
Machine Learning, 61, 129–150, 2005

DOI: 10.1007/s10994-005-1505-9

Combined SVM-Based Feature Selection and Classification

JULIA NEUMANN neumann@uni-mannheim.de
CHRISTOPH SCHNÖRR schnörr@uni-mannheim.de
GABRIELE STEIDL steidl@uni-mannheim.de
Department of Mathematics and Computer Science, University of Mannheim, D-68131 Mannheim, Germany

Editor: Dale Schuurmans
Published online: 11 July 2005

Abstract. Feature selection is an important combinatorial optimisation problem in the context of supervised pattern classification. This paper presents four novel continuous feature selection approaches directly minimising the classifier performance. In particular, we include linear and nonlinear Support Vector Machine classifiers. The key ideas of our approaches are additional regularisation and embedded nonlinear feature selection. To solve our optimisation problems, we apply difference of convex functions programming which is a general framework for non-convex continuous optimisation. Experiments with artificial data and with various real-world problems including organ classification in computed tomography scans demonstrate that our methods accomplish the desired feature selection and classification performance simultaneously.

Keywords: feature selection, SVMs, embedded methods, mathematical programming, difference of convex functions programming, non-convex optimisation

1. Introduction

1.1. Overview and related work

In the context of supervised pattern classification, *feature selection* aims at picking out some of the original input dimensions (*features*) (i) for performance issues by facilitating data collection and reducing storage space and classification time, (ii) to perform semantics analysis helping to understand the problem, and (iii) to improve prediction accuracy by avoiding the "curse of dimensionality" (cf. Guyon & Elisseeff, 2003).

According to Guyon and Elisseeff (2003), John, Kohavi, and Pfleger (1994), and Bradley (1998), feature selection approaches divide into *filters*, *wrappers* and *embedded approaches*. Most known approaches are filters which act as a preprocessing step independently of the final classifier (Hermes & Buhmann, 2000; Duda, Hart, & Stork, 2000). In contrast, wrappers take the classifier into account as a black box (John, Kohavi, and Pfleger, 1994; Weston et al., 2001). An example for a wrapper method for nonlinear SVMs is Weston et al. (2001), where instead of minimising the classification error, the features are selected to minimise a generalisation error bound. Finally, embedded approaches simultaneously determine features and classifier during the training process. The embedded methods in

Bradley and Mangasarian (1998) are based on a linear classifier. As for the wrapper methods, there exist only few embedded methods addressing feature selection in connection with nonlinear classifiers up to now. An embedded approach for the quadratic 1-norm SVM was suggested in Zhu et al. (2004). The authors penalise the features by the ℓ_1-norm and apply the nonlinear mapping explicitly. This makes the approach feasible only for low dimensional feature maps such as the quadratic one. In particular, original features are not suppressed so that no performance improvements or semantics analysis are possible. Finally, in Jebara and Jaakkola (2000) a feature selection method was developed as an extension to the so-called maximum entropy discrimination, i.e., from a discriminative (probabilistic) perspective.

1.2. Contribution

In this work, we focus on embedded approaches for feature selection. The starting point for our investigation is the approach of Bradley and Mangasarian (1998) that minimises the training errors of a linear classifier while penalising the number of features by a concave penalty approximating the ℓ_0-"norm". In this way, the linear classifier is constructed while implicitly discarding features. The first objective of our work is to extend this feature selection approach with the aim to improve the generalisation performance of the classifiers. Taking into account that the *Support Vector Machine* (SVM) provides good generalisation ability by its ℓ_2 regulariser, we propose new methods by introducing additional regularisation terms.

In the second part of our work, we construct *direct objective minimising feature selection* methods for nonlinear SVM classifiers. First, we generalise the approach for the quadratic SVM of Zhu et al. (2004) in two directions. We apply the approximate ℓ_0 penalty considered superior to the ℓ_1-norm in Bradley and Mangasarian (1998) and we focus on feature selection in the *original* feature space to further improve the performance and enable semantics analysis. Next we incorporate "kernel-target alignment" (Cristianini et al., 2002) within this framework which performs appropriate feature selection if, e.g., the Gaussian kernel SVM is used as classifier. This approach is essentially different from multiple kernel learning techniques addressed, e.g., in Bach, Lanckriet, and Jordan (2004).

Some of our new approaches require the solution of non-convex optimisation problems. To solve these problems, we apply a general difference of convex functions (d.c.) optimisation algorithm in an appropriate way. Moreover, we show that the *Successive Linearization Algorithm* (SLA) proposed in Bradley and Mangasarian (1998) for concave minimisation is in effect a special case of our general optimisation approach. A short summary of our algorithms has been announced in Neumann, Schnörr, and Steidl (2004).

Feature selection is especially profitable for high-dimensional problems. To illustrate this, we investigate as part of our in-depth method evaluation the problem of selecting a suitable subset from 650 image features in order to classify organs in computed tomography (CT).

1.3. Organisation

After reviewing the linear embedded approaches proposed in Bradley and Mangasarian (1998), we introduce our enhanced approaches both for linear and nonlinear classification

in Section 3. The d.c. optimisation approach and its application to our feature selection problems is described in Section 4. Numerical results illustrating and evaluating various approaches, including the CT organ classification, are given in Section 5.

1.4. Notation

We denote vectors and matrices by bold small and capital letters, respectively. The matrix $\mathbf{I}$ denotes the identity matrix in appropriate dimensions. The vector $\mathbf{0}$ signifies a vector of zeros and $\mathbf{e}$ a vector of ones. All vectors will be column vectors unless transposed by the superior symbol ^{T}. If $\mathbf{x} \in \mathbb{R}^n$ denotes a vector, in general, we will indicate its components by x_i $(i = 1, \ldots, n)$. We set $|\mathbf{w}| := (|w_1|, |w_2|, \ldots)^T$ and assume vector inequalities to hold componentwise. Furthermore, $[-\mathbf{v}, \mathbf{v}]$ for $\mathbf{v} \in \mathbb{R}^d$ signifies the cuboid $\{\mathbf{w} \in \mathbb{R}^d : -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v}\}$. We use the function $x_+ := \max(x, 0)$ and the indicator function χ_C of a feasible convex set C which is defined by $\chi_C(x) = 0$ if $x \in C$, and $\chi_C(x) = \infty$ otherwise.

2. Classifier regularisation and feature penalties

Given a training set $\{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\} : i = 1, \ldots, n\}$ with $\mathcal{X} \subset \mathbb{R}^d$, the first goal is to find a classifier $F : \mathcal{X} \to \{-1, 1\}$. We will introduce in Section 2.1 the linear classifier on which the presented embedded feature selection approaches are based, and then add penalties for feature suppression and for improving the generalisation performance in Section 2.2.

2.1. Robust linear programming

Our starting point are linear classification approaches for constructing two parallel bounding hyperplanes in $\mathbb{R}^d$ such that the differently labelled sets are maximally located in the two opposite half spaces determined by these hyperplanes. More precisely, one solves the minimisation problem

$$f_{\mathrm{RLP}}(\mathbf{w}, b) := \sum_{i=1}^{n} (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))_+ \to \min_{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}}. \tag{1}$$

If $(\mathbf{w}, b)$ is the solution of (1), then the classifier is $F(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^T \mathbf{x} + b)$. The linear method (1) was proposed as *Robust Linear Programming* (RLP) by Bennett and Mangasarian (1992). Note that these authors weighted the training errors by $1/n_{\pm 1}$, where $n_{\pm 1} = |\{i : y_i = \pm 1\}|$.

2.2. Regularisation and feature penalties

In general, optimisation approaches to statistical classification include an additional penalty term ρ besides a "goodness of fit" term as f_{RLP} in (1) whose competition is controlled by

a weight parameter $\lambda \in [0, 1)$:

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} (1-\lambda) f_{\mathrm{RLP}}(\mathbf{w}, b) + \lambda\rho(\mathbf{w}). \tag{2}$$

In the following, we consider different penalties.

2.2.1. SVM. In order to maximise the margin between the two parallel hyperplanes, the original SVM penalises the ℓ_2-norm of $\mathbf{w}$. Then (2) yields

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} (1-\lambda) \sum_{i=1}^{n} (1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \tag{3}$$

which can be solved by a convex Quadratic Program (QP). The *Support Vectors* (SVs) are those patterns $\mathbf{x}_i$ for which the dual solution is positive, which implies $y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 1$.

