Ciência e Natura
ISSN: 0100-8307
cienciaenaturarevista@gmail.com
Universidade Federal de Santa Maria
Brasil

Ciência e Natura, Santa Maria, v. 37 Part 2 2015, p. 76–86

ISSN impressa: 0100-8307 ISSN on-line: 2179-460X

Data Mining and Machine Learning: an Overview of Classifiers

Mehri Haghighi

Department of Computer Engineering, Payam Noor University, Sosangerd, Iran

Abstract

At the same time of information age, digital revolution has made necessary using some of technologies to analyze most of essential information. Data mining is a technique to make sense to the available data. The aim of data mining is extracting the information from a vast volume of data and transforming them into a comprehensible form for human. For this purpose, machine learning methods are used to classify data. In this study, we discuss six popular and useful classifiers in the data mining process.

Keywords: *Machine Learning, Classification, Support Vector Machine, decision Tree,K-nearst Neibora, bagging, Boosting.*

Recebido: dia/mês/ano Aceito: dia/mês/ano

1 Introduction

At the same time of information age, digital revolution has made necessary using some of technologies to analyze most of essential information. Data mining, is a technique to make sense to the available data. This technology has been became popular in the recent years more than before. Data mining is a set of techniques that allow us to move beyond the ordinary data processing and help extracting the information that are hidden in a vast volume of data. The data extraction process utilizes analytical tools to determine the relationships between data in large data base. Data mining include the method of machine learning, statistical technique and data base system. Data mining aims to extract information from a vast volume of data and transform them into a human comprehensible form (Salari and Adibnia, 1389, Sharma et al., 2013).

Classification is one of data mining (machine learning) technique that maps the data to predicted groups (classes). This technique provides intelligence decision making and it is used not only for studying and investigating the current instance, but also predicting the future behavior of same instance. Classification includes two phases: first, in the training phase the dataset is analyzed and in the second phase, the data is tested and the accuracy of the classification algorithm is achieved (Sharma et al., 2013).

In this paper, we discuss different and useful classification methods. For this purpose, regarding their execution, these methods are divided into two groups, independent classifiers and Ensemble classification (Ensemble learning). Moreover, in addition to introducing classification methods, this paper also evaluates their advantages and disadvantages in comparison to other methods.

2 Independed Classifiers

Independent classifiers are the same as ordinary classifiers that perform data separation using a specific mechanism. Among these classifiers are support vector machines (Vapnik, 1998), k-nearest neighbors (Fix and Hodges, 1951), decision trees (Breiman and Friedman, 1984), and random forest (Breiman, 2001) that in following, we explain the details of each classifier. We must note that these classifiers can be used as base learner in Ensemble Classifiers (part 3).

2.1 Support Vector Machine (SVM)

A linear separator shows as equation:

$$f(x): R^d \rightarrow R \qquad (1)$$

When there are two classes. If f(x) ≥0, the data attributed to the positive class otherwise, data belong to negative class.

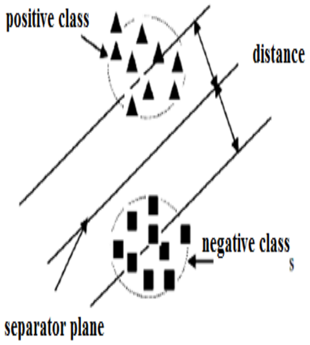
For set of points (x_i, y_i), that x_i is input data, y_i data class label and $y_i \in \{-1,1\}$, If a linear separator can be found to establish equation 2 for all i, this data then can be separated as linear:

$$y_i f(x_i) \geq 0 \qquad (2)$$

Support vector machine have been developed on base of optimal classification scheme in linear separation condition and ultimately, it will be changed to quadratic programming problem. In fact, the purpose of this classification method is finding the optimal solution. But in non-linear problems, SVM should be transferred to a high-dimensional feature space. In fact, if the distance between two classes is not separable linearly, mapping the upper space (feature space) is used for separating two classes. In this space, linear discriminative function is used. This function is known with the name of Kernel Function and it is called Kernel Trick. The common functions in SVM are linear function, Polynomial function and RBF (Radial Basis Function), that two recent functions are used in Feature Space (Yi et al.,2011, Pelckmans et al., 2005, Li and Cervantes, 2010).

