www.icgst.com

Learning of Decision Fusion Mappings for Pattern Recognition

Friedhelm Schwenker, Christian R. Dietrich, Christian Thiel, Günther Palm
Department of Neural Information Processing, University of Ulm
89069 Ulm, Germany
firstname.lastname@uni-ulm.de,
http://www.informatik.uni-ulm.de/neuro/

Abstract

Different learning algorithms for the decision fusion mapping of a multiple classifier system are compared in this paper. It is very well known that the confusion matrices of the individual classifiers are utilised in the *naive Bayes* combination of classifier outputs. After a brief review of the *decision templates*, the *linear associative memory* and the *pseudoinverse matrix* approaches it is demonstrated that all four adaptive decision fusion mappings share the confusion matrices as the essential ingredient.

Keywords: *Decision fusion mapping, Linear associative memory, Decision templates, Pseudoinverse matrix, Naive Bayes, Confusion matrix.*

1 Introduction

The reason behind this new approach is that in traditional pattern recognition, the best individual classifier has to be found, which is difficult to identify. Also, a single classifier can not make use of information from other discrimination functions.

Typical fusion strategies for the combination of classifiers in MCS are based on *fixed fusion mappings*, for instance averaging or multiplying the classifier outputs [1, 5]. In this paper we consider *supervised learning methods* to train the fusion mapping of the MCS. Whereas the MCS architecture (see Figure 1) is very similar to layered artificial neural networks, like radial basis function networks (RBFs) or multilayer perceptrons (MLPs), the training is rather different. For most artificial neural network architectures, e.g. MLPs, the parameters are typically trained simultaneously through a backpropagation-like training algorithm by a one-phase learning procedure. On the other hand a MCS is typically trained by two completely different training phases:

1. Building the *Classifier Layer* consisting of a set of *classifiers* where each classifier is trained separately, e.g. each classifier is trained on a specific feature subset.

2. Training of the *Fusion Layer* performing a mapping of the classifier outputs (soft or crisp decisions) into the set of desired class labels.

This two-phase MCS training is very similar to RBF learning procedures, where the network is learned over two or even three learning phases:

1. Training of the RBF kernel parameters (centres and widths) of the first layer through clustering or vector quantisation.

2. Supervised learning of the output layer (second layer) of the network.

3. Training both layers of the RBF network through a backpropagation-like optimisation procedure.

Typically, for these three-phase learning scheme a labeled training set is used to calculate the radial basis function centres, the radial basis function widths and the weights of the output layer, but it should be noticed that other learning schemes may be applied. For instance, if a large set of unlabeled data points is available this data can be used to train the parameters of the first RBF layer (centres and widths) through unsupervised clustering [13].

For the training of the two layer fusion architecture two labeled data sets may be used. It is assumed that I classifier decisions, here calculated on I different features, have to be combined. First, the classifiers $\mathcal{C}^i$, $i = 1, ..., I$, are trained, by utilising a training set, and then another labeled training set $\mathcal{R}$, which is not necessarily completely disjunct to the first, is sent to the previously trained first level classifiers to calculate the individual classifier outputs. These classifier outputs $\mathcal{C}^i(\mathbf{x}_i)$, $i = 1, ..., I$ together with the desired class labels are then used to train the decision fusion mapping $\mathcal{F}$. To train this fusion mapping different algorithms have been proposed.

In this paper *decision templates* [7, 8], *naive Bayes decision fusion* [15], *linear matrix memory* [3], and *Pseudoinverse matrix* [3] methods are studied and links between these fusion mappings are shown. The mappings are introduced in Section 2, while Section 3 presents experimental results on their performance. The closing Section explores the close links between the fusion mappings.

2 Adaptive fusion mappings

In this Section we briefly describe methods to train adaptive fusion mappings. It is assumed that a set of first level classifiers $\mathcal{C}^1, ..., \mathcal{C}^I$ has been trained through a supervised learning procedure using a labeled training set. This corresponds to phase 1 of the two-phase MCS learning scheme.