2.2.2. ℓ_1-SVM. In order to suppress features, i.e. components of the vector $\mathbf{w}$, ℓ_p-norms of $\mathbf{w}$ with $p < 2$ are used as feature penalties. In Bradley and Mangasarian (1998), the ℓ_1-norm (lasso penalty) $\rho(\mathbf{w}) = \|\mathbf{w}\|_1$ led to good feature selection and classification results. Accordingly, (2) reads

$$\min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} (1-\lambda) \sum_{i=1}^{n} (1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ + \lambda\mathbf{e}^T|\mathbf{w}| \tag{4}$$

which can be solved by a linear program. This penalty term was originally introduced in the statistical context of linear regression in the 'lasso' ('Least Absolute Shrinkage and Selection Operator') in Tibshirani (1996), and also applied in Zhu et al. (2004).

2.2.3. Feature selection concave (FSV). Feature selection can be further improved by using the so-called ℓ_0-"norm" $\|\mathbf{w}\|_0^0 = |\{i : w_i \neq 0\}|$ (Bradley & Mangasarian, 1998; Weston et al., 2003). Note that $\|\cdot\|_0$ is no norm because, unlike ℓ_p-norms ($p \geq 1$), the triangle inequality does not hold. Since the ℓ_0-"norm" is non-smooth, it was approximated in Bradley and Mangasarian (1998) by the concave functional

$$\rho(\mathbf{w}) = \mathbf{e}^T\left(\mathbf{e} - e^{-\alpha|\mathbf{w}|}\right) \approx \|\mathbf{w}\|_0^0 \tag{5}$$

with approximation parameter $\alpha \in \mathbb{R}_+$. Problem (2) with penalty term (5) yields with suitable constraints the mathematical program:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n, \mathbf{v}\in\mathbb{R}^d} \quad & (1-\lambda)\mathbf{e}^T\boldsymbol{\xi} + \lambda\mathbf{e}^T(\mathbf{e} - e^{-\alpha\mathbf{v}}) \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n, \\ & \boldsymbol{\xi} \geq \mathbf{0}\,, \\ & -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v} \end{aligned} \tag{6}$$

which is known as *Feature Selection concaVe* (FSV). Note that this problem is non-convex, and high quality solutions can be obtained by, e.g., the Successive Linearization Algorithm (SLA) presented in Section 4.2.1.

3. New feature selection approaches

3.1. Combined ℓ_p penalties

FSV performs well for feature selection. However, its classification accuracy can be improved by applying a standard SVM on the selected features only, as shown in Jakubik (2003) and also indicated in Weston et al. (2003). Therefore, since the ℓ_2 penalty term is responsible for the very good SVM classification results while the ℓ_1 and ℓ_0 penalty terms focus on feature selection, we suggest combinations of these terms. Consequently, we need two weight parameters $\mu, \nu \in \mathbb{R}_+$.

3.1.1. ℓ_2-ℓ_1-SVM. For the ℓ_2-ℓ_1-*SVM*, we are interested in solving the constrained convex QP

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n, \mathbf{v}\in\mathbb{R}^d} \quad & \frac{\mu}{n}\mathbf{e}^T\boldsymbol{\xi} + \frac{1}{2}\mathbf{w}^T\mathbf{w} + \nu\mathbf{e}^T\mathbf{v} \\ \text{subject to} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n, \\ & \boldsymbol{\xi} \geq \mathbf{0}, \\ & -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v}. \end{aligned} \tag{7}$$

It is advisable here to solve the dual problem because it involves fewer variables and has a simpler structure, similar to the SVM case.

3.1.2. ℓ_2-ℓ_0-SVM. For the ℓ_2-ℓ_0-*SVM* with approximate ℓ_0-"norm", we minimise

$$f(\mathbf{w}, b, \mathbf{v}) := \frac{\mu}{n}\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ + \frac{1}{2}\mathbf{w}^T\mathbf{w} + \nu\mathbf{e}^T(\mathbf{e} - e^{-\alpha\mathbf{v}}) + \chi_{[-\mathbf{v},\mathbf{v}]}(\mathbf{w}). \tag{8}$$

An appropriate approach to optimise (8) is developed in Section 4.

3.2. Nonlinear classification

For problems which are not linearly separable a so-called *feature map* ϕ is commonly used which maps the set $\mathcal{X} \subset \mathbb{R}^d$ into a higher dimensional space $\phi(\mathcal{X}) \subset \mathbb{R}^{d'}$ ($d' \geq d$). Then a linear classification approach (1) or (3) can be applied in the new feature space $\phi(\mathcal{X})$. This results in a nonlinear classification in the original space $\mathbb{R}^d$, i.e., in nonlinear

separating surfaces. Below, we consider two popular feature maps in connection with feature selection.

3.2.1. Quadratic FSV. We start with the simple quadratic feature map

$$\begin{aligned}\phi : \mathcal{X} \to \mathbb{R}^{d'}, \quad \mathbf{x} &\mapsto \left(\mathbf{x}^{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathbb{N}_0^d, 0 < \|\boldsymbol{\alpha}\|_1 \leq 2\right) \\ &= \left(x_1^{\alpha_1} x_2^{\alpha_2} \dots x_d^{\alpha_d} : \boldsymbol{\alpha} \in \mathbb{N}_0^d, 0 < \|\boldsymbol{\alpha}\|_1 \leq 2\right),\end{aligned}$$

where $d' = \frac{d(d+3)}{2}$. Straightforward application of the ℓ_0 penalty in $\mathbb{R}^{d'}$ by FSV leads to the minimisation problem

$$(1-\lambda)\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b))_+ + \lambda \mathbf{e}^T(\mathbf{e} - e^{-\alpha \mathbf{v}}) + \sum_{i=1}^{d'} \chi_{[-v_i, v_i]}(w_i)$$

for $\mathbf{w} \in \mathbb{R}^{d'}$, $b \in \mathbb{R}$ and $\mathbf{v} \in \mathbb{R}^{d'}$. This approach, as well as a similar one for the ℓ_1 penalty in Zhu et al. (2004), achieve feature selection only in the *transformed* feature space $\mathbb{R}^{d'}$. Our goal, however, is to select features in the *original* space $\mathbb{R}^d$ in order to get insight into our original problem, too, and to reduce the number of primary features. To this end, instead of penalising v_i for $\mathbf{v} \in \mathbb{R}^{d'}$, we examine for each w_i $(i = 1, \dots, d')$ which original features are included in computing ϕ_i. If $\mathbf{e}_j \in \mathbb{R}^d$ denotes the jth unit vector and $\phi_i(\mathbf{e}_j) \neq 0$, we penalise the corresponding v_j for $\mathbf{v} \in \mathbb{R}^d$:

$$\begin{aligned}f(\mathbf{w}, b, \mathbf{v}) :&= (1-\lambda)\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b))_+ + \lambda \mathbf{e}^T(\mathbf{e} - e^{-\alpha \mathbf{v}}) \\ &\quad + \sum_{i=1}^{d'} \sum_{\phi_i(\mathbf{e}_j) \neq 0} \chi_{[-v_j, v_j]}(w_i) \to \min_{\mathbf{w} \in \mathbb{R}^{d'}, b \in \mathbb{R}, \mathbf{v} \in \mathbb{R}^d}. \end{aligned} \tag{9}$$

In the following, we refer to (9) as *quadratic FSV*. In principle, the approach can be extended to other explicit feature maps $\boldsymbol{\phi}$, especially by choosing other polynomial degrees.

In the same manner as done for FSV here, it is possible to generalise the ℓ_2-ℓ_p-SVMs for $p = 0, 1$ by explicitly applying, e.g., the quadratic feature map. This leads to solving a sequence of convex QPs instead of LPs as will be seen in the next section.