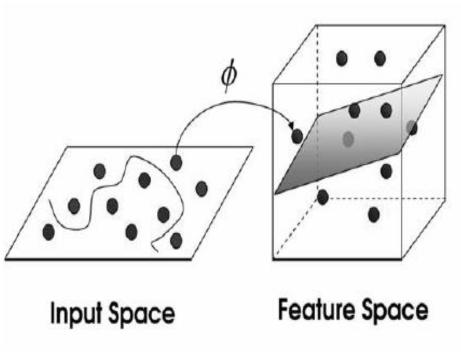
In SVM, maximizing the margin between two classes is concerned. Therefore, the hyper-plane is choose that its distance from the nearest data on both sides of linear separator be maximum. If there is such hyper-plane, it is known as maximum- margin hyper plane. In this classification method, support vector machines are data sample that they lie in area of separating plane of two classes. In (Cunhe and Chenggang, 2010), the method of data linear separation by support vector machine has been shown as:

Recebido: dia/mês/ano Aceito: dia/mês/ano



Figur1 : Method of data linear separation using support vector machine (Cunhe and Chenggang, 2010)

To make maximum margin, parallel to separator plane, two border planes are mapped, two planes spread apart to meet the data. The separator plane, that it has the maximum distance from the border planes, will be the best separator. In fact, the optimal hyper- plane in SVM, is the separator between support vectors. In the training step, only the points are kept as a learning model that are close to the hyper plane. These points are called support vectors. Fig. 2 is shown mapping the input space to feature space.



Figur 2 : mapping the data to feature space for non-linear classification in SVM (Kim, 2011)

Support vector machine, is a supervised classifier. Learning under supervision is a general method in machine learning that a set of input- output pairs give to system and base of it, system is tried to surround function of input to output. This learning require to some input data for system instruction (Shvaiko and Euzenat, 2013). Providing the adequate use of it, this algorithm will have good generalization power and despite high dimensions, the overfitting is kept away. The reason of generalization property SVM, is identical to maximizing the distance between two alignment and non- alignment categories. Overfitting is case that classifier is operated well only on training data, but don't have good operation on test data. In the other word, if the output of one classifier on training data is true 100% and its output on test data is true 50%, in fact, the classifier output can be true 75% on both of them. This condition is overfitting (Domingos, 2012).

The other property of SVM is flexibility in selecting one similarity function, but selecting the adequate core function in support vector machine can be its weakness. Selecting the inadequate core lead to weak efficiency or algorithm non-performance (Kim, 2011). This classifier is possible to meet the problem such as right non- function and increasing the calculating time and necessary memory for training and classification in condition that the training data be extensive (Xe and Geng, 2012).

2.2 K-Nearest Neighbors (KNN)

KNN, is a supervised classifier. This method suppose that data are in a feature space. Precisely, it can be said that data suppose as data spots in a metric space. These data can be as multi-dimensional vectors. Whereas, data lies as spots in feature space, the concept namely distance is created between them that it isn't Euclidean distance necessarily, although this kind of distance usually is used. Each of training data includes a set of vectors and class labels that these labels correlate with each vectors. In the simplest case, class labels are positive or negative. In fact, classification is accomplished in two positive and negative classes. Of course, KNN can be worked with ideal number of classes. Also, one k number give to algorithm. This number decide what number of neighbors influence on classification. If k=1, then the algorithm is called nearest neighbor algorithm.

In the other case, that k is more than 1, we try to find the nearest neighbor k and make the majority voting. We can summarize the process of method execution KNN and present mathematical relationship in classification process as follow (Chang and Liu, 2011):

1. Training samples are shown as feature vectors based on the standard vector model.

2. Unclassified sample t_i is shown with feature vector d_i.

Recebido: dia/mês/ano Aceito: dia/mês/ano

3. Similarity between unclassified sample d_i and training sample is calculated as follow:

$$s(d_i, d_j) = \cos(\theta) = \frac{d_i^T d_j}{||d_i||.||d_j||} \qquad (3)$$

Where, θ is the angle between vectors di and d_j and ||d|| is vector length.