To fix the notation, let $\Omega = \{1, ..., L\}$ be a set of class labels and $\mathcal{C}^i(\mathbf{x}_i^\mu) \in \Delta$ be the probabilistic classifier output of the i-th classifier $\mathcal{C}^i$ given the input feature vector $\mathbf{x}_i^\mu \in \mathcal{R}$, with Δ defined as

$$\Delta := \{(\mathbf{y}_1, ..., \mathbf{y}_L) \in [0,1]^L | \sum_{l=1}^{L} \mathbf{y}_l = 1\}. \quad (1)$$

Then for each classifier the outputs for the training set $\mathcal{R}$ are given by a $(L \times M)$-matrix C_i, $i = 1, ..., I$ where $M = |\mathcal{R}|$ and the μ-th column of C_i contains the classifier output $\mathcal{C}^i(\mathbf{x}_i^\mu)^{\mathrm{T}}$. Here the superscript T denotes the transposition. The desired classifier outputs $\omega^\mu \in \Omega$ for inputs $\mathbf{x}_i^\mu \in \mathcal{R}$ are given by the $(L \times M)$-matrix Y defined by the 1 of L encoding scheme for class labels

$$Y_{l,\mu} = \begin{cases} 1, & l = \omega^\mu \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

That is, corresponding to C_i the μ-th column $Y_{\cdot,\mu} \in \Delta$ contains the binary coded target output of feature vector $\mathbf{x}_i^\mu \in \mathcal{R}$.

In the following we discuss different approaches to combine the classifier outputs $\mathcal{C}^i(\mathbf{x}_i)$, $i = 1, ..., I$ into an overall classifier decision

$$\mathbf{z} := \mathcal{F}(\mathcal{C}^1(\mathbf{x}_1), ..., \mathcal{C}^I(\mathbf{x}_I)). \quad (3)$$

The four different learning schemes, namely *linear associative memory*, *decision template*, *pseudoinverse matrix* and *naive Bayes* are introduced to calculate the decision fusion mapping $\mathcal{F} : \Delta^I \to \Delta$ (see Eq. 3). In all these methods the fusion mapping $\mathcal{F}$ is realised through $(L \times L)$-matrices $V^1, ..., V^I$, calculated through a certain training algorithm. The overall classifier system is depicted in Figure 1.

2.1 Linear Associative Memory

A linear decision fusion mapping $\mathcal{F}$ may be realised through an associative matrix memory whose error-correcting properties have been shown in several numerical experiments and theoretical investigations [6]. In order to calculate the *memory matrix* V^i for each classifier $\mathcal{C}^i$ the stored classifier outputs C_i are adapted through a Hebbian learning rule [6] and V^i is given as the product of the classifier outputs C_i and the desired classifier outputs Y:

$$V^i := \underbrace{Y C_i^{\mathrm{T}}}_{W^i}. \quad (4)$$

In the case of crisp classifiers the matrix V^i is equal to the confusion matrix of classifier $\mathcal{C}^i$, where V^i_{ω,ω^*} is equal to the number of samples of class ω in the training set which were assigned by $\mathcal{C}^i$ to class ω^* [15]. For soft classifiers the ω-th row of V^i contains the accummulated soft classifier decisions of $\mathcal{C}^i$ for the feature vectors $\mathbf{x}_i^\mu \in \mathcal{R}^\omega$.[1]

In the classification phase these matrices are then used to combine the individual classifier decisions to calculate the overall classifier decision (see Eq 3). For a feature vector $X = (\mathbf{x}_1, ..., \mathbf{x}_I)$, the classifier outputs $\mathcal{C}^i(\mathbf{x}_i)$ are applied to the matrices V^i, $i = 1, ..., I$ and the outputs $\mathbf{z}^i \in \mathbb{R}^L$ are given by

$$\mathbf{z}^i := V^i(\mathcal{C}^i(\mathbf{x}_i))^{\mathrm{T}}. \quad (5)$$

The combined class membership estimate based on I feature vectors is then calculated as the average of the individual outputs of the second level classifiers

$$\mathbf{z} := \sum_{i=1}^{I} \mathbf{z}^i = \sum_{i=1}^{I} V^i \mathcal{C}^i(\mathbf{x}_i)^{\mathrm{T}} \quad (6)$$

and the final class label based on the combined class membership estimate $\mathbf{z}$ is then determined by the maximum membership rule

$$\omega := \underset{l \in \Omega}{\operatorname{argmax}}(\mathbf{z}_l). \quad (7)$$