3.2.2. Kernel-target alignment approach. Compared with linear SVMs, further improvements of classification accuracy in our context may be achieved by using Gaussian kernel SVMs, as has been confirmed by experiments in Jakubik (2003). Therefore, we also consider SVMs with the feature map $\phi : \mathcal{X} \to \ell_2$ induced by $K(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle$ for the Gaussian kernel

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = \mathbf{e}^{-\|\mathbf{x}-\mathbf{z}\|_{2,\boldsymbol{\theta}}^2 / 2\sigma^2} \tag{10}$$

with weighted ℓ_2-norm $\|\mathbf{x}\|_{2,\boldsymbol{\theta}}^2 = \sum_{k=1}^{d} \theta_k |x_k|^2$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{X}$. As the feature space has infinite dimension, feature selection as done for the quadratic feature map is no longer

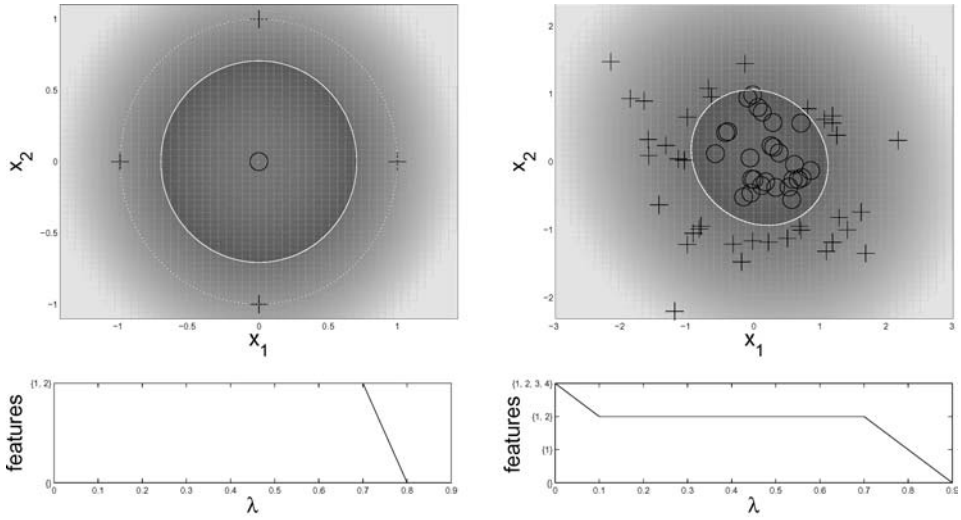

Figure 1. Quadratic classification problems with $y = \operatorname{sgn}(x_1^2 + x_2^2 - 1)$. *Top*: Training points and decision boundaries (*white lines*) computed by (9) for $\lambda = 0.1$, *left*: in $\mathbb{R}^2$, *right*: projection of $\mathbb{R}^4$ onto selected features. *Bottom*: Features determined by (9).

applicable. We apply the common SVM classifier without bias term b. For further information on nonlinear SVMs see, e.g., Schölkopf and Smola (2002). We obtain the commonly used kernel and classifier for $\boldsymbol{\theta} = \mathbf{e}$. Direct feature selection, i.e., the setting of as many θ_k to zero as possible while retaining or improving the classification ability, is a difficult problem. One possible approach is to use a wrapper as in Weston et al. (2001). Instead, we aim at directly maximising the alignment $\hat{A}(\mathbf{K}, \mathbf{y}\mathbf{y}^T) = \mathbf{y}^T\mathbf{K}\mathbf{y}/(n\|\mathbf{K}\|_F)$ proposed in Cristianini et al. (2002) as a measure of conformance of a kernel represented by $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^n$ with a learning task. To simplify this optimisation task, we drop the denominator which is justified in view of the boundedness of the kernel elements (10). To cope with unequal sample partitioning as, e.g., in Fig. 1 left, we replace $\mathbf{y}$ by $\mathbf{y}_n = (y_i/n_{y_i})_{i=1}^n$. This leads to

$$\mathbf{y}_n^T\mathbf{K}\mathbf{y}_n = \left\| \frac{1}{n_{+1}} \sum_{\{i:y_i=+1\}} \phi(\mathbf{x}_i) - \frac{1}{n_{-1}} \sum_{\{i:y_i=-1\}} \phi(\mathbf{x}_i) \right\|^2 \tag{11}$$

which is the class–centre distance in the feature space. A different view on the alignment criterion is obtained by considering the classifier F with $\mathbf{w} = \sum_{i=1}^n y_{ni}\phi(\mathbf{x}_i)$, $b = 0$ in feature space. Then maximising the correct class responses $\sum_{i=1}^n y_{ni}F(\mathbf{x}_i)$ also leads to the

expression above. Relaxing the binary $\boldsymbol{\theta} \in \{0,1\}^d$ to $\boldsymbol{\theta} \in [0,1]^d$ and adding penalty (5), we define as our *kernel-target alignment approach* to feature selection

$$f(\boldsymbol{\theta}) := -(1-\lambda)\frac{1}{2}\mathbf{y}_n^T \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n + \lambda \frac{1}{d}\mathbf{e}^T(\mathbf{e} - e^{-\alpha\boldsymbol{\theta}}) + \chi_{[\mathbf{0},\mathbf{e}]}(\boldsymbol{\theta}) \to \min_{\boldsymbol{\theta}\in\mathbb{R}^d}. \tag{12}$$

The scaling factors $\frac{1}{2}, \frac{1}{d}$ ensure that both objective terms take values in $[0,1]$. The minimisation problem (12) is subjected to bound constraints only, but the variable $\boldsymbol{\theta}$ is included in the exponential norm expressions in the first term as well as in the concave second term. As a result, the problem is likely to have many local minima and will be difficult to optimise. Considering the boundary values, it follows for $\boldsymbol{\theta} = \mathbf{0}$ that $\mathbf{K}_{\boldsymbol{\theta}} = (1)_{n\times n}$ and $\mathbf{y}_n^T \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n = 0$. For $\boldsymbol{\theta} \to \infty$, we have $\mathbf{K}_{\boldsymbol{\theta}} \to \mathbf{I}$ and $\mathbf{y}_n^T \mathbf{K}_{\boldsymbol{\theta}} \mathbf{y}_n \to \frac{1}{n_{+1}} + \frac{1}{n_{-1}}$.

4. D.C. decomposition and optimisation

Whereas RLP (1), SVM (3) and ℓ_1-SVM (4) are still convex QPs, adding the concave penalty term (5) makes problems FSV (6), the ℓ_2-ℓ_0-SVM (8), quadratic FSV (9) and, particularly, the kernel-target alignment approach (12) difficult to optimise due to possible local minima.

A robust algorithm for minimising non-convex problems is the *Difference of Convex functions Algorithm* (DCA) proposed in Pham and Hoai (1998) in a different context. It can be used to minimise a *non-convex* function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ which reads

$$f(\mathbf{x}) = g(\mathbf{x}) - h(\mathbf{x}) \to \min_{\mathbf{x}\in\mathbb{R}^d}, \tag{13}$$

where $g, h : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ are lower semi-continuous, proper convex functions cf. (Rockafellar, 1970). A property of this approach, particularly convenient for applications, is that f may be non-smooth. For example, constraints sets $C \ni x$ may be taken into account by adding a corresponding indicator function χ_C to the objective function f. In the next subsections, we first sketch the DCA and then apply it to our non-convex feature selection problems where the precise algorithm is determined by the appropriate decomposition of f in each case.

4.1. D.C. programming

According to Rockafellar (1970) and Pham and Hoai (1998), for a lower semi-continuous, proper convex function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ we use the standard notation

$$\begin{aligned}
\operatorname{dom} f &:= \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) < \infty\}, && \text{(domain)}\\
f^*(\tilde{\mathbf{x}}) &:= \sup_{\mathbf{x}\in\mathbb{R}^d} \{\langle \mathbf{x}, \tilde{\mathbf{x}}\rangle - f(\mathbf{x})\}, && \text{(conjugate function)}\\
\partial f(\mathbf{z}) &:= \{\tilde{\mathbf{x}} \in \mathbb{R}^d : f(\mathbf{x}) \geq f(\mathbf{z}) + \langle \mathbf{x}-\mathbf{z}, \tilde{\mathbf{x}}\rangle \quad \forall\, \mathbf{x} \in \mathbb{R}^d\} && \text{(subdifferential)}
\end{aligned}$$

for $\mathbf{z}, \tilde{\mathbf{x}} \in \mathbb{R}^d$. For differentiable functions we have $\partial f(\mathbf{z}) = \{\nabla f(\mathbf{z})\}$. According to Rockafellar (1970) [Theorem 23.5], it holds

$$\partial f(\mathbf{x}) = \arg\max_{\tilde{\mathbf{x}} \in \mathbb{R}^d} \{\mathbf{x}^T \tilde{\mathbf{x}} - f^*(\tilde{\mathbf{x}})\}, \quad \partial f^*(\tilde{\mathbf{x}}) = \arg\max_{\mathbf{x} \in \mathbb{R}^d} \{\tilde{\mathbf{x}}^T \mathbf{x} - f(\mathbf{x})\}. \tag{14}$$

In the remainder of this section, we will apply the following general algorithm:

Algorithm 4.1. *D.C. minimisation Algorithm* (*DCA*) (g, h, tol)

choose $\mathbf{x}^0 \in \operatorname{dom} g$ arbitrarily
for $k \in \mathbb{N}_0$

$$\textbf{do} \begin{cases} \text{select } \tilde{\mathbf{x}}^k \in \partial h(\mathbf{x}^k) \text{ arbitrarily} \\ \text{select } \mathbf{x}^{k+1} \in \partial g^*(\tilde{\mathbf{x}}^k) \text{ arbitrarily} \\ \textbf{if } \min\left(\left|x_i^{k+1} - x_i^k\right|, \left|\frac{x_i^{k+1} - x_i^k}{x_i^k}\right|\right) \leq tol \quad \forall\, i = 1, \ldots, d \\ \quad \textbf{then return } (\mathbf{x}^{k+1}) \end{cases}$$

The following theorem was proven in Pham and Hoai (1998) [Lemma 3.6, Theorem 3.7]:

Theorem 1 (*DCA convergence*)**.** *If* $g, h : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ *are lower semi-continuous, proper convex functions so that* $\operatorname{dom} g \subset \operatorname{dom} h$ *and* $\operatorname{dom} h^* \subset \operatorname{dom} g^*$, *then it holds for the DCA Algorithm* 4.1:

(i) *The sequences* $(\mathbf{x}^k)_{k \in \mathbb{N}_0}$, $(\tilde{\mathbf{x}}^k)_{k \in \mathbb{N}_0}$ *are well defined.*
(ii) $(f(\mathbf{x}^k) = g(\mathbf{x}^k) - h(\mathbf{x}^k))_{k \in \mathbb{N}_0}$ *is monotonously decreasing.*
(iii) *Every limit point of* $(\mathbf{x}^k)_{k \in \mathbb{N}_0}$ *is a critical point of* $f = g - h$. *In particular, if* $f(\mathbf{x}^{k+1}) = f(\mathbf{x}^k)$, *then* $\mathbf{x}^k$ *is a critical point of* f *in* (13).

Hence the algorithm converges to a local minimum that is controlled by the start value $\mathbf{x}^0$ and of course by the d.c. decomposition (13) of the objective. In case of non-global solutions, one may restart the DCA with a new initial point $\mathbf{x}^0$. However, Pham and Hoai (1998) state that the DCAs often converge to a global solution.

We point out that a similar optimisation approach has been proposed by Yuille and Rangarajan (2003), obviously unaware of previous related work in the mathematical literature (Pham Dinh, and Elbernoussi, 1988; Pham and Hoai, 1998) (Pham Dinh & Elbernoussi, 1988; Pham & Hoai, 1998). Whereas the approach by Yuille and Rangarajan (2003) assumes differentiable objective functions, our approach—adopted from (Pham & Hoai, 1998)—is applicable to a significantly larger class of non-smooth optimisation problems. This allows to include constraint sets in a natural way, for example.

4.2. Application to direct objective minimising feature selection

The crucial point in applying the DCA is to define a suitable d.c. decomposition (13) of the objective function. The aim of this section is to propose such decompositions for the different approaches under consideration.

4.2.1. FSV. Consider general problems of the form

$$\min_{\mathbf{x}\in X} f(\mathbf{x}) \tag{15}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is concave but not necessarily differentiable, and $X \subset \mathbb{R}^d$ is a polyhedral set. According to Mangasarian (1997) f always takes its minimum value at a vertex of the polyhedral feasible set X, and 'arg min' may be written as 'arg vertex min'. The symbol ∂f now denotes the *super*differential of f which, for concave f, is the analogue of the subdifferential for (not necessarily differentiable) convex functions. For such problems, and especially for the concave problem FSV (6), the following iterative algorithm was proposed in Bradley and Mangasarian (1998):

Algorithm 4.2. Successive Linearization Algorithm (SLA) (f, X)

choose $\mathbf{x}^0 \in \mathbb{R}^n$ arbitrarily
for $k \in \mathbb{N}_0$

$$\textbf{do} \begin{cases} \text{select } \mathbf{z} \in \partial f(\mathbf{x}^k) \text{ arbitrarily} \\ \text{select } \mathbf{x}^{k+1} \in \text{ arg vertex min}_{\mathbf{x}\in X}\, \mathbf{z}^T(\mathbf{x}-\mathbf{x}^k) \text{ arbitrarily} \\ \textbf{if } \mathbf{z}^T(\mathbf{x}^{k+1}-\mathbf{x}^k) = 0 \\ \quad \textbf{then return } (\mathbf{x}^k) \end{cases}$$

The algorithm produces a sequence of linear programs and terminates after a finite number of iterations (Mangasarian, 1997).

Now let us solve the general non-convex problems in the d.c. optimisation framework. It turns out that our new feature selection approaches not only generalise the FSV approach, but also that the SLA is a special case of the DCA: We show that the DCA applied to a *particular* d.c. decomposition (13) of (15) coincides with the SLA.

Corollary 2 (*SLA equivalence*). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be concave and $X \subset \mathbb{R}^d$ be a polyhedral set. Then for solving the concave minimisation problem* (15) *the SLA with $\mathbf{x}^0 \in X$ and DCA with $tol = 0$ are equivalent.*

Proof: Modelling problem (15) as a d.c. problem reads

$$\min_{\mathbf{x}\in\mathbb{R}^d} \chi_X(\mathbf{x}) - (-f(\mathbf{x})),$$

where the first term is defined as function g in (13), and the second one as h. Then we have in the DCA Algorithm 4.1

- $\mathbf{x}^0 \in \operatorname{dom} g \Leftrightarrow \mathbf{x}^0 \in X$, and for $k \in \mathbb{N}_0$:
- $\tilde{\mathbf{x}}^k \in \partial h(\mathbf{x}^k) \Leftrightarrow \tilde{\mathbf{x}}^k \in -\partial f(\mathbf{x}^k)$,
- $\mathbf{x}^{k+1} \in \partial g^*(\tilde{\mathbf{x}}^k) \overset{(14)}{\Leftrightarrow} \mathbf{x}^{k+1} \in \arg\min_{x\in X} -(\tilde{\mathbf{x}}^k)^T(\mathbf{x}-\mathbf{x}^k)$.

The problem given in the theorem has exactly the form for which the SLA Algorithm 4.2 is defined. Algorithm 4.2 and the above DCA are almost identical with $\mathbf{z} = -\tilde{\mathbf{x}}^k$. If we use $tol = 0$ in the DCA, choose our start value $\mathbf{x}^0 \in X$ in the SLA and apply, e.g., the simplex algorithm to obtain only vertex solutions, the algorithms are identical. □

4.2.2. ℓ_2-ℓ_0-SVM. A viable d.c. decomposition (13) for (8) reads

$$g(\mathbf{w}, b, \mathbf{v}) = \frac{\mu}{n} \sum_{i=1}^{n} (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))_+ + \frac{1}{2}\mathbf{w}^T \mathbf{w} + \chi_{[-\mathbf{v},\mathbf{v}]}(\mathbf{w}),$$
$$h(\mathbf{v}) = -\nu \mathbf{e}^T (\mathbf{e} - e^{-\alpha \mathbf{v}}).$$

Here and for the following problems, h is differentiable, so in the first step of DCA iteration $k \in \mathbb{N}_0$ we have $\tilde{\mathbf{x}}^k = \nabla h(\mathbf{x}^k)$. Combining the two DCA steps for each k by (14) leads to $\mathbf{x}^{k+1} \in \partial g^*(\nabla h(\mathbf{x}^k)) = \arg\max_{\mathbf{x}}\{\nabla h(\mathbf{x}^k)^T \mathbf{x} - g(\mathbf{x})\}$ so that we arrive at the constrained convex QP

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n, \mathbf{v}\in\mathbb{R}^d} \quad & \frac{\mu}{n}\mathbf{e}^T \boldsymbol{\xi} + \frac{1}{2}\mathbf{w}^T \mathbf{w} + \nu\alpha \mathbf{v}^T e^{-\alpha \mathbf{v}^k} \\ \text{subject to} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n, \\ & \boldsymbol{\xi} \geq \mathbf{0}, \\ & -\mathbf{v} \leq \mathbf{w} \leq \mathbf{v}. \end{aligned}$$

Note that the sequence of solutions to these QPs converges, due to Theorem 1, as f is bounded from below.

