The Support Feature Machine for Classifying with the Least Number of Features

Sascha Klement and Thomas Martinetz

Institute for Neuro- and Bioinformatics, University of Lübeck

Abstract. We propose the so-called Support Feature Machine (SFM) as a novel approach to feature selection for classification, based on minimisation of the zero norm of a separating hyperplane. Thus, a classifier with inherent feature selection capabilities is obtained within a single training run. Results on toy examples demonstrate that this method is able to identify relevant features very effectively.

Key words: Support feature machine, feature selection, zero norm minimisation, classification.

1 Introduction

The ever increasing complexity of real-world machine learning tasks requires more and more sophisticated methods to deal with datasets that contain only very few relevant features but many irrelevant noise dimensions. It is well-known that these irrelevant features will distract state-of-the-art methods, such as the support vector machine. Thus, feature selection is often a fundamental preprocessing step to achieve proper classification results, to improve runtime, and to make the training results more interpretable.

For many machine learning tasks, maximum margin methods have been confirmed to be a good choice to maximise the generalisation performance [1]. But, besides generalisation capabilities, other aspects, such as fast convergence, existence of simple error bounds, straightforward implementation, running time requirements, or numerical stability, may be equally important.

In recent years, as complexity and dimensionality of real-world problems have dramatically increased, two other aspects have gained more and more importance. These are sparsity and domain interpretability of the inference model. Both are closely connected to the task of variable or feature selection. Primarily, feature selection aims to improve or at least preserve the discriminative capabilities when using fewer features than the original classifier, regression or density estimator. In the following, we focus on feature selection for classification tasks.

Feature selection as an exhaustive search problem is in general computationally intractable as the number of states in the search space increases exponentially with the number of features. Therefore, all computationally feasible feature selection techniques try to approximate the optimal feature set, e.g. by Bayesian inference, gradient descent, genetic algorithms, or various numerical optimisation methods.

Commonly, these methods are divided into two classes: filter and wrapper methods. Filter methods completely separate the feature selection and the classification task [2]. The optimal feature subset is selected in advance, i.e. filtered out from the overall set of features without assessing the actual classifier. In practise, one could, for example, select those features with the largest Pearson correlation coefficients or Fisher scores before training the classifier.

Wrapper methods make use of the induction algorithm to assess the prediction accuracy of a particular feature subset. Well-known contributions to this class of feature selection algorithms are those of Weston et al. [3], who select those features that minimise bounds on the leave-one-out error, and Guyon et al. [4], who propose the so-called recursive feature elimination. Some types of support vector machines already comprise feature selection to some extend, such as the l_1-norm SVM [5] or the VS-SSVM (Variable Selection via Sparse SVMs) [6].

In the following, we propose the so-called Support Feature Machine (SFM) as a novel method for feature selection that is both simple and fast. To assess its performance, we will measure and discuss various aspects of feature selection methods, such as improvements to the test error when using only the selected features, sparsity of the solution, or the ability to identify relevant and irrelevant features.

The following sections are organised as follows. First, we briefly introduce the problem of finding relevant variables by means of zero norm minimisation. This leads to our contribution, the mathematical definition of the SFM. Using artificial linearly separable datasets, we illustrate various aspects of the SFM and compare the results to other feature selection methods.

We conclude with a critical discussion of the achievements and propose further extensions to the SFM.

2 Feature Selection by Zero Norm Minimisation

We make use of the common notations used in classification and feature selection frameworks, i.e. the training set

$$\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$$

consists of feature vectors $\boldsymbol{x}_i \in \mathbb{R}^d$ and corresponding class labels $y_i \in \{-1, +1\}$. First, we assume the dataset $\mathcal{D}$ to be linearly separable, i.e.

$$\exists\, \boldsymbol{w} \in \mathbb{R}^d,\, b \in \mathbb{R} \quad \text{with} \quad y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 0 \;\forall\, i \quad \text{and} \quad \boldsymbol{w} \neq \boldsymbol{0}\,, \tag{1}$$

where the normal vector $\boldsymbol{w} \in \mathbb{R}^d$ and the bias $b \in \mathbb{R}$ describe the separating hyperplane except for a constant factor. Obviously, if $\boldsymbol{w}$ and b are solutions to the inequalities, also $\lambda\, \boldsymbol{w}$ and $\lambda\, b$ solve them with $\lambda \in \mathbb{R}^+$.

