ELSEVIER

Pattern Recognition Letters 20 (1999) 1191–1199

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Support vector domain description

David M.J. Tax [*,1], Robert P.W. Duin

Pattern Recognition Group, Faculty of Applied Science, Delft University of Technology, F263 Lorentzweg 1, 2628 CJ Delft, The Netherlands

Abstract

This paper shows the use of a data domain description method, inspired by the support vector machine by Vapnik, called the support vector domain description (SVDD). This data description can be used for novelty or outlier detection. A spherically shaped decision boundary around a set of objects is constructed by a set of support vectors describing the sphere boundary. It has the possibility of transforming the data to new feature spaces without much extra computational cost. By using the transformed data, this SVDD can obtain more flexible and more accurate data descriptions. The error of the first kind, the fraction of the training objects which will be rejected, can be estimated immediately from the description without the use of an independent test set, which makes this method data efficient. The support vector domain description is compared with other outlier detection methods on real data. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Data domain description; Outlier detection; One-class classification; Support vector machines

1. Introduction

Most pattern recognition tasks deal with classification or regression problems. But there is a third, less well-known extension of the classification problem, the data domain description problem (also called one-class classification). In domain description the task is not to distinguish between classes of objects like in classification problems or to produce a desired outcome for each input object like in regression problems, but to give a *description* of a set of objects. This description should cover the class of objects represented by the training set, and ideally should reject *all* other possible objects in the object space. The data domain description is used for outlier detection or novelty detection, the detection of objects which differ in some sense significantly from the rest of the dataset.

Different methods for data domain description or outlier detection have been developed. When an underlying statistical law for the outlying patterns is assumed, this underlying distribution should be estimated (Ritter and Gallegos, 1997). When nothing about the outlier distribution can be assumed (or if an insufficient number of outlier examples is available), only a description of (the boundary of) the target class can be made. Most often a probability density of the data is estimated and new test objects which are under some probability threshold will be rejected. For instance, in

[*] Corresponding author. Tel.: +31-15-2781845; fax: +31-15-278-6740.

E-mail address: davidt@ph.tn.tudelft.nl (D.M.J. Tax)

[1] This work was partly supported by the Foundation for Applied Sciences (STW) and the Dutch Organization for Scientific Research (NWO).

PII: S 0 1 6 7 - 8 6 5 5 (9 9) 0 0 0 8 7 - 2

the paper of Tarassenko et al. (To appear), anomalities in mammographs are detected by applying Parzen density estimation and a mixture of Gaussians on the normal class. A drawback of these density methods is that they often require a large dataset, especially when high dimensional feature vectors are used. Also problems may arise when large differences in density exist: objects in low density areas will be rejected although they are legitimate objects.

In this paper, another method for data domain description is presented and analyzed (the idea was first presented in (Tax and Duin, 1999)). The method is inspired by the support vector machines by Vapnik (1995). For data domain description not the optimal separating hyperplane has to be found, but the sphere with minimal volume (or minimal radius) containing all objects. First we give a theoretical derivation of the basic method in Section 2. In Sections 3 and 4 we focus on choices for the parameters which are still free and look at some characteristics of the methods. Experimental results will be shown in Section 5, and we give conclusions in Section 6.

2. Theory

Of a data set containing N data objects, $\{x_i, i = 1, \ldots, N\}$, a description is required. We try to find a sphere with minimum volume, containing all (or most of) the data objects. This is very sensitive to the most outlying object in the target data set. When one or a few very remote objects are in the training set, a very large sphere is obtained which will not represent the data very well. Therefore, we allow for some data points outside the sphere and introduce slack variables ξ_i (analogous to (Vapnik, 1995)).

Of the sphere, described by center a and radius R, we minimize the radius

$$F(R, a, \xi_i) = R^2 + C\sum_i \xi_i, \tag{1}$$

where the variable C gives the trade-off between simplicity (or volume of the sphere) and the number of errors (number of target objects rejected).

