Domain approximation for condition monitoring

Alexander Ypma, Elzbieta Pekalska and Robert P. W. Duin

Pattern Recognition Group
Faculty of Applied Sciences Delft University of Technology
Lorentzweg 1, 2628 CJ, Delft The Netherlands
email: *ypma@ph.tn.tudelft.nl*

Keywords: machine condition monitoring, domain approximation, support vector methods, nonlinear mapping

Abstract

We propose a novel algorithm for extracting samples from a data set supporting the extremal points in the set. Since the density of the data set is not taken into account, the method could enable adaptation to gradually changing data characteristics (e.g. machine wear). Using knowledge about the clustering structure of the data (obtained with multidimensional scaling techniques), the complexity of the solution can roughly be determined.

1 Introduction

The condition of mechanical machinery can be monitored by measuring the vibration behaviour of its rotating parts [6]. Automatic recognition of machine wear and failure calls for methods that can deal with small sample sizes in high-dimensional spaces, undersampled fault classes and dynamically changing environments. Since normal behaviour will probably be determined by a few calibration measurements of extremal operating conditions (e.g. when putting the machine into practice), an accurate but parsimonious description of the borders of the domain in the feature space indicating normal behaviour is expected to emerge.

Failure detection can be performed by approximating the normal domain and rejecting samples not matching with this description. The rejection threshold should vary with the local resolution in the description [11]. Moreover, machine wear will usually be visible as a gradual shift of the borders of the admissible domain up to a point where a fault can be diagnosed (and this domain will consequently be labelled as faulty).

Conventional classification methods assume well-defined, static classes, while the problem of tailoring the complexity of the solution to a certain problem (e.g. choosing the number of hidden units in a neural network) is error-prone in cases with small sample sizes in high-dimensional spaces (*curse of dimensionality*). Moreover, tackling the former problem by taking a constructive approach can lead to an inherently difficult *subset selection problem [12]*. Recently proposed support vector methods circumvent these problems by editing the data set in a clever way, and expressing the solution in terms of the remaining (supporting) samples.

Inspired by this idea, we develop a novel method for the extraction of objects in a data set that describe the domain of the set in a parsimonious manner, without taking the density of the set into account.

2 Domain approximation with support objects

In Vapnik's support vector classifier, the *optimal separating hyperplane* for two separable classes of data $\{\mathbf{z}_i, i = 1, \ldots, l\}$ labeled by $y_i \in \{\pm 1\}$ is given by [9, 10]

$$f(\mathbf{z}) = sgn(\sum_{i=1}^{l} \alpha_i y_i (\mathbf{z} \cdot \mathbf{z}_i) + b) \qquad (1)$$

The coefficients α_i can be computed by a quadratic minimization procedure, and turn out to be nonzero only for the samples near the classification border (i.e. on the *margin*), the *support vectors* (fig. 1, dark objects).

Using Mercer's theorem, it can be derived [9] that a dot-product in a transformed space (obtained by some nonlinear mapping Φ) can be

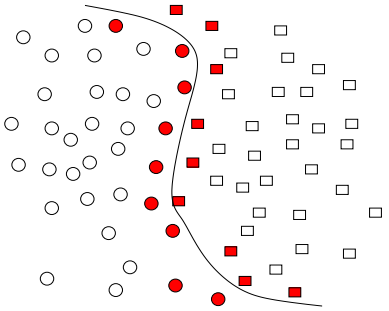

Figure 1: Support vector classifier

expressed as the application of a corresponding kernel $k(\cdot)$ in the original space $(\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})) = k(\mathbf{x}, \mathbf{y})$. By replacing the dot-product in (1) by the more general similarity measure $k(\cdot)$, discriminants of arbitrary complexity can be obtained.

Extending this idea, one could use an arbitrary (non-analytic) similarity measure by looking upon the $m \times m$ distance matrix $D = d(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, \ldots, m$ of a data set of size m as a set of m samples in an m-dimensional space [2]: there exist many situations where it is difficult to design a suitable similarity measure, but distances between objects can be derived intuitively. This approach suffers from the problem that we end up in a situation where we have as many samples as we have features, where the expected generalization error shows a peak. As a remedy, both the number of features or the number of samples can be reduced. When the rows in the distance matrix are considered as samples, and the columns as features, reducing the number of samples (data editing) results in a matrix with for each remaining sample (the *support objects*) the distances to each of the original objects.

