The Support Feature Machine: Classification with the Least Number of Features and Application to Neuroimaging Data

Sascha Klement
klement@inb.uni-luebeck.de
Institute for Neuro- and Bioinformatics, University of Lübeck, Lübeck 23562, Germany

Silke Anders
silke.anders@neuro.uni-luebeck.de
Department of Neurology, University of Lübeck, Lübeck 23562, Germany

Thomas Martinetz
martinetz@inb.uni-luebeck.de
Institute for Neuro- and Bioinformatics, University of Lübeck, Lübeck 23562, Germany

By minimizing the zero-norm of the separating hyperplane, the support feature machine (SFM) finds the smallest subspace (the least number of features) of a data set such that within this subspace, two classes are linearly separable without error. This way, the dimensionality of the data is more efficiently reduced than with support vector–based feature selection, which can be shown both theoretically and empirically. In this letter, we first provide a new formulation of the previously introduced concept of the SFM. With this new formulation, classification of unbalanced and nonseparable data is straightforward, which allows using the SFM for feature selection and classification in a large variety of different scenarios. To illustrate how the SFM can be used to identify both the smallest subset of discriminative features and the total number of informative features in biological data sets we apply repetitive feature selection based on the SFM to a functional magnetic resonance imaging data set. We suggest that these capabilities qualify the SFM as a universal method for feature selection, especially for high-dimensional small-sample-size data sets that often occur in biological and medical applications.

1 Introduction

Recent developments of massively parallel data acquisition systems call for sophisticated analysis methods that are capable of dealing with data sets that have many dimensions but comprise few samples. Such high-dimensional small-sample-size scenarios occur especially in biological and medical applications that rely on human data, such as tissue classification based on microarray gene data (Golub et al., 1999; Lockhart & Winzeler, 2000), identification of disease-specific genome mutations (Samani et al., 2007; McPherson et al., 2007; Raelson et al., 2007), or multivoxel pattern analysis (MVPA) in neuroimaging (Haynes, 2011). In these applications, theoretical and practical issues such as generalization bounds, run time, or memory footprint considerations require dedicated methods for the three basic machine learning tasks: classification, regression, and density estimation.

Maximum margin methods such as the support vector machine (Vapnik, 1999) have been shown to be a good choice for many classification problems in biological applications due to their excellent generalization capabilities. However, these methods may fail especially in high-dimensional small-sample-size scenarios (Weston et al., 2000). Moreover, in biological and medical applications, the primary goal is often not to achieve high prediction accuracy but to identify informative features. Thus, feature selection is not only needed to improve run time and achieve proper prediction results, but also to allow meaningful inferences about biologically significant features. Such feature selection methods have been proposed in a variety of flavors (for an overview, see Guyon & Elisseeff, 2003). However, these approaches have seldom addressed the question how the smallest subset of features that is required to solve a classification problem can be obtained.

Recently we proposed the support feature machine (SFM) as a novel method for feature selection that aims to identify the smallest subset of features that separates two classes (Klement & Martinetz, 2010a, 2010b, 2011). The basic idea of the SFM is to find the smallest subspace (the least number of features) in a data set such that within this subspace two classes are linearly separable without error. By minimizing the zero norm of the separating hyperplane, a classifier based on a minimal set of features is obtained within a single training run. In contrast to most feature selection methods that are conservative in the way they select features—they keep all features that might ever be relevant for classification and discard only those features that are irrelevant with high probability—the SFM is aggressive in discarding features and keeps only those that are definitely required to separate classes in the training data. Results on artificial data as well as real-world data show that the SFM identifies truly relevant features very effectively and in many cases more accurately than SVM-based feature selection (Klement & Martinetz, 2010a, 2010b).

We suggest that this feature makes the SFM a powerful tool not only to identify the most informative features within a data set but also to estimate the total amount of informative features of this data set. If the SFM is used for repetitive feature selection (RFS)—such that in each repetition, all features returned by the SFM are discarded from the data set and in the next repetition the SFM is trained on the remaining features only—then the percentage of features discarded from the data set before the test error reaches chance level might provide an estimate of the total number of informative features within this data set. Importantly, the accuracy of this estimate strongly depends on the selectivity of the applied feature selection method: if irrelevant (uninformative) features are falsely identified as being relevant, then the number of informative features in the data set is overestimated. In contrast, the estimate does not strongly rely on the sensitivity of feature selection: even if only one (truly relevant) feature is obtained in each repetition, the number of informative voxels will still be estimated correctly. Since the SFM is very conservative in the way it identifies relevant features, SFM-based RFS will likely provide an unbiased estimate of the true number of informative features of a data set.

The remainder of this letter is organized in three sections. In section 2, we review and reformulate the concept of the SFM. This new formulation now permits classification of unbalanced and nonseparable classes, which enables the SFM to be used for feature selection and classification in a large variety of different scenarios.[1] This is shown on artificial data in section 3. Finally, in section 4, we apply a repetitive version of the SFM to an fMRI (functional magnetic resonance imaging) data set to illustrate how the SFM can be used to identify not only the most informative features but also the total number of informative features in biological data sets. The appendix contains mathematical considerations as well as remarks on implementation alternatives.

Note on notations. In this work, we use lowercase boldface letters (e.g., $\boldsymbol{x}, \boldsymbol{y}$) for vectors and uppercase boldface letters for matrices (e.g., $\boldsymbol{A}$). Sets are typeset in uppercase calligraphic letters (e.g., $\mathcal{D}$).

We make use of the common notations used in classification and feature selection frameworks; a data set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$ consists of feature vectors, samples, or patterns $\boldsymbol{x}_i \in \mathbb{R}^d$ and corresponding class labels $y_i \in \{-1, +1\}$. The dimensionality of a vector is denoted by d, while n refers to the cardinality of the set. For simplicity, we define $\boldsymbol{z}_i = y_i \boldsymbol{x}_i$ and $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$. Further, for each class, we define a separate set of indices, $I^+ = \{i \,|y_i = +1\}$ and $I^- = \{i \,|y_i = -1\}$. The vectors $\mathbf{0}$ and $\mathbf{1}$ are vectors with all their entries being zero or one, respectively. For readability, we omit the length of these vectors where possible. The identity matrix $\boldsymbol{I}_d$ is a square matrix containing ones on the main diagonal and zeros elsewhere, and the zero matrix $\mathbf{0}_{n,d}$ has n rows and d columns all set to zero.

2 The Problem and the Approach

In the terminology of feature selection, the SFM is an embedded method that combines both feature selection and classification within a single

[1] A toolbox with source code and demo applications can be retrieved online from http://www.inb.uni-luebeck.de/tools-demos/support-feature-machine.

framework. The principle of structural risk minimisation is implemented by limiting the family of classification functions to those with the fewest number of parameters—in this case, dimensions or features. In this section, we first review and reformulate the basic concept of the SFM originally proposed in Klement and Martinetz (2010a, 2010b). As we will see, this new formulation of the SFM is more compact and intuitive than the original formulation and can easily be extended to unbalanced and nonseparable data. We conclude this section with some considerations regarding the behavior of the SFM in the limit.

2.1 SVM-Based Feature Selection. To introduce the basic concept of the SFM, we first show how it relates to the SVM-based feature selction proposed by Weston, Elisseeff, Schölkopf, and Tipping (2003). Let us assume that the data set $\mathcal{D}$ is linearly separable,

$$\exists\, \boldsymbol{w} \in \mathbb{R}^d,\ b \in \mathbb{R} \quad \text{with} \quad y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 0\ \forall i \quad \text{and} \quad \boldsymbol{w} \neq \boldsymbol{0}\,, \tag{2.1}$$

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, $\lambda\, \boldsymbol{w}$ and $\lambda\, b$ solve them with $\lambda \in \mathbb{R}^+$. In general, there is no unique solution to equation 2.1. A solution with the least number of features,

$$\begin{aligned} \text{minimizes} \quad & \|\boldsymbol{w}\|_0^0 \\ \text{subject to} \quad & y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 0 \quad \forall i \\ \text{and} \quad & \boldsymbol{w} \neq \boldsymbol{0}, \end{aligned} \tag{2.2}$$

with $\|\boldsymbol{w}\|_0^0 = \mathrm{card}\{w_i | w_i \neq 0\}$. Note that again any solution of equation 2.2 can be multiplied by a positive factor and is still a solution. Weston et al. (2003) proposed solving equation 2.2 with a variant of the SVM by

$$\begin{aligned} \text{minimizing} \quad & \|\boldsymbol{w}\|_0^0 \\ \text{subject to} \quad & y_i \left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 1 \quad \forall i\,. \end{aligned} \tag{2.3}$$

Indeed, as long as there exists a solution to equation 2.2 for which $y_i(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b) > 0$ for all $i = 1, ..., n$, solving equation 2.3 yields a solution to 2.2. Unfortunately, both equations are NP-hard and cannot be solved in polynomial time. Therefore, Weston et al. (2003) proposed to approximate equation 2.3 by

$$\begin{aligned} \text{minimizing} \quad & \sum_{j=1}^{d} \ln\left(\varepsilon + |w_j|\right) \\ \text{subject to} \quad & y_i\left(\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b\right) \geq 1 \quad \forall i \end{aligned} \tag{2.4}$$

with $0 < \varepsilon \ll 1$. If $\boldsymbol{w}_0$ and $\boldsymbol{w}^*$ optimize equations 2.3 and 2.4, respectively, then

$$\left\|\boldsymbol{w}^*\right\|_0^0 \leq \left\|\boldsymbol{w}_0\right\|_0^0 + \mathcal{O}\left(\frac{1}{\ln \varepsilon}\right), \tag{2.5}$$

that is, both solutions coincide as $\varepsilon \to 0$. Thus, by minimizing equation 2.4, an approximate solution to equation 2.3 is found. However, equation 2.4 is not convex, has many local minima, and is still hard to solve. Weston et al. (2003) proposed the following iterative scheme, which finds a local minimum of equation 2.4 by solving a sequence of linear programs:

```
Initialize $\boldsymbol{z} = (1, \ldots, 1)$.
**repeat**
    Minimize $|\boldsymbol{w}|$ such that $y_i\left(\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_i * \boldsymbol{z}) + b\right) \geq 1$
    Update $\boldsymbol{z} = \boldsymbol{z} * \boldsymbol{w}$
**until** *convergence*
```

This modification of the SVM effectively reduces the feature space used for classification. However, this approach does not necessarily return the least number of features required for classification. The number of features may be further reduced by discarding any margin maximization induced by the constraints $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq 1$. This is the basic idea of the SFM.

2.2 The Support Feature Machine: Basic Algorithm. The approach of Weston et al. (2003) performs a mixture of feature selection and margin maximization, which might be conflicting objectives. Taking a different approach, we adapt the definition of linear separability, equation 2.2, slightly, such that we

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{w}\|_0^0 \\ \text{subject to} \quad & y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 0 \quad \forall i \\ \text{and} \quad & \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-\right) = 1 \end{aligned} \tag{2.6}$$

with $\mu^+ = \frac{1}{n^+}\sum_{i\in I^+}\boldsymbol{x}_i$ and $\mu^- = \frac{1}{n^-}\sum_{i\in I^-}\boldsymbol{x}_i$ as the class-specific means. The first constraint is insensitive to any margin. The second constraint excludes the trivial solution $\boldsymbol{w} = \mathbf{0}$. As long as the input data are linearly separable with $y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) > 0$ for at least one $i \in \{1, \ldots, n\}$, we have

$$\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-) = \frac{1}{n^+}\sum_{i\in I^+} y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) + \frac{1}{n^-}\sum_{i\in I^-} y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) > 0, \tag{2.7}$$

and the equality constraint can be satisfied by scaling $\boldsymbol{w}$ and b appropriately. Hence, as long as the input data are linearly separable with $y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) > 0$

for at least one $i \in \{1, ..., n\}$ (not all i as in Westons's approach), solving equation 2.6 yields a solution to the ultimate problem, equation 2.2. Compared to the first version proposed in Klement and Martinetz (2010a, 2010b), the equality constraint is now more compact since the bias b no longer occurs. In this formulation, an extension to unbalanced and nonseparable classes is straightforward.

