulated as follows [Shakhnarovich, 2005; Shakhnarovich, 2003]. We are given an image which contains a human body. We also have an articulation model – a model of the body that describes the current 3D body configuration in terms of a set of limbs and rotational joints that connect them into a tree structure.

One can synthetically generate image of a humanoid with a computer graphics program like Poser (Figure 4.1). That image will correspond to the articulated model. The model is shown by plotting 2D projections a number of key joints (crosses) and the lines connecting them, which roughly correspond to limbs Figure 4.2. This model may be described by set of numbers, namely the coordinates of the joints. In fact, there are hundreds of parameters in these numbers that affect the resulting image: the articulated pose of additional body parts not accounted for by this coarse model, such as fingers; shape of the actual body parts, facial expression etc. Added to that could be the parameters that describe the scene, the objects in the background etc.

The goal of a computer graphics program like Poser is to start with these parameters and produce a realistic image.

The goal of computer vision is the opposite. In the context of articulated pose estimation this goal is to start from the Figure 4.1, and recover the relevant parameters (Figure 4.2) of the representation that “generated” the image, while ignoring the nuisance parameters.

Figure 4.1

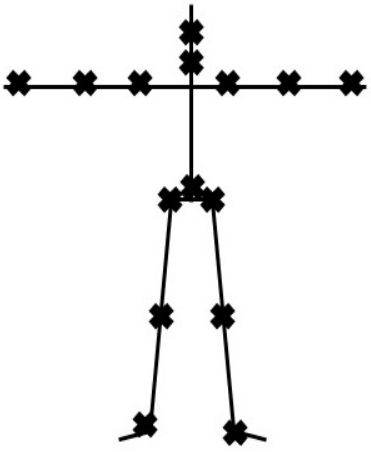

Figure 4.2

4.1. Example Based Pose Estimation

Now let we have a human image I represented as vector x (x belongs to some vector space X) and we want to retrieve the corresponding parameter vector denoted by θ belonging to some metric space Θ [Torralba, 2008]. The distance between two vectors $q, t \in \Theta$ will be denoted by $\rho_\Theta(q,t)$. We also have for some image feature vectors $x_1, x_2, \cdots x_m$ their corresponding parameter vectors i.e. $\theta_1, \theta_2, \cdots, \theta_m$. In example in the previous section parameter vector corresponding to image are the angles between those parts of body which are connected by joints (Figure 4.2). One approach to estimate θ based on k-NN is [Cover, 1968]

$$\theta = \frac{1}{k}\sum_{x_i \in F_{x,k}} \theta_i. \tag{4.1.1}$$

For theoretical ground of formula (4.1.1) we refer to [Cover, 1968].

In formula (1) the unknown parameter is estimated by using parameters corresponding to k nearest neighbors of x by means of metric ρ_X. Let us mention the following two difficulties in connection to formula (4.1.1):

- When the dimension of space X is big which is usual for multimedia applications the problem of searching k-NN may become to linear scan which is unacceptable in practice [Gionis, 1999]. Mention that in [Gionis, 1999] proposed method for approximate nearest neighbor search in sublinear time.
- The parameter values corresponding to k nearest neighbors of x may not be "close" to θ by means of metric ρ_Θ (Figure 4.1.1).

To come over the mentioned problem it is proposed [Shakhnarovich, 2005; Shakhnarovich, 2003; Torralba, 2008; Cipolla, 2013; Shakhnarovich, Darrel, Indyk, 2005] to consider the hash schema $H = \{h_1, \ldots, h_n\}$, $H: X \to E^n$ and $h_i: X \to \{0; 1\}$ such that the length of the hash code will be relatively small (to perform effective nearest neighbor search) and the "close" points in Θ by means of metric ρ_Θ will have "close" hash values in corresponding Hamming space with high probability.

