MULTI-CATEGORY CLASSIFICATION BY KERNEL BASED NONLINEAR SUBSPACE METHOD

Eisaku Maeda (maeda@eye.brl.ntt.co.jp) and Hiroshi Murase

NTT Basic Research Laboratories, Atsugi-shi, 243–0198, JAPAN

ABSTRACT

The Kernel based Nonlinear Subspace (KNS) method is proposed for multi-class pattern classification. This method consists of the nonlinear transformation of feature spaces defined by kernel functions and subspace method in transformed high-dimensional spaces. The Support Vector Machine, a nonlinear classifier based on a kernel function technique, shows excellent classification performance, however, its computational cost increases exponentially with the number of patterns and classes. The linear subspace method is a technique for multi-category classification, but it fails when the pattern distribution has nonlinear characteristics or the feature space dimension is low compared to the number of classes. The proposed method combines the advantages of both techniques and realizes multi-class nonlinear classifiers with better performance in less computational time. In this paper, we show that a nonlinear subspace method can be formulated by nonlinear transformations defined through kernel functions and that its performance is better than that obtained by conventional methods.

1. INTRODUCTION

There are two techniques for tackling pattern classification problems, the parametric and non-parametric approaches. As the form for the density functions of the patterns in the feature space are unknown in many practical problems, the non-parametric technique is usually more practical. The linear discriminant function is a widely investigated non-parametric classifier and it is simple and robust. When the true boundary between classes is complex, however, this function is fundamentally incapable of performing well and other classifiers described by nonlinear functions are needed. The Support Vector Machine (SVM) is a nonlinear classifier; it was recently studied and shown to offer good performance [9][1]. Its characteristics are that patterns are mapped to an extremely high-dimensional space by the nonlinear transformation defined by kernel functions and the optimization problems of classifiers result in quadratic programming problems. The computational cost for SVM optimization, however, increases exponentially with the number of training patterns (see Figure 4) !%Many more training patterns are needed for good generalization performance than required by linear classifiers whose capacity is $2d+1$ [2], because optimal SVMs are obtained by optimizing the linear function in the transformed high-dimensional space. Moreover, for multi-class problems multiple SVMs should be optimized. By contrast, the subspace method [10][5] can be used to design multi-category classifiers in much less computation time. As the method allocates the subspace for each class which well characterizes the class, it is more desirable for the feature space dimension to be as high as possible. Therefore, by combining the subspace method and the nonlinear transformation to high dimensional space defined by kernel functions, we can obtain a nonlinear classifier for multi-category classification that offers better performance in less computation time.

In this paper, we formulate the proposed nonlinear subspace method using kernel functions and provide experimental results on its performance. In the following, The propose method is called the "Kernel based Nonlinear Subspace method" hereafter, and abbreviate it as KNS method. Recently, a similar idea was reported independently [4][8].

2. NONLINEAR SUBSPACE METHOD

The kernel function $k(\mathbf{x}, \mathbf{y})$ is defined as

$$k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^t \phi(\mathbf{y}) = \sum_{i=1}^{d_\phi} \phi_i(\mathbf{x}) \phi_i(\mathbf{y}), \tag{1}$$

where ϕ is a nonlinear function and d_ϕ is the dimension of $\phi(\mathbf{x})$. In the SVM, the kernel function is not defined as a function of $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$, but of $\mathbf{x}$ and $\mathbf{y}$ as follows.

$$k(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^t \mathbf{y})^p \tag{2}$$

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2p^2}\right) \tag{3}$$

where, p is a constant. As shown in the SVM, the nonlinear transformations defined by the kernel functions give rise to two notable characteristics [9][1]. First, the dimension of the space transformed by ϕ is generally extremely high. As all the axes of this high dimension space are linearly independent, it is expected that there is a better classification hyper-plane than that in the original feature space. Second, the optimization can be conducted by means of pattern manipulation in the original space without knowing ϕ. This is an advantage from the viewpoint of computational costs.

