Characterizing one-class datasets

David M.J. Tax, Robert P.W. Duin

Information and Communication Theory Group
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
d.m.j.tax@ewi.tudelft.nl

Abstract

This paper aims at characterizing classification problems to find the main features that determine the differences in performance by different classifiers. It is known that, using the disagreements between the classifiers, a distance measure between datasets can be defined. The datasets can then be embedded and visualized in a 2-D scatterplot. This embedding thus reveals the structure of the set of problems. In this paper we focus on a specific pattern recognition problem, the problem of outlier detection or one-class classification, where classifiers have to detect if a new object resembles the training data or not. For this problem the outputs of many classifiers on many datasets are available. By inspecting the scatterplot of the datasets, two main features appear to characteristize the datasets; (1) their effective sample size and (2) the class overlap. By generating artificial datasets for which these variables are varied, these observations are confirmed experimentally.

1. Introduction

In pattern recognition we try to solve classification problems by using classifier models that are fitted to training data. All classifiers have a particular bias that make them suitable for specific datasets, and less for others. In practice we are forced to apply all the classifiers from our limited toolbox to find the best one. Except for artificial data we are never certain which classifier will perform best on a specific dataset. It is therefore not only of academic interest to find out what are the main characteristics in datasets which causes the classifiers to perform differently. These characteristics may point to specific approaches to solve an classification problem.

Many attempts has been made to characterize datasets using simple measures to predict which classifier works well [9, 12]. Indeed some conclusions concerning the domains of competence for some classifiers, were drawn. But the main conclusion was that real world datasets "reveal intricate relationships among the factors affecting the difficulty of the problem". The problem is far from solved.

In this paper we approach the problem from the other side. We start with a large set of classifiers and a large set of real world datasets and we try to find the structure of the datasets by comparing the output labels of the classifiers [1]. For this we use the classifier disagreements, indicating how often classifiers disagree [7]. The structure might point to the important characteristics of datasets, thus suggesting features on which classifiers can specialize.

[1]Most of the classifiers and datasets are also discussed in an overview paper on one-class classification that is submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

The results are given for a special type of classification problem, the one-class classification or the novelty detection problem [17, 10]. This considers a two-class classification problems in which one of the classes cannot be sampled reliably. This happens for instance when one tries to perform machine condition monitoring. Here a well operating machine should be distinguished from a machine that is breaking down. It is possible to sample from all normal operation conditions, but there are many different ways in which a machine can fail. Not only is it very hard to sample the space of all breaking machines, it is also very expensive. The ill-sampled class is called the *outlier* class, and this class should be distinguished from a well-sampled class which is called the *target* class.

In section 2 we first define and discuss the classifier disagreements. In section 3 we describe the classifiers and (very shortly, due to space constraints) the datasets that are used in this paper. In section 4 the results of the projected datasets is shown, together with an indication of the two main parameters characterizing the variation in the datasets. Extra experiments are performed to confirm that this indication is true.

2. The distance between datasets

Assume a training set $\mathcal{X}^{tr}$ containing N d-dimensional training objects $\mathbf{x}_i, i = 1...N$, $\mathbf{x} \in \mathbb{R}^d$. For the one-class classification problem, only training data of the target class are available, and therefore all labels are $+1$. A one-class classifier f consists of two parts. The first part is the proximity of an object to the target data, and the second part is a threshold function (with threshold θ) over this proximity to obtain a classification label. The definition of the proximity measure depends on the classifier. In general, the proximity measure can be constructed from a density estimation $\tilde{p}$

$$f(\mathbf{x}) = \mathbf{1}(\tilde{p}(\mathbf{x}) \geq \theta) \tag{1}$$

or from some distance to a model $\tilde{d}$:

$$f(\mathbf{x}) = \mathbf{1}(\tilde{d}(\mathbf{x}) \leq \theta) \tag{2}$$

where $\mathbf{1}(A)$ is the indicator function, returning 1 is A is true, and 0 otherwise. The threshold θ is determined by specifying the error on the target training data ε^t.

