Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki Nr 10, Rok 7, 2013, s. 25-37

Metrics and similarities in modeling dependencies between continuous and nominal data

Michał Grabowski*[1], Michał Korpusik[2]

[1]Warsaw School of Computer Science, Warsaw
[2]University of Warmia and Mazury, Olsztyn

Abstract

Classification theory analytical paradigm investigates continuous data only. When we deal with a mix of continuous and nominal attributes in data records, difficulties emerge. Usually, the analytical paradigm treats nominal attributes as continuous ones via numerical coding of nominal values (often a bit ad hoc). We propose a way of keeping nominal values within analytical paradigm with no pretending that nominal values are continuous. The core idea is that the information hidden in nominal values influences on metric (or on similarity function) between records of continuous and nominal data. Adaptation finds relevant parameters which influence metric between data records. Our approach works well for classifier induction algorithms where metric or similarity is generic, for instance k nearest neighbor algorithm or proposed here support of decision tree induction by similarity function between data. The k-nn algorithm working with continuous and nominal data behaves considerably better, when nominal values are processed by our approach. Algorithms of analytical paradigm using linear and probability machinery, like discriminant adaptive nearest-neighbor or Fisher's linear discriminant analysis, cause some difficulties. We propose some possible ways to overcome these obstacles for adaptive nearest neighbor algorithm.

Keywords: *k-nearest neighbors algorithm, data metrics, classification, continuous data, nominal data*

* E-mail: mgrabowski@poczta.wwsi.edu.pl.

Manuscript received September 29, 2013

1 Introduction

Classification theory seems to be divided into analytical paradigm investigating continuous data only and combinatorial paradigm investigating nominal data only. When we deal with a mix of continuous and nominal attributes in data records, difficulties emerge. Usually, the combinatorial paradigm treats continuous attributes as nominal ones via discretization. In spite of several objections that can be raised (for instance, we pretend in this way that continuous data are nominal, while they are not), the methods of discretization are more or less canonical, are based on theory which progresses and very often gives good results in applications (see [1]).

On the other hand, the analytical paradigm treats nominal attributes as continuous ones via numerical coding of nominal values (often a bit ad hoc). There are no canonical methods which find appropriate numerical coding of nominal values. Known algorithms are very different and apparently non uniform. We can raise well known objections against the idea of numerical coding of nominal values within analytical paradigm of classification theory.

- Usually attributes are inherently nominal (for example attribute sex). After numerical coding values like "one and a half of female sex" can appear in computations and their results may not be credible.
- Assume that the analysis of semantic of nominal values gives some numerical coding that can be accepted as reasonable. Then we have to deal with a nontrivial problem of proper rescaling remaining continuous attributes.

Majority of analytical paradigm algorithms heavily use continuous linear and probability machinery. It is hard to include discrete values without breaking out the basic principles behind these algorithms. This objection also concerns the reasonable Bayesian numerical coding of nominal values (*scoring methods*, see [2]). We propose a way of keeping nominal values within analytical paradigm without pretending that nominal values are continuous. The core idea is that the information hidden in nominal values influences metric (or on similarity function) between data records. We take the Hamming distance between nominal parts of data records as the leading parameter influencing metric (or similarity) between data records. Formal definitions based on kernel functions implementing our idea are given in section 2. Examples of basic kernel functions are given as well. As far as we know our approach works well for classifier induction algorithms where metric or similarity is generic (for instance k nearest neighbor algorithm or proposed here support of decision tree induction by similarity function between data).

In section 3, we define modifications of k-nearest neighbor algorithm in order to process mix of continuous and nominal data. Particular kernel functions are involved. Adaptation is used in order to find kernels parameters influencing metric or similarity between data records. The choice of these additional parameters depends on particular kernel function used by the algorithm. Equipped with similarity between data we define a new version of decision tree

induction. We think that classic rule induction algorithms can be enhanced with similarity between data as well. We end this section with a proposal of k-means clustering algorithm processing mix of continuous and nominal data.