In general, there is no unique solution to (1). Our goal is to find a weight vector $\boldsymbol{w}$ and a bias b which solve

$$\text{minimise } \|\boldsymbol{w}\|_0^0 \quad \text{subject to} \quad y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 0 \quad \text{and} \quad \boldsymbol{w} \neq \boldsymbol{0} \tag{2}$$

with $\|\boldsymbol{w}\|_0^0 = \text{card}\{w_i | w_i \neq 0\}$. Hence, solutions to (2) solve the classification problem (1) using the least number of features. Note, that any solution can be multiplied by a positive factor and is still a solution. Weston et al. [7] proposed to solve the above problem with a variant of the Support Vector Machine by

$$\text{minimising } \|\boldsymbol{w}\|_0^0 \quad \text{subject to} \quad y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 1\,. \tag{3}$$

Indeed, as long as there exists a solution to (2) for which $y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) > 0$ is valid for all $i = 1, ..., n$, solving (3) yields a solution to (2). Unfortunately, (2) as well as (3) are NP-hard and cannot be solved in polynomial time. Therefore, Weston et al. [7] proposed to approximate (3) by solving

$$\text{minimise} \quad \sum_{j=1}^{d} \ln\left(\epsilon + |w_j|\right) \quad \text{subject to} \quad y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 1 \tag{4}$$

with $0 < \epsilon \ll 1$. They showed that if $\boldsymbol{w}_0$ and $\boldsymbol{w}^*$ optimise (3) and (4), respectively, then

$$\|\boldsymbol{w}^*\|_0^0 \leq \|\boldsymbol{w}_0\|_0^0 + \mathcal{O}\left(\frac{1}{\ln \epsilon}\right)\,. \tag{5}$$

They also showed that using the following iterative scheme at least a local minimum of (4) is found:

1. Set $\boldsymbol{z} = (1, \ldots, 1)$.
2. Minimise $|\boldsymbol{w}|$ such that $y_i \left(\boldsymbol{w}^{\mathrm{T}} (\boldsymbol{x}_i \cdot \boldsymbol{z}) + b\right) \geq 1$.
3. Set $\boldsymbol{z} = \boldsymbol{z} \cdot \boldsymbol{w}$.
4. Repeat until convergence.

This iterative scheme simply applies linear programming.

2.1 Support Feature Machine

Instead of modifying the SVM setting as in (3), we slightly change (2) such that we

$$\text{minimise } \|\boldsymbol{w}\|_0^0 \quad \text{subject to} \quad y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 0 \quad \text{and} \quad \boldsymbol{w}^{\mathrm{T}} \boldsymbol{u} + \bar{y} b = 1 \tag{6}$$

with $\boldsymbol{u} = \frac{1}{n} \sum_{i=1}^{n} y_i \boldsymbol{x}_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$. The second constraint excludes $\boldsymbol{w} = \boldsymbol{0}$, since otherwise we would obtain $\bar{y} b = 1$ and $y_i b \geq 0$, which cannot be fulfilled for all i (we have labels $+1$ and -1). As long as there is a solution to (2) with $y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) > 0$ for at least one $i \in \{1, ..., n\}$, also $\sum_{i=1}^{n} y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) > 0$ is satisfied. Hence, solving (6) yields a solution to the ultimate problem (2).

