

LESS: a Model-Based Classifier for Sparse Subspaces

Cor J. Veenman* and David M.J. Tax†

Abstract In this paper we specifically focus on high dimensional data sets for which the number of dimensions is an order of magnitude higher than the number of objects. From a classifier design standpoint, such small sample size problems have some interesting challenges. The first challenge is to find, from all hyperplanes that separate the classes, a separating hyperplane which generalizes well for future data. A second important task is to determine which features are required to distinguish the classes. To attack these problems, we propose the LESS (Lowest Error in a Sparse Subspace) classifier that efficiently finds linear discriminants in a sparse subspace. In contrast with most classifiers for high dimensional data sets, the LESS classifier incorporates a (simple) data model. Further, by means of a regularization parameter the classifier establishes a suitable trade-off between subspace sparseness and classification accuracy. In the experiments we show how LESS performs on several high dimensional data sets and compare its performance to related state-of-the-art classifiers like among others linear ridge regression with the LASSO and the Support Vector Machine. It turns out that LESS performs competitively while using fewer dimensions.

Keywords: Classification, support vector machine, high dimensional, feature subset selection, mathematical programming.

*Corresponding author

†The authors are with the Department of Mediamatics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, P.O. Box 5031, 2600 GA, Delft, the Netherlands. E-mail: {C.J.Veenman, D.M.J.Tax}@ewi.tudelft.nl

1 Introduction

In nowadays applications data sets with hundreds or even thousands of features are no exception. For the representation of distributions in these high dimensional feature spaces there are hardly ever enough objects. According to a rule of thumb, for the estimation of data distributions the number of objects should be an order of magnitude higher than the number of dimensions. In this paper we consider data sets where the number of objects is even orders of magnitude lower than the number of features. Accordingly, the space is not just too sparsely sampled, almost any arbitrary hyperplane can separate the space into the desired classes.

A classifier that is known for being robust is the Nearest Mean Classifier. It has already been successfully applied to this type of data [22]. To be applied to such high dimensional data a suitable feature subset has to be selected for optimal performance. In [22] a naive filtering approach has been chosen. Alternative feature selection methods can be applied that are less greedy like forward selection or backward elimination, see e.g. [19]. Ultimately even a combinatorial optimization problem can be defined that aims at finding the feature subset with the highest classification performance in a wrapper framework [15]. The obvious drawback of the latter approach is that it is computationally intractable, though approximations through genetic algorithms, e.g. [14], simulated annealing [8] or tabu search [24] have been reported.

In this paper we choose another approach. Instead of selecting a suitable feature subset, we introduce a weighting factor for each dimension. The feature subset selection problem then turns into a problem of finding the weight factors, where a zero weight effectively rules out the respective feature. We propose the LESS classifier which is a Weighted Nearest Mean Classifier that balances classification errors with model sparseness. The LESS classifier can be seen as a variant of the L_1 Support Vector Machine. The main difference with related classifiers for high dimensional data sets like the SVM is that the LESS classifier employs a model of the class distributions. Accordingly, in general higher classification accuracy can be achieved in a lower dimensional subspace.

In the next Section, we motivate and propose the LESS classifier. In the following Section, we describe a number of related classifiers that have proven to be adequate for high-dimensional data. Then, we evaluate the LESS classifier and compare its performance to the described related classifiers and we discuss the results in the final Section.

2 The LESS Classifier Model

We first formalize the problem. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be a data set with n objects, where \mathbf{x}_i is a feature vector in a p -dimensional metric space. For the sake of simplicity we consider only two-class problems. Then, each object has a class label y_i being either -1 or $+1$. Let μ_1 be the mean vector of the n_1 objects $X_1 \subset X$ with label -1 , μ_2 the mean vector of the n_2 objects $X_2 \subset X$ with label $+1$, and μ_0 the mean vector of the whole data set. Further, Σ_k is the covariance matrix of objects X_k and the variance vector σ_k^2 is the diagonal of Σ_k . We denote the predicted label for object \mathbf{x} with $f(\mathbf{x})$.

