Ala'raj *et al. J Big Data* (2020) 7:15
https://doi.org/10.1186/s40537-020-00297-7

Journal of Big Data

RESEARCH Open Access

Improving binary classification using filtering based on *k*-NN proximity graphs

Maher Ala'raj[1*], Munir Majdalawieh[1] and Maysam F. Abbod[2]

*Correspondence:
maher.alaraj@zu.ac.ae
[1] College of Technological Innovation, Zayed University, Dubai 19282, UAE
Full list of author information is available at the end of the article

Abstract

One of the ways of increasing recognition ability in classification problem is removing outlier entries as well as redundant and unnecessary features from training set. Filtering and feature selection can have large impact on classifier accuracy and area under the curve (AUC), as noisy data can confuse classifier and lead it to catch wrong patterns in training data. The common approach in data filtering is using proximity graphs. However, the problem of the optimal filtering parameters selection is still insufficiently researched. In this paper filtering procedure based on k-nearest neighbours proximity graph was used. Filtering parameters selection was adopted as the solution of outlier minimization problem: *k*-NN proximity graph, power of distance and threshold parameters are selected in order to minimize outlier percentage in training data. Then performance of six commonly used classifiers (Logistic Regression, Naïve Bayes, Neural Network, Random Forest, Support Vector Machine and Decision Tree) and one heterogeneous classifiers combiner (DES-LA) are compared with and without filtering. Dynamic ensemble selection (DES) systems work by estimating the level of competence of each classifier from a pool of classifiers. Only the most competent ones are selected to classify a given test sample. This is achieved by defining a criterion to measure the level of competence of base classifiers, such as, its accuracy in local regions of the feature space around the query instance. In our case the combiner is based on the local accuracy of single classifiers and its output is a linear combination of single classifiers ranking. As results of filtering, accuracy of DES-LA combiner shows big increase for low-accuracy datasets. But filtering doesn't have sufficient impact on DES-LA performance while working with high-accuracy datasets. The results are discussed, and classifiers, which performance was highly affected by pre-processing filtering step, are defined. The main contribution of the paper is introducing modifications to the DES-LA combiner, as well as comparative analysis of filtering impact on the classifiers of various type. Testing the filtering algorithm on real case dataset (Taiwan default credit card dataset) confirmed the efficiency of automatic filtering approach.

Keywords: Binary classification, Heterogeneous combiner, k-NN, Proximity graphs, Data filtering, Feature selection

Introduction

Background

In different classifying problems, performance of classification model may vary a lot depending on its structure. Relatively simple classifiers like Decision Tree, Naïve Bayes

and Logistic Regression may have drastically lower accuracy than more complex classifiers like SVM or Neural Network, especially on data with big number of features [1]. Thus, building heterogeneous combiners using classifiers with different performance may be useless: combiner will have better performance than simple classifiers and worse than complex classifiers. The reason of it is that classifiers with lower performance have the same impact on the final result as more reliable. It can lead to deterioration of classification accuracy. We decide to solve the problem by setting combiner output as the linear combination of all its components with weights proportional to local accuracy of these classifiers. One of the ways to increase performance of simple classifiers is to change training data in the order to decrease noise and redundant features. We use filtering technique based on k nearest neighbours (k-NN) graphs (a node is connected to its k nearest neighbours) with automatic parameter evaluation, unified for all classifiers.

Research motivations

During various classification problems of datasets with large amount of features a big difference in performance of classifiers can be observed, which can lead to not sufficient performance of classifier combiners. The main reason of this difference lies in noise in training data, which confuse some simple classifiers like Decision Tree, Naïve Bayes or Logistic Regression. That is why it is important to develop efficient filtering method, which is classifier-independent and improve performance of simple classifier.

In this paper a new and efficient classifier-independent filtering method is proposed based on proximity graphs in combination with feature selection, which improve performance of simple classifiers. This method was applied on six binary class datasets and its result clearly show advantage of using filtering and feature selection as training data processing step for some classifiers.

Literature review

Data filtering

Data filtering is used for improving the results of classifiers machine learning [15], which should be trained on some training data before applying to the testing set. It upgrades the training set by removing the inaccurate samples, which stand out against the whole range. For example, a loan in the sample data which is labelled as bad one among many good loans, at the same time with similar characteristics, needs to be removed from the training set.

The motivation behind applying a data filtering algorithm in this paper lies in the belief that training a classifier with the filtered dataset can have several benefits [1, 9] such as:

- The decision boundaries are smooth and clear;
- It is easier for the classifiers to discriminate between the classes;
- Decreasing the size of the training; leaving in it the really important data;
- Improving the accuracy performance of the model;
- Computational costs can be reduced.

The obvious drawbacks of filtering that may decrease classifier performance are [9].

- Making the training data less expressive
- And decreasing training set size.

Related studies

In the research of Smith et al. [25], the potential benefits of instance filtering and hyper-parameter optimization (HPO) were estimated. While both HPO and filtering significantly improve the quality of the induced model, filtering has a greater potential effect on the quality of the induced model than HPO, motivating future work in it.

Filtering procedures is closely related to clustering. The most used technique of clustering is KNN algorithm. However, classical KNN algorithm is not adaptive, as KNN parameters are needed to be chosen manually. In the research of Shi et al. [23], a new clustering method is proposed. Firstly, the k nearest neighbors of all samples is calculated, and then a density method based on kNN is used to complete the clustering process. Clustering based on the density peak method was also researched in Chen et al. [5].

A fast exact nearest neighbour search algorithm overlarge scale data is proposed based on semi-convex hull tree, where each node represents a semi-convex hull, made of a set of hyper planes. When performing the task of nearest neighbour queries, unnecessary distance computations can be greatly reduced by quadratic programming [6].