2.1 Support objects algorithm

Consider a data set $X = \{\mathbf{x}_i\}, i = 1, \ldots, l$. For domain approximation, every object in the support set $J = \{\mathbf{y}_j \in X\}, j = 1, \ldots, k, k \leq l$ is now given a *receptive field* in $\mathbb{R}^m$ of radius r. A sample $\mathbf{x}_i$ in X lies in the receptive field R_p of a support object $\mathbf{y}_p$ when $p = arg\ min_j\ d(\mathbf{x}_i, \mathbf{y}_j)$ and $d(\mathbf{x}_i, \mathbf{y}_j) \leq r$.

We could choose the set of k support objects J such that corresponding radius $r(J)$ is minimized, while all original objects are captured by some support object's receptive field (fig. 2, black objects), i.e.

$$J = arg\ min_{S_k} r(S_k) \tag{2}$$

where $|S_k| = k$, $S_k \in 2^X$ i.e. S_k a subset of X of length k and the corresponding receptive field radius $r(S_k)$ is given by

$$r(S_k) = max_i d(\mathbf{x}_i, \mathbf{y}_j), \mathbf{x}_i \in R_j, \mathbf{y}_j \in S_k \tag{3}$$

Note that in this method, the problem of optimal subset selection is present. One could propose to use a greedy forward selection method for this purpose, but it is well-known that this will probably not lead to the global optimum.

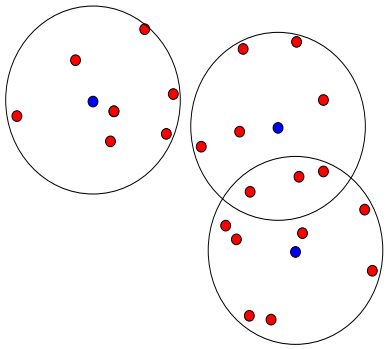

Figure 2: Approximation with support objects

We propose a variant of the *k-means clustering* algorithm, adapted for the domain approximation problem. In the basic approach, different trials are repeated with different random choices for the initial subset. During the algorithm, each support object represents the center of the samples in its receptive field. If a better center can be found within its receptive field (i.e. such that the radius can be decreased, or *relaxed*), swap the former and the latter objects, until the subset can't be improved any more. Ultimately, the best subset over all trials is retained. We refer to this first attempt as the *kcenters* algorithm.

2.2 Improvements of the basic method

The support set size k is also a parameter to be optimized, which makes a combined forward-backward search (e.g. used in branch-and-bound feature selection) difficult. However, we can use the following successive approximation scheme:

```
SUCCAPP(dataset):

k = 0; Jupdated = [];
while not done do
 increase k; Jinit = Jupdated;
 determine point p most remote to its
  receptive field center;
 add p to the initial support set Jinit;
 Jupdated = succ_kcenters(Jinit, dataset);
end;
```

Here, **succ_kcenters(Jinit, dataset)** denotes the *kcenters* algorithm performing just one trial, while initialization is done with the subset **Jinit**. As convergence criterion we choose the relative improvement in radius increasing the support set from size k to $k+1$.

Second, we propose a modification aiming at varying tolerance depending on the *local resolution*. It makes sense to increase the tolerance to new samples for parts of the data with larger interpoint distances (i.e. local resolution), and vice versa. Prior to successive approximation, the distance from each point to its nearest center is normalized by the distance to the point's nearest neighbouring point (not necessarily a support object). Hence, the optimization is done w.r.t. the *characteristic distance* in a receptive field. This has the effect of penalizing outlying subclusters in a receptive field (e.g. occurring because the support set size is smaller than the number of clusters in the data). This approach to a characteristic distance assumes homogeneously distributed samples in a receptive field, which will be true for high-dimensional spaces. After optimization of the global radius (now in terms of each receptive field's characteristic distance or local resolution), the radii in the original metric can be computed by multiplying with the local resolution. We estimate the "unfolding resolution" by taking the distance (in the original metric) of the most remote point in a receptive field (in the original metric) to its nearest neighbour (in the receptive field).