We will treat the similarity as some binary relationship $S: X^2 \to \{-1, +1\}$ [Shakhnarovich, 2005], i.e. we will suppose two human images I_1 and I_2 are similar if $S(x_1, x_2) = 1$ and are not similar if $S(x_1, x_2) = -1$. In [Shakhnarovich, 2005] it is considered the case when

$$S(x_1, x_2) = \begin{cases} +1 \ if \ \rho_\Theta(\theta_1, \theta_2) \le R, \\ -1 \ if \ \rho_\Theta(\theta_1, \theta_2) > R, \end{cases}$$

for some predefined value of R, where x_1 and x_2 are the corresponding feature vectors of I_1 and I_2. The meaningful interpretation of this formula in context of pose estimation problem is the following: two images are similar if they have "close" poses or "close" values of joint angel vectors.

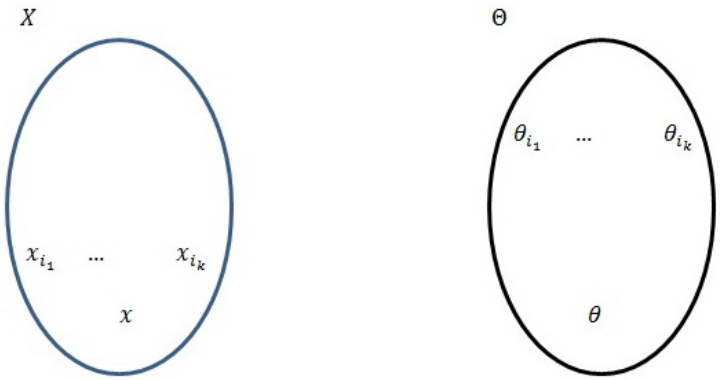

Figure 4.1.1

The general structure of hash function is [Shakhnarovich, 2005; Shakhnarovich, 2003; Torralba, 2008]:

$$H(x) = (h_1(x), \dots, h_n(x)), \tag{4.1.2}$$

where $\alpha_1, \dots, \alpha_n$ are real numbers.

Each h_i $i = 1, \dots, n$ has the form [Shakhnarovich, 2005; Shakhnarovich, 2003]:

$$h_{T,f}(x) = \begin{cases} 1 \; if \; f_i(x) \le T_i, \\ 0 \; if \; f_i(x) > T_i, \end{cases} \tag{4.1.3}$$

where $f_i: X \to R$ and T_i is some threshold. As f one can usually use the projection function [Shakhnarovich, 2005; Shakhnarovich, 2003; Gionis, 1999], i.e. $f(x) = x_d$, where by x_d is denoted the d[th] component of the vector x.

With each function h we can define a classifier by the following way

$$c(x, y) = \begin{cases} +1, if \; h(x) = h(y) \\ -1, otherwise. \end{cases}$$

As a measure of classification accuracy we will consider the following two values: expected true positive rate

$$R_{tp} = \underset{\substack{(x,y) \\ S(x,y)=+1}}{\mathrm{E}} [\Pr\,(c(x, y) = +1)]$$

and expected false positive rate

$$R_{tn} = \underset{\substack{(x,y) \\ S(x,y)=-1}}{\mathrm{E}} [\Pr\,(c(x,y) = +1)].$$

These values will be estimated from the available examples of similar and dissimilar pairs [Shakhnarovich, 2005; Shakhnarovich, 2003]. The empiric true positive rate will be denoted as

$$\hat{R}_{tp} = |\{(x,y)/S(x,y) = +1, c(x,y) = +1\}|.$$

By the same manner we will denote the false positive empirical rate:

$$\hat{R}_{fp} = |\{(x,y)/S(x,y) = -1, c(x,y) = 1\}|$$

The algorithm computing the optimal value of the threshold is brought in [Shakhnarovich, 2005].

Below the Algorithm 4.1.1 which computes the value of n and constructs the hash-function $H: X \to E^n$ is brought:

Algorithm 4.1.1: Similarity Sensitive Coding

Given: Set of similarity-labeled pairs $((x_1, y_1), l_1), \ldots, ((x_N, y_N), l_N)$,

Given: Lower bound on $\hat{R}_{tp} - \hat{R}_{fp}$ gap G

Output: Embedding $H: X \to E^n$ (n to be determined by the algorithm).