Assume that a classifier f_i is trained on the training set $\mathcal{X}^{tr}$ and it gives the output label l_{ik} after evaluating object $\mathbf{x}_k$ from an independent test set $\mathcal{X}$

$$l_{ik} = f_i(\mathbf{x}_k), \qquad \mathbf{x}_k \in \mathcal{X} \tag{3}$$

The disagreement between classifiers f_i and f_j is defined as:

$$D_{\mathcal{X}}(f_i, f_j) = \frac{1}{N} \sum_{k=1}^{N} \mathbf{1}\left(l_{ik} \neq l_{jk}\right) \tag{4}$$

Note that this forms an $C \times C$ disagreement matrix D, where C is the number of classifiers. Classifiers that perfectly agree have zero distance $D_{\mathcal{X}}(f_i, f_j) = 0$, while classifiers that always disagree have a maximal distance of $D_{\mathcal{X}}(f_i, f_j) = 1$.

Using these disagreements, we can define a distance measure between *datasets* $\mathcal{X}^m$ and $\mathcal{X}^n$, consisting of the average difference between the disagreements [7]:

$$G(\mathcal{X}^m, \mathcal{X}^n) = \frac{1}{C^2} \sum_{i,j}^{C} |D_{\mathcal{X}^m}(f_i, f_j) - D_{\mathcal{X}^n}(f_i, f_j)| \quad (5)$$

This forms an $M \times M$ distance matrix G between datasets, where M is the number of datasets under consideration. It therefore uses the agreement and disagreement pattern for defining the similarity between datasets. Given the distances between the datasets, they can be visualized in 2D by applying Multi-dimensional scaling [5]. This locates M points such that the distances between these points reproduces as well as possible the distances between the datasets [7]. A new dataset $\mathcal{X}$ can be mapped onto this projection by computing first the classifier disagreements (equation (4)), next the distances to the 'training' datasets $\mathcal{X}^n$ using (5), and finally finding a location such that these distances are preserved as well as possible. This procedure is not limited to 2D projections, although for visualization it is the most common approach.

3. Experimental setup

We train the one-class classifiers on the one-class dataset using 5 times 10-fold stratified crossvalidation. The threshold θ is set such that 10% of the training target data is classified as outlier, $\varepsilon^{\mathrm{t}} = 0.1$. The output labels generated by the classifiers are stored. When the classifiers fail to supply output labels, due to training/convergence problems or numerical problems, the corresponding term in equation (5) is disregarded (or equivalently, set to zero).

3.1. The one-class classifiers

All the classifiers used in this paper are defined in the Matlab toolbox `dd_tools` [18]. The following classifiers are defined:

Gaussian It models the target data with a unimodal Gaussian density, with the standard maximum likelihood estimates for the mean and covariance matrix:

$$\tilde{p}_G(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \quad (6)$$

For high dimensional datasets the covariance matrix is regularized: $\Sigma' = \Sigma + \lambda\mathcal{I}$, with $\lambda = 0.01$, and $\mathcal{I}$ is the identity matrix.

MCD Gaussian The standard Gaussian model lacks robustness; outliers in the training set can severely influence the Σ. Therefore a robust version of the Gaussian, the Minimum Covariance Determinant is used. It selects a subset of the data for which the determinent of the covariance matrix is minimal [15]. The current implementation works upto $p = 50$.

Mixture of Gaussians To make the unimodal Gaussian distribution more flexible, the Mixture of Gaussians is also used. The means and the covariance matrices are optimized using the standard Expectation-Maximization procedure [2]. In the experiments three clusters are used, with the same regularization for the individual covariance matrices.

Parzen The Parzen density estimator [13] is a mixture of, most often, Gaussian kernels centered on the individual training objects, but with a simplified covariance matrix: $\Sigma = h\mathcal{I}$. The width of the kernel h is found by optimizing the likelihood on the training set using a leave-one-out procedure [6].

Naive Parzen The Naive Parzen is a simplification of the Parzen density estimator, inspired by the Naive Bayes approach. A Parzen density is estimated in each feature dimension separately, and the probabilities are multiplied to give the final target probability.