Note that there is a trade-off between complexity of the solution (support set size) and generalization capability (figure 3):

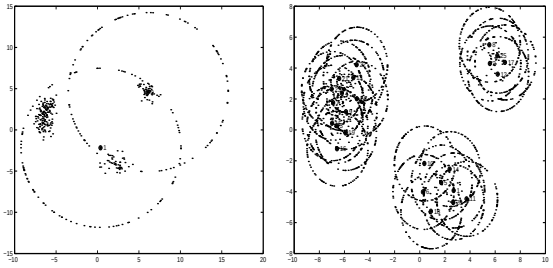

Figure 3: Trade-off between tolerance (left) and complexity (right)

Underfitting occurs when there are a small number of support objects (large receptive fields, yielding too much tolerance), while *overfitting* occurs when nearly each sample in the data set is a support object (small receptive fields, hardly tolerance to new objects). Estimation of an appropriate support set size can be done using the data clustering structure. High-dimensional data sets can be visualized using multidimensional scaling.

3 Multidimensional scaling techniques

Suppose that we consider n vectors in a m-dimensional space: $x_1, x_2, \ldots, x_n$ (stored in matrix $X \in \mathbb{R}^{n \times m}$). Our aim is to find n vectors in a d-dimensional space: $y_1, y_2, \ldots, y_n$ (stored in matrix $Y \in \mathbb{R}^{n \times d}$) in such a way that each vector y_i is the lower-dimensional image of x_i for $i = 1, 2, \ldots, n$. We are interested in a mapping that preserves the inherent structure of the data. We consider three methods for multidimensional scaling.

3.1 Classical scaling

Suppose that we have the coordinate matrix $X \in \mathbb{R}^{n \times m}$. Knowing that distances do not change under translations, we assume that Y has column means equal to 0. Then the square distance matrix D^2 of size n can be computed as follows:

$$D^2 = c\mathbf{1}' + \mathbf{1}c' - 2XX'$$

where $\mathbf{1}$ is a vector of ones, c is a vector with diagonal elements of matrix $B_D = XX'$ and $(\cdot)'$ denotes transposition. It can be shown that $B_D = -\frac{1}{2}JD^2J$, where J is the centering matrix: $J = I - \frac{1}{m}\mathbf{1}\mathbf{1}'$. We know that B_D is symmetric and has non-negative eigenvalues. Then we can find the factorization of B_D by eigendecomposition:

$$B_D = Q\Lambda Q' = (Q\Lambda^{\frac{1}{2}})(Q\Lambda^{\frac{1}{2}})'$$

where Q is an orthogonal matrix, $\Lambda^{\frac{1}{2}}$ is a diagonal matrix with diagonal elements $\lambda_{ii}^{\frac{1}{2}}$. Hence, we have the equation:

$$XX' = B_D = (Q\Lambda^{\frac{1}{2}})(Q\Lambda^{\frac{1}{2}})'$$

It can be proved that X and $Q\Lambda^{\frac{1}{2}}$ differ only by rotation, so we can take:

$$X = Q\Lambda^{\frac{1}{2}}.$$

In the method of *classical scaling*, we are given the square dissimilarity matrix Δ^2 and we want to find a configuration $\{y_i\}_{i=1}^n$ in a (lower dimensional) d-space. The matrix Δ is treated as if it was the square distance matrix D^2 for the configuration $\{y_i\}_{i=1}^n$. Then we compute the matrix B_Δ and perform an eigendecomposition. The result in the d-dimensional space is a matrix $Y \in \mathbb{R}^{n \times d}$ given by:

$$Y = Q_d\Lambda_d^{\frac{1}{2}}$$

where Λ_d stands for the first d eigenvalues greater than zero and Q_d stands for the first d columns of matrix Q.

3.2 Sammon and Kruskal mapping

In *Sammon mapping*, the inter-point distances between vectors in the lower-dimensional space are approximating the corresponding distances in the (original) m-space. We need a criterion for deciding whether one configuration is better than another. For this purpose one considers the error function E (*Sammon's stress*) that measures how well the present configuration of n points in the d-space fits the n points in the m-space:

$$E = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{(\delta_{ij}-d_{ij})^2}{\delta_{ij}}$$

We start from an initial configuration of vectors $\{y_i\}_{i=1}^{n}$ (e.g. randomly chosen or by taking d columns of matrix X with maximum variances) and calculate the stress. Next, we adjust the vectors in order to decrease the stress. We use e.g. a steepest descent algorithm to search for the minimum of the stress function. Having found the configuration in the d-space after i-th iteration, the new set at time $i+1$ is given by:

$$y_{pq}(i+1) = y_{pq}(i) - \alpha\Delta_{pq}(i), \quad 0 \le \alpha \le 1$$

where:

$$\Delta_{pq}(i) = \frac{\delta E(i)}{\delta y_{pq}(i)} \Big/ \left| \left(\frac{\delta^2 E(i)}{\delta y_{pq}(i)^2} \right) \right|$$

The idea of *Kruskal mapping* is similar to Sammon mapping: the inter-point distances in the lower-dimensional space approximate the corresponding distances in the m-dimensional space, but now we suppose that the original distances are transformed by some (monotonic, increasing) function and may be represented by the distances in the d-space.