4.2.3. Quadratic FSV. To solve (9), we use the d.c. decomposition

$$g(\mathbf{w}, b, \mathbf{v}) = (1 - \lambda) \sum_{i=1}^{n} (1 - y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b))_+ + \sum_{i=1}^{d'} \sum_{\phi_i(\mathbf{e}_j)\neq 0} \chi_{[-v_j, v_j]}(w_i),$$
$$h(\mathbf{v}) = -\lambda \mathbf{e}^T (\mathbf{e} - e^{-\alpha \mathbf{v}}),$$

which, analogously to the previous approach, in each DCA step $k \in \mathbb{N}_0$ leads to a linear problem

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^{d'}, b\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n, \mathbf{v}\in\mathbb{R}^d} \quad & (1 - \lambda)\mathbf{e}^T \boldsymbol{\xi} + \lambda\alpha \mathbf{v}^T e^{-\alpha \mathbf{v}^k} \\ \text{subject to} \quad & y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n, \\ & \boldsymbol{\xi} \geq \mathbf{0}, \\ & -v_j \leq w_i \leq v_j, \quad i = 1, \ldots, d';\ \phi_i(\mathbf{e}_j) \neq 0. \end{aligned}$$

4.2.4. Kernel-target alignment approach. For the function defined in (12), as the kernel (10) is convex in $\boldsymbol{\theta}$, we split f as

$$g(\boldsymbol{\theta}) = \frac{1-\lambda}{2n_{+1}n_{-1}} \sum_{\substack{i,\,j=1 \\ y_i \neq y_j}}^{n} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{2,\boldsymbol{\theta}}/2\sigma^2} + \chi_{[\mathbf{0},\mathbf{e}]}(\boldsymbol{\theta}),$$

$$h(\boldsymbol{\theta}) = \frac{1-\lambda}{2} \sum_{\substack{i,\,j=1 \\ y_i = y_j}}^{n} \frac{1}{n^2_{y_i}} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{2,\boldsymbol{\theta}}/2\sigma^2} - \frac{\lambda}{d}\mathbf{e}^T(\mathbf{e} - e^{-\alpha\boldsymbol{\theta}}).$$

Again h is differentiable, so by applying the DCA we find the solution in the first step of iteration k as

$$\tilde{\boldsymbol{\theta}}^k = \nabla h(\boldsymbol{\theta}^k) = -\frac{1-\lambda}{4\sigma^2} \sum_{\substack{i,\,j=1 \\ y_i = y_j}}^{n} \frac{1}{n^2_{y_i}} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{2,\boldsymbol{\theta}^k}/2\sigma^2} ((x_{il} - x_{jl})^2)_{l=1}^d - \frac{\lambda}{d}\alpha e^{-\alpha\boldsymbol{\theta}^k}.$$

In the second step, looking for $\boldsymbol{\theta}^{k+1} \in \partial g^*(\tilde{\boldsymbol{\theta}}^k) \overset{(14)}{=} \arg\max_{\boldsymbol{\theta}}\{\boldsymbol{\theta}^T\tilde{\boldsymbol{\theta}}^k - g(\boldsymbol{\theta})\}$ leads to solving the *convex non-quadratic* problem

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \frac{1-\lambda}{2n_{+1}n_{-1}} \sum_{\substack{i,\,j=1 \\ y_i \neq y_j}}^{n} e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2_{2,\boldsymbol{\theta}}/2\sigma^2} - \boldsymbol{\theta}^T\tilde{\boldsymbol{\theta}}^k \quad \text{subject to} \quad \mathbf{0} \leq \boldsymbol{\theta} \leq \mathbf{e} \tag{16}$$

with a valid initial point $\mathbf{0} \leq \boldsymbol{\theta}^0 \leq \mathbf{e}$. We solve the problems (16) efficiently by a trust region method using the function `fmincon` in MATLAB's optimisation toolbox (MathWorks, 2002). Alternatively, a penalty/barrier multiplier method with logarithmic-quadratic penalty function as proposed in Ben-Tal and Zibulevsky (1997) also reliably solves the problems.

5. Evaluation

To study the performance of our new methods in detail, we first present computer generated ground truth experiments illustrating the general behaviour and robustness of the nonlinear classification methods in Section 5.1. To evaluate the performance of the suggested approaches at large, we study various real-world problems in Section 5.2 and finally examine the high-dimensional research problem of organ classification in CT scans in Section 5.3.

5.1. Ground truth experiments

In this section, we consider artificial training sets in $\mathbb{R}^2$ and $\mathbb{R}^4$ where y is a function of the first two features x_1 and x_2. We examine specially designed points $(x_1, x_2) \in \mathbb{R}^2$ on

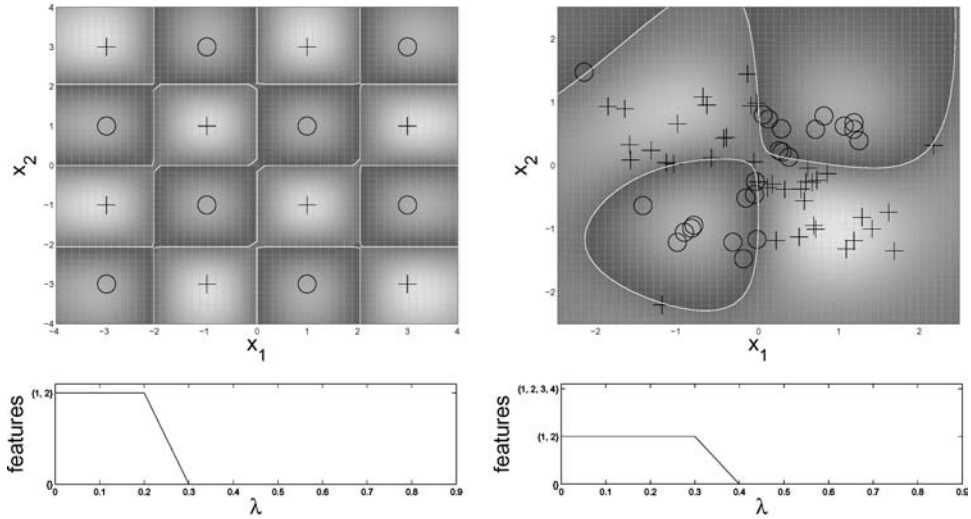

Figure 2. Chess board classification problems with $\frac{y+1}{2} = (\lfloor \frac{x_1}{2} \rfloor \bmod 2) \oplus (\lfloor \frac{x_2}{2} \rfloor \bmod 2)$. *Top*: Training points and Gaussian SVM decision boundaries (*white lines*) for $\sigma = 1$, $\lambda = 0.1$, *left*: in $\mathbb{R}^2$, *right*: zoomed projection of $\mathbb{R}^4$ onto selected features. *Bottom*: Features determined by (12).

the left of the figures and $n = 64$ randomly distributed points $(x_1, x_2, x_3, x_4) \in \mathbb{R}^4$ on the right.

The examples in Figure 1 show that our quadratic FSV approach indeed performs feature selection and finds classification rules for quadratic, not linearly separable problems. Ranking methods for feature selection as well as linear classification approaches do not appreciate the feature relevance for these problems.

For the non-quadratic chess board classification problems in Figure 2, our kernel-target alignment approach performs very well, in contrast to all other feature selection approaches presented. Again, the features by themselves do not contain relevant information and all linear methods are doomed to fail. In both test examples, only relevant feature sets are selected by our methods as can be seen in the bottom plots. Particularly the correct feature set $\{1, 2\}$ is selected for most values of λ. This clearly shows the favourable properties of *embedded* feature selection also in connection with nonlinear classification.

Figure 2 shows on the right a remarkable property: The alignment approach discards the two noise features even for $\lambda = 0$ which indicates that the alignment functional (11) incorporates implicit feature selection. This is due to the isotropic properties of the Gaussian kernel where the feature space distances are bounded by $\|\phi(\mathbf{x})\|^2 = \langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle = K(\mathbf{x}, \mathbf{x}) = 1$. As argued in Section 3.2.2, maximising the alignment term $\mathbf{y}_n^T \mathbf{K}_{\boldsymbol{\theta}} y_n$ amounts to maximising the class–centre distance of the feature vectors which lie on the unit sphere in ℓ_2. Adding

random features disturbs the original distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ and so distributes the feature vectors $\phi(\mathbf{x}_i)$ more uniformly on the sphere potentially moving the class means closer to each other. More precisely, adding features

$$\mathbf{x} \mapsto \begin{pmatrix} \mathbf{x} \\ \tilde{\mathbf{x}} \end{pmatrix}$$

leads for $\boldsymbol{\theta} = \mathbf{e}$ to kernel matrix elements

$$e^{(-\|\mathbf{x}-\mathbf{z}\|_2^2 - \|\tilde{\mathbf{x}}-\tilde{\mathbf{z}}\|_2^2)/2\sigma^2} = K(\mathbf{x}, \mathbf{z}) \cdot e^{-\|\tilde{\mathbf{x}}-\tilde{\mathbf{z}}\|_2^2/2\sigma^2}$$

for $\mathbf{x}, \mathbf{z} \in \mathcal{X}$. If the new features are random, roughly all off-diagonal elements are damped by the same factor α. Splitting the diagonal from the off-diagonal terms, the original alignment $\mathbf{y}_n^T \mathbf{K} \mathbf{y}_n =: (\frac{1}{n_{+1}} + \frac{1}{n_{-1}}) + c$ is reduced if $c > 0$ or $\mathbf{y}_n^T \mathbf{K} \mathbf{y}_n > \frac{1}{n_{+1}} + \frac{1}{n_{-1}}$. For large n_i, the value of the alignment term is reduced to $(\frac{1}{n_{+1}} + \frac{1}{n_{-1}}) + \alpha c$ by almost the factor α too. The implicit feature selection of the alignment functional does not apply for arbitrary kernels: The linear kernel, e.g., leads to a (nonnegative) alignment summand for each feature.

