Generalization Capabilities of Minimal Kernel-Based Networks

Martin A. Kraaijveld and Robert P.W. Duin

Pattern Recognition Group
Faculty of Applied Physics
Delft University of Technology
P.O. Box 5046
2600 GA Delft
The Netherlands

e-mail: martin@duttnph.tudelft.nl

Abstract

A kernel-based network is a two layer feedforward network in which the units compute a function of the *distance* between the weight vector and the input vector. We will discuss and analyze a class of kernel-based networks, which are based on the Parzen classifier. Although this class of networks has the highly desirable feature of consistency, it requires very high computational demands. Therefore, a new and an existing method will be discussed to minimize the size of these networks, thereby preserving the classification performance as good as possible. Moreover, we will present a theorem that explicitly states the relation between various parameters in the network design procedure and the confidence that one can have in the classification performance of the minimized network. The methods that are presented facilitate very powerful reduction of the network size and are essentially independent of the probability distributions.

Introduction

A large number of the multi-layer networks that are described in the neural networks literature consist of units that compute an inner product of a weight vector and an input vector, followed by a sigmoidal function, e.g. [Rumelhart 1986]. However, recently a number of authors have discussed the use of units that compute a distance measure between an input vector and a weight vector, possibly followed by gaussian shaped output function (e.g. [Robinson 1988] and [Poggio 1989]). The units in such a network can be considered as kernels in the feature space, so the network can be considered as a *kernel-based network*. To classify an unforeseen sample, the distance in feature space between all kernels and the sample is computed, these distances are weighed by the output function of the units, and an output unit computes a label by a (possibly weighted) summation of these weighted distances (see figure 1). Robinson and Poggio describe various training methods for these networks, like variants of backpropagation ([Robinson 1988]), or other gradient descent like procedures ([Poggio 1989]).
In this paper we will discuss and analyze another approach to design a kernel-based network, which is based on a method from the statistical pattern recognition literature. This method is called the *Parzen classifier* ([Parzen 1962], [Duda 1973], [Devijver 1982]) and has recently been introduced as a *Probabilistic Neural Network* ([Specht 1990]). A Parzen classifier, or Probabilistic Neural Network, can be considered as a special case of a kernel-based network, as will be made clear in one of the following paragraphs. An important advantage of the Parzen classifier is that it is provably consistent; i.e. if the number of samples in the learning set approaches infinity, the classifier has the same performance as the Bayes classifier. However, an important disadvantage is that a very large amount of storage is required to store the classifier and that a large amount of CPU-time is required to classify a sample. In this paper it will be discussed *how the size of the classifier, or equivalently the size of the kernel-based network, can be reduced, thereby preserving the classification performance as good as possible.*
This will be done in three steps. In the next paragraph we will define the model that is adopted for the class of networks that is subject of this paper. It will be shown that these networks are equivalent to the Parzen classifier. The following paragraph is dedicated to the discussion of network minimization. We will present how a method that was developed for the nearest neighbor classifier can be adapted to solve the network minimization problem partially. A method that was recently described in [Fukunaga 1989] will be used to solve the remaining part of the minimization problem. The last part of the paper is dedicated to an analysis of the confidence that one can have in the classification performance of the minimized network. For this analysis we will use some methods that were developed in the areas of statistical pattern recognition (e.g. [Vapnik 1982] and [Devroye 1988]), computational learning theory ([Blumer 1989]), and neural networks ([Baum 1989]).

