UDC 681.5

An Ensemble-Based Feature Selection Algorithm Using Combination of Support Vector Machine and Filter Methods for Solving Classification Problems

[1] Lev V. Utkin
[2] Yulia A. Zhuk
[3] Anatoly I. Chekh

[1] Saint-Petersburg State Forest Technical University, Russian Federation
5, Institutski pereulok, Saint-Petersburg, 194021
E-mail: lev.utkin@gmail.com
[2] Saint-Petersburg State Forest Technical University, Russian Federation
5, Institutski pereulok, Saint-Petersburg, 194021
E-mail: zhuk_yua@mail.ru
[3] Saint-Petersburg State Forest Technical University, Russian Federation
5, Institutski pereulok, Saint-Petersburg, 194021
E-mail: anatoly.chekh@gmail.com

Abstract. A new feature selection algorithm for solving classification problems is proposed. The algorithm exploits the ensemble-based methodology and iteratively combines classifiers in order to assign weights to features characterizing their importance in classification. The algorithm is based on the joint use of a filter method and the well known support vector machine. Moreover, the filter method uses only support vectors instead of the total training set to calculate the feature weights. Numerical experiments with publicly available data sets show that the proposed algorithm improves the classification accuracy.

Keywords: composition algorithm; support vector method; filtration method.

Introduction

Classification is one of the most common and studied statistical analysis tools which is often regarded as a part of a general framework of learning theory proposed by Vapnik [23]. A main goal of statistical machine learning is prediction of an unobserved output value y based on an observed input vector $\mathbf{x}$, which requires estimation of a predictor function f from training data consisting of pairs $(\mathbf{x}, y)$. In classification, the output variable is in one of a finite number of classes and the main task is to classify the output y corresponding to each input $\mathbf{x}$ into one of the classes by means of a separating function. We consider below only classification problems with two classes referred as binary classification problems.

The *binary classification* problem can be formally written as follows. Given n training data (examples, instances, patterns) $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$, in which $\mathbf{x}_i \in \mathbf{R}^m$ represents a feature vector involving m features and $y_i \in \{-1, 1\}$ represents the class of the associated examples, the task of classification is to construct an accurate classifier c : $\mathbf{R}^m \to \{-1, 1\}$ that maximizes the probability that $c(\mathbf{x}_i) = y_i$ for $i = 1, ..., n$. Generally $\mathbf{x}_i$ may belong to an arbitrary set, but we consider the special case for simplicity $\mathbf{R}^m$.

A classification problem is usually characterized by an unknown probability distribution $p(\mathbf{x}, y)$ (or a cumulative probability distribution function $F(\mathbf{x}, y)$) on $\mathbf{R}^m \times \{-1, +1\}$ defined by the training set or examples $\mathbf{x}_i$ and their corresponding class labels y_i. Many classification models accept the uniform distribution $p(\mathbf{x}, y)$ which means that every example in the training set has the probability $1/n$. In particular, the empirical risk functional [23] and the well known *support vector machine* (SVM) method [4] exploit the assumption of the uniform distribution.

One of the very popular approaches to classification is the *ensemble* methodology [7, 14, 20]. A basic idea of the classifier ensemble learning is to construct multiple classifiers from the original data and then to aggregate their predictions when classifying unknown samples. It is carried out by means of weighing several weak or base classifiers and by combining them in order to obtain a classifier that outperforms every one of them. The improvement in performance arising from ensemble combinations is usually the result of a reduction in variance of the classification error. This occurs because the usual effect of ensemble averaging is to reduce the variance of a set of classifiers.

Accuracy of classifiers can be substantially improved if a smaller subset of variables is used [2]. It can be done by using a procedure called feature selection which can be viewed as a process of determining what inputs should be presented to a classification algorithm. Many feature selection methods are based on the assumption that the feature set contains irrelevant and redundant features. Roughly speaking, *irrelevant features* contain no useful information improving the corresponding classifier or the classification model using the training sets with irrelevant features. *Redundant features* contain information which is already present in more informative features. Therefore, one of the aims of the feature selection is to select a set of non-irrelevant and non-redundant features and to remain relevant features which contain useful information. In many cases, the problem of feature selection is reduced to ranking features by assigning some weights to them, which show the importance of every feature in classification results.

In the paper, we propose a new algorithm for solving the problem of feature selection which includes some elements of the ensemble-based classifiers and use an iterative procedure for computing the weights of features in order to rank them. The proposed algorithm is based on exploiting a weighted modification of the SVM.

