národní
úložiště
šedé
literatury

**Classifier Based on Inverted Indexes of Neighbors**

Jiřina, Marcel
2008

Dostupný z http://www.nusl.cz/ntk/nusl-39998

Dílo je chráněno podle autorského zákona č. 121/2000 Sb.

Tento dokument byl stažen z Národního úložiště šedé literatury (NUŠL).

Datum stažení: 30.09.2024

Další dokumenty můžete najít prostřednictvím vyhledávacího rozhraní nusl.cz .

**Institute of Computer Science**

**Academy of Sciences of the Czech Republic**

# Classifier Based on Inverted Indexes of Neighbors

**Marcel Jiřina and Marcel Jiřina, jr.**

Technical Report No. V-1034

November 2008

Abstract

A new method for the classification of data into classes is presented. The method is based on the sum of reciprocals of neighbors' indexes. We show that neighbors' indexes are in close relation to the polynomial transform of the neighbors' distances. The sum of the reciprocals of indexes for all neighbors forms truncated harmonic series due to a finite number of its elements. For the neighbors of one class there is a sum of the selected elements of this truncated series. It is proved that the ratio of these sums gives just the probability that the point to be classified – the query point – is of that class. The classification ability is demonstrated on real-life data from the Machine Learning Repository and the results are compared with published results obtained through other methods.

Keywords:

multivariate data, correlation dimension, correlation integral, decomposition, probability density estimation, harmonic series, classification.

# Classifier Based on Inverted Indexes of Neighbors

*Marcel Jiřina* and *Marcel Jiřina, Jr.* (marcel@cs.cas.cz)

The presented material describes an elaborated yet simple classification method (IINC) that can outperform a range of standard classification methods of data mining, e.g. K-Nearest neighbors, Naïve Bayes Classifiers' as well as SVM.

First, we will provide a shot overview of the basic idea of the IINC and its features. Second, we will demonstrate the power of the IINC on data sets from two well-known repository real-life tasks.

**Classifier Background and Features**

In general, if we have estimates of the probability that a given sample (query point) belongs to a given class, we can easily construct a classifier. We just compare the individual probabilities and select the class with the highest probability. The presented IINC works in the same way, but the probabilities are estimated in a special way that is based on summing up the inverted indexes of neighbors.

We show a practical approach to the classification of data into two classes (extending the classifier to be able to classify to more than two classes is then straightforward).

Let all samples of the learning set regardless of the class be sorted according to their distances from the query point *x*. Let indexes be assigned to these points so that index 1 is assigned to the nearest neighbor, index 2 to the second nearest neighbor etc.

Let us compute sums $S_0(x) = \frac{1}{N_0} \sum_{i=1\,(c=0)}^{N} 1/i$ and $S_1(x) = \frac{1}{N_1} \sum_{i=1\,(c=1)}^{N} 1/i$, i.e. the sums of the reciprocals of the indexes of samples from class $c = 0$ and from class $c = 1$. $N_0$ and $N_1$ are the numbers of samples of class 0 and class 1, respectively, $N_0 + N_1 = N$, , $N$ is the total number of samples available.

The probability that point *x* belongs to class 0 is

$$p(c = 0 \mid x) \cong \frac{S_0(x)}{S_0(x) + S_1(x)}$$

and similarly the probability that point *x* belongs to class 1 is

$$p(c = 1 \mid x) \cong \frac{S_1(x)}{S_0(x) + S_1(x)}.$$

When a discriminant threshold $\vartheta$ is chosen (e.g. $\vartheta = 0.5$), then if $p(c = 1 \mid x) \geq \vartheta$ point *x* is of class 1 else it is of class 0. This is the same procedure as in other classification approaches where the output is the estimation of probability (naïve Bayes) or any real valued variable (neural networks). The value of the threshold can be optimized with regard to the minimum classification error.

**Features**

As shown, the IINC is very simple. It is based only on the sum of inverted indexes of the nearest neighbors. It opens the question whether it is as powerful as stated above.

In the above formulas the actual data do not appear directly, but are hidden behind the indexes that express their distance from a given sample (query point). To be able to get the indexes we have to sort the original data according to their distances from a particular sample. The only information we work with is their order, not the real distance! To compare distances we need proper metrics (just the $L_1$ (absolute) metrics yields the best results). And so on. In other words, there are many assumptions that have to be fulfilled (and fortunately they are fulfilled in standard classification tasks) to concentrate them into a simple presented classification algorithm IINC. To vindicate the correctness of the algorithm we offer a deeper mathematical insight into the IINC and demonstrate the IINC on real-life classification tasks.