The framework Weston et al. (2003) used to solve equation 2.4 is also suited to solve equation 2.6 (see Weston et al., 2003). Thus, we

$$\begin{array}{rl} \text{minimize} & \sum_{j=1}^{d} \ln\left(\varepsilon + |w_j|\right) \\ \text{subject to} & y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 0 \\ \text{and} & \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-}\right) = 1 \end{array} \tag{2.8}$$

with a similar iterative scheme (note that in practice, no choice for ϵ is needed as we never optimize the above equation directly):

```
Initialize $\boldsymbol{z} = (1, \ldots, 1)$
**repeat**
    Minimize $|\boldsymbol{w}|$ such that $y_i\left(\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_i * \boldsymbol{z}) + b\right) \geq 0$ and $\boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-}\right) = 1$
    Update $\boldsymbol{z} = \boldsymbol{z} * \boldsymbol{w}$
**until** *convergence*
```

Thus, by successively minimizing the one-norm, we aim to approximate the zero-norm minimizing solution as accurately as possible. By that, the SFM is a combinatorial feature selection method working on the original set of input features and taking class information into account. Importantly, in contrast to dimension-reduction methods like principal component analysis (PCA), which reduce the dimensionality of a transformed space, SFM reduces the dimension of the original feature space.

2.3 Extending the Support Feature Machine to Soft Separability. In general, if $n \leq d + 1$, then the data will be separable and the SFM has a solution. In the following, we introduce slack variables similar to a soft-margin SVM to allow for misclassifications during training. This is done for two reasons. First, if the input data are not separable in the intrinsic feature space (i.e., if the classes overlap), irrelevant features will be added to achieve separation of the training data. This might lead to an overestimation of the number of truly relevant features and diminish generalization performance. Second, even if the classes are in principle separable in the intrinsic feature space, the true separating hyperplane might not be identified correctly due to outliers. To address these problems, a mechanism is needed that allows for misclassifications and thereby provides a better estimate of the true dimensionality. (Note that we do not address the problem of intrinsically nonlinear decision borders here.)

We introduce slack variables ξ_i for each data point and a softness parameter C (Klement & Martinetz, 2010a) in the same way this is done for soft-margin SVMs; we

$$\begin{aligned} &\text{minimize} \quad \|\boldsymbol{w}\|_0^0 + C\|\boldsymbol{\xi}\|_0^0 \\ &\text{subject to} \quad \begin{cases} y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq -\xi_i \quad \forall\, i \\ \boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-) = \pm 1 \\ \xi_i \geq 0 \quad \forall\, i. \end{cases} \end{aligned} \tag{2.9}$$

As classification errors are allowed, $y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)$ may become negative and the pathological case where equation 2.7 is smaller or equal to zero may occur. Therefore, the optimizer needs to fulfill the latter constraint with $+1$ or -1. In practice, one needs to optimize for both variants and finally choose the solution with the lower objective function. To solve equation 2.9, we use the iterative approximation scheme described above. Importantly, one property of this approach is that the objective function explicitely trades off the number of features $\|\boldsymbol{w}\|_0^0$ and the number of misclassified training samples $\|\boldsymbol{\xi}\|_0^0$.

2.3.1 Unbalanced Data Sets. An important problem of classifiers with soft margins is their sensitivity to unbalanced data sets. If one class contains more samples than the other, many classifiers tend to behave like a majority classifier and ignore the smaller class. Several solutions to this problem have been proposed, such as rebalancing the data artificially, adjusting the output threshold of the classifier according to the class ratio, one-class classifiers, and cost-sensitive methods (Provost, 2000; Japkowicz, 2000; Chawla, Japkowicz, & Kotcz, 2004; He & Garcia, 2009).

Here we propose an approach in which the softness of the SFM is adjusted according to the class ratio by assigning individual misclassification costs to each class. The optimization problem then becomes

$$\begin{aligned} &\text{minimize} \quad \|\boldsymbol{w}\|_0^0 + C^+\|\boldsymbol{\xi}^+\|_0^0 + C^-\|\boldsymbol{\xi}^-\|_0^0 \\ &\text{subject to} \quad \begin{cases} y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq -\xi_i^+ \quad \text{for all } i \in I^+ \\ y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq -\xi_i^- \quad \text{for all } i \in I^- \\ \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-\right) = \pm 1 \\ \xi_i^+, \xi_i^- \geq 0\,. \end{cases} \end{aligned} \tag{2.10}$$

By scaling C^+ and C^- appropriately, one can avoid the situation that a much smaller class tends to be ignored. In practice, choosing C^+ and C^- such that

$C^+n^+ = C^-n^-$ enforces the proportion of misclassified samples to be equal for both classes.

2.3.2 Behavior in the Limit. The extension to soft separability reduces the impact of single outliers on the number of obtained features by trading off the number of features and the number of misclassified samples. To complete our softness extension, we consider the behavior of the soft SFM in the limit for $C^\pm \to \infty$ and $C^\pm \to 0$. In the first case, dominance of the slack term $C^+|\boldsymbol{\xi}^+\|_0^0 + C^-\|\boldsymbol{\xi}^-\|_0^0$ is equivalent to setting the slack variables to zero such that we obtain the hard SFM. The converse case, $C^\pm \to 0$, allows arbitrary choices of the slack variables ξ_i^+ and ξ_i^- such that the objective function becomes independent of the misclassification rate. Thus, the inequality constraints are fulfilled for all $\boldsymbol{w}$ and b. In the limit, the optimization problem, equation 2.10, simplifies to

$$\begin{aligned} &\text{minimize} && \|\boldsymbol{w}\|_0^0 \\ &\text{subject to} && \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-\right) = 1\,. \end{aligned}$$

The minimum, of course, is $\|\boldsymbol{w}\|_0^0 = 1$. The SFM approximates the zero-norm by the one-norm. We prove that the SFM finds the optimum $\|\boldsymbol{w}\|_0^0 = 1$ (i.e., exactly one feature) with using the proposed approximation. Assume that $k > 1$ features are obtained by minimizing $\|\boldsymbol{w}\|_1$. Without loss of generality, we assume that these are the first k features. Further, without loss of generality we assume $|\mu_1^+ - \mu_1^-| > |\mu_i^+ - \mu_i^-|$ for all $1 < i \leq k$ (note that equality has probability 0). We may rewrite the equality constraint as

$$\underbrace{\left(w_1 + \sum_{i=2}^{k} w_i \frac{\mu_i^+ - \mu_i^-}{\mu_1^+ - \mu_1^-}\right)}_{w_1'} \left(\mu_1^+ - \mu_1^-\right) = 1\,.$$

Thus, the equality can be fullfilled using only one feature but with a modified weight w_1'. Since

$$\|\boldsymbol{w}'\|_1 = |w_1'|_1 = |w_1| + \sum_{i=2}^{k} \left| w_i \frac{\mu_i^+ - \mu_i^-}{\mu_1^+ - \mu_1^-} \right| < |w_1| + \sum_{i=2}^{k} |w_i| = \|\boldsymbol{w}\|_1,$$

we would obtain a smaller objective function using exclusively feature 1, in contradiction to our initial assumption. Thus, in the limit for $C^\pm \to 0$, no more than one feature is obtained.

A very soft SFM will identify one and only one feature to be relevant, that is, the weight vector differs from zero in exactly one entry. Let j be the index of this nonzero entry. Then the equality constraint is solved with respect to w_j by

$$w_j = \frac{1}{\mu_j^+ - \mu_j^-},$$

where μ_j^+ and μ_j^- are the projection of the class-specific means onto the axes of feature j. As the SFM approximates the zero-norm by minimizing the one-norm, it will select the feature that minimizes $|w_j|$ and therefore maximizes the distance of the class-specific means $|\mu_j^+ - \mu_j^-|$. This is closely related to correlation-based feature selection methods such as the Fisher score and statistical distance measures such as Student's *t*-test. Consequently, we expect soft SFM to favor those features that maximize the correlation between feature value and class label, or the distance between the feature values of the two classes. In other words, the soft SFM is a trade-off between a hard SFM and correlation-based feature ranking.

3 Experiments on Artificial Data

Having reviewed and reformulated the basic principles of the SFM, we continue to show that the SFM outperforms Weston's SVM-based feature selection method in many simulations, particularly in high-dimensional small-sample-size scenarios. We first compare the capability of the SFM and Weston's method to identify the truly relevant features in a simple scenario with linearly separable but slightly unbalanced classes. For this, we use an experiment that is very similar to that described in Klement and Martinetz (2011) but now includes unbalanced classes. We then show how both methods perform if an increasing number of irrelevant features is added to the data, which is probably the most challenging setting for any feature selection method. Finally, we test the performance of the soft SFM on nonseparable data.

3.1 Basic Experiment.

For the basic experiment we generated artificial data sets with class ratio 60% versus 40%. The first k dimensions $x_i, \ldots, x_k$ were drawn as $x_i = \mathcal{N}(\mu \cdot y, 1)$. The remaining features $x_{k+1}, \ldots, x_d$ were noise drawn as $x_i = \mathcal{N}(0, 1)$. The parameter μ determines the distance between the means of the two classes. We ensured that both classes were linearly separable within the first k dimensions. This was achieved by removing samples that did not fulfill $y_i \sum_j x_{ij} > 0$, that is, we assumed all entries of the real weight vector in the first k dimensions to be one. Then we resampled all invalid data points and repeated both steps until convergence. (Note, however, that by chance, both classes might be separable with even fewer than k features.) The number of dimensions was set to $d = 100$. In the first experiment, we used a fixed sample size of $n = 100$ and varied the number of relevant features from $k = 1$ to $k = 20$. In the second

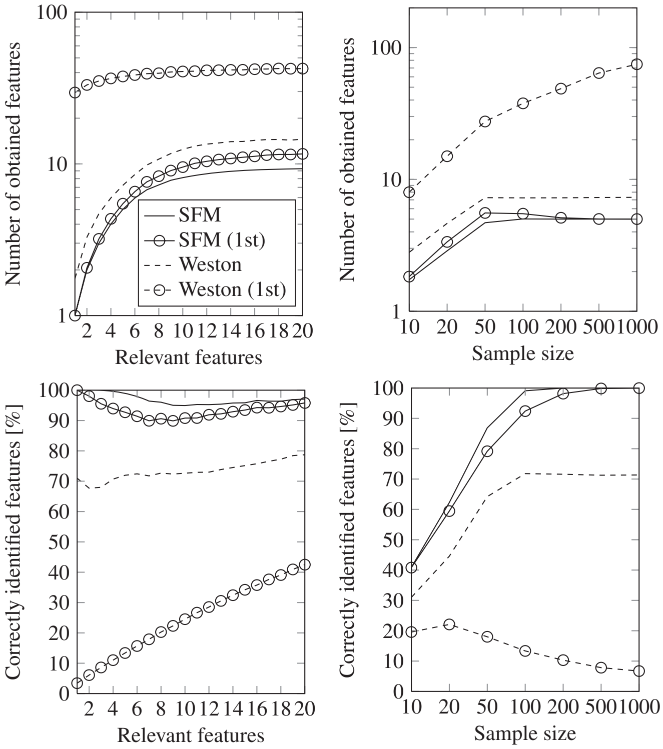


Figure 1: Feature selection performance of the SFM and Weston's method as a function of the number of truly relevant features and the number of data points. Shown are the average number (1000 runs) of obtained features (top) and the average percentage of correctly identified features (bottom) for the basic hard SFM and Weston's method after the first and after the final iteration. In the left column, the number of data points is fixed ($n = 100$), while in the right column, the intrinsic dimensionality is fixed ($k = 5$). The remaining parameters are equal in both settings ($d = 100$, $\mu = 0.3$) (Klement & Martinetz, 2011, slightly modified).

experiment, we set the number of relevant features to $k = 5$ and varied the sample size from $n = 10$ to $n = 1000$.

Figure 1 shows the average results for 1000 runs with Weston's method and a hard SFM. The SFM returned both a smaller total number of features and a higher percentage of correctly identified features for almost all scenarios. In scenarios with very low intrinsic dimensionality—for $k = 1, \dots, 4$—the SFM identified all relevant features correctly (see Figure 1, bottom left). In scenarios with $k = 5$, the SFM identified all relevant features correctly in every run if the number of data points exceeded 200. Weston's method failed to converge to the correct number of features even if the number of data points was further increased (to 1000), the percentage of correctly identified features stayed clearly below 80% (see Figure 1, bottom right). Thus, the SFM converged to the correct set of features, while the SVM-based approach got stuck in a local minimum even for large data sets.

Also note that in contrast to Weston's method, the SFM was close to the final solution in the first iteration.

