Computer Vision, Graphics, and Pattern Recognition Group
Department of Mathematics and Computer Science
University of Mannheim
D-68131 Mannheim, Germany

Reihe Informatik
10/2001

Efficient Feature Subset Selection for Support Vector Machines

Matthias Heiler, Daniel Cremers, Christoph Schnörr

Technical Report 21/2001
Computer Science Series
October 2001

Efficient Feature Subset Selection for Support Vector Machines

Matthias Heiler, Daniel Cremers, Christoph Schnörr

Computer Vision, Graphics, and Pattern Recognition Group
Department of Mathematics and Computer Science
University of Mannheim, 68131 Mannheim, Germany

heiler@uni-mannheim.de
http://www.ti.uni-mannheim.de/∼bmg

Abstract

Support vector machines can be regarded as algorithms for compressing information about class membership into a few support vectors with clear geometric interpretation. It is tempting to use this compressed information to select the most relevant input features. In this paper we present a method for doing so and provide evidence that it selects high-quality feature sets at a fraction of the costs of classical methods.

Keywords: support vector machine, feature subset selection, wrapper method

1 Introduction

The feature subset selection problem is an old problem studied in machine learning, statistics and pattern recognition [1]. For classification purposes, the problem can be stated as follows: Given a data set with features $X_1, X_2, \ldots, X_n$ and labels Y, select a feature subset such that a machine learning algorithm trained on it achieves good performance.

John et al. helped structuring the field by distinguishing *filter methods*, which select feature subsets based on general criteria independent of any specific learning algorithm, from *wrapper methods*, which tailor feature subsets to suite the inductive bias of a given learning algorithm [2]. The wrapper method treats feature selection as a search problem in the space of all possible feature subsets. It is well–known that exhaustive search through all possible feature subsets is the only way for selecting the *optimal* features [3, 4]. However, when there are n features this space has obviously 2^n elements which is generally too large to be searched exhaustively. Thus, numerous heuristic search algorithms have been proposed for determining a suboptimal feature subset in a computationally efficient way (e.g., [1, 5, 6]).

In this paper, we focus on a specific learning algorithm for classification, the support vector machine. In this context, application of the wrapper method has one severe disadvantage: It can be computationally expensive. This is due to the fact that to assess the quality of each feature subset the machine learning algorithm must be trained

and evaluated on it. Unfortunately, training SVMs can be slow rendering the wrapper method a costly procedure for feature selection, especially on large multiclass data sets.

To overcome this difficulty, we present a novel strategy for feature subset selection which is directly based on the support vector architecture and the representation of decision functions in terms of support vectors. The general idea is to train a support vector machine once on a data set containing all features, extract some relevance measure from the trained machine, and use this information to lead a hill-climbing search directly toward a good feature subset. Since the number of reiterations of the training procedure increases only linearly with the number of selected features, this algorithm can be orders of magnitudes faster than the wrapper method. Furthermore, we show that this computational efficiency can be obtained without sacrificing classification accuracy.

After completion of this work [7], the authors became aware of similar ideas reported in [8]. Whereas the latter work is applied in the context of visual object recognition [9], we focus directly on the feature selection problem and present here for the first time extensive numerical results which reveal the performance of our approach for established benchmark data sets [10].

2 SVM-based Feature Selection

Let us motivate the feature selection algorithm with a simple example: Assume we are given a two dimensional binary classification problem where only one input feature is relevant for classification. The other input feature contains noise. We train a SVM with linear kernel on this problem and find a separating hyperplane with maximal margin (Figure 1).

Now the key observation is that the normal $\vec{w}$ of the separating hyperplane will point in the relevant direction, i.e., it will be approximately colinear with the basis vector $\vec{e}_i$ that is used for the relevant feature and approximately orthogonal to the other one. This holds for any n-dimensional SVM with linear kernel: If we take away all the basis vectors which are orthogonal to $\vec{w}$ we will loose no information about class membership as the corresponding features have no influence on the SVM decision. Accordingly, we can define the importance of each feature X_k by its amount of colinearity with $\vec{w}$:

$$d_k = (\langle \vec{w}, \vec{e}_k \rangle)^2. \tag{1}$$

In the nonlinear case the SVM decision function [11] reads

$$f(\vec{x}) = \sum_i y_i \alpha_i \langle \phi(\vec{x}), \phi(\vec{x}_i) \rangle + b, \tag{2}$$

with a set of given training vectors $\vec{x}_i$, corresponding class labels y_i, Lagrange multipliers α_i associated with the SVM optimization problem, and some offset from the origin b. As the nonlinear mapping ϕ appears only inside the scalar product $\langle \phi(\vec{x}), \phi(\vec{x}_i) \rangle$ it is usually expressed in terms of a kernel function.