2.1. Projection to nonlinear class-subspace

When nonlinear function ϕ is unknown, the principal component vector $\mathbf{v}$ in a nonlinear space and the transformed vector of $\mathbf{z}$, $\phi(\mathbf{z})$, are unknown. By employing the kernel function technique, we can know the projection of $\phi(\mathbf{z})$ to $\mathbf{v}$, $\mathbf{v}^t \phi(\mathbf{z})$ [6]. Using sets of d dimensional patterns $\mathbf{x}_1, \ldots, \mathbf{x}_n$, $\mathbf{y}_1, \ldots, \mathbf{y}_m$, $\mathbf{z}_1, \ldots, \mathbf{z}_l$ and a function $\phi : \mathcal{R}^d \mapsto \mathcal{R}^{d_\phi}$, define pattern matrices $\mathbf{X}_\phi (\in R^{d_\phi \times n})$, $\mathbf{Y}_\phi (\in R^{d_\phi \times n})$, $\mathbf{Z}_\phi (\in R^{d_\phi \times n})$ like

$$\mathbf{X}_\phi = (\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)), \tag{4}$$

and define ϕ_M as the mean pattern vector of $\phi(\mathbf{x}_i)$ $(i = 1, \ldots, n)$. When a function $\tilde{\phi}$ is defined as

$\tilde{\phi} : \mathbf{x} \mapsto \phi(\mathbf{x}) - \phi_M(\mathcal{R}^d \mapsto \mathcal{R}^{d_\phi})$, $\mathbf{Y}_{\tilde{\phi}}$, $\mathbf{Z}_{\tilde{\phi}}$ can be written as

$$\mathbf{Y}_{\tilde{\phi}} = \mathbf{Y}_\phi - \frac{1}{n}\mathbf{X}_\phi \mathbf{1}_{nm} \quad (\in R^{d_\phi \times m}) \tag{5}$$

$$\mathbf{Z}_{\tilde{\phi}} = \mathbf{Z}_\phi - \frac{1}{n}\mathbf{X}_\phi \mathbf{1}_{nl} \quad (\in R^{d_\phi \times l}), \tag{6}$$

where $\mathbf{1}_{nn'}$ is an (n, n') matrix, all of whose elements are 1. Define a kernel matrix $K(\mathbf{Y}, \mathbf{Z})$ for a (d, m) matrix, $\mathbf{Y}$, and a (d, l) matrix, $\mathbf{Z}$, as the (m, l) matrix of which the (i, j) element is $\phi(\mathbf{y}_i)^t \phi(\mathbf{z}_j) (= k(\mathbf{y}_i, \mathbf{z}_j))$. It follows that $K(\mathbf{Y}, \mathbf{Z})$ can be written as

$$K(\mathbf{Y}, \mathbf{Z}) = \mathbf{Y}_\phi^t \mathbf{Z}_\phi. \tag{7}$$

If we define $G_X(\mathbf{Y}, \mathbf{Z})$ as the (m, l) matrix of which (i, j) element is $\tilde{\phi}(\mathbf{y}_i)^t \tilde{\phi}(\mathbf{z}_j)$, we obtain the following from (5), (6), (7),

$$\begin{aligned} G_X(\mathbf{Y}, \mathbf{Z}) &= \mathbf{Y}_{\tilde{\phi}}^t \mathbf{Z}_{\tilde{\phi}} \\ &= K(\mathbf{Y}, \mathbf{Z}) - \frac{1}{n} K(\mathbf{Y}, \mathbf{X}) \mathbf{1}_{nl} - \frac{1}{n} \mathbf{1}_{mn} K(\mathbf{X}, \mathbf{Z}) \\ &\quad + \frac{1}{n^2} \mathbf{1}_{mn} K(\mathbf{X}, \mathbf{X}) \mathbf{1}_{nl}. \end{aligned} \tag{8}$$