4 Experiments

Pump vibration was measured with three accelerometers mounted on a submersible pump (fig. 4) that was operating in three states: normal, presence of imbalance and presence of bearing failure [4]. Moreover, the bearing failure was measured at three different operating speeds. An increasing component in the acceleration spectrum at machine running speed is indicative of an imbalance, whereas bearing failures give rise to high-frequency modulations involving bearing geometry related fault frequencies, that can be resolved with envelope detection [6].

4.1 Analysis of vibration data

However, in [13] it was shown that taking a few principal components of the acceleration spectrum already suffices for accurate recognition,

Figure 4: Landustrie DECR 20-8 submersible pump

which indicated a low intrinsic dimensionality. This can be understood from the fact that a vibration signal $\mathbf{x}$ can be modeled as a harmonic series $\mathbf{s}$ plus additive white noise ϵ. Now the principal eigenvectors of the correlation matrix $R_{xx} = E[\mathbf{x}\mathbf{x}^T]$ span the signal subspace [7], which is probably low-dimensional. Due to nonlinearities and crosstalk during machine operation the data might reside in a nonlinear subspace, but since in previous experiments the linear approximation turned out to be already quite descriptive, a 256-points acceleration spectrum was obtained as feature vector and normalized w.r.t. mean and standard-deviation for further processing.

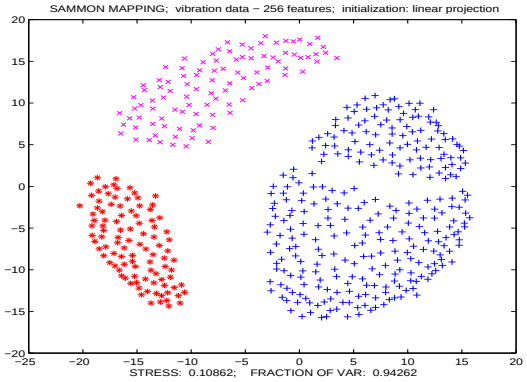

Figure 5: Sammon mapping of vibration data

We use the mapping methods presented above, in order to visualize the low-dimensional image of the original data and have an idea about its inherent high-dimensional structure. The data set consists of 500 vectors with 256 spectral features, including 3 non-overlapping classes. The result of

a Sammon mapping is presented in figure 5 (other methods show similar results). From this figure obtained we observe 3 non-overlapping clusters, so we can conclude that in the original space those clusters also appear.

Let us also consider another data set, obtained by taking the 100 first principal components of as 256-dimensional spectral feature set, now including 5 classes (normal, imbalance and bearing failure at three machine operating speeds). The mapped data obtained with classical scaling is presented in figure 6.

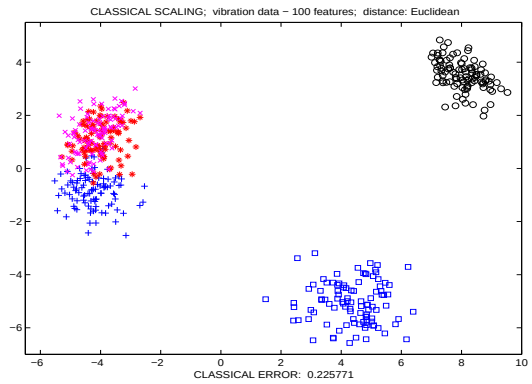

Figure 6: Visualization with classical scaling

For the other methods, the results are comparable. Again we observe 3 non-overlapping clusters. However, one of them now consists of 3 different overlapping subclusters, i.e. bearing failures at slightly different machine running speeds).

4.2 Evaluation of successive approximation algorithm

We investigated the performance of the successive approximation algorithm in domain description and novelty detection. First, we compared the algorithm to the *kcenters* algorithm with random initialization and arbitrary choice of support set size. Using the spectral features data set, we monitored the final radius of the surrounding spheres as a function of the number of spheres. The minimization was done in the original metric (no correction for local resolution), and we tried the *kcenters*-algorithm with both 1 and 5 trials.

