# Embedded Feature Selection for Support Vector Machines: State-of-the-Art and Future Challenges

Sebastián Maldonado[1] and Richard Weber[2]

[1] Universidad de los Andes, Faculty of Engineering and Applied Sciences
Av. San Carlos de Apoquindo 2200, Las Condes, Santiago, Chile
[2] Department of Industrial Engineering, University of Chile

**Abstract.** Recently, databases have incremented their size in all areas of knowledge, considering both the number of instances and attributes. Current data sets may handle hundreds of thousands of variables with a high level of redundancy and/or irrelevancy. This amount of data may cause several problems to many data mining algorithms in terms of performance and scalability. In this work we present the state-of-the-art the for embedded feature selection using the classification method Support Vector Machine (SVM), presenting two additional works that can handle the new challenges in this area, such as simultaneous feature and model selection and highly imbalanced binary classification. We compare our approaches with other state-of-the-art algorithms to demonstrate their effectiveness and efficiency.

**Keywords:** Embedded methods, Feature selection, SVM.

## 1 Introduction

Feature selection is an important topic in pattern recognition, especially in high-dimensional applications. A low-dimensional representation of the data reduces the risk of *overfitting* [3,5], improving model generalization. Feature selection is a combinatorial problem in the number of original features [3], and finding the optimal subset of variables is considered NP-hard.

Support Vector Machine (SVM) [10] is an effective classification method with significant advantages such as the absence of local minima, an adequate generalization to new objects, and a representation that depends on few parameters [5,10]. This method, however, does not directly determine the importance of the features used [5,6].

Several feature selection approaches for SVM have been proposed in the literature. An excellent review has been published by Guyon et al. [3]. Since then, several trends have arisen in concordance with the new challenges: First, given the increasing size of data sets, data mining methods are required to be more efficient in terms of training time and scalability. Data sets with millions of instances and a high level of irrelevant variables are more and more common for

C. San Martin and S.-W. Kim (Eds.): CIARP 2011, LNCS 7042, pp. 304–311, 2011.

new data mining applications, such as e. g. social network mining, and pattern recognition methods must adapt to the new challenges. In the same direction, model selection, meaning the process of fitting adjustable parameters to build the model, and feature selection are usually considered as different tasks. The advantages of developing a model selection framework that simultaneously performs feature selection and parameter fitting are the reduction of computational effort and avoiding the risk of overfitting [4]. Finally, several pattern recognition tasks involve classification with highly imbalanced data sets, and feature selection methods should be adapted to this challenge. In this paper we present two embedded methods for feature selection using SVM, comparing both approaches with well-known feature selection strategies and analyzing them in terms of the three challenges presented.

This paper is structured as follows. In Section 2 we provide a general overview of the different feature selection approaches. Section 3 introduces SVM for classification. Recent developments for embedded feature selection using SVM are reviewed in Section 4, providing experimental results using two real-world data sets. A summary of this paper can be found in Section 5, where we provide its main conclusions and address future challenges.

## 2 Feature Selection

Three main directions have been developed for feature selection: filter, wrapper, and embedded methods [3]. The first scheme (*filter methods*) uses statistical properties of the features to filter out irrelevant ones. This is usually done before applying any classification algorithm. Common filter methods are the Fisher Criterion Score, which is based on Fisher's Linear Discriminant Analysis (LDA), or entropy measures such as Information Gain [3]. This strategy has advantages, such as its simplicity, scalability and a reduced computational effort; but it ignores the interactions between the variables and the relationship between them and the classification algorithm.

*Wrapper methods* are computationally demanding, but generally provide more accurate results than filter methods. A wrapper algorithm explores the whole feature space to score feature subsets according to their predictive power. Since the exhaustive search for an optimal subset of features grows exponentially with the number of original variables, heuristic approaches have been suggested [3]. Commonly used wrapper strategies are the Sequential forward selection (SFS) and the Sequential backward elimination (SBE) [3]. In the first case, each candidate variable is included into the current set, and the resulting is evaluated. The variable whose inclusion resulted in the best evaluation is inserted in the current set. Subsequently, SBE starts with the variable set that consists of all the candidate variables, and the variable whose exclusion resulted in the best evaluation is considered to be eliminated from the current set. Advantages of wrapper methods include the interaction between subset of variables and the model. The main disadvantage is the high computational cost and the risk of overfitting [3]. Greedy strategies may also get stuck in a local optimum, leading to an unsatisfactory subset of features. To overcome this problem, several random search strategies have been proposed [3].