1-**nearest neigbor** This method uses the distance to the first nearest neighbor in the training set as proximity measure. Although this method is sensitive to outliers in the training set, no hyper parameters have to be optimized.

k-**nearest neigbor** Here the kth nearest neighbor is used, where k is optimized using a leave-one-out density estimation on the training data [8].

AUC-optimized k-NN This is the k-nearest neighbor data description where k is determined by optimizing the Area under the ROC curve [4, 20].

nearest neigbor distance ratio This method is the same as the 1 nearest neighbor, but the distance is normalized by the distance of the nearest object to *its* nearest neighbor in the training set.

PCA The principal component analysis classifier assumes that the data is located in a linear subspace. It finds a lower dimensional subspace, spanned by the basis vectors $\mathbf{W}$. It uses the reconstruction error, the distance between the original object and the mapped object, as the proximity measure.

autoencoder neural network This is a neural network approach to learn a low dimensional non-linear representation of the data [1, 16]. A standard feedforward neural network is trained to reproduce the input patterns $\mathbf{x}$ at its output layer. One of their hidden layers contains a small number of hidden units which works like an information bottleneck. The difference between the input $\mathbf{x}$ and output $\mathbf{x}'$ defines the proximity.

Support Vector Data Description The SVDD is a geometry-based model that fits a sphere around the data with the minimum volume, by optimizing the sphere center. The standard Euclidean distance can be rewritten in terms of inner products, making the "kernel trick" possible [19]. The RBF kernel is used with a fixed width of $\sigma = 1$.

L_1 **ball** This is a simplified version of the SVDD, where the Euclidean distance is replaced by the L_1 norm. The center of the sphere is fixed to the mean of the dataset, but the original features are rescaled such that all training data falls within the sphere.

k-**centers** This is a variant of the k-means clustering algorithm, but here the cluster centers are restricted to be one of the training objects. The proximity is the distance to the nearest cluster center.

Minimax Probability Machine This is a linear classifier that is placed such that the probability that a target object falls on the incorrect side of the decision boundary is bounded by a user-supplied value ε^{t} [11]. This method can also be

phrased in terms of inner products, and therefore also the kernel trick can be applied. The RBF kernel with $\sigma = 1$ is used in this paper.

Linear Programming dissimilarity The LPDD is a linear classifier that operates on distances between new objects and training target objects [14]. Objects with large distances to the target data are likely outlier objects. The LPDD therefore aims to place the decision boundary as close as possible to the origin in the distance space.

3.2. The datasets

In total 101 datasets are considered, mainly taken from the UCI repository [3]. An overview of the dataset with the dimensionalities and sample sizes, together with the classification performance, can be found at `http://ict.ewi.tudelft.nl/~davidt/occ/`. When the dataset is a multiclass classification problem, each of the classes is designated target class once, and the other classes are used as outlier. Objects with missing values are removed. In table 1 a small subset of the datasets is given, together with

Table 1: A listing of a subset of the 101 datasets. These datasets are explicitly mentioned somewhere in this paper.

nr	Dataset name, target class	obj/dim.
501	Iris, setosa	50/4
504	Beast cancer Wisconsin, malignant	458/9
505	Beast cancer Wisconsin, benign	241/9
506	Heart Cleveland, disease present	139/13
507	Heart Cleveland, disease absent	164/13
511	Biomed, healthy	127/4
512	Biomed, ill	67/4
515	Arrythmia, abnormal	237/278
519	Ecoli	52/7
530-539	Concordia, digit 0-9	400/256
571	Colon 2	40/1908
572	Leukemia 1	25/3571
585	Glass 5	13/9
591	Liver 2	200/6
601-611	Vowel 0-10	48/10
617	Survival, < 5 years	81/3
620	Page blocks	4913/10

their training set size and their dimensionality. Notice that for some datasets two or more versions exist. In these cases a multiclass problem is split into several one-class classification problems by designating each individual class to the target class once. Notice that the sample size ranges from 13 to 4913, and the dimensionality from 3 to 3571.

