Proc. Intern. Conference on Artificial Neural Networks, ICANN07 Porto, Portugal, September, 2007,
Lecture Notes in Computer Science, vol. 4668, Part I, Springer, Berlin, Germany, 2007, pp. 269-278.

Some Properties of the Gaussian Kernel for One Class Learning

Paul F. Evangelista[1], Mark J. Embrechts[2], and Boleslaw K. Szymanski[2]

[1] United States Military Academy, West Point, NY 10996
[2] Rensselaer Polytechnic Institute, Troy, NY 12180

Abstract. This paper proposes a novel approach for directly tuning the gaussian kernel matrix for one class learning. The popular gaussian kernel includes a free parameter, σ, that requires tuning typically performed through validation. The value of this parameter impacts model performance significantly. This paper explores an automated method for tuning this kernel based upon a hill climbing optimization of statistics obtained from the kernel matrix.

1 Introduction

Kernel based pattern recognition has gained much popularity in the machine learning and data mining communities, largely based upon proven performance and broad applicability. Clustering, anomaly detection, classification, regression, and kernel based principal component analysis are just a few of the techniques that use kernels for some type of pattern recognition. The kernel is a critical component of these algorithms - arguably the most important component.

The gaussian kernel is a popular and powerful kernel used in pattern recognition. Theoretical statistical properties of this kernel provide potential approaches for the tuning of this kernel and potential directions for future research. Several heuristics which have been employed with this kernel will be introduced and discussed.

Assume a given data set $\mathbf{X} \in \mathbb{R}^{N \times m}$. $\mathbf{X}$ contains N instances or observations, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$, where $\mathbf{x}_i \in \mathbb{R}^{1 \times m}$. There are m variables to represent each instance i. For every instance there is a label or class, $y_i \in \{-1, +1\}$. Equation 1 illustrates the formula to calculate a gaussian kernel.

$$\kappa(i,j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \tag{1}$$

This kernel requires tuning for the proper value of σ. Manual tuning or brute force search are alternative approaches. An brute force technique could involve stepping through a range of values for σ, perhaps in a gradient ascent optimization, seeking optimal performance of a model with training data. Regardless of the method utilized to find a proper value for σ, this type of model validation is common and necessary when using the gaussian kernel. Although this approach is feasible with supervised learning, it is much more difficult to tune σ for unsupervised learning methods. The one-class SVM, originally proposed by Tax and

Duin [16] and also detailed by Scholkopf et. al. [12], is a good example of an unsupervised learning algorithm where training validation is difficult due to the lack of positive instances. The one-class SVM trains with all negative instances or observations, and based upon the estimated support of the negative instances, new observations are classified as either inside the support (predicted negative instance) or outside of the support (predicted positive instance). It is quite possible, however, that there are very few or no positive instances available. This poses a validation problem. Both Tax and Duin [16] and Scholkopf et. al. [12] state that tuning the gaussian kernel for the one-class SVM is an open problem.

2 Recent Work

Tax and Duin [16] and Scholkopf et. al. [12] performed the groundbreaking work with the one-class SVM. Stolfo and Wang [15] successfully apply the one-class SVM to the intrusion data set that we use in this paper. Chen et. al. [3] uses the one-class SVM for image retrieval. Shawe-Taylor and Cristianini [14] provide the theoretical background for this method.

Tax and Duin [17] discuss selection of the σ parameter for the one-class SVM, selecting σ based upon a predefined error rate and desired fraction of support vectors. This requires solving the one-class SVM for various values of σ and the parameter C, referred to in this paper as $1/\nu N = C$. This method relies on the fraction of support vectors as an indicator of future generalization. Tuning of the parameter C does not significantly impact the ordering of the decision values created by the one-class SVM; tuning of σ influences the shape of the decision function and profoundly impacts the ordering of the decision values. When seeking to maximize the area under the ROC curve (AUC), the ordering of the decision values is all that matters. The technique in this paper is very different from the one which is mentioned in [17]. We use the kernel matrix directly, therefore the one-class SVM does not need to be solved for each change in value of σ. Furthermore, tuning the kernel matrix directly requires the tuning of only one parameter.

