A C T A U N I V E R S I T A T I S L O D Z I E N S I S
FOLIA OECONOMICA 235, 2010

Michał Trzęsiok[*]

THE IMPORTANCE OF PREDICTOR VARIABLES FOR INDIVIDUAL CLASSES IN SVM

Abstract. The model obtained from Support Vector Machines suffers from the lack of interpretation. It is usually very hard to extract the knowledge about the analyzed phenomenon from the classification model obtained by using SVMs because the classification task is realized in a high dimensional feature space. Although the method identifies the observations which are crucial for the form of the decision function, it does not show which variables are relevant and which are redundant either for the whole classification task or for each class separately.

Once the model is built, it is very valuable to recognize the relative importance of predictor variables for the shape of every class. The method we propose uses the sampling techniques, backward elimination and Rand index for evaluating whether the particular variable is redundant or not. As a result, we obtain the ranking of the predictor variables reflecting the relevant importance of the inputs for each class separately.

Key words: Support Vector Machines, relevant variables, backward elimination.

I. INTRODUCTION

Support Vector Machines (SVMs) are considered as a very powerful tool in Pattern Recognition. They are highly competitive in terms of accuracy within the group of classification methods [see Abe (2005)]. A number of researchers are working to modify and improve the performance of SVMs. One of the fields of SVM development is associated with the analysis of the importance of input variables or, more generally, with the knowledge extraction.

The feature selection problem is very important since the presence of redundant variables has a significant negative influence on the training time, model storage requirements, interpretability and sometimes even the generalization ability of the model [see Weston (2001)]. The interpretational issues connected with the SVMs models, especially when presenting the model and the results of classification to decision makers, was the main motivation for this paper. We tried to use simple intuitive techniques to develop a procedure for evaluating which predictor has a significant impact on the shape of a particular class in the SVMs classification model. Thus, the main goal of the paper was to develop the

[*] M.Sc., Department of Mathematics, Karol Adamiecki University of Economics, Katowice.

method for obtaining the profile description of the classes (after finishing learning SVMs) rather than to propose an additional element of learning.

As presented in Guyon *et al.* (2006), there are three approaches in feature selection: filters, wrappers and embedded methods. The first group of methods evaluates the importance of a variable independently from the classifier using different dependency measures, such as Pearson correlation, χ^2 or others. The whole procedure is performed at the pre–processing stage. Wrappers are much more often used for feature selection than filters since they involve the classifier to assess feature subsets (the original learning method is applied for the evaluation of the generated feature subsets). Embedded methods are close to wrappers but the learning method is modified in the way that the feature selection is incorporated into the algorithm. Since we are interested in describing the model built on the whole set of predictors given by the experts, we choose wrappers as the most appropriate approach. Within the wrappers, we use the backward elimination procedure. It is an iterative procedure where we start with all the features and delete one feature at a time. We delete the feature which deteriorates the previously chosen selection criterion the least. The alternative method – the forward selection – also generates the nested feature subsets but in this approach we start with an empty set and add into the model the predictor that improves the performance measure the most. Both procedures are local optimization techniques and yield different variable rankings. Backward elimination is usually slower but more stable than forward selection [see Abe (2005)].

II. THE OVERVIEW OF THE SVM ALGORITHM

In this section we briefly present the main ideas of Support Vector Machines. In the case of the two–class classification, first we transform data points from the training set to a higher dimensional feature space by non–linear mapping [see Vapnik (1998)]. Then we find the optimal hyperplane (maximizing the margin) separating the images of the data in the feature space. This hyperplane (linear boundary) in the feature space corresponds with the nonlinear classifier in the original data space.

Following Schölkopf and Smola (2002) we present the formalism of the SVMs algorithm. We are given the training set $D = \left\{\left(\mathbf{x^1}, y^1\right),\ldots,\left(\mathbf{x^N}, y^N\right)\right\}$, where $\mathbf{x^i} \in \mathbf{R}^d$ is the vector of predictors' values and $y^i \in \{-1,1\}$ defines the class the ith observation belongs to, $i \in \{1,\ldots,N\}$. Then the goal of supervised learning is to find a "good" predictive classification function $y = f(\mathbf{x})$, based on the

available training set. In order to obtain a model with a good generalization ability SVMs use the *structural risk minimization principle* [see Vapnik (1998)] as a criterion for finding a "good" decision function. To handle a case of linearly nonseparable classes, the training set is transformed to some higher dimensional feature space by non-linear mapping $\varphi : \mathbf{R}^d \to \mathbf{Z}$. There we construct the optimal separating hyperplane (i.e. the hyperplane with the largest margin):

$$\boldsymbol{\beta} \cdot \varphi(\mathbf{x}) + \beta_0 = 0, \tag{1}$$

where $\boldsymbol{\beta} \in \mathbf{Z}, \beta_0 \in \mathbf{R}$. This hyperplane separates the classes and defines the decision function:

$$f(\mathbf{x}) = \operatorname{sign}\left(\boldsymbol{\beta} \cdot \varphi(\mathbf{x}) + \beta_0\right). \tag{2}$$

