Feature Scaling in Support Vector Data Descriptions

David M.J. Tax and **Robert P.W. Duin**
Pattern Recognition Group
Faculty of Applied Science, Delft University of Technology
Lorentzweg 1, 2628 CJ Delft, The Netherlands
e-mail: {davidt,bob}@ph.tn.tudelft.nl

Abstract

In previous research the Support Vector Data Description is proposed to solve the problem of One-Class classification. In One-Class classification one set of data, called the target set, has to be distinguished from the rest of the feature space. This description should be constructed such that objects not originating from the target set, by definition the outlier class, are not accepted by the data description. In this paper the Support Vector Data Description is applied to the problem of image database retrieval. The user selects an example image region as target class and resembling images from a database should be retrieved. This application shows some of the weaknesses of the SVDD, particularly the dependence on the scaling of the features. By rescaling features and combining several descriptions on well scaled feature sets, performance can be significantly improved.

Introduction

When one class of the data is to be distinguished from the rest of the feature space, a closed boundary around this set should be defined. This One-Class classification is often solved using density estimation or a model based approach. In this paper we use a method inspired by the Support Vector Classifier (Vapnik 1998). Instead of using a hyperplane to distinguish between two classes, a hypersphere around the target set is used. This method is called the Support Vector Data Description (SVDD)(Tax & Duin 1998).

In general the problem of One-Class classification is harder than the problem of normal Two-Class classification. For normal classification the decision boundary is supported from both sides by examples of each of the classes. Because in the case of One-Class classification only one set of data is available, only one side of the boundary is covered. On the basis of one class it is hard to decide how tight the boundary should fit around the data in each of the directions. Even harder to decide is what the optimal scaling of the features should be to find the best separation of the target and outlier class.

To investigate the impact of this problem in a real application, the SVDD will be applied to an image database retrieval problem in which the user defines some interesting and desired image regions, and the application should retrieve the resembling images from the database. Only the user indicated regions will be used in training. The database is given by (Messer & Kittler 1999). Several features will be available, both color and texture features. In literature it is well known that color features perform very well in distinguishing the target images from the rest of the database(Antani, Kasturi, & Jain 1998). Some very advanced techniques for image database retrieval have been used to use both types of features. A well optimized retrieval procedure, including automatic feature selection and extraction is also found in (Messer & Kittler 1999).

In this paper we focus on the problem of the scaling of the data. First we will explain the Support Vector Data Description. Some characteristics of the image database and information about the queries will be given. Then the results of the queries by the SVDD will be shown and we conclude with the discussion.

Support Vector Data Description

For description of the domain of a dataset we capture it with a hypersphere with minimum volume. By minimizing the volume of the captured feature space, we hope to minimize the chance of accepting outliers. Inspired by the Support Vector Method by Vapnik (see (Vapnik 1995), or for a more simple introduction (Tax, de Ridder, & Duin 1997)) one can extend this idea to determine an arbitrary shaped region in the original feature space, the Support Vector Data Description (SVDD) method (Tax & Duin 1999).

Assume we have a data set containing N data objects, $\{\mathbf{x}_i, i = 1, .., N\}$ and the sphere is described by center $\mathbf{a}$ and radius R. To allow the possibility of outliers in the training set, the distance from $\mathbf{x}_i$ to the center $\mathbf{a}$ should not be strictly smaller than R^2, but larger distances should be penalized. Therefore slack variables ξ_i are introduced. An error function containing the volume of the sphere and the distance from the boundary of the outlier objects is minimized. The constraints that objects are within the sphere are imposed by applying