3 The One-Class SVM

The one-class SVM is an anomaly detection model solved by the following optimization problem:

$$\min_{R\in\mathbb{R},\zeta\in\mathbb{R}^N,c\in F} \qquad R^2 + \frac{1}{vN}\sum_i \zeta_i \tag{2}$$

$$\text{subject to} \qquad \| \Phi(\mathbf{x}_i) - c \|^2 \leq R^2 + \zeta_i \text{ and } \zeta_i \geq 0 \text{ for } i = 1, ..., N$$

The lagrangian dual is shown below in equation 3.

$$\max_{\alpha} \sum_{i} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

$$\text{subject to} \qquad 0 \leq \alpha_i \leq \frac{1}{vN} \text{ and } \sum_{i} \alpha_i = 1$$

Scholkopf et. al. point out the following reduction of the dual formulation when modeling with gaussian kernels:

$$\min_{\alpha} \sum_{i,j} \alpha_i \alpha_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

$$\text{subject to} \qquad 0 \leq \alpha_i \leq \frac{1}{vN} \text{ and } \sum_{i} \alpha_i = 1$$

This reduction occurs since we know that $\kappa(\mathbf{x}_i, \mathbf{x}_i) = 1$ and $\sum_i \alpha_i = 1$. Equation 4 can also be written as $\min \boldsymbol{\alpha}' \mathbf{K} \boldsymbol{\alpha}$. Shawe-Taylor and Cristianini [14] explain that $\boldsymbol{\alpha}' \mathbf{K} \boldsymbol{\alpha}$ is the weight vector norm, and controlling the size of this value improves the statistical stability, or regularization of the model.

All training examples with $\alpha_i > 0$ are support vectors, and the examples which also have a strict inequality of $\alpha_i < \frac{1}{vN}$ are considered non-bounded support vectors.

In order to classify a new test instance, $\mathbf{v}$, we would evaluate the following decision function:

$$f(v) = \kappa(\mathbf{v}, \mathbf{v}) - 2 \sum_{j} \alpha_j \kappa(\mathbf{v}, \mathbf{x}_j) + \sum_{j,k} \alpha_k \alpha_j \kappa(\mathbf{x}_k, \mathbf{x}_j) - R^2$$

Before evaluating for a new point, R^2 must be found. This is done by finding a non-bounded support vector training example and setting the decision function equal to 0 as detailed by Bennett and Campbell [1]. If the decision function is negative for a new test instance, this indicates a negative or healthy prediction. A positive evaluation is an unhealthy or positive prediction, and the magnitude of the decision function in either direction is an indication of the model's confidence. It is useful to visualize the decision function of the one class SVM with a small two dimensional dataset, shown in figure 1. The plots corresponding to a small σ value clearly illustrate that overtraining is occurring, with the decision function wrapped tightly around the data points. Large values of sigma simply draw an oval around the points without defining the shape or pattern.

4 Method

The behavior of the gaussian kernel is apparent when examined in detail. The values lie within the (0,1) interval. A gaussian kernel matrix will have ones along the diagonal (because $\| \mathbf{x}_i - \mathbf{x}_i \| = 0$). Additionally, a value too small for σ will force the matrix entries towards 0, and a value too large for σ will force matrix

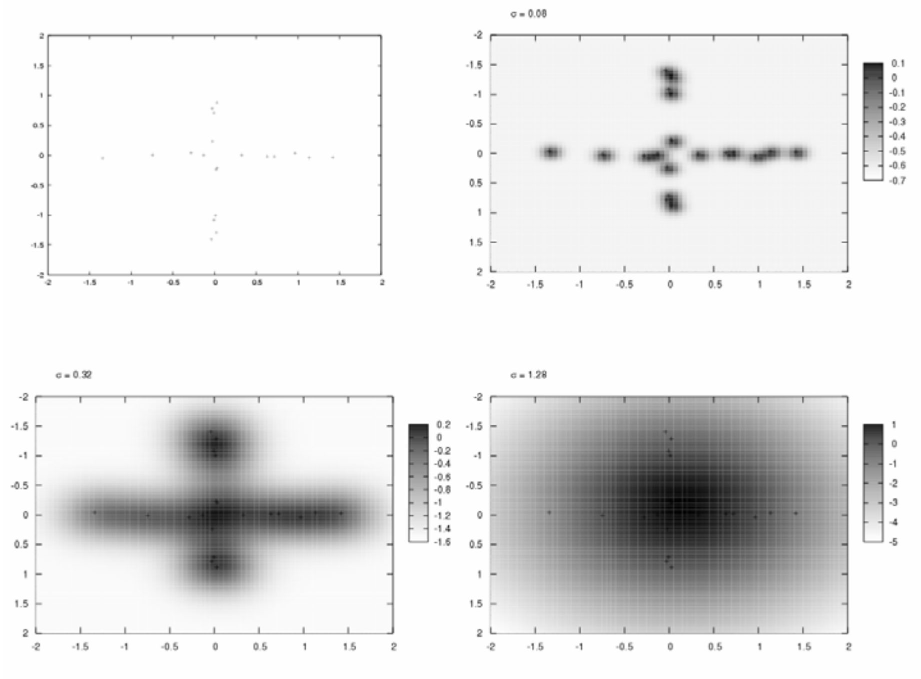

Fig. 1. Visualization of one class SVM for three different values for σ. The source data, named the cross data due to its pattern, is shown in the top left.