For the LESS classifier we established two important design criteria. First, it should have high classification performance on high dimensional data or small sample size problems. We consider data sets where the number of features is orders of magnitude higher than the number of objects, e.g. 10 – 100 objects with 1000 – 10,000 dimensions. Second, the classifier should use as few dimensions as possible in achieving its performance.

These objectives are also the underlying design principles of a number of related classifiers that we describe in the *Related Work* Section, i.e. the Support Vector Machine [7], [23], Liknon [3], Ridge Regression with the LASSO [20], and the Nearest Shrunken Centroids [21]. These classifiers achieve these objectives by formulating a minimization problem consisting of a classification error term and a complexity term. Among these classifiers only the Nearest Shrunken Centroids classifier incorporates a model of the class distributions. Especially, for the extreme low sample size problems that we study, such a model bias is expectedly beneficial.

Also the proposed LESS classifier employs a data model. We base the LESS classifier on the Nearest Mean Classifier, which assumes statistically independent features with equal variances, i.e. $\Sigma_1 = \Sigma_2 = \sigma^2 I$, see [9] (Section 2.6.1). Accordingly, for that classifier the classes can be modeled with their means only. An extension of the Nearest Mean Classifier that naturally allows for feature selection is the Weighted Nearest Mean Classifier. The Weighted Nearest Mean Classifier is defined as:

$$f(\mathbf{x}) = \begin{cases} -1, & \text{if } d_m^2(\mathbf{x}, \mu_1) - d_m^2(\mathbf{x}, \mu_2) < 0 \\ +1, & \text{otherwise} \end{cases} \quad (1)$$

where $d_m^2(\mathbf{x}, \mathbf{y})$ is the squared diagonally weighted Euclidean distance [2] as follows:

$$d_m^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})' M' M (\mathbf{x} - \mathbf{y}) = \sum_{j=1}^p m_j^2 (x_j - y_j)^2. \quad (2)$$

Here M is a diagonal matrix with $M_{jj} = m_j \geq 0$ and m_j is the weighting factor for feature or dimension j .

Features can be selected by setting the respective weighting factors $m_j > 0$. The actual values of the weighting factors take into account the possible varying quantities or units of the features. For instance, to classify apples and pears based on perimeter (cm), color wave length (nm), and weight (kg). In contrast with the Nearest Mean Classifier, the Weighted Nearest Mean Classifier only assumes the classes to have equal covariance matrices, i.e. $\Sigma_1 = \Sigma_2$, see [9] (Section 2.6.2).

The training of a standard Nearest Mean Classifier is trivial. The training of the Weighted Nearest Mean Classifier we define as finding proper weights for a given training data set under the condition that all training objects must be classified correctly. Formally, this can be written as:

$$\forall i : y_i \left(d_m^2(\mathbf{x}_i, \mu_1) - d_m^2(\mathbf{x}_i, \mu_2) \right) = \quad (3)$$

$$\forall i : y_i \sum_{j=1}^p m_j^2 \left((x_{ij} - \mu_{1j})^2 - (x_{ij} - \mu_{2j})^2 \right) = \quad (4)$$

$$\forall i : y_i \sum_{j=1}^p m_j^2 \left(2x_{ij}(\mu_{2j} - \mu_{1j}) + (\mu_{1j}^2 - \mu_{2j}^2) \right) \geq 1, \quad (5)$$

which imposes a margin between the classes similarly to the margin defined for the Support Vector Machine. In case the classes are not linearly separable we must allow for misclassifications. To this end, we introduce a slack variable ξ_i for each object constraint (5). These slack variables effectively release the constraints.

The objective of the LESS classifier is to find those weights that minimize the number of misclassifications ($\sum \xi_i$) while using as few features as possible ($\sum m_j^2$). The factors m_j that weigh the dimensions appear squared in the minimization term, so the optimization problem may seem quadratic. However, with some rewriting the problem turns into a linear optimization problem with linear constraints, also called a Linear Program (LP). We substitute $w_j = m_j^2$ and get the Lowest Error

in a Sparse Subspace (LESS):