4. Experiments

4.1. The two main directions in the projection

Applying Multi-dimensional Scaling on the averaged differences in disagreements (5), results in a 2D position for each dataset. In the left subplot of figure 1 all the datasets are shown. The numbers in the plot are the identifiers of the datasets. After inspection of the datasets it appears that the two main directions in the plot indicate the effective sample size, which is the ratio between the number of training objects over the dimensionality:

$$SS_{\mathcal{X}} = \frac{\text{\# training objects}}{\text{dimensionality}} \tag{7}$$

and the average AUC performance:

$$\text{perf}_{\mathcal{X}} = \frac{1}{M} \sum_{i=1}^{M} AUC(f_i, \mathcal{X}) \tag{8}$$

where $AUC(f_i, \mathcal{X})$ is the Area under the ROC curve [4] of classifier f_i on dataset $\mathcal{X}$.

In the right of figure 1 the average AUC performance and the sample size is plotted [2]. Indeed, the dataset on the bottom of the graph have far higher sample sizes (Liver dataset, nr 591, has 200 objects in 6D) than the dataset on the top (Leukemia dataset, nr 572, with 25 objects in 3571D). Furthermore, datasets on the (lower) right have a far lower average performance (Arrythmia, nr 515, has an average AUC of 0.33, worse than random!), while datasets in the upper right are very well separable (Concordia handwritten digit 0, nr 530, has an average AUC of more than 0.96). This gives the first indication that these are the main variables in the dataset differences.

In the next sections we manipulate these two features of some of the datasets to check if the directions suggested in the figures correspond to these features. The procedure is that first, for each of the manipulated datsets, the classifiers from section 3.1 are trained. The classifier disagreements are computed and the disagreement differences (5) are mapped into the 2D space.

4.2. Sample size

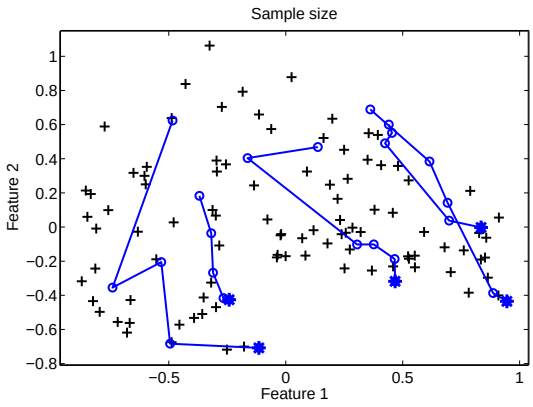

Figure 2: Traces of a few datasets for which the sample size is reduced.

To manipulate the sample size of a classification problem is simple: we start with a well sampled dataset, and reduce the number of training samples. Some results are shown in figure 2. From left to right five traces of datasets are shown. The traces start at the bottom with the star and move up via the circles. First trace on the left is the Liver dataset (nr 591, with 200, 100, 50, 25 and 15 objects in 6D), second the Biomed dataset (nr 511, with 127, 100, 60, 30 and 15 objects in 4D), next the Breast cancer Wisconsin (nr 504 with 458, 200, 100, 50, 25 and 15 objects in 9D), next the Vowel 1 dataset (nr 602 with 48, 24, 12, 6 and 3 objects in 10D) and in the extreme right corner the Vowel 2 dataset (nr 603, with also 48, 24, 12, 6 and 3 objects). Although the traces are a bit noisy (in particular for small sample sizes) they follow the of the reduced sample direction as it was suggested in figure 1.

[2] The values are rescaled for a clear visual presentation in gray scale.