The results of introductory experiment with k-nn algorithm on real data set are described in section 4. This algorithm behaves considerably better, when nominal data are processed in accordance with our approach. Classification accuracy given by cross-validation method is about 6-8% better than the accuracy of algorithm with ad hoc nominal numerical coding. It seems that the idea has some significance. Algorithms of analytical paradigm using linear and probability machinery, like discriminant adaptive nearest-neighbor (*DANN*) or Fisher's linear discriminant analysis (*LDA*), cause difficulties when we try to generalize them for mix of continuous and nominal data. The source of these problems is the fact, that these algorithms are not generic with respect to metric between data.

In section 5, we sketch some possible way to overcome these difficulties for DANN algorithm. We propose to use the results of computational geometry on low-distortion embedding of finite metric spaces into linear spaces (see [3], [4]). The algorithms like Fisher's linear discriminant analysis are immune against approach which use embedding of suitable finite metric spaces into linear spaces. Our work on generalization for mix of continuous and nominal data of analytical paradigm algorithms using linear and probability machinery is at the very beginning stage yet.

2 Metrics and similarities via kernel functions

Let p be a number of continuous attributes. Let s be a number of nominal attributes. Let for $j = 1, 2, \ldots, s$ A_j = be a finite domain of values of the j-th nominal attribute. We present records as a pairs of continuous and nominal data $r = (x, n)$ where $x \in R^p$ and $n \in A_1 \times \ldots \times A_s$. In the sequel p denotes the dimension of continuous data and s denotes the number of nominal attributes.

Let $n_1 = (v_1^1, \ldots, v_s^1), n_2 = (v_1^2, \ldots, v_s^2) \in A_1 \times \ldots \times A_s$ be the nominal parts of data records $r_1=(x_1, n_1)$, $r_2=(x_2,n_2)$. We define the *Hamming distance* between nominal parts of records r_1, r_2 as follows:

$$H(n_1, n_2) = \frac{|\{ j \,|1 \le j \le s \text{ and } v_j^1 \ne v_j^2\}|}{s}$$

For a given $0 \le \varepsilon < 1/s$ we define the *Hamming distance* $H_\varepsilon(n_1, n_2)$ that we shall deal with.

$$H_\varepsilon(n_1, n_2) = if\ H(n_1, n_2) > 0\ then\ H(n_1, n_2)\ else\ \varepsilon$$

Parameter ε is involved since it is useful in defining some kernel functions like *spherical kernel* defined in the sequel. We define now kernel function. Let $0 \le \varepsilon < 1/s$ be an arbitrary set real number.

Def. 1. A function $K_\varepsilon : [\varepsilon, +\infty) \times R^p \times R^p \rightarrow R^+$ is a kernel function if the following conditions hold:

1. $\forall x \in R^p . \forall a \in [\varepsilon, +\infty) . K_\varepsilon(a, x, x) = 0$
2. $\forall x, y \in R^p . \forall a \in [\varepsilon, +\infty) . K_\varepsilon(a, x, y) = K_\varepsilon(a, y, x)$
3. $\forall x, y, z \in R^p . \forall a \in [\varepsilon, +\infty) . K_\varepsilon(a, x, y) \leq K_\varepsilon(a, x, z) + K_\varepsilon(a, z, y)$
4. $\forall a_1, a_2 \in [\varepsilon, +\infty) . \forall x, y \in R^p . (a_1 > a_2 \rightarrow K_\varepsilon(a_1, x, y) > K_\varepsilon(a_2, x, y))$ □

The first three conditions say that for set $a \in [\varepsilon, +\infty)$ the function $K_\varepsilon(a,\cdot,\cdot): R^p \times R^p \rightarrow R^+$ is a metric on R^p. Fourth one says that parameterized by *a* distance $K_\varepsilon(a,\cdot,\cdot)$ is growing when *a* is growing.