**Mathematical background of IINC**

Let us consider partial influences of individual points on the probability that point *x* is of class *c*. Each point of class *c* in the neighborhood of point *x* adds a little to the probability that point *x* is of class *c*, where *c* = {0, 1} is the class mark. This influence grows larger the closer the point considered is to point *x* and vice versa. This observation is based on the finding of [4] that the nearest neighbor has the largest influence on the proper estimation to what class point *x* belongs. Let us assume – for proof see [1] – that the influence to the probability that point *x* is of class *c* (the nearest neighbor of class *c*) is 1, the influence of the second nearest neighbor is 1/2, the influence of the third nearest neighbor is 1/3, etc. We show further that just these values of influence lead to improved classification. Let $p_1(c|x, r_i)$ be the probability that query point *x* is of class *c* if neighbor point number *i* is of the same class as point *x*, *K* is a constant that is used to normalize the probability that point *x* belongs to any class to 1:

For the first (nearest) point *i* = 1 $p_1(c \mid x, r_1) = K.1$,

for the second point *i* = 2 $p_1(c \mid x, r_2) = K\frac{1}{2}$ ,

and so on, generally for point No. *i* $p_1(c \mid x, r_i) = K\frac{1}{i}$ .

Individual points are independent and then we can sum up these probabilities. Thus we add the partial influences of *k* individual points together by summing up

$$p(c \mid x, r_k) = \sum_{i=1(c)}^{k} p_1(c \mid x, r_i) = K \sum_{i=1(c)}^{k} 1/i\,.$$

The sum goes over indexes *i* for which the corresponding samples of the learning set are of class *c.* Let

$$S_c = \sum_{i=1(c)}^{k} 1/i$$

and let

$$S = \sum_{i=1}^{N} 1/i$$

(This is, in fact, so-called harmonic number $H_N$, the sum of truncated harmonic series.) The estimation of the probability that query point *x* belongs to class *c* is

$$p(x \mid c) = \frac{S_c}{S}\,.$$

The approach is based on the hypothesis that the influence, the weight of a neighbor, is proportional to the reciprocal of its order number just as it is to its distance from the query point.

The hypothesis above is equivalent to the assumption that the influence of individual points of the learning set is governed by Zipfian distribution (Zipf's law).

It is also possible to show that the use of 1/i has a close connection to the correlation integral and correlation dimension and thus to the dynamics and true data dimensionality of processes that generate the data we wish to separate. It generally leads to better classification.

**Theorem.** Let the task of classification into two classes be given. Let the size of the learning set be *N* and let both classes have the same number of samples. Let *i* be the index of the *i*-th nearest neighbor of point *x* (without considering neighbor's class) and $r_i$ be its distance from point *x*. Then

$$p(c \mid x) = \lim_{N\to\infty} \frac{\sum_{i=1(c)}^{N} 1/i}{\sum_{i=1}^{N} 1/i} \tag{1}$$

(the upper sum goes over indexes *i* for which the corresponding samples are of class *c*) is the probability that point *x* belongs to class *c*. The proof can be found in [1].

**Note.** For a different number of samples of one and the other class formula (1) has the form

$$p(c \mid x) = \lim_{N\to\infty} \frac{\frac{1}{N_c} \sum_{i=1(c)}^{N} 1/i}{\frac{1}{N_0} \sum_{i=1(0)}^{N} 1/i + \frac{1}{N_1} \sum_{i=1(1)}^{N} 1/i} \ .$$

It is only a recalculation of the relative representation of different numbers of samples of one and the other class.

For more than two classes, say *C* classes, the equation is

$$p(c \mid x) = \lim_{N\to\infty} \frac{\frac{1}{N_c} \sum_{i=2(c)}^{N} 1/r_i^q}{\sum_{k=1}^{C} \frac{1}{N_k} \sum_{i=2(c)}^{N} 1/r_i^q} \ .$$

**Demonstrations of the IINC on Real-life Tasks**

***Tasks from UCI Machine Learning Repository - a Comparison with Published Results***

The classification ability of the IINC presented here was tested using real-life tasks from UCI Machine Learning Repository [2].

Four tasks of classification into two classes for which data from previous tests were known were selected: "German", "Heart", "Adult", and "Ionosphere".

The task “German” decides whether a client is good or bad to be lent money to. In this data errors are weighted so that not to lend money to good a client means error weight 1, and lending money to a bad client means error weight 5.