2.2 Decision Templates

The concept of decision templates is a simple, intuitive, and robust aggregation idea that evolved from the *fuzzy template* which was introduced by KUNCHEVA, see [8, 9]. Decision templates are calculated as the mean of the classifier outputs for inputs $\mathbf{x}_i^\mu$ of class $\omega \in \Omega$

$$\mathcal{T}_i^\omega := \frac{1}{|\mathcal{R}^\omega|} \sum_{\mathbf{x}_i^\mu \in \mathcal{R}^\omega} \mathcal{C}(\mathbf{x}_i^\mu). \quad (8)$$

In the case of I input features for each class the decision template $\mathcal{T}^\omega$ is given by the $(I \times L)$-matrix

$$\mathcal{T}^\omega := \begin{bmatrix} \mathcal{T}_1^\omega \\ \vdots \\ \mathcal{T}_I^\omega \end{bmatrix} \in \Delta^I. \quad (9)$$

[1] Independent from the classifier type (soft or crisp) we consider W^i a confusion matrix. It is defined to compare the individual fusion schemes (see Eq. 4,11,12 and 14).

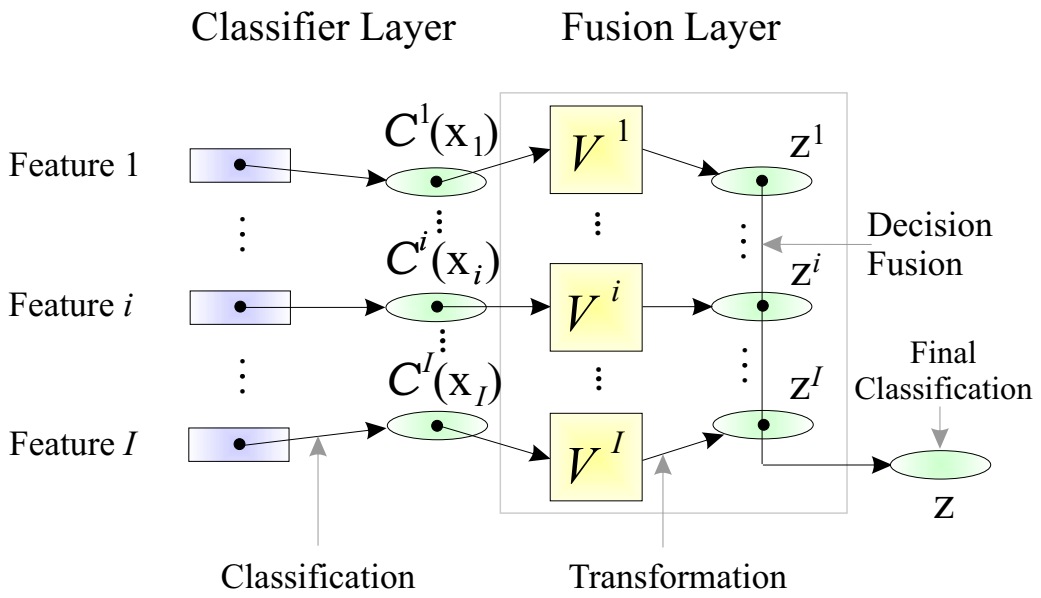

Figure 1: Two layer MCS architecture consisting of a classifier layer and an additional fusion layer. In general the combination of the classifier outputs $\mathcal{C}^i(\mathbf{x}_i)$, $i = 1, ..., I$ is accomplished through a fusion mapping $\mathcal{F}(\mathcal{C}^1(\mathbf{x}_1), ..., \mathcal{C}^I(\mathbf{x}_I))$. In this paper we restrict ourselves to separable linear fusion mappings, where the classifier outputs $\mathcal{C}^i(\mathbf{x}_i)$ are multiplied with matrices V^i, $i = 1, ..., I$. The resulting decisions $\mathbf{z}^1, ..., \mathbf{z}^I$ are combined with decision fusion. To produce independent answers, the classifiers in this example work on different features.