$$\min \sum_{j=1}^p w_j + C \sum_{i=1}^n \xi_i, \quad (6)$$

$$\text{subject to: } \forall i : y_i \sum_{j=1}^p w_j \left(2x_{ij}(\mu_{2j} - \mu_{1j}) + (\mu_{1j}^2 - \mu_{2j}^2) \right) \geq 1 - \xi_i, \quad (7)$$

$$\forall i : \xi_i \geq 0, \quad (8)$$

$$\forall j : w_j \geq 0. \quad (9)$$

Fortunately, these Linear Programming problems can be solved efficiently even with thousands of variables (w_j 's and ξ_i 's in this case) and constraints. It can be seen that in contrast with the Nearest Shrunken Centroids classifier [21], the LESS classifier finds the optimal feature weights in a combinatorial fashion. As such, the LESS classifier is more flexible, or, in other words, it has less model bias.

We now focus on the LESS constraints as formulated in (7). These constraints can be written as:

$$\forall i : y_i \mathbf{w}' \Phi(\mathbf{x}_i) \geq 1 - \xi_i, \quad (10)$$

$$\text{with } \mathbf{w} = (w_1 \ w_2 \ \cdots \ w_p)' \quad \text{and} \quad \Phi(\mathbf{x}_i) = (\phi(x_{i1}) \ \phi(x_{i2}) \ \cdots \ \phi(x_{ip}))', \quad (11)$$

$$\text{where } \phi(x_{ij}) = 2x_{ij}(\mu_{2j} - \mu_{1j}) + (\mu_{1j}^2 - \mu_{2j}^2). \quad (12)$$

So, the function ϕ scales and translates the data vectors using the class means. Denoted as such LESS shows similarities with SVM formulations. We have a closer look at this in the *Related Work* Section. We call the mapping (12) ϕ_μ and the corresponding LESS version LESS_μ . Further, we propose a few other mappings leading to other LESS variants with interesting properties.

First, the choice for the Nearest Mean Classifier was motivated for its low complexity, and therefore its stability. Nevertheless, the computation of the mean per dimension as we did so far is sensitive for outliers. An alternative, more robust estimate of the class prototype is the median $\tilde{\mu}_k = \text{median}(X_k)$. The median can easily be substituted in (7) or (12) leading to a more robust LESS realization. We denote LESS with medians as class prototypes with $\text{LESS}_{\tilde{\mu}}$ and the mapping function with $\phi_{\tilde{\mu}}$.

Second, we propose a mapping that scales the distances to the class means with the variance in the respective dimension. This mean/variance mapping $\phi_{\mu\sigma}$ results in the non-linear (quadratic) $\text{LESS}_{\mu\sigma}$

classifier and is defined as:

$$\phi_{\mu\sigma} = \frac{(x_{ij} - \mu_{1j})^2}{\sigma_{1j}^2} - \frac{(x_{ij} - \mu_{2j})^2}{\sigma_{2j}^2} \quad (13)$$

Clearly, the previous extensions to LESS can also be combined. Especially, the robust class prototypes can be combined with variance scaling resulting in LESS $_{\tilde{\mu}\tilde{\sigma}}$. This means that in (13) μ_k must be substituted with the median $\tilde{\mu}_k$. It is then also more natural to replace the variances σ_k^2 with the median squared deviation $\tilde{\sigma}_k^2$ in the same equation. This results in the mapping $\phi_{\tilde{\mu}\tilde{\sigma}}$, where $\tilde{\sigma}_k^2$ is defined as:

$$\tilde{\sigma}_{kj}^2 = \text{median}(\{(x_j - \tilde{\mu}_{kj})^2 \mid x \in X_k\}) \quad (14)$$

3 Related Work

In this section, we review four related classifiers. The first three classifiers can be formulated as a mathematical programming problem, either a linear program or a convex quadratic program. The last classifier we review can be computed directly. Consequently, all these models can be optimized efficiently.