5.2. *Real-world data*

We compare our approaches with RLP (1) and FSV (6) favoured over the ℓ_1-SVM in Bradley and Mangasarian (1998) and standard linear and Gaussian kernel SVMs as well as with the fast SVM–based filter method for feature selection Heiler, Cremers, and Schnörr (2001) ranking the features according to the linear SVM decision function.

5.2.1. Data sets and preprocessing. To test all our methods on real-world data, we use several data sets from the UCI repository Blake and Merz (1998) as well as the high-dimensional Colon Cancer data set from Weston et al. (2003). The problems mostly treat medical diagnoses based on genuine patient data and are resumed in Table 1 where we use distinct short names for the databases. (See also Bradley & Mangasarian, 1998) for a brief review of most of the data sets used.) It is essential that the features are normalised, especially for the kernel-target alignment approach as their variances influence its sensible objective with initially equal weights. In experiments, it shows that otherwise features with large variances are preferred. So we rescale the features linearly to zero mean and unit variance.

5.2.2. Choice of parameters. As to the parameters, we set $\alpha = 5$ in (5) as proposed in Bradley and Mangasarian (1998) and $\sigma = \frac{\sqrt{d}}{2}$ in (10) which maximises the alignment of the problems. We start the DCA with $\mathbf{v}^0 = \mathbf{e}$ for the ℓ_2-ℓ_0-SVM, FSV and quadratic FSV and with $\boldsymbol{\theta}^0 = \mathbf{e}/2$ for the kernel-target alignment approach, respectively. We stop on $\mathbf{v}$ with $tol = 10^{-5}$ resp. $tol = 10^{-3}$ for $\boldsymbol{\theta}$. To determine the weight parameters, we discretise their range of values and perform a parameter selection step minimising the error on an independent validation set before actually applying the feature selection algorithm. The validation set is chosen randomly as one half of each run's (cross-validation) training set

Table 1. Statistics for data sets used.

Data set	No. of features d	No. of samples n	Class distribution n_{+1}/n_{-1}
wpbc60	32	110	41/ 69
wpbc24	32	155	28/127
liver	6	345	145/200
cleveland	13	297	160/137
ionosphere	34	351	225/126
pima	8	768	500/268
bcw (breast-cancer-wisconsin)	9	683	444/239
sonar	60	208	111/97
musk	166	476	207/269
microarray	2000	62	22/40

to select $\ln \mu \in \{0, \ldots, 10\}$, $\ln \nu \in \{-5, \ldots, 5\}$ or $\lambda \in \{0.05, 0.1, 0.2, \ldots, 0.9, 0.95\}$ for (quadratic) FSV or $\lambda \in \{0, 0.1, \ldots, 0.9\}$ for the kernel-target alignment approach. In case of equal validation error, we choose the larger values for (ν, μ) resp. λ. In the same manner, the SVM weight parameter λ is chosen according to the smallest $\frac{1-\lambda}{\lambda} \in \{e^{-5}, e^{-4}, \ldots, e^{5}\}$ independently of the selected features. For the filter method, we successively include features until the validation error does not drop 0.1 per cent below the current value five times. The final classifier is then built from the training and validation sets. To solve the elementary optimisation problems, we use the CPLEX solver library (Ilog, 2001), MATLAB's QP solver `quadprog` for the common SVMs as well as its constrained optimisation method `fmincon` documented in MathWorks (2002).

5.2.3. Results. We first partition the data equally into a training, a validation and a test set. The validated parameters and test results for the linear classifiers are summarised in Table 2 where the number of features is determined as $|\{j = 1, \ldots, d : |w_j| > 10^{-8}\}|$. As a result of the validation, the optimal combination for (μ, ν) mostly falls within the range of discretised values. Further, the methods are often stable for large regions of values for ν or for the ratio μ/ν. Our linear methods achieve feature selection and are often able to improve the classification performance compared with the baseline RLP classifier. Especially for the very high dimensional 'microarray' data, both our linear feature selection methods ℓ_2-ℓ_1-SVM and ℓ_2-ℓ_0-SVM are more accurate than even the linear SVM.

For more thorough cross-validation experiments, the aggregate results are summarised in Table 3 for linear and Table 4 for nonlinear classifiers. The number of features is again determined as $|\{j = 1, \ldots, d : |w_j| > 10^{-8}\}|$ again resp. $|\{j = 1, \ldots, d : |\theta_j| > 10^{-2}\}|$. The results for the quadratic FSV on data set 'musk' are not given due to the high problem dimension. It is clear that all proposed approaches perform feature selection: linear FSV discards most features followed by the kernel-target alignment approach, the SVM ranking

Table 2. Feature selection and linear classification performance (number of features, test error [%]) and weight parameters that minimise classification error on validation set.

	RLP (1)		Linear SVM (3)		Ranking		FSV (6)			ℓ_2-ℓ_1-SVM (7)			ℓ_2-ℓ_0-SVM (8)		
Data set	dim	err	dim	err	dim	err	dim	err	λ^*	dim	err	$(\ln \mu^*, \ln \nu^*)$	dim	err	$(\ln \mu^*, \ln \nu^*)$
wpbc60	32	44	32	31	2	31	0	31	0.95	27	33	$(1, -4)$	27	33	$(1, -5)$
wpbc24	32	25	32	22	1	22	0	22	0.95	19	20	$(10, 5)$	13	22	$(8, 5)$
liver	6	28	6	30	6	30	2	33	0.3	6	30	$(9, 5)$	6	31	$(5, 1)$
cleveland	13	17	13	16	6	18	4	23	0.05	9	17	$(8, 5)$	7	17	$(2, -2)$
ionosphere	33	12	34	11	9	14	2	14	0.2	19	11	$(9, 5)$	3	15	$(6, 3)$
pima	8	26	8	27	6	27	1	29	0.05	7	27	$(6, -1)$	8	27	$(5, -3)$
bcw	9	4	9	4	8	4	1	9	0.2	9	4	$(3, -2)$	8	4	$(5, -3)$
sonar	48	34	60	32	15	35	20	31	0.05	30	35	$(6, 2)$	58	26	$(10, -4)$
musk	113	27	166	18	3	39	17	22	0.05	163	18	$(10, -3)$	160	18	$(10, -4)$
microarray	41	**40**	2000	**10**	4	**30**	1	**15**	0.3	21	**5**	$(1, 0)$	18	**5**	$(0, -3)$

Table 3. Feature selection and linear classification tenfold cross-validation performance (average number of features, average test error [%], error variance [%]), bold numbers indicate lowest errors of feature selection methods including Table 4.