4. The nearest- k of unclassified sample neighbors d_i is determined.

5. Based on k-nearest neighbors, weight of candidate category is determined as follow:

$$p(d_i, C_k) = \sum_{j=1}^{k} S(d_i, d_j)\delta(d_j, C_k) \qquad (4)$$

In this relation, $S(d_i,d_j)$ is similarity of vectors d_i and d_j. Also, $\delta(d_j,C_k)$ that is classification function is defined as follow:

$$\delta(d_j, C_k) = \begin{cases} 1 & d_j \in C_k \\ 0 & d_j \notin C_k \end{cases} \qquad (5)$$

In classification function, C_k is class or category k. This function can have two values. If sample d_j be desired in class, 1 is allocated to function. Otherwise, amount of classification function is zero.

6. Comparing the weight of each category, the unclassified sample t_i is classified to category with maximum weight.

It can be said that KNN is a non-parametric learning algorithm and lazy. The non-parametric means that the algorithm does not consider non-assumption for distribution of educational data. This property is benefit in real world, because a lot of real data (such as Gaussian Mixture) have not been made based on theoretical assumptions. The other property of KNN is its lazy and this means that KNN doesn't use training data for any generalization. In the other word, there is no explicit training phase and or if there is, it's very fast and low. The other property KNN is non-generalization and this means that this classifier is kept all of training data, because all data requires in trial phase. This property KNN is against SVM technique which the non-support vectors can be ignored during the trial phase. Many of lazy algorithms and specially KNN decide on the basis of existing training data that in the best case benefit a sub-branch of training set for decision. There is a duality here and it was not training phase and against a costly testing phase. Cost are memory and time here. At the worst case, if all training data are used in final algorithm decision, it will certainly spend more time. Also, more memory is required to store all training data (Chang and Liu, 2011, Jing et al., 2013, Suguna and Thanushkodi, 2010).

2.3 Decision Tree (DT)

Decision tree is a supervised learning method that this is usually used for classification. Data collection is taught and modeled in decision tree. Therefore, each time a new data sample is examined, it will be classified based on the created model (Mahmood Ali and Rajamani, 2012). Also, samples were grouped in a way that grow from the roots to downward. In a decision tree, each internal node or non-leaf is specified with a feature. This feature is introduced a question in relation to input data. For finding the best attribute in each node, given a small subset of training samples is enough that pass from that node. In each internal node, there is a branch on the number of possible answers to this question that each is marked with amount of that answer. The leaves of this tree has been specified with a class or a category of answers that the amount written on the leaves is formed the output.

The various decision trees have been developed. These algorithms in terms of efficiency have accuracy and different cost. So classification is important in a matter to know which algorithm is the best for use. One type of this algorithm is ID3 (Jackson, 1988) and is one of the oldest type of decision tree. This algorithm, despite making a simple and benefit tree, when complexity is increased, is faced with reduced accuracy (Thirumuruganathan, 2010). Therefore, Intelligent Decision Tree Algorithm (IDA) (Tu and Chung and, 1992) and Algorithm C4.5 (Wu et al., 2008, Quinlan, 1993), have been developed. In follow, algorithms ID3, IDA and C.4.5 are introduced:

ID3:

ID3 is a supervised learning algorithm. This algorithm is tried to specify the features that the samples of a class are separated from the other classes. For development and formulation ID3, information theory (Breiman, 1996) and pattern recognition have been used. A key feature of information theory is information term that often is obtained its mathematical sense as a numerical measurable value based on probabilistic model. So that the solution of many of important problems of storage and data transfer can be

Recebido: dia/mês/ano Aceito: dia/mês/ano

formulated with this criterion. Data measurement function, it is called entropy, is used as a standard function. Entropy is determined disorder and lack of purity in a series of samples (Navada et al., 2011).

IDA:

In algorithm IDA, divergence standard is used instead of entropy. Divergence have been defined as degree that everything of it have swing. The difference between ID3 and IDA can be summarized in the following cases:

ID3 uses a greedy search method. To build lower level trees, ID3 often produce inaccurate trees. Classification process in IDA is more accurate than ID3 computationally. Analysis time is shown that IDA is more efficient than ID3 computationally. Dependence relationships between variables is not being considered in algorithm ID3, which is generally worse than the result of classification process in decision tree. This problem has been solved successfully by IDA. It can be concluded that IDA than ID3 is more efficient and effective algorithm (Navada et al., 2011).

C4.5:

ID3 have limitations among which it is hypersensitivity to a feature a lot. To be able to use this algorithm or any other classification algorithm as the search factor in internet that includes many features or value, we must overcome this limitation. This problem solved by C4.5. C4.5 is enumerated as developed version of algorithm ID3.