entries towards 1. There is also a property of all kernels, which we will refer to as the fundamental premise of pattern recognition, which simply indicates that for good models, the following relationship consistently holds true:

$$(\kappa(i,j)|(y_i = y_j)) > (\kappa(i,j)|(y_i \neq y_j)) \tag{5}$$

Consistent performance and generalization of the fundamental premise of pattern recognition is the goal of all kernel based learning. Given a supervised dataset, a training and validation split of the data is often used to tune a model which seems to consistently observe the fundamental premise. However, in an unsupervised learning scenario positive labeled data is limited or non-existent, and furthermore, models such as the one-class SVM have no use for positive labeled data in the training data.

A first approach towards tuning a kernel matrix for the one-class SVM might lead one to believe that the matrix should take on very high values, indicating that all of the kernel entries for the training data is of one class and therefore should take on high values. Although this approach would first seem to be consistent with the fundamental premise in equation 5, this approach would be misguided. The magnitude of the values within the kernel matrix is not an important attribute. The important attribute is actually the spread or the variance of the entries in the kernel matrix. At first this may seem to be anomalous with equation 5, however a closer examination of the statistics of a kernel matrix illustrates why the variance of the kernel matrix is such a critical element in model performance.

Shawe-Taylor and Cristianini point out that small values of σ allow classifiers to fit any set of labels, and therefore overfitting occurs [14]. They also state that large values for σ impede a classifiers ability to detect non-trivial patterns because the kernel gradually reduces to a constant function. The following mathematical discussion supports these comments for the one-class SVM. Considering again the one-class SVM optimization problem, posed as $\min \boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha}$ assuming the a gaussian kernel, if the sigma parameter is too small and $\kappa(i,j) \rightarrow 0$, the optimal solution is $\alpha_i = 1/N$. Equation 4, the objective function, will equate to $1/N$ (since $\sum_i (1/N)^2 = 1/N$). If the sigma parameter is too large and $\kappa(i,j) \rightarrow 1$, the optimal solution is the entire feasible set for $\boldsymbol{\alpha}$. Given these values for the variables and parameters, the objective function will now equate to 1. The brief derivation for the case when $\kappa(i,j) \rightarrow 1$ follows:

$$\alpha'\mathbf{K}\alpha = \sum_{i=1}^{N} \alpha_i^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} 2\alpha_i \alpha_j \kappa(i,j)$$

$$= \sum_{i=1}^{N} \alpha_i^2 + \sum_{i=1}^{N} \alpha_i \sum_{j=1...i-1,i+1...N} \alpha_j$$

$$= \sum_{i=1}^{N} \alpha_i^2 + \sum_{i=1}^{N} \alpha_i(1-\alpha_i) = \sum_{i=1}^{N} \alpha_i^2 + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i^2 = \sum_{i=1}^{N} \alpha_i = 1$$

The objective function bounds are $(1/N, 1)$, and the choice of σ greatly influences where in this bound the solution lies.

4.1 The Coefficient of Variance

In order to find the best value for σ, a heuristic is employed. This heuristic takes advantage of the behavior of the one-class SVM when using the gaussian kernel. The mean and the variance of the non-diagonal kernel entries, $\kappa(i,j)|i \neq j$, play a crucial role in this heuristic. We will refer to the mean as $\bar{\kappa}$ and the variance as s^2. For any kernel matrix where $i, j \in \{1, ..., N\}$, there are $N^2 - N$ off diagonal kernel entries. Furthermore, since all kernel matrices are symmetrical, either the upper or lower diagonal entries only need to be stored in memory, of which there are $l = (N^2 - N)/2$. From here forward, the number of unique off diagonal kernel entries will be referred to as l.