If we write the eigenvalues of $\mathbf{X}_{\tilde{\phi}}^t \mathbf{X}_{\tilde{\phi}}$ as $\lambda_i (\lambda_1 \geq \ldots \geq \lambda_n)$, $r (= \mathrm{rank}(\mathbf{X}_{\tilde{\phi}}))$ of them are positive eigenvalues, and are identical to the positive eigenvalues of $\mathbf{X}_{\tilde{\phi}} \mathbf{X}_{\tilde{\phi}}^t$. If we write normalized orthogonal eigenvectors of $\mathbf{X}_{\tilde{\phi}}^t \mathbf{X}_{\tilde{\phi}}$ and $\mathbf{X}_{\tilde{\phi}} \mathbf{X}_{\tilde{\phi}}^t$ corresponding to an eigenvalue λ_i as $\mathbf{u}_i$ and $\mathbf{v}_i$ respectively, and define matrices $\mathbf{U}$, $\mathbf{V}$ and Λ as

$$\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_r) \quad (\in \mathcal{R}^{n \times r}) \tag{9}$$

$$\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_r) \quad (\in \mathcal{R}^{d_\phi \times r}) \tag{10}$$

$$\Lambda = \begin{bmatrix} \sqrt{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda_r} \end{bmatrix} \quad (\in \mathcal{R}^{r \times r}), \tag{11}$$

the relation

$$\mathbf{X}_{\tilde{\phi}} = \mathbf{V} \Lambda \mathbf{U}^t \tag{12}$$

holds. As $\mathbf{v}_i$ represents the i-th principal vector of the pattern set $(\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n))$ and the relation about a singular decomposition

$$\mathbf{v}_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{X}_{\tilde{\phi}} \mathbf{u}_i \tag{13}$$

holds, the projection of a vector $\tilde{\phi}(\mathbf{z})$to the $\mathbf{v}_i$ can be written as

$$\mathbf{v}_i^t \tilde{\phi}(\mathbf{z}) = \frac{1}{\sqrt{\lambda_i}} \mathbf{u}_i^t \mathbf{X}_{\tilde{\phi}}^t \tilde{\phi}(\mathbf{z}) = \frac{1}{\sqrt{\lambda_i}} \mathbf{u}_i^t G_X(\mathbf{X}, \mathbf{z}). \tag{14}$$

As this projection $\mathbf{v}_i^t \tilde{\phi}(\mathbf{z}_j)$ is an (i, j) element of $\mathbf{V}^t \mathbf{Z}_{\tilde{\phi}}$, we can obtain

$$\mathbf{V}^t \mathbf{Z}_{\tilde{\phi}} = \Lambda^{-1} \mathbf{U}^t G_X(\mathbf{X}, \mathbf{Z}). \tag{15}$$

Λ and $\mathbf{U}$ can be obtained as eigenvalues and eigenvectors, respectively, of the matrix $\mathbf{X}_{\tilde{\phi}}^t \mathbf{X}_{\tilde{\phi}}$, i.e. $G_X(\mathbf{X}, \mathbf{X})$. This result means we can obtain the projection of any pattern $\mathbf{z}$ to $\mathbf{v}_i$ if only K defined by (7) is given.

2.2. Classification criteria

In terms of classification criterion, there are two major approaches to subspace classification: the CLAFIC method [10][5] and the projection distance method [3]. The KNS method can be developed from either approaches. Here, the latter case is shown. Define $\lambda_i \quad (i = 1, \ldots, d')$ as the d' largest eigenvalues of $G_X(\mathbf{X}, \mathbf{X}) = \mathbf{X}_{\tilde{\phi}}^t \mathbf{X}_{\tilde{\phi}}$, $\mathbf{u}_i$ and $\mathbf{v}_i$ as the normalized orthogonal eigenvectors of $\mathbf{X}_{\tilde{\phi}}^t \mathbf{X}_{\tilde{\phi}}$ and $\mathbf{X}_{\tilde{\phi}} \mathbf{X}_{\tilde{\phi}}^t$ corresponding to the eigenvalue λ_i, and define $\mathbf{U}_{d'}$, $\mathbf{V}_{d'}$ and $\Lambda_{d'}$ using d' $\mathbf{u}_i$s, $\mathbf{v}_i$s and λ_is as in (9), (10) and (11). Define $D(\mathbf{z})$ as the projection distance of a pattern $\phi(\mathbf{z})$ to a class subspace spanned by $\mathbf{v}_i \quad (i = 1, \ldots, d')$. The squared projection distance, $D^2(\mathbf{z})$, can be obtained from the following equation,