Solution C4.5 for overcoming to the mentioned problem is use of a measure called information gain (Quinlan, 1986). Information gain of a feature is amount of decreasing entropy that is achieved by separating samples by this feature.

C4.5 is enumerated as development ID3 that it can calculates unavailable values, features range of continuous values, decision trees trimming, others. In this algorithm with estimating the probability of various possible outcomes, we can categorize cases which characteristics are unknown. Algorithm ID3 have special difficulties. This algorithm only can be operated on nominal data. Also, ID3 is not able to deal with a set of noisy data and algorithm may not be strong in these conditions. C4.5 advantage is that, unlike ID3, in terms of noise well operates. Avoiding overfitting problem and dealing with missing attribute is the other features of C4.5. Also, this algorithm can help trees to become rules (Navada et al., 2011).

Against IDA it can be said that, C4.5 well operate than this algorithm. C4.5 is operated with continuous features, while IDA only pay attention to discrete features. Therefore, C4.5 can be used in many of real life condition. Algorithm C4.5 can also pay attention to discrete features and also well operate against continuous features. C4.5 creates a threshold limit then features list is divided to two categories. One that its values is more than threshold limit and the other one,its values is less or equal with threshold limit. Now, j48 Java implementation is of decision tree C4.5 (Mohmood Ali et al, 2012).

2.4 Random Forest (RF)

In 2001, Breiman introduced the concept of random forests based on bagging theory (Breiman and Friedman, 1984). One decision forest is group of different decision trees that they operate in parallel form. Random Forest is presented a different content of Boosting, although both of them use a group learning method. The difference of these two classifiers is that each tree is classified data from the other trees in random forest independently, while in Boosting, each base classifier help to develop better final result via improvement of obtained result from own previous counterpart. Breiman suggested that random operation has applied in selecting features set and training data sample in classification process. His reason was no need to all features in making true result. He said that even in some cases, use of all features lead to decreasing accuracy and precision in final result.

The random forest is one of supervised machine learning techniques. This algorithm uses the decision tree as base classifier. In this process, multiple decision trees are made. Random operation in Random Forest is implemented in two forms: first, random sampling means the same method that is utilized in Bagging. Second, random choosing of input features for producing discrete base decision trees. The power of this discrete trees and correlation between them is a key matter in Random Forest and error of classifier generalization is proceed of it.

Recebido: dia/mês/ano Aceito: dia/mês/ano

A Random Forest is a set of classifiers with tree structure and each tree is produced a unity vote for classification. For a new sample, sample is given to each forest tree and each tree is produced a vote about sample class. Prediction of class for new sample is performed with collecting the votes of all trees and then majority voting. This process can be defined as follow (Khoshgoftar et al., 2007):

1. Sampling B_i from original data that it is done randomly and by replacing the original data. Here, $|B_i| = ||D||$ and samples are obtained from D randomly. Now, these samples set up training sets and having a decision tree algorithm, tree t_i is created:

A) For each tree node, set of features or attributes are limited to a set of k feature:

$$(x_1, x_2, \dots, x_k)$$

B) For a tree in standard random forest, pruning is not done.

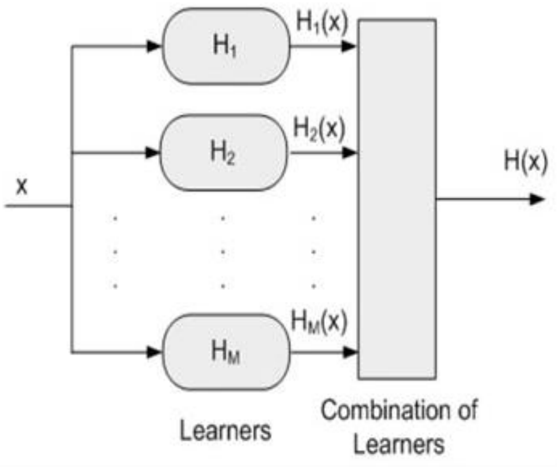
2. Steps 1 and 2 repeat for i=1,...,n and n tree t_i is made.

3. Votes (decisions) are collected from trees t_i and, new sample is classified using majority voting.

In the other word, first, main process in Random Forest is choosing k sample from a set of primary training data that using Bagging technique is done randomly. Ultimately, between classifier K is voting for choosing F of optimal classification (Han et al., 2013).