This has to be minimized under the constraints

$$(x_i - a)^{\mathrm{T}}(x_i - a) \leqslant R^2 + \xi_i \quad \forall_i, \xi_i \geqslant 0. \tag{2}$$

Incorporating these constraints in (1), we construct the Lagrangian,

$$L(R, a, \alpha_i, \xi_i) = R^2 + C\sum_i \xi_i - \sum_i \alpha_i\{R^2 + \xi_i - (x_i^2 - 2ax_i + a^2)\} - \sum_i \gamma_i \xi_i, \tag{3}$$

with Lagrange multipliers $\alpha_i \geqslant 0$ and $\gamma_i \geqslant 0$. Setting the partial derivatives to 0, new constraints are obtained:

$$\sum_i \alpha_i = 1, \quad a = \frac{\sum_i \alpha_i x_i}{\sum_i \alpha_i} = \sum_i \alpha_i x_i, \quad C - \alpha_i - \gamma_i = 0 \quad \forall_i. \tag{4}$$

Since $\alpha_i \geqslant 0$ and $\gamma_i \geqslant 0$ we can remove the variables γ_i from the third equation in (4) and use the constraints $0 \leqslant \alpha_i \leqslant C \ \forall_i$.

Rewriting Eq. (3) and resubstituting Eqs. (4) give to maximize with respect to α_i:

$$L = \sum_i \alpha_i(x_i \cdot x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j), \tag{5}$$

with constraints $0 \leqslant \alpha_i \leqslant C$, $\sum_i \alpha_i = 1$.

The second equation in (4) states that the center of the sphere is a linear combination of data objects, with weight factors α_i which are obtained by optimizing Eq. (5). Only for a small set of objects the equality in Eq. (2) is satisfied: these are the objects which are on the boundary of the sphere itself. For those objects the coefficients α_i will be non-zero and are called the support objects. Only these objects are needed in the description of the sphere. The radius R of the sphere can be obtained by calculating the distance from the center of the sphere to a support vector with a weight smaller than C. Objects for which $\alpha_i = C$ have hit the upper bound in (4) and are outside the sphere. These support vectors are considered to be outliers. We will discuss the parameter C in more detail in the next section.

To determine whether a test point z is within the sphere, the distance to the center of the sphere has to be calculated. A test object z is accepted when this distance is smaller than the radius, i.e., when $(z-a)^{\mathrm{T}}(z-a) \leqslant R^2$. Expressing the center of the sphere in terms of the support vectors, we accept objects when

$$(z \cdot z) - 2\sum_i \alpha_i (z \cdot x_i) + \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \leqslant R^2. \tag{6}$$

3. Generalizing to other kernels

The method just presented only computes a sphere around the data in the input space. Normally, data are not spherically distributed, even when the most outlying objects are ignored. So, in general, we cannot expect to obtain a very tight description. Since the problem is stated completely in terms of inner products between vectors (Eqs. (5) and (6)), the method can be made more flexible, analogous to (Vapnik, 1995). Inner products of objects $(x_i \cdot x_j)$ can be replaced by a kernel function $K(x_i, x_j)$, when this kernel $K(x_i, x_j)$ satisfies Mercer's theorem. This implicitly maps the objects x_i into some feature space and when a suitable feature space is chosen, a better, more tight description can be obtained. No explicit mapping is required, the problem is expressed completely in terms of $K(x_i, x_j)$.

Therefore, we replace all inner products $(x_i \cdot x_j)$ by a proper $K(x_i, x_j)$ and the problem of finding a data domain description is now given by (see (5))

$$L = \sum_i \alpha_i K(x_i, x_i) - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j), \tag{7}$$

with constraints $0 \leqslant \alpha_i \leqslant C$, $\sum_i \alpha_i = 1$. A test object z is accepted when (see (6))

$$K(z, z) - 2\sum_i \alpha_i K(z, x_i) + \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \leqslant R^2. \tag{8}$$

Different kernel functions K result in different description boundaries in the original input space. The problem is to find a suitable kernel function $K(x_i, x_j)$. We discuss two choices: a polynomial kernel and a Gaussian kernel.

The first choice for kernel $K(x_i \cdot x_j)$ is the extended inner product: $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$, where the free parameter d is the degree of the polynomial kernel. As argued by Vapnik (1995), this kernel maps the objects into the high dimensional feature space by adding products of the original features, up to degree d. (For example, a 2D vector (x_1, x_2) is mapped to $(x_1, x_2, x_1x_2, x_1^2, x_2^2)$ when a polynomial kernel with $d = 2$ is used.)