Assume we are given a kernel function $K_\varepsilon(a, x, y)$. Similarity function between data records $r_1 = (x_1, n_1),\ r_2 = (x_2, n_2)$ is defined by

$$\rho(r_1, r_2) = K_\varepsilon(H_\varepsilon(n_1, n_2), x_1, x_2)$$

The function ρ is a similarity between data records since it is reflexive and symmetric.

Examples of kernel functions

1. Additive kernel (the idea of Jakub Zahorski [5]).

Let $0 \leq \alpha \leq 1$ be a given weight and let $\varepsilon=0$.

$$K_\varepsilon(a, x, y) = \alpha * (renormalized\ to\ [0,1]\ Euclidian\ distance\ between\ x, y) + (1 - \alpha) * a$$

induced similarity function by additive kernel is a metric for data records.

2. Spherical Gaussian kernel

Let $a \in R^+$ be given and let $\varepsilon=0$. (The Hamming distance $H(n,m)$ will be taken as a). Let $v = 1/(2\pi * a^{2/p})$. Let $\sum_a$ be a diagonal matrix with v value on diagonal. Gaussian multivariate density function f with $\sum_a$ as the covariance matrix is taken. We use the known formula:

$$f_a(x, y) = e^{\left(\left(-\frac{1}{2}\right) * (x-y)^T \sum_a{}^{-1} (x-y)\right)} / (2\pi)^{p/2} \left|\sum_a\right|^{1/2}$$

where $|\sum_a|$ is determinant of $\sum_a$ and $x, y \in R^p$. Hence, the given $a \in R^+$ is viewed as the "height" of spherical Gaussian multivariate density. We define now spherical Gaussian kernel (Fig. 1) by

$$K_\varepsilon(a, x, y) = Euclidian\ distance\ in\ R^{p+1}\ of\ the\ points\ \big(x, f_a(x, x)\big), \big(y, f_a(x, y)\big).$$

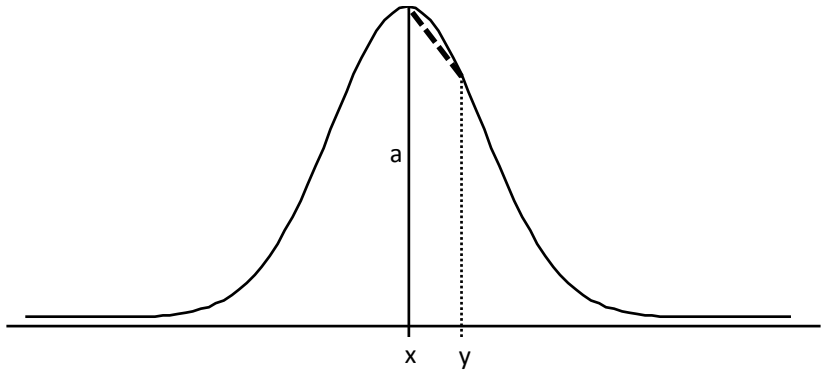

Figure 1. Spherical Gaussian kernel for *p=1*

3. Elliptical Gaussian kernel

Let $a \in R^+$ be given and let *ε=0*. Let $\sum_a$ be an arbitrary symmetric matrix such that $a = 1/((2\pi)^{p/2} * |\sum_a|^{1/2}$ (*a* is viewed as the height of Gaussian density). Again we use Gaussian multivariate density $f_a(x, y)$ with $\sum_a$ as covariance matrix now:

$$f_a(x,y) = e^{\left(\left(-\frac{1}{2}\right)*(x-y)^T {\sum_a}^{-1}(x-y)\right)} / (2\pi)^{p/2} \, |\sum_a|^{1/2}$$

We define elliptical Gaussian kernel by

$K_\varepsilon(a, x, y) = Euclidian\ distance\ in\ R^{p+1}\ of\ the\ points\ \big(x, f_a(x,x)\big), \big(y, f_a(x,y)\big)$

It does not have anything in common with probability here. We have chosen Gaussian kernels in order to obtain exponential influence of parameter *a* on similarity between *x, y*. Other kernels can give similar effect also.

