Improving Performance of a Binary Classifier by Training Set Selection

Cezary Dendek[1] and Jacek Mańdziuk[2]

[1] Warsaw University of Technology, Faculty of Mathematics and Information Science,
Plac Politechniki 1, 00-661 Warsaw, POLAND,
dendekc@student.mini.pw.edu.pl,
[2] Warsaw University of Technology, Faculty of Mathematics and Information Science,
Plac Politechniki 1, 00-661 Warsaw, POLAND,
(phone: (48 22) 621 93 12; fax: (48 22) 625 74 60)
mandziuk@mini.pw.edu.pl,
WWW home page: http://www.mini.pw.edu.pl/~mandziuk/

Abstract. In the paper a method of training set selection, in case of low data availability, is proposed and experimentally evaluated with the use of *k-NN* and neural classifiers. Application of proposed approach visibly improves the results compared to the case of training without postulated enhancements.
Moreover, a new measure of distance between events in the pattern space is proposed and tested with *k-NN* model. Numerical results are very promising and outperform the reference literature results of *k-NN* classifiers built with other distance measures.

1 Introduction

The problem of binary classification based on small corpus of data, which implies the use of simple classifier models, is quite often encountered in several application fields. In order to improve classifier's accuracy in such cases the training set should be selected with special care due to significant influence of each element on the classification quality (caused by a small size of the data set).

In this paper a new approach to the problem of training set selection is proposed in Sect. 2 and experimentally evaluated in the context of supervised training with feed-forward neural networks and *k-NN* models. The idea relies on the iterative training set creation based on the test of the hypothesis that elements of the training set fulfill the model assumptions and can be properly classified by chosen (target) classifier. Presented algorithm provides a generalization of approach described in [1] in the context of iterative reevaluation of typicality of each observation.

Additionally, a new measure of distance between elements of the pattern space is proposed in Sect. 3 and evaluated with *k-NN* model. Proposed distance function expresses the empirical probability that two vectors are representations of the same event.

The benchmark data sets and results of numerical evaluation of proposed training set construction method and distance measure efficacy are presented in Sections 4 and 5, resp. Conclusions and directions for future research are placed in the last section.

Presented measure of distance is a continuation of authors' previous works [2, 3] related to properties of metrical structure of pattern space.

2 Outlier extraction in the case of binary classification

2.1 Introduction

The usual approach during the task of fitting a model belonging to the class of models to given set of observations is to select a model that performs best on this set.

This approach however can be misleading in case of modeling phenomena of complex structure when only a small data set is available, which naturally limits the possible degree of complexity of the model. In such case some non-typical observations from data set can hinder the quality of a fit by their influence during the training phase (these observations should be regarded as outliers in the context of selected model and the whole data set).

Hence, in order to improve the performance of the chosen model one may consider to eliminate these non-typical, misleading observations from the training set.

2.2 Algorithm of non-typical observation extraction in case of *k-NN* classifiers

Having a distance measure defined on pattern space it is possible to extract patterns that are non-typical for the considered models. In the case of binary classification the following algorithm can be proposed utilizing the *k-NN* approach:

1. Let TS be a set of all observations.
2. $\forall x \in TS$ find k closest observations from the remaining set $TS \setminus \{x\}$. The distance is calculated only based on the parts of patterns that do not belong to dimensions being predicted, since these (predicted) dimensions will be unknown during the test phase. Hence, a part of a pattern belonging to predicted dimensions is set to 0.
3. Among those k observations, a fraction of them belonging to the same class as x defines a *confidence coefficient* of x.
4. If k is reasonably large and a confidence coefficient of x is close to 0, then x should be considered as non-typical observation for the *k-NN* classifier.

In described algorithm given element x is conditionally classified under the condition of its neighborhood (defined by parameter k and distance function). The whole process can be considered as *k-NN* leave-one-out crossvalidation (denoted by *1-CV*). Generally speaking, typicality of observation is a function of data set and the considered classification algorithm. These observations lead to generalization of the algorithm described above to other binary classifier classes.

2.3 Algorithm of non-typical observation extraction in general case of binary classifiers

1. Let TS be a set of all observations.
2. Perform *1-CV* using given class of models m times (independence of each particular model with respect to the training set is important as well as m being reasonably large; m cannot however be higher than number of models in class being evaluated). Let $C_{i,TS\setminus\{x\}}(x)$ be the binary correctness of classification of element x in step i, $i \in \{1, \ldots, m\}$.