Feature selection approaches

Three main groups of methods have been developed for feature selection: filter, wrapper, and embedded methods.

The first group of methods called *filter methods* uses statistical properties of the features to filter out poorly informative ones. Filter methods constitute a preprocessing step to remove irrelevant features. This step is performed independently from the specific learning algorithm. Selection by means of the filter methods is usually carried out before applying any classification algorithm. An excellent review of filter methods is provided by Altidor et al. [1]. Other interesting and comprehensive reviews can be found in [15, 21].

When it is assumed that the two classes are distributed with a multivariate Gaussian distribution with different mean values, but with an equal covariance matrix, and it is assumed that features are independent, then the relevance of feature *i* for discrimination is measured by the *t*-statistics as follows:

$$r(i) = \frac{\left|\mu_i^+ - \mu_i^-\right|}{\sqrt{\frac{n^+(\sigma_i^+)^2 + n^-(\sigma_i^-)^2}{n^+ + n^-}}},$$

where n^+ and n^- denote the number of examples in positive and negative class; μ_i^+ and μ_i^- are the mean values of the *i*-th feature in the positive and negative classes, respectively; σ_i^+ and σ_i^- are the respective standard deviations.

The *t*-test is a well-known statistical method for detecting differential features between two samples in training data [16,17,24]. A feature having a higher *t*-statistics between two classes is assumed to have higher class separability. This measure is such a filter method which computes the importance of each feature independently of the other features by comparing that feature's correlation to the output labels.

Closely related to the *t*-test statistical measure is the Fisher criterion score $F(i)$ defined for feature *i* as

$$F(i) = \left|\frac{\mu_i^+ - \mu_i^-}{(\sigma_i^+)^2 + (\sigma_i^-)^2}\right|.$$

A feature *i* can be considered better than a feature *j* if $F(i) > F(j)$. Fisher criterion score are sensitive to all changes in the values of features.

Another measure is the Information Gain which is commonly used in the fields of information theory and machine learning. Information Gain measures the number of bits of information gained about the class prediction by knowing the value of a given feature when predicting the class.

Chi-squared test is also a popular statistical test of the divergence between the observed and expected distribution of a feature. In feature selection, it tests whether the distribution of a feature differs between groups. The chi-square score uses the summation of squared differences between observed and expected values divided by expected values. It is based on the χ^2-statistics, and it evaluates features independently with respect to the class labels. The larger the Chi-squared, the more relevant the feature is with respect to the class [1].

Relief-F [5, 12] is an instance-based feature selection method which evaluates a feature by how well its value distinguishes samples that are from different groups but are similar to each other. For each feature *s*, Relief-F selects a random sample and *k* of its nearest neighbors from the same class and each of different classes. Then *s* is scored as the sum of weighted differences in different classes and the same class. If *s* is differentially expressed, it will show greater differences for samples from different classes, thus it will receive higher score (or vice versa).

Gheyas and Smith [9] mention also additional used filter methods including Mann-Whitney-Wilcoxon U-test [6], mutual information [18], Pearson correlation coefficients [3], principal component analysis [11].

A second approach (*wrapper methods*) generally provides more accurate solutions than the filter methods, but it is computationally demanding [13]. According to the wrapper methods, feature selection is wrapped in a learning algorithm. A wrapper algorithm explores the feature space to score feature subsets according to their predictive power, optimizing the subsequent induction algorithm that uses the respective subset for classification. One of the well-known wrapper methods is the Recursive Feature Elimination (RFE).

RFE is a recently proposed feature selection algorithm described by Guyon et al. [10]. The algorithm is based on the assumption that removing a redundant feature leads to small changes of the risk measure or the cost function. Hence, we have to find and order differences between the risk measure *R* being minimized and the risk measure $R(t)$ caused by removing the *t*-th feature.

The wrapper methods are often used in combination with the filter methods. For example, Mundra and Rajapakse [17] enhance the support vector machine recursive feature elimination (SVM-RFE) method for gene selection by incorporating a minimum-redundancy maximum-relevancy filter method. In spite of the efficiency of wrapper methods, Smialowski el al. [22] show some problems with their using.

The third approach (*embedded methods*) performs feature selection in the process of model building. One of the interesting embedded methods is the so-called l_0-SVM or Concave Feature Selection (FSV), based on the minimization of the zero norm: $\|\mathbf{w}\|_0 = |\{i \ : \ w_i \neq 0\}|$.