Machine learning algorithms are of vital importance to many medical problems, they can help to diagnose a disease, to detect its causes, to predict the outcome of a treatment, etc. K-Nearest Neighbors algorithm (KNN) is one of the simplest algorithms and is widely used in predictive analysis. To optimize its performance and to accelerate its process a new solution to speed up KNN algorithm based on clustering and attributes filtering was proposed if the research of Cherif [7]. It also includes another improvement based on reliability coefficients which insures a more accurate classification. Results of the proposed approach exceeded most known classification techniques with an average f-measure exceeding 94% on the considered breast-cancer dataset.

The traditional KNN method has some shortcomings such as large amount of sample computation and strong dependence on the sample library capacity. A method of representative sample optimization based on CURE algorithm is proposed in the research of Chen [4]. On the basis of this, presenting a quick algorithm QKNN (Quick k-nearest neighbor) to find the nearest k neighbor samples, which greatly reduces the similarity calculation. The experimental results show that this algorithm can effectively reduce the number of samples and speed up the search for the k nearest neighbor samples to improve the performance of the algorithm.

Researches related to impact of filtering on the performance of various classifiers were conducted in the papers of Brodley and Friedl [3], Frénay and Verleysen [8], Guyon and Elisseeff [11], Gieseke et al. [10], Saez et al. [22]. The classification problems were studied in Ko et al. [14], Peterson et al. [21], Vriesmann et al. [27], Woods et al. [28] and Xiao and He [30].

In the paper of Netti and Radhika et al. [19], the authors present a novel method to minimize the loss of accuracy in Naïve Bayes Classifier due to the assumption of Independence among predictors. The experimental results show that the proposed method performed well and improved the accuracy when compared to the traditional Naïve

Bayes Classifier. In the paper of Mansourifar and Shi [17], a novel type of perceptron called L-Perceptron was proposed. The work of Malladi Tejasvi et al. [26] has focused on detecting specific aspects of banknotes using methods of Machine Learning.

Dynamic ensemble selection (DES) is the problem of finding, given an input x, a subset of models among the ensemble that achieves the best possible prediction accuracy. DES method based on Probabilistic Classifier Chains was proposed by Anil [18]. Experimental results on 20 benchmark data sets show the effectiveness of the proposed method against competitive alternatives, including the aforementioned multi-label approaches.

Most DES differ from each other only on the selection scheme. Zhu et al. [31] propose Dynamic Ensemble Selection with Local Expertise Consistency (DES-LEC) that focus on generating a learners pool dedicated to the latter selection phase. Experiment results on 4 medical data sets suggest that DES-LEC is able to improve the performance over the DES systems that select from a regular learners pool.

A new oracle based Dynamic Ensemble Selection (DES) method in which an Ensemble of Classifiers (EoC) is selected to predict the class of a given test instance (xt) was described by Pereira et al. [20]. The competence of each classifier is estimated on a local region (LR) of the feature space represented by the most promising k-nearest neighbors (or advisors) related to xt according to a discrimination index (D) originally proposed in the Item and Test Analysis (ITA) theory. A robust experimental protocol based on 30 classification problems and 20 replications have shown that the proposed DES compares favorably with 15 state-of-the-art dynamic selection methods and the combination of all classifiers in the pool.

Methods

All the experiments for this study were performed using MATLAB 2016b version, on a PC with 3.4 GHz, Intel CORE i5 and 8 GB RAM, using the Microsoft Windows 7 operating system.

Datasets

A collection of binary classification datasets from UCI depository was employed in the process of empirical model evaluation. All datasets are different in number of entries, features and percentage of positive entries. It allows to test the algorithm on different cases and see the difference in classification accuracy for each one.

- German credit (22 features, 1000 entries, 70% of positive entries) will be denoted as dataset A. The dataset was created by Professor Dr. Hans Hofmann from Institute of Statistics of Hamburg University. In this dataset bank credit attributes for 1000 credits is provided.
- Data banknote authentication (4 features, 1372 entries, 56% of positive entries). Will be denoted as dataset B. The dataset was provided by Helene Darksen (University of Applied Sciences, Ostwestfalen-Lippe. Data was extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400×400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained.

- Haberman (3 features, 306 entries, 74% of positive entries). Will be denoted as dataset C [12]. The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.
- Ionosphere (34 features, 351 entries, 64% of positive labels). Will be denoted as dataset D. [24]. The dataset represent classification of radar returns from the ionosphere, which were collected by a system in Goose Bay, Labrador. 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; their signals pass through the ionosphere.
- Seismic bumps (18 features, 2584 entries, 93% of positive labels). Will be denoted as dataset E. The dataset was provided by Marek Sikora and Lukasz Wrobel from Institute of Computer Science, Silesian University of Technology. The data describe the problem of high energy (higher than 10^4 J) seismic bumps forecasting in a coal mine. Data come from two of longwalls located in a Polish coal mine.
- WDBC (30 features, 569 entries, 63% of positive labels). Will be denoted as dataset F. This dataset was provided by Dr. William H. Wolberg, General Surgery Dept. University of Wisconsin, Clinical Sciences Center, Madison.

Data analysis and selecting filtering parameters

To detect outliers, for each point we compute weighted average of its neighbour labels, with weights proportional to power of distance:

$$P(i) = \sum_{j \in N(i)} l(j) * w_{ij} \tag{1}$$

where

$$w_{ij} = \frac{d(i,j)^p}{\sum_{k \in N(i)} d(i,k)^p}. \tag{2}$$

The notation $d(i,j)$ above means Euclidean distance between point i and point j, $l(j)$ is label of point j, $N(i)$ is neighbour of point i according to selected proximity graph. Then we mark point i as outlier if one of the conditions are true:

- $P(i) < threshold$ and $l(i) = 1$.
- $P(i) \geq threshold$ and $l(i) = 0$.