A Class of Kernel-Based Networks

Let us start with some formal definitions. Consider a two class classification problem in a d-dimensional feature space. The two classes are labeled +1 and -1, so a *sample* is denoted as $(\mathbf{x}, \theta)$, where $\mathbf{x}$ is a vector in $\Re^d$, and $\theta \in \{-1, +1\}$. A *design set* is defined as a sequence of $n + m$ samples drawn independently at random from some distribution D on $\Re^d \times \{-1, +1\}$. The first n samples of the design set are called the *learning set* and the remaining m samples are called the *test set*. The learning set will be used to construct the network and the test set will be used

0-7803-0164-1/91/0000-0843$01.00©1991 IEEE

to derive guaranteed estimates of the classification performance of the network. A *classification function* is defined as a function f: $\Re^d \rightarrow \{-1, +1\}$, and a *kernel function* K(x, r) is defined as the normalized indicator function of a hypersphere located at x = 0 with radius r in d-dimensional space:

$$K(x, r) = \begin{cases} M^{-1} & \| x \| \leq r \\ 0 & \| x \| > r \end{cases}$$

where M is the volume of a hypersphere of radius r in d dimensions.

A *kernel-based network* is defined as a network consisting of a set of N kernels in d-dimensional space which implements the following classification function:

$$f(x, r) = \text{sgn}\left(\sum_{i=1}^{N} \theta_i K((x - x_i), r)\right)$$

i.e. a kernel-based network can be considered as a voting scheme in which kernel i has weight $K((x - x_i), r)$.

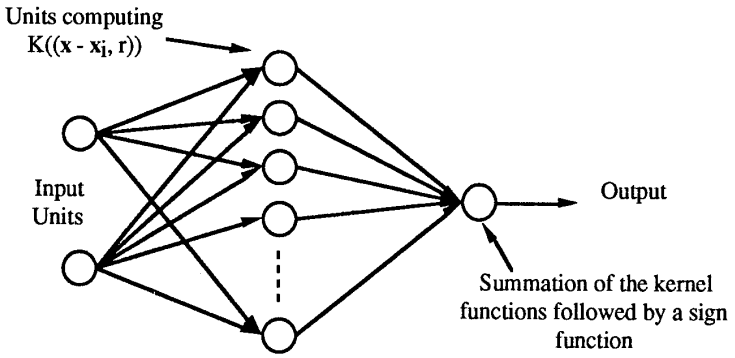

Figure 1: Schematic overview of a kernel-based network.

The previous definitions essentially describe a class of classifiers which are known as *Parzen classifiers* ([Parzen 1962], [Duda 1973], [Devijver 1982]), or *Probabilistic Neural Networks* ([Specht 1990]). When the N = n kernels of the network are placed on the position of the n samples of the learning set a *non-parametric* Parzen estimate of the density of the distributions of the classes is derived. As the Bayes rule prescribes, the classifier then selects the class with the highest a-posteriori probability.
One of the attractive aspects of using a Parzen classifier is that it is provably consistent under very mild conditions. Provided that an appropriate value for the radius of the kernels r is selected, the consistency of a kernel-based network can be guaranteed, since we have not violated any of these conditions for consistency with our choice for K() ([Parzen 1962], [Duda 1973]).
Unfortunately there is no theoretical result known that predicts an optimal value for r given a finite set of samples, without knowledge of the underlying distributions. This has been subject of many previous studies, e.g. [Koontz 1972], [Duin 1976] and [Devroye 1988]. The approach that we will follow in a subsequent paragraph, is that a radius which has a good performance on the test set is selected.

Minimization of Kernel-Based Networks

Although the network that was described above has the highly desirable feature of consistency, one can worry about the large amounts of data that are involved to store the network, and the large amounts of computation that are required to classify a sample. Especially when N gets very large this results in considerable demands for storage and CPU time, even on modern computers. The dilemma that is encountered now is: the network should be as large as possible in order to increase the quality of the density estimates and thereby the classification performance of the network, but as small as possible in order to facilitate fast and efficient operation. In the rest of this paragraph we will therefore discuss a number of methods to reduce the network size, by which we try to preserve the network performance as good as possible. The first method is one of the contributions of this paper, the second was recently described in the literature.