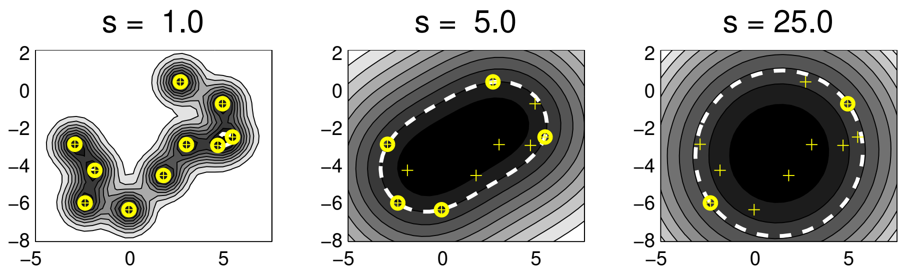

Figure 1: Distance to the center of the hypersphere, mapped back on the input space for a Gaussian kernel. The darker the color, the smaller the distance. The white dashed line indicates the surface of the hypersphere. The small circles indicate the support objects.

Lagrange multipliers:

$$L(R, \mathbf{a}, \alpha_i) = R^2 + C\sum_i \xi_i - \sum_i \alpha_i\{R^2 - (\mathbf{x}_i^2 - 2\mathbf{a}\cdot\mathbf{x}_i + \mathbf{a}^2)\} - \sum_i \gamma_i\xi_i$$

with Lagrange multipliers $\alpha_i \geq 0$ and $\gamma_i \geq 0$. This function has to be minimized with respect to $R, \mathbf{a}$ and ξ_i and maximized with respect to α_i and γ_i.

Setting the partial derivatives of L to R and $\mathbf{a}$ to zero, gives:

$$\sum_i \alpha_i = 1$$

$$\mathbf{a} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\sum_i \alpha_i} = \sum_i \alpha_i \mathbf{x}_i \tag{1}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \gamma_i = 0 \tag{2}$$

From the last equation $\alpha_i = C - \gamma_i$ and because $\alpha_i \geq 0, \gamma_i \geq 0$, Lagrange multipliers γ_i can be removed when we demand that

$$0 \leq \alpha_i \leq C \tag{3}$$

Resubstituting these values in the Lagrangian gives to maximize with respect to α_i:

$$L = \sum_i \alpha_i(\mathbf{x}_i\cdot\mathbf{x}_i) - \sum_{i,j}\alpha_i\alpha_j(\mathbf{x}_i\cdot\mathbf{x}_j) \tag{4}$$

with $0 \leq \alpha_i \leq C$, $\sum_i \alpha_i = 1$.

This function is a standard Quadratic Optimization problem and it should be maximized with respect to α_i. In practice it appeared that a large fraction of the α_i become zero. The objects for which $\alpha_i > 0$ are called the Support Objects, and these are important in the computation of the center $\mathbf{a}$. All other objects with $\alpha_i = 0$ can be disregarded. This can drastically reduce the computation.

Object $\mathbf{z}$ is accepted when:

$$\begin{aligned}(\mathbf{z}-\mathbf{a})(\mathbf{z}-\mathbf{a})^T &= (\mathbf{z} - \sum_i \alpha_i\mathbf{x}_i)(\mathbf{z} - \sum_i \alpha_i\mathbf{x}_i) \\ &= (\mathbf{z}\cdot\mathbf{z}) - 2\sum_i \alpha_i(\mathbf{z}\cdot\mathbf{x}_i) + \sum_{i,j}\alpha_i\alpha_j(\mathbf{x}_i\cdot\mathbf{x}_j) \\ &\leq R^2\end{aligned} \tag{5}$$

In general this does not give a very tight description. The model of a hypersphere is assumed, which will not be satisfied in the general case. Analogous to the method of Vapnik (Vapnik 1998), the inner products $(x \cdot y)$ in equations (4) and (5) can be replaced by kernel functions $K(\mathbf{x}, \mathbf{y})$ which gives a much more flexible method. When the inner products are replaced by Gaussian kernels for instance, we obtain:

$$(\mathbf{x}\cdot\mathbf{y}) \rightarrow K(\mathbf{x}, \mathbf{y}) = \exp(-(\mathbf{x}-\mathbf{y})^2/s^2) \tag{6}$$