Parameters p, threshold, type and parameters of proximity graph to minimize the proportion of outliers (outlier rate) for each dataset. We calculate the outlier rate by dividing the number of outliers by the total number of entries. After comparing several types of proximity graphs K-NN proximity graph was chosen according to the fact that it is fast to build and gives lower outlier rate. K-NN proximity graph is basically a graph that connect each vertex with K other vertices that has the smallest Euclidean distance to it.

Let us illustrate the filtering algorithm. We have target point with label 0 and three points from the training set that are closest neighbours of it. Distances are 1, 5 and 7 respectively. Suppose that power p is equal to -0.35 and threshold is equal to 0.5.

After applying Eq. (2) to the set we get the following weights for training points: 0.48, 0.27, 0.24. Therefore, for the target point $P = 0.48 \cdot 1 + 0.27 \cdot 0 + 0.27 \cdot 0 = 0.75$. As P is greater than threshold, we can mark target point as an outlier.

As it can be seen from Table 1, German credit and Haberman datasets have 23% and 24% of outliers respectively. This explains why it is so hard to improve accuracy on these datasets more than 77% and its increase of even a few per cent is a decent result. Further accuracy increase for these datasets is possible, but it requires tremendous efforts. By the contrary, level of outliers for other datasets are much lower, which leads to higher accuracy even using single and simple classifiers.

In order to calculate the accuracy for each method me use fivefold cross-validation method 10 times. Therefore, the final accuracy of each single classifier and DES-LA combiner is an average number of this parameter for all 50 testing sets.

Base classifiers development

In this paper, six classifiers and one combiner are used, these methods as well as being well known are easy to implement. Below classifiers are listed along with their parameters used.

- *Decision Tree* (number of estimations to split the leaf -10, empirically evaluated probabilities for each class) Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed. This process is continued on the training set until meeting a termination condition. The tree is constructed in a top-down recursive divide-and-conquer manner. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers. An over-fitted model has a very poor performance on the unseen data even though it gives an impressive performance on training data. This can be avoided by pre-pruning which halts tree construction early or post-pruning which removes branches from the fully-grown tree.
- *Logistic Regression classifier (*nominal type of model). Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. It computes

Table 1 Filtering parameters for each dataset

	German	Banknote authn.	Haberman	Ionosphere	Seismic bumps	WDBC
Neighbour count	25	20	20	8	10	20
Power	−0.35	0	0	−4.8	−0.8	−2.1
Threshold	0.48	0.75	0.45	0.11	0.62	0.49
Outlier rate	0.23	0	0.24	0.07	0.06	0.02

the probability of an event occurrence. It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.

Properties of logistic regression:

- The dependent variable in logistic regression follows Bernoulli distribution.
- Estimation is done through maximum likelihood.
- *Naïve Bayes* (normal distribution for each feature). Naive Bayes is a probabilistic classifier inspired by the Bayes theorem under an assumption which is that the attributes are conditionally independent. The classification is conducted by deriving the maximum posterior probability with the above assumption applying to Bayes theorem. This assumption greatly reduces the computational cost by only counting the class distribution. Even though the assumption is not valid in most cases since the attributes are statistically dependent, Naive Bayes has able to perform quite well. Naive Bayes can suffer from a problem called the zero-probability problem. When the conditional probability is zero for a particular attribute, it fails to give a valid prediction. This needs to be fixed explicitly using a Laplacian estimator.
- *Support Vector Machine* (Radial basis kernel function, kernel scale automatically evaluated for each dataset except German (for which kernel scale is 1.8). The objective of the support vector machine algorithm is to find a hyperplane in transformed feature space that distinctly separates the data points of different classes. Our objective is to find a hyperplane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane.
- *Neural Network* (10 hidden layers, method of gradient descent learning with adaptive learning rate, transfer function for the first layer is hyperbolic tangent, for the second layer is linear). Artificial Neural Network is a set of connected input/output artificial neurons where each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input vectors. The disadvantage of the classifier is the poor interpretability of model compared to other models like Decision Trees due to the unknown symbolic meaning behind the learned weights. However, Neural Networks have performed impressively in most of the real-world applications. It has high tolerance to noisy data and able to classify untrained patterns. Usually, Artificial Neural Networks perform better with numerical inputs and outputs.
- *Random Forest* (number of trees—60, method–classification). Random forest by itself is a homogeneous combiner which consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes the prediction

of Random Forest. Uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this effect is that the trees cancel out its individual mistakes.

- *DES-LA combiner.*
  Data for training is the same for all classifiers, no matter whether filtering is used or not.

DES-LA combiner

The idea of DES-LA (Dynamic ensemble selection based on local accuracy) combiner is similar to DCS-LA (dynamic classifier selection based on local accuracy) [29]. In original DCS-LA algorithm for each test point, we select one classifier with the best local accuracy, while in our case we assign output of combiner as a linear combination of single classifiers with weights proportional to local accuracy of these classifiers (according to the point 2.6 of the algorithm).

1. Input: training and testing data, training actual labels, training predicted labels for all classifiers, testing predicted labels for all classifiers, K—number of neighbours to consider at each testing point, P—power of distance. Value of K and P are chosen the same as in filtering algorithm. If we take $K = 1$, then the algorithm will lose its generalizing ability (the ability to produce the correct result for data not previously encountered in the algorithm) as a new record will get a class from the closest one. If you set a too big value, then many local features will not be revealed.
2. *For all test samples do*
   2.1. Find k nearest neighbours (K-NN) from training set to the test sample (by the Euclidean distance between samples as the vectors of numerical features),
   2.2. Get corresponding distances $D(KNN)$ from these data entries to the test sample and actual labels $L(KNN)$ of these entries.
   2.3. For each classifier compute difference between its ranking and $L(KNN)$, which is the error of each classifier at each neighbour entry
   2.4. Compute weighted average error of each classifier with weights proportional to distances $D(KNN)$ in the power $P < 0$, so we have a vector of six numbers, name it M. The smaller error is, the more we can trust the classifier at testing sample.
   2.5. Proceed with equation $W_i = \max(M) - M_i \, \forall i \in \overline{1..k}$, which converts vector of errors so classifier with the greatest error will have zero value of W_i, and classifier with the smallest error will have maximal value of W_i.
   2.6. Normalize vector W or: $W_i = \frac{M_i}{\sum M_i} \, \forall i \in \overline{1..k}$
   2.7. Calculate weighted average of single classifier predictions at current testing entry with weights W, assign this value as ranking of DES-LA combiner at this entry.
3. *End for*