	RLP			Linear SVM			Ranking			FSV			ℓ_2-ℓ_1-SVM			ℓ_2-ℓ_0-SVM		
Data set	dim	err	var	dim	err	var	dim	err	var	dim	err	var	dim	err	var	dim	err	var
wpbc60	32.0	40.9	2.7	32.0	33.6	1.5	4.9	36.4	2.1	0.4	36.4	1.7	12.4	**35.5**	1.2	13.4	37.3	1.4
wpbc24	32.0	27.7	1.1	32.0	18.1	1.0	1.8	18.1	1.0	0.0	18.1	1.0	12.6	**17.4**	0.9	2.9	18.1	1.0
liver	6.0	31.9	0.7	6.0	32.5	0.7	4.5	33.3	0.7	2.1	36.2	1.0	6.0	35.1	1.0	5.0	34.2	1.6
cleveland	13.0	16.2	0.6	13.0	15.8	0.5	6.9	**16.2**	0.4	1.8	23.6	1.0	9.9	16.5	0.5	8.2	16.5	0.4
ionosphere	33.0	13.4	0.1	34.0	13.4	0.1	10.0	14.0	0.2	2.3	21.7	1.0	24.8	13.4	0.3	14.0	15.7	0.6
pima	8.0	22.5	0.3	8.0	23.2	0.2	5.5	**24.0**	0.1	0.6	30.1	0.4	6.6	25.1	0.2	6.1	24.7	0.2
bcw	9.0	3.4	0.0	9.0	2.9	0.0	8.7	**3.1**	0.0	2.4	4.8	0.0	8.7	**3.2**	0.0	7.9	**3.1**	0.0
sonar	51.6	27.9	0.7	60.0	26.0	0.3	10.0	27.9	0.6	4.6	27.4	0.4	50.4	**22.6**	0.1	40.3	23.6	0.2
musk	116.0	20.6	0.2	166.0	15.3	0.1	12.6	29.2	0.4	4.0	28.2	0.2	125.1	18.3	0.3	105.2	16.8	0.2

method and then the ℓ_2-ℓ_0-SVM, then the ℓ_2-ℓ_1-SVM. In addition, for all approaches the test error is often smaller than for RLP. The quadratic FSV performs well mainly for special problems (e.g., 'liver' and 'ionosphere'), but the classification is good in general for all other approaches. For the kernel-target alignment approach, apart from the apparent feature reduction, also the number of SVs is generally reduced which can be seen in Table 4. This allows again faster classification and may also lead to a higher generalisation ability. The average number of DC iterations given in Table 4 for a run with ten validation calls and the final evaluation is still moderate.

Table 4. Feature selection and nonlinear classification tenfold cross-validation average performance (number of features, test error [%], error variance [%], number of DCA iterations, number of Support Vectors), bold numbers indicate lowest errors of feature selection methods including Table 3.

	Gaussian SVM				Quad. FSV			Kernel-target alignment				
Data set	dim	err	var	SVs	dim	err	var	dim	err	var	DCA iter	SVs
wpbc60	32.0	32.7	2.3	94.3	3.2	37.3	1.7	4.4	**35.5**	3.0	248.1	92.0
wpbc24	32.0	16.8	0.9	123.8	0.0	18.1	1.0	1.9	18.1	1.0	215.2	131.5
liver	6.0	33.3	0.8	233.1	3.2	**32.5**	0.8	2.5	35.4	1.5	242.6	262.3
cleveland	13.0	15.8	0.5	241.0	9.2	32.3	1.4	3.2	23.6	0.3	139.6	224.4
ionosphere	34.0	7.1	0.2	159.7	32.9	10.5	0.4	6.6	**7.7**	0.3	192.2	109.6
pima	8.0	23.4	0.2	481.1	4.7	29.9	0.4	1.4	27.0	0.2	202.2	444.2
bcw	9.0	2.9	0.0	229.0	5.9	9.4	0.1	2.8	4.2	0.0	74.9	160.5
sonar	60.0	12.5	0.8	159.1	60.0	24.0	0.7	9.6	27.4	0.6	268.2	110.7
musk	166.0	5.5	0.1	311.7	–	–	–	41.0	**15.5**	0.2	676.5	218.9

Table 5. Feature selection and classification time relative to fastest method.

Data set	RLP	Lin. SVM	Ranking	FSV	ℓ_2-ℓ_1-SVM	ℓ_2-ℓ_0-SVM	Gauss. SVM	quad. FSV	Kernel-target align.
wpbc60	5	67	105	5	1	9	38	454	263

Average runtimes for the final feature selection and classifier training during cross-validation are given in Table 5. Taking into account that RLP is FSV for $\lambda = 0$ and that the common SVMs are determined by the MATLAB solver, the runtimes reflect the problem types. Also the kernel-target alignment approach is tractable. Besides, one should be aware that the final classification is fast for all approaches.

We already pointed out in Section 5.1 that the alignment approach performs feature selection implicitly which means without feature penalty ($\lambda = 0$). To illustrate this, the respective results are given in Table 6. Of course the number of selected features is larger than with feature penalty as in Table 4, but many features are discarded inherently along with a sound classification performance. Note that this gives a reliable feature selection approach without any necessity for parameter selection.

5.3. Organ classification in CT scans

The results on the 'microarray' data set in the previous section already indicate that feature selection methods are more important in higher dimensions. The evaluation of medical data is a prominent area where this occurs. Due to the unknown relevant factors and problem nature, at first often large feature sets are collected.

Here, we study the classification of specific organs in CT scans where no satisfactory algorithms exist up to now. However this automatic detection is essential for the treatment of,

Table 6. SVM tenfold cross-validation performance (average number of features, average test error [%]) with features chosen by (12) for $\lambda = 0$.

	Kernel-target alignment	
Data set	dim	err
wpbc60	9.0	38.2
wpbc24	6.5	17.4
liver	4.0	29.6
cleveland	4.2	19.9
ionosphere	8.9	7.1
pima	2.0	25.9
bcw	3.0	4.0
sonar	13.6	24.5
musk	48.4	14.3

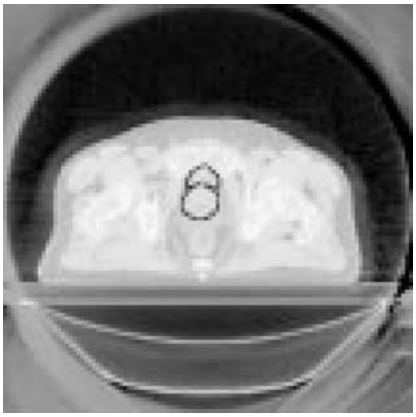

Figure 3. Sample CT slice from data set 'organs22' with contours of both organs.

e.g., cancer patients. The data originates from three-dimensional CT scans of the masculine hip region. An exemplary two-dimensional image slice is depicted in Figure 3. To label the images, the adjacent organs bladder and prostate have been masked manually by experts. The contours of both organs are shown in Figure 3 where the organs are very difficult to distinguish visually.

As described in Schmidt (2004), the images are filtered by a three-dimensional steerable pyramid filter bank with 16 angular orientations and four decomposition levels. Then local histograms are built for the filter responses with ten bins per channel. Including the original grey values, this results in 650 features per image voxel. The task is to label each voxel with the correct organ. Here, the high-dimensional feature space is induced by the filtering which requires many directions due to the already three input image dimensions. In total,

Table 7. Feature selection and linear classification performance for CT data (number of features, test error [%]) with weight parameters chosen to minimise classification error on validation set.

	RLP		Lin. SVM		FSV		ℓ_2-ℓ_1-SVM		ℓ_2-ℓ_0-SVM	
Data set	dim	err	dim	err	dim	err	dim	err	dim	err
organs4	225	13.2	650	1.1	4	2.3	61	0.9	18	0.7
organs20	242	15.2	650	1.4	6	3.6	79	1.5	43	2.7
organs22	231	11.7	650	1.3	3	11.4	106	2.2	66	2.2

for problem 'organs22', the data for the region where bladder or prostate are contained amount to $117 \times 80 \times 31$ feature vectors $\in \mathbb{R}^{650}$.

In our experiments, we consider three different patients or data sets. For each of those, we select 500 feature vectors from each class. From those, we use 334 arbitrary samples for training and test, respectively, during the parameter validation and then train our final classifier on all 1000 training vectors. Note that, by choosing an equal number of training samples from both classes different from the entire test set where $\frac{n_{+1}}{n_{-1}} \in [\frac{1}{12}, \frac{1}{4}]$, we put more weight on the errors of the smaller class 'prostate'.

As done in Schmidt (2004), we also apply an SVM classifier with χ^2 kernel

$$K_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = e^{-\rho \sum_{k=1}^{d} \theta_k \frac{(x_k - z_k)^2}{x_k + z_k}}$$

for $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$ with $\rho = 2^{-11}$ on unmodified features. According to Haasdonk and Bahlmann (2004), the kernel is positive definite. Nevertheless, we include a bias term b as in the linear case. This kernel achieved a performance significantly superior to the Gaussian kernel for histogram features in Chapelle, Haffner, and Vapnik (1999). In order to apply the kernel-target alignment approach for feature selection, one has to replace the Gaussian kernel by the new kernel which is still convex in $\boldsymbol{\theta}$ in Section 4.2.4.

In our experiments, we include the filter method (Heiler, Cremers, & Schnörr, 2001) for the χ^2 SVM now determining the ranking and number of features by cross-validation on the final training set. For the other approaches, we use the same parameter settings as in the previous section. Then the results for the three patients are given in Table 7 for linear and in Table 8 for nonlinear classification methods.

Table 8. Feature selection and nonlinear classification performance for CT data (number of features, test error [%]) with weight parameters chosen to minimise classification error on validation set.