4. Spherical kernel

Let $0 < \varepsilon < 1/s$ be given. Let us consider the set $D = \{\varepsilon, 1/s, \ldots, s\text{-}1/s, 1\}$ of all possible values of Hamming distance H_ε between nominal parts of data records. Let for $a \in D$, $S_p(a)$ be a *p*-dimensional sphere in R^{p+1} lying with its south pole somewhere on *p*-dimensional hyperplane *P* in R^{p+1}. We consider the spheres $S_p(a)$ with their north pole removed. For each $a \in D, h_a: P \to S_p(a)$ is the stereographical embedding of *P* in $S_p(a)$.We define spherical kernel by

$K_\varepsilon(a, x, y) = spherical\ distance\ on\ S_p(a)\ of\ the\ points\ h_a(x), h_a(y).$

Ellipsoids or other affine images of spheres could be used instead of spheres as well.

3 Algorithms

In this section we show that our approach works for classifier induction algorithms using similarity (or metric) between data in order to construct a classifier. The known example is the k nearest (k most similar, when similarity instead of metric is used) neighbor algorithm (*k-nn algorithm*). We use defined in [6] tolerance between data sets in order to inject similarity between data into computation of decision tree induction algorithm. Thus we obtain a new version of decision tree induction where our approach is applicable. We think that a similar result can be obtained for rule induction algorithms. The third example where our approach works is defined here modification of k means clustering algorithm processing records of continuous and nominal data. No values like "one and a half of female sex" are involved.

K-nearest neighbor for continuous and nominal data modulo a kernel function

Assume we are given a similarity function (or metric) between data records. K-nn works as follows: for a given record *r* which is to be classified compute the set *N* of k most similar to *r* records of the training set. Next we assign *r* to the most frequent class in *N*.

1. K-nn modulo additive kernel
Adapt a weight $0 \leq \alpha \leq 1$ (see the definition of additive kernel) in order to minimize, computed via cross-validation, classification error. Classify by k-nn with respect to metric induced by additive kernel with optimal *a* selected by adaptation.

2. K-nn modulo spherical Gaussian kernel
Rescaling the height of spherical multivariate Gaussian density factor *h>0* as adaptation parameter is taken. The definition of the similarity function that we work with is the following: $\rho(r_1, r_2) = K_\varepsilon(h * H_\varepsilon(n_1, n_2), x_1, x_2)$ where K_ε is the spherical Gaussian kernel. K-nn modulo this kernel works as follows: adapt the rescaling factor *h* in order to minimize, computed via cross-validation, classification error. Classify by k-nn with respect to similarity induced by spherical Gaussian kernel with optimal *h* selected by adaptation.

Several different techniques where used as adaptive methods in our introductory experiment.

3. K-nn modulo elliptical Gaussian kernel
Covariance matrix in Gaussian multivariate density formula determines the elliptical shape of its density function. We propose to adapt this shape for each possible Hamming distance $a \in D = \{0, 1/s, \ldots, s-1/s, 1\}$ separately. Let for a given $a \in D$, $\sum_a$ be a symmetrical matrix such that $a = 1 / ((2\pi)^{p/2} * |\sum_a|^{1/2})$ – we enforce here that the height of Gaussian multivariate density is equal to *a*. K-nn modulo elliptical Gaussian kernel works as follows: adapt matrices $\sum_a$ in order to minimize, computed via cross-validation, classification error. Classify

by k-nn with respect to similarity induced by elliptical Gaussian kernel with optimal matrices $\sum_a$ selected by adaptation. Unfortunately, we have quite a lot of adaptation parameters: $(s+1)*p^2/2$, where p is the dimension of continuous data, s is the number of nominal attributes in data records. It is plain enough that the problem of Gaussian elliptical kernel shape adaptation requires further study.

Decision tree induction for continuous and nominal data modulo a kernel function

The essence of decision tree induction is the test choice criterion: which one of the currently available tests should be placed into the node under construction? We shall sketch a formalization of the following heuristics: the test should divide the current training set into subsets as dissimilar as possible while the elements within each subset should be as similar as possible. The similarity function between data sets defined in [6] is used.