Performance indicator measures

It is worth pointing out that from the perspective of risk management, the scores of the methods are more valuable than the binary result of classification—credible or not

credible clients. In order to reach a reliable and robust conclusion of the predictive accuracy of the proposed approach, five performance indicator measures are implemented, specifically: (1) accuracy, (2) sensitivity, (3) specificity, (4) area under the curve (AUC), and (5) Brier Score. These performance indicators were chosen because they are popular in binary classification and they give a comprehensive view on all aspects of model performance. Accuracy is the percentage of the correctly classified entries with respect to all samples. But it does not say anything about the classification performances for negative and positive classes separately. As such, this is a criterion that estimates the discriminating ability of the model [16]. Sensitivity is the proportion of entries who have the target condition (reference standard positive) and give positive test results. Specificity is the proportion of entries without the target condition and give negative test results. Sensitivity and specificity measures are especially valuable for unbalanced datasets, where proportion of positive and negative labels is different (German dataset or Seismic Bumps).

The area under the curve (AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative example. It measures the classifiers skill in ranking a set of patterns according to the degree to which they belong to the positive class, but without actually assigning patterns to classes. Higher the AUC, better the model is at correctly predicting positive and negative entries. According to [13], the AUC can be used to estimate the model's performance without any prior information about the error costs. Finally, the Brier Score, which is also known as the mean squared error [2], measures the calibration of the probability predictions of the classifier. The Brier score can be thought of as either a measure of the "calibration" of a set of probabilistic predictions, or as a "cost function". More precisely, across all items $i \in \{1 \ldots N\}$ in a set of N predictions, the Brier score measures the mean squared difference between the predicted probability assigned to the possible outcomes for each item and the actual value of outcome.

The lower the Brier Score the better the classifier performance.

Experimental results and discussion

Impact of filtering on single classifiers performance

Filtering drastically increases accuracy for Naïve Bayes, Decision Tree and Neural Network classifiers, while for Logistic Regression classifier, SVM and Random Forest filtering impact is close to zero. In addition, DES-LA shows 1.6% increase after applying filtering because this combiner works better if performance of its constituents is similar.

In DES-LA classifier, we use both filtered and unfiltered random forest predictions, because of the fact, that filtering does not always increase accuracy of random forest.

From the Table 2 it is seen that the best classifiers are SVM and DES-LA with the highest values of classification accuracy almost for all datasets. Naïve Bayes and decision tree classifier are the worst. An explanation for it could be that Naïve Bayes usually have not really high accuracy while working with statistically dependent features. Moreover, a decision tree can be easily over-fitted, which can lead to accuracy decrease.

In the Table 3 it is seen that the biggest increase shows decision tree for German and Haberman datasets. At the first sight it seems that filtering doesn't have a big impact on

Table 2 Classifier accuracy without filtering

Classifier	German	Banknote authn.	Haberman	Ionosphere	Seismic bumps	WDBC
DT	0.692	0.981	0.686	0.879	0.898	0.919
LR	0.76	0.99	0.742	0.868	0.931	0.954
NB	0.724	0.841	0.747	0.821	0.855	0.934
SVM	0.761	0.999	0.716	0.937	0.933	0.974
NN	0.741	0.978	0.731	0.871	0.933	0.955
RF	0.762	0.993	0.686	0.932	0.91	0.961
DES-LA	0.742	0.997	0.737	0.934	0.927	0.964

Table 3 Classifier accuracy with filtering

Classifier	German	Banknote authn.	Haberman	Ionosphere	Seismic bumps	WDBC
DT	0.746	0.981	0.747	0.894	0.933	0.931
LR	0.76	0.99	0.744	0.878	0.934	0.961
NB	0.757	0.841	0.752	0.814	0.927	0.931
SVM	0.761	0.999	0.736	0.929	0.934	0.969
NN	0.747	0.979	0.742	0.863	0.934	0.948
RF	0.743	0.992	0.748	0.922	0.934	0.948
DES-LA	0.768	0.996	0.747	0.928	0.934	0.963

Table 4 Difference between performance measures with and without filtering over all datasets

Classifier	DT (%)	LR (%)	NB (%)	SVM (%)	NN (%)	RF (%)	DES-LA (%)
ACC	2.97	0.38	1.65	0.11	0.07	0.35	0.57
Sens	6.14	0.85	3.07	0.26	1.41	2.59	1.12
Spec	7.12	0.74	7.92	− 0.15	− 2.98	5.42	− 1.08
AUC	− 1.53	4.53	3.76	− 1.54	− 2.56	4.23	0.17
BS	− 1.32	1.61	0.05	0.11	0.81	1.00	− 0.13

classification accuracy, but as we have in most cases skewed datasets, even increase of a few per cent is also very good result.

As we observe from the Table 4, for almost all datasets, Decision Tree and Naïve Bayes shows big increase in accuracy, so after filtering they show almost as good results as more complicated classifiers like SVM and Random Forest. When to talk about AUC, the best impact filtering has on Logistic Regression classifier, Naïve Bayes and Random Forest, with increase almost in 5% in every case. When for other classifiers, except DES-LA, the value of AUC even decreases. The reason of decreasing AUC for some classifiers lies in negative consequences of data filtering (reducing data variability).