3.2 Experiment with Increasing Dimensionality. For many biological applications, a feature selection method should not only be able to deal with high-dimensional data sets but should also scale well; adding irrelevant features should not significantly degrade the performance. To assess how the performance of the hard SFM and Weston's method degrades when irrelevant features are added to the data, we used an artificial dataset that has been used in a variety of publications but was originally proposed by Weston et al. (2003). The data set contains two equally sized classes where the first six dimensions are informative but redundant and the remaining dimensions contain gaussian noise. With probability 0.7, the first three features are drawn as $x_i = y\mathcal{N}(i, 1)$, and the second three features are drawn as $x_i = \mathcal{N}(0, 1)$. With probability 0.3, the setting is inverted: the first three features are drawn as $x_i = \mathcal{N}(0, 1)$ and the second three as $x_i = y\mathcal{N}(i - 3, 1)$. The remaining k^* features are drawn as $x_i = \mathcal{N}(0, 20)$ with $k^* = 10, 10^2, 10^3, 10^4$. Additionally, we ensured the training set to be linearly separable within the six informative dimensions. We sampled n training points ($n = 20, 50, 100, 200, 500$) and 5000 test data points. Note that this setting is slightly different from the one we originally evaluated in Klement and Martinetz (2010a). Here, the largest number of irrelevant features is 10,000 (instead of 4096), we use a more convient spacing for k^*, and, to obtain results with higher confidence, the results are averaged across 1000 (instead of 100) runs).

Table 1 compares the capability of both feature selection methods to identify relevant features. Compared to Weston's approach, the SFM returns a smaller number of features and more likely the truly relevant features. Even in very high-dimensional small-sample-size scenarios, the SFM can identify the relevant dimensions very effectively: as the number of data points increases, the number of features found to be relevant increases but does not exceed 6—the number of truly relevant features. The percentage of correctly identified features decreases when the number of noise dimensions increases. However, only in extremely high-dimensional small-sample-size scenarios does the percentage of correctly identified features drop below 90%.

The sampling scheme causes the number of correctly identified features not to converge to 100% for large n. Due to the experimental design, some features provide better separability than others. Then, by chance, one of the irrelevant features may be favored by the SFM (and also Weston's method)

Table 1: Impact of an Exponentially Increasing Number of Irrelevant Features on Feature Selection Performance, with Six Features Being Relevant.

n	$k^* = 10$	$k^* = 100$	$k^* = 1000$	$k^* = 10{,}000$
SFM, number of obtained features				
20	2.0 (±0.6)	2.0 (±0.6)	2.0 (±0.6)	1.9 (±0.6)
50	2.3 (±0.6)	2.4 (±0.6)	2.4 (±0.6)	2.5 (±0.7)
100	2.7 (±0.7)	2.6 (±0.6)	2.6 (±0.7)	2.6 (±0.7)
200	3.1 (±0.7)	3.2 (±0.8)	3.2 (±0.8)	3.1 (±0.7)
500	4.1 (±0.8)	4.2 (±0.8)	4.2 (±0.8)	4.2 (±0.8)
Weston, number of obtained features				
20	2.6 (±0.8)	2.8 (±0.9)	3.0 (±1.1)	3.2 (±1.1)
50	3.2 (±1.0)	3.4 (±1.1)	3.3 (±1.1)	3.4 (±1.1)
100	3.8 (±1.1)	4.0 (±1.2)	4.0 (±1.2)	3.9 (±1.2)
200	4.6 (±1.2)	4.9 (±1.4)	4.9 (±1.4)	4.8 (±1.2)
500	5.9 (±1.3)	6.2 (±1.5)	6.4 (±1.5)	6.2 (±1.4)
SFM, percentage of correctly identified features				
20	98.5% (±8.0%)	88.9% (±20.2%)	66.5% (±29.1%)	46.9% (±32.9%)
50	99.6% (±3.5%)	98.8% (±6.5%)	96.7% (±10.9%)	85.9% (±22.4%)
100	99.7% (±3.1%)	99.1% (±5.3%)	97.3% (±8.8%)	96.9% (±9.4%)
200	99.4% (±3.9%)	98.5% (±6.0%)	96.6% (±8.9%)	95.1% (±11.0%)
500	98.8% (±5.0%)	96.4% (±8.7%)	94.3% (±10.8%)	92.2% (±11.7%)
Weston, percentage of correctly identified features				
20	94.8% (±12.8%)	81.2% (±23.7%)	58.0% (±29.6%)	32.5% (±26.1%)
50	94.0% (±12.1%)	87.6% (±16.0%)	85.0% (±18.0%)	79.7% (±20.3%)
100	93.1% (±12.0%)	87.0% (±16.0%)	83.4% (±17.5%)	83.1% (±17.1%)
200	89.0% (±13.6%)	82.9% (±15.9%)	80.3% (±16.0%)	80.6% (±16.0%)
500	82.2% (±12.9%)	76.2% (±14.5%)	73.5% (±14.7%)	74.1% (±14.6%)

Note: Shown are the average (over 1000 runs) number of features found to be relevant (±std).

instead of the weakest relevant feature. This effect gets amplified for large sample sizes as the solution space becomes smaller and smaller.

3.3 Experiment with Nonseparable Classes. Finally, we constructed an artificial problem where the two classes are not linearly separable. The probabilities of the classes $y = 1$ and $y = -1$ were equal in both the training and the test sets. The first k dimensions $x_1, \ldots, x_k$ were drawn normally distributed as $x_i = \mathcal{N}(\mu \cdot y, 1)$. The remaining features $x_{k+1}, \ldots, x_d$ were noise drawn as $x_i = \mathcal{N}(0, 1)$. The parameter μ was used to adjust the distance between the class centers. Both the training and the test sets were sampled according to the above procedure, each containing n data points. The softness parameter C was varied in 100 steps logarithmically spaced between 0.01 and 100. This basic setting is the same as in Klement and Martinetz (2010a). Here, we extend this basic setting to examine the soft SFM in four different scenarios with variable numbers of relevant and irrelevant

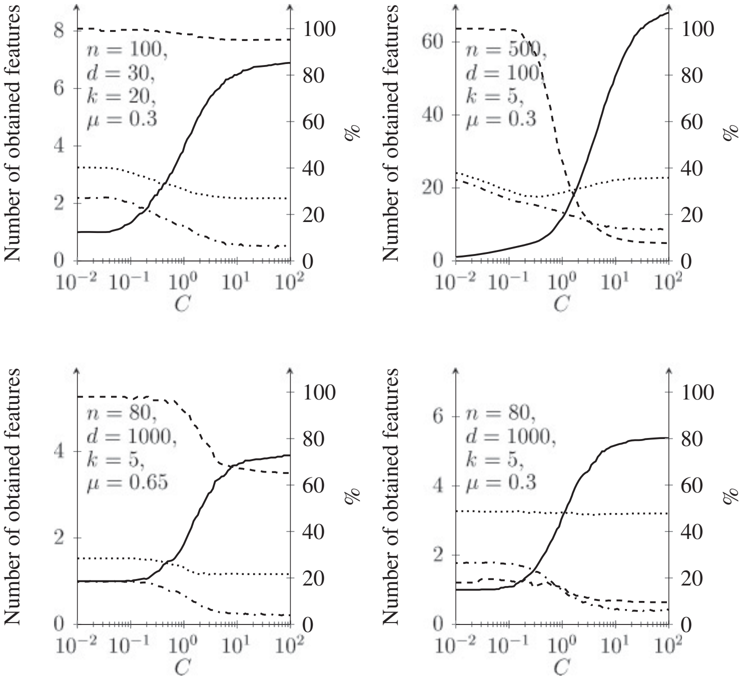


Figure 2: Behavior of the soft SFM for linearly nonseparable classes. Shown are four scenarios and the resulting number of obtained features (solid line, left axis), percentage of correctly identified features (dashed, right axis), training error (dash-dotted, right axis), and test error (dotted, right axis) averaged over 100 runs.

features. In the first scenario, the data contain a high percentage of relevant features. In the second scenario, we reduced the number of relevant features and increased the number of irrelevant features. Finally, in the third and fourth scenarios, we further increased the number of irrelevant features and reduced the number of data points.

Figure 2 shows the averaged results of 100 runs. In all four scenarios, increasing softness—allowing more training errors—results in a smaller number of obtained features. This is in line with the theoretical consideration that the soft SFM directly trades off the number of obtained features and the training error and that a very soft SFM will return a single feature. As the number of obtained features decreases, the percentage of correctly identified features increases. However, this does not necessarily result in increased prediction performance. In the second scenario, the test error reaches a minimum just after the point where the percentage of correctly identified features sharply increases. In the first and third scenarios, however, the test error increases with increasing softness, and in the fourth scenario, the test error remains almost constant throughout all softness values. Below, we discuss the four data scenarios in detail.

3.3.1 Many Relevant Features, Few Irrelevant Features, Large Sample Size. In this case the data contain a large percentage of relevant features and the number of data points exceeds the number of dimensions (see Figure 2, top left). Note, however, that features might be correlated and not all features might actually be required to separate the classes. The SFM identified almost no irrelevant feature as being relevant, independent of the softness. However, only a small fraction of the truly relevant features was identified (at most 7 out of 20). The test error increased slightly with increasing softness.

3.3.2 Few Relevant Features, Many Irrelevant Features, Large Sample Size. In this scenario, the data contain few relevant and many irrelevant features, but the number of data points still exceeds the number of dimensions (see Figure 2, top right). Increasing the softness resulted in a smaller set of obtained features, while the percentage of correctly identified features increased. For the parameters chosen here, the optimal test error (in this case 27.6%) is achieved for medium softness ($C = 0.34$), approximately at that point where almost all truly relevant features and almost no irrelevant features were obtained. In particular, 5.42 features were obtained, of which 86.7% were correctly identified features on average. Thus, in such scenarios, optimizing for the test error is a valid approach to identify the true set of relevant features.

3.3.3 Few Relevant Features, Many Irrelevant Features, Small Sample Size. The third and fourth scenarios both represent high-dimensional small-sample-size data with few relevant and many irrelevant features (see Figure 2, bottom). Note that this is very challenging because the information content is very small. As before, increasing softness resulted in a smaller number of obtained features while the percentage of correctly identified features increased. In the third scenario (see Figure 2, bottom left), we chose a larger class distance than in the previous scenarios (μ was increased from 0.30 to 0.65). The test error increased slightly with increasing softness but stayed well below chance. In the fourth scenario, we used the same class distance as in the first two scenarios. Although the percentage of correctly identified features still increased with increasing softness, the test error remained almost constant and never fell well below chance (see Figure 2, bottom right). Thus, in this scenario, the class overlap seems to be too large to obtain meaningful results.

Some authors have recommended using a soft- rather than a hard-margin SVM even if the training data are linearly separable to improve the prediction accuracy (see, Hastie, Rosset, Tibshirani, & Zhu, 2004). The third and fourth scenarios are both scenarios with $n \ll d$. In these scenarios, the softness had little impact on the test error but largely influenced the number of obtained features.

In sum, applying a soft SFM to the four exemplary data sets shows that the soft SFM performs well on data sets with nonseparable classes even in scenarios with few relevant and many irrelevant features, as long the class overlap is not too large relative to the number of data points. In high-dimensional small-sample-size scenarios with few relevant features and strongly overlapping classes, the performance of the SFM drops to chance, possibly because the information content is no longer sufficient to allow valid feature selection. However, we expect that in such scenarios, performance of any feature selection method will be severely hampered.

4 Application to Neuroimaging Data

So far, we have demonstrated that the SFM identifies truly relevant features in artificial data sets very effectively, particularly if the data contain few relevant and many irrelevant dimensions. In this section, we illustrate how the SFM might be used to identify not only the most informative features but also the total number of informative features in biological data sets. For this purpose, we chose an fMRI (functional magnetic resonance imaging) data set.

In fMRI, researchers are almost always faced with high-dimensional small-sample-size scenarios (≈100,000 dimensions versus ≈100 samples). To reduce the feature space, feature selection and classification are often performed either independently (e.g., principal component analysis; Carlson, Schrater, & He, 2003; Strother et al., 2002) or recursively (e.g., recursive feature elimination, Martino et al., 2008), but there are also approaches that, like the SFM, combine feature selection and classification within a single framework (e.g., Elastic Net, Carroll, Cecchi, Rish, Garg, & Rao, 2009; sparse penalized discriminant analysis (SPDA), Grosenick, Greer, & Knutson, 2008; and sparse logistic regression, Yamashita, Sato, Yoshioka, Tong, & Kamitani, 2008; Ryali, Supekar, Abrams, & Menon, 2010). In addition to representing extreme high-dimensional small-sample-size scenarios, fMRI data—like many high-dimensional real-world data sets—often contain several informative feature subsets that all permit linear separation (i.e., the data are highly redundant). In such scenarios, one might not only be interested in finding the most informative features, but also in identifying all informative features (Rasmussen, Hansen, Madsen, Churchill, & Strother, 2012). In this case, repetitive feature selection (RFS) might provide an estimate of the total number of informative features even if the number of features that carry information alone or in combination with others cannot be determined exactly (the sample size is usually too small to capture all sources of variance and to accurately describe the decision border).