Support vector machine

In order to solve a classification problem, we have to find a real valued function $f(\mathbf{x}, \mathbf{w}, b)$ called the separating function whose sign determines the class label prediction. The separating function $f(\mathbf{x}, \mathbf{w}, b)$ may be parameterized with some parameters $\mathbf{w} = (w_1, ..., w_m)$, b that are determined from the training examples by means of a learning algorithm. In particular, the function f may have the form $f(\mathbf{x}, \mathbf{w}, b) = \langle w, \varphi(\mathbf{x}) \rangle + b$, where φ is a feature map such that the data points are mapped into an alternative higher-dimensional feature space G. In other words, this is a map into an inner product space G such that the inner product in the image of φ can be computed by evaluating some simple kernel $K(\mathbf{x}, \mathbf{y}) = \langle \varphi(\mathbf{x}), \varphi(\mathbf{y}) \rangle$, such as the Gaussian kernel

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\|\mathbf{x} - \mathbf{y}\|^2 / c\right).$$

Given the training data the linear separating training problem is to minimize the following risk functional:

$$R(\mathbf{w}, b) = \int_{\mathbf{R}^m \times \{-1,1\}} l(\mathbf{x}, y) \mathbf{d} F(\mathbf{x}, y).$$

Here the loss function $l(\mathbf{x}, y)$ usually takes a non-zero value when the sign of the separating function (the class label prediction) does not coincide with the class label y. The minimization of the risk functional is carried out over the parametric class of functions $f(\mathbf{x}, \mathbf{w}, b)$. In other words, the function $f(\mathbf{x}, \mathbf{w}_{\text{opt}}, b_{\text{opt}})$ provides the minimum of $R(\mathbf{w}, b)$.

In SVM, it is taken the hinge loss function $l(\mathbf{x}, y) = \max\left\{0, 1 - y\left(\left\langle \mathbf{w}, \varphi(\mathbf{x})\right\rangle + b\right)\right\}$ as a loss function. Moreover, the integral in the expression for the risk functional is replaced by the sum

$$R(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, y_i).$$

Under the above conditions, taking into account regularization and after some modification, the optimization problem for computing parameters $\mathbf{w}$, b becomes

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \to \min_{\mathbf{w},\xi,b}$$

subject to

$$y_i\left(\left\langle \mathbf{w}, \varphi(\mathbf{x}_i)\right\rangle + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1,...,n.$$

This is a quadratic programming problem. Here C is a user-specified positive parameter, which controls the trade-off between classification violation and margin maximization. In practice, the optimal parameter C can be selected using cross-validation, where the training data is randomly splitted into two parts: a training part and a validation part. The training part is used to compute $\mathbf{w}, b$ with different C, and then estimate its performance on the validation part. The value of C with the smallest validation error is then chosen as the optimal regularization parameter. The introduction of the regularization term $\|\mathbf{w}\|^2$ makes the solution more stable.

By using the well-known optimization methods for solving the quadratic programming problem, we get the dual form (Lagrangian) as

$$L(\phi) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\phi_i\phi_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n}\phi_i \to \max_{\phi}$$

subject to

$$0 \leq \phi_i \leq C,\ i = 1,...,n,\ \sum_{i=1}^{n}\phi_i y_i = 0.$$

Here $\phi = (\phi_1,...,\phi_n)$ are Lagrange multipliers, i.e., optimization variables. The function $f(\mathbf{x}, \mathbf{w}, b)$ can be rewritten in terms of Lagrange multipliers as

$$f(\mathbf{x}, \mathbf{w}, b) = \sum_{i=1}^{n}\phi_i K(\mathbf{x}_i, \mathbf{x}) + b.$$

The dual form allows us to obtain the non-linear separating function in a simple way. There are many procedures and software tools for solving the above optimization problem.

A very important peculiarity of the SVM is that the solution to the optimization problem is defined only by a part of data points which is called by a set of *support vectors* (SV). The condition for the *i*-th data point to be a support vector is $\phi_i > 0$.

The proposed algorithm

An interesting idea underlying a feature selection method proposed by Mundra and Rajapakse [17] is to use only support vectors in *t*-statistics computation to improve the performance of the *t*-statistics. We modify their method and extend it on the ensemble-based framework.

First, we introduce the weights of features $v = (v_1,...,v_m)$, $v_1 + ... + v_m = 1$, which are computed by using one of the measures determining the filter methods, for example, *t*-statistics, Fisher criterion score,

Information Gain, chi-squared test, etc. In fact, the weights v are normalized values of the corresponding filter measure. It should be noted that the weights of feature are computed on the basis of a part of training examples, namely, on the basis of support vectors obtained by using the SVM. This coincides with the method proposed by Mundra and Rajapakse [17].