It is worth pointing out that filtering doesn't have really big impact on performance of complex classifiers like Neural Network, Random Forest or SVM. The reason of it is that they are robust against noise in the training data. Therefore, filtering doesn't increase their accuracy very much. But the performance of simple classifiers shows big increase in accuracy after filtering pre-processing step.

Table 5 Decision tree filtering impact analysis

Dataset	Accuracy gap comparing to best classifier (without filtering) (%)	Accuracy gap comparing to best classifier (with filtering) (%)	Accuracy increase (%)
German	7	2.2	5.4
Banknote authn.	1.8	1.8	0
Haberman	6.1	0.5	6.2
Iono-sphere	5.8	3.5	1.5
Seismic bumps	3.5	0.1	3.5
WDBC	5.5	3.8	1.2

Decision tree is considered as not the best classification method. However, filtering procedure made this classifier perform almost as good as the best classifiers such as SVM, RF and DES-LA. According to Table 5 the first and second columns tell difference in accuracy of Decision Tree and the accuracy of the best classifier for each dataset. The third column shows us increase of accuracy for DT. Therefore, Decision tree is the classifier which is the most affected (in a positive way) from filtering preprocessing. In comparison, Naïve Bayes performance increased in German dataset (+3.2%) and in seismic bumps dataset (+7.2%), for other datasets, it remains almost the same.

As results of filtering, accuracy of DES-LA combiner shows big increase with low-accuracy datasets (2.6% increase for German dataset and 1% increase for Haberman dataset). For high-accuracy dataset DES-LA performance remains almost the same with and without filtering.

Comparing with benchmarks:

For *seismic bumps* dataset, benchmark is 93.1% ± 0.6% for Random Forest, while in this investigation result of Random Forest is 93.4%, which lies in this range. Naïve Bayes Classifier algorithm proposed in Netti et al. [19] has an accuracy of 81.1% while our method shows 92.7% accuracy with filtering option enabled.

For *Haberman* dataset, benchmark accuracy value is 75.18%, according to the study of Mansourifar et al. [17]. Filtering pre-processing step allows Naïve Bayes Classifier reach a 75.2% of accuracy, which is comparable to the results, obtained in the mentioned study.

For *data banknote authentication* dataset SVM gives 99.9% of accuracy, which is better than benchmark 98% of accuracy. In the work of Malladi Tejasvi et al. [26] the logistic regression and decision tree detected fake notes with the accuracy of 99.27% and 98.91% respectively. In the current paper these classifiers achieve 99% and 99.8% accuracy respectively.

For *WDBC* dataset benchmark is 90.26% (CED method), which is also surpassed by this investigation (97.4% for SVM).

For *ionosphere* dataset, benchmark is 88.64%, while 93.4% for DES-LA combiner without filtering.

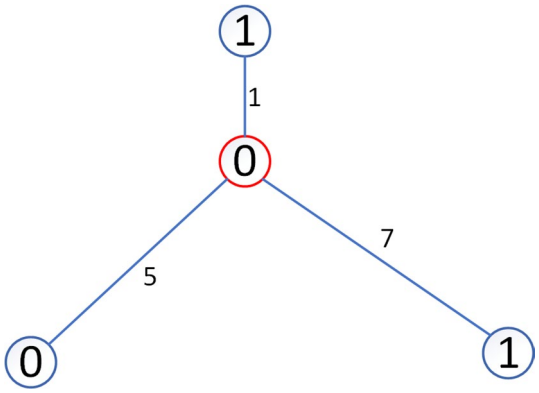

Fig. 1 Illustration of filtering algorithm in 2-dimentional space

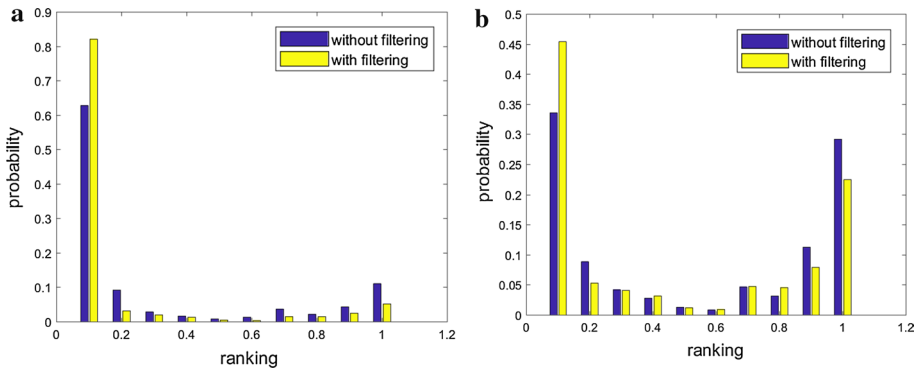

Fig. 2 Actual ranking distribution for testing labels with label 0 and 1 respectively (Decision Tree classifier, German dataset)

Ranking distribution change due to filtering and feature selection

Figure 1 shows the bar plot of ranking distribution for testing instances with actual label 0 and actual label 1 respectively. It is found out that for Decision Tree classifier that was trained with pre-filtered data tends to be sharper in its decisions (decrease number of rankings near 0.5). Also filtering increase skewness of results (increase sensitivity and decrease specificity). This is logical, as filtering procedure more often removes training entries from non-dominant class.

In the Fig. 2 the bar plots of ranking distribution for testing instances with actual label 0 and actual label 1 respectively are displayed. We found out that filtering pre-processing stage makes Decision Tree more confident in its decisions (decrease number of rankings near 0.5). Also filtering increase skewness of results (increase sensitivity and decrease specificity). This is logical, as filtering procedure tends to remove more training entries from non-dominant class.