As noted in section 1, the accuracy of this estimate will depend strongly on the selectivity of the applied feature selection method. If irrelevant (uninformative) features are falsely identified as being relevant, then the proportion of relevant features in the data set is overestimated. In contrast, the estimate does not strongly rely on the sensitivity of the feature selection; even if only one (truly relevant) feature is returned in each repetition, the proportion of informative voxels will still be estimated correctly. Since the SFM is very restrictive in the way it selects relevant features and returns a high percentage of truly relevant features (see section 3 and appendix A), the estimate of informative features obtained with a repetitive SFM (rSFM) likely represents an unbiased estimate of the true number of informative features. This makes the SFM a potentially powerful tool to identify not only the most informative features but also the total number of informative features in biological data sets.

4.1 FMRI Data Set. FMRI data acquired from 12 healthy female participants (mean age 21.6 years, range 19–26 years) on a 3 Tesla scanner (Philips Medical Systems) were used for analysis. Functional imaging was divided into 4 runs per subject; during each run 67 functional whole-brain images were acquired (T_2^* weighted echoplanar images, 42 horizontal interleaved slices, tilt angle -30^o, 3 mm slice thickness, in plane resolution 3×3 mm^2, FOV 240×240 mm^2, TE 35 ms, TR 3000 ms). Participants were shown short text messages (either *happy* or *sad*) through fMRI-compatible video goggles and asked to decide whether they wanted to press a button in their left or right hand immediately whenever a text message appeared on the screen, but to hold their decision in mind and to execute their decision only when a go signal (two arrow heads, one pointing to the left and one pointing to the right) appeared on the screen. Participants were instructed to respond as quickly as possible when the go signal appeared by pressing the selected button with their left or right thumb, respectively. During each run, 12 trials (mean duration 5 scans) were presented in pseudo-randomized order, using the following timing parameters: stimulus presentation time, 1000 ms; delay, 2000 or 3500 ms; go signal, 300 ms; inter trial interval, 8700 to 13,200 ms (steps of 1500 ms). The study was approved by the Ethics Committee of the University of Lübeck.

Image preprocessing and BOLD (blood oxygen level dependent) activity estimation were conducted with SPM5 (Wellcome Department of Imaging Neuroscience, London, UK), and results were visualized with the BrainNet Viewer (National Key Laboratory of Cognitive Neuroscience and Learning, Beijing Normal University, China) and MRicron (www.mccauslandcenter.sc.edu/mricro/mricron/). Preprocessing included removal of the first two functional scans of each run, slice acquisition time correction, concurrent spatial realignment and correction of image distortions, normalization into standard MNI (Montreal Neurological Institute) space, and spatial smoothing with an 8 ms gaussian kernel.

Individual activity maps for left-hand and right-hand button presses were estimated for each participant and run using a standard GLM procedure. This procedure returned eight activation maps for each participant (four for left-hand button presses and four for right-hand button presses). Thus, the overall data set, in the following referred to as the button press data set consisted of 48 maps (12 participants × 4 runs) per class, each of which contained 50,989 in-brain voxels identified with the brain mask published by Tzourio-Mazoyer et al. (2002).

4.2 Mass Univariate Analysis. The traditional approach to identify voxels in fMRI data that show different levels of activity during two conditions (i.e., that are discriminative) are voxel-wise univariate analyses. Here, we used such a mass univariate approach as baseline measure against which the performance of the rSFM was compared. For this, univariate activation maps of each participant were averaged for each class and fed into a group-level voxel-wise paired *t*-test. The overlap between voxels with high absolute *t*-values and voxels identified as being relevant with the rSFM was assessed at different thresholds. Thresholds were defined in terms of the fraction of above-threshold voxels (i.e., percentage of voxels) rather than in terms of probability of error to keep the number of above-threshold voxels constant across the univariate and rSFM-based approach. Six thresholds were used: 1%, 2.5%, 5%, 20%, 34%, and 50%. The second threshold (2.5%) corresponded to a (uncorrected) voxel-wise probability of error of $p = 0.001$, which is a commonly used (although debated; see Nichols, 2012) threshold in voxel-wise analysis of fMRI data. Voxels that were identified as discriminative at this threshold are listed in Table 2. The second last threshold (34%) corresponded to the rSFM-based estimate of the upper limit of informative voxels in the button press data set (see below). The other thresholds were chosen to cover a wide range of different thresholds. Corresponding (uncorrected) voxel-wise probabilities of error are given in Table 3 (column 4).

4.3 Support Vector Machine with Random Feature Selection. To obtain a baseline estimate of the number of informative voxels in the button press data set, we trained a hard-margin SVM on randomly chosen d-dimensional feature subsets ($d = 1, 2, 4, \ldots, 32{,}768$) in a leave-one-participant-out cross-validation scheme with 44 samples in each class (11 participants × 4 runs). This procedure was repeated 1000 times for each subset size. The test error was below chance for the large majority of repetitions even if only a single dimension was selected at random, and close to zero if more than 1000 features were included, indicating that a large fraction of voxels carried relevant information (see Figure 3).

4.4 Repetitive Feature Selection (RFS) with the SFM. The basic idea of SFM-based RFS is to train an SFM on the complete data set, remove all features from the data set that are found to be relevant by the SFM, retrain on

Table 2: Discriminative Voxels as Identified by Voxel-Wise *t*-Statistics.

	Significant Voxels	
Anatomical Region Name	Left	Right
Postcentral gyrus	25.0% (319)	16.0% (204)
Precentral gyrus	7.4% (94)	7.5% (96)
Cerebellum VI	4.1% (52)	6.1% (78)
Inferior parietal lobe	5.3% (67)	1.3% (16)
Cerebellum IV/V	2.5% (32)	3.1% (39)
Putamen	1.5% (19)	0.0% (0)
Supplementary motor area	1.3% (17)	0.0% (0)
Superior parietal lobe	1.3% (16)	1.3% (16)
Pallidum	1.0% (13)	0.0% (0)
Supramarginal gyrus	0.9% (11)	0.4% (5)
Unassigned	9.0% (115)	
Other regions (<1%)	5.1% (65)	

Notes: Shown are the 2.5% most significant voxels (uncorrected voxel-wise $p \leq 0.001$). Anatomical regions were identified by an automatic labeling procedure (Tzourio-Mazoyer et al., 2002; Schmahmann et al., 1998). Only regions that contain at least 1% of all significant voxels across hemispheres are listed. Numbers in brackets are numbers of discriminative voxels in each region. The majority of discriminative voxels form large clusters in each hemisphere, including part of the precentral and postcentral gyri (motor and somatosensory cortex) and part of the cerebellum, respectively.

the reduced data set, discard the obtained relevant features, retrain again, and so on, until the data set is no longer separable within the remaining features (Klement & Martinetz, 2010a).

```
Initialize the set of active features $\mathcal{F}_0 \leftarrow \{1, \ldots, d\}$
Set $t \leftarrow 0$
**repeat**
  Train a support feature machine using the feature set $\mathcal{F}_t$
  **if** *a solution was found* **then**
    Store the results, i.e., $\boldsymbol{w}_t$ and $b_t$
    Store the set of relevant features, i.e., $\mathcal{R}_t = \{i \mid w_{t,i} \neq 0\}$
    Update the set of active features, i.e., $\mathcal{F}_{t+1} = \mathcal{F}_t \setminus \mathcal{R}_t$
    Reduce all feature vectors to $\mathcal{F}_{t+1}$
    Set $t \leftarrow t + 1$
  **end**
**until** *until the data set is no longer separable within the remaining features*
```

If the SFM correctly identifies the smallest informative feature subset in each run, the size of the returned feature subsets will monotonously increase as more and more features are discarded. However, in practice, this might not always be the case because the optimization might terminate

Table 3: Observed and Expected Overlap Between Voxels Identified as Being Relevant by the rSFM and Voxels Identified as Being Discriminative by Univariate *t*-Statistics.

Percent Voxels Identified as Relevant/Discriminative (threshold)	Percent Overlap (observed)	Percent Overlap (expected)	Voxel-Wise Probability of Error (univariate analysis)
1%	53%	1%	0.0001
2.5%	64.3%	2.5%	0.001
5%	61.6%	5%	0.006
20%	67.7%	20%	0.093
34%	72.7%	34%	0.23
50%	77.6%	50%	0.41

Notes: The overlap expected by chance (third column) for two subsets covering the same percentage q is q^2 with respect to the whole set and q with respect to one subset. Thus, percentages in the first and third columns are the same. Additionally, the fourth column shows the voxel-wise p-value of the least discriminative among all relevant voxels at this threshold.

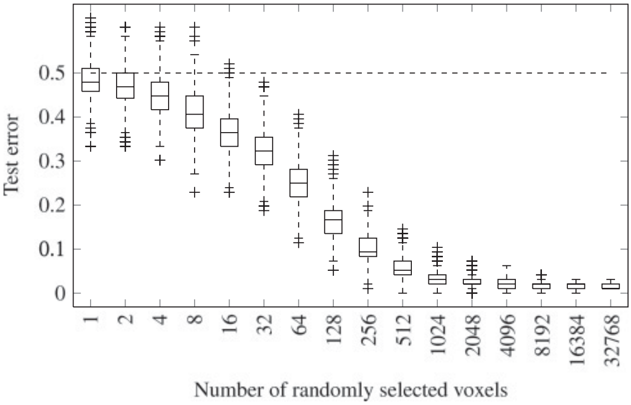


Figure 3: Support vector machine with random feature selection. The box plot visualizes the distribution of the leave-one-participant-out cross-validation error (median, lower, and upper quartile, outliers).

in a local optimum due to the data set configuration or because of numerical issues of the technical implementation of the SFM. To correct for such inaccuracies, we sorted the obtained feature subsets according to their size, starting with the smallest feature subset. This way, we obtained a sequence of monotonously increasing feature subsets that, according to our definition, represents a sequence of feature subsets that are less and less informative.

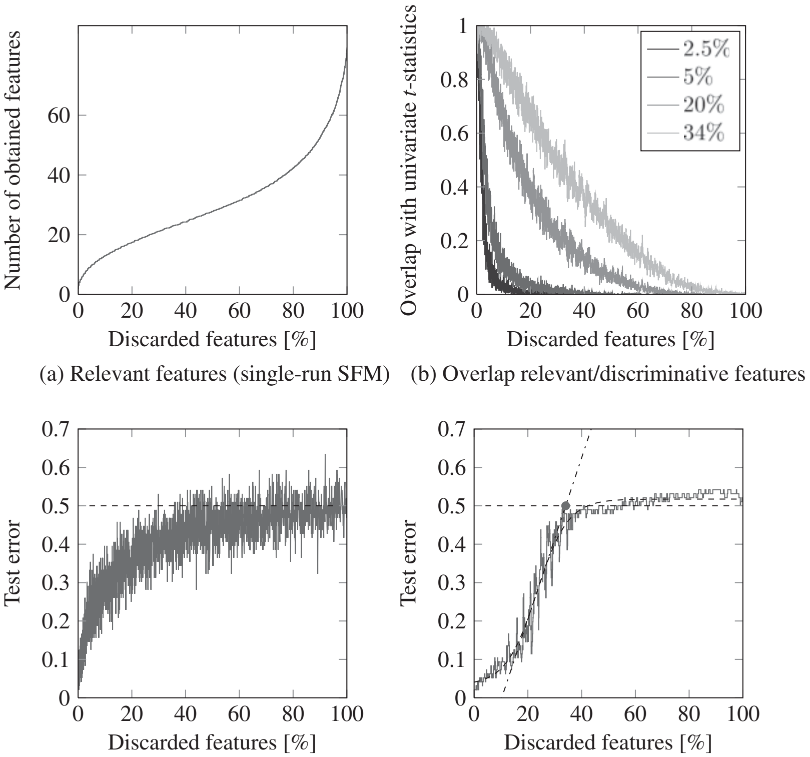
To assess both the information content of each feature subset and the information content in the remaining features in each repetition, we used a leave-one-participant-out cross-validation scheme similar to that described above. In each repetition, information content was accessed as the test error of the SFM and the test error of a soft SVM, respectively. The soft SVM was trained on all features that remained in the data set after all features that had been identified as being relevant by the SFM had been removed. The optimal softness of the SVM ($C \in \{10^{-8}, 10^{-7}, \ldots, 10^{7}, 10^{8}\}$) for each cross-validation was estimated by a nested cross-validation scheme in which the SVM was trained on $10 \times 4 = 40$ samples in each class and tested on the 11th subject. Once the optimal softness parameter was determined, the SVM was retrained on all 11 subjects and tested on the 12th subject. This way, a function representing the test error of an optimized soft SVM over runs was derived for each of the 12 participants.