Let $n: = 0$.

Assign equal weights $W(i) = 1/N$ to all N pairs.

for all d = 1, . . . , dim(X) do

Let $f(x) = x_d$

Apply Algorithm of finding best threshold to obtain a set of m thresholds $\{T_t^d\}_{t=1}^m$ and associated true positive $\{\hat{R}_{tp_t}^d\}_{t=1}^m$ and false positive $\{\hat{R}_{fp_t}^d\}_{t=1}^m$ rates

for all $t = 1, \ldots, n$ do

if $\hat{R}_{tp_t}^d - \hat{R}_{fp_t}^d \geq G$ then

$n: = n + 1$;

$h_n(x) = h_{f,T_t^d}(x)$;

end for

Return H = $[h_1, \ldots, h_N]$

5. Genomic Sequence Processing

The nearest neighbor approach has also many biological applications. Let we have a finite alphabet Σ (for biological applications it is useful to consider when $|\Sigma| = 4$ and $|\Sigma| = 20$ corresponding to the cases of nucleotide sequences and amino-acid sequences). Let us consider two strings $a = a_1 a_2 \dots a_n$ and $b = b_1 b_2 \dots b_m$, where $a_i \in \Sigma, b_j \in \Sigma,\ 1 \le i \le n,\ 1 \le j \le m$. To refer the i[th] element of string a we will use the notation $a[j]$. For $i < j$ we use $a[i,j]$ as an alternative notation of string $a[i] \dots a[j]$. Consider we have an operation of insertion the symbol "-" somewhere in string a or b.

An alignment (global) [Gusfield, 1997; Mount, 2013] A of two strings a and b is a finite set of transformations by inserting the symbol "-" in a or b, such that after these transformations we get correspondingly the sequences a' and b' of the same length l.

We escape the case when $a'[i] = b'[i] = " - "$ for some $i,\ 1 \le i \le l$. Let us denote the set of all possible alignments by $\mathcal{A}$. It is obvious that $\mathcal{A}$ is finite. For each alignment A we define its value

$$S_A(a,b) = \sum_{\substack{i=1 \\ a'[i]\neq"-" \text{ or } b'[i]\neq"-"}}^{l} s(a'[i], b'[i]) - kw, \tag{5.1}$$

where s is a some function defined over $\Sigma \times \Sigma$, k is the number of symbols "-" in alignment and w is a number which can be interpreted as a "cost" of each occurrence of symbol "-". The function s can be represented as a matrix of size $|\Sigma| \times |\Sigma|$. Such matrixes are called substitution matrixes. For detailed information about calculation of these matrixes we refer to [Dayhoff, 1978; Henikoff, 1992].

The optimal alignment is an alignment A_o such that has maximal value which we will denote by $S(a,b)$.

$$S(a,b) = \max_{A \in \mathcal{A}} S_A(a,b). \tag{5.2}$$

The local alignment is alignment of substring $a[i,j]$ and $b[p,q]$ such that have the maximal value of global alignments overall substrings of a and b, i.e.

$$S_{loc}(a,b) = \max_{\substack{0 \le i \le j \le n \\ 0 \le p \le q \le m}} S(a[i,j], b[p,q]). \tag{5.3}$$

We call the number $S_{loc}(a,b)$ the value of local alignment. Algorithms computing local and global alignments for given two sequences and substitution matrix were brought in [Needleman, 1970; Smith, 1981].

Now let we have a set of strings maybe of different lengths. For the given sequence it is required to find all sequences most "similar" to the given sequence by means defined above (the score of local or

global alignment is maximal). The mentioned problem named as sequence nearest neighbor problem. There may be different variations of notion “similarity” namely gaps may be allowed with different cost functions [Waterman, 1976], or the edit distance and some other distances may be considered. The appropriate selection of similarity measure depends on concrete application.