Let ϱ be a similarity function on data records and $tr \in R^+$ be a certain threshold. We work with environments $n(r) = \{r' | \rho(r, r') < tr\}$, where *r, r'* are data records. Let *X, Y* be the sets of data records. The degree of inclusion of *X* in *Y* is defined as follows:

$$\vartheta(X,Y) = \frac{|\{r \in X | \exists r' \in Y.\ r' \in n(x)|}{|X|}$$

Therefore if $r \in X$ has a similar to *r* record $r' \in Y$ then *r* is considered as element of *Y*. The similarity between data sets *X, Y* is defined (see [6]) as follows:

$$\tau(X,Y) = \min(\vartheta(X,Y), \vartheta(Y,X))$$

The value $\tau(X,Y)$ near 0 means that *X, Y* are dissimilar; $\tau(X,Y)$ near 1 means that *X, Y* are similar. The similarity $\tau(X,Y)$ has nothing to do with Lopez de Mantaras distance between tests (see [7]). Let $T=\{r_1,\ldots,r_n\}$ be a current training set and let us assume that a test *t* divides *T* into subsets $T_1,\ldots,T_m$. Let $s = \sum_{i<j} \tau(T_i, T_j)$, $d_{ij} = \tau(T_i, T_j)/s$ – renormalized to [0,1] similarity $\tau(T_i, T_j)$. Let *E* be the entropy of renormalized to [0,1] similarities $\tau(T_i, T_j)$: $E = -\sum_{i<j} d_{ij} * \log(d_{ij})$. Let *m* be the arithmetic mean of the similarities $\tau(T_i, T_j)$, i<j. Finally, let $E(t)$ be the classic entropy of the test *t*, measuring disorder in the inherited from *T* partitions of the sets T_i into classification categories.

We can now formulate our test choice criterion: choose the test *t* such that $(\frac{E(t)*m}{E}) \rightarrow min$. This means that the chosen test *t* should minimize disorder in partitions of T_i into classes and mean similarity between T_i, T_j should be small (i.e. T_i, T_j should be dissimilar) and renormalized to [0,1] similarities $\tau(T_i, T_j)$ should be more or less close to each other. Notice that our test choice criterion depends on accepted threshold *tr* of similarity environments *n(r)*.

Let us choose a kernel function *K*. Assume that adaptation parameters $Q=\{q_1,\ldots,q_a\}$ are related to kernel *K*. Decision tree induction modulo kernel *K* works as follows: adapt threshold *tr* and parameters *Q* in order to minimize, computed via cross-validation, classification error. Decision trees constructed at the consecutive stages of adaptation are built by decision tree induction equipped with defined above test choice criterion. This criterion is determined by

the current threshold *tr* and by similarity function induced by kernel *K* and current values of parameters *Q*. Classify by decision tree T which is built with respect to selected parameters by adaptation. These parameters define test choice criterion that the decision tree induction algorithm is working with.

Tree built by the classic decision tree induction with entropy driven test choice criterion tends to overfit to training set (see [8]). We hope that our version of decision tree induction is less vulnerable to overfitting. Decision trees process continuous and nominal attributes equally well and no numerical coding of nominal values is involved. Nevertheless, if the dependencies between continuous and nominal data are controlled by a more subtle kernel, then presented version of decision tree induction can make sense. On the other hand we wanted to show that some known classifier induction algorithms can be supported by similarity function or by metric.

K clustering for continuous and nominal data modulo a kernel function

The heart of k means clustering is the algorithm computing the mean (with respect to metric, or similarity, under consideration) record $r=(x,n)$ of the given finite set X of records. The problem is that we are not allowed to use linear space structure (for example no computations like $(r + r')/2$, where r, r' are data records). The algorithm must be generic with respect to metric (or similarity) between data records since otherwise our approach will not be applicable. We define here a preliminary generic, with respect to metric function, algorithm computing mean record.