Compared to the linear case the influence of a feature X_k on the decision boundary is no longer independent from the other features. It varies depending on where in the

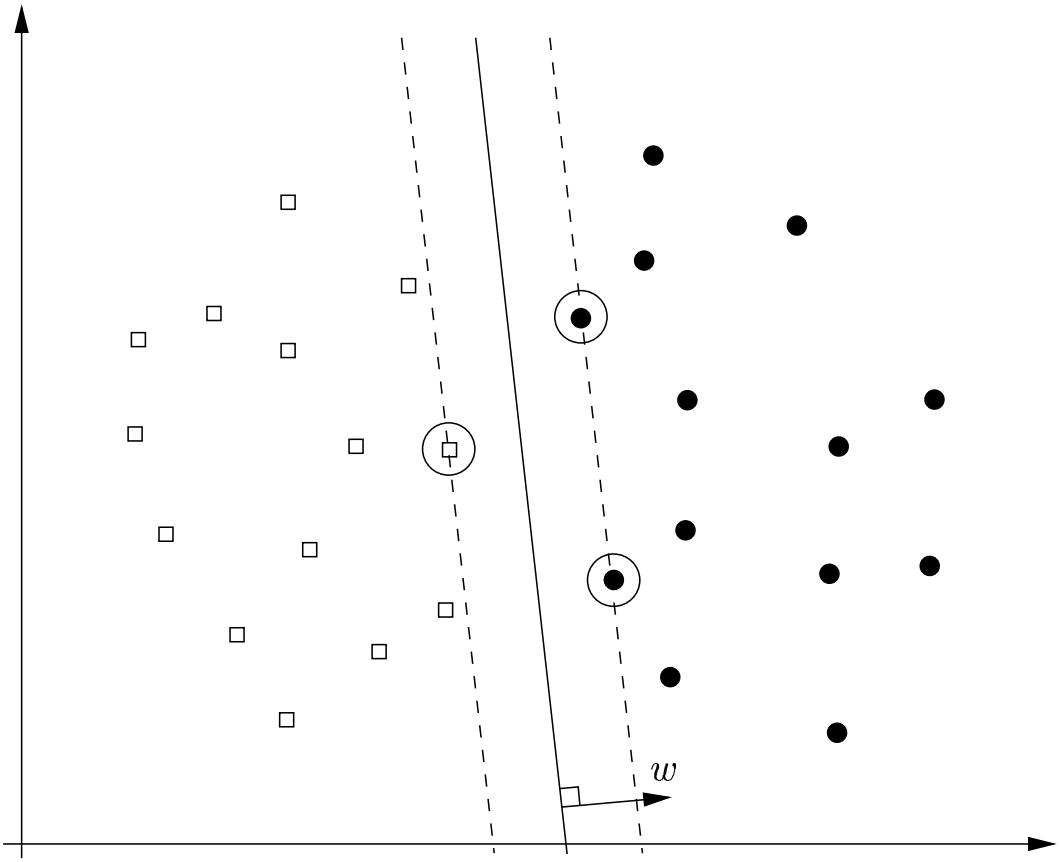

Figure 1: Separating hyperplane for feature selection. Circles indicate the support vectors.

input space it is determined. However, for a given point in input space, $\vec{x}$, we can define the influence of X_k as the squared partial derivative with respect to X_k:

$$d_{k,\vec{x}} = (\langle \nabla f(\vec{x}), \vec{e}_k \rangle)^2, \tag{3}$$

Note that for SVMs with linear kernel this definition reduces to the measure of colinearity defined in (1).

In order to evaluate (3), we have to select meaningful points $\vec{x}$. To this end, we can take advantage of the information-compressing capabilities of the support vector machine: The SVM decision function (2) essentially is linear in the mapped input vectors $\phi(\vec{x}_i)$. More precisely, it is linear in the mapped input vectors for which the SVM optimization process yields non-zero Lagrange parameters $\alpha_i > 0$. These input vectors are the support vectors $SV = \{\vec{x}_i : \alpha_i > 0\}$, and in practice their number is often small compared to the number of all input vectors [11, p. 135]. It is clear that the features which have little influence on the support vectors also have small effect on the SVM decision. Thus, a good measure of the importance of a feature is its average influence

evaluated at the support vectors:

$$d_k = \frac{1}{|SV|} \sum_{\vec{x}_i \in SV} \frac{d_{k,\vec{x}_i}}{\sum_k d_{k,\vec{x}_i}}. \tag{4}$$

Note that the denominator $\sum_k d_{k,\vec{x}}$ ensures that each support vector's contribution sums to one. This is to avoid that outliers and support vectors located at very narrow margins dominate the overall result too much. Or, equivalently, it increases the influence of support vectors at clear, well-separated margins.