*Embedded methods* attempt to find an optimal subset of features in the process of model building. These methods depend directly on the nature of the classification method used. In general, embedded methods present important advantages in terms of variable and model interaction, capturing accurately the dependencies between variables, being computationally less demanding than wrapper methods. [3]. However, these techniques are conceptually more complex, and modifications to the classification algorithm may lead to a poor performance. In Section 4 several embedded approaches for SVM will be presented.

## 3 Support Vector Machine for Binary Classification

This section introduces SVM for binary classification as developed by Vapnik [10]. Given training vectors $\mathbf{x}_i \in \Re^n$, $i = 1, ..., m$ and a vector of labels $\mathbf{y} \in \Re^m$, $y_i \in \{-1, +1\}$, SVM provides the optimal hyperplane $f(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x} + b$ to separate the training classes. For a linearly separable problem, this hyperplane maximizes the sum of the distances to the closest positive and negative training instances, which is called *margin*. In order to maximize this margin, we need to classify correctly the vectors $\mathbf{x}_i$ of the training set into two different classes $y_i$, using the smallest norm of coefficients $\mathbf{w}$ [10]. For a non-linear classifier, SVM maps the data points into a higher dimensional space $\mathscr{H}$, where a separating hyperplane with maximal margin is constructed. The dual formulation of SVM can be stated as follows:

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \tag{1}$$

subject to

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C \qquad\qquad i = 1, ..., m.$$

The mapping is performed by a kernel function $K(\mathbf{x}, \mathbf{y})$ which defines an inner product in $\mathscr{H}$. From a variety of available kernel functions, the polynomial and the Gaussian kernel are chosen in many applications [4,5]:

1. Polynomial function: $K(\mathbf{x}_i, \mathbf{x}_s) = (\mathbf{x}_i \cdot \mathbf{x}_s + 1)^d$, where $d \in \mathbb{N}$ is the degree of the polynomial.
2. Radial basis function: $K(\mathbf{x}_i, \mathbf{x}_s) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_s||^2}{2\rho^2}\right)$ , where $\rho > 0$ is the parameter controlling the width of the kernel.

The selection of the best Kernel function is still a matter of research [6]. Empirically, we have achieved best classification performance with the Gaussian Kernel [5,6].

# 4 Embedded Feature Selection for SVM

According to the emerging challenges related to feature selection as identified in the introduction, we present recently developed algorithms and show how they can contribute to the future trends. Section 4.1 presents state-of-the-art embedded methods for SVM. Section 4.2 presents our previously developed approaches, which address the mentioned trends. Finally, Section 4.3 presents numerical results for two well-known benchmark data sets.

## 4.1 Related Work and Analysis

There are different strategies for embedded feature selection. First, feature selection can be seen as an optimization problem. For example, the methods presented in Neumann et al. [7] add an extra term that penalizes the cardinality of the selected feature subset to the standard cost function of SVM. By optimizing this modified cost function features are selected simultaneously to model construction. Another embedded approach is the Feature Selection ConcaVe (FSV) [1], based on the minimization of the "zero norm" : $\|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|$. Note that $\|\cdot\|_0$ is not a norm because the triangle inequality does not hold [1], unlike $l_p$-norms with $p > 0$. Since $l_0$-"norm" is non-smooth, it was approximated by a concave function:

$$\|\mathbf{w}\|_0 \approx \mathbf{e}^T(\mathbf{e} - \exp(-\beta|\mathbf{w}|) \tag{2}$$

with an approximation parameter $\beta \in \Re_+$ and $\mathbf{e} = (\mathbf{1}, ..., \mathbf{1})^{\mathbf{T}}$. The problem is finally solved by using an iterative method called Successive Linearization Algorithm (SLA) for FSV [1]. Weston et al. [12] proposed an alternative approach for zero-"norm" minimization ($l_0$-SVM) by iteratively scaling the variables, multiplying them by the absolute value of the weight vector $\mathbf{w}$. Perkins et al. consider simultaneously the three objectives goodness-of-fit, a regularization parameter for structural risk minimization, and feature penalization, considering a secuencial forward selection strategy [8]. An important drawback of these methods is that they are limited to linear classification functions [3,5].