This kernel does, in general, not result in good tight descriptions. For higher degrees d, the influence of objects most remote from the origin of the coordinate system increases and overwhelms all other inner products. This effect is shown in Fig. 1 with a two-dimensional dataset containing 10 objects. For different values of the degree $(d = 1, 10, 25)$ a sphere description is computed.

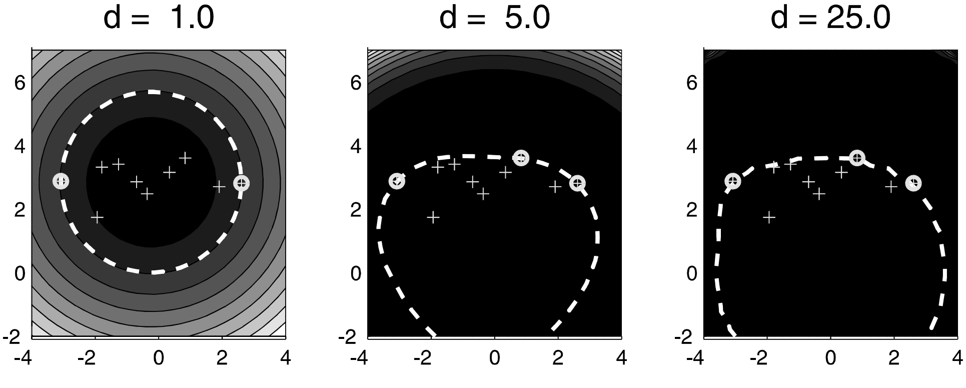

Fig. 1. Distance to the center of the hypersphere, mapped back on the input space for a polynomial kernel. The darker the color, the smaller the distance. The white dashed line indicates the surface of the hypersphere. The small circles indicate support objects.

The distance to the center of the sphere is plotted in the original input space. The dashed white line crossing the support vectors (indicated by the small circles) is the boundary of the description. The objects in the upper part are most distant from the origin and although these objects are not the most outlying objects in the data in two dimensions, they become the support vectors when higher degrees are used. Note that a large part of the input space becomes accepted. This description results in a very large and sparse sphere in the original two-dimensional input space.

To suppress the growing distances for larger feature spaces, a Gaussian kernel $K_G(x_i,x_j) = \exp(-(x_i - x_j)^2/s^2)$ is more appropriate. Eq. (7) then becomes

$$L = 1 - \sum_i \alpha_i^2 - \sum_{i \neq j} \alpha_i \alpha_j K_G(x_i, x_j), \tag{9}$$

and the acception rule, Eq. (8), becomes

$$-2 \sum_i \alpha_i K_G(z, x_i) \leqslant R^2 - C_X - 1, \tag{10}$$

where C_X only depends on the support vectors and the α_i and not on the test object z.

In Fig. 2, again a 2D artificial dataset containing 10 objects is shown. Now a support vector domain description with a Gaussian kernel for different values of s is used. The width parameter s ranges from very small ($s = 1.0$ in the leftmost figure) to large ($s = 25.0$ in the rightmost figure). Note that the number of support vectors decreases and that the description becomes more sphere-like.

We can derive explicit solutions for Eq. (7) for the two different extreme situations, one for very small values and one for very large values of s. For very small s, $K_G(x_i,x_j) \simeq 0$, $i \neq j$ and $L = 1 - \sum_i \alpha_i^2$. This is maximized when $\alpha_i = 1/N$ and L becomes $1 - 1/N$. This is similar to the Parzen density estimation, where each object supports a kernel (see Eq. (10)). All distances to the center of the sphere become $1 - 1/N$.

For very large s, $K_G(x_i,x_j) = 1$ and $L = 1 - \sum_i \alpha_i^2 - \sum_{i \neq j} \alpha_i \alpha_j$. This is maximized when all $\alpha_i = 0$ except for one $\alpha_j = 1$ and all distances to the sphere center become 0. This infinitely large sphere will not be obtained in practice and s will not be large enough to give equal $K_G(x_i,x_j)$ for all pairs i,j. In the rightmost subplot of Fig. 2 a realistic limit situation is plotted. The data description is again the smallest sphere which covers the complete dataset, without outliers. A Taylor expansion of Eq. (9) shows that when higher orders are ignored, Eq. (5) is obtained (up to a scaling and offset factor).