In fig. 7 the successive approximation algorithm (the bottom line) can be seen to yield smaller final radius (vertical axis), while the radius is monotonically decreasing with support set size (horizontal axis). The radius obtained with 1 trial random initialization fluctuates randomly around the 5 trial random initialization results. It is clear that the stabilizing effect of

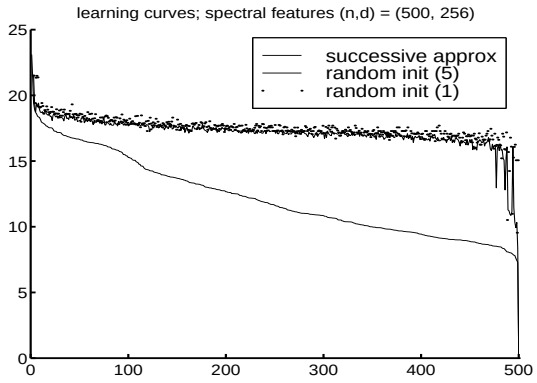

Figure 7: Succ. approximation vs. random initialization

more trials is at the expense of increased computing time, whereas successive approximation needs for each support set size only one run of the algorithm. Moreover, using more spheres leads to significant improvement with successive approximation, whereas the random case reaches a plateau. This suggests that there is a relatively homogeneous distribution of distances in high-dimensional spaces, since the sphere radius can constantly be relaxed. With increasing support set size, the probability that a random initial subset guess is adequate decreases, hence the radius will be less improved using random initialization.

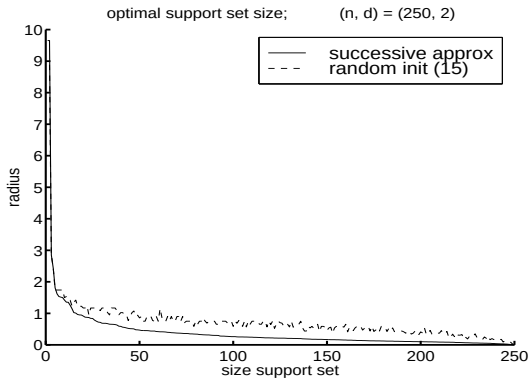

Figure 8: Determination of optimal support set size

Using data consisting of the first two principal components of the previous data set, we repeated the previous experiment (now with 15 trial random initialization). Again (fig. 8), the successive approximation algorithm proved superior, but the difference is much smaller. Due to the three clusters in the data, combined with small dimensionality of the space, a sudden decrease in the radius occurs with three spheres (to a very small value). Moreover, the clusters are already represented adequately by the spheres, resulting in

only marginal improvement in radius using more spheres.

4.3 Generalization to novel data

Next, we checked the capability of algorithm for incorporating gradually changing data. From the 256-dimensional principal components data set, new data sets with gradually changing domain of support were constructed by generating new samples out of the old ones based on distances and directions of the nearest neighbours of samples in the original set. Here, the offset of a new sample from its corresponding original sample is Gaussian distributed with zero mean and with standard deviation s times the mean signed difference between the point under consideration and its nearest neighbours in the original set. For s running from 0 (original set) to 5, different validation sets were contructed. To track generalization, an independent test set of the same origin as the original set was used.

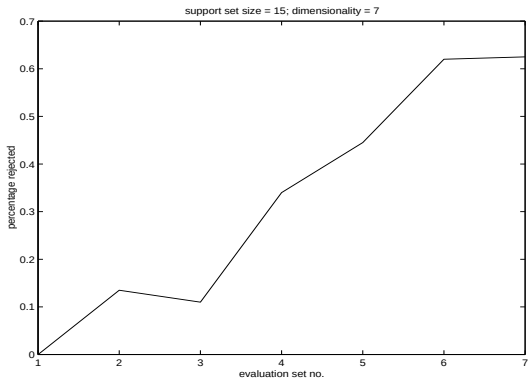

Figure 9: Rejection of novel data with successive approximation algorithm

It was observed that increasing feature size and support set size decreased generalization drastically (i.e. test samples and validation data for moderate values of s were frequently rejected). Results for sensible values of dimensionality (7) and number of spheres (15) are shown in fig. 9. The first set is the train set, the second set the test set, and sets 3 to 7 are validation sets with $s = 1, \ldots, 5$. Since the domain is fitted as tight as possible, a test set already shows some rejections. Then the rejection rate increases with the amount of novelty, up to a point where always a certain fraction of the new samples lies somewhere on the original domain (the distribution of the offset has zero mean).
Results with correction for local resolution were still not satisfactory (fig. 10). Final radii vary with local resolution, but are quite large, probably because of the nonsymmetric scaling back and forth into the original metric.