It is first necessary to understand the statistic used in this heuristic, the coefficient of variance. The coefficient of variance is commonly referred to as 100 times the sample standard deviation divided by its mean, or $\frac{100s}{\bar{x}}$. This statistic describes the relative spread of a distribution, regardless of the unit of scale. Due to the scale and behavior of $\bar{\kappa}$ and s, this coefficient of variance monotonically increases for gaussian kernels as σ ranges from 0 to ∞. Using the sample variance rather than the standard deviation, different behavior occurs. The monotonic increase of the coefficient of variance occurs because when σ is small, $s > \bar{\kappa}$; however, as σ increases, there is a cross-over point and then $s < \bar{\kappa}$. However, the variance of $\kappa(i,j)|i \neq j$ is always smaller than the mean of $\kappa(i,j)|i \neq j$. This property is what makes the variance of a kernel matrix such a critical component for the direct tuning method. The proof follows. For the sake of notation simplicity, $x_k \in (0,1)$, $k \in [l]$, will represent off diagonal kernel entries, that is entries $\kappa(i,j)|i \neq j$.

$$VAR(x_k) \leq \bar{x} \implies \frac{\sum_k x_k^2 - 2\bar{x}\sum_k x_k + l\bar{x}^2}{l-1} \leq \frac{(l-1)\bar{x}}{l-1}$$

$$\implies \frac{l\bar{x}^2 - 2l\bar{x}^2 - (l-1)\bar{x} + \sum_k {x_k}^2}{l-1} \leq 0$$

$$\implies \sum_k x_k^2 - l\bar{x}^2 - (l-1)\bar{x} \leq 0 \implies \sum_k x_k^2 - \sum_k x_k + \frac{\sum_k x_k}{l}(1 - \sum_k x_k) \leq 0$$

$$\implies l\sum_k x_k^2 - \sum_k x_k(l-1) - (\sum_k x_k)^2 \leq 0$$

$$\implies \sum_k x_k^2 - (\sum_k x_k)^2 + (l-1)\sum_k x_k^2 - (l-1)\sum_k x_k \leq 0$$

$$\implies \underbrace{\sum_k x_k^2 - (\sum_k x_k)^2}_{\text{always } \leq 0} + (l-1) \underbrace{\left(\sum_k x_k^2 - \sum_k x_k\right)}_{\text{always } \leq 0 \text{ for } 0 \leq x_k \leq 1} \leq 0$$

The fact that the variance of $\kappa(i,j)|i \neq j$ is always smaller than the mean of $\kappa(i,j)|i \neq j$ indicates that $s^2/\bar{\kappa}$ is a fraction. Furthermore, as σ ranges from 0 to ∞, this fraction ascends to a global max. In order to protect against division by zero and roundoff error, a small value, ϵ, can be added to the denominator. The results in the following objective function for optimization which can be solved quickly with a gradient ascent algorithm. Solving the optimization problem in equation 6 leads to the best choice for σ.

$$\max_{\sigma} \frac{s^2}{\bar{\kappa} + \epsilon} \ \forall (i \neq j) \tag{6}$$

such that

$$\kappa(i,j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \ , \ \bar{\kappa} = \frac{\sum_{i=1}^{N}\sum_{j=i+1}^{N}\kappa(i,j)}{l} \ , \ s^2 = \frac{\sum_{i=1}^{N}\sum_{j=i+1}^{N}(\kappa(i,j) - \bar{\kappa})^2}{l-1}$$

Figure 2 illustrates the impact of sigma on the kernel matrix. The dataset used to create figure 2 was the ionosphere data which is available from the UCI repository. These data are all of the negative class, and it is evident that both the kernel element values and the dispersion of these values changes as σ changes. The optimal value for σ for this dataset is 1, and the color visualization of the kernel matrix when $\sigma = 1$ clearly indicates that there is dispersion in the kernel matrix. However, it is also noticeable that the central tendency of the kernel entries for this optimal σ is a small value, closer to zero than one. This is consistent for all of the datasets. This is why it is important to use a metric that detects relative dispersion and is not biased by the magnitude of the kernel entries.

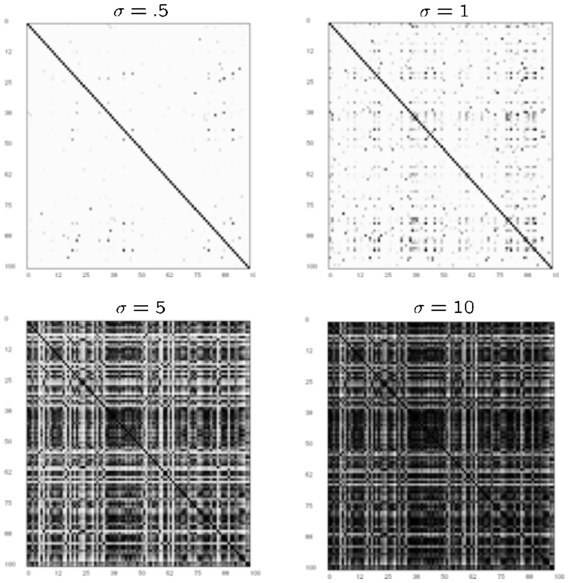

Fig. 2. Color visualization of a kernel matrix with various values for σ. This visualization involves 100 observations where kernel entries close to 0 are light; darker entries represent values closer to unity.