In the case of moderate values of s (middle plot in Fig. 2) just a fraction of the objects become support objects. Eq. (10) shows that in this case an edited and weighted Parzen density estimation is obtained. This does not estimate the total density of the data, but tries to describe just the boundary of the dataset.

The parameter C gives the upper boundary for the parameters α_i and thus limits the influence of the individual support vectors on the description, Eq. (10). When an object x_1 obtains $\alpha_i = C$, the description will not be adapted any further

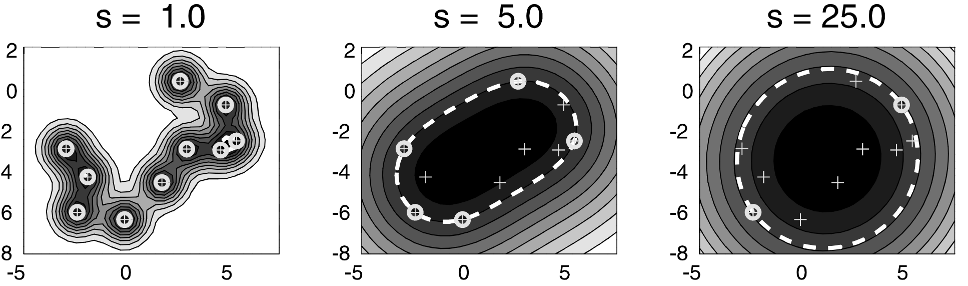

Fig. 2. Distance to the center of the hypersphere, mapped back on the input space for a Gaussian kernel. The darker the color, the smaller the distance. The white dashed line indicates the surface of the hypersphere. The small circles indicate the support objects.

towards this object and it will stay outside the sphere. Because of the constraints $\sum_i \alpha_i = 1$ and $\alpha_i \geqslant 0$, only the choices for which C can have any influence on the solution of Eq. (9) is when $1/N \leqslant C \leqslant 1$. For $C < 1/N$ no solution can be found because then the constraint $\sum_i \alpha_i = 1$ can never be met, while for $C > 1$ one can always find a solution (α_i's are always less or equal to 1).

When C is restricted to small values, the cost of being outside the sphere is not very large and a larger fraction of the objects is allowed to be outside the sphere. In practice the value of C is not very critical. In the experiments of this paper, $C = 0.25$ is chosen and in none of the cases an outlier is detected in the target class. When a smaller $C = 0.2$ or a larger $C = 0.4$ is used, the same results are obtained.

4. Generalization

To get an indication of the generalization or the overfitting characteristics of the SVDD, we have to get an indication of (1) the number of target patterns that will be rejected (errors of the first kind) by this description and (2) of the number of outlying patterns that will be accepted (errors of the second kind).

We can estimate the error of the first kind by applying the leave-one-out method on the training set containing the target class (Vapnik, 1995). When we leave out an object from the training set which is not a support object, the original solution is found and all training objects will be found. When a support object is left out, the optimal sphere description can be made smaller, because this support object is on the boundary of the sphere. This left-out object will then be rejected, while the rest of the training objects will still be accepted (because the method is trained on these data). Thus, the error can be estimated by

$$E[P(\text{error})] = \frac{\#SV}{N}, \tag{11}$$

where $\#SV$ is the number of support vectors.

When we use a Gaussian kernel, we can regulate the number of support vectors by changing the width parameter s. Therefore, we can also set the error of the first kind. When the number of support vectors is too large, we have to increase s, while when the number is too low, we have to decrease s. To check how well the estimate of Eq. (11) is, we plotted in Fig. 3 the estimation of the errors of the first kind as a function of the width parameter s. The method was applied to a two-dimensional dataset containing 10 objects. Also the error, estimated on an independent test set of 100 objects, is shown. We can conclude that this estimate works well.

So when a description of a dataset is required, we can set beforehand a bound on the expected rejection rate of the target data. The Lagrangian from Eq. (9) is solved and the expected error for this solution is obtained via Eq. (11). When this error is too large, the width parameter s is increased, or when this error can still increase, the width parameter s is decreased. This guarantees that the width parameter in the SVDD is adapted for the problem at hand, given the error.