Ridge Regression with LASSO

With ridge regression a linear classifier is learned by minimizing the squared distance to the class labels $\{-1, +1\}$. Additionally, the squared L_2 -norm of the weight vector is minimized. In [20], a modification was proposed that replaces the L_2 -norm of the weight vector with the L_1 -norm. This modification, with the so-called Least Absolute Shrinkage and Selection Operator (LASSO), was motivated by earlier work in [6]. The classifier is defined as follows:

$$\min \sum_{i=1}^n (y_i - \mathbf{w}'\mathbf{x}_i - b)^2 + C|\mathbf{w}| \quad (15)$$

The advantage of this formulation is that in the final solution to this minimization problem most weight entries w_j are effectively forced to 0 instead of to some small number. Accordingly, the method implicitly selects features or dimensions to be used in the resulting linear classifier, similarly to the proposed LESS classifier.

The problem as denoted in (15) is a convex quadratic programming problem without additional constraints for which efficient solutions exist.

Linear Support Vector Machine (SVM)

The last decade the Support Vector Machine [7], [23] has become a heavily researched and applied classifier. This is partially for its theoretical foundations in computational learning theory and, of course, for its good performance. Although the theory leaves room for improvement, in particular when the classifier must allow for misclassifications, the SVM is certainly a classifier with high potential. Here we only consider the linear Support Vector Machine, which is adequate for the high-dimensional data under consideration. The linear Support Vector Machine is defined as:

$$\min \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i, \quad (16)$$

$$\text{subject to: } \forall i : y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \quad (17)$$

Since the squared L_2 -norm of the feature weight vector is minimized, the SVM generally uses all features though some of them with small weights. The slack variables ξ_i , however, are linearly minimized. Accordingly, many objects have $\xi_i = 0$, so that they are effectively ruled out. The remaining objects are called support vectors or support objects, hence the name of the classifier.

The corresponding optimization problem is a convex quadratic programming (QP) problem with linear constraints. Usually this problem is rewritten into its dual form. Using that formulation, non-linear kernels can easily be substituted. Since we mainly consider linear classifiers we stay with the primal problem notation, which is easier to interpret and it compares more easily to the other classifiers we review.

If we compare the SVM equations (16)-(17) to the LESS model (6)-(12) we see remarkable similarities. The main difference is that for LESS the data vectors are mapped, either linearly $(\phi_\mu, \phi_{\bar{\mu}})$ or non-linearly $(\phi_{\mu\sigma}, \phi_{\bar{\mu}\bar{\sigma}})$, and that a different weight vector norm is minimized. Further, LESS has an explicit bias term $(\mu_{1j}^2 - \mu_{2j}^2)$, while the bias term b in (17) is estimated. Moreover, for LESS the weights $w_j \geq 0$.

L_1 Support Vector Machine (Liknon)

The Liknon classifier was designed for class prediction with a low number of relevant features [3], which are the same design criteria as for LESS. The model is very similar to the linear SVM model (16)-(17). The only difference is that the L_1 -norm of the weight vector is minimized instead of the

squared L_2 -norm. As such Liknon is considered an L_1 -SVM. For more on L_1 -SVM classifiers see for instance [5] and [11]. The motivation for applying the L_1 -norm is the same as for the LASSO (see previous Section). That is, the L_1 -norm forces the weight entries w_j to 0 so the classifier explicitly selects a feature subset. The minimization problem is stated as:

$$\min |\mathbf{w}| + C \sum_{i=1}^n \xi_i \quad (18)$$

$$\text{subject to: } \forall i : y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \quad (19)$$

This seemingly non-linear problem can be rewritten into a linear optimization problem with linear constraints by substituting $\mathbf{w} = (\mathbf{u} - \mathbf{v})$ and $|\mathbf{w}| = (\mathbf{u} + \mathbf{v})$ with $u_j, v_j \geq 0$.

As an SVM variant also Liknon resembles LESS to a certain extent. In addition to the L_2 SVM similarities, the L_1 -norm that is minimized in (18) is the same as the weight vector minimization term in (6) given that for LESS the weights $w_j \geq 0$.