6.1 Search for Homologous Proteins

At the present there are available many protein databases [Altschul, 1990]. For the new sequenced protein/gene etc. the problem rises to find its properties/functions [Mount, 2013; Gabaldon, 2004; Sleator, 2010]. The experimental way may be costly by means of time/money etc. One of the possible approaches based on homologous proteins with known functions. Two sequences will be called homologous if they have the same ancestor. Proteins having similar sequences are usually homologous. So the function of unknown protein may be predicted by knowing the function of homologous (similar) proteins.

There exist empiric algorithms finding similar sequences for a given sequence (Blast, Fasta etc.) [Mount, 2013; Gusfield, 1997; Altschul, 1990]. Here the meaningful implementation of the algorithm BLAST for proteins [Mount, 2013] is brought below:

Input: Query sequence S[1,n]

- For each sequence a list of words of length 3 L={S[1,3], S[2,4],…,S[n-2,n]} using substitution matrix Blossum62 construct all words with alignment score $\geq$some threshold T. Such words will be called hits;
- For each hit scan the database for exact-match;
- Extend the alignment.

Return high-scoring alignments.

Mention that BLAST is heuristic algorithm. For analyzing of implementation we refer to [Altschul, 1990].

6.2 Protein Secondary Structure Prediction

Now let we have a protein database and for each protein the corresponding secondary structure is available in database. For the protein with unknown secondary structure it is required to predict its secondary structure using the database. The nearest neighbor method may be applied also, to keeping in mind the hypothesis that the homologous sequences have the same secondary structure tendencies [Levin, 1986]. The algorithm described in [Levin, 1986; Levin, 1997] and [Keedwell, 2005]

Algorithm 6.2.1

Input Sequence S of length n, integer m, threshold q.

For each k=1, k≤n-m+1, k++

find set $F_{S,k}$ of sequences of length m from database which align with S[k,k+m] with score ≥q.

Consider the matrix a n×p comment: p-the number of secondary structure conformation, usually p=3, [Keedwell, 2005].

In the ith row and jth column of the matrix write sum of all alignment scores in $F_{S,k}$where in the i-th place corresponds j-th secondary structure conformation.

Return $\alpha_1, \dots, \alpha_n$ where $\alpha_i = \max_{j=1,\dots,p} a_{ij}$.

For example of the performance of algorithm we refer to [Keedwell, 2005].

Conclusion

The problem of nearest neighbor search becomes important in many applied domains such as:

- Textual data mining (document filtering, spam detection, plagiarism detection, etc.),
- Machine vision (human pose estimation, image classification, image search, etc.),
- Computational biology (Search for homologous proteins, protein secondary structure prediction, etc.).

The major difficulty is the amount of available data, when the linear scan becomes practically unrealizable. So the requirements on algorithm are to consider data points from the database as few as possible and to be effective from computational viewpoint.

Bibliography

[Aggarwal, 2012] C. Aggarwal, C. Zhai, Mining text data, Springer, 522pages, 2012

[Altschul, 1990] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, Basic local alignment search tool, Journal of Molecular Biology, 215, pp. 403-410, 1990

[Andoni, 2009] A. Andoni, Neares Neighbor Search: the Old the New, and the Impossible, PhD. Thesis, MIT 2009, 178p.

[Aslanyan, 2013] L. H. Aslanyan, H. E. Danoyan, On the Optimality of the Hash-Coding Type Nearest Neighbour Search Algorithm, Selected works of 9th CSIT conference, pp. 1-6, 2013

[Aslanyan, 2014] L. H. Aslanyan, H. E. Danoyan, Complexity of Hash-Coding Type Search Algorithms with Perfect Codes, JNIT Journalof Next Generation Information Technology, Vol. 5, number 4, pp.26-35, 2014

[Aslanyan, Danoyan, 2013] L. H. Aslanyan, H. E. Danoyan, On the optimality of a hash-coding type search algorithm, Proceedings of the 9th conference CSIT, Yerevan, Armenia, pp. 55-57, 2013

[Cipolla, 2013] R. Cipolla, S. Battiato, G. M. Farinella, Machine learning for computer vision, Springer 2013