The chance that outlying objects will be accepted by the sphere description, the error of the second kind, cannot be estimated by this measure. In general, only a good description of the target class in the form of a training set is available. All other patterns are considered outliers. To get an estimate for the error of the second kind, data

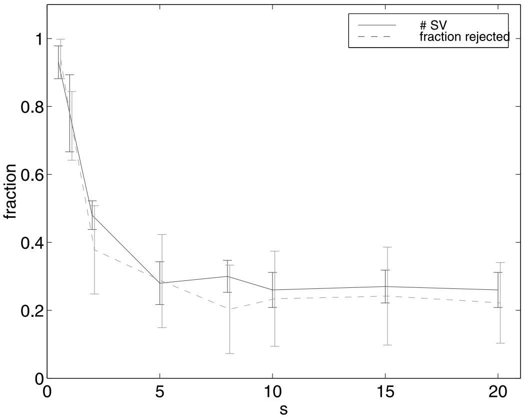

Fig. 3. Comparison between the fraction of objects which are support objects and the fraction of test points which is rejected, with respect to the parameter s. The target class consists of the 10 objects shown in Figs. 1 and 2.

around the original data set should be created and tested. This method then requires a way of obtaining or creating data around the training data, but not *in* the training set. Also the number of test patterns should be sufficiently high for a reasonable estimate, which can be a problem in higher dimensional feature spaces.

In the experiments in this paper we circumvent this problem by using classification problems for testing the method. From the classification tasks we take one class as being the outlier class, and all other classes will be used as the target class. In this way artificial outliers can be constructed. This means that a performance bias is introduced. These classification problems often contain overlapping classes and by using the classification problems in this way, the performance of the outlier methods will be lower than that of normal classification methods for the classification task. Still, it gives an indication of the performances when different outlier methods are compared.

5. Experiments

The SVDD method is compared with four other outlier detection methods: normal density estimation, Parzen windows, a k-nearest-neighbor distance comparison and an instability method. These methods are described in more detail in (Tax and Duin, 1998). The first two methods rely on a density estimation of the data. The third method compares the distance from a test object $\boldsymbol{x}$ to its nearest neighbor $\mathrm{NN}^{\mathrm{tr}}(\boldsymbol{x})$ in the training set with the distance from this nearest neighbor $\mathrm{NN}^{\mathrm{tr}}(\boldsymbol{x})$ to *its* nearest neighbor $\mathrm{NN}^{\mathrm{tr}}(\mathrm{NN}^{\mathrm{tr}}(\boldsymbol{x}))$ in the trainset. The instability method is specially designed for outlier detection in classification tasks. By training several simple classifiers, such as linear classifiers, on bootstrapped versions of the training set, one obtains variations in the classifier outputs. Objects which experience large variations in these outputs are likely to reside in low density or low confidence areas and will be rejected.

These methods will be compared to the SVDD method with a Gaussian kernel. The width parameter s is found by using the procedure mentioned in Section 4. The variable C is set to 0.25. On the basis of the performance on the training set samples, we set a target rejection threshold value of 10% on the different measures. This relatively large value is chosen, because some datasets contain a small number of objects, and using a 10% rejection rate ensures that some of the target objects will indeed be rejected. After that the performance on a test set containing the target class and one containing the outlier class is measured. This means that the optimal performance is reached when all outlying objects are rejected and 90% of the target class is accepted.

All methods are applied to a set of standard datasets taken from the UCI Machine Learning Dataset Repository (Blake et al., 1998). The datasets considered are listed in Table 1. As explained in the previous section, one of the classes is considered as the outlier class, the rest is target class. To estimate the errors (of the first and the second kind) n-fold cross-validation with $n = 5$ is used.