3 Ensemble classifiers

The idea of making ensemble classifier was formed at the end of ninety's decade, and instead of creating a single complex classifier, the design of a combination of some weak classifiers was targeted. For example, instead of training a big Neural Network, some simpler neural network can be trained and their separate output can be combined for the final output production. It let us have faster training and have concentration on each neural network in part of training set. In Fig.5, the concept of ensemble classifiers is observed. Pattern of input X is classified by each weak learner. Then, outputs of these weak learners are combined for classification making final decision. Assuming no correlation of separate classifiers, majority voting on the ensemble classifiers, should lead to better results than using a classifier unit (Ferreira, 2007).



Figur 3 : Concept of Ensemble Classifiers (Ferreira, 2007)

In [32] a relation is expressed for output of an ensemble classifier. According to a linear combination of weak classifiers output, output of the group and the final decision of classification are calculated as follows:

$$H(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m H_m(x)\right) \quad (6)$$

In this relation, α_m is weight of each weak classifier H_m.

Although, group of classifiers can be taught in different way, but Boosting techniques are suitable for this purpose. Boosting have been composed the sequential of a linear combination of a linear combination of base classifiers. This technique focuses on difficult samples of classification. In this classification method, base learnings can be support vector machine, neural networks and or decision tree.

Boosting and Bagging (Breiman, 1996) are among ensemble learning methods that include a complete family of similar methods and use the voting for combining the trained base models by a unit learning algorithm. In Bagging, the origin of base models is based on chance, while in Boosting it is tried to produce supplement base models by subsequent models of learning and faults of previous models are brought into account. This method is began with learning the initial base model in set of total learning together with samples with equal weight. For subsequent base models, we ask them samples predicts correctly which not predicted correctly by previous base models. Therefore, weight of this samples is increased (or weight of not predicted correctly samples is decreased) and a new base model is learned. Base models of new learning are stopped when some stop standards are satisfied (for example, when the precision of new base model be lower and or equal to 0/5). At the end of this process, ensemble prediction is obtained by weighted voting, that more weight is given to base models with upper precision. Weight of all classifiers, which have voted for a particular class, gathered and the class with the highest vote had been predicted.

3.1 Bagging Classifier

Bagging was proposed in 1996 for improving classification with combining classifications of training sets. Bagging have been used Boostrap Aggregating for different estimation. In Bagging, it is supposed that set of training data is representative of population under study and species of realized states of population can be simulated from this dataset. When each new sample is entered to each classifiers, a majority agreement is used to desire class is diagnosed.

Parameter T is considered as the number of iterations. As a result of iterations, T of Boostrap sample is produced named s_1 ،s_2 ,..., s_T. From each sample S_i, one classifier called c_i have been provided by same learning algorithm. The final classifiers called c^* are created by gathering classifier T. In the other word, the final classification of sample x have been produced with a uniform voting on c_1،c_2و ...،c_T (Kotsiantis, 2011).

The main core in Bagging is majority voting on result of significant amount of Boostrap samples. As an initial approximation, majority voting is helped to disregard the effect of random changes. This technique can be used for evaluating the precision of utilized approximations in data mining methods via sampling with replacing the training data (Kotsiantis, 2011).

3.2 Boosting Classifier

Boosting in binary matters is utilized three type of weak classifier. First, classifier is trained random subset of training data. Second, classifier is trained on samples that half of them were classified by first classifier correctly and the other half were classified by mistake. Third classifier is trained on samples which two previous classifiers don't agree about them. Finally, these three classifiers are combined using majority voting. It has been demonstrated about Boosting that error of classification is less than the best classifiers. In fact, Boosting is subset of reinforcement methods that it can be achieved in arbitrary small error on set of training data (Ferreira, 2007).

An interesting feature of some Boosting methods is that they are offering a theoretical guarantee the accuracy and precision (Li et al., 2008, Haratian Nezhad et al., 1388). It can be shown that the predicted group error on training data can be reduced quickly. To do this, we will increase the number of base learners. The only prerequisite for reducing the error is that individual error be less than 5.0 in the group. Usually, this condition is estimated easily for the binary classification. While ensuring a small error in the set of learning is not a guarantee of a small error in the unseen samples (Keykha et al., 2010). Now, AdaBoost (Adaptive Boosting) is a popular algorithms and the first practical approach of Boosting learning method (Galar et al., 2012).