The structural risk minimization principle is applied within SVMs by the fact that we seek for the optimal separating hyperplane (the one with the largest margin), not just any separating hyperplane. The problem of finding the optimal separating hyperplane (i.e. finding the normal vector $\boldsymbol{\beta}$ and the intercept β_0) can be formulated as the following optimization problem:

$$\begin{aligned} &\underset{\boldsymbol{\beta}\in\mathbf{Z},\beta_0,\rho\in\mathbf{R},\boldsymbol{\xi}\in\mathbf{R}^N}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \nu\rho + \frac{1}{N}\sum_{i=1}^{N}\xi_i, \\ &\text{subject to} \quad y^i\left(\boldsymbol{\beta} \cdot \varphi(\mathbf{x}^i) + \beta_0\right) \geq \rho - \xi_i, \\ &\text{and} \qquad \rho \geq 0, \quad \xi_i \geq 0, \quad i \in \{1,\ldots,N\}, \end{aligned} \tag{3}$$

where the slack variables $\xi_1,\ldots,\xi_N \geq 0$ are introduced to allow some observations from the training set to be misclassified (to allow noise in the training set), ρ is an additional variable to be optimized and ν replaces regularization parameter C in the the original SVM formulation. Parameter $\nu \in (0,1]$ is specified beforehand and controls the trade-off between the generalization ability of the model and the fitting of the training data. We can solve problem (3) using the Lagrange multipliers method. The dual form of the Lagrangian is:

$$\underset{\boldsymbol{\alpha}\in\mathbf{R}^N}{\text{maximize}} \quad -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \mathrm{K}(\mathbf{x^i},\mathbf{x^j}), \tag{4}$$

$$\text{subject to} \quad 0 \le \alpha_i \le \frac{1}{N}, \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{4}$$

$$\text{and} \sum_{i=1}^{N}\alpha_i \ge \nu, \quad i \in \{1,\ldots,N\},$$

where $\mathrm{K}(\mathbf{u},\mathbf{v}) = \varphi(\mathbf{u})\cdot\varphi(\mathbf{v})$ already denotes the *kernel function* representing the dot product in the high dimensional feature space $\mathbf{Z}$. Within SVMs one of the kernel functions is used to define this dot product. The most popular kernels for SVMs are:

- Radial Basis Function $\mathrm{K}(\mathbf{u},\mathbf{v}) = \exp(-\gamma\|\mathbf{u}-\mathbf{v}\|^2)$,
- kth degree polynomial $\mathrm{K}(\mathbf{u},\mathbf{v}) = \gamma(\mathbf{u}\cdot\mathbf{v}+\delta)^k$,

where $\gamma,\delta \ge 0$, $k \in \mathbf{N}$ are parameters. The use of the kernel function implies that we construct the optimal separating hyperplane in the high dimensional feature space $\mathbf{Z}$ (and thus the optimal decision function) without explicitly performing calculations in this space. It can be shown [see Schölkopf and Smola (2002)] that then the decision function takes the final form:

$$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{N}\alpha_i y^i \mathrm{K}(\mathbf{x^i},\mathbf{x}) + \beta_0\right). \tag{5}$$

Many of the Lagrange multipliers α_i in the solution of the optimization task (4) are equal zero [see Vapnik (1998)]. Since the decision function (5) uses the linear combination of the images of the observations, only the observations corresponding to nonzero Lagrange multipliers have discriminative power. These observations are called *support vectors*.

In the case of m classes ($m \ge 3$), the common approach is to build $\frac{m(m-1)}{2}$ binary classifiers (one–against–one multi–class SVMs) and use the majorization scheme to assign the observation to a particular class.

III. VARIABLE IMPORTANCE EVALUATION

Building the Ranking of Predictors

Performing an exhaustive search through all variable subsets is usually not feasible due to the computational costs. Therefore, we decide to develop another basic procedure using the greedy technique – the backward elimination. The computational costs for this approach are much lower than for the exhaustive search, although it is still expensive to compute if d – the number of predictors – is large.