Several embedded approaches consider backward feature elimination in order to establish a ranking of features, using SVM-based contribution measures to evaluate their relevance. One popular method is known as Recursive Feature Elimination (SVM-RFE) [4]. The goal of this approach is to find a subset of size $r$ among $n$ variables ($r < n$) which maximizes the classifier's performance. The feature to be removed in each iteration is the one whose removal minimizes the variation of $W^2(\boldsymbol{\alpha})$:

$$W^2(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i,\mathbf{x}_s) \tag{3}$$

The scalar $W^2(\boldsymbol{\alpha})$ is a measure of the model's predictive ability and is inversely proportional to the margin. Features are eliminated applying the following procedure:

1. Given a solution $\boldsymbol{\alpha}$, for each feature $p$ calculate:

$$W^2_{(-p)}(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i^{(-p)}, \mathbf{x}_s^{(-p)}) \tag{4}$$

   where $\mathbf{x}_i^{(-p)}$ represents the training object $i$ with feature $p$ removed.

2. Eliminate the feature with smallest value of $|W^2(\boldsymbol{\alpha}) - W^2_{(-p)}(\boldsymbol{\alpha})|$.

Another ranking method that allows kernel functions was proposed by Rakotomamonjy [9], which considers a *leave-one-out* error bound for SVM, the *radius margin bound*[10] $LOO \leq 4R^2||\mathbf{w}||^2$, where $R$ denotes the radius of the smallest sphere that contains the training data. This bound is also used in Weston et al. [11] through the *scaling factors* strategy. Feature selection is performed by scaling the input parameters by a vector $\boldsymbol{\sigma} \in [0,1]^n$. Large values of $\sigma_j$ indicate more useful features. The problem consists in choosing the best kernel of the form:

$$K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \equiv K(\boldsymbol{\sigma} * \mathbf{x}_i, \boldsymbol{\sigma} * \mathbf{x}_s) \tag{5}$$

where $*$ is the component-wise multiplication operator. the method presented by Weston et al. considers the gradient descent algorithm for updating $\boldsymbol{\sigma}$. Canu and Grandvalet [2] propose to limit the use of the attributes by constraining the scaling factors using a parameter $\sigma_0$, which controls the norm of $\boldsymbol{\sigma}$.

### 4.2 Proposed Methods for Embedded Feature Selection

We consider two approaches that attempt to perform feature selection and model selection (hyperparameter setting) in the same algorithm. The main idea is to define a procedure that simultaneously defines both the classifier and the selected features, instead of the standard two-step methodology that first selects features and then constructs the classifier via model selection for a given subset of variables. The first approach is a ranking method called Holdout SVM (HO-SVM) [5], which defines a new contribution measure based on the number of errors in a validation subset. Then, a backward feature elimination procedure is performed, pruning those features whose removal keeps this contribution measure small, until an explicit stopping criterion is reached: when the elimination of variables lead to a degradation of the predictive performance, i.e. the number of errors in the validation set grows by removing any feature. Algorithm 1 formally presents this approach.

The second approach, called Kernel-Penalized SVM (KP-SVM) [6], uses the scaling factors principle to penalize the use of features in the dual formulation of SVM (1). This penalization is performed by considering an additional term that penalizes the zero norm of the scaling factors, in a similar way as in (2). The respective optimization procedure is done by updating the scaling factors using a variation of the gradient descent approach, as presented in Algorithm 2.