Figure 3 shows that Naïve Bayes classifier results and explanations are similar: increase sharpness of and skewness of results. It means that filtering cause more categorical decisions of Naïve Bayes classifier. This fact leads to increase of AUC and Brier score for this classifier.

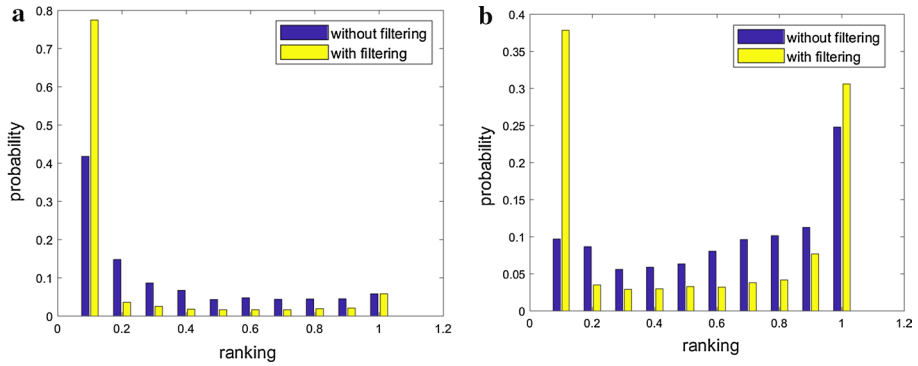

Fig. 3 Actual ranking distribution for testing labels with label 0 and 1 respectively (Naïve Bayes classifier, German dataset)

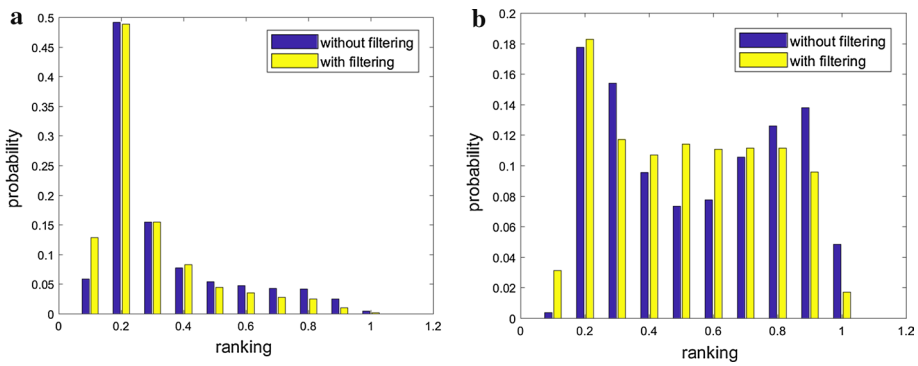

Fig. 4 Actual ranking distribution for testing labels with label 0 and 1 respectively (DES-LA combiner, German dataset)

According to the Fig. 4, DES-LA combiner results are slightly different (increase sharpness of decisions in case of dominant label, and become less certain in other case) In this case filtering also increase skewness of results: Sensitivity increase from 86 to 92% while specificity drops from 46 to 41%.

Real case test: large defaults payments dataset

The dataset reflects costumer's default payments in Taiwan. We will compare the predictive accuracy of the probability of default among previously introduced data mining methods. The size of the data set is 30000, which is large enough to test the efficiency of filtering using KNN approach. Number of non-default payments is 23364, while number of default payments is 6636 (proportion of default payments in dataset is 22%). In the dataset the following 23 variables are used as explanatory ones:

- X1: Amount of the given credit, which includes both the individual consumer credit and his/her family (supplementary) credit
- X2: Gender (1 = male; 2 = female)
- X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others)
- X4: Marital status (1 = married; 2 = single; 3 = others)
- X5: Age (year)

- X6–X11: History of past payment. We denoted tracked payment records from September to April, 2005 by X6–X11 respectively. The measurement scale for the repayment status is: −1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.
- X12–X17: Amount of bill statement. We denoted amount of bill statement from September to April, 2005 by X12–X17 respectively.
- X18–X23: Amount of previous payment. We denoted amount paid from September to April, 2005 by X18–X23 respectively.

We can divide the variables into 2 groups: numerical and categorical. The examples of the first are: X1(amount of given credits), X5(age), X6-X11(history of past payment), etc. The second group contains such variables: X2(gender), X3(education), X4(marital status).

For this dataset filtering procedure based on k-nearest neighbours proximity graph was used, however the method of finding nearest neighbours for this dataset is based on KD-tree algorithm. The choice of a KD-tree method seems to be the best adapted to our task. The advantage of using it is the fact that it allows finding k-nearest neighbours with O(k*log(n)) complexity. A k-d tree is a data structure that partitions space by repeatedly choosing a point in the data set to split the current partition into halves. It accomplishes this by alternating the dimension, which is called a cutting dimension for the level and performs the split (refer to Fig. 5).

We will denote by Q the closest point and by X that point for which we are looking for a neighbour. In the beginning, Q and minimum distance (between X and Q) are null and infinite respectively. The process of searching the closest point starts at the root and should be followed by the certain rules:

1. *Input:* set of points, root.
2. *For root do:*

   2.1. If we reach a null node-return.
   2.2. If the boundary box of the present root has no point closer than the minimum distance then return, meaning skip traversing that sub-tree at all.