4.2 Why Direct Tuning of the Kernel Matrix Works

As mentioned previously, it is desirable to minimize $\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha}$. $\boldsymbol{\alpha}$ is sparse when there are few support vectors, and this sparseness is typically desirable to minimize complexity and improve regularization. However, meaningless sparse solutions for α can occur as all $\kappa(i, j) \to 0$. Meaningful sparse solutions for $\boldsymbol{\alpha}$ occur when the kernel matrix entries are not concentrated either towards 0 or 1, but are showing good dispersion and there is payoff in the optimization to select the few instances which clearly define the margin of the density approximated by the one-class SVM. Although the variance, s^2, is a good indicator of the dispersion in the kernel matrix, it is biased towards a larger σ since variance is affected by unit of scale or magnitude. $s^2/(\bar{\kappa}+\epsilon)$ is robust against unit of scale. This statistic illustrates the dispersion of a distribution regardless of scale. For the one class SVM, the optimal value for σ and maximum value for $s^2/(\bar{\kappa}+\epsilon)$ typically occurs as σ increases from 0 and the kernel entries first begin to illustrate dispersion. When there is maximal dispersion in the kernel matrix, the values assigned to $\boldsymbol{\alpha}$ will reflect the behavior and relationships of observations, which is the purpose of the statistical learning. Maximal dispersion of the kernel matrix also supports the fundamental premise of pattern recognition. When σ is not properly tuned, the values assigned to $\boldsymbol{\alpha}$ will be erroneously based upon the behavior of the optimization problem or the optimization algorithm used to solve the problem.

5 Experimental Results

In order to evaluate the performance of the direct tuning heuristic, several experiments were conducted. The data included three benchmark sets: banana, chessboard, and ionosphere (from UCI repository). Additionally, two computer intrusion datasets named Schonlau and Sick, after their respective creators, are also examined. The Schonlau data involves determining authenticity of a user based on UNIX commands. This data was originally discussed in ([4, 5, 13]) and the actual data used in this paper was also discussed in ([6–8]). The Sick data was originally examined in [11]. The parameter ν is set to .5 for every experiment.

Dataset	**dimensions**	**positive**	**negative**	**comment**
Banana	2	50	50	see ([9])
Chessboard	2	50	50	www.cs.wisc.edu/ ∼olvi/data/check1.txt
Ionosphere	34	126	225	UCI Repository
Schonlau	26	231	4769	www.schonlau.net
Sick	137	250	5143	see ([11])

For each dataset, one half of the negative class was used for training the one-class SVM and the test data comprised of the other half of the negative class and all members of the positive class. The same split remained consistent for all experiments for the purposes of control and consistency. The performance measure utilized is the area under the receiver operating characteristic curve (AUC).

The three benchmark datasets clearly illustrated optimal performance when $s^2/(\bar{\kappa}+\epsilon)$ is optimal. For each of the benchmark solutions, a gradient ascent algorithm was tested and converged on the optimal σ within 4-6 iterations de-

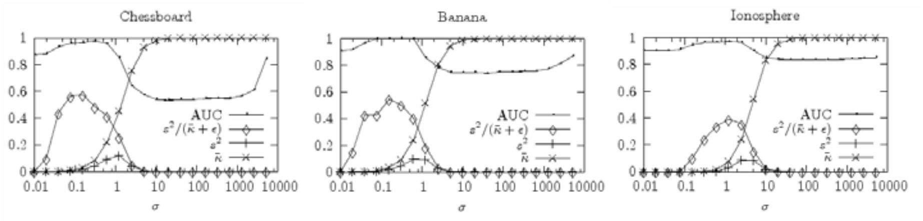

Fig. 3. Experimental results for three benchmark datasets.

pending on the initial values and dataset. An initial value of 1 for σ seemed to work well since most of the optimal values for σ ranged between .1 and 10.