In order to align the decision template combining scheme in such a way that the combination is done as proposed in Figure 1, for each feature space a linear matrix operator V^i is calculated. Let $\mathcal{T}_i^\omega \in \Delta$ be the decision template of the i-th feature space and target class ω as defined in Eq. 8. Then for each classifier $i = 1, ..., I$, a $(L \times L)$-decision template V^i is given by the decision templates of the i-th feature space

$$V^i := \begin{bmatrix} \mathcal{T}_i^1 \\ \vdots \\ \mathcal{T}_i^L \end{bmatrix} \in \Delta^L. \tag{10}$$

It can be computed similar to the associative memory matrix (see Eq. 4) by using the stored classifier outputs and the desired classifier outputs

$$V^i := (YY^{\mathrm{T}})^{-1} \underbrace{(YC_i^{\mathrm{T}})}_{W^i}. \tag{11}$$

Multiplying the confusion matrix W^i with $(YY^{\mathrm{T}})^{-1}$ from the left is equivalent to the row wise normalisation of W^i with the number of training patterns per class.

Assuming the normalised correlation as similarity measure [7, 8] the combination of the classifier outputs with decision templates is equivalent to the combination of classifier outputs with the linear associative memory. Thus, for a set of input vectors $X = (\mathbf{x}_1, ..., \mathbf{x}_I)$ and a set of classifier outputs $\mathcal{C}^i(\mathbf{x}_i)$, $i = 1, ..., I$ the combined class membership estimate is given by Eq. 6.

2.3 Pseudoinverse Solution

Another linear decision fusion mapping $\mathcal{F}$ can be calculated as an optimal least squares solution between the outputs of the second layer $V^i C_i$ and the desired classifier outputs Y. Such a mapping is given by the pseudoinverse solution which defines another type of linear associative memory [3]. The linear matrix operator V^i based on the pseudoinverse solution (also called generalised inverse matrix) is given by

$$V^i := Y \lim_{\alpha \to 0_+} C_i^{\mathrm{T}}(C_i C_i^{\mathrm{T}} + \alpha I)^{-1} = YC_i^+ \tag{12}$$

with C_i^+ being the pseudoinverse of C_i and I for once being the identity matrix. Provided the inverse matrix of $C_i C_i^{\mathrm{T}}$ exists, which is always the case for full rank matrices C_i, the pseudoinverse solution is given through

$$V^i = Y \; C_i^{\mathrm{T}}(C_i C_i^{\mathrm{T}})^{-1} = \underbrace{(YC_i^{\mathrm{T}})}_{W^i}(C_i C_i^{\mathrm{T}})^{-1}. \tag{13}$$

As with the linear associative memory and the decision template, the matrix operators $V^1, ..., V^I$ based on the pseudo-inverse solution are used for a linear combination of the classifier outputs. Therefore, the combined class membership is given by Eq. 6 as well.

2.4 Naive Bayes Decision Fusion

In the naive Bayes fusion scheme the classifiers $\mathcal{C}^1, ..., \mathcal{C}^I$ are assumed to be crisp and mutually independent [7], therefore in [15] it is called *Bayes combination*. In the case of I feature spaces for each feature

space a $(L \times L)$-matrix operator is calculated based on the stored classifier outputs C_i and the desired classifier outputs Y:

$$V^i := \underbrace{(YC_i^{\mathrm{T}})}_{W^i}(C_i C_i^{\mathrm{T}})^{-1}. \tag{14}$$

In contrast to decision templates (see Eq. 11) the confusion matrices W^i, $i = 1, ..., I$ are normalised column wise and V^i is called *label matrix*. Due to the fact that a classifier $\mathcal{C}^i(\mathbf{x}_i)$ is typically error-bearing, the individual confusions describe the uncertainty of $\mathcal{C}^i$ to classify $\mathbf{x}_i$. Therefore, the (m, l)-th entry of V^i is an estimate of the conditional probability

$$\hat{P}(\omega = m | \mathcal{C}^i(\mathbf{x}_i) = l), \tag{15}$$

that is the probability that the true class label is m given that $\mathcal{C}^i(\mathbf{x}_i)$ assigns the crisp class label $l \in \Omega$.