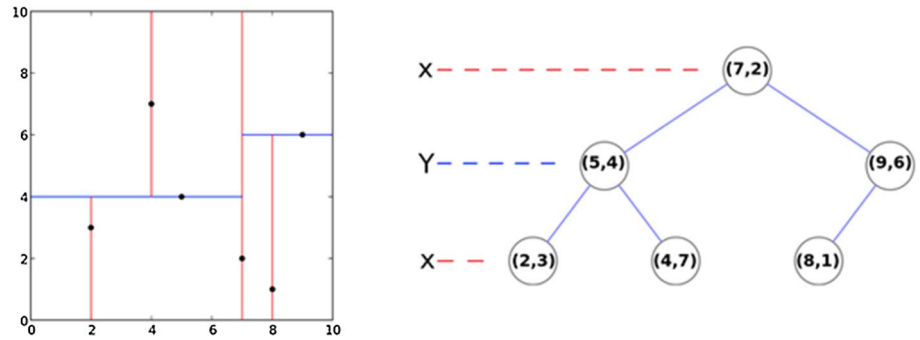

Fig. 5 Illustration of kd-tree algorithm for finding nearest neighbour in 2-dimentional space

Table 6 Classifier performance without filtering on large default payments dataset

Classifier	Decision Tree	Logistic regression	Naive Bayes	SVM	Neural Network	Random Forest	DES-LA
Accuracy	0.657	0.737	0.774	0.816	0.793	0.817	0.816
Sensitivity	0.781	0.926	0.837	0.956	0.985	0.943	0.947
Specificity	0.217	0.073	0.551	0.323	0.118	0.371	0.354
AUC	0.5	0.499	0.736	0.703	0.686	0.762	0.749
Brier Score	0.302	0.196	0.184	0.152	0.155	0.139	0.141

Table 7 Classifier performance with filtering on large default payments dataset

Classifier	Decision Tree	Logistic regression	Naive Bayes	SVM	Neural Network	Random Forest	DES-LA
Accuracy	0.813	0.818	0.79	0.815	0.794	0.817	0.835
Sensitivity	0.951	0.958	0.875	0.956	0.985	0.954	0.95
Specificity	0.327	0.324	0.491	0.322	0.124	0.334	0.365
AUC	0.688	0.713	0.712	0.714	0.683	0.729	0.769
Brier Score	0.179	0.162	0.192	0.15	0.17	0.161	0.159

2.3. If the present root is closer to X than the minimum distance, we update Q and the minimum distance.

2.4. To choose the next sub-tree to be explored we will compare the cutting dimension value of X to that of the root. If its dimension value is smaller than we traverse left first, right second. We want to emphasize that we are calling both of them but at a certain order with the hope that the first traversed sub-tree would give a closer point than any point the other one could offer. So next time when we would traverse the other sub-tree we can do a quick check and completely skip traversing it.

3. *End for*

To get k neighbours you should repeat the process k times ignoring previously selected points. In general, we need a k-d tree when we have higher dimensional data points. But when the dimension is too high other approaches might work better. When to talk about our task, this decision helps us to improve the performance of the whole algorithm by decreasing time needed for the k-nearest neighbours selection.

During the generation of proximity graph for each testing set data point, such parameters were automatically selected:

(1) Distance power: − 1.05.
(2) Threshold: 0.522.
(3) Outlier percentage: 19.75%

Distance power is negative because the larger the distance is, the smaller impact has the neighbour to each point. After selecting the neighbours for each testing set point, we apply all single classifiers and DES-LA combiner to them.

As we can see from Table 6 the results table, simple classifiers like Decision Tree, Logistic regression and Naïve Bayes performs much worse than more complex classifiers like Random Forest. The best classifier is Random Forest, that has the highest accuracy and AUC, and lowest Brier Score.

By looking at Table 7 filtering procedure hugely improves accuracy of Decision Tree, Logistic regression and Naïve Bayes, and has almost no impact on Random Forest. However, with all classifiers now have more similar performance, DES-LA combiner now manages to overcome all of single classifiers by 1.8%.

Conclusion

The main idea of filtering is to find such parameters for proximity graph, as well as threshold and distance power parameters, which will minimize the ratio of outliers. We apply filtering with the same parameters for all classifiers and observe the decrease of accuracy difference between stronger classifiers like SVM and simple classifiers like Naïve Bayes or Decision Tree. This fact makes possible to construct combiners, which overcome the best of the single classifiers results. We demonstrate this fact by implementing DES-LA combiner and compare its results with and without filtering. Testing the approach on real case dataset (Taiwan default credit card dataset) confirmed the efficiency of automatic filtering approach.

One of a few disadvantages of filtering procedure is that it increases skewness of results: for datasets with dominant positive class filtering increase sensitivity and decrease specificity, for datasets with dominant negative class everything is vice versa. The main contributions of the study are:

(1) Steady accuracy increases for simple classifiers, regardless of the dataset.

Development a novel single classifier combiner based on local accuracy. This classifier takes into account results of all six classifiers with corresponding weights that are proportional to the local accuracy of each classifier.

(2) Parameter selection for filtering is fully automatic and does not require manual adjustment.

In the future, we plan to extend filtering method and DES-LA combiner to multi-class classification problems and do verification on UCI multi-class datasets. It is also planned to apply filtering method on more classical classifiers and combiners to find the other ones which are affected by filtering at most.

Abbreviations

AUC: Area under the curve; k-NN: k nearest neighbour; SVM: Support vector machines; DES-LA: Dynamic ensemble selection based on local accuracy; LR: Linear regression; NB: Naïve Bayes; RF: Random Forests; NN: Neural Networks; DT: Decision Tree; BS: Brier Score; OLA: Overall local accuracy; DCS: Dynamic classifier selection; DES: Dynamic ensemble selection.

Acknowledgements

None.

Authors' contributions
MA designed and carried out experiments and data analysis and drafted the manuscript. MM and MA participated in research coordination and checked, read and approved the final manuscript. All authors contributed in revising the manuscript. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
All datasets used for supporting the conclusions of this paper are available from the public data repository at the website of http://archive.ics.uci.edu/ml/index.php.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests

Author details
[1] College of Technological Innovation, Zayed University, Dubai 19282, UAE. [2] Department of Electrical and Computer Engineering, Brunel University London, Uxbridge UB8 3PH, UK.