Second, we use the SVM for computing parameters $\mathbf{w}, b$. At that, the weights v are assigned to every feature for all training examples, i.e., every vector $\mathbf{x}_j$, $j = 1,...,n$, is replaced by the vector $< v, \mathbf{x}_j >= (v_1 x_1^{(j)},...,v_m x_m^{(j)})$. The set of support vectors SV is determined after this procedure for computing the weights v by means of filter methods.

Third, we compute the error rate $\varepsilon_t = \sum_{i: c_t(x_i) \neq y_i} 1/n$ which is defined as a number of misclassified examples from the training set divided on n. The error rate characterizes how correctly the weights are assigned to features.

The above three steps are repeated T times. Finally, the weights of features can be computed as the weighted sum $v_{\text{final}} = \sum_{t=1}^{T} \varepsilon_t v_t$.

The formal algorithm for computing the weights of features is given below.

1) Given S is the set of all examples (the training set).
2) Assign the initial uniform feature weights $v \leftarrow (1/m,...,1/m)$.
3) $t \leftarrow 1$
4) **Repeat**
5) Compute the vectors $\mathbf{x}_j \leftarrow < v, \mathbf{x}_j >$, $j = 1,...,n$.
6) Train classifier c_t by using SVM with weights v_t for features and select support vectors $SV \subseteq S$.
7) Compute weights $v_t = (v_1,...,v_m)$ of features by using a filter method such that the weights v_t are computed on the basis of support vectors *SV*.
8) Compute error rate $\varepsilon_t \leftarrow \sum_{i: c_t(x_i) \neq y_i} 1/n$.
9) $t++$
10) **Until** $t > T$.
11) $v_{\text{final}} \leftarrow \sum_{t=1}^{T} \varepsilon_t v_t$.

Numerical analysis of the proposed algorithm

We illustrate the algorithm proposed in this paper via several real examples, all computations have been performed using the statistical software R [19]. We investigate the performance of the proposed algorithm and compare it with the standard SVM by considering the error rate ε_1 when the feature weight vector is $v = (1/m,...,1/m)$, and the error rate $\varepsilon_{\text{final}}$ when the feature weight vector is v_{final}. The t-statistics is used as a filter method.

The proposed algorithm has been evaluated and investigated by the following publicly available data sets: Haberman's Survival Data Set, Pima Indian Data Set, Mammographic masses, Parkinsons, Lung cancer, Breast Cancer Wisconsin (Diagnostic) Data Set. All data sets are from the UCI Machine Learning Repository [8]. Table 1 provides the number of examples in the training sets and the number of features for the considered data sets, while more detailed information can be found from, respectively, the data resources.

Table 1:

Numbers of examples and numbers of features for data sets

Data Set	n	m
Haberman's Survival	306	3
Pima Indian Diabetes	768	8

Mammographic masses	961	4
Parkinsons	195	23
Lung cancer	32	57
Breast Cancer Wisconsin (Diagnostic)	699	9

Table 2 shows the comparative performance of the proposed algorithm. One can see from Table 2 that $\mathcal{E}_{\text{final}} \leq \mathcal{E}_1$ for all data sets. This implies that the proposed algorithm provides outperforming results in comparison with the standard SVM.

Table 2:

Values of the error rate for the initial feature weights and for final combined feature weights

Data Set	$\mathcal{E}_1$	$\mathcal{E}_{\text{final}}$
Haberman's Survival	0.281	0.255
Pima Indian Diabetes	0.314	0.249
Mammographic masses	0.213	0.201
Parkinsons	0.359	0.195
Lung cancer	0.246	0.246
Breast Cancer Wisconsin (Diagnostic)	0.124	0.091

Conclusion

A feature selection algorithm has been proposed in the paper. The algorithm combines some peculiarities of the ensemble-based methodology, of the SVM, of the feature selection filter methods. This combination allows us to obtain a method with the better classification quality.

We have studied only one algorithm which combines feature weights after several iterations in the simplest way. However, it is interesting to study how the feature weights can be updated in accordance with the error rate value at each iteration similarly to the well-known adaptive boosting method like AdaBoost. This is a direction for future research.

Another interesting direction for future research is how to incorporate additional information about the training data taking into account the structure of training examples. For instance, the proposed algorithm might be modified and improved when we would know that all features are binary. There are several ways for improving the proposed algorithm in this case and every way might lead to outperforming classifiers.