3. A fraction of proper classifications of element x defines a *confidence coefficient* ($CC_{CV(TS)}(x)$) of given pattern (under the condition of model used and the data set):
$$CC_{CV(TS)}(x) = \frac{\sum_{i=1}^{m} C_{i,TS\setminus\{x\}}(x)}{m}, \tag{1}$$
where $CC_{CV(TS)}(x)$ is calculated using *1-CV* on the TS set.
4. $\forall x$ if m is reasonably large and $CC_{CV(TS)}(x)$ is smaller than predefined threshold α the observation is considered as a non-typical one, being a member of non-typical observation set (NTS) defined as follows:
$$NTS = \{x \in TS : CC_{CV(TS)}(x) < \alpha\}, \tag{2}$$
$$TS := TS \setminus NTS. \tag{3}$$

In case of classifiers with time-consuming training phase, application of presented approach can be hindered by overall calculation time. In such a case, instead of application of *1-CV*, a whole data set can be used as both training and testing set. In most cases the sets generated by both methods should be highly positively correlated.

Non-typicality of the elements of the set created with the use of the algorithm described by equations (1)-(3) depends on the other elements of the data set (including other non-typical observations), which may cause improper categorization of some observations as non-typical.

In order to reduce this effect re-categorization of non-typical observations (NTS set) with respect to typical ones (TS set) is proposed. The procedure can be expressed by the following algorithm.

2.4 Algorithm of outliers extraction in general case of binary classifiers

1. $\forall x \in NTS$ calculate its confidence coefficient ($CC_{TS}(x)$) by building m classifiers with use of TS as training set. Observe that in this case the calculation is performed *without applying the* 1-CV *procedure*.
2. Move elements of $CC_{TS}(x) > \beta$, $\beta \geq 0$ from NTS to TS, for a given, pre-defined threshold β.
$$TTS = \{x \in NTS : CC_{TS}(x) > \beta\}, \tag{4}$$
$$NTS := NTS \setminus TTS, \tag{5}$$
$$TS := TS \cup TT. \tag{6}$$
3. If any elements have been moved (i.e. $|TTS| > 0$), goto step 1. Otherwise, finish the procedure. NTS is the set containing outliers.

2.5 Remarks

Algorithm presented in sections 2.3 and 2.4 can be regarded as *forward* ($TS \rightarrow NTS$) and *backward* ($NTS \rightarrow TS$) steps of outlier selection procedure. Those steps can be repeated iteratively until stabilization of NTS set.

The confidence coefficient of elements belonging to NTS set, calculated during the phase of outlier extraction can be used to estimate the efficiency of cooperative voting classifier as well as a number of classifiers used in cooperative model.

3 Distance in the training patterns space

Pattern space has naturally defined structure of metrical space which is obtained by its immersion into $\mathbb{R}^n$. This approach however does not preserve structure of probability space, which can be used to improve accuracy of estimators.

Improved immersion can be obtained with the use of *Cumulative Density Functions* (CDF) by transformation of pattern space. Let CDF_i denotes univariate CDF calculated on i-th dimension of pattern space. Transformation of pattern is defined as follows:

$$(CDF(x))_i := CDF_i(x_i)$$

CDF transformation results in uniform distribution of patterns in each dimension. In order to provide an effective measure of distance between two events correlation has to be taken into consideration, by applying *Mahalanobis-derived distance*. Let Σ denotes covariance matrix of $CDF_i(x_i)$. Distance between events A and B is defined as:

$$d_{cdf}(A,B) := \sqrt{\Big(CDF\,(A) - CDF(B)\Big)^T \Sigma^{-1} \Big(CDF(A) - CDF(B)\Big)}$$

where $CDF(A)$ and $CDF(B)$ are representations of these events according to CDF. A classical Mahalanobis distance defined as a measure of distance between a vector and a set of all observations can be obtained as follows:

$$D_{cdf}(A) := \sqrt{\Big(CDF(A) - \frac{1}{2})\Big)^T \Sigma^{-1} \Big(CDF(A) - \frac{1}{2}\Big)}$$

due to uniform distribution of CDF_i values in the set $[0,1]$, for each i.

4 Data sets

In order to provide experimental support of presented method validity and generate results that can be compared to other sources BUPA Liver Disorders and Pima Indians Diabetes data sets available at *UCI Machine Learning Repository* [4] has been chosen. Both of them contain observations of rather complex medical phenomena of catastrophic nature and provide only correlated predictors. Size of the sample is limited to 345 and 768 elements, respectively. Both sets are rather challenging tasks for classification methods.