After learning, the label matrices $V^1, ..., V^I$ are used to calculate the final decision of the classifier ensemble. The classification works as follows: Let $\mathcal{C}^1(\mathbf{x}_1) = \omega_1, ..., \mathcal{C}^I(\mathbf{x}_I) = \omega_I$ be be the class decisions of the individual classifiers for a new sample. Then by the independence assumption [4], the estimate $\mathbf{z}_m$ of the probability that the true class label is m, is calculated as [15]

$$\mathbf{z}_m := \alpha \prod_{i=1}^{I} \hat{P}(\omega = m | \mathcal{C}^i(\mathbf{x}_i) = \omega_i), \quad m = 1, ..., L \tag{16}$$

where α is a normalising constant that ensures $\sum_{m=1}^{L} \mathbf{z}_m = 1$. This type of Bayesian combination and various methods for uncertainty reasoning have been studied extensively in [11].

3 Experiments

All four schemes have been tested using two benchmark sets, namely our photos of fruits and Coil20. The fruits data set comes from a robotic environment, consisting of 840 pictures from 7 different classes (apples, oranges, plums... [2]). Four features were extracted (3 histograms left/middle/right using Sobel edge detector, mean colour values from the HSV representation) and then classified separately with a Gaussian RBF network which had been initialised using the K-Means algorithm (for details see [14]). Coil20 is the well known data set from the Columbia Object Image Library [12], consisting of 1440 greyscale pictures from 20 different classes, from which 4 features were extracted (concatenation of 7 invariant hu moments [10], orientation histograms utilising sobel or canny edge detectors respectively on the grey scale image as well as on the black/white channel of the opponent colour system) and then classified separately with a k-nearest-neighbour classifier. The accuracy after combining the answers from the different classifiers using one of the four fusion schemes was then measured using 5x5 times cross validation with the fruits and 5x6 times with the Coil20 data set. Please note that no effort has been made to achieve high accuracy or select good features, the emphasis was on comparing the fusion architectures.

	Fruits	Coil20
Linear assoc. memory	94.93	98.87
Decision templates	95.71	99.06
Pseudoinverse	95.81	99.28
Naive Bayes	89.17	78.14

Table 1: Accuracy of the four fusion schemes on two different data sets in %.

As can be seen in Table 1, the naive Bayes scheme falls well behind for the two data sets tested. This could not be attributed to it working only on crisp answers, as the accuracy of the other fusion schemes did not change much in experiments when also fed only with crisp classifier outputs. The linear associative memory and decision templates fusion schemes do yield different results as we did not use the normalised correlation as similarity measure for the later (as shown in Table 2) but the S_1 measure introduced in [9].

It is striking that three of the schemes achieve very similar results, demonstrating that despite coming from different research directions they are quite alike, in fact all using the confusion matrix W^i.

4 Discussion

Finally we want to explore the close ties between the four combining schemes. All are realised using $(L \times L)$-matrices $V^1, ..., V^I$, obtained using a certain training algorithm (see Table 2 for an overview). These matrices are based on the classifier outputs $C_1, ..., C_I$ of the corresponding first level classifiers $\mathcal{C}^1, ..., \mathcal{C}^I$ and the desired output Y.

The confusion matrix $W^i = YC_i^{\mathrm{T}}$ (see Eq. 4, 11, 12 and 14) is the main ingredient in all combination schemes, with W^i describing the uncertainty of the classifier which is taken into account for the training of the fusion layer. From another point of view, the confusion matrix W^i could be regarded as prior knowledge about the classifier $\mathcal{C}^i$ [15].

In three of the four combining schemes the combination of the classifier decisions is a matrix-vector product (see Eq. 6). The only exception is the naive Bayes fusion scheme.

By assuming the normalised correlation as similarity measure, classifier fusion via decision templates is very similar to the combination with linear associative memories. Provided, that the number of feature vectors for each class in $\mathcal{R}$ is equal to $\kappa \in \mathbb{N}$, the normalisation term of the decision template $(YY^{\mathrm{T}})^{-1}$

	Matrix $V^i =$	Usage
Linear assoc. memory	(YC_i^{T})	$\mathbf{z} := \sum_{i=1}^{I} V^i \mathcal{C}^i(\mathbf{x}_i)^{\mathrm{T}}$
Decision templates	$(YY^{\mathrm{T}})^{-1}(YC_i^{\mathrm{T}})$	"
Pseudoinverse	$(YC_i^{\mathrm{T}})(C_iC_i^{\mathrm{T}})^{-1}$	"
Naive Bayes	$(YC_i^{\mathrm{T}})(C_iC_i^{\mathrm{T}})^{-1}$	$\mathbf{z} := \prod_{i=1}^{I} V^i \mathcal{C}^i(\mathbf{x}_i)^{\mathrm{T}}$ $\prod$ = element-wise product

Table 2: Overview of the central matrix $V^i =$ in the different approaches, and what to do in each case to get the classification result z, which is an estimate for the class membership. The final decision is hence obviously for the class with the highest probability (see Eq. 7).