Once we have calculated the importance measure $\{d_k\}_{k=1\ldots n}$ we use a simple hill-climbing search to determine the optimal number of relevant features. Specifically, we rank the features according to their d_k values and, starting from an empty feature set, subsequently add features with highest rank until the accuracy of a SVM trained on the selected features stops increasing.

3 Experiments

To evaluate the performance of the SVM feature selection method we ran it on a number of data sets from the UCI machine learning repository [10]. For comparison, we also ran the wrapper method with hill-climbing search. Note that overfitting is a general problem with feature selection on small data sets [12, 13]. We tried to avoid it by using 10-fold crossvalidation during the feature selection process, as well as on a completely separate test set for assessing the quality of the selected features. For the hill-climbing search we used the stopping criterion proposed by Kohavi et al. [12]. For all experiments we employed the LIBSVM package with default parameters set [14].

Table 1 summarizes the results of our experiments. For each of the data sets examined we could reduce the number of features used for classification without sacrificing accuracy. More precisely speaking, a one-tailed t-test revealed at the 5% level no statistically significant decrease in classification performance after feature selection. On the contrary, for the chess, the led and the optidigit data sets we found our feature selection to significantly increase classification accuracy. Note that the t-test revealed no difference in performance between the wrapper and the SVM-based selection method.

The led data benefitted very much from feature selection. This is not surprising as led is a synthetic data set consisting of 7 relevant features and 17 features containing random noise. Both feature selection algorithms reliably extract the 7 relevant features, however some irrelevant features, which appear to be predictive on the given data, are also included. As this data set contains binary features only, we could easily estimate the Shannon entropy h, compute the information gain $h(y) - h(y|X_k)$ for each individual feature X_k [15], and compare it to our relevance measure d_k. Figure 2 visualizes both relevance measures: The strong correlation ($r = 0.99$) between them is apparent as well as the clear distinction between the 7 relevant features on the left and the random features on the right. Note that for experiments comprising continuous features computing the information gain is often not straightforward while evaluating the SVM-based relevance measure is.

Table 1: Performance of different feature selection methods on a test set. Feature selection significantly reduces the number of features without sacrificing classification accuracy.

	No Selection		Wrapper		SVM Selection	
Data set	Features	Accuracy	Features	Accuracy	Features	Accuracy
breast cancer	9	95.71	5	95.42	6	95.71
chess	35	**33.33**	5	**86.67**	4	**86.67**
crx	43	83.73	5	85.15	5	86.03
diabetes	8	74.13	4	75.19	5	74.91
led	24	**66.27**	11	**73.10**	10	**74.70**
mfeat	649	97.60	n.a.	n.a.	31	97.50
glass	9	60.63	4	61.36	5	60.54
mushroom	125	99.95	8	99.95	14	100.00
optidigit	64	**97.68**	36	**98.32**	36	**98.39**