Since we have linear constraints, for solving (6) we can employ the same framework Weston et al. [7] used for solving their problem. Also (5) applies. However, our experiments show that by solving

$$\text{minimise} \sum_{j=1}^{d} \ln\left(\epsilon + |w_j|\right) \quad \text{subject to} \quad y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 0 \quad \text{and} \quad \boldsymbol{w}^{\mathrm{T}} \boldsymbol{u} + \bar{y} b = 1$$

with the iterative scheme

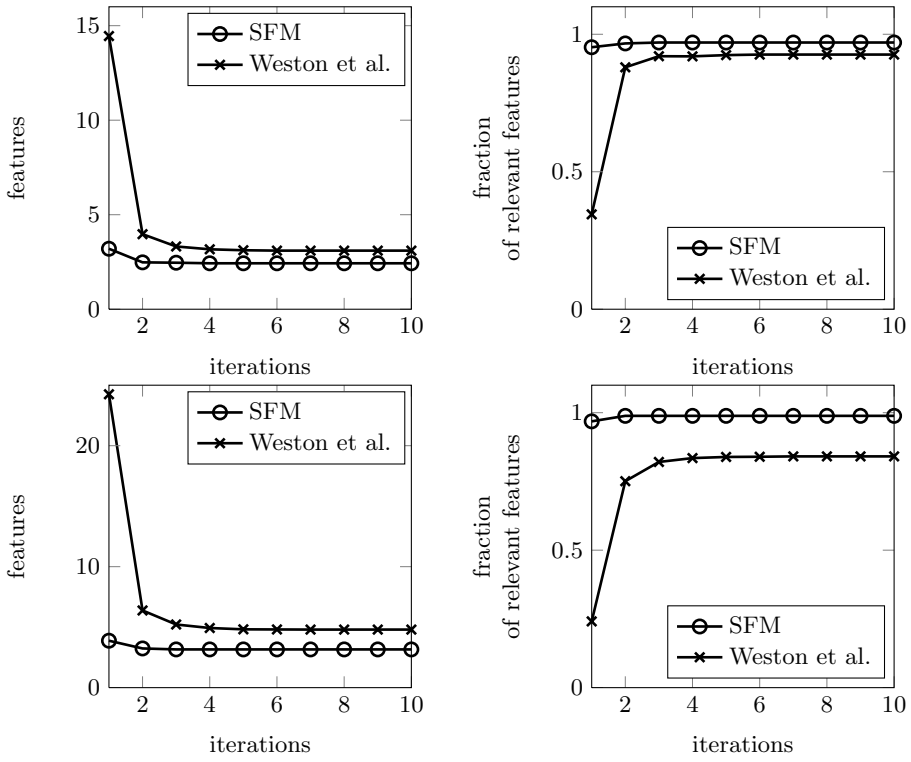

Fig. 1. Comparison of the SFM and the method proposed by Weston et al. The top row shows results for $n = 50$ data points, the bottom row for $n = 200$ data points (averaged over 100 runs).

1. Set $\boldsymbol{z} = (1, \ldots, 1)$.
2. Minimise $|\boldsymbol{w}|$ such that $y_i \left(\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_i \cdot \boldsymbol{z}) + b\right) \geq 0$ and $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{u} + \bar{y}b = 1$.
3. Set $\boldsymbol{z} = \boldsymbol{z} \cdot \boldsymbol{w}$.
4. Repeat until convergence

we obtain significantly better solutions to the ultimate problem then by solving (4). It seems that the new cost function is much less prone to local minima.

2.2 Experiments

For learning tasks, such as classification or regression, one normally assesses a method's performance via the k-fold cross-validation error, or via the test error on a separate dataset. For feature selection, besides the test error, also the number of selected features and the amount of truly relevant features are important. Since in real-world scenarios these values are almost never known, we used artificial examples to compare the results of the SFM and the method proposed by Weston. The toy examples were constructed according to Weston

et al. [7], i.e. the input data consist of 6 relevant but redundant features and 196 noise dimensions. Additionally, we required the datasets to be separable within the 6 relevant dimensions. Figure 1 shows the results for 100 independent runs using $n = 50$ and $n = 200$ data points. Apparently, the SFM returns both a lower total number of features and a higher percentage of truly relevant features. The convergence speed is also slightly better, and already after one iteration the SFM solution is quite close to the final solution.