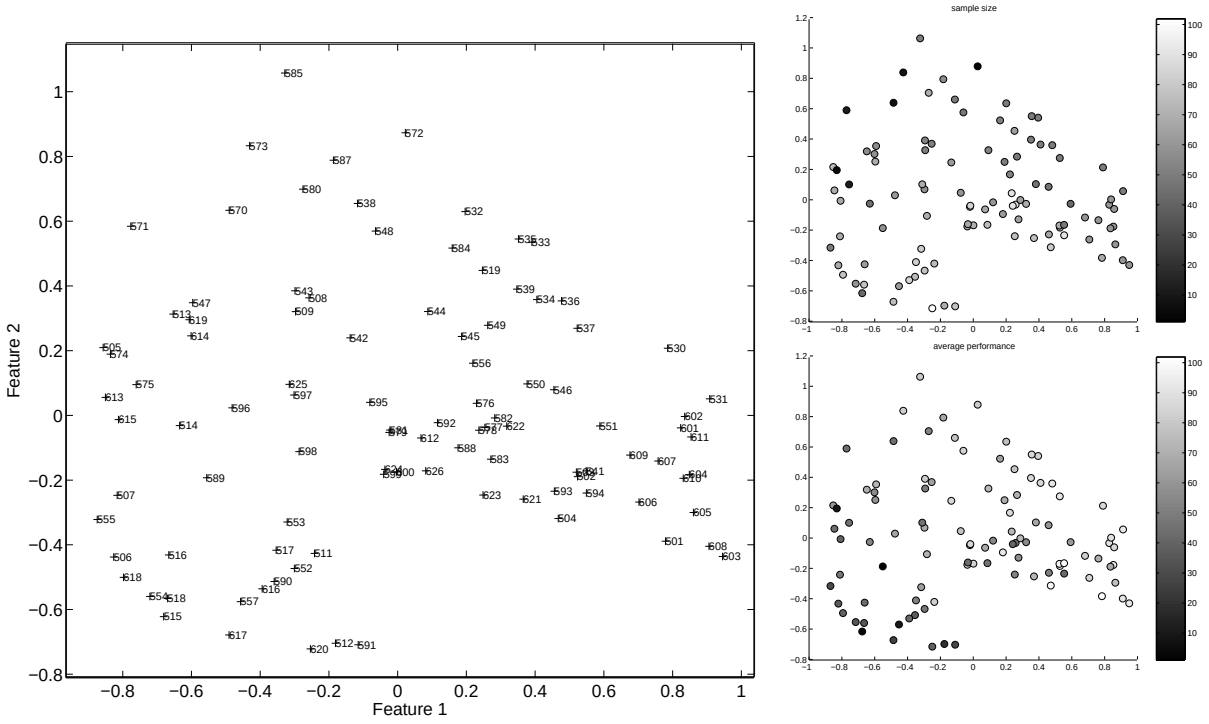

Figure 1: A 2D scatterplot of all the datasets based on the average disagreement between the classifiers. For all classifiers $\varepsilon^t = 0.1$. In the left plot the datasets are numbered according to `http://ict.ewi.tudelft.nl/davidt/occ/`. The right plots show (top) the effective sample size and (bottom) the average performance of these datasets. The values are rescaled to improve the visual presentation using a grey-level colour coding.

4.3. Class overlap

To check the interpretation of the class overlap direction is less straightforward. This requires that the distributions themselves are manipulated such that the overlap in the target and outlier class is varied. In this paper three possibilities are considered:

1. reduce the dimensionality of well separable classes. By reducing the dimensionality it is hoped that the separability decreases and the class overlap increases.
2. shift the means of the target and outlier class. We start with poorly separable datasets and computing the difference vector between the means of the two classes. The outlier class is now shifted by a few multiples of the difference vector. Notice that only the outlier data is changed, and it is not necessary to retrain the classifiers.
3. flip labels of the target and outlier data. By starting with very well separable data and randomly flipping labels from target to outlier, and vise versa, the class overlap is increased.

Next we discuss the results by the three methods.