Let ϱ be a metric (or similarity) under consideration. Let X be a finite set of records for which the mean record is to be found. Let us assume for a while that we have determined a finite set Y of candidates for mean record of X. Let r be an arbitrary chosen record. Let $\sigma_r = \sum_{r' \in X} \rho(r, r')$. Let $E_r = -\sum_{r' \in X} \frac{\rho(r,r')}{\sigma_r} * \log(\frac{\rho(r,r')}{\sigma_r})$ – the entropy of renormalized to [0,1] distances $\varrho(r,r')$. We choose r from Y, as mean record of X, such that $\frac{\sigma_r}{E_r} \rightarrow min$. It means that the sum σ_r should be possibly small while the distances $\varrho(r,r')$, $r' \in X$ should be close to each other. Now we should construct a finite set Y of candidates for mean record of X.

Assume that $X = \{r_1, \ldots, r_n\}$ and for $i = 1, \ldots, n$ $r_i = (x_i, n_i)$, $x_i \in R^p, n_i \in A_1 \times \ldots \times A_s$, where for $j=1,\ldots,s$ A_j is the finite domain of j-th nominal attribute. Let $CX = \{x_1, \ldots, x_n\}$. Let R be a cube in R^p covering CX. Let CY be a finite set of a lattice points of cube R. The density of lattice is determined by chosen parameter $\varepsilon>0$.

We take $Y = CY \times (A_1 \times \ldots \times A_s)$ as the set of candidates for mean record of X. k means clustering modulo a kernel K works with respect to distance between records of continuous and nominal data induced by kernel K and it uses defined above algorithm computing mean records of current clusters.

Above version of k means clustering shows that it possible to develop clustering algorithms processing records of continuous and nominal data with no numerical coding of nominal values.

4 Introductory experiment

During our work some preliminary tests were made. We found out that kernel approach works better than other solutions when we have to deal with continuous-nominal attributes mix. We used k nearest (most similar) neighbor algorithm with different distance and kernel functions. Each function classification accuracy (for a given parameter k) is a mean value of series of tests using 10-fold cross validation. As a classification rule we simply adopt most frequent class amongst k nearest neighbors. Tested functions are: Euclidean distance, additive kernel and spherical Gaussian kernel (Fig. 2). Several algorithms from MATLAB's Optimization Toolbox were used to adapt kernels parameters in order to minimize classification error.

For our experiment we used Australian credit approval data set (see [9]). This data set is interesting, because there's a good mixture of attributes – continuous, nominal with small and large number of values. Before any test was conducted all continuous values were rescaled to $\boldsymbol{N(0,1)}$ normal distribution and nominal values were coded as natural numbers. In this particular data set records are classified into one of two classes. It's one of the reasons why we've decided to additionally examine Euclidean distance function with Bayesian coding.