	Gaussian SVM		χ^2 SVM		χ^2 SVM ranking		Kernel-target align.	
Data set	dim	err	dim	err	dim	err	dim	err
organs4	650	1.5	650	0.8	25	1.2	16	1.6
organs20	650	2.3	650	1.1	32	1.8	29	1.9
organs22	650	2.2	650	1.9	22	2.7	35	3.9

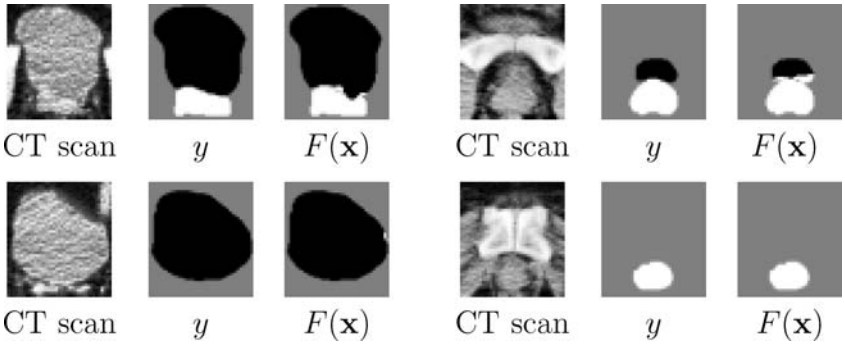

Figure 4. Sample results for ℓ_2-ℓ_0-SVM on classification problem 'organs4'; classes are marked black and white.

The data sets seem to be well linearly separable which also results in much lower classification and training times. Even more, the Gaussian SVM yields astonishingly bad results compared with its linear and χ^2 variants although reasonable values for the weight λ are selected and our chosen kernel width σ produces an alignment of around 12 per cent on the training set which is maximised for a near kernel width $\in [\sigma/2, \sigma]$. This slight overestimation of σ is due to the sparsity of the histogram features. The error of the Gaussian SVM always increased compared with its validation error of 0.3–2.1 per cent whereas it decreased for the other SVMs. But the scant superiority of Gaussian SVMs over linear ones is also consistent with Chapelle, Haffner, and Vapnik (1999).

Both our linear methods perform very well: They sometimes reduce the classification error compared with RLP and linear SVM on the whole feature set and reliably reduce the number of features. The alignment approach and the filter method select very few features only, in particular only few features corresponding to each filter subband. So the alignment approach well copes with the redundancy of the histogram features. The classification results for the ℓ_2-ℓ_0-SVM on the data set 'organs4' may be visually compared with the mask considered as ground truth in Figure 4. The organs are classified with a high accuracy although the classes are again difficult to distinguish visually. The dimension reduction also leads to a reduced classification time for all feature selection approaches which is essential in real-time medical applications.

6. Summary and conclusions

We proposed several novel methods that extend existing linear embedded feature selection approaches towards better generalisation ability by improved regularisation, and constructed feature selection methods in connection with nonlinear classifiers. In order to apply the DCA, we found appropriate splittings of our non-convex objective functions.

Our results show that embedded nonlinear methods, which have been rarely examined up to now, are indispensable for feature selection. In the experiments with real data, effective feature selection was always carried out by our methods in conjunction with a small

classification error. So direct objective minimising feature selection is profitable and viable for different types of classifiers. In higher dimensions, the curse of dimensionality affects the classification error even more such that our methods are also more important here.

For multi-class classification problems solved by a sequence of binary classifiers, one could select features for every binary classifier or apply one of the embedded approaches for all classifiers simultaneously. This is left for future research.

The approaches may also be extended to incorporate other feature maps in the same manner as quadratic FSV. For the kernel-target alignment approach, an application to kernels other than the Gaussian is possible as we have shown in the experiments with histogram features.

Acknowledgments

This work was funded by the DFG, Grant Schn 457/5.

Further thanks to Stefan Schmidt for the reliable collaboration concerning the evaluation of the CT data. as well as to Dr. Pekar and Dr. Kaus, Philips Research Hamburg, for providing the data and stimulating discussions.

References

Bach, F., Lanckriet, G., & Jordan, M. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the 21st International Conference on Machine Learning*. New York, NY: ACM Press.

Ben-Tal, A., & Zibulevsky, M. (1997). Penalty/Barrier multiplier methods for convex programming problems. *SIAM Journal on Optimization, 7:2*, 347–366.

Bennett, K. P., & Mangasarian, O. L. (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software, 1*, 23–34.

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases.

Bradley, P. S. (1998). Mathematical programming approaches to machine learning and data mining. Ph.D. thesis, University of Wisconsin, Computer Sciences Dept., Madison, WI, TR-98-11.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In J. Shavlik (Ed.), *Proceedings of the 15th international conference on machine learning* (pp. 82–90). San Francisco, CA: Morgan Kaufmann.

Chapelle, O., Haffner, P., & Vapnik, V. N. (1999). SVMs for histogram-based image classification. *IEEE Transactions on Neural Networks, 10:5*, 1055–1064.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., & Kandola, J. (2002). On kernel-target alignment. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems 14* (pp. 367–373). Cambridge, MA: MIT Press.

Duda, R., Hart, P., & Stork, D. (2000). *Pattern classification*. New York, NY: John Wiley & Sons, second edition.

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

Haasdonk, B., & Bahlmann, C. (2004). Learning with distance substitution kernels. In C. E. Rasmussen, H. H. Bülthoff, M. A. Giese, & B. Schölkopf (Eds.), *Pattern recognition, proc. of 26th DAGM symposium*, Vol. 3175 of LNCS. (pp. 220–227). Berlin: Springer.

Heiler, M., Cremers, D., & Schnörr, C. (2001). Efficient feature subset selection for support vector machines. Technical Report TR-01-021, Comp. science series, Dept. of Mathematics and Computer Science, University of Mannheim.

Hermes, L., & Buhmann, J. M. (2000). Feature selection for support vector machines. In *Proc. of the International Conference on Pattern Recognition (ICPR'00)*, Vol. 2 (pp. 716–719).

Ilog, Inc.: 2001, 'ILOG CPLEX 7.5'.

Jakubik, O. J. (2003). Feature selection with concave minimization. Master's thesis, Dept. of Mathematics and Computer Science, University of Mannheim.

Jebara, T., & Jaakkola, T. (2000). Feature selection and dualities in maximum entropy discrimination. In I. Bratko & S. Dzeroski (Eds.), *Proceedings of the 16th international conference on machine learning* (pp. 291–300). San Francisco, CA: Morgan Kaufmann.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In R. S. Michalski & G. Tecuci (Eds.), *Proc. of the 11th international conference on machine learning* (pp. 121–129). San Francisco, CA: Morgan Kaufmann.

Mangasarian, O. L. (1997). Minimum-support solutions of polyhedral concave programs. Technical Report TR-1997-05, Mathematical Programming, University of Wisconsin.

MathWorks. (2002). Optimization toolbox user's Guide. The MathWorks, Inc.

Neumann, J., Schnörr, C., & Steidl, G. (2004). SVM-based feature selection by direct objective minimisation. In C. E. Rasmussen, H. H. Bülthoff, M. A. Giese, & B. Schölkopf (Eds.), *Pattern recognition, proc. of 26th DAGM symposium*, Vol. 3175 of LNCS (pp. 212–219). Berlin: Springer.

Pham Dinh, T., & Elbernoussi, S. (1988). Duality in d.c. (difference of convex functions optimization. Subgradient Methods. In *Trends in Mathematical Optimization*, Vol. 84 of Int. Series of Numer. Math. Basel: Birkäuser Verlag (pp. 277–293).

Pham Dinh, T., & Hoai An, L. T. (1998). A D.C. Optimization Algorithm for Solving the Trust-Region Subproblem. *SIAM Journal on Optimization, 8:2*, 476–505.

Rockafellar, R. T. (1970). *Convex analysis*. Princeton, NJ: Princeton University Press.

Schmidt, S. (2004). Context-sensitive image labeling based on logistic regression. Master's thesis, Dept. of Mathematics and Computer Science, University of Mannheim.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B, 58:1*, 267–288.

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. (2001). Feature Selection for SVMs. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems 13* (pp. 668–674). Cambridge, MA: MIT Press.

Yuille, A., & Rangarajan, A. (2003). The convex-concave procedure. *Neural Computation, 15*, 915–936.

Zhu, J., Rosset, S., Hastie, T., & Tibshirani, R. (2004). 1-norm support vector machines. In S. Thrun, L. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems 16*. Cambridge, MA: MIT Press.

Received August 17, 2004
Revised March 16, 2005
Accepted April 3, 2005