The task “Heart” indicates the absence or presence of a heart disease in a patient.

The task “Adult” determines whether a person earns over $ 50000 a year.

For the task “Ionosphere” the targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not show this; their signals pass through the ionosphere.

We do not describe these tasks in detail here as all can be found in [2]. For each task the same approach to testing and evaluation was used as described in [2]. Especially splitting the data set into two disjoint subsets, the learning set and the testing set, and the use of cross validation were the same as in [2]. For our method the discriminant threshold was tuned accordingly.

The testing should show the classification ability of IINC method for some tasks and also show its classification ability relatively to other published methods and results for the same data sets.

In Table 1 the results are shown together with the results of other methods as given in [2]. For each task the methods were sorted according to the classification error, the method with the best – the smallest - error first.

Table 1. Comparison of the classification error of IINC method for different tasks with results of other classifiers as given in [2].

| **“German”** | | **“Heart”** | |
|---|---|---|---|
| Algorithm | Error | Algorithm | Error |
| **IINC** | 0.1580 | **IINC** | 0.1519 |
| SVM | 0.297 | Bayes | 0.374 |
| Discrim | 0.535 | Discrim | 0.393 |
| LogDisc | 0.538 | LogDisc | 0.396 |
| Castle | 0.583 | Alloc80 | 0.407 |
| Alloc80 | 0.584 | SVM | 0.411 |
| Dipol92 | 0.599 | QuaDisc | 0.422 |
| Smart | 0.601 | Castle | 0.441 |
| Cal | 0.603 | Cal5 | 0.444 |
| Cart | 0.613 | Cart | 0.452 |
| QuaDisc | 0.619 | Cascade | 0.467 |
| KNN | 0.694 | KNN | 0.478 |
| Default | 0.700 | Smart | 0.478 |
| Bayes | 0.703 | Dipol92 | 0.507 |
| IndCart | 0.761 | Itrule | 0.515 |
| Back Prop | 0.772 | Bay Tree | 0.526 |
| BayTree | 0.778 | Default | 0.560 |
| Cn2 | 0.856 | BackProp | 0.574 |
| **“Adult”** | | **“Ionosphere”** | |
| Algorithm | Error | Algorithm | Error |
| FSS Naive Bayes | 0.1405 | IB3 | 0.0330 |
| NBTree | 0.1410 | **IINC** | 0.0331 |
| C4.5-auto | 0.1446 | backprop | 0.0400 |

| | | | |
|---|---|---|---|
| IDTM (Decision table) | 0.1446 | Ross Quinlan's C4 | 0.0600 |
| HOODG | 0.1482 | nearest neighbor | 0.0790 |
| C4.5 rules | 0.1494 | "non-linear" perceptron | 0.0800 |
| OC1 | 0.1504 | "linear" perceptron | 0.0930 |
| C4.5 | 0.1554 | SVM | 0.1400 |
| Voted ID3 (0.6) | 0.1564 | | |
| SVM | 0.1590 | | |
| CN2 | 0.1600 | | |
| Naïve-Bayes | 0.1612 | | |
| **IINC** | 0.1617 | | |
| Voted ID3 (0.8) | 0.1647 | | |
| T2 | 0.1684 | | |
| 1R | 0.1954 | | |
| Nearest-neighbor (4) | 0.2035 | | |
| Nearest-neighbor (2) | 0.2142 | | |

***Tasks from UCI Machine Learning Repository – Comprehensive Tests***

Data sets ready for a run with a classifier were prepared by Paredes and Vidal and are available on the net [7]. We used all data sets in this corpus. Each task consists of 50 pairs of training and testing sets corresponding to 50-fold cross validation. For DNA data [8], Letter data (Letter recognition [2]), and Satimage (Statlog Landsat Satellite [2]) the single partition into training and testing sets according to specification in [2] was used. We also added the popular Iris data set [2] with ten-fold cross validation.