Equation (4) now changes into:

$$L = 1 - \sum_i {\alpha_i}^2 - \sum_{i\neq j}\alpha_i\alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{7}$$

and and the formula to check if a new object $\mathbf{z}$ is within the sphere (equation (5)) becomes:

$$1 - 2\sum_i \alpha_i K(\mathbf{z}, \mathbf{x}_i) + \sum_{i,j}\alpha_i\alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \leq R^2 \tag{8}$$

So a more flexible description than the rigid sphere description is obtained. In figure 1 the resulting data description is shown for a simple 2D dataset. For different values for the width parameter s in the kernel (equation (6)) the resulting decision boundary is plotted. Note that the number of support vectors changes for different s. As shown in (Tax & Duin 1999) rejection rate of the target set, i.e. the error on the target set can be estimated by the number of support vectors:

$$E[P(\text{error})] = \frac{\#SV}{N} \tag{9}$$

where $\#SV$ is the number of support vectors. The number of support vectors can be regulated by changing the width parameter s and therefore also the error on the target set. The upper bound C on α_i can be set when an assumption on the fraction of outliers in the target set is made.

When examples of outliers are available, the SVDD can be adapted to use these to obtain a tighter description(Tax & Duin 1999). Note that we cannot set a priori restrictions on the error on the *outlier* class. In general we only have a good representation of the target class and the outlier class is per definition everything else.

Image database

Although the SVDD does not assume a certain distribution for the target class, it requires well-scaled data: distances in feature space should be more or less homogeneous. To investigate if this assumption is satisfied in practical situations, we apply the method to the image database from Messer(Messer & Kittler 1999). The image database contains 3483 images, ranging from capture TV images to images from the MPEG-7 test set. The user chooses interesting regions in a target image. On the pixels extracted from the selected image patches an One-Class classifier is trained. Finally the resembling images (both in color and texture) should be retrieved from the database. In the practical application the speed of the retrieval is important, but in this paper we only focus on the retrieval accuracy.

In figure 2 five test queries are defined (originally from (Messer & Kittler 1999)). For each of the target images the user has defined 2 regions. To test the performance of the One-Class classifiers, also 5 to 8 other target images are defined, which resemble the target images. To give a ranking of the images, an error has to be defined. Most importantly the pixels should be accepted by the description. Furthermore the pixels are weighted by the distance to the center of the sphere or the size of the region to which they belong. The errors which are used, will be explained further in the next section.

The color and texture features of the image are calculated beforehand to accelerate the query. For each pixel in each image 33 features are calculated. They are listed in table 1.

set	dim	type of feature	σ
1	9	discrete cosine transform	$1 \cdot 10^4$
2	8	gabor filters	$2 \cdot 10^3$
3	3	energy (green/int/red)	$3 \cdot 10^{-2}$
4	3	entropy (green/int/red)	$6 \cdot 10^{-2}$
5	3	mean (green/int/red)	$2 \cdot 10^0$
6	3	variance (green/int/red)	$6 \cdot 10^2$
7	4	wavelet transform	$1 \cdot 10^3$

Table 1: List of the 33 features, divided in 7 feature sets. Last column gives the average standard deviation of the data obtained from the brick wall region from the Cathedral image.

To further lower the computational burden in the processing of the complete image database, all images are segmented by clustering the pixels in the 33 dimensional feature space. On average about 30 to 50 cluster centers are obtained for each of the images. Only the cluster centers, called the *indices*, are used in the testing. This reduces the images from 300×200 pixels to about 50 indices. The drawback is that only an average of the cluster is considered and that the variance structure of the cluster is lost. An example of a clustered image with indices is given in figure 3.