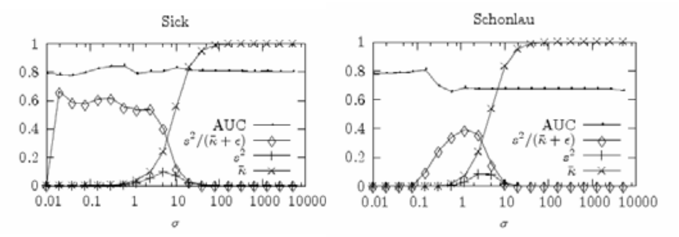

Fig. 4. Experimental results for two computer intrusion datasets.

The two computer intrusion datasets did not clearly indicate this same type of ideal performance. The Schonlau dataset performed the worst by far, and the Sick dataset did indicate a region of good performance. The value of σ seemed indifferent for the Sick data.

6 Conclusions

Direct tuning of the gaussian kernel matrix is a novel and promising approach for the necessary tuning of the powerful gaussian kernel. The heuristic proposed is grounded and supported with underlying theory. The significance of tuning the gaussian kernel for unsupervised learning applies directly to ensemble methods. Techniques such as bagging [2], random subspace selection [10], and fuzzy aggregation techniques [6, 7] can be employed for unsupervised learning with the gaussian kernel.

Future research could involve automated tuning approaches for supervised learning, however this approach will clearly have to outperform the traditional technique of validation. Direct tuning for supervised learning would be faster, but first it needs to be shown that it is also just as accurate as validation.

References

1. Kristin P. Bennett and Colin Campbell. Support Vector Machines: Hype or Hallelujah. *SIGKDD Explorations*, 2(2), 2001.
2. Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
3. Yunqiang Chen, Xiang Zhou, and Thomas S. Huang. One-Class SVM for Learning in Image Retrieval. Thessaloniki, Greece, 2001. Proceedings of IEEE International Conference on Image Processing.
4. William DuMouchel, Wen Hua Ju, Alan F. Karr, Matthius Schonlau, Martin Theus, and Yehuda Vardi. Computer Intrusion: Detecting Masquerades. *Statistical Science*, 16(1):1–17, 2001.
5. William DuMouchel and Matthius Schonlau. A Fast Computer Intrusion Detection Algorithm Based on Hypothesis Testing of Command Transition Probabilities. pages 189–193. The Fourth International Conference of Knowledge Discovery and Data Mining, August 1998.
6. Paul F. Evangelista, Piero Bonissone, Mark J. Embrechts, and Boleslaw K. Szymanski. Fuzzy ROC Curves for the One Class SVM: Application to Intrusion Detection. In *Proceedings of the International Joint Conference on Neural Networks*, Montreal, Canada, August 2005.
7. Paul F. Evangelista, Piero Bonissone, Mark J. Embrechts, and Boleslaw K. Szymanski. Unsupervised Fuzzy Ensembles and Their Use in Intrusion Detection. In *Proceedings of the European Symposium on Artificial Neural Networks*, Bruges, Belgium, April 2005.
8. Paul F. Evangelista, Mark J. Embrechts, and Boleslaw K. Szymanski. Computer Intrusion Detection Through Predictive Models. pages 489–494, St. Louis, Missouri, November 2004. Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining and Complex Systems.
9. D. Frossyniotis, A. Likas, and A. Stafylopatis. A Clustering Method Based on Boosting. *Pattern Recognition Letters*, 25:641–654, 2004.
10. Tin Kam Ho. The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
11. Alexander Hofmann, Timo Horeis, and Bernhard Sick. Feature Selection for Intrusion Detection: An Evolutionary Wrapper Approach. Budapest, Hungary, July 2004. International Joint Conference on Neural Networks.
12. Bernhard Schölkopf, John C. Platt, John Shawe Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the Support of a High Dimensional Distribution. *Neural Computation*, 13:1443–1471, 2001.
13. Matthius Schonlau and Martin Theus. Detecting Masquerades in Intrusion Detection Based on Unpopular Commands. *Information Processing Letters*, 76(1-2):33–38, 2000.
14. John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
15. Salvatore Stolfo and Ke Wang. One Class Training for Masquerade Detection. Florida, 19 November 2003. 3rd IEEE Conference Data Mining Workshop on Data Mining for Computer Security.
16. David M.J. Tax and Robert P.W. Duin. Support Vector Domain Description. *Pattern Recognition Letters*, 20:1191–1199, 1999.
17. David M.J. Tax and Robert P.W. Duin. Support Vector Data Description. *Machine Learning*, 54:45–66, 2004.