In figure 3 the traces of dataset for which randomly features are removed. Again the traces start with the star. The top dataset is the Concordia handwritten digit 3 (nr 533, 400 objects in 256-, 200-, 150-, 100- and 50D, left next to it Ecoli (nr 519, 52 objects in 7-, 6-, 5-, 4-, and 2D), the dataset right of that is Concordia digit 0 (nr 530, 400 objects in 256-, 200-, 150-, 100- and 50D), the extreme right dataset is the Vowel 3 dataset (nr 604, 48 objects in 10-, 8-, 6- and 4D) and the bottom dataset is Breast cancer (nr 504, 458 objects in 9-, 7-, 5- and 3D). The traces are not completely "clean", and it appears that

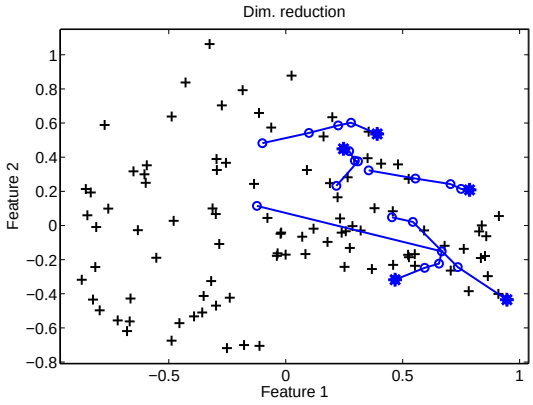

Figure 3: Traces of a few datasets for which features are randomly removed.

reducing the number of features has a more complicated effect on the dataset complexity. For some datasets class overlap decreases, while the sample size is not heavily affected (the Concordia datasets and the Vowel dataset). For these datasets most of the features seem to be informative, and removing them hurts the average performance. For the Breast and Ecoli dataset the class overlap actually seems to improve. Apparently, in these datasets many features are not very informative, and the classifies improve when features are randomly removed.

In figure 4 the traces of datasets are shown for which the class means are moved apart. From top to bottom we have the

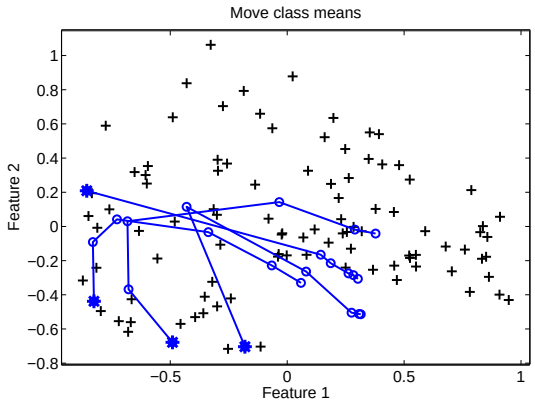

Figure 4: Traces of a few datasets for which randomly the means of the two classes are moved apart.

Breast cancer (target benign, nr505, 241 objects in 9D), Heart Cleveland (nr 506, 139 objects in 13D), Survival ($<$ 5 years, nr 617, 81 objects in 3D) and Biomed (nr 512, 67 objects in 4D). In all the cases the difference in the class mean is multiplied by 0.5, 1, 2, 3 and 4. The traces of these datasets are more consistent, but they fail to cover the complete range from very poor performance to very good performance. In other words, they never reach the far right end of the plot. Their curved trajectories actually suggest that the class overlap characteristic is not linear in this plot. It shows that by simplifying the classification problem by separating the two classes, the actual sample size increases. Less samples are required to make a good classifier, pushing the dataset not only in the direction of higher averaged accuracy, but also down, in the direction of higher sample sizes.

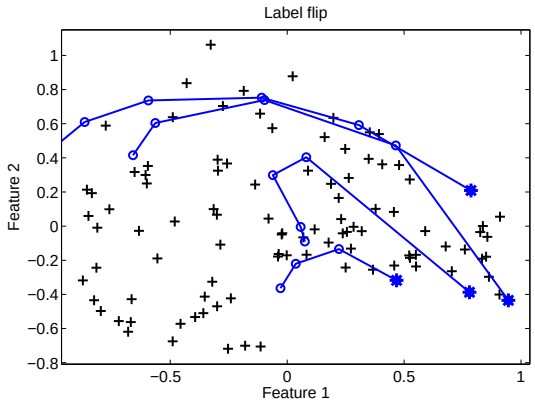

Figure 5: Traces of a few datasets for which randomly labels are flipped.