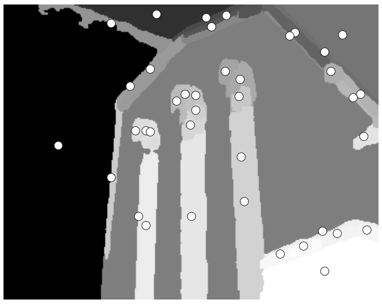

Figure 3: Clustered (or segmented) Cathedral image, containing 38 clusters. The cluster centers (indices) and they approximate positions are given by the dots.

Finally a total retrieval error is defined which is independent on the image database size and the definition of the query. It is the chance that a random method (a method which just randomly ranks the images having an uniform distribution over the whole database) shows the same ranking as the method under investigation. When the total database consist of M images and n images are ranked, the distribution of average rank m of the n images by the random method will be distributed like:

$$p(\overline{m}, n, M) = \mathcal{N}\left(\overline{m}; \mu = \frac{M+1}{2}, \sigma^2 = \frac{(M-1)^2}{12n}\right) \tag{10}$$

where $\mathcal{N}(\mathbf{x}; \mu, \sigma^2)$ is the Normal distribution. In general the average ranking obtained by a non-random method will be better. Integration over m upto the average rank $\overline{m}$ of equation (10) gives an indication how (un-)likely the results would be in case of a random method:

$$\mathcal{E}(\overline{m}, n, M) = \int_{-\infty}^{\overline{m}} p(m, n, M) dm \tag{11}$$

Assume we have an image database containing $M = 3500$ images, and a query with $n = 5$ target images is defined. When the images are ranked $(1, 2, 3, 4, 5)$, $\overline{m} = 3$, the error becomes $\mathcal{E} = 5.09 \cdot 10^{-15}$, while for ranking $(1500, 1550, 1600, 1650, 1700)$ $\mathcal{E} = 0.2526$. The results of the experiments will be shown in a more readable format as $\log_{10}(\mathcal{E})$, giving -14.29 and -0.59 respectively.

Figure 2: The five defined queries, Cathedral, Queen, Hut, Flag, Newsreader, each with two user defined regions.

Experiments

In all experiments the Support Vector Data Description is trained on the user defined regions such that about 10% of the training pixels is expected to fall outside the description (estimated by equation (9)). The training time used by the Quadratic Optimization routine was restricted by using just 200 target objects (i.e. pixels randomly drawn from the user defined regions) per description. Using larger training sets gave sometimes better results, but required long training times (longer than 5 minutes).

Normal features

To see how hard the image retieval problem is, first an SVDD is trained on the original 33 features, without preprocessing. In table 3 the definitions of the errors is shown. First it is determined which indices are accepted by the SVDD. After that the accepted indices are ranked according to distance to center of the SVDD, the region size of the index or a combination of both. Finally these values are summed (of multiplied in error 4) over all indices to obtain a final score for the image.

nr	error
1	sum over region sizes of accepted indices
2	sum over distances from accepted indices to centers
3	sum over distance $\times$ (1+region size)
4	product over region sizes

Table 3: Errors defined for queries where all features are used in one SVDD

In the third column in table 2 the results are shown for the five query images on the complete image database. The results show that the four different ranking errors (table 3) give about the same performance. Results on the Cathedral, Queen and Newsreader queries are acceptable, with all desired images within the first 250 (clearly better than random guessing). The results on the Hut and Flag queries on the other hand are disappointing. It appeared that in the Hut query only the training image is recognized well (it is ranked $91, 4, 5, 43$ in the four error measures). All other images are ranked low or are rejected completely.

In the Flag query, most of the desired images are ranked very high. Only two images are not well recognized. These images are shown in figure 4. It appeared that these images are rotated over 90 degrees. This transformation is not considered irrelevant, and therefore the images are rejected.

Figure 4: Poorly ranked images in the Flag query.

