Robustness Analysis of Eleven Linear Classifiers in Extremely High–Dimensional Feature Spaces

Ludwig Lausser[1] and Hans A. Kestler[1,2,⋆]

[1] Department of Internal Medicine I, University Hospital Ulm, Germany
ludwig.lausser@uni-ulm.de
[2] Institute of Neural Information Processing, University of Ulm, Germany
hans.kestler@uni-ulm.de

Abstract. In this study we address the linear classification of noisy high-dimensional data in a two class scenario. We assume that the cardinality of the data is much lower than its dimensionality. The problem of classification in this setting is intensified in the presence of noise. Eleven linear classifiers were compared on two-thousand-one-hundred-and-fifty artificial datasets from four different experimental setups, and five real world gene expression profile datasets, in terms of classification accuracy and robustness. We specifically focus on linear classifiers as the use of more complex concept classes would make over-adaptation even more likely. Classification accuracy is measured by mean error rate and mean rank of error rate. These criteria place two large margin classifiers, SVM and ALMA, and an online classification algorithm called PA at the top, with PA being statistically different from SVM on the artificial data. Surprisingly, these algorithms also outperformed statistically significant all classifiers investigated with dimensionality reduction.

1 Introduction

Classification is one of the basic tasks in machine learning. Many different classification methods were proposed (see e.g. [1,2,3]). In the standard inductive setting, a classifier will be selected according to a set of training examples and its accuracy is tested on a set of test examples. Problems arise if a collected dataset contains more features than samples. In this case even simple classifiers have the complexity to adapt perfectly to a given training set and loose their ability of generalization (overfitting) [4]. Dimensionality reduction methods, like for example PCA, ICA, can antagonize this problem but complicate the interpretation of a classifier in terms of its original input space [5,6]. The problem of overfitting is increased in real life applications. The single datapoint can be affected by measurement errors and a classifier will adapt to a noisy dataset.

Aim of this investigation is the influence of different types of noise on the performance of linear classifiers for high-dimensional data of low cardinality.

2 Classification

Classification is the task of predicting a categorial label $y \in \mathbf{Y}$ of a datapoint $x \in \mathbf{X}$. A classifier is a mapping $c : \mathbf{X} \rightarrow \mathbf{Y}$. In the following we will concentrate on binary

[⋆] Corresponding author.

F. Schwenker and N. El Gayar (Eds.): ANNPR 2010, LNAI 5998, pp. 72–83, 2010.

classification $\mathbf{Y} = \{+1, -1\}$ and real valued input spaces $\mathbf{X} \subseteq \mathbb{R}^n$. A classifier is chosen from a concept class $\mathbf{C}$, a set describing all classifiers fulfilling some model assumptions. The aim is to find the classifier $c^* \in \mathbf{C}$ which minimizes the number of errors over the distribution of all possible labeled pairs $D(x,y)$

$$c^* = \underset{c}{\operatorname{argmin}} \frac{1}{2} \int |c(x) - y| dD(x,y). \tag{1}$$

The distribution of $D(x,y)$ is usually not known. In this case a classifier c is selected (trained) by a learning algorithm $t(\mathbf{C},\mathbf{S}) \rightarrow c$ according to a finite set $\mathbf{S}$ of m examples

$$\mathbf{S} = \mathbf{S}(\mathbf{P},\mathbf{N}) = \{(x,+1)\,|\,x \in \mathbf{P}\} \cup \{(x,-1)\,|\,x \in \mathbf{N}\}. \tag{2}$$

Here $\mathbf{P}$ denotes the set of k (positive) examples of the first class and $\mathbf{N}$ denotes the set of l (negative) examples of the second class. The error rate of a classifier is estimated on an independent (test-) dataset $\mathbf{S}' = \mathbf{S}(\mathbf{P}',\mathbf{N}')$ with $\mathbf{S} \cap \mathbf{S}' = \emptyset$. This estimator can be formalized as

$$f_{err} = \frac{1}{2|\mathbf{S}'|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathbf{S}'} |c(\mathbf{x}) - \mathbf{y}|. \tag{3}$$

2.1 Linear Classifiers

The concept class of linear classifiers is given by

$$\mathbf{C}_{lin} = \{c(x) = \operatorname{sign}(\omega^T x - \theta)\,|\,\omega \in \mathbb{R}^n, \theta \in \mathbb{R}\}. \tag{4}$$

The decision boundaries of these classifiers are linear equations of the form

$$\omega^T x = \theta. \tag{5}$$

ω and θ are normally substituted by $\omega := \omega/||\omega||_2$ and $\theta := \theta/||\omega||_2$ in order to gain a unique representation of each classifier. Here $||\cdot||_2$ denotes the Euclidian norm. In a geometric interpretation ω can be seen as the norm vector of the line. The threshold θ can be seen as the line's distance to the origin.

In this analysis we focus on datasets with higher dimensionality than cardinality ($m \ll n$), which is the basic scenario in many tasks, like image analysis [7], speech recognition [8] and gene expression analysis [9]. Although linear classifiers are very simple models, they tend to overfit on such datasets. This was shown, for example, by Cover's theorem [10] stating that a database of m-datapoints (in general position) within a n-dimensional space can be separated in an arbitrary way with probability

$$P(m,n) = \left(\frac{1}{2}\right)^{m-1} \sum_{k=0}^{n-1} \binom{m-1}{k}. \tag{6}$$

If the ratio n/m is greater than 0.5, $P(m,n)$ is rapidly increasing towards 1 and a classifier without any training error can be found for an arbitrary dataset of these dimensions.

3 Training Algorithms

This section contains a brief description of the eleven training algorithms that were used in this study. The algorithms are divided into model-based algorithms (3.1), linear and quadratic programming algorithms (3.2) and iterative algorithms (3.3).

3.1 Model Based Classifiers

The algorithms listed here were created with assumptions on the class densities.

*Fisher Linear Discriminant Analysis (***LDA***).* The LDA classifier is built with the assumption, that both class densities are Gaussians with a common covariance Σ. The hyperplane calculated by this algorithm minimizes the error for datapoints chosen according to these class densities. For this the inverse of Σ is needed. On a real dataset the estimate $\hat{\Sigma}$ of Σ has to be used. $\hat{\Sigma}$ will become singular for datasets with higher dimensionality than cardinality. In this case the inverse of $\hat{\Sigma}$ is usually replaced by the Moore-Penrose Inverse. Besides the standard **LDA (mean)**, we have used a variant **LDA (median)**, for which the estimation of the class centroid was done by applying the median feature-wise.

*Nearest Centroid (***NC***).* The nearest centroid algorithm assumes, that both class densities are Gaussians with a common covariance of form $c \cdot \mathbf{I}$, $c \in \mathbb{R}$ ($\mathbf{I}$ is the identity matrix). In this way the NC can be seen as a special case of LDA. Under these assumptions only the class centroids have to be calculated for the final classification. For a new example the Euclidian distances to all centroids are calculated. The datapoint will receive the label of its nearest centroid.

*Nearest Shrunken Centroid (***NSC***) [11].* The nearest shrunken centroid is a feature reducing version of the NC. Here, additionally the class independent (overall) centroid is calculated. The main idea of the NSC is, that feature dimensions in which a class centroid is near to the overall centroid are not useful for characterizing the class. The class-wise centroids are shrunken feature–wise towards the overall centroid. If a single entry of a centroid gets negative, it is set to zero. The amount of shrinkage is determined by a set of parameters Δ. In this study experiments for $i \in \{1,\ldots,30\}$ different sets of shrinking parameters $\Delta_{ij} = i/30 * \max\{|d_{0j}|,|d_{1j}|\}$ were done. Here d_{0j} and d_{1j} denote the distances of the class-wise centroids to the overall centroids in feature dimension j.

3.2 Linear and Quadratic Programming Training Algorithms

This section contains algorithms, which optimize an objective function by a linear or quadratic program. In order to handle non-linear separable datasets a penalty term of slack variables ξ_i is added to the objective function. The tradeoff between the penalty term and the original objective function can now be regulated by a cost parameter C.

*Support Vector Machine (***SVM***) [2].* The support vector machine searches for the hyperplane, which maximizes the Euclidian distance between the hyperplane and the datapoints next to it (maximal L2 margin). This can be formulated as a quadratic problem for minimizing the Euclidian norm $||\omega||_2$ of ω.

$$\begin{aligned}
\min_{\omega,\xi} \quad & \|\omega\|_2^2 + C\sum_{i=1}^{N} \xi_i \\
\text{s.t.} \quad & \forall i : y_i(\omega^T x_i) - \theta \geq 1 - \xi_i \\
& \forall i : \xi_i \geq 0
\end{aligned}$$

LIKNON *[12].* The LIKNON algorithm can be seen as the L1 variant of the SVM. Minimizing the L1 norm $||\omega||_1$ of ω forces many ω_i to zero. The corresponding features of the datapoints will not be used for the final classification. In this way a feature reduction is achieved. The optimization problem of the LIKNON algorithm can be formalized as a linear program.

$$\begin{aligned}
\min_{\omega,b,\xi} \quad & \|\omega\|_1 + C\sum_{i=1}^{N} \xi_i \\
\text{s.t.} \quad & \forall i : y_i(\omega^T x_i) - \theta \geq 1 - \xi_i \\
& \forall i : \xi_i \geq 0
\end{aligned}$$

LESS *[13].* The LESS classifier belongs to the group of weighted centroid classifiers. Linear programming is used here to find a weight vector w, which minimizes the tradeoff between its L1 norm and the penalization term. Here again a feature selection is implicitly performed.

$$\begin{aligned}
\min_{w,\xi} \quad & \|w\|_1 + C\sum_{i=1}^{N} \xi_i \\
\text{s.t.} \quad & \forall i : y_i \sum_{j=1}^{M} w_j(2x_{ij}(\mu_{0,j} - \mu_{1j}) + (\mu_{0,j}^2 - \mu_{1j}^2)) \geq 1 - \xi_i \\
& \forall i : \xi_i \geq 0 \qquad \forall j : w_j \geq 0
\end{aligned}$$

μ_0 and μ_1 denote the class-wise centroids.

3.3 Iterative Training Algorithms

The algorithms in this section adapt the linear model in an iterative way. During each iteration, i.e. presentation of a data point, the classifier will be modified. Many iterative algorithms are designed for the online learning setting. In this scenario new labeled data points will be available one by one. The classifier will be adapted after receiving a new datapoint. In this study the online learning setting was simulated by iterating 10000 times through permuted versions of the original dataset.

Perceptron *[14].* The perceptron algorithm is one of the classical iterative algorithms. The hyperplane will be updated until it separates the data points correctly. No objective function is considered for the choice of the hyperplane.

ALMA *[15] and* **ROMMA** *[16].* The two algorithms ALMA (approximate maximal margin algorithm) and ROMMA (relaxed online maximum margin algorithm), approximate a maximum margin solution of the L2 margin in an iterative way.

*Passive Aggressive Algorithm (***PA***) [17].* The update rule of PA utilizes the hinge loss $l(\omega^*;(x,y)) = \max(0, 1-y(\omega x-\theta))$. Here ω^* denotes the vector of all classifier parameters $(\omega_1,\dots,\omega_n,-\theta)^T$. If a datapoint is classified correctly with a margin greater or equal to one, the hinge loss is equal to zero. Otherwise, the loss is increasing according to the distance between this margin and the datapoint. In the linear separable case, an update step of PA has to fulfill the constraint $l(\omega^*;(x_t,y_t)) = 0$. By this constraint not only a correct classification of x_t but also a minimal distance between the classifier and x_t is enforced. In each iteration t the classifier will be selected, which has the minimal modification of ω_t^*. If the classification of x_t fulfills the constraint, no modifications have to be done and PA is passive. Otherwise, PA forces aggressively the correct classification of x_t. For the linear inseparable case, the optimization problem can be formalized as

$$\omega_{t+1}^* = \operatorname*{argmin}_{\omega^* \in \mathbb{R}^{n+1}} \frac{1}{2}||\omega^* - \omega_t^*||_2^2 + C\xi^2$$
$$\text{s.t.} \qquad l(\omega^*;(x_t,y_t)) \leq \xi$$

This is equal to the PA-II variant proposed in [17].

4 Experimental Setup

The first part of this study is an empirical comparison of the classifiers in several artificial noise settings. For all experiments we use different datasets with a dimensionality of $n = 100$ and 25 datapoints for each of the two classes. A graphical visualization of the experimental setup can be found in Figure 1. For all algorithms various parameters settings were tested prior to the results given here. The best found parameter values were chosen and fixed for the results given in the following. We will first introduce some notation used within this section. The vector $\mathbf{1}$ is the vector, which is equal to 1 at each position. The vector $\mathbf{1}_x$ is equal to 1 in the first x positions and 0 on the other positions. The vector $\bar{\mathbf{1}}_x$ is defined as $\mathbf{1} - \mathbf{1}_x$. The function $d : \mathbb{R}^n \longrightarrow \mathbb{R}^{n\times n}$ converts a vector $v \in \mathbb{R}^n$ into a $n \times n$ - dimensional diagonal matrix. The main diagonal of this matrix is filled with the elements of v. A Gaussian distribution with mean μ and covariance Σ will be denoted by$\mathcal{N}(\mu,\Sigma)$. We will write $s(k,\Phi)$ to denote a function which creates a set of k datapoints chosen according to distribution Φ.

4.1 Breakdown Experiments

Here the test error of a classifier trained on samples $\mathbf{P} = s(k,\Psi_1)$ and $\mathbf{N} = s(l,\Psi_0)$ is compared to a classifier trained on contaminated samples $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{N}}$. A contaminated version $\tilde{\mathbf{X}} = \tilde{s}(\mathbf{X},i,\Phi)$ of a sample $\mathbf{X} \in \{\mathbf{P},\mathbf{N}\}$ is generated by replacing $i \leq |\mathbf{X}|$ examples by new ones, which were chosen according to distribution Φ. The number of contaminated datapoints is increased from 0 to $|\mathbf{X}|$ (class breakdown). For all experiments the test sets are chosen as $\mathbf{P}' = s(k,\Psi_1)$ and $\mathbf{N}' = s(l,\Psi_0)$. Each test was repeated on ten different samples. A table of the concrete experiments can be found in Table 1. The $mean_x$ experiment was done for $x \in \{5,10,25,50\}$. The $sd_{x'}$ experiment for $x' \in \{10^2,10^3,10^4\}$.

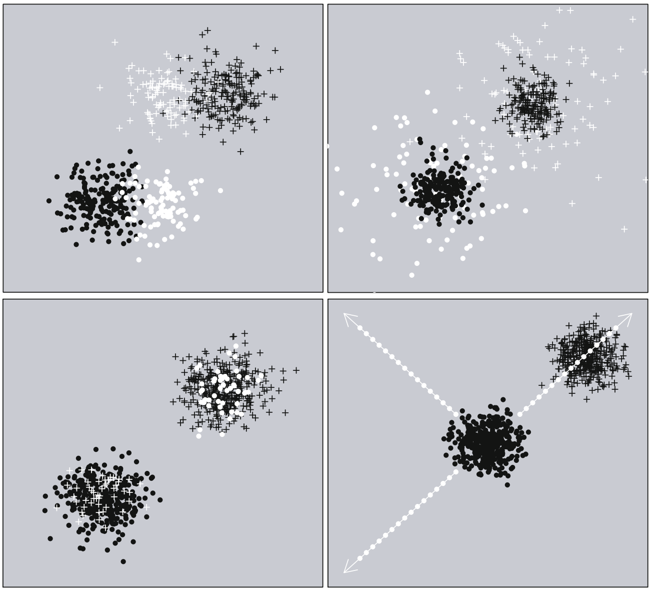

Fig. 1. The four settings of the artificial data experiments: Upper left: *mean* breakdown experiment. The distribution of the noisy datapoints differs from the distribution of the original datapoints in the first x dimensions of their common mean vector. The mean vector of the noisy datapoints is equal to 0 for the first x dimensions. Upper right: *sd* breakdown experiment. The noisy datapoints are chosen according to a Gaussian distribution with higher standard deviation than the standard deviation of the Gaussian of the original one. Lower left: *class* breakdown experiment. In this setting the noisy datapoints are chosen according to the distribution of the other class. Lower right: single outlier experiment. In this experiment a single noisy datapoint is moved in a certain direction. The datapoint is moved either towards the other class ($forward$), or away from the other class (*backwards*), or orthogonal to the other class (*sideways*).

Table 1. Breakdown experiments that were performed

experiment	$\mathbf{P}/\mathbf{P}'$	$\mathbf{N}/\mathbf{N}'$	$\tilde{\mathbf{P}}$	$\tilde{\mathbf{N}}$
$mean_x$	$s(25,\mathscr{N}(\mathbf{1},d(\mathbf{1})))$	$s(25,\mathscr{N}(-\mathbf{1},d(\mathbf{1})))$	$\tilde{s}(\mathbf{P},i,\mathscr{N}(\bar{\mathbf{1}}_x,d(\mathbf{1})))$	$\tilde{s}(\mathbf{N},i,\mathscr{N}(-\bar{\mathbf{1}}_x,d(\mathbf{1})))$
sd_x	$s(25,\mathscr{N}(\mathbf{1},d(\mathbf{1})))$	$s(25,\mathscr{N}(-\mathbf{1},d(\mathbf{1})))$	$\tilde{s}(\mathbf{P},i,\mathscr{N}(\mathbf{1},xd(\mathbf{1}))$	$\tilde{s}(\mathbf{N},i,\mathscr{N}(-\mathbf{1},xd(\mathbf{1})))$
$class$	$s(25,\mathscr{N}(\mathbf{1},d(\mathbf{1})))$	$s(25,\mathscr{N}(-\mathbf{1},d(\mathbf{1})))$	$\tilde{s}(\mathbf{P},i,\mathscr{N}(-\mathbf{1},d(\mathbf{1})))$	$\tilde{s}(\mathbf{N},i,\mathscr{N}(\mathbf{1},d(\mathbf{1})))$

4.2 Single Outlier Experiments

In this test a classifier trained on samples $\mathbf{P} = s(k,\Psi_1)$ and $\mathbf{N} = s(l,\Psi_0)$ is compared to a classifier trained on samples $\mathbf{P}_x^\tau = \bar{s}(\mathbf{P},x,\tau) = \mathbf{P} \cup \{10^\tau x\}$ and $\mathbf{N}$. Here x is a random point from the corresponding unit sphere and $\tau \in \{\ 1,\ldots,5\}$. For each x, ten different datasets were resampled. Some characteristics of the used datasets are given in Table 2.

Table 2. Single outlier experiments

experiment	$\mathbf{P}/\mathbf{P}'$	$\mathbf{N}/\mathbf{N}'$	$\mathbf{P}_x^\tau$
forward	$s(25,\mathcal{N}(\bar{\mathbf{1}}_1,d(\bar{\mathbf{1}}_1)))$	$s(25,\mathcal{N}(-\bar{\mathbf{1}}_1,d(\bar{\mathbf{1}}_1)))$	$\bar{s}(\mathbf{P},-\bar{\mathbf{1}}_1/\|\|-\bar{\mathbf{1}}_1\|\|,\tau)$
backwards	$s(25,\mathcal{N}(\bar{\mathbf{1}}_1,d(\bar{\mathbf{1}}_1)))$	$s(25,\mathcal{N}(-\bar{\mathbf{1}}_1,d(\bar{\mathbf{1}}_1)))$	$\bar{s}(\mathbf{P},\bar{\mathbf{1}}_1/\|\|\bar{\mathbf{1}}_1\|\|,\tau)$
sideways	$s(25,\mathcal{N}(\bar{\mathbf{1}}_1,d(\bar{\mathbf{1}}_1)))$	$s(25,\mathcal{N}(-\bar{\mathbf{1}}_1,d(\bar{\mathbf{1}}_1)))$	$\bar{s}(\mathbf{P},\mathbf{1}_1/\|\|\mathbf{1}_1\|\|,\tau)$

Table 3. Real data sets

name	#Fea	#Pos	#Neg
Bittner [18]	8067	19	19
Golub [19]	3571	47	25
Buchholz/Kestler [9]	169	37	25
Notterman [20]	7457	18	18
West [21]	7129	25	24

4.3 Experiments on Real Datasets

The classifiers were additionally compared on real data sets. For this setting a 10 × 10 cross-validation was chosen. A 10 × 10 cross-validation is a 10-fold repetition of a 10-fold cross-validation test on permuted variants of the initial dataset. The result will be the average error of the 10 single experiments. In a single 10-fold cross-validation test a dataset is divided into ten equal part. Nine parts are used to train the classifier and one part is used for testing. This procedure is repeated for all ten parts of the data. The mean error of these tests is calculated. The used data sets are chosen from the field of gene expression analysis. A list of the used data sets is given in Table 3.

5 Results

mean Breakdown results. The LDA-based algorithms are the only classifiers that are influenced for $mean_5$ and $mean_{10}$ (data not shown). They show error rates between 30% and 50% in these experiments, all other classifiers have zero error. Performance decreases for all other classifiers starting with $mean_{10}$, but is still much better than the LDA classifiers. For 25 and 50 noisy dimensions ($mean_{25}$, $mean_{50}$) performance decreases uniformly to 2% to 50% starting with 15 noisy datapoints. All classifiers are robust to this kind of noise, if the number of noisy datapoints is low. The NC-based classifiers are more robust on an increasing number of noisy datapoints and are only deteriorating for 24 or 25 noisy datapoints. Performance of SVM and LIKNON on lower noise levels is inferior to the iterative algorithms.

sd Breakdown results. The results of the *sd* breakdown experiments can be seen in Figure 2. The single rows contain the results of the sd_{100}, the sd_{1000} and the sd_{10000} breakdown experiment. The classifiers ALMA, PA, LIKNON and SVM are not influenced in any experiment until all datapoints were replaced by noisy datapoints. One noisy datapoint is enough to increase the error rate of the NC-based classifiers. This effect increases with a higher standard deviation. The LDA-based algorithms fluctuate around their initial error rate of about 20%. For all *sd* experiments there is a number of datapoints for which the error rate of LDA (median) is rapidly increasing towards 50%. The LDA (mean) has an error rate of 50% only for the number of 25 noisy datapoints.

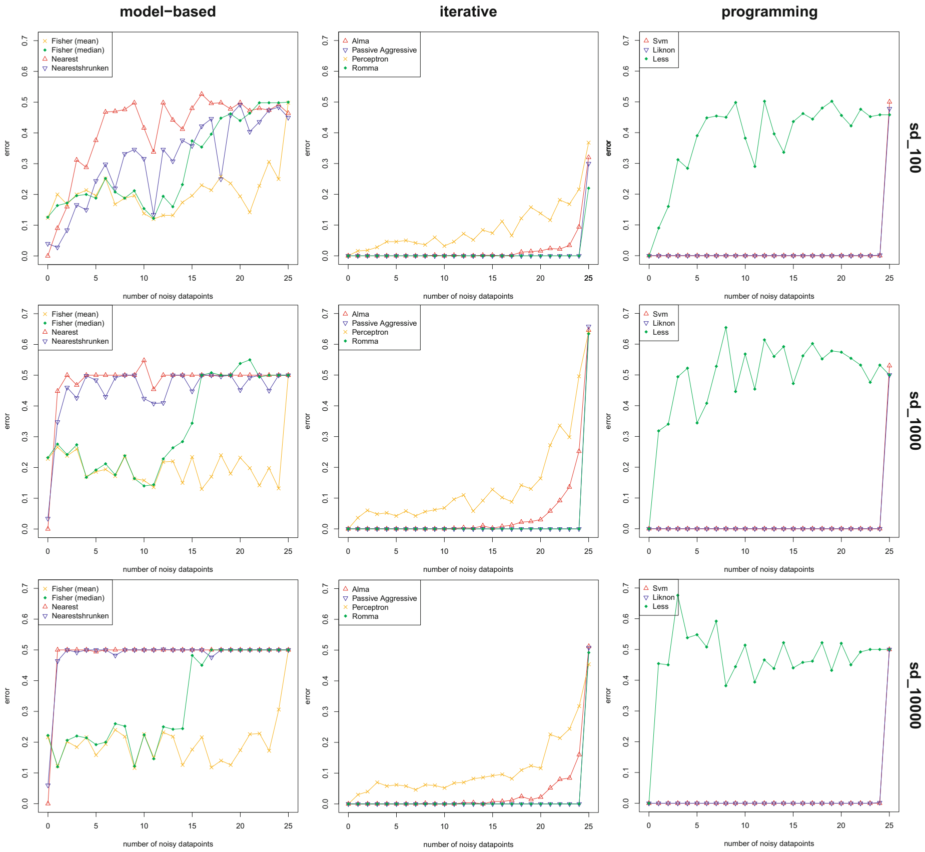

Fig. 2. Error curves of the *sd* breakdown experiment. The mean error of ten repetitions is shown. The number of noisy datapoints per class is given on the horizontal axis. The rows include the results from sd_{100}, sd_{1000}, and sd_{10000} (top to bottom).

class Breakdown results. The results of the *class* breakdown experiment are given in Figure 3. The classifiers show a linear increasing error rate according to the increasing number of noisy datapoints. An exception to this are the model-based classifiers. The classifiers NC, NSC and LESS show a flat error curve until a level of 13 noisy datapoints per class is reached. The LDA-based classifiers fluctuate around the 50% error rate in the range of 5 and 20 noisy datapoints.

Single outlier results. The error curves of the single outlier experiments are given in Figure 4. Only the model based classifiers were influenced in the experiments *backwards* and *sideways*. A exception to this is the NSC in the *sideways* experiment. The other classifiers are only affected in the *forward* experiment.

Average ranking on artificial datasets. The rank over all classifiers was calculated for all single experiments and noise levels. The mean rank is shown in Table 4. The best

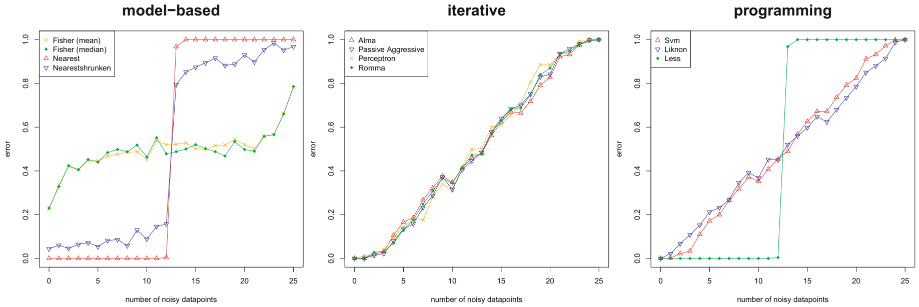

Fig. 3. Error curves of the *class* breakdown experiment over an increasing number of noisy datapoints per class. The mean error of ten repetitions is shown.

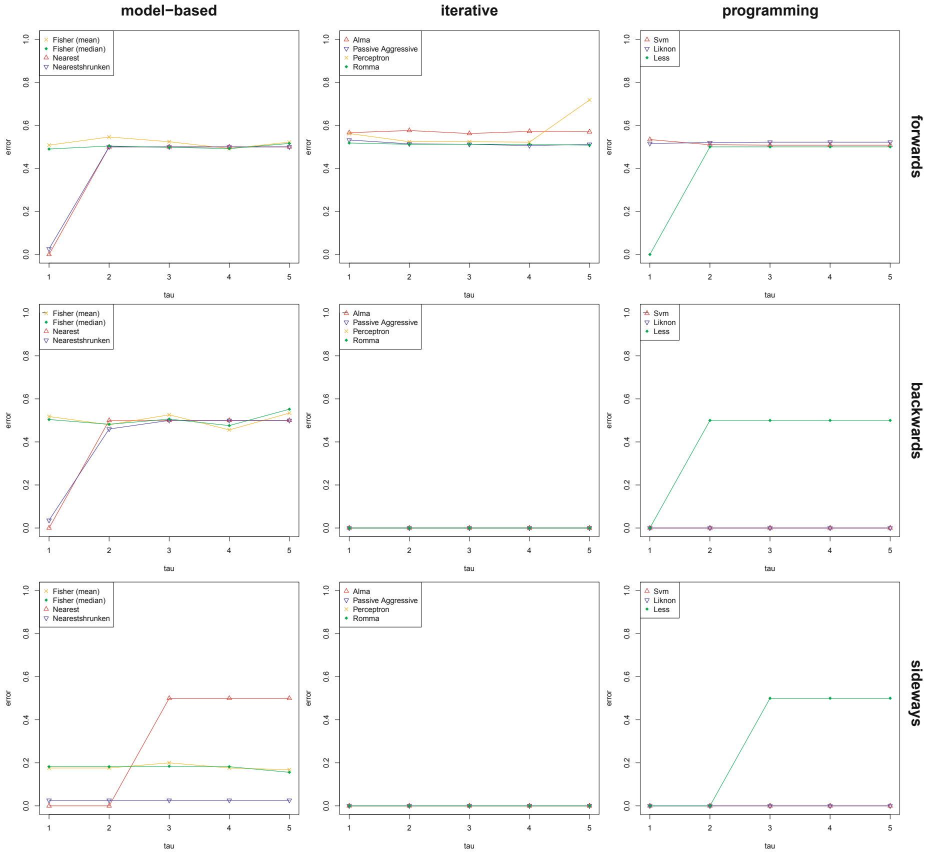

Fig. 4. Error curves of the single outlier experiments. The mean error of ten repetitions is shown. The distance from the outlier to the class centroid is given on the horizontal axis.

Table 4. Average ranks over all experiments on artificial datasets

LDA (mean)	LDA (median)	NC	NSC	PER	ROMMA	ALMA	PA	SVM	LIK	LESS
8.62	9.02	6.48	6.91	5.49	4.41	4.62	4.41	4.60	5.06	6.37

Table 5. Results of the real data experiments. The results of the 10 × 10 cross-validation are given by the mean errors in percent and standard deviations. The last two columns show the average error and the average rank over all datasets.

	Bittner	Golub	Notterman	Buchholz	West	Average error	Average rank
LDA (mean)	46.84 ± 6.88	42.22 ± 6.84	34.17 ± 7.53	45.32 ± 7.35	44.69 ± 6.83	42.65	9.9
LDA (median)	46.84 ± 7.00	44.31 ± 6.56	35.28 ± 7.30	45.48 ± 6.44	47.35 ± 9.47	43.85	10.7
NC	50.53 ± 1.11	2.50 ± 0.59	2.78 ± 0.00	27.26 ± 0.92	39.39 ± 4.41	24.49	6.4
NSC	8.68 ± 2.17	4.17 ± 1.13	5.28 ± 2.43	29.19 ± 2.21	15.10 ± 1.72	12.48	5.4
PER	28.42 ± 3.88	11.39 ± 2.60	5.83 ± 2.05	26.77 ± 3.58	16.33 ± 2.89	17.75	6.9
ROMMA	20.53 ± 7.21	5.83 ± 2.68	6.39 ± 2.64	26.77 ± 3.15	17.96 ± 3.94	15.50	6.8
ALMA	10.53 ± 0.00	2.22 ± 0.97	2.78 ± 0.00	19.03 ± 5.20	9.59 ± 1.38	8.83	1.9
PA	9.74 ± 2.50	2.64 ± 0.44	2.78 ± 0.00	19.52 ± 3.68	10.41 ± 1.79	9.02	2.8
SVM	14.74 ± 2.83	2.64 ± 0.44	2.78 ± 0.00	16.13 ± 3.88	10.61 ± 2.32	9.38	3.2
LIK	13.95 ± 2.79	7.64 ± 1.35	3.06 ± 0.88	24.52 ± 4.08	17.96 ± 4.49	13.43	5.5
LESS	41.58 ± 4.77	2.50 ± 0.59	4.72 ± 1.87	28.23 ± 2.97	30.00 ± 6.24	21.41	6.5

average ranks were achieved by PA (4.41), ROMMA (4.41), SVM (4.60), and ALMA (4.62). We found significant differences between PA and SVM (Wilcoxon rank sum test: $p = 0.0028$) and ROMMA and SVM (Wilcoxon rank sum test: $p = 0.0006$). We also found a significant difference between SVM, PA, ROMMA and all classifiers with dimensionality reduction NSC, LESS, and LIKNON (9 Wilkoxon rank tests, all $p <$ 0.000014 after Holm correction for multiple testing).

Cross-validation results on real datasets. The results of the cross–validation experiments can be seen in Table 5. The LDA variants show high error rates for all datasets. Compared to the other centroid based classifiers the NSC has better error rates on the datasets Bittner and West. On these datasets the improvement is better than 10%. Among the classifiers, which try to maximize the margin, ALMA has best error rates on the datasets Bittner, Golub and West. The SVM achieves equal or better result on the other datasets. The results of PA are comparable to the results of SVM and ALMA.

6 Conclusion

In this study a set of eleven linear classifiers were compared in terms of noise robustness and classification rates. The classifiers were tested on real and artificial high dimensional datasets. The artificial datasets fulfilled the model assumptions of the classifiers LDA and NC. Within these tests, a small number of undirected noisy datapoints lead to rapid increasing error rates. NC-like algorithms are more influenced by this effect than the LDA-like ones. This can be seen in the *sd* breakdown experiments and the single outlier experiments. If the noise is directed, the NC-like algorithms perform different to the LDA-like algorithms. In the *class* breakdown experiment, 50% of all datapoints could be replaced by noisy ones. The LDA-based classifiers show a mean error rate of about 50% in this experiment. The centroid based classifiers are superior to the others in the early stages of the *forward* single outlier test. Effects of the noise variant chosen in the *mean* breakdown experiment could only be seen for high values of x. In this case, all NC-based classifiers were more robust for higher noise rates. Only the LDA-based algorithms were highly affected of this kind of noise. This is not too surprising and supports the findings of Raudys & Duin [22], that when the total number of learning samples approaches the dimensionality some of the eigenvalues of the sample covariance matrix become extremely large while the others become extremely small. This negatively affects classifier performance. The other classifiers are more robust in the *sd* breakdown experiments. Especially the large margin classifiers are unaffected by this scenario. These classifiers are more sensitive to direct noise. This can be seen by their linear increasing error rates in the *class* breakdown.

The effects of feature selection were different for single algorithms. The NSC obtained equal or better results than NC in the *sd* breakdown experiments and in the *class* breakdown experiments for higher noise rates. LESS is more comparable to NC, but becomes more instable in the *sd* breakdown experiments. LIKNON has lower error rates than the SVM in the *mean* breakdown experiments. The top mean rank over all artificial experiments was gained by PA, ROMMA, ALMA and SVM. Surprisingly, these algorithms also outperformed statistically significant all classifiers with dimensionality reduction (NSC, LESS, LIKNON). This might be due to the problem of finding a meaningful subset of features in these very high-dimensional spaces of low cardinality [23]. Also as volume of the feature space increases exponentially with dimensionality, noise on each of the coordinates does not affect the location of the datapoint too much and thus margin classifiers seem to be superior in this setting. This is also supported by the good performance of the single outlier experiments in which the pure feature selection algorithm NSC scored worse than margin algorithms.

On the real datasets the large margin classifiers were slightly better than the centroid based classifiers. There were two examples among the used datasets (Bittner and West), which could hardly be classified by NC and LESS. The error curves of the NSC has comparable results to the other classifiers on this datasets. Concerning the error rates over all real datasets, the top three classifiers are ALMA, PA and SVM. The overall best performance for these types of high-dimensional data of low cardinality is given be PA, as it scores top on the artificial data and second on the expression profiles.

Acknowledgments. This work is supported by the Stifterverband für die Deutsche Wissenschaft (HAK) and the German Science Foundation, SFB 518, Project C05.

References

1. Breiman, L., Friedman, J., Stone, C., Olshen, R.: Classification and Regression Trees. Chapman & Hall/CRC (1984)
2. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
3. Rojas, R.: Neural Networks: A Systematic Introduction. Springer, Heidelberg (1996)

4. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, New York (2001)
5. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine 2(6), 559–572 (1901)
6. Ans, B., Hérault, J., Jutten, C.: Adaptive neural architectures: Detection of primitives. In: Proceedings of COGNITIVA 1985, pp. 593–597 (1985)
7. Lu, J., Plataniotis, K., Venetsanopoulos, A.: Face recognition using LDA-based algorithms. IEEE Transactions on Neural Networks 14(1), 195–200 (2003)
8. Zolnay, A., Kocharov, D., Schlüter, R., Ney, H.: Using multiple acoustic feature sets for speech recognition. Speech Commun. 49(6), 514–525 (2007)
9. Buchholz, M., Kestler, H.A., Bauer, A., et al.: Specialized DNA arrays for the differentiation of pancreatic tumors. Clinical Cancer Research 11(22), 8048–8054 (2005)
10. Cover, T.M.: Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. IEEE Transactions on Electronic Computers 14(3), 326–334 (1965)
11. Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Diagnosis of multiple cancer types by shrunken centroids of gene expression. PNAS 99(10), 6567–6572 (2002)
12. Bhattacharyya, C., Grate, L.R., Rizki, A., et al.: Simultaneous classification and relevant feature identification in high-dimensional spaces: application to molecular profiling data. Signal Process. 83(4), 729–743 (2003)
13. Veenman, C.J., Tax, D.M.: Less: A model-based classifier for sparse subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(9), 1496–1500 (2005)
14. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psych. Rev. 65(6), 386–407 (1958)
15. Gentile, C.: A new approximate maximal margin classification algorithm. Journal of Machine Learning Research 2 (2001)
16. Li, Y., Long, P.M.: The Relaxed Online Maximum Margin Algorithm. Machine Learning 46(1-3), 361–387 (2002)
17. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online Passive-Aggressive Algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
18. Bittner, M., Meltzer, P., Chen, Y., et al.: Molecular classification of cutaneous malignant melanoma by gene expression profiling. Nature 406(6795), 536–540 (2000)
19. Golub, T., Slonim, D., Tamayo, P., et al.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 286(5439), 531–537 (1999)
20. Notterman, D., Alon, U., Sierk, A., Levine, A.: Transcriptional Gene Expression Profiles of Colorectal Adenoma, Adenocarcinoma, and Normal Tissue Examined by Oligonucleotide Arrays. Cancer Research 61(7), 3124–3130 (2001)
21. West, M., Blanchette, C., Dressman, H., et al.: Predicting the clinical status of human breast cancer by using gene expression profiles. PNAS 98(20), 11462–11467 (2001)
22. Raudys, S., Duin, R.: Expected classification error of the fisher linear classifier with pseudo-inverse covariance matrix. Pattern Recognition Letters 19(5), 385–392 (1998)
23. Dougherty, E.R.: Feature-selection overfitting with small-sample classifier design. IEEE Intelligent Systems 20(6), 64–66 (2005)