ELSEVIER

Pattern Recognition Letters 19 (1998) 227–235

Pattern Recognition Letters

Data equalisation with evidence combination for pattern recognition

Geok See Ng [a,*], Harcharan Singh [b]

[a] *Division of Software Systems, School of Applied Science, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore*

[b] *Division of Computing Systems, School of Applied Science, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Singapore*

Received 21 February 1997; revised 1 October 1997

Abstract

In this paper, data equalisation is applied to output nodes of individual classifier in a multi-classifiers system such that the average difference of the output activation values is smaller. This helps in overall competitiveness of the output nodes of individual classifier. This will then improve the accuracy rate of a combined classifier which aggregates the outputs of the front-end classifiers.

Keywords: Data equalisation; Recognition system; Neural network; Evidence combination; Multi-classifier systems

1. Introduction

In this paper, data equalisation is applied to the output of a multi-classifiers system. Each classifier is a neural network with different architecture. The output of the classifier is then combined by a Combined Classifier using the Dempster–Shafer theory method (Shafer, 1976) with a new support function (Ng and Singh, 1997). Ten thousand hand-written digits are used for training and five thousand digits are used for testing.

The problem of many multi-classifiers systems (Xu et al., 1992; Quinlan, 1996) is to improve the accuracy rate. Notwithstanding the many satisfactory results that have been reported for constrained domains, pattern classification problems involving noisy patterns and many classes remain difficult. Xu et al. (1992) suggested that a number of classifiers be combined and used in parallel to overcome the difficulties. This configuration has the advantage that learning and classification procedures of different types can be used simultaneously to complement one another and improve the overall accuracy of recognition.

Breiman's bagging (Breiman, 1996) and Freund and Schapire's boosting (Freund and Schapire, 1996) are recent methods for improving accuracy of multi-classifiers systems. Bagging produces replicate training data by sampling with replacement from the training instances. Boosting uses all training instances, maintains a weight

* Corresponding author.

0167-8655/98/$19.00
PII S0167-8655(98)00005-1

for each training instance and adjusts the weights to cause the classifier to focus on different instances. These two methods manipulate the training data. However, data equalisation in this paper does not manipulate the training data. Instead, it manipulates the output activation values of each classifier. Boosting boosts a ''weak'' classifier to a ''strong'' classifier. However, data equalisation does not affect the strength of each classifier. Instead, it increases the competitiveness among classes, not classifiers. Although boosting generally increases accuracy, it leads to a deterioration on some training data sets. Data equalisation does not have any effect on training data sets.

In order to analyse the output activation values of the recognition system, a rejection-accuracy approach is used. It is obvious that the recognition system should reject the input patterns that are inside the overlapped region of different classes because these are the patterns that degrade the accuracy of the systems. LeCun et al. (1989) have used different rejection processes for zip code recognition systems. A rejection-accuracy plane is used as a tool for discussing the trade-offs between accuracy of recognition and fraction of accepted input patterns.

The proposed data equalisation uses a transformation function which is a cumulative distribution of output activation values of the front-end classifiers. The motivation of this technique is to increase the dynamic ranges of the output activation values. This will then increase the overall competitiveness of the output nodes of individual classifier. These outputs are then aggregated by a Combined Classifier (CC) into a single output. The experimental results show that by having data equalisation, CC's accuracy rate is improved.

2. The accuracy-rejection approach

2.1. The accuracy-rejection plane

To avoid unnecessary complications, the following notation for the probabilities associated with events is used: $p(\text{event}_1,\text{event}_2,\dots)$ is the probability that event_1, event_2, and so on happen. For example, $p(w_1,w_2,\text{equal})$ is the probability that the responses of both Classifiers 1 and 2 are wrong and that the output classes are equal. The specific $p()$ function is uniquely identified by its arguments.

In the following, an abstract description of a classifier that processes a vector $\boldsymbol{x}$ of inputs and provides both a decision (i.e., a class) and a binary value about the confidence in the classification is considered. If the confidence *flag* is set to zero (uncertain response), the pattern is *rejected* by the classifier. The performance of a classifier can be described by a point in the *rejection-accuracy plane* (in short *R-A*), that is, by its probability of an accurate response, given that the input pattern is accepted: $A = p(\text{correct}\,|\,\text{accept})$, and by its probability of rejection $R = p(\text{reject})$. Note that the *accuracy* is always conditional on the acceptance of the pattern. If several classifiers are involved, $R_i = p(\text{reject}_i)$ and $A_i = p(\text{correct}_i\,|\,\text{accept}_i)$ are defined for the ith classifier.

In general, the *R-A* values depend on the internal parameters of the classifier. By varying these parameters one obtains the accuracy as a function of the rejection rate: $A(R)$, a function that depends on the specific rejection scheme. The $A(R)$ function is increasing with R if a growing fraction of the accepted cases consist of correctly classified patterns.

The *R-A* co-ordinates introduce a partial ordering of different classifiers: Classifier X is better than Classifier Y if it has both a greater accuracy and a smaller rejection. In other cases (for example, if X has greater accuracy but also greater rejection than Y), the preference is decided by a *compromise* between the two requirements of high accuracy and low rejection. Introducing a parameter λ to regulate the relative importance of the two requirements, the optimal classifier for a given application can be defined as the one that maximises $U = A - \lambda R$ with $\lambda \geqslant 0$. The left-hand side of Fig. 1 graphically illustrates how the gradient of the compromise function U, equal to $(-\lambda,1)$, introduces a complete ordering of the classifiers, apart from ties. Other performance criteria can be based on the simultaneous satisfaction of two in-equalities of the type: $A \geqslant A_{\min}$ and $R \leqslant R_{\max}$ (the right-hand side of Fig. 1).