Note that this method requires training 188 SVMs for a single SFM (11 participants × 17 softness values + the final run). To keep the run time in a reasonable range, the SVM was trained and tested only on every 10th SFM repetition.

Training and testing the repetitive SFM on the complete data set in a leave-one-participant-out scheme required 31,647 training runs (i.e., for each of the 12 cross-validations approximately 2500 repetitions before all voxels were discarded) and took 83 hours on an Intel Core 2 Quad 2.4GHz machine with 4GB RAM (using the Mosek optimization toolbox for solving the SFM). Training and testing SVMs on the remaining features for every 10th SFM repetition took another 27 hours.

Because we used a leave-one-participant-out scheme for cross-validation, the feature set size and error functions obtained during each validation did not have the same length (i.e., the number of repetitions until all features are consumed differed across validation runs). Thus, these functions needed to be resampled before averaging. We chose a resampling procedure in which feature subsets were first sorted according to their size, and each $x \in 1, \ldots, d$ was then assigned the performance value of the last run in which fewer than x features were removed. These piece-wise constant curves were then averaged across all leave-one-participant-out cross-validations.

The smallest feature subset obtained by the rSFM contained 2.4 voxels on average. The largest feature subset contained an average of 77.7 voxels, which is below the upper bound (i.e., number of data points, $n = 88$ samples, minus 1—VC-dimension of a linear classifier: see Figure 4a). Critically, the comparison of voxels obtained by the rSFM and voxels identified as being discriminative by univariate t-statistics revealed a large overlap across different thresholds that was by far larger than the overlap expected by chance (Table 3 and Figure 5). Furthermore, this overlap was largest for feature sets removed early in the repetitive feature selection scheme (i.e., small feature subsets) and decreased continuously toward the end of the RFS (see



(a) Relevant features (single-run SFM) (b) Overlap relevant/discriminative features

(c) Test error (single-run SFM) (d) Test error (SVM on remaining features)

Figure 4: Analysis of the fMRI data set with the rSFM and an optimized soft SVM trained on the remaining features. Shown are the (a) average number of relevant features (b), the average overlap between the features identified with the rSFM, and those that were found to be discriminative with voxel-wise t-statistics (top 2.5% of all features, $p \leq 0.001$ as well as top 5%, 20% and 34%) (c), the average leave-one-participant-out cross validation error of the SFM, and (d) the average leave-one-participant-out cross-validation error of an SVM trained on the remaining features for every 10th SFM repetition. To approximate the number of features (voxels) that carry information, a sigmoid function was fitted to the test error function of the SVM (dashed). The straight line (dash-dotted) through the inflexion point of the sigmoid crosses chance level at 34% (black dot).

Figure 4b). This indicates that the SFM very quickly consumes significant features before other features are included.

Having shown that the rSFM indeed identifies discriminative voxels as being informative, we continued to pursue our last question: whether the rSFM can be used to estimate the number of relevant voxels within a data

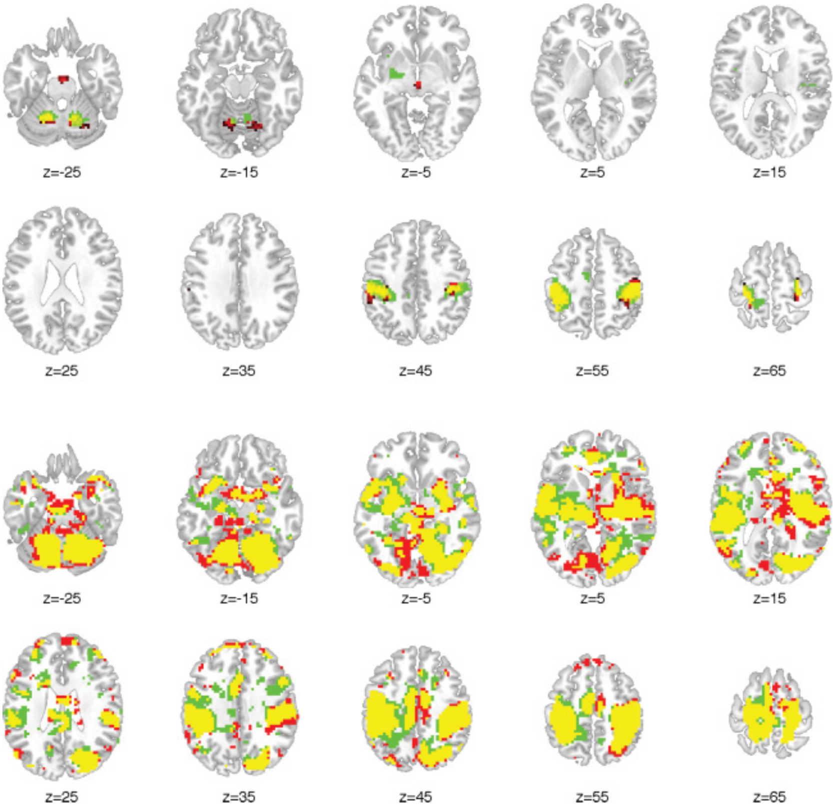
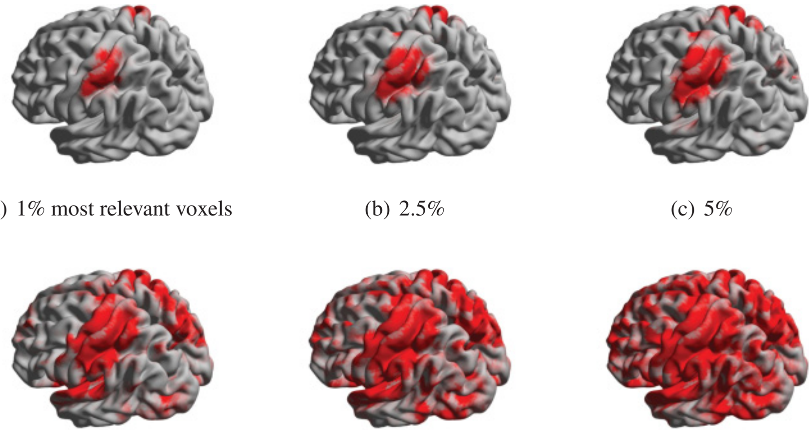


Figure 5: Overlap between relevant voxels (identified by the rSFM) and discriminative voxels (identified with voxel-wise *t*-statistics). Voxels (features) found to be relevant by the rSFM (2.5% most relevant voxels in at least half of all participants in the top rows/ 34% most relevant voxels in at least half of all participants in the bottom rows) are red. Voxels identified as being discriminative by voxel-wise *t*-statistics (most significant 2.5%/34% of all voxels, the former corresponding to an uncorrected voxel-wise $p = 0.001$) are green. Overlapping regions are yellow. Voxels are projected onto horizontal slices of a standard anatomical template (MNI space, most ventral slice $z = -25$, most dorsal slice $z = 65$, spacing 10 mm). Discriminative voxels are mainly located in the precentral/postcentral gyrus (motor and somatosensory cortex) and, at the more lenient threshold, in the SMA (supplementary motor area), with a high degree of overlap between the two methods.

set. The test error of both the rSFM and the SVM converged to chance level as more and more features were discarded, indicating that the remaining voxels carry less and less information (see Figures 4c and 4d). However, due to large repetition-to-repetition fluctuations, it is difficult to estimate



(a) 1% most relevant voxels (b) 2.5% (c) 5%

(d) 20% (e) 34% (f) 50%

Figure 6: Location of voxels identified as relevant by the rSFM. Numbers indicate percentage of voxels. Color intensity indicates depth below surface and consistency across participants: bright red regions are close to the surface and were consistently identified across participants.

the exact point where the test error is no longer below chance, that is, the point where all informative voxels are discarded and the remaining voxels do no longer carry information. We pursued this estimation by fitting a sigmoid function,

$$f(x) = \alpha_0 + \frac{\alpha_1}{1 + e^{-\frac{x - \alpha_2}{\alpha_3}}},$$

to the test error curve of the SVM (see Figure 4d). The coefficients α_0 to α_3 were estimated using least-squares approximation. The point of intersection with chance level of a straight line through the inflexion point (with the same slope as the sigmoid at that point) was used as an estimate of the upper limit of informative voxels in the data set. In the button press data set, this point was reached at approximately 34% discarded voxels. Although we do not know the true number of informative voxels in the button-press data set, visual inspection of the distribution of the identified voxels provides some preliminary evidence for the validity of this estimate. Figure 6 shows the distribution of discarded voxels over repetitions. The second-last plot marks the point where all (even weakly) informative voxels are identified according to the point-of-intersection criterion (34%). As can be seen, these voxels were mainly located in two dense clusters in the motor and somatosensory cortex of both hemispheres, where button-press-related activity would be expected. Consistent with our assumption that pushing the rSFM beyond this limit would return uninformative voxels, voxels in the next plot (50%) are more or less scattered across the whole brain.

Taken together, the results indicate that the SFM identifies informative features effectively even if the data set contains several informative feature subsets that all permit linear separation (such as the fMRI data set used here). Furthermore, the SFM seems to be a promising tool to estimate the total number of informative voxels (or, more generally, features) in highly redundant high-dimensional small-sample-size data sets. Further experiments and validations in different directions will be a next step.

5 Summary and Conclusion

The goal of this letter was threefold. First, we reformulated the concept of the SFM to make it applicable to a wide range of different scenarios. We then showed that the SFM allows identifying relevant features very effectively and, in many cases, more accurately than SVM-based feature selection, particularly in high-dimensional small-sample-size scenarios. Finally, we applied a repetitive version of the SFM to an fMRI data set to illustrate how the SFM can be used to identify not only the most informative features of a data set but also to estimate the total number of informative features in this data set.

In appendix A we derive a condition that is necessary for both the SFM and Weston's approach to converge to the zero-norm minimizing solution. With this condition, we provide additional plausibility considerations as to why the SFM finds the least number of features with higher probability than Weston's approach.

In sum, we think we have shown that the SFM is both an effective and, as shown in appendix B, efficient method for feature selection that will open new avenues for data analysis in many functional biological applications, including the rapidly growing field of information-based neuroimaging. Open issues for further research include possible extensions of the SFM to nonlinear classification problems and multiple classes, as well as a more comprehensive comparison with existing approaches in neuroimaging applications.

Finally, we note that we see the SFM as a method that might be used in different ways for repetitive feature selection. For example, if one is not primarily interested in identifying the total number of informative features but to find those features that permit most accurate classification, a slightly different rSFM might be useful: as before, the SFM will be trained on the complete data set in the first repetition, but this time an SVM will be trained on all features found to be relevant. In the next repetition, features found to be relevant will be added to the feature set only if the classification of the SVM trained on all relevant features improves relative to the previous repetition. This heuristic aims at finding the smallest subset of features that contains all information, which is recommended for optimizing classification performance. However, the evaluation of this (and other) approaches requires comprehensive analysis and experiments, which are beyond the scope of this letter. They will be the focus of future studies.

Appendix: Mathematical Considerations and Implementation

We provide some mathematical considerations that give hints of the superiority of the SFM for feature selection compared to SVM-based methods such as the one proposed by Weston et al. (2003). Further, we provide technical details on how to transform the SFM into a standard linear program to be solved with conventional optimization packages.

A.1 Mathematical Considerations.

The SFM enforces linear separation with an additional constraint on the mean decision value to avoid the trivial solution $\boldsymbol{w} = \boldsymbol{0}$. However, it is not obvious why this method should be better suited to identify the minimum number of relevant features than, say, the closely related method by Weston et al. (2003). Both methods converge to a local minimum of the target problem, equation 2.2. On artificial and real-world data sets, we observe the SFM to identify relevant features more effectively than other SVM-based feature selection methods.