In figure 5 the traces of dataset are shown for which the class labels are flipped. For these experiments well-sampled dataset are used, therefore the traces start from datasets located at the right side of the graph. The dataset on the top tight is Concordia digit 0 (nr 530, 400 objects in 256D, flipping 25, 50, 100, 200 and 300 labels), on the bottom right the Vowel 2 dataset (nr 603, 48 objects in 10D, flipping 5, 10, 20 and 30 labels), next to the Vowel is the Iris Setosa dataset (501, 50 objects in 4D, also flipping 5, 10, 20 and 30 labels) and finally the Breast cancer Wisconsin (nr 504, 458 objects in 9D, flipping 50, 100, 200 and 300 labels). These datasets show a very clear tendency to move to the high class overlap area and indeed almost reach the left end of the graph. This very clearly suggests that the second high variance direction from right to left indicates the class overlap in the dataset. The curved traces here also indicate that this is a non-linear structure in this projection.

4.4. Individual classifier performances

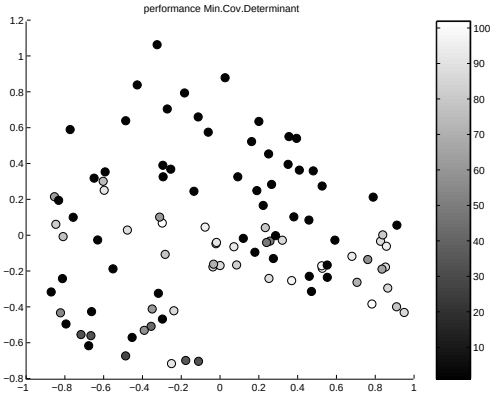

Figure 6: The AUC performance of the Minimum Covariance Determinant classifier encoded in grey scale.

It is now possible to investigate the the datasets for which each classifier performs well or not. Most classifier follow roughly the pattern as it is shown in the bottom right picture of figure 1, some classifiers have a more specific focus. In figure 6 the performance is shown for the Minimum covariance determinant classifier. Here a clear band of classifiers is classified well by this classifier. For higher dimensional datasets (mainly in the top of the figure) the procedure fails; the method is only implemented for $d < 50$. But also for datasets where the classes overlap or where the two classes are near and a complicated decision boundary is required (bottom left), the model performs poor.

4.5. The other variabilities in the data

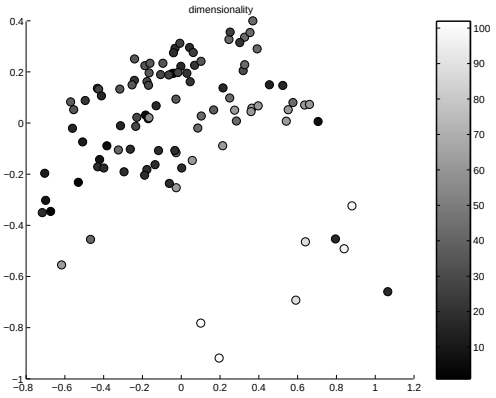

Figure 7: MDS scatterplot of the datasets using the second and third features, showing two clusters of datasets. The grey-level colour coding indicates the dimensionality, suggesting that the two clusters are the high and low dimensionality problems.

In figure 7 the second and the third dimensions of the MDS plot are shown. Here a clustering in the datasets is visible. The third dimension seems to encode data dimensionality, but it is not very clear (there are two outlier datasets in the lower cluster; the glass datasets with 13 and 29 objects in 9D). The lower right cluster contains the datasets in the very high dimensional spaces (dimensionalities larger than 1000), the other cluster contains the datasets upto 256D. The gap suggest that the set of datasets is not covering all dimensionalities, and that datasets with dimensionalities around 500-600 are lacking.

5. Conclusions

For the specific problem of one-class classification or novelty detection we investigated the main variables that determine the variability in the classification of objects by different classifiers. Using the classifier disagreements a similarity between datasets is defined allowing for the visualization of the datasets in a (2D) projection space using MDS. In this paper the outputs of 19 classifiers on 101 datasets are used. It appears that the effective sample size (the ratio beteen the number of objects and the dimensionality) and the average performance are the main variables that describe the variance in real world one-class datasets. Given these datasets, the scatterplot using the first two features shows a reasonably well sampled space; the classifiers almost uniformly fill the space.