**Algorithm 1.** HO-SVM Algorithm for Feature Selection

```
1. Initial Model selection: set $C$ and kernel parameter $\rho$, $\boldsymbol{\sigma} = (1, ..., 1)$
2. repeat
   (a) Random split of the training data in subsets $TRAIN$ and $VAL$
   (b) SVM Training (Formulation (1))using $TRAIN$ for a given subset of features
       $\boldsymbol{\sigma}$, kernel of the form presented in (5).
   (c) for each feature $p$ with $\sigma_p = 1$, do determine $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, the number of
       classification errors when feature $p$ is removed.
   (d) remove feature $j$ with the smallest value of $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$:

       $$E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma}) = \sum_{l \in VAL} \left| y_l^v - sign\left( \sum_{i \in TRAIN} \alpha_i y_i K_{\boldsymbol{\sigma}}(\mathbf{x}_i^{(-p)}, \mathbf{x}_l^{v(-p)}) + b \right) \right| \quad (6)$$

       where $VAL$ is the Validation subset and $\mathbf{x}_l^v$ and $y_l^v$ are the objects and labels
       of this subset, respectively. $\mathbf{x}_i^{(-p)}$ ($\mathbf{x}_l^{v(-p)}$) means training object $i$ (validation
       object $l$) with feature $p$ removed.
3. until the smallest value of $E_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ is greater than $E(\boldsymbol{\alpha}, \boldsymbol{\sigma})$, which is the num-
   ber of errors in the Validation subset using all features as indicated by the current
   vector $\sigma$, i.e. without removing any further feature.
```

**Algorithm 2.** Kernel Width Updating and Feature Elimination

```
Initial Model selection: set $C$ and kernel parameter $\boldsymbol{\sigma} = \rho \cdot \mathbf{e}$;
cont=true; t=0;
while(cont==true) do
    train SVM (Formulation (1), kernel of the form presented in (5)) for a given $\boldsymbol{\sigma}$;
    $\boldsymbol{\sigma}^{t+1} = \boldsymbol{\sigma}^t - \gamma \Delta F(\boldsymbol{\sigma}^t)$;
where $\gamma$ is the gradient descent parameter. For a given feature $j$, the gradient for
kernel updating $\Delta_j F(\boldsymbol{\sigma})$ is:

    $$\Delta_j F(\boldsymbol{\sigma}) = \sum_{i,s=1}^{m} \sigma_j (x_{i,j} - x_{s,j})^2 \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s, \boldsymbol{\sigma}) + C_2 \beta exp\left(-\beta \sigma_j\right) \quad (7)$$

    for all $(\sigma_j^{t+1} < \epsilon)$ do
        $\sigma_j^{t+1} = 0$;
        end for
where $\epsilon$ is the threshold for feature selection: when a kernel variable $\sigma_j$ in the
iteration $t + 1$ is below a threshold $\epsilon$, we consider this feature irrelevant, and we
eliminate it by setting $\sigma_j = 0$.
    if $(\boldsymbol{\sigma}^{t+1} == \boldsymbol{\sigma}^t)$ then
        cont=false;
        end if
    $t = t + 1$;
    end while;
```

### 4.3 Experimental Results

We applied the proposed approaches for feature selection and the alternative embedded methods FSV and SVM-RFE on two well-known benchmark data sets: A real-world data set from the UCI data repository, and a DNA microarray data set. Wisconsin Breast Cancer data set (WBC) contains 569 observations described by 30 continuous features, while Colorectal Microarray data set (CRMA) contains the expression of the 2000 genes with highest minimal intensity across 62 tissues. Results in terms of mean classification accuracy over 100 realizations using the test subset are shown in Table 1, where the first two rows consider the stopping criterion for HO-SVM and the latter two rows the stopping criterion for KP-SVM. From this table we obtain that the proposed approaches outperform the alternative methods in terms of classification performance for a given number of selected features, while KP-SVM is particularly effective for high-dimensional data sets, such as CRMA, obtaining significantly better results for a small number of attributes. For the method KP-SVM, convergence is achieved in 25 iterations for WBC and 75 iterations for CRMA. Therefore, this method is more efficient than backward approaches, since the number of iterations to reach convergence is smaller than the number of original variables.