As we have seen in our simulations, in contrast to the approach of Weston et al., the SFM finds the relevant features basically with the first step. Weston's approach can hardly catch up with the following iterations. It seems that the first step is important and decides whether we will converge to a good minimum. The first step is equivalent to minimizing the one-norm instead of the zero-norm. In the following, we derive necessary conditions for finding a zero-norm solution by minimizing the one-norm (see also Klement & Martinetz, 2011)—for both the SFM and the related SVM-based method by Weston et al. Based on this condition, we explain why the SFM approach finds a zero-norm minimizing solution more frequently by comparing their behavior in a simple illustrative scenario.

First, we introduce some simplifications and notations to improve the readability of the admittedly complex plausibility considerations. In the following, we assume the data set $\mathcal{D}$ to be balanced and linearly separable without bias,

$$\exists\, \boldsymbol{w} \in \mathbb{R}^d \quad \text{with} \quad y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w} \geq 0 \ \forall\, i \quad \text{and} \quad \boldsymbol{w} \neq \boldsymbol{0}\,,$$

where the normal vector $\boldsymbol{w} \in \mathbb{R}^d$ describes the separating hyperplane except for a constant factor. For abbreviation, we define $\boldsymbol{z}_i = y_i \boldsymbol{x}_i$, $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$, and $\bar{\boldsymbol{z}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i = \frac{1}{2}(\frac{1}{n^+}\sum_{i\in I^+} \boldsymbol{x}_i - \frac{1}{n^-}\sum_{i\in I^-} \boldsymbol{x}_i) = \frac{1}{2}(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-)$. Using this notation, Weston et al. aim to

$$\text{minimize} \quad ||\boldsymbol{w}||_0 \quad \text{subject to} \quad \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} \geq \mathbf{1}\,, \tag{A.1}$$

while in the SFM setting, we aim to

$$\text{minimize} \quad ||\boldsymbol{w}||_0 \quad \text{subject to} \quad \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} \geq \mathbf{0} \quad \text{and} \quad \bar{z}^{\mathrm{T}}\boldsymbol{w} = \frac{1}{2}\,. \tag{A.2}$$

Both, equations A.1 and A.2, solve 2.2, but which setting does it more effectively in the first iteration when we replace the zero-norm by the one-norm,

$$\text{minimize} \quad ||\boldsymbol{w}||_1 \quad \text{subject to} \quad \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} \geq \mathbf{1}\,, \tag{A.3}$$

or, in the case of the SFM,

$$\text{minimize} \quad ||\boldsymbol{w}||_1 \quad \text{subject to} \quad \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} \geq \mathbf{0} \quad \text{and} \quad \bar{z}^{\mathrm{T}}\boldsymbol{w} = \frac{1}{2}\,. \tag{A.4}$$

First, we focus on equation A.1. When do we find a solution to this equation A.1 by solving equation A.3? We denote the solution space of equation A.1 by Ω and define the following two weight vectors:

$$\boldsymbol{w}_0 = \underset{\boldsymbol{w}\in\Omega}{\arg\min} \quad ||\boldsymbol{w}||_1 \quad \text{subject to} \quad \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} \geq \mathbf{1}, \tag{A.5}$$

$$\boldsymbol{w}_1 = \underset{\boldsymbol{w}\in\mathbb{R}^d}{\arg\min} \quad ||\boldsymbol{w}||_1 \quad \text{subject to} \quad \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} \geq \mathbf{1}. \tag{A.6}$$

$||\boldsymbol{w}_0||_0 = k$, that is, at least k features are necessary to separate the input data.

In the following, we assume $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ to be unique. This is only a minor restriction, as nonuniqueness will occur only in degenerate cases. Since $\boldsymbol{Z}$ is drawn from a probability distribution, the probability of these cases is of measure zero. The probabilistic nature of the input data also ensures that all quadratic submatrices of $\boldsymbol{Z}$ have full rank. Among all solutions of equation A.1, $\boldsymbol{w}_0$ is the solution with the lowest one-norm. Note that if $\boldsymbol{w}_1$ is in Ω, then $\boldsymbol{w}_1 = \boldsymbol{w}_0$. Since in practice, equation A.1 cannot be solved directly, Ω is in general unknown, as is $\boldsymbol{w}_0$. However, both are well defined. In contrast, $\boldsymbol{w}_1$ is the solution on the entire $\mathbb{R}^d$ and can efficiently be found by linear programming. If $\boldsymbol{w}_0 = \boldsymbol{w}_1$ for a specific data set, then the optimal feature set can be obtained by optimizing for the one-norm. Without loss of generality, for the following considerations, we assume:

1. All entries of the weight vector are positive: $w_{0,i} \geq 0$. Otherwise, we invert the corresponding input dimension.

2. The training data are ordered such that the design matrix $\boldsymbol{Z} = (\hat{\boldsymbol{Z}}\ \check{\boldsymbol{Z}})$ with $\hat{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{w}_0 = \mathbf{1}$ and $\check{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{w}_0 > \mathbf{1}$. So only the first columns of $\boldsymbol{Z}$ correspond to active constraints; the constraints are fulfilled with equality. Let k^* be the number of active constraints.
3. The dimensions of $\mathcal{D}$ are sorted such that exactly the first k dimensions of $\boldsymbol{w}_0$ are nonzero:

$$w_{0,i} \begin{cases} > 0 & i = 1, \ldots, k \\ = 0 & \text{otherwise} \end{cases} \quad \text{such that} \quad \boldsymbol{w}_0 = \begin{pmatrix} \hat{\boldsymbol{w}}_0 \\ \mathbf{0} \end{pmatrix}.$$

In total the design matrix $\boldsymbol{Z}$ is organized as

$$\boldsymbol{Z} = \begin{pmatrix} \hat{\boldsymbol{Z}}_1 & \check{\boldsymbol{Z}} \\ \hat{\boldsymbol{Z}}_2 & \end{pmatrix} \quad \text{with} \quad \hat{\boldsymbol{Z}}_1 \in \mathbb{R}^{k\times k^*}, \quad \hat{\boldsymbol{Z}}_2 \in \mathbb{R}^{d-k\times k^*}, \quad \check{\boldsymbol{Z}} \in \mathbb{R}^{d\times n-k^*},$$

where the dimensions n and d are known in advance. The following lemma holds for the relation between k and k^*:

Lemma 1. *If $\boldsymbol{w}_0$ contains k nonzero entries, exactly k equations in $\boldsymbol{Z}^T\boldsymbol{w}_0 \geq \mathbf{1}$ are active: $k = k^*$.*

Proof. In linear programming theory, a basic feasible solution is defined to be a solution located in one of the corners of the solution space defined by the constraints. The fundamental theorem of linear programming (found in many textbooks on linear programming, e.g., in Dantzig & Thapa, 2003; Vanderbei, 2008) states that if an optimal solution exists, then a basic optimal solution also exists. In other words, optimal solutions are located in the corners of the solution space, which is exploited by the simplex method for solving linear programming problems.

As stated before, equation A.1 may have multiple solutions. Each solution may involve a different set of features. Let Λ_i be the linear subspace spanned by the k features of a particular solution to equation A.1. Then $\Omega \subset \bigcup_i \Lambda_i$, and

$$\begin{aligned} \boldsymbol{w}_0 &= \underset{\boldsymbol{w}\in\Omega}{\arg\min} \quad ||\boldsymbol{w}||_1 \quad \text{subject to} \quad \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} \geq \mathbf{1} \\ &= \underset{\boldsymbol{w}\in\bigcup_i \Lambda_i}{\arg\min} \quad ||\boldsymbol{w}||_1 \quad \text{subject to} \quad \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} \geq \mathbf{1} \end{aligned}$$

is valid. Thus, $\boldsymbol{w}_0$ can be obtained by a sequence of linear programs. All of them are feasible and nondegenerate. Therefore, an optimal solution exists, and it is a basic optimal solution of a linear program. The weight vector $\boldsymbol{w}$ contains k nonzero entries, so $\hat{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{w}_0 = \hat{\boldsymbol{Z}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}_0 = \mathbf{1}$, that is, the initial

d-dimensional problem is equivalent to a k-dimensional one. In a basic solution, k constraints are active and, hence, $k^* = k$ follows.

We proceed with the main theorem, which provides a necessary condition for finding $\boldsymbol{w}_0$ with a linear program (i.e., by solving equation A.6).

Theorem 1. *For $\boldsymbol{w}_1 = \boldsymbol{w}_0$, it is necessary that* $\left\|\hat{\boldsymbol{Z}}_2\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\left(\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\right)^{-1}\mathbf{1}\right\|_\infty < 1$.

Proof. If $\boldsymbol{w}_0 = \boldsymbol{w}_1$, for each infinitesimal disparity vector $\boldsymbol{\Delta}$ with $\hat{\boldsymbol{Z}}^{\mathrm{T}}(\boldsymbol{w}_0 + \boldsymbol{\Delta}) = \mathbf{1}$ and $\check{\boldsymbol{Z}}^{\mathrm{T}}(\boldsymbol{w}_0 + \boldsymbol{\Delta}) > \mathbf{1}$, we have

$$\begin{aligned}
&||\boldsymbol{w}_0 + \boldsymbol{\Delta}||_1 > ||\boldsymbol{w}_0||_1 \\
&\quad\Leftrightarrow \sum_{i=1}^{d} |w_{0,i} + \Delta_i| > \sum_{i=1}^{d} |w_{0,i}| = \sum_{i=1}^{d} w_{0,i} \\
&\quad\Leftrightarrow \sum_{i=1}^{k} |w_{0,i} + \Delta_i| + \sum_{i=k+1}^{d} |\underbrace{w_{0,i}}_{=0} + \Delta_i| > \sum_{i=1}^{k} w_{0,i} \\
&\quad\Leftrightarrow \sum_{i=1}^{k} w_{0,i} + \Delta_i + \sum_{i=k+1}^{d} |\Delta_i| > \sum_{i=1}^{k} w_{0,i} \\
&\quad\Leftrightarrow \sum_{i=1}^{k} \Delta_i + \sum_{i=k+1}^{d} |\Delta_i| > 0\,.
\end{aligned} \tag{A.7}$$

Next, we make use of the particular structure of the matrix $\hat{\boldsymbol{Z}}$ and split the disparity vector into an upper and a lower part: $\boldsymbol{\Delta}^{\mathrm{T}} = \left(\boldsymbol{\Delta}_1^{\mathrm{T}}\boldsymbol{\Delta}_2^{\mathrm{T}}\right)$ with $\boldsymbol{\Delta}_1 \in \mathbb{R}^k$, $\boldsymbol{\Delta}_2 \in \mathbb{R}^{d-k}$. A closed formulation for $\boldsymbol{\Delta}_1$ is derived by rearrangement and using $\hat{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{\Delta} = 0$:

$$\begin{aligned}
&\hat{\boldsymbol{Z}}^{\mathrm{T}}\boldsymbol{\Delta} = \hat{\boldsymbol{Z}}_1^{\mathrm{T}}\boldsymbol{\Delta}_1 + \hat{\boldsymbol{Z}}_2^{\mathrm{T}}\boldsymbol{\Delta}_2 = \mathbf{0} \\
&\quad\Leftrightarrow \hat{\boldsymbol{Z}}_1^{\mathrm{T}}\boldsymbol{\Delta}_1 = -\hat{\boldsymbol{Z}}_2^{\mathrm{T}}\boldsymbol{\Delta}_2 \\
&\quad\Leftrightarrow \hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\boldsymbol{\Delta}_1 = -\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_2^{\mathrm{T}}\boldsymbol{\Delta}_2 \\
&\quad\Leftrightarrow \boldsymbol{\Delta}_1 = -\left(\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\right)^{-1}\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_2^{\mathrm{T}}\boldsymbol{\Delta}_2 \\
&\quad\Rightarrow \mathbf{1}^{\mathrm{T}}\boldsymbol{\Delta}_1 = -\underbrace{\mathbf{1}^{\mathrm{T}}\left(\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\right)^{-1}\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_2^{\mathrm{T}}}_{:=\boldsymbol{\alpha}^{\mathrm{T}}}\boldsymbol{\Delta}_2
\end{aligned} \tag{A.8}$$

Finally, equation A.7 can be expressed using $\boldsymbol{\alpha}$ and $\boldsymbol{\Delta}_2$:

$$\sum_{i=1}^{k} \Delta_i + \sum_{i=k+1}^{d} |\Delta_i| = -\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\Delta}_2 + ||\boldsymbol{\Delta}_2||_1 = \sum_{i=k+1}^{d} -\alpha_{i-k}\Delta_i + |\Delta_i| > 0. \tag{A.9}$$

Equation A.9 has to hold for any infinitesimal $\boldsymbol{\Delta}_2$, which is the case if and only if $|\alpha_{i-k}| < 1$ holds for all $k+1 \leq i \leq d$, that is, if and only if

$$||\boldsymbol{\alpha}||_\infty = \left\| \hat{\boldsymbol{Z}}_2 \hat{\boldsymbol{Z}}_1^{\mathrm{T}} \left(\hat{\boldsymbol{Z}}_1 \hat{\boldsymbol{Z}}_1^{\mathrm{T}} \right)^{-1} \mathbf{1} \right\|_\infty < 1. \tag{A.10}$$

($\boldsymbol{\Delta}_2 = \mathbf{0}$ is excluded according to equation A.8 since then $\boldsymbol{\Delta} = \mathbf{0}$.)