$$D^2(\mathbf{z}) = \tilde{\phi}^t(\mathbf{z}) \tilde{\phi}(\mathbf{z}) - \sum_{i=1}^{d'} \left(\mathbf{v}_i^t \tilde{\phi}(\mathbf{z}) \right)^2 \tag{16}$$

$$= G_X(\mathbf{z}, \mathbf{z}) - \| \mathbf{V}_{d'}^t \tilde{\phi}(\mathbf{z}) \|^2 \tag{17}$$

$$= G_X(\mathbf{z}, \mathbf{z}) - \| \Lambda_{d'}^{-1} \mathbf{U}_{d'}^t G_X(\mathbf{X}, \mathbf{z}) \|^2. \tag{18}$$

$D^2(\mathbf{z})$ is calculated for each class subspace and $\mathbf{z}$ is classified in the class in which D^2 is minimum. This result shows that KNS can be applied if only the kernel function is given, it does not need to know the nonlinear function ϕ explicitly.

3. RESULTS AND DISCUSSION

We compared the classification performance of our proposed (KNS) method with that of conventional methods such as k-nearest neighbor rule (kNN), Support Vector Machine (SVM), and linear subspace method (SS), from four viewpoints: performance against nonlinearly distributed patterns, performance against multiple category patterns, performance stability as regards parameter changes, and computational cost.

3.1. Two-class problem with a nonlinear boundary

First, we investigated the ability of KNS to classify two categories that have a nonlinear boundary. x_1, y_1, x_2 and y_2 are randomly sampled values from normal distributions with a mean and a variance (μ, σ^2) of $(0, 10)$, $(10, 5)$, $(3, 10)$ and $(20, 5)$ respectively. 2-dimensional patterns of class 1 and 2 were generated by the equation shown in the inset of Figure 1. Figure 1 shows an example of 200 patterns and the optimal boundary determined by KNS. The patterns of class 1 and 2 are shown by open and closed circles respectively. In the example, we chose the 2nd order polynomial function for the kernel (p=2) and set the dimension of the class-subspace to 3. In the following sections, this version of KNS is written as KNS(poly,p=2,d'=3). As shown in Figure 1, KNS can even classify even two-class patterns that have a nonlinear complex boundary.

Figure 2 shows the classification error rates with the proposed method (KNS), SS, SVM and kNN. Each classifier was optimized using various numbers (10 to 300) of training patterns and error rates were measured using a different set of 100 test patterns. The measurements were repeated 100 times and their means and standard errors are shown in the figure. The error rate curve for each method consists of a pair of curves: the upper and lower curves represent

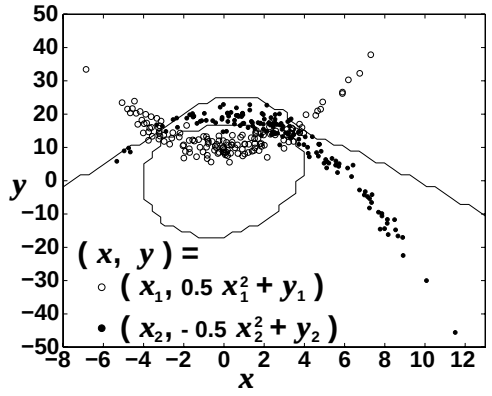

Figure 1: An example of artificial patterns and the classification boundary determined by the proposed method (KNS).

the errors rate for test patterns and training patterns respectively. This kind of analysis provide us a insight about both the classification performance for a small number of training patterns and the potential performance for an infinite number of training patterns. As this pattern distribution is strongly nonlinear, the SS method can not classify patterns correctly. By contrast, KNS showed much better performance than SS and matched that of kNN. Moreover, the convergence to the optimal error rate is faster in KNS than in kNN. The error rate of the SVM(poly,p=1) or SVM(poly,p=2) was much worse than KNS because a higher than 2nd order polynomial function is needed to describe the boundary in this example. Accordingly, SVM(poly, p=3) is slightly better than KNS. SVM(rbf,p=2) showed equivalent performance to SVM(poly,p=3) with slower convergence. These results suggest that KNS can show good performance even when the feature space dimension is low compared with the number of classes or when the distributions of classes overlap.