Received: 26 November 2019 Accepted: 21 February 2020
Published online: 05 March 2020

References

1. Ala'raj M, Abbod MF. A new hybrid ensemble credit scoring model based on classifiers consensus system approach. Expert Syst Appl. 2016;104:36–55.
2. Brier GW. Verification of forecasts expressed in terms of probability. Mon Whether Rev. 1950;78(1):1–3.
3. Brodley CE, Friedl MA. Identifying mislabeled training data. J Artif Intell Res. 1999;11(1):131–67.
4. Chen S. (2017). K-nearest neighbor algorithm optimization in text categorization. IOP Conference Series: Earth and Environmental Science.
5. Chen Y, Hu X, Fan W, Shen L, Zhang Z, Liu X, Li H. Fast density peak clustering for large scale data based on kNN. Knowledge-Based Syst. 2020;187:104824.
6. Chen Y, Zhou L, Bouguila N, Zhong B, Wu F, Lei Z, Du J, Li H (2018). Semi convex hull tree: fast nearest neighbor queries for large scale data on GPUs. IEEE International Conference on Data Mining, ICDM, IEEE, p. 911–916.
7. Cherif W. Optimization of K-NN algorithm by clustering and reliability coefficients: application to breast-cancer diagnosis. The First International Conference On Intelligent Computing in Data Sciences, Procedia Computer Science. 2018;127(2018):293–9.
8. Frénay B, Verleysen M. Classification in the presence of label noise: a survey. IEEE Transactions on Neural Networks and Learning Systems. 2014;25(5):845–69.
9. Garcia V, Marqués A, Sánchez JS. On the use of data filtering techniques for credit risk prediction with instance-based models. Expert Syst Appl. 2012;39:13267–76.
10. Gieseke F, Heinermann J, Oancea CE, Igel C. Buffer kd trees: processing massive nearest neighbor queries on GPUs. ICML. 2014;2014:172–80.
11. Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res. 2003;3:1157–82.
12. Haberman, S. J. (1976). Generalized Residuals for Log-Linear Models, Proceedings of the 9th International Biometrics Conference, Boston, p. 104–122.
13. Hand DJ. Measuring classifier performance: a coherent alternative to the area under the ROC curve. Mach Learn. 2009;77:103–23.
14. Ko A, Sabourin R, Britto A Jr. From dynamic classifier selection to dynamic ensemble selection. Pattern Recognit. 2008;41(5):1718–31.
15. Kubica J, Moore A. (2003). Probabilistic noise identification and data cleaning. In: Proceedings of the third IEEE International Conference on Data Mining, pages 131–138, 2003.
16. Lessmann S, Baesens B, Seow H, Thomas LC. Benchmarking state-of-the-art classification algorithms for credit scoring: an update of re- search. Eur J Oper Res. 2015;247:124–36.
17. Mansourifar H, Shi W (2018) Toward efficient breast cancer diagnosis and survival prediction using L-perceptron. arXiv preprint arXiv:1811.03016.
18. Narassiguin A., Elghaze H, Alex Aussem A (2017). Dynamic ensemble selection with probabilistic classifier chains. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2017: Machine Learning and Knowledge Discovery in Databases, p. 169–186.
19. Netti K, Radhika Y. Minimizing loss of accuracy for seismic hazard prediction using Naive Bayes Classifier. IRJET. 2016;3(4):75–7.
20. Pereira M., Britto A., Oliveira L., Sabourin R. (2018). Dynamic ensemble selection by K-nearest local Oracles with Discrimination Index. 2018 IEEE 30th International conference on tools with artificial intelligence (ICTAI), volume: 1, p. 765–771.

21. Peterson A. H. and Martinez T. R. (2005). Estimating the potential for combining learning models. In: Proceedings of the ICML workshop on meta-learning, p. 68–75
22. Saez JA, Luengo J, Herrera F. Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. Pattern Recognit. 2013;46(1):355–64.
23. Shi Bing, Han Lixin, Yan Hong. Adaptive clustering algorithm based on kNN and density. Pattern Recognit Lett. 2018;104:37–44.
24. Sigillito VG, Wing SP, Hutton LV, Baker KB. Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Tech Dig. 1989;10:262–6.
25. Smith MR, Martinez T, Giraud-Carrier C. (2015) The Potential benefits of data set filtering and learning algorithm hyperparameter optimization. MetaSel'15 In: Proceedings of the 2015 international conference on meta-learning and algorithm selection, volume 1455, p. 3–14.
26. Tejasvi Malladi, A. Nayeemulla Khan, A.Shahina (2019). Perfecting counterfeit banknote Detection-a classification Strategy. International Journal of Innovative Technology and Exploring Engineering (IJITEE), p. 434–440.
27. Vriesmann LM, Britto AS, Luiz SO, Koerich AL, Sabourin R (2015). Combining overall and local class accuracies in an oracle-based method for dynamic ensemble selection. 2015 International Joint Conference on Neural Networks (IJCNN).
28. Woods K, Kegelmeyer WP, Bowyer K. Combination of multiple classifiers using local accuracy estimates. IEEE Trans Pattern Anal Mach Intell. 1997;19(4):405–10.
29. Xiao J, Xie L, He Changzheng, Xiaoyi J. Dynamic classifier ensemble model for customer classification with imbalanced class distribution. Expert Syst Appl. 2012;39(2012):3668–75.
30. Xiao J, He CZ. Dynamic classifier ensemble selection based on GMDH. Proceeding of the second international joint conference on computational sciences and optimization. Washington: IEEE; 2009. p. 731–4.
31. Zhu Y, Zhang Y, Pan Y (2015). Dynamic ensemble selection with local expertise consistency. 2015 IEEE Conference on computational intelligence in bioinformatics and computational biology (CIBCB).

Publisher's Note