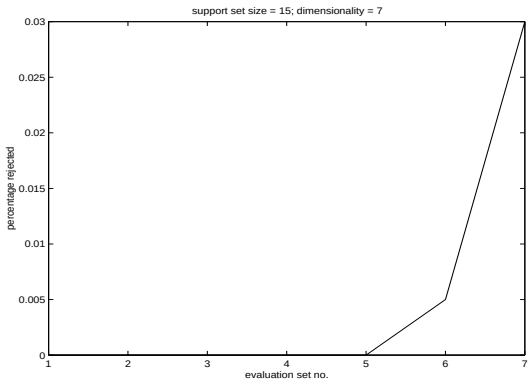

Figure 10: Rejection with correction for local resolution

5 Discussion

We proposed a novel algorithm for the extraction of objects capturing the data set using multidimensional spheres with minimal radius. The algorithm was demonstrated to be an adequate and efficient way for approximating the domain of a dataset. For data with low intrinsic dimensionality, the use of spheres may hamper practical utilization, since tolerances in singular directions are equal to tolerances in relevant directions, leading to frequent acceptation of non-matching data points. For low-dimensional spaces, insight into the clustering structure may aid in choosing the right support set size.

Optimization could be performed w.r.t. the characteristic distances in a receptive field, enabling adaptation of radii to the local resolution. However, the current way of scaling distances back and forth w.r.t. the original metric yields quite large values for final radii.

Moreover, a predefined tolerance specifying the admissible amount of novelty should be incorporated in the method, since the proper rejection threshold will typically vary with each practical application. Automatic complexity control could then be based on this predefined tolerance level.

Since the density of points is never used in the algorithm, but the description is made on the basis of border points, the algorithm bears potential for on-line learning.

6 Acknowledgment

This work was partially supported by the Dutch Foundation for Applied Sciences (STW), project no. DTN-44.3584.

References

[1] I. Borg and P. Groenen. *Modern multidimensional scaling.* Springer-Verlag, New York, 1997.

[2] R. P. W. Duin. Relational discriminant analysis and its large sample size problem. *submitted to ICPR'98*, 1998.

[3] J. B. Kruskal. *Multidimensional scaling and other methods for discovering structure.* Statistical methods for digital computers, Vol. 3 of Mathematical methods for digital computers. John Wiley & Sons, Inc., 1977.

[4] R. Ligteringen, A. Ypma, E. E. E. Frietman, and R. P. W. Duin. Machine diagnostics by neural networks, experimental setup. In *Proc. of ASCI'97*, volume 1, pages 185 – 190. ASCI - TWI-TUD, Delft, The Netherlands, 1997.

[5] B. J. F. Manly. *Multivariate statistical methods, 2nd edition.* Chapman & Hall, 1994.

[6] J. S. Mitchell. *An introduction to machinery analysis and monitoring - 2nd ed.* PennWell Publ. Comp., 1993.

[7] P. Pajunen, J. Joutsensalo, J. Karhunen, and K. Saarinen. Maximum likelihood estimation of equispaced sinusoids in rotating machine fault detection. In *Proc. of the ICSPAT'95, Boston, USA*, pages 1164–1168, 1995.

[8] J. W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18:401–409, 1969.

[9] B. Schölkopf. *Support vector learning.* PhD thesis, TU Berlin, 1997.

[10] V. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, New York, 1995.

[11] A. Ypma and R. P. W. Duin. Novelty detection using self-organizing maps. In *Proc. of ICONIP'97*, pages 1322–1325. Springer-Verlag Singapore, 1997.

[12] A. Ypma and R. P. W. Duin. Using the wavenet for function approximation. In *Proc. of ASCI'97*, volume 1, pages 236 – 240. ASCI - TWI-TUD, Delft, The Netherlands, 1997.

[13] A. Ypma, R. Ligteringen, E. E. E. Frietman, and R. P. W. Duin. Recognition of bearing failures using wavelets and neural networks. In *Proc. of TFTS'97*, pages 69–72. University of Warwick, Coventry (UK), 1997.