We start with the set of all variables and recursively delete one predictor in every step. The feature which deteriorates the previously chosen selection criterion the least is deleted. In the literature [Guyon *et al.* (2006), Rakotomamonjy (2003)], typically the minimum expected prediction error is used as the selection criterion, but since it is unknown we need to estimate it e.g. by cross–validation (CV). This significantly increases the computational cost of the whole procedure. We propose a simple modification. The classification resulting from the model built on the set of all predictors can be treated as a pattern. It should be noticed that at this stage the SVMs hyperparameters are tuned and then the best kernel and hyperparameter values are used for building every model with reduced number of predictors. We do so because we only want to observe the influence of deleting variables in the training set – not the changes associated with other aspects of the learning process. Then we can compare the pattern with the classification results obtained from the model with one of the predictors excluded. We use the maximum classification agreement between the pattern and the model with a reduced number of predictors as the criterion for choosing the variable to be deleted. As a classification agreement measure we use the Rand index, which is more general because it can be used for comparing clustering results, but is also suitable for this purpose. We summarize the procedure in the algorithm presented in Table 1.

Table 1. Algorithm for evaluating the ranking of predictor variables using backward elimination and the Rand index

Step 1.	Tune the SVMs model on training set *D* using all the predictors (the best model is used as the pattern). Take the working data set *S* equal to the training set *D*.
Step 2.	Generate different modifications of data set *S* by excluding one of the input variables from data set *S* at a time and build SVMs models on these data sets using the same parameters as in Step 1.
Step 3.	Compare the Rand indexes of the classification results from Step 2 with the pattern (prediction from Step 1).
Step 4.	Delete from the data set *S* the input variable associated with the largest Rand index (deleting this predictor changes the results of the classification the least). Compute the value of the prediction error estimator using cross–validation for the model with a reduced number of predictors.
Step 5.	Go to Step 2 and repeat the procedure (on the data set *S* with the reduced number of predictors) until only one input variable is left.

In the first few iterations of the procedure we identify the least important variables and then we continue until we end up with only one – the most important variable. Thus, we read the variable labels in the reverse order to find the ranking of predictors.

Redundant Variables Identification

Once the ranking of predictors is built we can use different criteria to indicate which input variables are redundant, e.g.:

- specifying the threshold value for the Rand index,
- specifying the threshold value for the prediction error (estimated by CV–error).

When estimating the prediction error by cross–validation we can also compute the standard error. We would need to do it for each estimated prediction error. Then we can use the smallest prediction error increased by its standard error as the threshold [as proposed in Hastie *et al.* (2001)], i.e. we choose the least complex model within one standard error of the best.

The whole procedure of evaluating the importance of predictors is performed for each class separately in order to obtain the profile description of the classes. In other words we use the one–against–all strategy, treating all the observations not belonging to a given class as the second class and evaluating the influence of input variables on the decision function obtained from two–class SVMs. The procedure is repeated for every class in dataset.

IV. AN EXAMPLE ILLUSTRATING THE PROCEDURE

We used the benchmark real–world data set *Glass* taken from the "`mlbench`" package from the statistical language **R** in which all the computations were performed. As the benchmark dataset donors state, the study of the classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence if it is correctly identified. The set *Glass* consists of 214 observations, described by 9 numerical predictors: V.1 – refractive index and variables: V.2,...,V.9 – representing a weight percentage of a particular element in corresponding oxide in the glass type. The elements are: Na, Mg, Al, Si, K, Ca, Ba and Fe, respectively. One nominal variable defines the class. There are six classes of objects in this data set representing six types of glass: building windows float processed, building windows non float processed, vehicle windows float processed, containers, tableware and headlamps.

The results of the procedure presented in Table 1 realized for Class 1 (for illustration) are shown in Table 2. Since it was only the mechanism of the procedure that we wanted to present in Table 1 we do not present details for other classes.

Table 2. The results of the procedure for the variable importance evaluation for Class 1 in dataset *Glass*

No. of iteration	Deleted predictor	Rand Index	CV prediction error	Standard error
1	V.5	0,936	0,229	0,020
2	V.8	0,919	0,210	0,024
3	V.3	0,910	**0,205**	**0,029**
4	V.2	0,869	0,248	0,048
5	V.1	0,853	0,266	0,037
6	V.9	0,837	*0,234*	0,014
7	***V.6***	0,744	0,252	0,012
8	***V.7***	0,568	0,355	0,030
9	***V.4***			

Source: own results.

We can easily see that the best model (the one with the minimum prediction error – in boldface) is not the model built on the complete set of predictors but the one, where variables: V.5, V.8 and V.3 were deleted (these inputs are irrelevant for the Class 1 boundary). As proposed by Hastie *et al.* (2001) we choose the model with the fewest input variables with the prediction error not greater than the minimum prediction error (0,205) increased by its standard error (0,028). In the given example, this allowed to extend the set of redundant predictors to {V.5, V.8, V.3, V.2, V.1, V.9}. Reading the list of deleted predictors (the second column of Table 2) in reverse we obtain the ranking of input variables (from the most to the least important). Thus variable V.4 is the one with the largest discriminative power for the problem of distinguishing objects of Class 1 from any other observations.