So far, all considerations apply for equation A.1. However, with the following minor changes, a similar condition can be derived for the SFM problem, equation A.2:

1. The design matrix $\boldsymbol{Z}$ is extended by an additional column, the vector $\bar{\boldsymbol{z}}$.
2. The weight vectors $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ are defined analogously:

$$\boldsymbol{w}_0 = \arg\min_{\boldsymbol{w}\in\Omega} ||\boldsymbol{w}||_1 \;\text{ subject to } \left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\right)^{\mathrm{T}} \boldsymbol{w} \geq \mathbf{0} \;\text{ and } \bar{\boldsymbol{z}}^{\mathrm{T}}\boldsymbol{w} = \frac{1}{2},$$

$$\boldsymbol{w}_1 = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} ||\boldsymbol{w}||_1 \;\text{ subject to } \left(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\right)^{\mathrm{T}} \boldsymbol{w} \geq \mathbf{0} \;\text{ and } \bar{\boldsymbol{z}}^{\mathrm{T}}\boldsymbol{w} = \frac{1}{2}.$$

3. If $\boldsymbol{w}_0$ contains k nonzero entries, exactly k constraints are active. The last of these constraints is the equality constraint $\bar{\boldsymbol{z}}^{\mathrm{T}}\boldsymbol{w} = 1/2$.
4. The design matrix $\boldsymbol{Z}$ and the weight vector $\boldsymbol{w}_0$ are ordered in the same way as before: the first k entries of $\boldsymbol{w}_0$ are nonzero, and the first k columns of $\boldsymbol{Z}$ correspond to active constraints. The kth column is $\bar{\boldsymbol{z}}$.

Theorem 1 and its proof now apply exactly in the same way to the SFM.

Both approaches are very closely connected; however, they are not identical. The slight difference leads to a significantly lower number of features, as we have seen in the experiments. Due to the complexity of both approaches, it is not possible to give a rigorous mathematical proof for the superior performance of the SFM, equation A.2, compared to Weston's approach, equation A.1. However, within a simplified scenario and with approximate arguments, we can use the result of theorem 1 to make superior performance plausible.

We consider the simplest scenario analog to our preliminary experiments in section 3.1. Assume the elements of each vector $\boldsymbol{z}_i$ to be drawn from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with the expected value:

$$\mu = \begin{cases} c & i = 1, \ldots, k \\ 0 & \text{otherwise} \end{cases}$$

Thus, only the first k features are relevant; all others are irrelevant. For Weston's approach, equation A.1, we have $\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}_0 = \mathbf{1}$ and obtain

$$\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}_0 = \hat{\boldsymbol{Z}}_1\mathbf{1} \approx k \cdot c \cdot \mathbf{1} \Leftrightarrow \hat{\boldsymbol{w}}_0 \approx k \cdot c \cdot \left(\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\right)^{-1}\mathbf{1}$$

such that

$$||\boldsymbol{\alpha}||_\infty = \left\|\hat{\boldsymbol{Z}}_2\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\left(\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\right)^{-1}\mathbf{1}\right\|_\infty \approx \left\|\frac{\hat{\boldsymbol{Z}}_2\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}_0}{k \cdot c}\right\|_\infty = \left\|\frac{\hat{\boldsymbol{Z}}_2\mathbf{1}}{k \cdot c}\right\|_\infty = \left\|\frac{\boldsymbol{\varepsilon}_k}{c}\right\|_\infty .$$

The entries of the vector $\boldsymbol{\varepsilon}_k$ are distributed as $\mathcal{N}(0, \frac{\sigma^2}{k})$. In contrast, for the SFM, equation A.2, where the last column of $\hat{\boldsymbol{Z}}$ is the mean of all $\boldsymbol{z}_i$, we have $\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}_0 = \frac{1}{2}\binom{\mathbf{0}}{1}$ and obtain

$$\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}_0 = \hat{\boldsymbol{Z}}_1\,\frac{1}{2}\begin{pmatrix}\mathbf{0}\\1\end{pmatrix} \approx \frac{c}{2} \cdot \mathbf{1} \Leftrightarrow \hat{\boldsymbol{w}}_0 \approx \frac{c}{2} \cdot \left(\hat{\boldsymbol{Z}}_1\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\right)^{-1}\mathbf{1}$$

and

$$||\boldsymbol{\alpha}||_\infty \approx \left\|\frac{\hat{\boldsymbol{Z}}_2\hat{\boldsymbol{Z}}_1^{\mathrm{T}}\hat{\boldsymbol{w}}_0}{c/2}\right\|_\infty = \left\|\frac{\hat{\boldsymbol{Z}}_2}{c}\begin{pmatrix}\mathbf{0}\\1\end{pmatrix}\right\|_\infty = \left\|\frac{\boldsymbol{\varepsilon}_n}{c}\right\|_\infty .$$

Here, the entries of $\boldsymbol{\varepsilon}_n$ are distributed as $\mathcal{N}(0, \frac{\sigma^2}{n})$. Obviously, for $k \ll n$, the probability that all elements of $\boldsymbol{\alpha}$ stay below 1 and, hence, that the condition in theorem 1 is fulfilled, is much larger for the SFM. As expected, the larger c is, the easier it is for both approaches to be successful. Note that we assumed that the elements of $\hat{\boldsymbol{Z}}_1$ and $\hat{\boldsymbol{Z}}_2$ are independent stochastic variables. Of course, since $\hat{\boldsymbol{Z}}_1$ and $\hat{\boldsymbol{Z}}_2$ are selected by the respective algorithm according to certain criteria, this is not really the case.

To summarize, the above considerations are not a proof for a superior performance of the SFM on every data set; however, it provides some insight into why we observe it to identify relevant features more effectively.

A.2 Implementation. Next, we show how the mathematical formulations are transformed to be solved by conventional optimization frameworks. Given a particular optimization problem, it is in general impossible to say beforehand which of the numerous commercially or freely available solvers will perform best. Therefore, we chose four major optimization packages and performed an empirical analysis of their performance in solving the SFM. Of course, this list is only a small excerpt of the confusingly vast world of linear programming toolboxes; however, they cover the main concepts and algorithms, including the simplex algorithm, interior-point methods, presolvers, and others.

First, the SFM needs to be transformed into the standard linear program:

$$\begin{aligned} &\text{minimize} && \boldsymbol{f}^{\mathrm{T}}\boldsymbol{x} \\ &\text{subject to} && \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ &&& \boldsymbol{A}^{\mathrm{eq}}\boldsymbol{x} = \boldsymbol{b}^{\mathrm{eq}} \\ &&& \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\,. \end{aligned}$$

We derive two alternative formulations that differ in size and sparsity of the constraint matrices. Depending on the problem size—the dimension and the sample size—either of the alternatives may be better suited. The iterative SFM algorithm is require to solve the nonlinear optimization problem:

$$\begin{aligned} &\text{minimize} && |\boldsymbol{w}|_1 \\ &\text{subject to} && y_i\big(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\big) \geq 0 \quad \text{for all } i \\ &&& \boldsymbol{w}^{\mathrm{T}}\big(\boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-}\big) = 1. \end{aligned} \tag{A.11}$$

For linearization, each entry of the weight vector is split into a positive and a negative component where only one may be active: $w_i = w_i^{+} - w_i^{-}$ with $w_i^{+}, w_i^{-} \geq 0$ and $w_i^{+} \cdot w_i^{-} = 0$. As either w_i^{+} or w_i^{-} or both are 0, $|w_i| = w_i^{+} + w_i^{-}$ holds. Thus, equation A.11 is transformed into a linear program, where we seek to

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{d} w_i^{+} + w_i^{-} \\ &\text{subject to} && y_i\left(\sum_{j=1}^{d}\big(w_j^{+} - w_j^{-}\big)x_{ij} + b\right) \geq 0 \quad \text{for all } i \\ &&& \sum_{j=1}^{d}\big(w_j^{+} - w_j^{-}\big)\big(\mu_j^{+} - \mu_j^{-}\big) = 1 \\ &&& w_i^{+}, w_i^{-} \geq 0 \quad \text{for all } i. \end{aligned} \tag{A.12}$$

Note that the constraint $w_i^+ \cdot w_i^- = 0$ is not required. Assume the optimal solution is found and both variables take positive values. Then one could reduce each of them by $\min(w_i^+, w_i^-)$ without affecting w_i; the overall weight vector stays the same. However, the objective function is reduced by $2 \cdot \min(w_i^+, w_i^-)$, which is a contradiction to the initial assumption of w_i^+ and w_i^- being optimal. The input matrices and vectors for equation A.12 take the following values:

$$\begin{aligned}
\boldsymbol{f} &= \begin{pmatrix} \mathbf{1}^{\mathrm{T}} & \mathbf{1}^{\mathrm{T}} & 0 \end{pmatrix}^{\mathrm{T}} \in \mathbb{R}^{2d+1} \quad \text{(objective function)} \\
\boldsymbol{x} &= \begin{pmatrix} \boldsymbol{w}^{+\mathrm{T}} & \boldsymbol{w}^{-\mathrm{T}} & b \end{pmatrix} \quad \text{(target variable)} \\
\boldsymbol{A} &= \begin{pmatrix} -y_1\boldsymbol{x}_1^{\mathrm{T}} & y_1\boldsymbol{x}_1^{\mathrm{T}} & -y_1 \\ \vdots & \vdots & \vdots \\ -y_n\boldsymbol{x}_n^{\mathrm{T}} & y_n\boldsymbol{x}_n^{\mathrm{T}} & -y_n \end{pmatrix} \in \mathbb{R}^{n\times 2d+1} \quad \text{(inequality constraint)} \\
\boldsymbol{b} &= \mathbf{0} \in \mathbb{R}^{n} \\
\boldsymbol{A}^{\mathrm{eq}} &= \begin{pmatrix} \boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-} & \boldsymbol{\mu}^{-} - \boldsymbol{\mu}^{+} & 0 \end{pmatrix} \in \mathbb{R}^{1\times 2d+1} \quad \text{(equality constraint)} \\
\boldsymbol{b}^{\mathrm{eq}} &= 1
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{l} &= \begin{pmatrix} \mathbf{0}^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} & -\infty \end{pmatrix} \quad \text{(lower bounds of the variables)} \\
\boldsymbol{u} &= \begin{pmatrix} \infty \cdots \infty & \infty \cdots \infty & \infty \end{pmatrix} \quad \text{(upper bounds of the variables)}
\end{aligned}$$

Here, two conflicting variable naming schemes are mixed to avoid uncommon notations. So, $\boldsymbol{x}$, the variable to be optimized, should not be confused with the input data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, and the bias b is to be distinguished from the equality constraint vector $\boldsymbol{b}$. We use ∞ and $-\infty$ to indicate that the variables have no upper or lower bound, respectively. The above formulation is memory inefficient, as it requires the inequality constraint matrix to be stored twice: once with a positive and once with a negative sign. The second minor issue is related to the number of nonzero entries in the constraint matrices. In general, linear programming solvers are most efficient on sparse matrices, and problem formulations should minimize redundancy. In our case, we should seek an alternative formulation involving the training data only once. The key idea is the substitution $s_i = w_i^+ + w_i^-$. Thus, we get

$$w_i^+ = s_i - w_i^- \Rightarrow \; w_i = w_i^+ - w_i^- = s_i - 2w_i^- \Rightarrow \; \frac{1}{2}\left(s_i - w_i\right) = w_i^- \geq 0$$

and, vice versa,

$$w_i^- = s_i - w_i^+ \Rightarrow \; w_i = w_i^+ - w_i^- = 2w_i^+ - s_i \Rightarrow \; \frac{1}{2}\left(s_i + w_i\right) = w_i^+ \geq 0.$$