Next, we evaluated the generalisation performance of the SFM. Table 1 shows mean and standard deviations in comparison to the SVM without feature selection and to the method proposed by Weston et al. For each method and training set size, the experiment was repeated 100 times. Within each repetition 10000 data points were sampled (6 relevant, 196 noise dimensions), n data points were used for training ($n = 20, 50, 100, 200, 500$) and the remaining for evaluating the test error. Again, only linearly separable training datasets were allowed. Obviously, the SFM significantly outperforms a standard SVM approach, but is slightly worse than Weston's method.

Table 1. Mean and standard deviation of the test error using different methods and training set sizes for the toy example. The methods are: Standard hard-margin Support Vector Machine (SVM), the method proposed by Weston et al. (Weston) and the Support Feature Machine (SFM).

n	SVM	Weston	SFM
20	28.8% (± 2.2%)	8.9% (± 8.0%)	17.5% (± 7.8%)
50	19.0% (± 1.9%)	2.7% (± 1.5%)	6.6% (± 3.7%)
100	12.2% (± 1.5%)	1.7% (± 0.7%)	3.8% (± 1.7%)
200	6.7% (± 0.9%)	1.2% (± 0.5%)	2.1% (± 0.9%)
500	3.1% (± 0.5%)	0.8% (± 0.2%)	1.1% (± 0.4%)

2.3 Implementation Issues

As with many machine learning algorithms, normalisation is an essential preprocessing step also for the SFM. For all experiments, we normalised the training datasets to zero mean and unit variance and finally scaled all vectors to have a mean norm of one. This last step is necessary in high-dimensional scenarios to keep the outcome of scalar products in a reasonable range. The test sets were normalised according to the factors obtained from the corresponding training sets.

For solving the optimisation problems, we used the MOSEK optimisation software. To avoid numerical issues, numbers that differed by no more than a specific implementation-dependent number — normally closely connected to the machine epsilon — were considered to be equal.

3 Conclusions

We proposed a novel method for combined feature selection and classification — the so-called Support Feature Machine. Experiments on artificial as well as real-world datasets demonstrated that the SFM can identify relevant features very effectively and may improve the generalisation performance significantly with respect to an SVM without feature selection. The implementation only requires linear programming solvers and may therefore be established in various programming environments.

So far, we focused on linear classifiers, mostly for high-dimensional low-sample size scenarios because these scenarios seem to be the most relevant ones in practical applications, such as the analysis of microarray datasets.

In some scenarios, it is necessary to allow for nonlinear classification to achieve proper classification performance. One might think of ways to incorporate kernels into the SFM to allow for arbitrary class boundaries. Nevertheless, the main focus of the SFM was to provide results that may easily be interpreted both in terms of feature selection and classification, so nonlinearities would slacken this demand.

In total, the results we obtained using the SFM approach are quite promising, however, we need to justify our results on real-world datasets. In a follow-up paper, we will show, that even an exponentially increasing number of irrelevant features does not significantly reduce the performance of the SFM. Additionally, we will extend the standard SFM approach to non-separable scenarios. Further work will include experiments on more challenging real-world scenarios with practical relevance. Finally, we seek for an iterative optimisation method to be independent from proprietary optimisation toolboxes.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
2. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence (1997) 273–323
3. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: Feature Selection for SVMs. In: Advances in Neural Information Processing Systems. (2000)
4. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning **46** (2002) 389–422
5. Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-norm support vector machines. In Thrun, S., Saul, L., Schölkopf, B., eds.: Advances in Neural Information Processing Systems 16, Cambridge, MA, MIT Press (2004)
6. Bi, J., Bennett, K.P., Embrechts, M., Breneman, C.M., Song, M.: Dimensionality Reduction via Sparse Support Vector Machines. Journal of Machine Learning Research **3** (2003) 1229–1243
7. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of the Zero-Norm with Linear Models and Kernel Methods. Journal of Machine Learning Research **3** (2003) 1439–1461