3.2. KNS for multiple-category problems

We used binary patterns (72 × 76 pixels in size) of 48 category hand-printed Japanese katakana characters provided by ETL-5 [7]. 40 × 50 pixel regions were segmented from each pattern and normalized by the mean and the standard deviation of gray levels in each image. These patterns were compressed by KL expansion from 2000- to 10- or 64-dimensional patterns, in which the cumulative proportions were 0.47 and 0.86, respectively. Finally we obtained 208 of 10 or 64 dimensional patterns for each class. Although these patterns are not necessarily suitable for character recognition, they are reasonable for estimating of classifier performance, because other researchers can regenerate the same patterns, the patterns are distributed in a bounded region in the feature space, they are classified into more than two classes, and they are less arbitrary than artificial patterns. There were 100 training patterns and 100 test patterns.

Table 1 shows classification error rates of various methods for the training patterns and test patterns. The results for various patterns with 10 or 64 pattern dimensions and 10, 20 or 48 classes are shown in each line. As regards SS and KNS, various values of parameters, p and cp, were examined and the best recorded results were shown. When

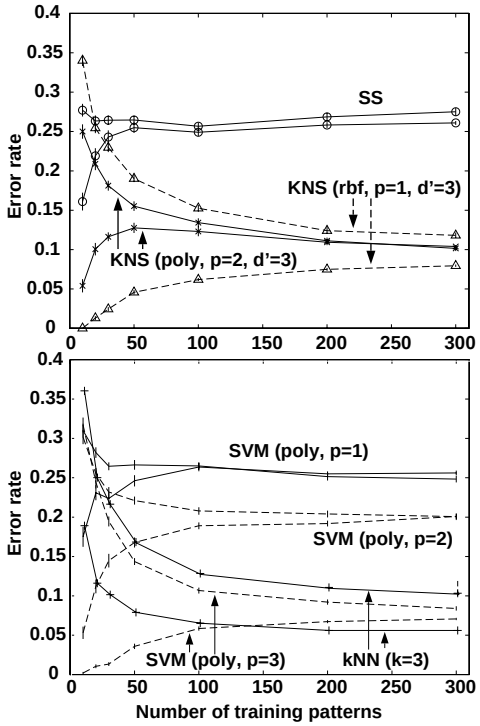

Figure 2: Error rates of the proposed (KNS) and conventional (SS, kNN and SVM) methods.

the feature space dimension, d, was 10, the error rate of SS increased in accordance with the increase in the number of classes, because of the increase in the pattern distribution overlap among the classes. By contrast, KNS performed much better than either SS or kNN. This suggests that the transformation to a high dimensional space in the KNS method improved the classification performance. When d was 64, KNS and SS showed similar results. This suggests that the pattern distribution in this experiment has little nonlinearity. Their error rates were much better than that of kNN, which suggests that the convergence of the kNN performance under training patterns is very slow in a high dimensional feature space.

3.3. Performance stability as regards parameter changes and computational costs

Figure 3 shows an analysis of the performance stability of the KNS and SS methods versus subspace dimensions d'. The upper figure shows error rates for the training and test patterns in various d' and the lower figure shows the corresponding cumulative proportion values. It is clearly shown that while SS is very sensitive to d', the optimal values in KNS are broader. This suggest that the KNS method is more manageable than the SS method despite its nonlinear property. The performance stability as regards the parameter value p of the kernel function was also investigated. While KNS(rbf) was very stable around the optimal value of p, KNS(poly) was more sensitive to p. Its optimal performance, however, was comparable to or sometimes better than that of KNS(rbf).