The transformed optimization problem,

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{d} s_i \\
\text{subject to} \quad & y_i\big(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\big) \geq 0 \quad \text{for all } i \\
& \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-}\right) = 1 \\
& s_i + w_i \geq 0 \quad \text{for all } i \\
& s_i - w_i \geq 0 \quad \text{for all } i,
\end{aligned}$$

has the same optimum but is memory efficient and much sparser. The input matrices and vectors now take the following values:

$$\boldsymbol{f} = \begin{pmatrix} \boldsymbol{0}^{\mathrm{T}} & \boldsymbol{1}^{\mathrm{T}} & 0 \end{pmatrix}^{\mathrm{T}} \in \mathbb{R}^{2d+1} \quad \text{(objective function)}$$

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{w}^{\mathrm{T}} & \boldsymbol{s}^{\mathrm{T}} & b \end{pmatrix} \quad \text{(target variable)}$$

$$\boldsymbol{A} = \begin{pmatrix} -y_1\boldsymbol{x}_1^{\mathrm{T}} & \boldsymbol{0}^{\mathrm{T}} & -y_1 \\ \vdots & \vdots & \vdots \\ -y_n\boldsymbol{x}_n^{\mathrm{T}} & \boldsymbol{0}^{\mathrm{T}} & -y_n \\ \boldsymbol{I}_d & -\boldsymbol{I}_d & \boldsymbol{0} \\ -\boldsymbol{I}_d & -\boldsymbol{I}_d & \boldsymbol{0} \end{pmatrix} \in \mathbb{R}^{(n+2d)\times(2d+1)} \quad \text{(inequality constraint)}$$

$$\boldsymbol{b} = \boldsymbol{0} \in \mathbb{R}^{n}$$

$$\boldsymbol{A}^{\mathrm{eq}} = \begin{pmatrix} \boldsymbol{\mu}^{+} - \boldsymbol{\mu}^{-} & \boldsymbol{0}^{\mathrm{T}} & 0 \end{pmatrix} \in \mathbb{R}^{1\times 2d+1} \quad \text{(equality constraint)}$$

$$\boldsymbol{b}^{\mathrm{eq}} = 1$$

$$\boldsymbol{l} = \begin{pmatrix} -\infty \cdots -\infty & \boldsymbol{0}^{\mathrm{T}} & -\infty \end{pmatrix} \quad \text{(lower bounds of the variables)}$$

$$\boldsymbol{u} = \begin{pmatrix} -\infty \cdots & \infty & \infty \cdots \infty & \infty \end{pmatrix} \quad \text{(upper bounds of the variables)}$$

In the first approach, the constraint matrix $\boldsymbol{A}$ has $n(2d+1)$ nonzero entries, while in the reformulated version, $n(d+1) + 4d$ entries are nonzero. Thus, for $n > 4$, the reformulated version has fewer entries. However, the complexity of linear programming solvers due to numerous processing steps—presolving, scaling, solving—makes an a priori run-time prediction impossible. So other less obvious aspects than the number of nonzero entries might get important in practice and require an empirical run-time evaluation.

The soft SFM approach, equation 2.9, is reformulated in the same way. In the initial problem,

$$\begin{aligned}
\text{minimize} \quad & \|\boldsymbol{w}\|_1 + C\|\boldsymbol{\xi}\|_1 \\
\text{subject to} \quad & y_i\big(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\big) \geq -\xi_i \\
& \boldsymbol{w}^{\mathrm{T}}\big(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-\big) = \pm 1 \\
& \xi_i \geq 0\,.
\end{aligned}$$

the substitution $w_i = w_i^+ - w_i^-$ with $w_i^+, w_i^- \geq 0$ leads to

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{d} w_i^+ + w_i^- + C\sum_{i=1}^{n} \xi_i \\
\text{subject to} \quad & y_i\left(\sum_{j=1}^{d} \big(w_j^+ - w_j^-\big)x_{ij} + b\right) \geq -\xi_i \\
& \sum_{j=1}^{d} \big(w_j^+ - w_j^-\big)\big(\mu_j^+ - \mu_j^-\big) = \pm 1 \\
& w_i^+, w_i^-, \xi_i \geq 0 \quad \text{for all } i.
\end{aligned}$$

Again, the size and the structure of the inequality constraint matrix $\boldsymbol{A}$ are the crucial factors:

$$\boldsymbol{A} = \begin{pmatrix} -y_1\boldsymbol{x}_1^{\mathrm{T}} & y_1\boldsymbol{x}_1^{\mathrm{T}} & & -y_1 \\ \vdots & \vdots & -\boldsymbol{I}_n & \vdots \\ -y_n\boldsymbol{x}_n^{\mathrm{T}} & y_n\boldsymbol{x}_n^{\mathrm{T}} & & -y_n \end{pmatrix} \in \mathbb{R}^{n\times 2d+n+1}. \tag{A.13}$$

Here, $n(2d+2)$ entries are nonzero, and as in the hard-margin case, the input data need to be stored twice. This is again avoided by substituting $s_i = w_i^+ + w_i^-$ to get the linear program:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{d} s_i + C\sum_{i=1}^{n} \xi_i \\
\text{subject to} \quad & y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq -\xi_i \\
& \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-\right) = \pm 1 \\
& s_i + w_i \geq 0 \\
& s_i - w_i \geq 0 \\
& \xi_i \geq 0 \quad \text{for all } i.
\end{aligned}$$

Now the inequality constraint matrix is

$$\boldsymbol{A} = \begin{pmatrix} -y_1\boldsymbol{x}_1^{\mathrm{T}} & & & -y_1 \\ \vdots & \boldsymbol{0}_{n,d} & -\boldsymbol{I}_n & \vdots \\ -y_n\boldsymbol{x}_n^{\mathrm{T}} & & & -y_n \\ -\boldsymbol{I}_d & -\boldsymbol{I}_d & \boldsymbol{0}_{d,n} & \boldsymbol{0} \\ \boldsymbol{I}_d & -\boldsymbol{I}_d & \boldsymbol{0}_{d,n} & \boldsymbol{0} \end{pmatrix} \in \mathbb{R}^{(n+2d)\times(2d+n+1)} \tag{A.14}$$

with $4d + dn + 2n$ nonzero entries.

Technical issues. As with many other machine learning algorithms, normalization is an essential preprocessing step for any of the proposed SFM variants. For all experiments, we normalized the training data sets to zero mean and unit variance and finally scaled all vectors to have a mean norm of one. This last step is sometimes beneficial in high-dimensional scenarios to keep the outcome of scalar products in a reasonable range. The test sets were normalized according to the factors obtained from the corresponding training sets.

For hard SFMs, either no solution exists or a solution where all data points are correctly classified. Since the optimizer uses numerical approximation methods with certain accuracy thresholds, some constraints may be marginally violated. Thus, some data points may be located on the wrong side of the hyperplane, but very close to it, producing a nonzero training error even in the hard case.

To avoid numerical issues, numbers that differed by no more than a specific implementation-dependent number—the machine epsilon—were considered to be equal.

Acknowledgments

We thank Jana Heussen for help with fMRI scanning. This work was partly funded by the Bundesministerium für Bildung und Forschung (Federal Ministry of Education and Research, grant 01GQ1105 to S.A.).

References

Carlson, T. A., Schrater, P., & He, S. (2003). Patterns of activity in the categorical representations of objects. *J. Cognitive Neuroscience*, *15*(5), 704–717.

Carroll, M. K., Cecchi, G. A., Rish, I., Garg, R., & Rao, A. R. (2009). Prediction and interpretation of distributed neural activity with sparse models. *NeuroImage*, *44*(1), 112–122.

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, *6*, 1–6.

Dantzig, G., & Thapa, M. (2003). *Linear programming, theory and extensions*. New York: Springer-Verlag.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, *286*(5439), 531–537.

Grosenick, L., Greer, S., & Knutson, B. (2008). Interpretable classifiers for FMRI improve prediction of purchases. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *16*(6), 539–548.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Hastie, T., Rosset, S., Tibshirani, R., & Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, *5*, 1391–1415.

Haynes, J.-D. (2011). Special issue on multivariate decoding and brain reading. *NeuroImage*, *56*.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*, 1263–1284.

Japkowicz, N. (2000). Learning from imbalanced data sets: A comparison of various strategies. In *Learning from imbalanced data sets: Papers from the AAAI Workshop* (Tech. Rep. WS-00-05). AAAI Press.

Klement, S., & Martinetz, T. (2010a). A new approach to classification with the least number of features. In *Proceedings of the 9th International Conference on Machine Learning and Applications* (pp. 141–146). San Mateo, CA: IEEE Computer Society.

Klement, S., & Martinetz, T. (2010b). The support feature machine for classifying with the least number of features. In K. I. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Lecture Notes in Computer Science*: *Vol. 6353*. *Artificial neural networks* (pp. 88–93). Berlin: Springer-Verlag.

Klement, S., & Martinetz, T. (2011). On the problem of finding the least number of features by L1-norm minimisation. In T. Honkela (Ed.), *Lecture Notes in Computer Science: Vol. 6791*. *Proceedings of the 21st International Conference on Artificial Neural Networks* (pp. 315–322). Berlin: Springer-Verlag.

Lockhart, D. J., & Winzeler, E. (2000). Genomics, gene expression and DNA arrays. *Nature*, *405*, 827–836.

Martino, F. D., Valente, G., Staeren, N., Ashburner, J., Goebel, R., & Formisano, E. (2008). Combining multivariate voxel selection and support vector machines for mapping and classification of FMRI spatial patterns. *NeuroImage*, *43*(1), 44–58.

McPherson, R., Pertsemlidis, A., Kavaslar, N., Stewart, A., Roberts, R., Cox, D. R., et al. (2007). A common allele on chromosome 9 associated with coronary heart disease. *Science*, *316*(5830), 1488–1491.

Nichols, T. E. (2012). Multiple testing corrections, nonparametric methods, and random field theory. *NeuroImage*, *62*(2), 811–815.

Provost, F. (2000). Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI-2000 Workshop on Imbalanced Data Sets*. Palo Alto, CA: AAAI.

Raelson, J. V., Little, R. D., Ruether, A., Fournier, H., Paquin, B., Van Eerdewegh, P., et al. (2007). Genome-wide association study for Crohn's disease in the Quebec founder population identifies multiple validated disease loci. *Proceedings of the National Academy of Sciences*, *104*(37), 14747–14752.

Rasmussen, P. M., Hansen, L. K., Madsen, K. H., Churchill, N. W., & Strother, S. C. (2012). Model sparsity and brain pattern interpretation of classification models in neuroimaging. *Pattern Recognition*, *45*(6), 2085–2100.

Ryali, S., Supekar, K., Abrams, D. A., & Menon, V. (2010). Sparse logistic regression for whole-brain classification of FMRI data. *NeuroImage*, *51*(2), 752–764.

Samani, N. J., Erdmann, J., Hall, A. S., Hengstenberg, C., Mangino, M., Mayer, B., et al. (2007). Genomewide association analysis of coronary artery disease. *New England Journal of Medicine*, *357*(5), 443–453.

Schmahmann, J. D., Doyon, J., Mcdonald, D., Holmes, C., Lavoie, K., Hurwitz, A. S., et al. (1998). Three-dimensional MRI atlas of the human cerebellum in proportional stereotaxic space. *NeuroImage*, *10*, 233–260.

Strother, S. C., Anderson, J., Hansen, L. K., Kjems, U., Kustra, R., Sidtis, J., et al. (2002). The quantitative evaluation of functional neuroimaging experiments: the NPAIRS data analysis framework. *Neuroimage*, *15*, 747–771.

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., et al. (2002). Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *NeuroImage*, *15*(1), 273–289.

Vanderbei, R. (2008). *Linear programming: Foundations and extensions*. Berlin: Springer-Verlag.

Vapnik, V. N. (1999). *The nature of statistical learning theory*. Berlin: Springer.

Weston, J., Elisseeff, A., Schölkopf, B., & Tipping, M. (2003). Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, *3*, 1439–1461.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for SVMs. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems*, *13*. Cambridge, MA: MIT Press.

Yamashita, O., Sato, M.-a., Yoshioka, T., Tong, F., & Kamitani, Y. (2008). Sparse estimation automatically selects voxels relevant for the decoding of FMRI activity patterns. *NeuroImage*, *42*(4), 1414–1429.

Received June 13, 2012; accepted December 27, 2012.