The last column in table 1 shows the standard deviations of the data extracted from the wall region from the Cathedral image. Clearly the large difference in scale of the features can deteriorate the performance of the SVDD. Therefore the data is rescaled to unit variance

		all features		seperate features		outlier objects	
name query	error	no sc.	scaling	no sc.	scaling	no sc.	scaling
Cathedral	1	-13.01	-12.98	-9.23	-12.31	-12.94	-14.23
	2	-13.15	-13.18	-10.57	-7.84	-13.27	-14.23
(5 target	3	-12.87	-12.99	-10.07	-9.88	-12.82	-14.23
images)	4	-13.01	-12.99	-14.03	-13.95	-12.95	-14.23
	5			-14.01	-13.95		
	6			-14.04	-14.11		
Queen	1	-12.31	-11.82	-8.64	-12.17	-13.32	-13.49
	2	-12.46	-11.11	-13.01	-13.47	-12.45	-13.74
(5 target	3	-12.33	-11.05	-6.86	-7.13	-12.47	-13.47
images)	4	-12.84	-11.68	-12.63	-13.96	-13.51	-13.59
	5			-12.24	-13.72		
	6			-12.69	-13.96		
Hut	1	-4.31	-10.01	-10.28	-12.23	-4.99	-6.86
	2	-4.57	-10.04	-2.93	-4.89	-5.10	-6.86
(5 target	3	-4.40	-10.04	-13.23	-13.15	-5.03	-6.86
images)	4	-4.34	-10.02	-8.17	-6.60	-5.02	-6.86
	5			-8.12	-6.57		
	6			-8.22	-6.60		
Flag	1	-4.30	-8.90	-0.09	-7.18	-6.34	-2.15
	2	-2.11	-8.54	-7.10	-15.21	-4.41	-2.14
(8 target	3	-3.04	-8.64	-0.00	-0.00	-5.69	-2.15
images)	4	-4.41	-8.64	-1.99	-20.84	-6.59	-2.15
	5			-2.10	-17.57		
	6			-1.86	-10.07		
Newsreader	1	-10.16	-12.22	-1.24	-13.68	-8.69	-14.05
	2	-13.19	-13.62	-11.70	-12.98	-12.44	-14.27
(5 target	3	-12.15	-13.65	-0.08	-0.37	-11.47	-14.26
images)	4	-10.46	-13.51	-14.27	-14.29	-9.41	-14.13
	5			-14.12	-14.18		
	6			-14.24	-14.27		

Table 2: Query results for the different data processing procedures: one SVDD using all features (normal and scaled to unit variance), descriptions for seperate feature sets (also normal and scaled) and descriptions trained with outliers. The final error is defined in equation (11). The ranking errors are defined in tables 3 and 4.

in the target set and the experiments are repeated. The results are shown in the fourth column in table 2. The results on the Newsreader and especially the Hut and Flag query are improved. In most queries just a few of the desired images are poorly recognized. In some of the desired Hut images, one of the panels is absent, or partly obscured by an antenna (see figure 5). In the Flag query, the most important cue seems to be the color: the best matching images all contain the bright flag colours. The texture of the flag is not detected very well.

Separating the feature sets

To improve performance, especially in the Flag query, separate SVDDs are trained on the individual feature sets (see table 1). It is hoped that it will not only make training an SVDD easier and more efficient, it will also give the possibility to use several feature combination rules. In the Flag query this means that the One-Class classifier can be forced to use the texture information.

	distance to center	region size	weighted train perf	feature sets	indices	comb. rule
1	X			sum	sum	OR
2		X		sum	sum	OR
3			X	sum	sum	OR
4	X			prod	sum	AND
5		X		prod	sum	AND
6			X	prod	sum	AND

Table 4: Errors defined for image ranking in case of descriptions on separate feature sets

Table 4 shows the definitions of the used errors. The third column 'weighted by train perf.' indicates that the ranking is also weighted by how well the extracted indices from the training image are accepted by the One-Class classifiers. When all indices of the training image are rejected by the SVDD's, the corresponding feature set gets a low weight. A sum or a product combination over feature sets gives an OR-operation and an AND-operation respectively. In the cathedral query this means the difference between 'blue sky OR brick wall' and 'blue sky AND brick wall'.