In Table 2, the performances of the outlier detection methods on all UCI datasets are shown. For each method, the performance on a target validation set (left) and an outlier test set (right) is shown. Each of the classes is outlier class once (indicated in the first column). Results on the balance-dataset already show that the estimation of the errors of type 1 on the training set is not very precise for the Parzen density estimation and

Table 1
UCI datasets used for the evaluation of the data description methods

Name	# Objects	# Classes	# Features
Balance-scale	625	3	4
Breast-cancer-Wisconsin	699	2	9
Ionosphere	351	2	34
Iris	150	3	4

Table 2
Outlier detection performances on the UCI datasets[a]

Class no.	Set size	Gauss	Parzen	kNN	Instab	SVDD
Balance data						
1	337, 288	0.13, 0.74	0.46, 1.00	0.00, 0.65	0.12, 0.73	0.14, 0.89
2	576, 49	0.11, 0.30	0.40, 0.51	0.00, 0.00	0.08, 0.76	0.12, 0.13
3	337, 288	0.12, 0.74	0.40, 1.00	0.00, 0.65	0.12, 0.76	0.11, 0.88
Breast cancer data						
1	241, 458	0.14, 0.46	0.91, 1.00	0.10, 0.17	0.00, 0.00	0.09, 0.94
2	458, 241	0.11, 0.99	0.28, 1.00	0.07, 0.45	0.00, 0.00	0.10, 0.99
Ionosphere data						
1	126, 225	0.36, 0.06	0.91, 0.98	0.11, 0.03	0.00, 0.00	0.13, 0.00
2	225, 126	0.11, 0.90	0.94, 1.00	0.09, 0.67	0.00, 0.00	0.11, 0.90
Iris data						
1	100, 50	0.13, 1.00	0.33, 1.00	0.12, 1.00	0.11, 0.46	0.11, 1.00
2	100, 50	0.13, 0.93	0.30, 0.97	0.09, 0.49	0.12, 0.15	0.11, 0.40
3	100, 50	0.12, 0.91	0.43, 1.00	0.09, 0.51	0.14, 0.58	0.09, 0.90

[a] The first column gives the class which is considered as outlier. In the second column, the target (left) and outlier (right) set sizes are given. In the other columns, the leftmost number in each column gives the performance for a test set containing the target class and the rightmost number the performance on an outlier set containing the outlier class.

the kNN method. In both cases the error on the 'target' class is far larger than the predefined 0.1. All methods perform poorly on the case in which the second class is considered outlier. This can be understood by looking at the distribution of the data, where class 2 is between classes 1 and 3. Only the instability method is able to reject objects from the second class.

In the breast-cancer data set, the second class is clearly easier to distinguish than the first class. Looking at the origin of the data, this means that by describing the benign class, the malignant class can be rejected quite well. All methods perform well in describing class 1, except for the instability method. Since the original dataset contains only two classes, the instability method could not be used. When one class is considered as the outlier class, the instability method cannot train simple classifiers on the remaining class. Also visible is that the Parzen method overtrains heavily and performs poorly when class 1 is the outlier class. The SVDD performs best overall.

In the ionosphere dataset, the Parzen density estimation again overtrains and the instability method cannot be used because only two classes are available. From the results we see that class 1 is almost Gaussian distributed and class 2 is scattered around it. The SVDD cannot distinguish one class 2 object from class 1.

Finally, the performance of the outlier methods are applied on the iris dataset. Here, all methods work reasonably well, which indicates that the data distributions of the classes are well clustered. Only the Parzen density estimation slightly overtrains.

From these results we can conclude that the SVDD works comparably and often better than the other outlier methods, from the simple Gaussian distribution to the nearest neighbor method. Another advantage of the SVDD is that an estimate of the error on the target set can be obtained immediately by looking at the fraction of support vectors. This guarantees that the scale of the SVDD, set by the width parameter s, is adjusted to the data and no extra leave-one-out estimation is required (like in the Parzen estimation).

6. Conclusions

Data domain description is an important tool for robust and confident classification. Data which do not resemble a target class should be rejected. In this paper we propose a sphere shaped data description which does not have to make a probability density estimation. The sphere description depends on a few target objects, the support objects and new test objects only have to be compared with these support objects by an inner product or some more general kernel function. By adapting the kernel function, this method becomes more flexible than just a sphere in the input space. The SVDD also allows for target objects not included in the sphere description. An extra parameter C is introduced to give the trade-off between the number of errors made on the training set and the size of the sphere description. In practice, the size of this parameter is not very crucial for finding a good solution.