In Table 3 the results obtained by different methods are summarized. The methods are as follows:

| | |
|---|---|
| L2 | The nearest neighbor method, data by [5] |
| 1-NN L2 | The nearest neighbor method computed by authors |
| sqrt-NN L2 | The k-NN method witt k equal to square root of the number of samples of the learning set computed by authors |
| Bayes 10 | The Bayes naive method with ten bins histograms, computed by authors |
| CDM | The learning weighted metrics method with class dependent Mahalanobis, data by [5] |
| CW | The learning weighted metrics method with class dependent weighting by [5], data by [5] |
| PW | The learning weighted metrics method with prototype dependent weighting by [5], data by [5] |
| CPW | The learning weighted metrics method with class and prototype - dependent weighting by [5], data by [5] |
| posit. L1 | The learning weighted metrics method [6] with positions weighting and Manhattan L1 metrics |
| posit. L2 | The learning weighted metrics method [6] with positions weighting and Euclidean L2 metrics |
| diff. L1 | The learning weighted metrics method [6] with coordinate differences weighting and Manhattan L1 metrics |
| diff. L2 | The learning weighted metrics method [6] with coordinate differences weighting and Euclidean L2 metrics |
| **IINC L1** | The method presented here with Manhattan L1 metrics |

**IINC L2** The method presented here with Euclidean L2 metrics

In Table 2 in each row the best result is denoted by bold numerals. Furthermore, in the last column, the values for IINC better with L2 metrics than with L1 metrics are shown in italics. There are 6 such cases out of a total of 24.

Table 2. Classification error rates for different datasets and different approaches. Empty cells denote not available data. For legend see text above.

| \Method | L2 | 1-NN L2 | sqrt-NN | Bayes | SVM | CDM | CW | PW | CPW | posit. L1 | posit. L2 | diff. L1 | diff. L2 | **IINC L1** | **IINC L2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Australian | 34.37 | 20.73 | 15.50 | 13.88 | 35.99 | 18.19 | 17.37 | 16.95 | 16.83 | 17.64 | 19.00 | 17.86 | 21.51 | **13.31** | 14.75 |
| Balance | 25.26 | 23.61 | 32.06 | 15.17 | 45.48 | 35.15 | 17.98 | 13.44 | 17.6 | 17.85 | **16.17** | 34.48 | 37.74 | 32.58 | *30.80* |
| Cancer | 4.75 | 5.07 | 3.25 | 2.68 | 16..34 | 8.76 | 3.69 | 3.32 | 3.53 | 17.70 | **3.18** | 26.46 | 26.49 | 3.28 | 3.48 |
| Diabetes | 32.25 | 29.48 | 26.46 | **24.19** | 29.64 | 32.47 | 30.23 | 27.39 | 27.33 | 34.90 | 26.49 | 34.90 | 34.90 | 26.21 | *25.52* |
| Dna | 23.4 | 25.72 | 34.06 | 6.66 | | 15 | 4.72 | 6.49 | **4.21** | 20.83 | 24.37 | 42.24 | 41.57 | 27.82 | 31.03 |
| German | 33.85 | 32.76 | 30.90 | 24.97 | 27.25 | 32.15 | 27.99 | 28.32 | **27.29** | 29.02 | 29.23 | 29.87 | 30.00 | 30.91 | 31.13 |
| Glass | **27.23** | 32.72 | 42.10 | 47.37 | | 32.9 | 28.52 | 26.28 | 27.48 | 43.43 | 30.29 | 46.89 | 43.77 | 33.01 | 35.18 |
| Heart | 42.18 | 25.11 | 16.89 | 17.44 | 38.89 | 22.55 | 22.34 | 18.94 | 19.82 | 19.04 | 21.56 | 21.37 | 22.52 | 17.96 | ***17.93*** |
| Ionosphere | *19.03* | 14.05 | 14.70 | **9.26** | | | | | | 29.39 | 17.58 | 29.70 | 30.03 | 10.82 | 14.81 |
| Iris | *6.91* | 5.91 | 7.91 | 9.82 | 6.55 | | | | | **4.91** | 6.91 | 25.82 | 11.82 | 7.91 | ***4.91*** |
| Led17 | *20.5* | 11.50 | **0.12** | 0.00 | | | | | | 7.64 | 2.67 | 24.78 | 37.72 | 0.46 | *0.45* |
| Letter | 4.35 | 4.80 | 18.70 | 28.98 | 40.53 | 6.3 | 3.15 | 4.6 | **4.2** | 6.23 | 5.90 | 7.95 | 8.05 | 4.85 | 4.98 |
| Liver | 37.7 | 39.59 | 41.48 | 39.42 | 37.68 | 39.32 | 40.22 | **36.22** | 36.95 | 40.96 | 42.00 | 40.70 | 40.43 | 38.29 | 39.13 |
| Monkey1 | *2.01* | 2.01 | 9.27 | 28.01 | 23.54 | | | | | 2.01 | 2.82 | **1.45** | 1.47 | 4.79 | 4.79 |
| Phoneme | *18.01* | 11.83 | 20.71 | 21.47 | 21.71 | | | | | 14.72 | **14.61** | 29.27 | 29.27 | 17.55 | 18.06 |
| Satimage | 10.6 | 10.65 | 15.20 | 19.15 | 44.85 | 14.7 | 11.7 | **8.8** | 9.05 | 11.40 | 11.70 | 76.95 | 75.90 | 11.00 | 11.55 |
| Segmen | *11.81* | 3.81 | 11.41 | 9.85 | | | | | | 5.18 | 5.35 | 9.96 | 10.62 | **4.12** | 5.05 |
| Sonar | *31.4* | 18.37 | 32.51 | 31.46 | | | | | | 21.11 | 21.89 | 46.63 | 46.63 | **19.89** | 22.85 |
| Vehicle | 35.52 | 30.51 | 31.51 | 38.40 | | 32.11 | 29.38 | **29.31** | 28.09 | 30.48 | 31.01 | 36.83 | 34.96 | 29.40 | *29.34* |
| Vote | 8.79 | 8.74 | 9.60 | 9.70 | | 6.97 | 6.61 | 5.51 | **5.26** | 7.97 | 7.45 | 7.17 | 11.98 | 8.52 | 8.89 |
| Vowel | 1.52 | **1.19** | 46.68 | 26.64 | | 1.67 | 1.36 | 1.68 | 1.24 | 3.52 | 3.89 | 5.55 | 6.17 | 2.73 | 2.74 |
| Waveform 21 | 24.1 | 23.73 | **14.71** | 19.26 | | | | | | 18.50 | 18.63 | 25.56 | 25.19 | 16.15 | 16.38 |
| Waveform 40 | 31.66 | 28.22 | **16.24** | 20.31 | | | | | | 20.50 | 22.61 | 32.25 | 32.78 | 17.59 | 18.08 |
| Wine | 24.14 | 5.42 | 6.15 | 4.50 | | 2.6 | 1.44 | 1.35 | **1.24** | 5.27 | 6.06 | 72.04 | 67.42 | 4.24 | 5.66 |