AdaBoost:

AdaBoost is a training a strong classifier with a linear combination of a set of weak learner (Seyedhosseini et al., 2011). Also, this algorithm focuses on very hard training samples using a sample weighting strategy. In the other words, the underlying idea in AdaBoost is that weight to be allocated to them instead of random data sampling and weight of a sample is updated based on its importance in classification by subsequent weak classifiers.

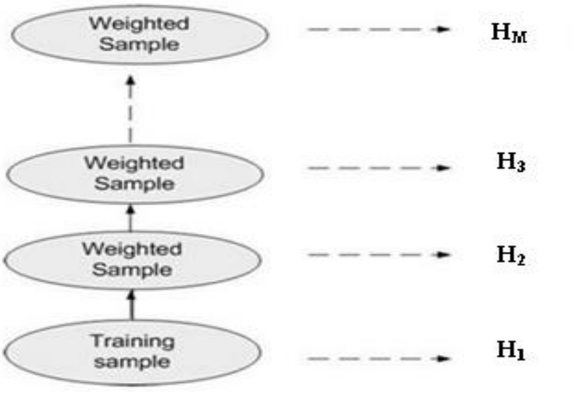
Base classifiers, which they are the same weak learners, are chosen to minimize the error in each iteration step during training process. As mentioned above, AdaBoost provides a simple and useful method for ensemble classifiers production. Group performance depends on diversity of base classifiers as well as performance of each base classifiers. However, AdaBoost algorithm has been focused on the problems of minimizing the error. Therefore, the

Recebido: dia/mês/ano Aceito: dia/mês/ano

developments have been introduced on this algorithm to inject diversity and enhance performance of AdaBoost classifiers (Ki An and Hyun Kim,2010). Then, AdaBoost algorithm and its methodology are studied more precisely:

At the first, this algorithm creates a set of classifiers (this algorithm is sometimes also called hypothesis) then what has been announced by a separate single category is combined together using a weighted majority voting. These classifiers, with training a weak classifier and then using the samples that are taken from a distribution, are produced. This distribution is updated as iteration. Updating ensures that the samples, which were classified by previous classifiers incorrectly, are adjusted in a set of training data of subsequent classifier with high probability. This matter causes the subsequent classifier focuses more attention to these cases and in the other words will focus on difficult samples.

Regarding to a set of training samples, AdaBoost is preserved a weight distribution, W, on samples. At the first, this distribution is set uniformly. Then, AdaBoost calls learning algorithm frequently in a series of cycles. Weight distribution is updated on patterns of training set inter iterations based on classification precision in previous classifiers. Samples that are not classified correctly, for the next iteration will gain weight. However, weight of samples corrected classification is reduced. Amount of change on weight of each sample is proportional to rate of sample classification error. The idea of AdaBoost is seen in Figure 6. Training set is always same in each iteration and weight is allocated to each input sample based on its correct or incorrect classification by previous classifiers. Increasing and decreasing weight allow us to focus on difficult samples for current classifiers. Difficult sample is one that is not classified correctly by previous classifiers (Ferreira, 2007, Seyedhosseini et al., 2011)



Figur 4: Graphic view of classification in AdaBoost Algorithm (Li et al., 2008)

Also, process of algorithm execution can be introduced as follow cycle:

1. Input: a set of labeled training samples

$$\{(\boldsymbol{x_1}, \mathbf{y_1}), \dots, (\boldsymbol{x_N}, \mathbf{y_N})\}$$

Algorithm of base learning, number of cycles (orbits) T.

2. Primary Value: weight of training samples: $w_i^1=1/N$ for all i=1,….,N

3. Do it for t=1,…,T

i. For training a base classifier (h_t) on weighted training samples, use the base leaning algorithm.

ii. Calculating the training error h_t

$$h_t\colon \varepsilon_t = \sum_{i=1}^{N} w_i^t, y_i \neq h_i(x_i)$$

iii. Formulating the weight for base classifier

$$h_t\colon \alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$$

iv. Updating the weight of training samples

$$w_i^{t+1} = \frac{\exp\{-\alpha_t y_i h_t(x_i)\}}{c_t}, i = 1, \dots, N$$

Where c_t is a normalized constant and $\sum_{i=1}^{N} w_i^{t+1} = 1$

4. Output:

$$f(x) = sign\left(\sum_{t=1}^{T} \alpha_t\, h_t(x)\right)$$

In cycle t, AdaBoost is provided training samples with a weight distribution w_t for base learning that at the first, this weight is same for all samples. In response, base learner teaches a classifierh_t. Weight distribution w_t is updated