The rankings of predictors for every class in dataset Glass are presented in Table 3.

Table 3. Rankings of predictors for every individual class in dataset Glass obtained as the result of the proposed procedure involving backward elimination and Rand index

Position in ranking	Class 1 Building windows float processed	Class 2 Building windows non float processed	Class 3 vehicle windows float processed	Class 4 containers	Class 5 tableware	Class 6 headlamps
1	**V.4**	**V.4**	**V.9**	**V.6**	**V.8**	**V.8**
2	**V.7**	**V.3**	V.7	**V.9**	**V.7**	V.5
3	**V.6**	**V.1**	V.6	**V.7**	**V.6**	V.7
4	V.9	**V.8**	V.5	**V.5**	**V.3**	V.4
5	V.1	V.9	V.4	V.4	**V.9**	V.9
6	V.2	V.5	V.8	V.3	**V.5**	V.6
7	V.3	V.7	V.3	V.8	V.2	V.3
8	V.8	V.2	V.2	V.2	V.4	V.2
9	V.5	V.6	V.1	V.1	V.1	V.1

Source: own results.

These rankings are very easy for interpretation. For example we see that Class 3 – vehicle windows – is very well separable using only one predictor V.9 (weight percentage of the iron oxide). For separating Class 4 objects it is enough to use predictors: V.6, V.9, V.7 and V.5. We can also conclude that with only one exception, predictor V.1 – refractive index – is identified as a redundant variable. It is very easy to obtain the profile description of every class from Table 3.

V. CONCLUSION

We proposed an intuitive procedure for evaluating the importance of variables for every individual class in SVMs models. The procedure is a simple modification of the recursive backward elimination technique. Instead of using the minimum CV–prediction error as the criterion for deleting input variables when creating the ranking we use the maximum classification agreement (Rand index) measuring the differences in the prediction between the model built with the whole set of variables and the reduced set of predictors. This modification is justified by the observation that deleting the redundant variable does not change the results of classification significantly. The advantage of using the Rand index instead of the CV–classification error is the reduction of computational costs of the algorithm. The algorithm is not optimal since it uses the greedy searching technique. If the number of observations is too large to perform the procedure effectively it is suggested to call it on the random subset or on the subset consisting of all support vectors identified by the model used as the pattern.

Moreover, the ranking resulting from the procedure is very easy for interpretation. We obtain the profile description of every class and it gives additional important information about the analyzed phenomenon. The example on the benchmark real–world data set demonstrates the usefulness of the presented procedure.

REFERENCES

Abe S. (2005), *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*, Springer.

Guyon I., Gunn S., Nikravesh M., Zadeh L. (red.) (2006), *Feature Extraction, Foundations and Applications*. Springer.

Hastie T., Tibshirani R., Friedman J. (2001), *The Elements of Statistical Learning*, Springer Verlag, N.Y.

Rakotomamonjy A. (2003), Variable Selection Using SVM–based Criteria, "*Journal of Machine Learning Research*", 3, p. 1357–1370.

Schölkopf B., Smola A. (2002), *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond,* MIT Press, Cambridge.

Vapnik V. (1998), *Statistical Learning Theory*, John Wiley & Sons, N.Y.

Weston J., Mukherjee S., Chapelle O., Pontil M., Piaggio T., Vapnik V. (2001), Feature selection for SVMs, "*Advances in Neural Information Processing Systems*", 13, MIT Press, p. 668–681.

Michał Trzęsiok

WPŁYW ZMIENNYCH OBJAŚNIAJĄCYCH NA KSZTAŁT POSZCZEGÓLNYCH KLAS W METODZIE WEKTORÓW NOŚNYCH

W metodzie wektorów nośnych (SVM) funkcja dyskryminująca wyznaczana jest poprzez transformację danych w przestrzeń o znacznie większym wymiarze, gdzie poszukuje się optymalnej hiperpłaszczyzny rozdzielającej klasy parami. Na skutek tej transformacji działanie metody SVM przypomina działanie „czarnej skrzynki", co oznacza, iż bardzo trudno interpretować wyniki tak otrzymanej klasyfikacji. Po zbudowaniu modelu często ważnym problemem jest znalezienie stosownego opisu klas oraz rozpoznanie, które zmienne objaśniające miały największy wpływ na kształt poszczególnych klas (zidentyfikowanie zmiennych charakterystycznych).

Głównym celem przeprowadzonej analizy jest przedstawienie procedury wykorzystującej techniki próbkowania, selekcję oraz miarę zgodności klasyfikacji, do oceny wpływu poszczególnych zmiennych diagnostycznych na kształt każdej z klas.