The first solution to the network reduction problem is based on the observation that the density estimates which are implicitly used by the Parzen classifier, are currently used in the context of classification. The difference is that in a classification context it is not necessary to represent the density of the class with locally the lowest density, since

the Bayes rule prescribes that a newly classified sample should be assigned to the class with the highest density. The procedure that will be described therefore aims at removing those kernels which belong to the class with the lowest density (see figure 2). It is essentially a variant of a procedure developed for the nearest neighbor classifier which is called the *multi-edit algorithm* [Devijver 1982]. It consists of 5 steps:

1. Diffusion: Make a random partition of the learning set S into P subsets S_1,.....,S_P, P > 2.
2. Classification: Classify the samples in S_i using the Parzen classifier (i.e. a kernel-based network) using $S_{(i+1) \text{ Mod } P}$ as a training set.
3. Editing: Discard all samples that were misclassified at step 2.
4. Diffusion: Pool all the remaining data to constitute a new set S.
5. Termination: If the last I iterations produced no editing then exit with the final set S, else go to step 1.

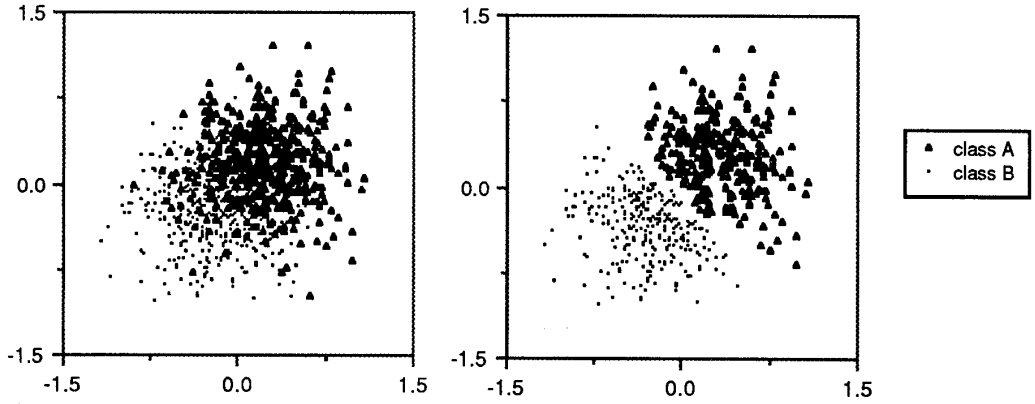

Figure 2: Network minimization by the editing technique. The left figure shows the kernels of the original learning set in the feature space. In the right figure all kernels in the overlapping part of the distributions have been removed by the editing technique.

When the density estimates of the Parzen classifier are based on a sufficiently large number of samples the variance in the density estimates approaches zero. Therefore, in step 2, the probability that a sample of the class with the lowest density will be discarded approaches one. In other words; with probability approaching one, all kernels in the overlapping part of the distributions are removed. (N.B. although these arguments are relatively straightforward, a similar discussion for editing with the nearest neighbor classifier requires a considerable mathematical treatment to arrive at a comparable statement, e.g. see [Devijver 1982]).
With the editing step two important goals are reached. In the first place the number of kernels is decreased, which results in lower demands with respect to storage and CPU time of the classifier. In the second place the consistency of the classifier is preserved, since only those kernels which are not relevant for the classification result are removed.

Another way of looking at the result of the editing algorithm is that the densities of the classes have been transformed into two new densities which do not overlap. These transformed densities will now be the subject of the next minimization method. This method was recently published in [Fukunaga 1989].
Fukunaga describes a method which aims at selecting a subset of kernels to represent the densities. A density which is represented by N kernels, will then be represented by a smaller number, say K kernels. The problem to be solved then is: *which subset of K out of N kernels represents the underlying density best.* The similarity function that Fukunaga proposes is the entropy of the densities approximated by N and K kernels:

$$\int \ln\left(\frac{\hat{p}_K(x)}{\hat{p}_N(x)}\right)\hat{p}_N(x)\,dx$$

Although this problem can not be solved completely, since the number of possible subsets can be prohibitively large, Fukunaga reports very good results with an iterative, but suboptimal, procedure to select the subsets.