In the following experiments either the "raw" data sets were used or the transformed ones obtained by application of *Empirical Cumulative Distribution Function* (ECDF) which resulted in normalization of data and uniformity of marginal distributions.

5 Results

Both data sets has been evaluated with use of a neural network and *k-NN* classifier. Application of outlier removal algorithm described in Sect. 2 resulted in significant increase of classifier quality in both cases.

5.1 k-NN classifier

During evaluation *k-NN* models with standard distance function and distance defined in Sect. 3 has been used. Discrimination levels has been set as follows: $\alpha = 1$, $\beta = 0.5$.

Observe that in case of classifiers using the ECDF transformation, the metric function was modified by repeating the calculation of the matrix Σ based *exclusively on elements of the set* $TS \setminus NTS$. The re-calculation of Σ after outliers removal from TS enhances classification capabilities. This property is owed to the fact that after removing a non-empty set NTS from TS the correlation matrix has changed and the "old" correlation matrix calculated for TS does not fit the current model for the set $TS \setminus NTS$.

Numerical results of classifiers quality (misclassification rates in per cent) for *1-CV* estimation are presented in Tables 1, 2 and 3. Table 1 presents misclassification results in the case of standard Euclidean metric (i.e. without ECDF transformation) in the "raw" *1-CV* case and with the use of outlier removal algorithm described in Sect. 2.4. Table 2 provides a similar comparison in the case of ECDF transformation and standard distance. The results of the combination of the outlier removal algorithm with ECDF transformation with (column 3) and without (column 2) recalculation of Mahalanobis-based distance metric (Σ) vs standard *1-CV* (column 1) in ECDF space are presented in Table 3.

Table 1. *k-NN* misclassification rate, standard distance in Euclidean space.

	BUPA Liver Disorders		**Pima Indians Diabetes**	
k	**initial *1-CV***	***1-CV* after outlier removal**	**initial *1-CV***	***1-CV* after outlier removal**
3	36.52	33.62	30.60	26.30
5	33.62	32.75	28.52	27.99

Table 2. *k-NN* misclassification rate, standard distance in ECDF image space.

	BUPA Liver Disorders		**Pima Indians Diabetes**	
k	**initial *1-CV***	***1-CV* after outlier removal**	**initial *1-CV***	***1-CV* after outlier removal**
3	35.07	30.14	27.73	23.96
5	31.88	29.28	26.56	25.26

The results presented in the Tables fully confirm the efficacy of the proposed outliers removal algorithm. Moreover, the advantage of using ECDF transformation is also clear. Finally, the results support application of the Mahalanobis-derived metric - in all cases

Table 3. *k-NN* misclassification rate, Mahalanobis-derived distance in ECDF image space

	BUPA Liver Disorders			Pima Indians Diabetes		
k	**initial *1-CV***	***1-CV* after outlier removal**	***1-CV* after outlier Σ removal**	**initial *1-CV***	***1-CV* after outlier removal**	***1-CV* after outlier Σ removal**
3	33.91	28.12	26.96	27.60	24.09	24.09
5	31.59	30.43	28.70	26.56	25.00	24.87

Table 4. k-NN misclassification rate comparison

distance maeasure	**estimation method**	**BUPA Liver Disorders**	**Pima Indians Diabetes**
Mahalanobis-based with outlier removal (Table 3)	leave-one-out CV	26.96	24.09
Mahalanobis-based (Table 3)	leave-one-out CV	31.59	26.56
weighted distances [5]	100 x 5–CV	36.22	27.39
adaptive distance measure [6]	10–CV	32.94	28.16
boosting distance estimation [7]	100 x $20/80$	33.58	28.91
cam weighted distance [8]	leave-one-out CV	35.3	24.7

except for *3-NN* and Pima Indians Diabetes set, presented in Table 3 the use of this metric improved the results compared to the cases of standard metric in the ECDF space (Table 2) and Cartesian metric on the untransformed data (Table 1). The advantage is especially visible in the case of Liver Disorders data.

In summary, the use of the Mahalonobis-based distance metric combined with the algorithm of outliers removal allowed, within the *k-NN* models, to achieve comparable or better results than the best ones presented in the literature. In particular the following comparisons with other definitions of the metric functions within the considered data sets has been taken into account: weighted distances [5], adaptive distance measure [6], boosting distance estimation [7] and cam weighted distance [8]. Table 4 presents the summary of results accomplished by the *k-NN* classifiers.