In the fifth and sixth column of table 2 the results for the separate SVDD's without and with scaling is shown. The separate feature sets are scaled well and the rescaling does not improve performance very much. The results are somewhat worse for the Cathedral, Queen and Newsreader with respect to the original results. Only when the error contains a product over the feature sets (errors 4,5,6), performance is improved.

Figure 5: Image of Hut, with one of the panels partly occluded by an antenna.

For the Hut image the OR-operation over the feature sets performs very well. It appeared that in some of the desired images one of the panels is partly obscured by an antenna, thus destroying the texture characterization of the original panel. An example is shown in figure 5. By using the OR-operation, the influence of the wrongly characterized panels can be reduced.

Finally the Flag query still suffers from the same problems as before. The two desired images which are rotated 90 degrees are not ranked very high. Best performance is reached using an AND-operation on the feature sets, while only considering the distance to the centers to the SVDD's.

Training with example outliers

Finally the SVDD was adapted to also include example outliers in the training. One SVDD in the 33 dimensional feature space was trained, and again ranking errors given in table 3 are used. The results are shown in columns 7 and 8 in table 2. For the Cathedral, Queen and Newsreader the performance improves a bit, but for Hut and Flag it deteriorates, especially after scaling! It appears that the desired images of the Hut and Flag query are not well clustered, and using example outliers, the SVDD overtrains over the training image.

Conclusion

The Support Vector Data Description is made to seperate one class of data from the rest of the feature space. The method depends on well defined distances between objects and to investigate the influence of ill-defined distances in the data, the SVDD is applied to the problem of image database retrieval. In this problem objects are characterized by color and texture feature with very different scales.

It appeared that in this application the color feature is an important feature and that it is well clustered. Scaling both color and texture features to unit variance improves performance for image queries in which color features are most important. For queries where texture features are more important, the different color and texture features are better separated and treated separately. Combining the descriptions from the different feature sets opens the possibility of applying AND and OR operations on the different features. Then a distinction between 'blue sky AND brick wall' and 'blue sky OR brick wall' can be made. Of course this will require user input.

A normal operation procedure might therefore be, that an SVDD is trained on all 33 features, which are scaled to unit variance. In most cases this gives very acceptable results, especially when queries are focussed on color. When the user is not satisfied, data descriptions in seperate feature sets should be trained. The user then has to indicate how they should be combined (by AND or OR combinations). Of course this application can be improved in several ways. For instance, the user should be able to reject some of the high scoring images. These images than can be used as negative examples for the next query. This is a subject for further research.

Acknowledgements

We would like to thank Kieron Messer for making available the image database with all preprocessed images. This work was partly supported by the Foundation for Applied Sciences (STW) and the Dutch Organisation for Scientific Research (NWO).

References

Antani, S.; Kasturi, R.; and Jain, R. 1998. Pattern recognition methods in image and video databases: past, present and future. In *Advances in Pattern Recognition, Proceedings of SPR'98 and SSPR'98*, 31–53. Berlin: IAPR.

Messer, K., and Kittler, J. 1999. A region-based image database system using colour and texture. *Pattern Recognition Letters* 20:1323–1330.

Tax, D., and Duin, R. 1998. Outlier detection using classifier instability. In Amin, A.; Dori, D.; Pudil, P.; and Freeman, H., eds., *Advances in Pattern Recognition, Lecture notes in Computer Science*, volume 1451, 593–601. Berlin: Proc. Joint IAPR Int. Workshops SSPR'98 and SPR'98 Sydney, Australia.

Tax, D., and Duin, R. 1999. Support vector domain description. *Pattern Recognition Letters* 20(11-13):1191–1199.

Tax, D.; de Ridder, D.; and Duin, R. 1997. Support vector classifiers: a first look. In *Proceedings ASCI'97*. ASCI.

Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc.

Vapnik, V. 1998. *Statistical Learning Theory*. Wiley.