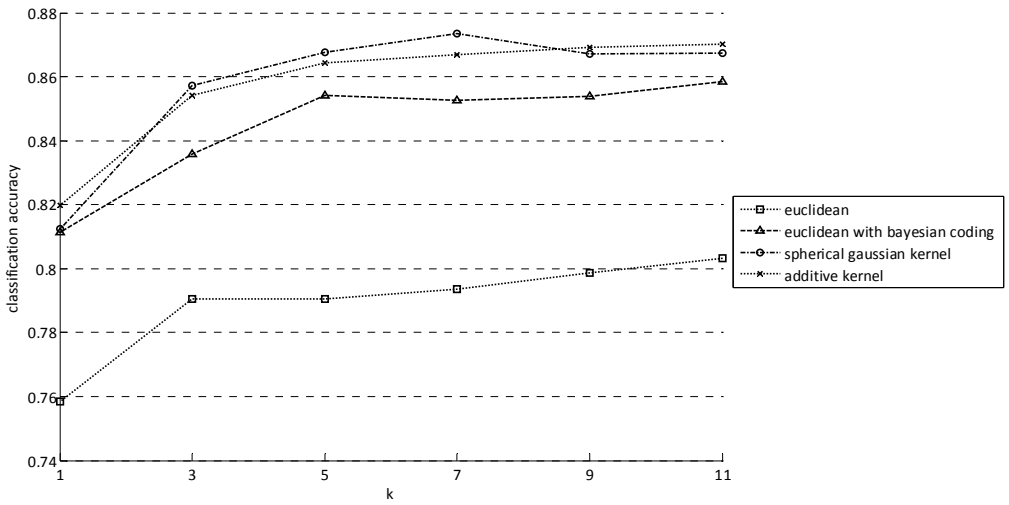

Figure 2. Classification accuracy for tested functions

Presented figure shows classification accuracy (0...1) for different distance and kernel functions. In general additive kernel produced best results. Classification accuracy increase varies from 0.61 to almost 0.74 comparing to regular Euclidean distance. Best results were obtained for $0.2 < \alpha < 0.3$. It looks like, in selected range, additive kernel classification accuracy monotonically grows within growth of the k parameter. Spherical Gaussian kernel also produced very good results. Its adaptation parameter (as well as result) dependency on k is not trivial and needs to be examined more. Classification accuracy gains almost 0.8 in the best solution. Although Euclidean distance with Bayesian coding gave quite good results one must realize that it works fine only when it deals with two classes of data. Our approach does not suffer from this weakness. Table below presents the best result for each distance/kernel function.

Table 1. Best result for tested functions

Function	Best classification accuracy
Euclidean distance	0.8031449
Euclidean distance with Bayesian coding	0.8584058
Spherical Gaussian kernel	0.8735178
Additive kernel	0.8701449

Introductory experiment has shown that the idea of kernel functions is worth exploring further. All tests have a very basic nature. There are still a lot of open questions and tests to run. We're looking forward to obtain results from different data sets. Remaining kernels need to be examined as well. Also we would like to compare our results with other algorithms including these from combinatorial paradigm (with continuous values discretization).

5 Final remarks

The question whether our approach can be applied to more complicated algorithms, especially ones using linear space and probability machinery, is of great importance. Two examples of such algorithms are considered here: DANN - discriminant adaptive nearest neighbor ([10]) and LDA – well known Fisher's linear discriminant analysis. We point out some difficulties related to generalization of these algorithms for mix of continuous and nominal data. While DANN algorithm seems to be tractable, LDA is more problematic.

We begin with reminding the DANN algorithm (see [10]) processing continuous data $x \in R^p$. The DANN metric at a point x_0, which is to be classified, is defined by $\rho(x, x_0) = (x - x_0)^T \sum (x - x_0)$ where:

$$\sum = W^{-\frac{1}{2}}\left[W^{-\frac{1}{2}}BW^{-\frac{1}{2}} + \varepsilon I\right]W^{-\frac{1}{2}} = W^{-\frac{1}{2}}[B^* + \varepsilon I]W^{-\frac{1}{2}}.$$

Matrix W is the within-class covariance matrix and B is the between classes covariance matrix. They are computed using only 50 nearest neighbors of x_0. The above formula first spheres the neighborhood data with respect to W and then stretches the neighborhood in the zero-eigen-value directions of B^* (the between-classes matrix for the sphered neighborhood data). The ε parameter rounds the neighborhood to an ellipsoid. Thus DANN adapts to the shape of boundary between classes in the neighborhood of x_0. It computes k nearest with respect to DANN metric, neighbors of x_0 and then assigns x_0 to the most frequent class in the computed neighborhood.

Let T be a training set of records $r=(x,n)$ of continuous and nominal data and let r_0 be a query record. Assume that we are working with a kernel K such, that the induced by K similarity ϱ is the metric (for instance, additive kernel). Let us consider the finite metric space $X = (T \cup \{r_0\}, \rho)$. Assume that there is an embedding e of X into linear space R^q for some dimension q with Euclidian metric on R^q. Then we compute DANN algorithm in the space R^q with the training set $e(T)$ and with the query point $e(r_0)$. The within-class covariance and between- classes covariance matrices W and B considered in R^q make sense since the original distances from the space X are preserved. Unfortunately, there are finite metric spaces X for which an embedding e does not exist (for any q, see [3], [4]). The computational geometry proposes some remedy: embedding e exists if we allow that e expands or contracts a bit (low-distortion) original distances between points from X. If the distortion of e is really low, DANN algorithm working in R^q should give satisfactory results. There is a beautiful theory about low-distortion embedding of finite metric spaces into linear spaces. Again, see [3], [4] for formal definitions and results.