**Table 1.** Comparison of four embedded methods for SVM

|  | $n$ | FSV | RFE-SVM | HO-SVM | KP-SVM |
|---|---|---|---|---|---|
| WBC | 12 | 94.70±1.3 | 95.47±1.1 | **97.69±0.9** | * |
| CRMA | 100 | 91.17±6.7 | 95.61±5.4 | **96.36±5.3** | * |
| WBC | 15 | 95.23±1.1 | 95.25±1.0 | * | **97.55±0.9** |
| CRMA | 20 | 92.03±7.7 | 92.52±7.2 | * | **96.57±5.6** |

## 5 Conclusions and Future Challenges

In this paper we present two embedded methods for feature selection using SVM. A comparison with other embedded techniques shows the advantages of our approach in terms of effectiveness and dimensionality reduction. We also present three different challenges regarding the future of feature selection. The first trend is the importance of considering the process of model selection as a whole, including both feature selection and hyperparameter setting [4]. Several embedded methods attempt to establish a ranking of features from a training set, being necessary a second step that finally leads to the intended model, defining the adequate number of ranked variables. This second step, usually done via cross-validation, is both time consuming and may lead to overfitting, especially when the feature ranking is done using non-linear functions. Both methods presented, HO-SVM and KP-SVM, performs the model selection as a whole, determining the selected number of features in the same algorithm. Alternative approaches, such as FSV and SVM-RFE, require the mentioned additional step, and can be compared with the proposed approaches only using their stopping criterion.

The second presented trend is the increasing size of the data sets, making too complex methods less tractable for large scale pattern recognition. The main benefit of KP-SVM is that we can reach convergence in a small number of iterations, even if the number of variables is very high, making it computationally less intensive. An additional advantage is that ranking methods based on greedy search present difficulties when data sets are high dimensional.

The third and last presented trend is the extension to highly imbalanced data sets, a very relevant topic in pattern recognition. The proposed approaches can be easily adapted to this task. For example, HO-SVM may consider a cost function $C_{(-p)}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ instead of the number of errors, establishing asymmetric costs for the Type I and Type II errors. For KP-SVM, the proposed formulation could consider different costs of errors by penalizing the vector $\xi$ differently, depending on the label of the instance $i$. As future work, we consider the implementation of these models to compensate for the undesired effects caused by imbalanced data sets in model construction; an issue which occurs for example in the domains of Spam filtering, microarray analysis and fraud detection.

**Acknowledgments.** Support from the Chilean "Instituto Sistemas Complejos de Ingeniería" (ICM: P-05-004-F, CONICYT: FBO16) is greatly acknowledged (www.sistemasdeingenieria.cl).

## References

1. Bradley, P., Mangasarian, O.: Feature selection vía concave minimization and support vector machines. In: Int. Conference on Machine Learning, pp. 82–90 (1998)
2. Canu, S., Grandvalet, Y.: Adaptive scaling for feature selection in SVMs. In: Advances in NIPS, vol. 15, pp. 553–560. MIT Press, Cambridge (2002)
3. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature extraction, foundations and applications. Springer, Berlin (2006)
4. Guyon, I., Saffari, A., Dror, G., Cawley, G.: Model selection: Beyond the Bayesian frequentist divide. JMLR 11, 61–87 (2009)
5. Maldonado, S., Weber, R.: A wrapper method for feature selection using Support Vector Machines. Information Sciences 179(13), 2208–2217 (2009)
6. Maldonado, S., Weber, R., Basak, J.: Kernel-Penalized SVM for Feature Selection. Information Sciences 181(1), 115–128 (2011)
7. Neumann, J., Schnörr, C., Steidl, G.: Combined SVM-Based Feature Selection and Classification. Machine Learning 61(1-3), 129–150 (2005)
8. Perkins, S., Lacker, K., Theiler, J.: Grafting: Fast incremental feature selection by gradient descent in function space. JMLR 3, 1333–1356 (2003)
9. Rakotomamonjy, A.: Variable Selection Using SVM-based Criteria. JMLR 3, 1357–1370 (2003)
10. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
11. Weston, J., Mukherjee, S., Chapelle, O., Ponntil, M., Poggio, T., Vapnik, V.: Feature selection for SVMs. In: Advances in NIPS, vol. 13. MIT Press, Cambridge (2001)
12. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: The use of zero-norm with linear models and kernel methods. JMLR 3, 1439–1461 (2003)