The overall procedure that we arrived at is now: editing of the kernels with the editing algorithm described above, followed by an approximation of the edited densities with Fukunaga's method. Can we assume that this overall procedure is consistent ? The answer is yes; when we optimize the entropy of the edited densities approximated by N and K (out of N) kernels, it is evident that we can approximate the edited densities arbitrary well with the best subset

of K kernels, when $K \rightarrow \infty$. Therefore we can say that the method is consistent as long as we let $K \rightarrow \infty$, $N \rightarrow \infty$ and $K/N \rightarrow 0$.

Generalization Capabilities of Kernel-Based Networks

It is clear from the previous paragraph that the classification performance of a kernel-based network is largely based on the quality of the search process for the *best* subset of K kernels. Although we have discussed two reasonable procedures which can guide us through this search, it is clear that numerous other approaches can be adopted to solve this optimization problem.

However, an important question that remains in practice concerns the confidence in the resulting classifier, since in practical situations one will not work with an infinite learning set and the search process will presumably result in a suboptimal subset of kernels. The rest of this paper is therefore devoted to answering the question: *How can we be sure that the classification performance of the minimized network is good enough for a desired application?* To solve this problem in an analytic way we will now adopt a technique which enables the use of some theoretical results. We will use the remaining m samples of the design set, the test set, to judge the quality of the classifier. The search process is now formulated as: search for the subset with the best performance on the test set. Note, that there is still a need for the procedures of the previous paragraph, since an exhaustive search is not feasible in most practical cases. The original question is now reformulated as: *how large should the test set be, in order to have a sufficiently high confidence in the classifier that performed best on the test set ?* The tools that will be used for this analysis are similar to those used by Baum and Haussler [Baum 1989], originally developed by Vapnik [Vapnik 1982]. We will treat two separate cases:

- The desired size of the test set for a network with K, chosen from N kernels, with fixed common radius.
- The desired size of the test set for a network with K, chosen from N kernels, with variable common radius. In this case the radius of the kernels is selected which has the best performance on the test set.

To facilitate the analysis we need some additional definitions. With *e* we denote the base of the natural logarithm, with *ln* we denote the natural logarithm, and with *log* the logarithm base 2. The *error* $\varepsilon(f)$ of a classification function f, with respect to a distribution D, is defined as the probability that $\theta \neq f(\mathbf{x})$ for a random sample $(\mathbf{x}, \theta)$. The (empirical) estimate of the error, i.e. the frequency of errors on a test set of m samples, is denoted by $\hat{\varepsilon}(f)$. Let F^N be the set of classification functions on $\Re^d$ which can be computed by all kernel based network with N kernels, and let X be a test set of m points in $\Re^d$. A *dichotomy* of X induced by any kernel based network $f \in F^N$ is a partitioning of X is into two disjoint subsets X^+ and X^-, such that $f(\mathbf{x}) = +1$ for $\mathbf{x} \in X^+$ and $f(\mathbf{x}) = -1$ for $\mathbf{x} \in X^-$. We denote the number of dichotomies which can be induced on X by the classification functions in F^N by $\Sigma_F(N, X)$, and the maximum of $\Sigma_F(N, X)$ over all sets X of m points in $\Re^d$ is denoted $S_F(N, m)$. When we can label a set of m points X on all 2^m ways with the classification functions in F^N we say that X is *shattered* by F^N. The *capacity V* of F^N is the maximum number of points that can be shattered by F^N. Note that what is called the capacity is closely related to the *capacity* of [Cover 1965], and is equal to the *index* of [Devroye 1988] and to the *Vapnik-Chervonenkis dimension* of [Blumer 1989] and [Baum 1989]. The proofs in this paragraph are motivated and built upon the theory of these last three references. For our analysis we will need the following two theorems:

Theorem 1 *[Vapnik 1982, p.148]: Let the set of classification functions consist of L elements, let X be a test set of m samples, drawn independently according to the distribution D, and for each classification function f_i $(1 \leq i \leq L)$ let the frequency of errors on X equal $\hat{\varepsilon}(f_i)$. Then the probability that there is a classification function f_i for which:*

$$\left(\frac{\varepsilon(f_i) - \hat{\varepsilon}(f_i)}{\sqrt{\varepsilon(f_i)}}\right) > \chi \qquad \textit{is less than:} \qquad L \exp\left(-\frac{1}{2}\chi^2 m\right)$$

Theorem 2 *[Blumer 1989, p. 962]: If the capacity of a set of classification functions F^N is V, then for any $0 < \gamma \leq 1$, $0 < \varepsilon, \delta < 1$ and sample size m, greater than:*

$$m \geq \max\left(\frac{8}{\gamma^2 \varepsilon} \ln\left(\frac{8}{\delta}\right), \frac{16\,V}{\gamma^2 \varepsilon} \ln\left(\frac{16}{\gamma^2 \varepsilon}\right)\right)$$

the probability that there exists a classification function $f \in F^N$ with $\varepsilon(f) > \varepsilon$ such that $\hat{\varepsilon}(f) \leq (1 - \gamma)\, \varepsilon(f)$ is at most δ.

Note that theorem 1 deals with the case of sets with a finite number of classification functions, whereas theorem 2 deals with sets which can contain an infinite number of classification functions. It is important to realize that both theorems are essentially independent of the distribution D.

We will now treat the case of the selection of a subset of K out of N kernels, given a certain fixed radius r_0 for all kernels. We will compute the size of the test set in order to have a desired confidence in the selected classifier. From theorem 1 the following corollary is obtained.

Corollary 1: *Let $0 < \gamma \leq 1$ and $0 < \varepsilon, \delta < 1$. Let the set of classification functions F^N consist of all kernel-based networks which are designed by selecting a subset of K out of N kernels with fixed radius r_0. If one can find a network $f \in F^N$ which erroneously classifies at most a fraction $(1 - \gamma)\, \varepsilon(f)$ of a test set with sample size greater than:*

$$m \geq \frac{2\left(K \ln\left(\frac{eN}{K}\right) + \ln\left(\frac{1}{\delta}\right)\right)}{\gamma^2 \varepsilon}$$

then the probability that this network has error $\varepsilon(f) > \varepsilon$ is at most δ.

proof: As in [Blumer 1989, proposition A3.1] we will express χ as a fraction of $\sqrt{\varepsilon}$ in theorem 1. Let $\chi = \gamma \sqrt{\varepsilon}$. If $\hat{\varepsilon}(f) \leq (1 - \gamma)\, \varepsilon(f)$ then $\varepsilon(f) - \hat{\varepsilon}(f) \geq \gamma\, \varepsilon(f)$, which implies $((\varepsilon(f) - \hat{\varepsilon}(f)) / \sqrt{\varepsilon(f)}) \geq \gamma \sqrt{\varepsilon(f)}$. For a network with error $\varepsilon(f) > \varepsilon$ now follows $((\varepsilon(f) - \hat{\varepsilon}(f)) / \sqrt{\varepsilon(f)}) \geq \gamma \sqrt{\varepsilon} = \chi$. The probability that such a network exists in a set of L networks is exactly given by theorem 1. By choice of m we will now assure that this probability is less than δ. Since the number of networks L in F^N is equal to N over K, which is less than $(e\, N / K)^K$, we have:

$$L \exp\left(-\frac{1}{2}\chi^2 m\right) = \binom{N}{K} \exp\left(-\frac{1}{2}\chi^2 m\right) < \left(\frac{eN}{K}\right)^K \exp\left(-\frac{1}{2}\gamma^2 \varepsilon\, m\right) \leq \delta$$

The result now directly follows. ❑

The case in which the common radius of all the kernels is a free variable which is chosen in order to optimize the performance of the network on the test set, is slightly more complicated. The following theorem presents a lower bound for the required size of the test set.