This observation is verified and confirmed by varying the sample size and average performance of an artificial datasets and check where these datasets are mapped onto the 2D projection. The sample size direction can easily be confirmed, but to vary the class overlap is more complicated. Three approaches have been tried, moving the means of the datasets, randomly swapping the labels and reducing the dimensionality. All three approaches indeed change the class overlap, but it appears that it also influences the effective sample size, resuling in heavily curved trajectories in the projection.

These two main features of the one-class datasets suggest that one should develop a set of classifiers that cover the wide ranges of sample size and class overlap. First one can focus on classifiers that can exploit high sample sizes, or very low sample sizes. Second, one should construct classifiers that are capable of utilizing objects from the outlier class when the class overlap is not very large. When the class overlap is large, one has to focus on classifiers that are robust against outliers, or be sure that the training set does not contain outlier objects.

Further features become increasingly harder to interpret. This is probably caused by the fact that it is not clear what the main characteristics are, and are really not named yet. The third feature probably indicates the dimensionality of the datasets. When it is added, two clusters of datasets appear. This suggest that the sampling of the datasets is not sufficient in this direction and that datasets with dimensionality around 500 are lacking.

6. References

[1] P. Baldi and K. Hornik. Neural networks and principal component analysis: learning from examples without local minima. *Neural networks*, 2:53–58, 1989.

[2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Walton Street, Oxford OX2 6DP, 1995.

[3] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[4] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[5] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1995.

[6] R.P.W. Duin. On the choice of the smoothing parameters for Parzen estimators of probability density functions. *IEEE Transactions on Computers*, C-25(11):1175–1179, 1976.

[7] R.P.W. Duin, E. Pekalska, and D.M.J. Tax. The characterization of classification problems by classifier disagreements. In J. Kittler, M. Petrou, and M. Nixon, editors, *Proceedings 17th International Conference on Pattern Recognition (22-26 August 2004, Cambridge UK)*, volume 2, pages 140–143. IEEE Computer Society, Los Alamitos, CA, 2004.

[8] S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, and K.-R. Mueller. From outliers to prototypes: ordering data. *Neurocomputing*, 2005. To appear.

[9] T.K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002.

[10] M.W. Koch, M.M. Moya, L.D. Hostetler, and R.J. Fogler. Cueing, feature discovery and one-class learning for synthetic aperture radar automatic target recognition. *Neural Networks*, 8(7/8):1081–1102, 1995.

[11] G.R.G. Lanckriet, L. El Ghaoui, and M.I. Jordan. Robust novelty detection with single-class mpm. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press: Cambridge, MA, 2003. E,.

[12] E.B. Mansilla and T.K. Ho. On classifier domains of competence. In *Proc. of the 17th international conference on pattern recognition*, volume 1, pages 136–139, 2004.

[13] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathenatical Statistics*, 33:1065–1076, 1962.

[14] E. Pekalska, D.M.J. Tax, and R.P.W. Duin. One-class LP classifier for dissimilarity representations. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press: Cambridge, MA, 2003.

[15] P.J. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223, 1999.

[16] C. Surace, K. Worden, and G. Tomlinson. A novelty detection approach to diagnose damage in a cracked beam. In *Proceedings of SPIE*, pages 947 – 943, 1997.

[17] D.M.J. Tax. *One-class classification*. PhD thesis, Delft University of Technology, http://ict.ewi.tudelft.nl/˜davidt/thesis.pdf, June 2001.

[18] D.M.J. Tax. Dd_tools, data description toolbox for Matlab, Augustus 2005. version 1.3.0.

[19] D.M.J. Tax and R.P.W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[20] L. Yan, R. Dodier, M. C. Mozer, and R. Wolniewicz. Optimizing classifier performance via the Wilcoxon-Mann-Whitney statistic. In *In The Proceedings of the International Conference on Machine Learning (ICML)*, pages 848–855, 2003.