(see Eq. 11) can be written as

$$(YY^{\mathrm{T}})^{-1} = \mathtt{diag}(\frac{1}{\kappa}, ..., \frac{1}{\kappa}). \tag{17}$$

In comparison to the linear associative memory matrix the decision vector $\mathbf{z}$ of the decision template is multiplied by $\frac{1}{\kappa}$ and therefore the class decisions of both fusion schemes are equivalent because both are based on the maximum membership rule (see Eq. 7).

Regarding the coefficients of the matrix operators $V_1, ..., V_I$, fusion with the naive Bayes combination is very similar to the pseudoinverse matrix solution. If crisp classifiers are employed to classify the feature vectors in $\mathcal{R}$, the term $C_iC_i^{\mathrm{T}}$ (see Eq. 12 and Eq. 14) is a diagonal matrix containing the number of feature vectors of the individual classes. Let κ^ω be the number of feature vectors classified to class $\omega \in \Omega$ in $\mathcal{R}$. Then the normalisation term is given by

$$C_iC_i^{\mathrm{T}} = \mathtt{diag}(\kappa^1, ..., \kappa^L). \tag{18}$$

If $\kappa^\omega \neq 0$ for each $\omega \in \Omega$ the inverse of $C_iC_i^{\mathrm{T}}$ is given through

$$(C_iC_i^{\mathrm{T}})^{-1} = \mathtt{diag}(\kappa^1, ..., \kappa^L)^{-1} = \mathtt{diag}(\frac{1}{\kappa^1}, ..., \frac{1}{\kappa^L}). \tag{19}$$

But then the combination schemes for the naive Bayes combination (see Eq. 16) and the pseudoinverse solution (see Eq. 12) are different, leading to different decisions.

5 Acknowledgements

This work was partially supported by the DFG (German Research Society) contract SCHW 623/3-2.

References

[1] C. Dietrich, F. Schwenker, and G. Palm. Classification of time series utilizing temporal and decision fusion. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, pages 378–387. Springer, 2001.

[2] R. Fay, U. Kaufmann, F. Schwenker, and G. Palm. Learning Object Recognition in a NeuroBotic System. In H.-M. Groß, K. Debes, and H.-J. Böhme, editors, *3rd Workshop on SelfOrganization of AdaptiVE Behavior SOAVE 2004*, number 743 in Fortschritt-Berichte VDI, Reihe 10, pages 198–209. VDI, 2004.

[3] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.

[4] E. T. Jaynes. *Probability Theory: The Logic of Science*. Washington University, St. Louis, MO 63130, U.S.A., 1996.

[5] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[6] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.

[7] L. I. Kuncheva. *Fuzzy Classifier Design*. Physica Verlag, Heidelberg New York, 2000.

[8] L. I. Kuncheva. Using measures of similarity and inclusion for multiple classifier fusion by decision templates. *Fuzzy Sets and Systems*, 122(3):401–407, 2001.

[9] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin. Decision templates for multiple classifier fusion. *Pattern Recognition*, 34(2):299–314, 2001.

[10] M. K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Trans. Info. Theory*, IT-8:179–187, 1962.

[11] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers, San Francisco, 1988.

[12] S. Nene, S. Nayar, and H. Murase. Columbia Object Image Library: COIL. Technical report CUCS-006-96, COIL 20, Department of Computer Science, Columbia University, Feb. 1996.

[13] F. Schwenker, H. A. Kestler, and G. Palm. Three phase learning for radial basis function networks. *Neural Networks*, 14:439–458, 2001.

[14] C. Thiel. Multiple Classifier Fusion Incorporating Certainty Factors. Master's thesis, University of Ulm, Germany, 2004.

[15] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications in handwritten character recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 22(3):418–435, 1992.