Theorem 3: *Let $0 < \gamma \leq 1$, $0 < \varepsilon, \delta < 1$, $N \geq 2$ and $1 \leq K \leq N$. Let the set of classification functions F^N consist of all kernel-based networks which are designed by selecting a subset of K out of N kernels of which the common radius can be varied. If one can find a network $f \in F^N$ which erroneously classifies at most a fraction $(1 - \gamma)\, \varepsilon(f)$ of a test set with sample size greater than:*

$$m \geq \max\left(\frac{8}{\gamma^2 \varepsilon} \ln\left(\frac{8}{\delta}\right),\ \frac{32\left(\log K + K \log\left(\frac{eN}{K}\right)\right)}{\gamma^2 \varepsilon} \ln\left(\frac{16}{\gamma^2 \varepsilon}\right)\right)$$

then the probability that this network has error $\varepsilon(f) > \varepsilon$ is at most δ.

proof: Since we can choose from infinitely many values for r, theorem 1 is not applicable in this case. We therefore proceed as in [Baum 1989] by first deriving an upper bound for the number of dichotomies $S_F(N, m)$ in order to use this result to upper bound the capacity of the network.
Suppose that a subset of K kernels is chosen and consider one fixed sample $(\mathbf{x}_i, \theta_i)$ of the test set. At most K sign changes of the classification function can be induced on this sample by varying the common radius of the K kernels (see also [Devroye 1988]). For m samples in the test set we therefore conclude that the number of dichotomies with K kernels is less than or equal to $(mK + 1)$. Since at most N over K subsets of K kernels can be chosen, the upper bound for the number of dichotomies for the resulting network is:

$$S_F(N, m) \leq (m K + 1)\binom{N}{K} < m K \left(\frac{eN}{K}\right)^K$$

The upper bound for the capacity of the network is now derived by demanding: $S_F(N, m) \leq 2^m$. It is easily verified that for $N \geq 2$ and $1 \leq K \leq N$ this inequality holds for $m \geq 2 (\log K + K \log (e\, N / K))$. The upper bound for the capacity V is therefore: $V \leq 2 (\log K + K \log (e\, N / K))$. The result now directly follows from theorem 2. ❑

Discussion.

In some ways our results can be considered as a special case of those derived by Baum and Haussler [Baum 1989]. Their results are applicable for any feedforward network with *any* topology, provided that it consists of units with a binary output. Note that since the capacity of a hyperspheric kernel in $\Re^d$ is equal to the capacity of a linear threshold function in $\Re^d$ (i.e. d + 1, see [Blumer 1989]) we can apply all theorems of Baum and Haussler for networks with linear threshold units directly to the class of networks with hyperspheric kernels described here.

However, for the class of kernel-based networks that is described in this paper, more is known about the topology of the network. Furthermore the weights of these networks (i.e. the positions $\mathbf{x}_i$ of the kernels) can not be considered as real degrees of freedom, since we are only allowed to either select or discard weights by selecting or discarding the corresponding kernels. Due to this decrease in the effective number of degrees of freedom we can present bounds which are slightly better then those of an arbitrary network. The bound of Baum and Haussler for the capacity of a arbitrary feed forward network with linear threshold units is $V \leq 2\ W \log (e\ N)$, where W is the number of weights in the network and N is the number of units. This is roughly a factor d larger than our bound of
$V \leq 2\ (\log K + K \log (e\ N / K)) \approx 2\ K \log N$, since the number of weights in a network with one hidden layer of K units is roughly Kd.
However, their results are independent of a separate learning set, whereas we do require a learning set to design the network. Furthermore, it is interesting to notice that our results are a factor $(1/\varepsilon)$ better than those that could be derived by the approach of [Devroye 1988]. This is due to the fact that theorem 1 and theorem 2 are based on Vapnik's results on the uniform *relative* deviation of frequencies from their probabilities.
Unfortunately, the decrease in the required sample size of the test set does not necessarily imply that a kernel-based network will perform better than a network of the broad class described by Baum and Haussler. A network which has more degrees of freedom generally has more abilities to approximate the Bayes classifier. However, networks with more degrees of freedom generally also need a larger design set. Moreover, in one of the preceding paragraphs it was made clear that there exist some very good ways of designing kernel-based networks, by which one can guarantee the consistency of the classification function. Clearly, this is a result that is still lacking for many other learning procedures (e.g. backpropagation).
Some final remarks. First, the methods that were described in this paper are all independent of the distribution of the classifier. The Parzen density estimate, as well as the analysis of the preceding paragraph, do not depend on any assumptions on the probability densities. It is also clear that the method facilitates very powerful compression of a classifier; e.g. it is not hard to imagine that one can find a reasonable approximation of a multi variate normal density with a moderate number of kernels. Second, the initial choice in the definition of a kernel function for a normalized indicator function of a hypersphere was mainly made for ease of understanding. In [Devroye 1988] it is argued that a kernel function may be the indicator function of any star-shaped set of unit Lebesgue measure. A set S is star shaped if $x \notin S$ implies that $cx \notin S$ for all $c \geq 1$. The notion of the radius will in this case be replaced by the notion of a scaling factor.