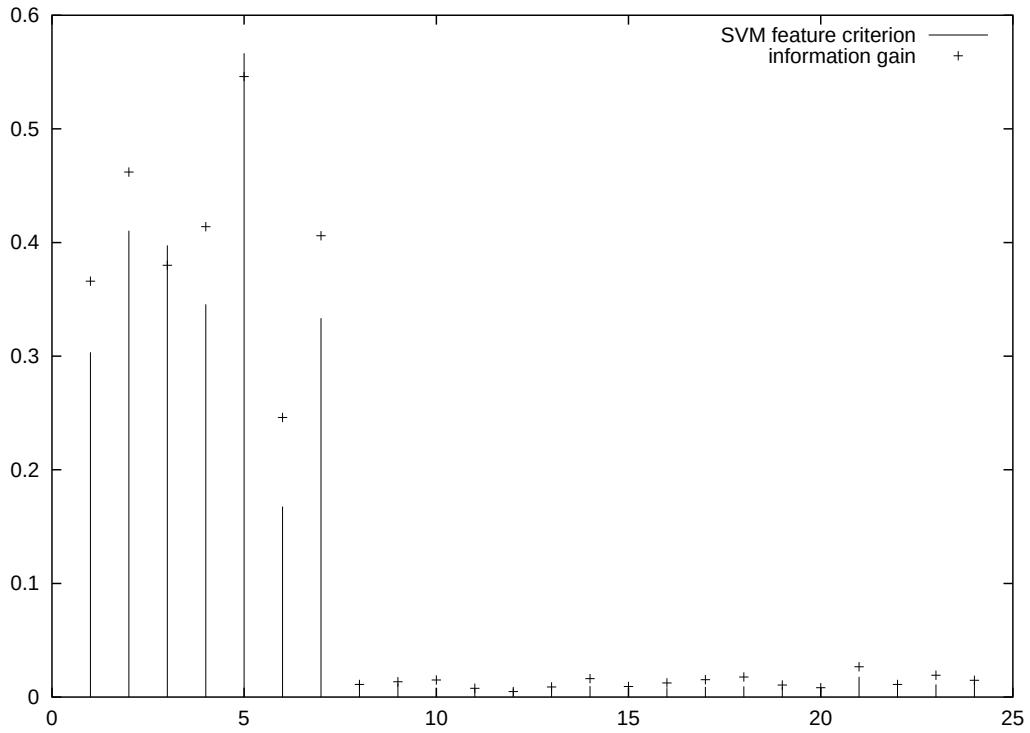

Figure 2: Feature selection on the led data set.

Table 2: CPU time used for the feature selection process. SVM selection is much faster than the wrapper method as less classifiers must be trained.

Data set	Wrapper	SVM Selection
breast cancer	19.29	4.073
chess	45.03	1.48
crx	355.74	17.92
diabetes	45.15	19.35
glass	13.87	3.46
led	253.25	23.04
mfeat	n.a.	59162.13
mushroom	27518.39	2977.34
optidigit	77742.152	2843.34

Our results show that the wrapper and the SVM method selected features of equal quality in all cases examined. Consequently, it is interesting to compare the methods in terms of speed. Table 2 shows the CPU time in seconds used by each method for the different data sets. We can see that especially for the larger data sets the SVM-based feature selection has a clear advantage over the wrapper. This is not surprising as the wrapper needs to train a larger number of SVMs. Specifically, to select a subset of d features from a set of n given features the wrapper methods examines $(d^2 + d(2r + 1))/2$ SVMs, where $r = (n - d)$ denotes the number of removed features, while the method we propose examines $(d + 1)$ SVMs only – one for computing the relevance measure d_k and r during hill-climbing. Thus, incorporating the information collected from the SVM reduces the run-time complexity from quadratic to linear.

4 Conclusion

We propose a method that utilizes the information-compressing capabilities of the support vector machine for feature selection. It is easy to understand, simple to implement, fast to execute, and it performs as accurately as the wrapper method on a number of real-world data sets.

Software

For our experiments we used the LIBSVM package by [14].

References

[1] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall International, 1982.

[2] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of ICML-94, 11th International Conference on Machine Learning*, pages 121–129, New Brunswick, NJ, 1994.

[3] T. Cover and J. van Campenhout. On the possible orderings in the measurement selection problem. *IEEE Trans. on Systems, Man, and Cybernetics*, 7:657–661, 1977.

[4] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.

[5] Ron Kohavi and George H. John. Wrappers for feature subset selection. In *Proceedings of the Tenth International Conference on Computational Learning Theory*, pages 245–271, 1997.

[6] P. Somol, P. Pudil, J. Novovicova, and P. Paclik. Adaptive floating search methods in feature selection. *Patt. Recog. Letters*, 20(11):1157–1163, 1999.

[7] Matthias Heiler. Optimization criteria and learning algorithms for large margin classifiers. Master's thesis, University of Mannheim, Germany, Department of Mathematics and Computer Science, Computer Vision, Graphics, and Pattern Recognition Group, D-68131 Mannheim, Germany, 2001.

[8] Theodoros Evgeniou. *Learning with kernel machine architectures*. PhD thesis, Massachusetts Institute of Technology, 6 2000.

[9] T. Evgeniou, M. Pontil, C. Papageorgiou, and T. Poggio. Image representations for object detection using kernel classifiers. *Asian Conference on Computer Vision*, pages 687–692, 2000.

[10] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[11] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.

[12] Ron Kohavi and Sommerfield Dan. Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In Usama M Fayyad and Ramasamy Uthurusamy, editors, *First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, 1995.

[13] T. Scheffer and R. Herbrich. Unbiased assessment of learning algorithms. In *IJCAI-97*, pages 798–803, 1997.

[14] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[15] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.