***Standalone Serious Real-live Comprehensive Classification Task***

This data set was available for tests described in [3] as one of many studies for data processing relating ATLAS experiment at CERN, Geneva, Switzerland. For the description of the particle physics problem we cite [3] in Table 3 as follows:

Table 3. Problem formulation from the point of view of physics.

> Identification of hadronic $\tau$ decays will be the key to the possible Higgs boson discovery in the wide range of the MSSM parameter space [1]. The $h/H/A \to \tau\tau$ and $H^{\pm} \to \tau\nu$ are promising channels in the mass range spanning from roughly 100 GeV to 800 GeV. The sensitivity increases with large $\tan\beta$ and decreases with rising mass of the Higgs boson. The $H \to \tau\tau$ decays will give access to the Standard Model and light Minimal Supersymmetric Standard Model Higgs boson observability around $m_H = 120$ GeV, with Higgs boson produced by vector-boson fusion [2]. The hadronic $\tau$ identification is also very important in searching for supersymmetric particles, particularly at high $\tan\beta$ values [3].
>
> ...
>
> The same signal and background samples, as discussed in [4], are used to evaluate performance of the proposed methods. As signal, we consider reconstructed candidates from tau decays in $pp \to W \to \tau\nu$ and $pp \to Z \to \tau\tau$ events. As background, we consider candidates from QCD shower in the same $pp \to W \to \tau\nu, pp \to Z \to \tau\tau$ events and in QCD dijet events (sample with $p_T^{hard} > 35$ GeV).
>
> (Note that references relate to [3].)

The data set consists of 7 dimensional vectors of real numbers and class mark, which differentiates between signal samples (events) and background samples. The data set is split into a learning and a testing set, each of 3279 samples.