5.2 Neural classifier

Both data sets has been evaluated with use of a neural network classifier (MLP with one hidden layer), using various sizes of hidden layer and backpropagation as a training method.

Classifier quality has been estimated with use of leave-one-out crossvalidation performed 10 times (the results presented in teh paper are the average values over these 10 trials). The initial estimation of the network's classification quality concerned the full

data sets. In the final estimation, the observations classified as outliers were removed from the training set and appeared only in the test set.

In case of Pima Indians Diabetes data set initial non-typical observation set has been obtained with the use of training and testing performed on full data set, due to high computational cost of full crossvalidation and in order to provide an empirical support for a claim presented in Sect. 2.5.

For both data sets values of discriminant coefficients has been set to $\alpha = 0.5$, $\beta = 0.5$. Discrimination has been made with a support of $m = 20$ independent (with respect to the training set) classifiers of a given architecture.

The results of *1-CV* estimation of the neural classifiers with and without application of the outliers removal algorithm are presented in Table 5. Similarly to the case of *k-NN* classifiers the use of the removal of non-typical observations from the training set clearly enhances the efficacy of neural classifiers. In effect, the MLP-based classifiers outperform the *k-NN* ones further decreasing the misclassification figures to the levels of 24.46% and 21.35%, respectively for Liver Disorders and Pima Indians Diabetes data sets.

Based on the comprehensive comparison of various classification methods presented in [9] for the case of Pima Indians Diabetes set the results accomplished by the algorithm using ECDF transformation of the data combined with outliers removal method presented in this paper can be regarded as very promising.

Table 5. Neural network misclassification rate.

	BUPA Liver Disorders		**Pima Indians Diabetes**	
number of hidden units	**initial 1-CV**	**1-CV after outlier removal**	**initial 1-CV**	**1-CV after outlier removal**
2	28.79	26.32	24.84	22.27
3	29.37	24.54	24.77	21.59
4	28.60	25.39	24.85	21.43
5	29.08	24.46	24.22	21.35

6 Conclusions

The problem of classifier construction based on small corpus of data is quite common in real-life situations. In case of fitting a general model (constructed without additional knowledge about modelled phenomena) training set selection can improve overall classifier's efficacy.

In the paper a method of training set selection is proposed and experimentally evaluated with *k-NN* and neural network classifiers. It is shown that proposed approach produces in average better results than training without its use in case of two sample problems of small corpus of data.

Additionally, a new measure of distance between events in the pattern space is proposed and evaluated with *k-NN* model. Crossvalidational estimate of resulting model quality has been compared with numerical results provided by other researchers, concerning the *k-NN* classifiers built with other distance measures.

Tests in other problem domains are under research. Other possible uses of presented distance measure (especially in the context of training sequence construction presented in [2]) are considered as future research plans.

Acknowledgement

This work was supported by the Warsaw University of Technology grant number 503G 1120 0007 007.

References

1. Sane, S.S., Ghatol, A.A.: Use of instance typicality for efficient detection of outliers with neural network classifiers. In Mohanty, S.P., Sahoo, A., eds.: ICIT, IEEE Computer Society (2006) 225–228
2. Dendek, C., Mandziuk, J.: Including metric space topology in neural networks training by ordering patterns. In Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E., eds.: ICANN (2). Volume 4132 of Lecture Notes in Computer Science., Springer (2006) 644–653
3. Mańdziuk, J., Shastri, L.: Incremental class learning approach and its application to handwritten digit recognition. Inf. Sci. Inf. Comput. Sci. **141**(3-4) (2002) 193–217
4. A. Asuncion, D.N.: UCI machine learning repository (2007)
5. Pardes, R., Vidal, E.: Learning weighted metrics to minimize nearest-neighbor classification error. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28**(7) (2006) 1100–1110
6. Wang, J., Neskovic, P., Cooper, L.N.: Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recogn. Lett. **28**(2) (2007) 207–213
7. Amores, J., Sebe, N., Radeva, P.: Boosting the distance estimation. Pattern Recogn. Lett. **27**(3) (2006) 201–209
8. Zhou, C.Y., Chen, Y.Q.: Improving nearest neighbor classification with cam weighted distance. Pattern Recogn. **39**(4) (2006) 635–645
9. Tsakonas, A.: A comparison of classification accuracy of four genetic programming-evolved intelligent structures. Inf. Sci. **176**(6) (2006) 691–724