In this paper two kernel types are considered: the polynomial and the Gaussian kernel. In general, the polynomial kernel does not give tight descriptions of the training data. On the other hand, the Gaussian kernel seems to work very well. In the SVDD using a Gaussian kernel, another free parameter, the width of the kernel s, can be adapted. By choosing different extremes for this width parameter, the sphere method obtains more or less flexible descriptions. For very small values for s, a Parzen density estimation is obtained. In that case, all target objects become support objects. For very large values of s, just one prototype for the complete data set is used and almost the complete training set can be disregarded. Applying a moderate value for the width parameter, an edited and weighted Parzen estimation is obtained.

An extra feature of this SVDD method is that the error on the target class can be estimated immediately by calculating the fraction of target objects which become support objects. Setting the error on the training set beforehand, the width s can be set such that the fraction of support objects is equal to this error. Since the SVDD focuses on the boundary description and not on the complete data density, the required number of objects is smaller than for, e.g., the Parzen density estimation. We can conclude that the SVDD gives both an efficient and robust method for describing a dataset.

For further reading, see (Ypma and Pajunen, 1999).

Discussion

Gimel'farb: Can you tell why the SVDD approach works so poorly ? Since it depends on your choice of the kernel function, then if $s = 0$, it is simply a nearest neighbor classifier. And such a nearest neighbor classifier, in this case, cannot perform so poorly.

Tax: No, but I said that the error on my target set is about 10%. So, I tune my parameters in such a way that I will reject about 10% and 10% of my training set will be support vectors. In that case, the description will be different from the normal Parzen estimator. It will be a more crude approximation of the boundary. If I had more points, it would be comparable, better than the Parzen estimator.

Gimel'farb: One more question: why do you need to restrict yourself to single sphere approximation, because by using the earlier results of Vapnik, you can approximate any distribution of your training points by a minimal number of spheres, and in that case, you have a much better description.

Tax: First of all, if you have a multi-modal distribution, for instance, three Gaussian distributions, and if you have enough training points, it will automatically find three spheres. If you do not have enough data and you still restrict yourself to an error of 10% on your target set, it will still give you just one complete blob. So it only finds that solution for which it finds enough justification in the data. If you are very strict on the number of errors you make, it will give very broad, very crude approximations.

Gimel'farb: But this means that you should not restrict yourself to a fixed error on the target set, because in that case, the result depends on your data. Sometimes, if you fix the rejection rate, then, even for beautiful data, you intentionally obtain bad results.

Tax: True, but if you know that you have three clusters, I would not recommend this method. I would then rather take three Gaussians. If that is the prior knowledge that you have, then use it.

Kanal: You might assume one, two, three clusters, and so on, to see which assumption gives the best results.

Tax: But then, the tricky part is always to find a good threshold value. And here, I find my threshold on the basis of the number of support vectors, and that gives a more direct link to how good or bad the description is. From the SVDD, you cannot find directly the number of clusters in the data.

References

Blake, C., Keogh, E., Merz, C., 1998. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, University of California, Irvine, Department of Information and Computer Sciences.

Ritter, G., Gallegos, M.T., 1997. Outliers in statistical pattern recognition and an application to automatic chromosome classification. Pattern Recognition Letters 18, 525–539.

Tarassenko, L., Hayton, P., Brady, M., To appear. Novelty detection for the identification of masses in mammograms.

Tax, D., Duin, R., 1998. Outlier detection using classifier instability. In: Amin, A., Dori, D., Pudil, P., Freeman, H. (Eds.), Advances in Pattern Recognition Proc. Joint IAPR Internat. Workshops SSPR'98 and SPR'98, Sydney, Australia. Lecture Notes in Computer Science, Vol. 1451. Springer, Berlin, pp. 593–601.

Tax, D., Duin, R., 1999. Data domain description using support vectors. In: Verleysen, M. (Ed.), Proc. European Symposium Artificial Neural Networks 1999. D. Facto, Brussel, pp. 251–256.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, New York.

Ypma, A., Pajunen, P., 1999. Rotating machine vibration analysis with second-order independent component analysis. In: Proc. 1st Internat. Workshop Independent Component Analysis and Signal Separation, ICA'99, pp. 37–42.