[Cover, 1968] T. M. Cover. Estimation by the nearest neighbor rule. IEEE Transactions on Information Theory, 14:21–27, January 1968,

[Dayhoff, 1978] M. Dayhoff , R. Schwartz, B. Orcutt, A model of evolutionary change in proteins, Atlas of protein sequence and structure, Vol. 5, No. suppl 3. pp. 345-351, 1978,

[Gabaldon, 2004] T. Gabaldon, T; M. A. Huynen, Prediction of protein function and pathways in the genome era". Cellular and Molecular Life Sciences 61 (7-8): pp. 930–944, 2004

[Gionis, 1999] A. Gionis, P. Indyk, R. Motwani, Similarity Search in High Dimensions via Hashing, Proceedings of the 25th VLDB Conference, Edinburg, Schotland, pp.518-529, 1999

[Gusfield, 1997] D. Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 534 pages, 1997

[Henikoff, 1992] S. Henikoff and J. Henikoff, Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci USA., vol. 89, pp.10915–10919, 1992

[Keedwell, 2005] E. Keedwell and A. Narayanan, Intelligent Bioinformatics, Wiley, 280p. 2005

[Levin, 1986] J. M. Levin, B. Robson and Jean Gamier, An algorithm for secondary structure determination in proteins based on sequence similarity, FEBS, vol. 205, num. 2, pp. 303-308, 1986

[Levin, 1997] J. M.Levin, Exploring the limits of nearest neighbor secondary structure prediction, Protein Engineering vol.10 no.7 pp.771–776, 1997

[Mount, 2013] D. Mount, Bioinformatics: Sequence and Genome Analysis, second edition, Cold Spring Harbor Laboratory Press, 665 pages, 2013,

[Needleman, 1970] S. Needleman, C. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J Mol Biol.; 48(3):443-453, 1970

[Rivest, 1974] R. Rivest. On the optimality of Elias's algorithm for performing best-match searches. Information Processing, pp. 678–681, 1974

[Salton, 1988] G. Salton and C. Buckley, Term-Weighting approaches in automatic text retrieval, Information Processing & Management Vol. 24, no. 5, pp. 513-523, 1988

[Sebastiani, 2002] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, 34(1), 1-47pp., 2002,

[Shakhnarovich, 2003] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In Proc. IEEE International Conference on Computer Vision, volume 2, 2003,

[Shakhnarovich, 2005] G. Shakhnarovich, Learning Task-Specific Similarity, PhD Thesis, MIT 2005,

[Shakhnarovich, Darell, Indyk 2005] G. Shakhnarovich, T. Darell and P. Indyk, Nearest neighbor methods in Learning and Vision: Theory and Practice, MIT Press, 2005

[Sleator, 2010] R. D. Sleator, P. Walsh, An overview of in silico protein function prediction, Arch Microbiol, no. 192, pp. 151-155, 2010

[Smith, 1981] T. Smith, and M. Waterman, Identifcation of common molecular subsequences, Journal of Molecular Biology, 147, pp. 195-197, 1981

[Torralba, 2008] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2008,

[Wagner, 1974] R. Wagner, M. Fischer, The String-to-String Correction Problem, Journal of the ACM, Volume 21 Issue 1, pp. 168-173, Jan. 1974

[Waterman, 1976] M. S. Waterman, T. F. Smith and W. A. Beyer Some Biological Sequence Metrics, Advances in Mathematics, 20, pp. 367-387, 1976

[Yang, 1997] Y. Yang and J. Pedersen, A comparative study on feature selection in text categorization, In Proceedings of ICML-97, 14th International Conference on Machine Learning, 412–420pp., 1997

Authors' Information

Hayk Danoyan *– Institute for informatics and automation problems of NAS RA, 1, P. Sevak street, Yerevan 0014, Armenia, e-mail: hed@ipia.sci.am*

Major Fields of Scientific Research: Nearest Neighbor Search, Discrete optimization, Coding theory, Machine Learning, Bioinformatics