Figure 4 shows the computation time for various classification techniques needed to design classifiers and to classify

Table 1: Classification error (%) for multiple-category problems

d: dimension				10		10		10				64
# of classes				10		20		48				48
# of patterns	p	cp	train 1000	test 1000	train 2000	test 2000	train 4800	test 4800	p	cp	train 4800	test 4800
kNN (k=1)			0.0	11.7	0.0	22.5	0.0	27.4			0.0	13.9
kNN (k=5)			7.9	11.9	13.7	21.2	18.2	27.0			10.7	15.7
SS		0.81	14.0	16.8	25.8	29.8	36.1	39.5		0.96	0.3	6.6
KNS (rbf)	15	0.96	0.0	8.0	0.0	16.2	0.0	19.2	100	0.98	0.0	5.8
KNS (poly)	2	0.96	2.1	8.7	5.5	18.1	7.4	20.2	2	0.99	0.0	6.9

p: parameter value of the kernel function cp: cumulative proportion of class subspaces

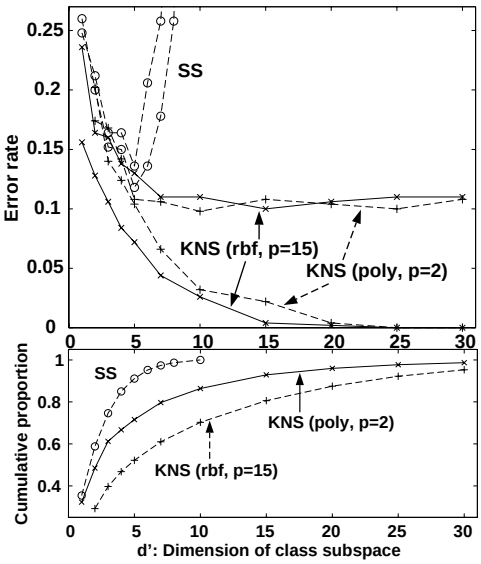

Figure 3: Effect of subspace dimension on error rate.

test patterns for two-class problems. The time is plotted as a function of the number of training patterns. The computational costs for SVM increase to an impractical level at more than 1000 patterns. KNS is 100 faster than SVM. Moreover, while the computational time of SVM is proportional to the square of the number of classes, that of KNS increases linearly to the number of classes.

4. CONCLUSION

In this paper, we proposed a novel pattern classification technique, the Kernel based Nonlinear Subspace (KNS) method. This method is practical for multi-class problems and shows better performance than conventional methods such as the linear subspace method and k-nearest neighbor rule. Moreover it is effective even for two class problems with nonlinear characteristics and its performance is comparable to that of the support vector machine in much less computation time. These results confirm that the KNS method can effectively solve difficult problems where the pattern distribution is nonlinear multiple classes are involved.

Acknowledgments The authors thank Drs. K. Ishii, Y. Shiraki and N. Ueda for their valuable comments.

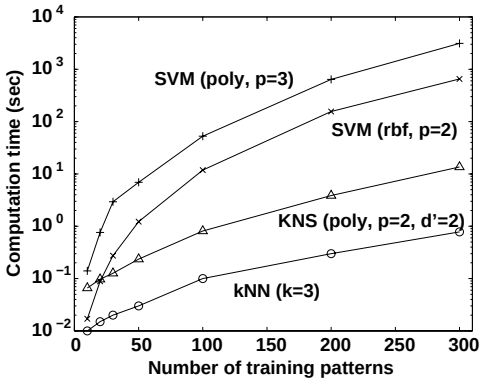

Figure 4: Computational costs of classifiers.

5. REFERENCES

[1] C. Burges. A tutorial on Support Vector Machines for pattern recognition. In *Data Mining and Knowledge Discovery*, pages 1–43. Kluwer Academic Publishers, 1998.

[2] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[3] M. Ikeda, H. Tanaka, and T. Motooka. Projection distance method in hand-written character recognition (in Japanese). 24:106 – 112, 1983.

[4] E. Maeda and H. Murase. Support Vector Machine and kernel based nonlinear subspace method (in Japanese). *Technical Report of IEICE*, PRMU98, 1998.

[5] Erkki Oja. *Subspace Methods of Pattern Recognition*. Research Studies Press Ltd., 1983.

[6] B. Schölkopf, A. Smola, and K-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299 – 1319, 1998.

[7] The Electrotechnical Laboratory. ETL character database. In *ICDAR'93*, volume B, ETL-5, 1993.

[8] K. Tsuda. Subspace method in the Hilbert space (in Japanese). *Technical Report of IEICE*, NC98:47–54, 1998.

[9] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[10] S. Watanabe. *Knowing & Guessing — quantitative study of inference and information*. John Wiley & Sons, Inc., 1969.