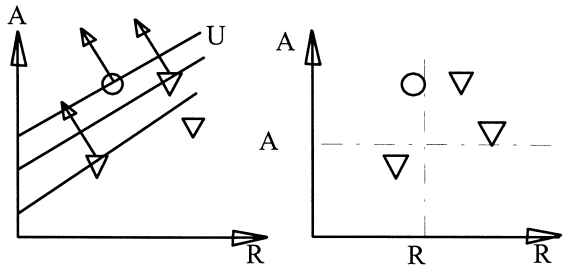

Fig. 1. Selection of the optimal classifier (point O in the R-A plane).

2.2. The common consent scheme for classifiers

The rejection is based only on the comparison of the outputs provided by the set of classifiers. An inaccurate classification is signaled (with a high probability) by the disagreement between the responses of different classifiers. If a pattern of Class X is wrongly classified as belonging to Class Y, the chance that other classifiers wrongly classify it as belonging to a different Class Z must be substantial. This can occur if there are many possible output classes with complex decision boundaries, for example, if the various classifiers have different input features, architectures, initialisations, or learning algorithms.

2.3. Common consent rules

Consider two classifiers in four possible outcomes of the classification. The classifier responses can be:
1. all correct, and therefore all equal, event (c_0,c_1),
2. all wrong but all equal, event (w_0,w_1,equal),
3. all wrong and unequal, event $(w_0,w_1,\text{unequal})$,
4. one correct and one wrong, event (c_0,w_1) or (w_0,c_1).

The common consent scheme accepts a pattern if the two responses are the same. The following probabilities for the composite classifier are obtained:

$$p(\text{accept}) = p(c_0,c_1) + p(w_0,w_1,\text{equal}), \tag{1}$$

$$p(\text{correct}\,|\,\text{accept}) = \frac{p(\text{correct,accept})}{p(\text{accept})} = \frac{p(c_0,c_1)}{p(c_0,c_1) + p(w_0,w_1,\text{equal})}. \tag{2}$$

In fact, events of the type (c_0,w_1) are always rejected because the response class cannot be equal. It is straightforward to generalise to the case of N classifiers, obtaining the following R-A values:

$$R = 1 - p(c_0,c_1,\ldots,c_{N-1}) - p(w_0,w_1,\ldots,w_{N-1},\text{equal}), \tag{3}$$

$$A = \frac{p(c_0,c_1,\ldots,c_{N-1})}{p(c_0,c_1,\ldots,c_{N-1}) + p(w_0,w_1,\ldots,w_{N-1},\text{equal})} \approx 1 - \frac{p(w_0,w_1,\ldots,w_{N-1},\text{equal})}{p(c_0,c_1,\ldots,c_{N-1})}, \tag{4}$$

where the last approximation is valid for a high signal-to-noise ratio, that is, for $p(w_0,w_1,\ldots,w_{N-1},\text{equal}) < p(c_0,c_1,\ldots,c_{N-1})$.

In general, the performance of the common consent scheme depends on the *joint* probability distribution for the output responses produced by input patterns extracted from the different classes. From an operational point of view, a specific scheme can be evaluated by combining the responses provided by the individual classifiers on a suitable test set. It can be obtained also by calculating the probabilities needed in Eqs. (3) and (4). If the individual responses are available, then the actual test is suggested for a general recognition problem.

2.4. Independent confusion

Assume that the distribution of confused cases among the different classes is known. For each classifier n in a team of N classifiers, define $p_n(\omega_j \mid \omega_i)$ as the conditional probability that a pattern is recognised as belonging to class ω_j given that the correct class is ω_i. The probabilities describe the accuracy for the different classes and the spread of the wrong classifications among the incorrect classes.

The results can be obtained by CC in the approximation of independence among the different classifiers. Therefore, the probability of a set of output responses given an input class is the product of the individual probabilities:

$$p(w_0, w_1, \ldots, w_{N-1}, \text{equal}) = \sum_{i=0}^{C-1} p(\omega_i) \sum_{j=0,\, j\neq i}^{C-1} \left[\prod_{n=0}^{N-1} p_n(\omega_j, \omega_i) \right], \tag{5}$$

$$p(w_0, w_1, \ldots, w_{N-1}) = \sum_{i=0}^{C-1} p(\omega_i) \prod_{n=0}^{N-1} p_n(\text{wrong} \mid \omega_i) = \sum_{i=0}^{C-1} p(\omega_i) \prod_{n=0}^{N-1} \left[1 - p_n(\omega_i \mid \omega_i)\right], \tag{6}$$

$$p(c_0, c_1, \ldots, c_{N-1}) = \sum_{i=0}^{C-1} p(\omega_i) \prod_{n=0}^{N-1} p_n(\omega_i \mid \omega_i). \tag{7}$$

From the above equations, it is straightforward to derive estimates for the accuracy and rejection probabilities in a system composed of several classifiers.

3. