Outlier Detection Using Ball Descriptions with Adjustable Metric

David M.J. Tax[1], Piotr Juszczak[1], Elżbieta Pękalska[2], and Robert P.W. Duin[1]

[1] Information and Communication Theory Group
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
[2] School of Computer Science, University of Manchester
Manchester M13 9PL, United Kingdom
D.M.J.Tax@tudelft.nl

Abstract. Sometimes novel or outlier data has to be detected. The outliers may indicate some interesting rare event, or they should be disregarded because they cannot be reliably processed further. In the ideal case that the objects are represented by very good features, the genuine data forms a compact cluster and a good outlier measure is the distance to the cluster center. This paper proposes three new formulations to find a good cluster center together with an optimized ℓ_p-distance measure. Experiments show that for some real world datasets very good classification results are obtained and that, more specifically, the ℓ_1-distance is particularly suited for datasets containing discrete feature values.

Keywords: one-class classification, outlier detection, robustness, ℓ_p-ball.

1 Introduction

In this paper we consider a special classification problem in which one of the classes is sampled well, but in which the other class cannot be sampled reliably [1,2]. An example is machine condition monitoring, where failure of a machine should be detected. It is possible to sample from all normal operation conditions (called the *target* class), but to sample from the failure class (the *outlier* class) is very hard. Furthermore it is also very expensive. Therefore a classifier should be constructed that mainly relies on examples of healthy machines and that can cope with a poorly sampled class of failing machines.

In the most ideal case the target class forms a tight, spherical cluster and all outliers are scattered around this cluster. To identify outliers one has to measure the distance from an object to the cluster center and threshold this distance. Clearly, when the threshold on the distance (or radius of the ball) is increased, the error on the target class decreases but at the cost of the outlier data that is accepted. The optimal ball has a minimum volume while it still encloses a large fraction of the target data.

According to the central limit theorem the target class has a Gaussian distribution when the target objects are noisy instantiations of one prototype disturbed by a large number of small noise contributions. The Mahalanobis distance

D.-Y. Yeung et al. (Eds.): SSPR&SPR 2006, LNCS 4109, pp. 587–595, 2006.

to the cluster center has to be used to detect outliers. But one should take care that a robust estimate of the class shape is used, because outliers in the training set severely deteriorate the maximum likelihood estimates for the Gaussian distribution [3]. The Minimum Determinant Covariance estimator is a practical implementation of a robust mean and covariance estimator [4].

When the assumption of many small noise contributions does not hold, other distance measures can be used. One flexible parameterization of a distance is the ℓ_p-distance. This distance has one free parameter p that rescales distances non-linearly along individual axis before adding the contributions to the final distance. Thresholding this distance defines a ℓ_p-ball as the decision boundary around the target class. The advantage of the ball description is that only few parameters have to be fitted to get a good description of the target class. This is particularly useful when the outlier detector is applied in high dimensional feature spaces and with small training set sizes. A second advantage is that it is possible to compute the volume captured by the ball analytically (see for instance, [5] pg. 11). This allows for an estimate of the error on the outlier class [6] and therefore for model evaluation between outlier detection methods.

In this paper we propose the use of the ℓ_p-distance measure to a center for the description of a class, resulting in a ball-shaped decision boundary. Three models are formulated in section 2. In the first formulation the volume of the ℓ_p distance ball is minimized by weighing the features, while the parameter p and the center of the ball are fixed. In the second we fix the p and the weights of the features, but optimize the center to minimize the volume. In the last formulation we optimize both the center as the p. In section 3 the methods are compared on real world datasets and we end with a conclusion in 4.

2 Theory

We start with a training set $\mathcal{X}^{tr} = \{\mathbf{x}_i, i = 1, ..., l\}$ containing l target objects, represented in an n dimensional feature space: $\mathbf{x} \in \mathbb{R}^n$. This dataset may contain some outliers, but they are not labeled as such. The ℓ_p-distance is defined as:

$$\|\mathbf{x} - \mathbf{z}\|_p = \sqrt[p]{\sum_{j=1}^{n} |x_j - z_j|^p}, \quad p > 0. \tag{1}$$

To detect outliers with respect to the training set $\mathcal{X}^{tr}$, we threshold the distance to some center $\boldsymbol{a}$. This defines the classifier f_p:

$$f_p(\mathbf{x}; \boldsymbol{a}) = \begin{cases} \text{target} & \|\mathbf{x} - \boldsymbol{a}\|_p^p \leq w_0, \\ \text{outlier} & \text{otherwise.} \end{cases} \tag{2}$$

A well performing classifier f_p minimizes both the error on the target class (i.e. the ball encloses almost all the target objects) as the error on the outlier class (i.e. the ball covers a minimum volume in the feature space). By a suitable

placing of $\boldsymbol{a}$, by minimizing the threshold (or radius) w_0, weighting of features and optimizing p the two errors are minimized. In the next three sections we propose three formulations to optimize ℓ_p-balls.

2.1 w-Ball: The Weighted-Feature ℓ_p-Ball

In the first formulation, the feature axis are weighted such that the ball has minimum radius w_0. The center $\boldsymbol{a}$ and the parameter p are fixed beforehand. The w_0 is minimized by varying the weight w_j on each individual feature. To avoid the trivial solution of zero weights for all the features, the sum of the weights is fixed to one and all zero-variance directions are removed. [1] To make the solution less sensitive to outliers in the training data, the constraints are weakened by introducing slack variables ξ_i:

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \; w_0 + C\sum_{i=1}^{l} \xi_i \tag{3a}$$

$$\text{s.t.} \; \sum_j w_j |x_{ij} - a_j|^p \leq w_0 + \xi_i, \quad \xi_i \geq 0 \quad \forall i \tag{3b}$$

$$\sum_j w_j = 1, \quad w_j \geq 0, \quad \forall j \tag{3c}$$

A reweighted ℓ_p-distance is used for the evaluation of a new object. That means that each term in the sum in equation (1) is multiplied by w_j. This formulation is called the 'weighted-features' ℓ_p-ball, or w-ball.

In the experiments center $\boldsymbol{a}$ is set to the mean vector of dataset $\mathcal{X}^{tr}$. This formulation is a linear programming problem that can be solved efficiently using standard optimization toolboxes, even for high dimensional feature spaces. Parameter C determines the tradeoff between w_0 and ξ_i. A large C indicates that ξ_i should remain small in comparison to w_0 (see (3a)), resulting in a very large ball. When C is small, the slack ξ_i is allowed to grow and the radius w_0 stays reasonably small. In practice the w-ball is still not robust against outliers [7]. This is caused by the fact that an outlier influences the location $\boldsymbol{a}$ of the ball. Varying C has just a minor effect on the final solution. To get a robust ball description, the center of the ball has to be optimized such that outliers do not have any influence on the solution, even when they are located far away. This is achieved with a formulation given in the next section.

In the left subplot of figure 1 the decision boundaries for the w-ball are shown for $p = 1, 2$ and 6. The optimization reweighs the features such that the balls fit the data best. Depending on p, the shape becomes more diamond-like ($p = 1$) or more box-like ($p = 6$). The two objects on the far right still influence the solution, although they are outside the decision boundary. When the outliers on the right side are moved much further to the right, the weight for this feature is decreased (to satisfy constraint (3b)). When this weight w_1 decreased to zero,

[1] When the k-th feature does not show a variance, the optimal solution is $w_k = 1$ and all other $w_i = 0, i \neq k$.

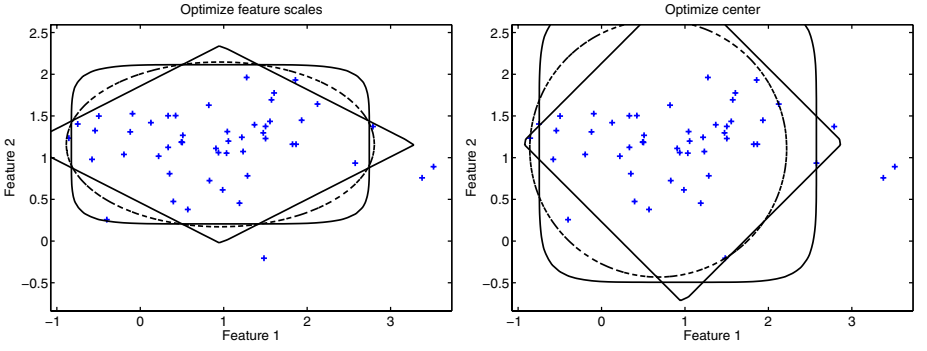

Fig. 1. The decision boundaries of the w-ball (left) and the c-ball (right) on the same dataset, and varying $p, p = 1, 2, 6$. The diamond-shaped boundary is obtained for $p = 1$. For increasing p the boundary becomes more square. For the w-ball $C = 10$ and for the c-ball $f = 0.9$ (see text for explanation of f).

the ball degenerates to a 'strip', effectively performing a feature reduction by removing this feature from the solution.

2.2 c-Ball: The ℓ_p-Ball with Variable Center

For a robust formulation a quantile function is defined. Denote $\tilde{\boldsymbol{y}} = (y_{(1)}, y_{(2)}, ..., y_{(l)})$ the sorted version of $\boldsymbol{y}$, with $y_{(1)} < y_{(2)} < ... < y_{(l)}$. The quantile function is defined as:

$$\mathcal{Q}_f(\boldsymbol{y}) = y_{([fl])}, \tag{4}$$

where $[c]$ returns the nearest integer value of c. Thus, $\mathcal{Q}_0(\boldsymbol{y})$ is the minimum element of $\boldsymbol{y}$, $\mathcal{Q}_1(\boldsymbol{y})$ the maximum and $\mathcal{Q}_{0.5}(\boldsymbol{y})$ the median.

The center $\boldsymbol{a}$ is optimized such that the object furthest away is as near as possible to this center. To be robust against outlier objects in the training set, we only consider a fraction f of the objects. When we define $y_i = \sum_j |x_{ij} - a_j|^p$ the following optimization problem can be formulated:

$$\min_{\boldsymbol{a}} \mathcal{Q}_f((y_1, y_2, ..., y_l)) \tag{5}$$

This formulation is called the the 'centered' ℓ_p-ball, or c-ball. Due to the very non-linear quantile function this optimization cannot be solved very efficiently. In this paper we use a general purpose multivariate non-linear optimizer (based on the Nelder-Mead minimization[8]). It should be noted that this optimization becomes very slow for high dimensional feature spaces (say, $n > 100$). In these cases a standard gradient descent method is applied [2]. On the other hand, the

[2] Note that the gradient and the Hessian of (5) is very simple to compute when the f% quantile $y_{([fl])}$ has been found. Define $k = [fl]$, then the gradient becomes $\frac{\partial y_k}{\partial a_j} = p \cdot \text{sign}(a_j - x_{kj})|a_j - x_{kj}|^{p-1}$ and the Hessian $\frac{\partial^2 y_k}{\partial^2 a_j} = p(p-1)\text{sign}(a_j - x_{kj})|a_j - x_{kj}|^{p-2}$ and $\frac{\partial^2 y_k}{\partial a_i \partial a_j} = 0$ for $i \neq j$.

solution is insensitive to the most remote $(1-f)\times 100\%$ of the data, making it an estimator with a breakdown value of $\lfloor (1-f)l \rfloor$ [9].

In the right subplot of figure 1 the decision boundaries for the c-ball's are shown for $p=1, p=2$ and $p=6$. The models do not take the difference in variance of the different features into account, resulting in a wider data description than the w-ball. On the other hand, the c-ball is robust against the outlier objects on the right side (the centers are optimized to reject 10% of the data, i.e. $f=0.9$). Moving these objects even further away will not change the solution as it is shown in the figure. Also notice that the locations of the centers of the balls vary, depending on the p.

2.3 p-Ball: The ℓ_p-Ball with Variable Center and Metric p

In the last formulation also the p in the ℓ_p-distance is optimized, together with the ball center, while fixing the weight per feature. Because p changes, the metric changes and it is not possible to compare solutions in different spaces by just comparing the radii of the balls. To compare balls in different spaces, the volumes of the balls have to be compared. The unit ball of ℓ_p is defined as $B_p^n = \{\mathbf{x} \in \mathbb{R}^n; \|\mathbf{x}\|_p \leq 1\}$. The volume of the unit ball is given by [5]:

$$\mathrm{vol}(B_p^n) = \frac{(2\Gamma(1+1/p))^n}{\Gamma(1+n/p)} \tag{6}$$

The volume of a ball with radius r is $\mathrm{vol}(B_p^n)r^n$. Using this, the following optimization problem can now be formulated:

$$\min_{\boldsymbol{a},p,r} \ \mathrm{vol}(B_p^n)r^n \tag{7a}$$

$$\text{s.t.} \ \ \mathcal{Q}_f((\|\mathbf{x}_i - \boldsymbol{a}\|_p^n)) \leq r^n, \quad p>0 \tag{7b}$$

where r is the ball radius. This is called the p-optimized ℓ_p-ball, or p-ball. Again, the optimization is made more robust by considering the f-fraction quantile.

Notice that in this formulation both the degree p and the center of the ball are optimized, resulting in an even more complex optimization problem. Again a general multivariate nonlinear optimizer has to be used [3]. To avoid problems with the constrained variable p ($p>0$), a variable substitution is applied and a new unconstrained variable $q=\log(p)$ is introduced. This makes it possible to use an unconstrained optimization procedure.

In figure 2 the decision boundaries for the p-ball are shown for the same data as used in figure 1. The fraction f is set to $f=0.9, f=0.8, f=0.5$. Both the location as the shape of the balls is adapted to capture 90%, 80% or 50% of the data. The resulting optimized values for p are $1.26, 2.21$ and 5.39 respectively. Objects outside the decision boundary are completely ignored in the minimization of the ball volume, and can therefore be randomly moved around without affecting the solution.

[3] Here also the gradient and Hessian can be computed, but this is considerably more complicated than in section 2.2.

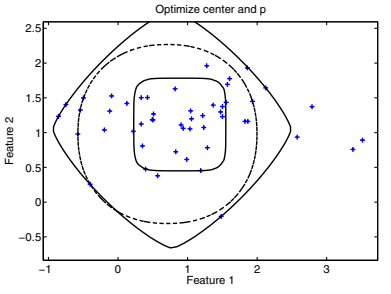

Fig. 2. The decision boundaries of the p-ball for different values of f, $f = 0.9, f = 0.8, f = 0.5$

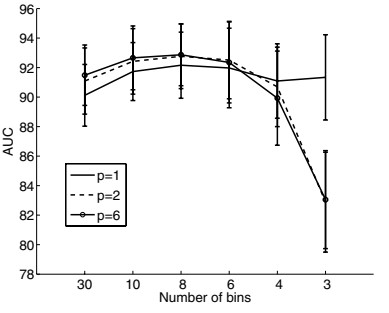

Fig. 3. The AUC performances on dataset 13 for a varying number of bins B in which the features are discretized

3 Experiments

The three methods, the w-ball, c-ball and p-ball, are compared with some standard classifiers on datasets, mainly taken from the UCI repository [10]. These datasets are standard multiclass problems and to convert them into an outlier detection problem, we use one of the classes as target class, and all other classes are considered outlier. Furthermore, we consider datasets for which the target class is reasonably clustered: it does not contain several clusters, and is not distributed in a subspace. The datasets are preprocessed to have unit variance for all features (where the scaling factors are obtained from the training set).

Table 1. Characteristics of the datasets: the number of objects in the target class and the outlier class, and the dimensionality of the data

nr	dataset	tar/out	dim.	nr	dataset	tar/out	dim.
1	Iris virginica	50/100	4	9	Concordia16 digit 3	400/3600	256
2	Breast malignant	458/241	9	10	Colon 2	40/22	1908
3	Breast benign	241/458	9	11	Thyriod normal	93/3679	21
4	Heart diseased	139/164	13	12	Waveform 1	300/600	21
5	Heart healthy	164/139	13	13	Pageblocks text	4913/560	10
6	Biomed diseased	67/127	5	14	Satellite, cotton crop	479/3956	36
7	Arrhythmia normal	237/183	278	15	Satellite, damp grey soil	415/4020	36
8	Ecoli periplasm	52/284	7				

In table 1 the list of datasets with their characteristics is shown. For the two-class Breast and Heart datasets, each of the two classes is used as the target class once. This is to show that for each class separately, different optimal solutions are found. On the datasets some standard classifiers are fitted. First a simple Gaussian distribution is applied, using the maximum likelihood estimates for the mean and covariance matrix. The second method uses the Minimum Covariance Determinant algorithm to estimate a robust covariance matrix [4]. The third method is the Parzen density estimator, that optimizes its width parameter

Table 2. AUC performances of the one-class classifiers on 15 real world datasets. The best performances (and the ones that are not significantly worse according to a t-test, at a 5% confidence level) are indicated in bold. The experiments are done using five times ten-fold stratified cross-validation. The standard deviations are given between brackets.

	datasets				
classifiers	1	2	3	4	5
Gauss	**97.8 (0.5)**	98.5 (0.1)	82.2 (0.2)	63.8 (0.7)	80.0 (0.7)
Min.Cov.Determinant	97.6 (0.2)	NaN (0.0)	73.5 (0.1)	66.7 (1.8)	NaN (0.0)
Parzen	96.8 (0.9)	99.1 (0.1)	68.1 (0.5)	65.6 (0.6)	79.3 (0.4)
k-center	96.0 (0.9)	98.4 (0.2)	72.6 (13.6)	67.3 (2.6)	79.3 (2.3)
Support vector DD	97.3 (0.4)	NaN (0.0)	69.8 (1.0)	64.4 (0.5)	78.4 (0.6)
w-ball $p = 1$	**98.3 (0.4)**	98.0 (0.1)	97.4 (0.2)	**78.9 (0.8)**	73.3 (1.7)
w-ball $p = 2$	**98.0 (0.5)**	97.7 (0.2)	**97.7 (0.1)**	71.3 (3.2)	45.9 (1.3)
w-ball $p = 6$	97.0 (0.4)	97.5 (0.4)	91.1 (0.5)	70.9 (2.7)	40.2 (6.5)
c-ball $p = 1$	96.4 (0.9)	**99.3 (0.1)**	97.5 (0.1)	77.4 (0.4)	**83.9 (0.6)**
c-ball $p = 2$	96.5 (0.5)	99.0 (0.1)	97.3 (0.1)	73.0 (0.3)	82.6 (0.9)
c-ball $p = 6$	96.0 (0.6)	98.5 (0.2)	91.6 (0.2)	63.3 (0.4)	79.5 (0.7)
p-ball	96.0 (0.6)	**99.3 (0.1)**	96.6 (0.3)	72.9 (0.8)	82.6 (0.7)
classifiers	6	7	8	9	10
Gauss	60.8 (0.8)	76.8 (0.4)	92.9 (0.3)	91.3 (0.0)	68.4 (3.6)
Min.Cov.Determinant	53.5 (1.2)	NaN (0.0)	NaN (0.0)	NaN (0.0)	NaN (0.0)
Parzen	48.3 (0.5)	77.3 (0.5)	92.9 (0.5)	92.4 (0.0)	**63.6 (22.4)**
k-center	46.9 (5.2)	77.8 (1.1)	87.0 (2.3)	91.0 (0.6)	68.1 (2.1)
Support vector DD	53.0 (2.1)	52.7 (9.4)	92.2 (1.0)	36.7 (0.5)	**63.6 (22.4)**
w-ball $p = 1$	**71.8 (1.2)**	70.4 (0.8)	91.6 (0.7)	84.4 (0.0)	57.1 (3.6)
w-ball $p = 2$	69.0 (1.1)	**80.9 (0.5)**	91.5 (0.5)	82.9 (0.0)	56.8 (3.0)
w-ball $p = 6$	62.3 (1.1)	70.3 (1.9)	90.1 (0.4)	65.0 (1.1)	56.2 (4.0)
w-ball $p = 1$	**72.7 (0.6)**	78.4 (0.4)	**95.3 (0.4)**	92.6 (0.0)	66.9 (2.1)
w-ball $p = 2$	67.9 (0.4)	78.2 (0.3)	94.6 (0.5)	90.5 (0.0)	71.1 (1.5)
w-ball $p = 6$	61.1 (1.0)	76.2 (0.3)	93.3 (0.4)	85.2 (0.2)	**77.2 (0.9)**
p-ball	66.0 (0.5)	76.5 (0.4)	93.3 (0.4)	**92.6 (0.0)**	70.2 (1.1)
classifiers	11	12	13	14	15
Gauss	84.3 (0.0)	89.9 (0.0)	59.9 (5.9)	88.0 (0.0)	83.0 (0.0)
Min.Cov.Determinant	NaN (0.0)	89.9 (0.0)	93.5 (0.0)	89.6 (0.2)	78.6 (0.1)
Parzen	90.6 (0.0)	90.0 (0.0)	50.6 (5.1)	99.0 (0.0)	39.9 (0.0)
k-center	53.3 (3.0)	87.8 (1.8)	55.9 (3.7)	97.5 (1.5)	85.0 (1.5)
Support vector DD	56.0 (0.0)	41.7 (0.0)	50.1 (5.6)	37.6 (0.0)	21.1 (0.0)
w-ball $p = 1$	96.9 (0.0)	91.2 (0.0)	91.7 (0.1)	**99.1 (0.0)**	91.2 (0.0)
w-ball $p = 2$	99.0 (0.0)	91.6 (0.0)	91.8 (0.1)	98.8 (0.0)	92.3 (0.0)
w-ball $p = 6$	**99.1 (0.0)**	90.5 (0.0)	91.0 (0.1)	98.4 (0.0)	92.4 (0.0)
c-ball $p = 1$	93.1 (0.0)	92.1 (0.0)	92.2 (0.0)	98.7 (0.0)	92.6 (0.0)
c-ball $p = 2$	88.5 (0.0)	93.0 (0.0)	93.0 (0.0)	98.5 (0.0)	**92.7 (0.0)**
c-ball $p = 6$	83.4 (0.0)	91.6 (0.0)	**93.8 (0.0)**	96.9 (0.0)	91.4 (0.0)
p-ball	93.6 (0.0)	**93.0 (0.0)**	93.0 (0.1)	98.5 (0.0)	92.6 (0.0)

using leave-one-out on the training set [11]. The fourth method uses the k-centroid method that places several centers and minimizes the largest distance from any training object to its nearest center. Finally, the support vector data description [12] is used, that is fitting a ball in a Gaussian kernel space. The features are rescaled to unit variance, and therefore the width parameter σ in the RBF kernel was fixed to $\sigma = 1$ which gave acceptable results in most cases.

These standard methods are compared to the w-ball, c-ball and p-ball with varying values for p (when applicable). Five times ten-fold stratified cross-validation is applied, and the average Area Under the ROC curve [13] is reported. The results are shown in table 2. In some cases the classifier could not

be trained (for instance, the minimum covariance determinant classifier has a constraint that it cannot be estimated on datasets with more than 50 features). For these cases NaN outputs are shown.

The first observation that can be made is that for many datasets, datasets 1, 2, 4, 5, 6 and 8, the ℓ_1-metric outperforms all the others, even when a different formulation is used (i.e. c-ball instead of w-ball). It appears that all these classes have discrete features, suggesting that the ℓ_1 (city-block) distance is indeed very suited for the description of discrete data. This is tested by discretizing the features of dataset 15 (where the w-ball performs better for higher p) and training an w-ball with $p = 1, 2, 6$. The AUC performances are shown in figure 3. It shows that by reducing the number of bins, the relative performance of the ℓ_1 metric improves while that of the ℓ_2 and ℓ_6 significantly decreases.

Secondly, for high dimensional data, like datasets 7, 10 and 11, the ball-shaped models appear to be simple enough (and therefore stable enough) to outperform more complex models. Often the performance is not that significantly better than that, for instance, of the Gaussian model, but in some cases it can be significant (see dataset 11). The difference in performance between w-ball and c-ball can often be traced to the number of outliers (or the noise) present in the training set. When the ball center can be represented well by the mean of the training set, like in datasets 1, 3, 4, 7, and 14, the w-ball is to be preferred. In other datasets, like 2 and 8, the target class shows a long tail with remote outliers, shifting the mean of the target class out of the main cluster. In these cases the more robust center estimate has to be used.

Finally, the most flexible approach, the p-ball, rarely shows the very best performance, models with a fixed p perform on average slightly better. The p-ball slightly overfits, but fortunately, the optimized value for p is always close to the p of the best performing ball. Clearly, a validation set has to be used to finally judge the best value for p in the w-ball or c-ball. When this validation data is not available, the p-ball is to be preferred.

4 Conclusions

For many outlier detection problems for which the target data is characterized by good features, outliers can be detected well by measuring the distance to a suitable cluster center and thresholding this distance. This paper proposes three new approaches to optimize the cluster center and the distance measure, such that the genuine data is described well by an ℓ_p ball. In the first formulation the feature weights are optimized, by solving a linear programming problem. The second formulation optimizes the cluster center in a robust gradient descent approach. In the last formulation not only the center but also the parameter p is optimized, using a general multivariate nonlinear optimizer.

The results on real world data show that datasets with discretized feature values benefit from the use of the ℓ_1 metric. On the other hand, the optimization of the p in the ℓ_p metric appears to be sensitive to overtraining. When one considers relatively outlier-free data, it is advantageous to fix the center of ℓ_p ball

and optimize the scaling of the features. When significant outliers are present, or the target class distribution is significantly asymmetric, the ℓ_p ball has to optimized using a robust procedure.

Obviously, the single ball solution can be extended to a *set* of balls by using the standard k-means clustering algorithm. In k-means clustering often the Euclidean distance to cluster prototypes is used. This can be replaced by the ℓ_p-distance to cluster centers resulting in a generalized Lloyds algorithm [14]. The cluster centers, the feature weights and possibly the p can be optimized using one of the three ball fitting approaches as they are presented in this paper.

Acknowledgments. This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministry of Economic Affairs.

References

1. Tax, D.: One-class classification. PhD thesis, Delft University of Technology, http://ict.ewi.tudelft.nl/~davidt/thesis.pdf (2001)
2. Koch, M., Moya, M., Hostetler, L., Fogler, R.: Cueing, feature discovery and one-class learning for synthetic aperture radar automatic target recognition. Neural Networks **8**(7/8) (1995) 1081–1102
3. Huber, P.: Robust statistics: a review. Ann. Statist. **43** (1972) 1041
4. Rousseeuw, P., Van Driessen, K.: A fast algorithm for the minimum covariance determinant estimator. Technometrics **41** (1999) 212–223
5. Pisier, G.: The volume of convex bodies and Banach space geometry. Cambridge University Press (1989)
6. Tax, D., Duin, R.: Uniform object generation for optimizing one-class classifiers. Journal for Machine Learning Research (2001) 155–173
7. Barnett, V., Lewis, T.: Outliers in statistical data. 2nd edn. Wiley series in probability and mathematical statistics. John Wiley & Sons Ltd. (1978)
8. Nelder, J., Mead, R.: A simplex method for function minimization. Computer journal **7**(4) (1965) 308–311
9. He, X., Simpson, D., Portnoy, S.: Breakdown robustness of tests. Journal of the American Statistical Association **85**(40) (1990) 446–452
10. Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
11. Duin, R.: On the choice of the smoothing parameters for Parzen estimators of probability density functions. IEEE Transactions on Computers **C-25**(11) (1976) 1175–1179
12. Tax, D., Duin, R.: Support vector data description. Machine Learning **54**(1) (2004) 45–66
13. Bradley, A.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30**(7) (1997) 1145–1159
14. Lloyd, S.: Least squares quantization in PCM. IEEE Transactions on Information Theory **28**(2) (1982) 129–137