Nearest Shrunken Centroids (NSC)

The last classifier that we consider is not formulated as a mathematical programming problem. This classifier assigns objects to the class to which shrunken centroid they are closest, therefore the name Nearest Shrunken Centroids [21]. The shrunken centroids are the means of the classes, for which each of the feature components are reduced by a factor Δ until the feature component becomes 0. This classifier is defined as follows:

$$f(\mathbf{x}) = \begin{cases} -1, & \text{if } \sum_{j=1}^p \frac{(x_j - |\mu_{1j}|_\Delta)^2}{(s_1 + s_0)^2} - \frac{(x_j - |\mu_{2j}|_\Delta)^2}{(s_2 + s_0)^2} < 0, \\ +1, & \text{otherwise,} \end{cases} \quad (20)$$

where s_0 is a regularization term and $|\mu_{kj}|_\Delta = \mu_{0j} + m_k(s_j + s_0)|d_{kj}|_\Delta$ is the shrunken centroid with:

$$|d_{kj}|_\Delta = \text{sign}(d_{kj})(|d_{kj}| - \Delta) \quad \text{and} \quad s_j^2 = \frac{1}{n-2}(\sigma_{1j}^2 + \sigma_{2j}^2). \quad (21)$$

In this expression Δ is the shrinkage parameter. Without shrinkage ($\Delta = 0$) the means are not scaled or $|\mu_{kj}|_\Delta = \mu_{kj}$. Finally, d_{kj} is defined as:

$$d_{kj} = \frac{\mu_{kj} - \mu_{0j}}{m_k(s_j + s_0)} \quad \text{and} \quad m_k = \sqrt{\frac{1}{n_k} + \frac{1}{n}} \quad \text{with } k \in \{1, 2\}. \quad (22)$$

Dataset	Ref.	class I (n_1)	class II (n_2)	total (n)	features (p)
Colon	[1]	22	40	62	1908
Leukemia	[12]	25	47	72	3571
Metastasis	[22]	34	44	78	4919
Ionosphere	[18]	126	225	351	34
Sonar	[13]	97	111	208	60

Table 1: Overview of the characteristics of the data sets used in the experiments.

By scaling the classes with the variances the NSC results in a quadratic classifier similar to $\text{LESS}_{\mu\sigma}$.

4 Experiments

We tested the classifiers on artificial and several real-life high-dimensional data sets. The real-life data sets range from biological tissue classification with micro-arrays to some well-known data sets from the Machine Learning Database Repository [4]. In Table 1, we list the characteristics of the real-life data sets.

For the optimization of the linear programs for LESS and Liknon we used the GLPK solver [10]. The Support Vector Machine is implemented using the quadratic programming solver from [16]. Further, we used the efficient LASSO implementation as made available by the authors from [17].

In the experiments, we implemented LESS with variance scaling slightly different from (13). In order to protect against degenerate cases, where the class variance in a certain dimension is very low, we added the mean of the variances over all dimensions $\text{mean}_j(\sigma_{kj}^2)$ to the variance per dimension σ_{kj}^2 . Moreover, we added two LESS_{μ} variants in order to separate the benefits of the data mapping (12) from the other differences between LESS and Liknon. The first is LESS_{μ}^{\pm} for which we removed the constraint that all weights should be non-negative. The minimisation term (6) then turns into the Liknon term (18) subject to (7) and (8). The second modification is $\text{LESS}_{\mu}^{\pm b}$ which additionally estimates a bias term in (10) leading to (18) subject to:

$$\forall i : y_i \mathbf{w}'(\Phi_{\mu}(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \quad (23)$$

Accordingly, $\text{LESS}_{\mu}^{\pm b}$ equals Liknon where the data is mapped with Φ_{μ} . All described methods have one fundamental free parameter, called C or Δ . Since the value of these parameters can not be derived directly from the model or the data, we determine them with a ten-fold cross-validation proto-

col. For the accurate tuning of the classifier parameter we use 10-fold cross-validation repeated three times for a range of parameter values. The parameter value that results in the lowest average error is considered optimal for the respective data(sub)set. The evaluation of the consequent classifier is done through 10-fold cross-validation repeated 10 times. In each fold of the cross-validation procedure the respective classifier parameter is optimized. This value is used to train the classifier on the whole fold. As usual, the trained classifier is tested on the left-out part.

4.1 Experimental Results

We started with the micro-array data sets that were preprocessed as described in the respective publications [1], [12], [22]. In Table 2, we list the errors for all classifiers. With the Colon data set [1] the differences between the classifiers are small. Only LASSO stays behind. The Leukemia data set [12] is easier for the classifiers that use more features, being the SVM (all features) and NSC (± 2700), see Table 3. Remarkably, the LESS implementation that utilizes the median and median squared deviation scaling performs significantly better than the other LESS versions. Moreover, the number of features that this classifier used is still very low. The Breast Cancer metastasis data set [22] is a more difficult data set. On this data set the SVM performs worst and also Liknon has a higher error. The NSC again utilizes substantially more features on the average than all other classifiers (except the SVM of course that always uses all features). On the Ionosphere data set the quadratic LESS with variance scaling performs clearly the best. Also the number of features that it uses is low. Also on the Sonar data set this quadratic LESS variant outperforms all other classifiers. Only the other LESS versions use fewer features on this data set. The other quadratic classifiers, $LESS_{\tilde{\mu}\tilde{\sigma}}$ and NSC, do not achieve similar high performance. The last lines in Table 2 and Table 3 give the averages over all data sets. It follows that $LESS_{\mu\sigma}$ has the lowest average error and standard $LESS_{\mu}$ uses the fewest dimensions on the average.

Among the LESS versions the ones with robust class prototype estimation (using $\tilde{\mu}$ and $\tilde{\sigma}$) are especially interesting. We did an additional experiment to exemplify the conditions under which these classifiers can be profitable. We constructed an artificial data set to investigate the feature selection capabilities and the robustness of the classifiers. This data set consist of two Gaussian distributed classes in a 200 dimensional feature space with unit variance in all feature directions. The means of the two classes differ one unit in the first 20 dimensions, and are identical in the next 180 dimensions.

Dataset	LESS $_{\mu}$	LESS $_{\mu}^{\pm}$	LESS $_{\mu}^{\pm b}$	LESS $_{\mu\sigma}$	LESS $_{\bar{\mu}}$	LESS $_{\bar{\mu}\bar{\sigma}}$	SVM	LASSO	Liknon	NSC
colon	11.7 ± 2.6	12.8 ± 2.4	12.0 ± 2.3	12.7 ± 2.8	11.5 ± 1.5	13.5 ± 1.1	13.8 ± 1.8	16.2 ± 2.9	11.9 ± 2.8	12.7 ± 2.0
leukemia	10.3 ± 2.2	9.9 ± 2.4	9.5 ± 1.5	10.8 ± 1.7	10.3 ± 2.0	7.8 ± 2.8	2.7 ± 0.7	8.8 ± 2.9	11.7 ± 1.4	3.5 ± 0.7
metas	33.4 ± 3.4	33.2 ± 2.5	31.3 ± 2.2	34.0 ± 3.8	33.7 ± 2.9	31.7 ± 4.4	39.8 ± 2.7	31.6 ± 2.1	36.8 ± 3.5	31.5 ± 3.9
iono	18.0 ± 1.1	19.0 ± 0.6	16.0 ± 1.0	9.5 ± 0.7	21.3 ± 1.0	19.9 ± 1.4	16.4 ± 0.6	21.8 ± 1.0	16.6 ± 1.0	22.4 ± 0.8
sonar	24.5 ± 1.8	24.9 ± 2.4	24.3 ± 1.8	21.0 ± 1.0	24.5 ± 2.2	23.8 ± 1.2	24.6 ± 1.8	26.4 ± 1.4	26.8 ± 1.5	26.9 ± 1.5
Average	19.6	20.0	18.6	17.6	20.3	19.3	19.4	21.0	20.8	19.4

Table 2: The mean and standard deviation of the generalization error obtained with cross-validation tests.

Dataset	LESS $_{\mu}$	LESS $_{\mu}^{\pm}$	LESS $_{\mu}^{\pm b}$	LESS $_{\mu\sigma}$	LESS $_{\bar{\mu}}$	LESS $_{\bar{\mu}\bar{\sigma}}$	SVM	LASSO	Liknon	NSC
colon	13.3 ± 1.2	14.0 ± 0.9	13.7 ± 1.1	28.8 ± 2.2	16.7 ± 1.0	15.5 ± 1.5	1908	30.0 ± 3.1	20.7 ± 1.6	60.4 ± 31.8
leukemia	7.0 ± 0.8	7.8 ± 1.1	1.9 ± 0.4	9.4 ± 1.0	6.7 ± 0.7	9.3 ± 0.7	3571	26.2 ± 2.5	7.8 ± 1.8	2.7e3 ± 258
metas	5.0 ± 1.4	5.7 ± 2.4	6.1 ± 2.2	13.8 ± 1.8	4.1 ± 1.7	12.3 ± 1.8	4919	5.7 ± 2.0	17.8 ± 3.1	87.3 ± 153
iono	15.7 ± 1.5	15.0 ± 3.2	28.0 ± 1.8	11.1 ± 0.4	13.9 ± 1.0	6.1 ± 0.2	34	31.2 ± 1.5	27.7 ± 1.1	6.8 ± 1.2
sonar	12.9 ± 1.9	11.5 ± 3.7	12.3 ± 1.7	18.6 ± 2.1	18.1 ± 1.3	16.1 ± 1.9	60	25.6 ± 3.5	25.5 ± 3.8	23.2 ± 1.2
Average	10.8	10.8	12.4	16.3	11.9	11.9	2100	23.8	19.9	578.2

Table 3: The mean and standard deviation of the number of dimensions found in the cross-validation tests.

Result	LESS $_{\mu}$	LESS $_{\mu}^{\pm}$	LESS $_{\mu}^{\pm b}$	LESS $_{\mu\sigma}$	LESS $_{\bar{\mu}}$	LESS $_{\bar{\mu}\bar{\sigma}}$	SVM	LASSO	Liknon	NSC
Errors	39.1 ± 11	39.6 ± 10	35.9 ± 8.9	42.7 ± 3.9	28.9 ± 14	14.5 ± 4.0	18.2 ± 4.4	44.9 ± 4.9	37.1 ± 11	31.2 ± 3.4
Dims	11.2 ± 9.2	11.4 ± 9.7	8.3 ± 7.6	7.4 ± 5.5	8.4 ± 6.1	11.5 ± 3.6	200	10.5 ± 10	16.4 ± 12	163 ± 78

Table 4: The mean and standard deviation of the error and the number of dimensions in the validation tests with the artificial data sets.

One of the objects is an outlier, because its first 20 feature values are put to $(20, 20, \dots, 20)$. For these experiments, the validation is done using a large reference data set (5000 objects per class) that is generated according to the same data distribution.

In Table 4 we list the errors and resulting number of dimensions, computed as described above, for all classifiers. For these artificially generated data sets, we see the clear advantage of using the robust version of LESS. The $LESS_{\bar{\mu}\bar{\sigma}}$ clearly outperforms the other classifiers. The non-robust versions of LESS, using μ and σ are disturbed by the outlier. Also the SVM performs well, but for this performance it uses all features.

From the experiments the following conclusions appear to be justified with respect to the LESS classifier. With extremely low sample size problems all LESS versions perform similarly. This is partially because the variances cannot be estimated reliably. Accordingly, assuming similar variances for each class may be better. Also, using the robust version (see e.g. $LESS_{\bar{\mu}\bar{\sigma}}$ on Leukemia) seems to be a good choice, but more experiments are needed. When the number of objects allows for a more reliable estimation of the class variances per dimension, like with the Ionosphere data set and the Sonar data set, this certainly improves the performance of the LESS classifier. The constraint that forces all weights w_j to be non-negative as incorporated in the LESS model does not appear to be significant. This follows from the comparable performance of $LESS_{\mu}$ and $LESS_{\mu}^{\pm}$. The introduction of a bias term as in $LESS_{\mu}^{\pm b}$ appears to influence the model. Especially, with respect to the number of utilized dimensions. More experiments are needed to investigate when this bias is beneficial.

If we compare the performance of the LESS classifier to the other classifiers, then it is clear that LESS uses the fewest features. Moreover, except for the Leukemia data set LESS or one of its variants has the lowest error. Especially on the Leukemia dataset the SVM classifier performs clearly better than the other classifiers. Though, it has to be noted that it does not result in a sparse solution.