The requirement that a kernel K should induce a metric between data records is the serious disadvantage of the proposed version of DANN algorithm. We suspect that kernels needed for analyzing complicated mixtures of continuous and nominal data does not satisfy this property.

Approach via embedding of finite metric space into linear space R^q does not work well for Fisher's LDA algorithm. For example, even low distortion of embedding e of finite (!) metric space can give the following arrangement in R^2 of four records:

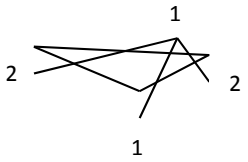

Figure 3. Records arrangement

No line discriminates classes 1 and 2. What can we do? Let p be the dimension of continuous data and A_j be the finite domain of j-th nominal attribute and ϱ be the metric induced by a kernel that we work with. Maybe an embedding with low distortion of the whole metric space $(R^p \times A_1 \times ... \times A_s, \rho)$, into a manifold with a suitable structure would help. Unfortunately, interesting manifolds have a lot of geometry but no linear space structure. It seems that a structure with metric and with defined notions of line, hyperplane, orthogonal projection is sufficient for modeling LDA algorithm. We do not dare to call this vision "the approach". We merely argue that maybe it is too early to say that LDA algorithm is definitely not tractable in presented context.

6 Conclusion

Our approach works well for algorithms using metric between data in a generic way. For instance k-nn algorithm or our versions of k means clustering and decision tree induction. It is plain enough that the approach sketched for DANN algorithm is at the very beginning stage of research. A lot of work developing theory giving foundations to our approach should be done. For instance further examining kernels properties can lead to discovery of embedding metric spaces in question into interesting manifolds. Simultaneously a lot of work with serious experiments should be done. It is the task for many researchers we think, and that is why we have decided to share our core idea at the current stage.

References

[1] Hung Son Nguyen, *Approximate Boolean Reasoning: Foundations and Applications in Data Mining*, in: *Transactions on Rough Sets V*, (eds.) Peters J.F., Skowron A., LNCS 4100, 2006

[2] Koronacki J., Ćwik J., *Statystyczne systemy uczące się, Statistical Learning* (in polish), Akademicka Oficyna Wydawnicza EXIT, Warszawa 2008

[3] Linial N., *Finite metric spaces – Combinatorics*, *Geometry and Algorithms*, Symposium on Computational Geometry, 2002

[4] Indyk P. , Matousek J., *Low-Distortion Embedding of Finite Metric Spaces*, Handbook of Discrete and Computational Geometry (2nd edition), (eds.) Goodman J.E., O'Rourke J., CRC Press, LLC 2004

[5] Zahorski J., private communication

[6] Doherty P., Łukaszewicz W., Skowron A., Szałas A., *Knowledge Representation Techniques. A rough set approach*, "Studies in Fuzziness and Soft Computing" 202, Springer-Verlag 2006

[7] Lopez de Mantaras R., *A Distance–Based Attribute Selection Measure for Decision Tree Induction*, "Machine Learning" 1991, Vol. 6

[8] Hastie T., Tibshirani R., Friedman J., *The Elements of Statistical Learning*, Springer Series in Statistics, 2001

[9] Frank A., Asuncion A., *UCI Machine Learning Repository*, University of California, Irvine, School of Information and Computer Science, 2010, http://archive.ics.uci.edu/ml

[10] Hastie T., Tibshirani R., *Discriminant Adaptive Nearest Neighbor Classification*, "IEEE Pattern Recognition and Machine Intelligence", Vol. 18, No. 6