Recebido: dia/mês/ano Aceito: dia/mês/ano

after each cycle based on prediction results on training samples. Easy samples that were classified correctly, gain the lower weight and difficult samples that have not been classified, gain more weight. So, AdaBoost is focused on samples with more weight which it seems to be harder for base learner. This process continues for cycles T and eventually, AdaBoost is combined all base classifiers with a final hypothesis f linearly. Greater weights are given to base classifiers with less training error. The theoretical important property AdaBoost is that if base classifiers have slight precision better than half sequentially, then the training error depended on final hypothesis quickly goes to zero. This means that the base classifiers only require to act a little better than random case (Li et al., 2008).

This algorithm is one of the multiple learning methods. Strong theoretical foundations, precise calculation and simplicity are its features. According to IEEE International Conference on Data Mining (ICDM), AdaBoost is one of the top ten data mining algorithms. It also KNN, C4.5 and SVM have been set among the most powerful algorithms (Salari and Adibnia, 1389). However, education can be, in some cases, time-consuming. For example, in the training of large databases may use this algorithm is not efficient in terms of time.

Because classification complexity in such cases is high and convergence of learning algorithm are faced with a range of complex decision-making and convergence rate decreases. In this case, the poor classification of the first cycle has affected on the further weighting process and the focus of subsequent classifiers make difficult on strict sample (Li et al., 2008).

4 Conclusion

Classification methods are used in various fields of data mining. In each of these fields, it is important to carefully choose the best classifier. Some of these, such as Boosting and Bagging, are run classification process by combining a number of base classifiers. AdaBoost is the most important algorithm Boosting. This algorithm is training a strong classifier with a linear combination of a set of weak learners. In contrast, an ensemble classifiers called an ensemble learner, there are single classifiers (non-ensemble) such as support vector machines, decision trees, k-nearest neighbor and random forest. These classifiers can be used as base learner in ensemble classification methods.

References

Bernard, S., Heutte, L., and Adam, S. (2007). Using Random Forests for Handwritten Digit Recognition, in *9th IAPR/ IEEE International Conference on Document Analysis and Recognition ICDAR'07*, Brazil, pp. 1043-1047.

Breiman. L. (1996). Bagging Predictors, *Machine Learning,* vol. 24, pp. 123-140.

Breiman, L.(2001). Random forests,*Machine Learning* vol. 45, pp. 5-32.

Breiman, L., Friedman, J. H. (1984). Classification and regression trees, Monterey.

Chang, Y., and Liu, H. (2011). Semi-supervised classification algorithm based on the KNN, presented at the 2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN).

Cunhe, L., Chenggang, W. (2010). A New Semi-Supervised Support Vector Machine Learning Algorithm Based on Active Learning, presented at the Future Computer and Communication (ICFCC).

Domingos, P. (2012). A Few Useful Things to Know about Machine Learning, *Magazine Communications of the ACM,* vol. 55, pp. 78-87.

Ferreira, A. (2007). Survey on Boosting Algorithms for Supervised and Semi-supervised Learning, 12 October.

Fix, E., Hodges, J.(1951).Discriminatory analysis. Nonparametric discrimination: Consistency properties. *Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas.*

Freund, Y., Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *journal of computer and system sciences,* vol. 55, pp. 119-139.

Galar, M., Fernandez, A.,Varrenechea, E., Bustince, H., Herrera, F. (2012). A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches, *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND*

Recebido: dia/mês/ano Aceito: dia/mês/ano

CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, vol. 42.

Han, J., Liu, Y., Sun, X. (2013). A scalable random forest algorithm based on MapReduce, *Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on*, pp.849-852.

Haratian Nezhad, A., Shadegar, B., Assare, A. R. (1388). Automatic and Optimal Orientation of Anthologies in OWL Document, 15th Annual International Conference of Iran Computer Community, Computer Community, Center of Power Technology Development, Tehran, Iran, 1388.

Jackson, A. H. (1988). Machine learning, *Ezpert Systems,* vol. 5, pp. 132-150.

Jing, Y., Gou, H, Zhu, Y. (2013). An Improved Density-Based Method for Reducing Training Data in KNN, *2013 International Conference on Computational and Information Sciences*.