References:

1. W. Altidor, T.M. Khoshgoftaar, J. Van Hulse, and A. Napolitano. Ensemble feature ranking methods for data intensive computing applications. *Handbook of Data Intensive Computing*, pages 349–376. Springer, New York, 2011.

2. S. Bierman and S. Steel. Variable selection for support vector machines. *Communications in Statistics- Simulation and Computation*, 38(8):1640–1658, 2009.

3. J. Biesiada and W. Duch. Feature selection for high-dimensional data- a Pearson redundancy based filter. *Computer Recognition Systems 2. Advances in Soft Computing*, volume 45, pages 242–249. Springer, Berlin Heidelberg, 2008.

4. V. Cherkassky and F.M. Mulier. *Learning from Data: Concepts, Theory, and Methods*. Wiley-IEEE Press, UK, 2007.

5. T.G. Dietterich. Machine-learning research: Four current directions. *Artificial Intelligent Magazine*, 18:97–136, 1997.

6. M.P. Fay and M.A. Proschan. Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys*, 4:1–39, 2010.

7. A.J. Ferreira and M.A.T. Figueiredo. Boosting algorithms: A review of methods, theory, and applications. *Ensemble Machine Learning: Methods and Applications*, pages 35–85. Springer, New York, 2012.

8. A. Frank and A. Asuncion. UCI machine learning repository, 2010.

9. I.A. Gheyas and L.S. Smith. Feature subset selection in large dimensionality domains. *Pattern Recognition*, 43(1):5–13, 2010.

10. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

11. I.T. Jolliffe. *Principal component analysis*. Springer, New York Berlin Heidelberg, 2005.

12. K. Kira and L.A. Rendell. The feature selection problem: Traditional methods and a new

algorithm. *Proceedings of Ninth National Conference on Artificial Intelligence*, pages 129–134, 1992.

13. R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

14. L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, New Jersey, 2004.

15. In-Hee Lee, G.H. Lushington, and M. Visvanathan. A filter-based feature selection approach for identifying potential biomarkers for lung cancer. *Journal of Clinical Bioinformatics*, 1(11):1–8, 2011.

16. Xiaobo Li, Sihua Peng, Jian Chen, Bingjian Lu, Honghe Zhang, and Maode Lai. SVM-T-RFE: A novel gene selection algorithm for identifying metastasis-related genes in colorectal cancer using gene expression profiles. *Biochemical and Biophysical Research Communications*, 419(2):148-153, 2012.

17. P.A. Mundra and J.C. Rajapakse. Gene and sample selection for cancer classification with support vectors based t-statistic. *Neurocomputing*, 73:2353-2362, 2010.

18. H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

19. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005.

20. L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.

21. Y. Saeys, I. Inza, and P. Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

22. P. Smialowski, D. Frishman, and S. Kramer. Pitfalls of supervised feature selection. *Bioinformatics*, 26(3):440–443, 2010.

23. V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

24. T. Yang, V. Kecman, L. Cao, and C. Zhang. Combining support vector machines and the t-statistic for gene selection in DNA microarray data analysis. *Advances in Knowledge Discovery and Data Mining*, volume 6119, pages 55–62. Springer, Berlin / Heidelberg, 2010.

УДК 681.5

Композиционный алгоритм отбора признаков для решения задач классификации на основе комбинации метода опорных векторов и методов фильтрации

[1] Лев Уткин
[2] Юлия Жук
[3] Анатолий Чех

[1] Санкт-Петербургский государственный лесотехнический университет, Россия
Санкт-Петербург, 194021, Институтский переулок, 5
E-mail: lev.utkin@gmail.com
[2] Санкт-Петербургский государственный лесотехнический университет, Россия
Санкт-Петербург, 194021, Институтский переулок, 5
E-mail: zhuk_yua@mail.ru
[3] Санкт-Петербургский государственный лесотехнический университет, Россия
Санкт-Петербург, 194021, Институтский переулок, 5
E-mail: anatoly.chekh@gmail.com

Аннотация. Предлагается новый алгоритм отбора признаков для решения задач классификации. Алгоритм использует модели композиции и итерационно комбинирует классификаторы для назначения весов признаков, характеризующих их значимость в задаче классификации. Алгоритм основан на совместном использовании методов фильтрации и известного метода опорных векторов. Кроме того, применяемый метод фильтрации использует только опорные вектора вместо всей обучающей выборки для вычисления весов признаков. Числовые эксперименты с использованием известных данных показали, что предлагаемый алгоритм повышает точность классификации.

Ключевые слова: композиционный алгоритм; метод опорных векторов; метод фильтрации.