Acknowledgements.

This work was sponsored by the Dutch Government as a part of the SPIN/FLAIR-DIAC project, and by the Foundation of Computer Science in the Netherlands (SION) with financial support from the Dutch Organization for Scientific Research (NWO).

Literature.

- [Baum 1989] Baum, E.B., and Haussler, D., "What Size Net Gives Valid Generalization?", Neural Computation 1, pp. 151-160, 1989.
- [Blumer 1989] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, K., "Learnability and the Vapnik-Chervonenkis Dimension", Journal of the ACM, Vol. 36, No. 4, Oct. 1989, pp. 929-965.
- [Cover 1965] Cover, T.M., "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition", IEEE Trans. Electron. Computers, Vol. EC-14, pp. 326-334, 1965.
- [Devijver 1982] Devijver, P.A. and Kittler, J.,"Pattern Recognition, A Statistical Approach", Prentice Hall, 1982.
- [Devroye 1988] Devroye, L., "Automatic Pattern Recognition: A Study of the Probability of Error", IEEE Transactions on PAMI, Vol. 10, No. 4, July 1988.
- [Duda 1973] Duda, R.O., and Hart, P.E., "Pattern Classification and Scene Analysis", John Wiley and Sons, New York, 1973.
- [Duin 1976] Duin, R.P.W., "On the Choice of Smoothing Parameters for Parzen Estimators of Probability Density Functions", IEEE Transactions on Computers, 1976, pp. 1175-1179.
- [Fukunaga 1988] Fukunaga, K. and Hayes, R.R., "The Reduced Parzen Classifier", IEEE Transactions on PAMI, Vol. 11, No. 4, July 1989.
- [Koontz 1972] Koontz, W.L.G., and Fukunaga, K., "Asymptotic Analysis of a Nonparametric Clustering Technique", IEEE Transactions on Computers, Vol. c-21, No. 9, September 1972.
- [Parzen 1962] Parzen, E., "On the Estimation of a Probability Density Function and the Mode", Ann. Math. Statist., Vol. 33, pp. 1065-1076, 1962.
- [Poggio 1989] Poggio, T., and Girosi, F., "A Theory of Networks for Approximation and Learning", MIT AI-lab Memo 1140, July 1989.
- [Robinson 1988] Robinson, A.J., Niranjan, M. and Fallside, F, "Generalizing the Nodes of the Error Propagation Network", Technical Report TR.25, Cambridge University Engineering Department, Nov. 1988.
- [Rumelhart 1986] Rumelhart, D.E., and McClelland, J.L.,"Parallel Distributed Processing",ch.8, MIT-press 1986.
- [Specht 1990] Specht, D.F., "Probabilistic Neural Networks", Neural Networks, Vol. 3, pp. 109-118, 1990.
- [Vapnik 1982] Vapnik, V.N., "Estimation of Dependences Based on Empirical Data", Springer Verlag, 1982.