Keikha, M. M., Nematbakhsh, M. A., Tork Ladani, B. (2010). Adaptive Similarity Aggregation Method for Ontology Matching, *Computer Modeling and Simulation (EMS), 2010 Fourth UKSim European Symposium on* pp. 391-396.

Khoshgoftar, T. M., Golawala, M., Van Hulse, J. (2007). An Empirical Study of Learning from Imbalanced Data Using Random Forest.," presented at the 19th IEEE Conference on Tools with Artificial Intelligence.

Ki An, T., Hyun Kim, M. (2010). A New Diverse AdaBoost Classifier, *2010 International Conference on Artificial Intelligence and Computational Intelligence.*

Kim, B. (2011). Support Vector Machine & Classification using Weka, *SNU Biointelligence Lab*.

Kotsiantis, S. (2011). Combining bagging, boosting, rotation forest and random subspace methods, *Artif. Intell. Rev.* 35.

Li, X., Cervantes, J. (2010). A Novel SVM Classification Method for Large Data Sets, presented at the 2010 IEEE International Conference on Granular Computing (GrC), San Jose.

Li, X., Wang, L., Sung, E. (2008). AdaBoost with SVM-based component classifiers, *Engineering Applications of Artificial Intelligence,* vol. 21, pp. 785-795.

Mahmood Ali, M., Rajamani, L. (2012). Decision tree induction: Priority classification, *2012 International Conference on Advanced in Engineering, Sience and Management,* 30-31 March, pp. 668-673.

Mohamed, W. N. H. W., Mohd Salleh, M. N., Omar, A. H. (2012). A Comparative Study of Reduced Error Pruning Method in Decision Tree Algorithms, *2012 IEEE International Conference on Control System, Computing and Engineering,* 23-25 Nov.

Navada, A., Ansari, A. N., Patil, S., Sonkamble, B. A. (2011). Overview of Use of Decision Tree algorithms in Machine Learning," *2011 IEEE Control and System Graduate Research Colloquium*.

Pelckmans, K., Suykens, J. A. K., De Moor. B. (2005). Building sparse representations and structure determination on LS-SVM substrates, *Neurocomputing,* vol. 64, pp. 137-159.

Quinlan, J. R. (1993) *C4.5: programs for machine learning*, San Francisco, CA, USA: Morgan Kaufmann Publishers.

Quinlan, J. R. (1986). Induction of decision trees," *Machine Learning,* vol. 1, pp. 81-106.

Salari, S.M., Adibnia, F. (1389). 10 Algorithm of Data mining Tops, 13th Student Conference of Iran Electric Engineering, University of Tarbiat Modaress, Tehran, Iran, 24-26 Shahrivar.

Seyedhosseini, M., Paiva, A. R. C., Tasdizen, T. (2011). Fast AdaBoost training using weighted novelty selection, presented at the Proceedings of International Joint Conference on Neural Networks (IJCNN), California,USA.

Sharma, S. Agrawal, J. Agarwal, S.(2013). Machine Learning Techniques for Data Mining: A Survey, IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Enathi, 26-28 Dec.

Shvaiko, P., Euzenat, J. (2013). Ontology Matching: State of the Art and Future Challenges, *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,* vol. 25.

Suguna, N., Thanushkodi, K. (2010). An Improved k-Nearest Neighbor Classification Using Genetic

Recebido: dia/mês/ano Aceito: dia/mês/ano

Algorithm, *IJCSI International Journal of Computer Science Issues,* vol. 7.

Thirumuruganathan, S. (2010). *A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm.*

Tu, P.-L., Chung, J.-Y. (1992). A New Decision-Tree Classification Algorithm for Machine Learning," *Fourth International Conference on Tools with Artificial Intelligence(TAI '92),* pp. 370-377.

Vapnik, V. (1998). Statistical Learning Theory, *Wiley-Interscience,* New York.

Wu, J., Kim, Y. S., Song, C.-H., Lee, W. D. (2008). A New Classifier to Deal with Incomplete Data, *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing.*

Xu, Q., Geng, S. (2012). A Fast SVM Classification Learning Algorithm Used to Large Training Set, presented at the 2012 International Conference on Intelligent Systems Design and Engineering Application, Sanya, Hainan.

Yi, D., Wei, C., Shengfeng, L. (2011). A new Regression Method Based on SVM .

Recebido: dia/mês/ano Aceito: dia/mês/ano