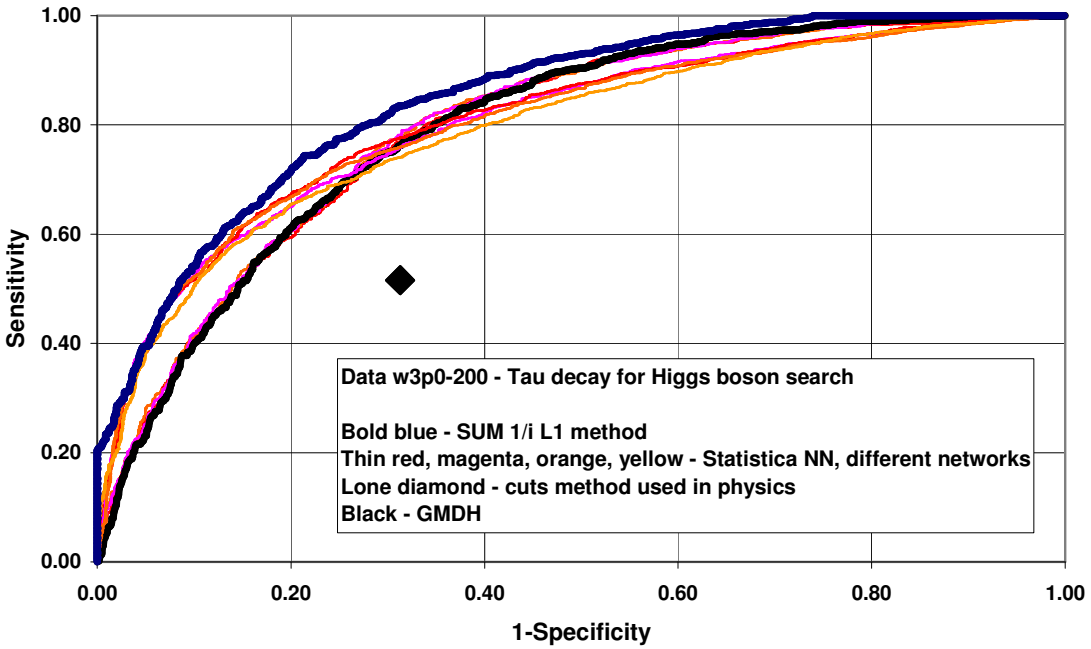

Fig. 1. ROC curves for different separation/classification tools including the “cut” method.

In Fig. 1 well-known ROC curves are shown for different separation/classification tools including the “cut” method popular in physics studies.

The result obtained with “cuts” method is depicted by the black diamond.

The result obtained by GMDH-MIA algorithm is depicted by the lower bold black line.

The results obtained by Statistica Neural Networks are depicted by two sets of red, magenta, orange and yellow lines. Each set corresponds to four best results out of ten networks generated. The set going more to the left at level 0.4 or 0.6 of signal acceptance corresponds to its being set as a classifier; the other set (more close to the black line of GMDH-MIA) corresponds to its being set as an approximator.

The upper bold blue line was obtained by the IINC method described in this report with L1 metrics.

**Conlusion**

The IINC seems to provide better classification then other classifiers in most tasks even though it is not the best all the time. This could make it a welcome alternative to standard classification methods.

The idea of the classifier above is subject to patent pending under number PV 2008-245; Z 7576 submitted on 22nd April 2008 to the INDUSTRIAL PROPERTY OFFICE, Antonína Čermáka 2a, 160 68 Prague, Czech Republic.

**Acknowledgements**

This work was supported by the Ministry of Education of the Czech Republic under the project Center of Applied Cybernetics No. 1M0567, and No. MSM6840770012 Transdisciplinary Research in the Field of Biomedical Engineering II.methods.

**References**

[1] Jiřina M. and Jiřina M., Jr., Classifier Based on Inverted Indexes of Neighbors II – Theory and Appendix, Technical Report, Institute of Computer Science AS CR, No. V-1041, November 2008.

[2] A. Asuncion, D.J. Newman, (2007). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

[3] F. Hakl, M. Jirina, E. Richter-Was. Hadronic tau's identification using artificial neural network. ATLAS Physics Communication, ATL-COM-PHYS-2005-044, CERN, Geneve, Switzerland, last revision: 26 August 2005, 12pp.

[4] T.M. Cover, P.E. Hart: Nearest neighbor Pattern Classification. IEEE Transactions in Information Theory, Vol. IT-13, No. 1, january 1967, pp. 23-27.

[5] R. Paredes, E. Vidal, Learning Weighted Metrics to Minimize Nearest Neighbor Classification Error. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 7, July 2006, pp. 1100-1110.

[6] M. Jiřina, M. Jiřina, jr. Learning Weighted Metrics Method with a Nonsmooth Learning Process. Technical report V-1026, Institute of Computer Science AS CR, 2008, 15pp.

[7] S. M. Lucas, Algoval: Algorithm Evaluation over the Web, [online], 2008, [cited November 23, 2008]. Available: <http://algoval.essex.ac.uk/data/vector/UCI/>

[8] R. Paredes: CPW: Class and Prototype Weights learning, [online], 2008, [cited November 23, 2008]. Available: <http://www.dsic.upv.es/~rparedes/research/CPW/index.html>