5 Conclusions

In this paper, we introduced the LESS classifier, which stands for Lowest Error in a Sparse Subspace. The LESS classifier is based on the Nearest Mean Classifier where each dimension has an added weighting factor such that the relevance and importance of each feature can be expressed. For the learning of the weighting factors from a training data set the classifier model is formulated as a Linear

Programming problem. Accordingly, it can be optimized effectively and efficiently.

Besides the standard LESS classifier we proposed a number of extensions. First, we showed how the variance per class can be incorporated in case the classes have different variances. The consequent quadratic classifier can still easily be optimized as a Linear Program. Further, we showed that also robust estimations of the mean and variance can be applied with the same profitable optimization properties.

The main difference between the LESS and other classifiers like the SVM, Liknon, and LASSO is that LESS utilizes a model of the data. Such a model can be beneficial in cases where the data is insufficient to reliably estimate a suitable discriminant based on training data only. This is especially true for small sample size problems. In the experiments, we compared these classifiers to the proposed LESS classifier alternatives. We compared both the resulting generalization error and the number of utilized features. It turns out that all LESS variants have some benefits. Overall the LESS classifiers perform competitively while they utilize the lowest number of features.

We also showed that the LESS model inherently contains a data mapping, which is part of the difference between LESS, SVM and Liknon. In the experiments we incorporated this data mapping in the Liknon classifier. It turned out that the mapping is a fundamental part of the strength of the LESS classifier. Further research should be directed towards exploiting the data mapping in other classifiers.

Acknowledgements This research is supported by PAMGene within the BIT program ISAS-1. We thank Dr. Berwin Turlach for his help and for making the LASSO software available.

References

- [1] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences (PNAS)*, 96(12):6745–6750, 1999.
- [2] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

-
- [3] C. Bhattacharyya, L.R. Grate, A. Rizki, D. Radisky, F.J. Molina and M.I. Jordan, M.J. Bissel, and I.S. Mian. Simultaneous classification and relevant feature identification in high-dimensional spaces: Application to molecular profiling data. *Signal Processing*, 83:729–743, 2003.
- [4] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
- [5] P.S. Bradley and O.L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proc. 15th International Conf. on Machine Learning*, pages 82–90. Morgan Kaufmann, San Francisco, CA, 1998.
- [6] L. Breiman. Better subset selection using non-negative garotte. Technical report, University of California, Berkeley, 1993.
- [7] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [8] J.C.W. Debus and V.J. Rayward-Smith. Feature subset selection within a simulated annealing data mining algorithm. *Journal of Intelligent Information Systems*, 9(1):57–81, 1997.
- [9] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, 2001.
- [10] Free Software Foundation. *GNU Linear Programming Kit*. <http://www.gnu.org>.
- [11] G. Fung and O.L. Mangasarian. Data selection for support vector machine classifiers. In R. Ramakrishnan and S. Stolfo, editors, *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 2094, pages 64–70, Boston, August 2000. ACM.
- [12] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

- [13] R.P. Gorman and T.J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1:75–89, 1988.
- [14] M. Karzynski, Á Mateos, J. Herrero, and J. Dopazo. Using a genetic algorithm and a perceptron for feature selection and supervised class learning in DNA microarray data. *Artificial Intelligence Review*, 20:39–51, 2003.
- [15] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, December 1997.
- [16] *Mosek Optimization Toolbox*. <http://www.mosek.com>.
- [17] M.R. Osborne, B. Presnell, and B.A. Turlach. On the LASSO and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000.
- [18] V.G. Sigillito, S.P. Wing, L.V. Hutton, and K.B. Baker. Classification of radar returns from the ionosphere using neural networks. *John Hopkins APL Technical Digest*, 10:262–266, 1989.
- [19] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, London, 1999.
- [20] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.
- [21] R. Tibshirani, T. Hastie, B. Balasubramanian, and G. Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences (PNAS)*, 99(10):6567–6572, 2002.
- [22] L.J. van 't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A.M. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, and S.H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, January 2002.
- [23] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [24] H. Zhang and G. Sun. Feature selection using tabu search method. *Pattern Recognition*, 35:701–711, 2002.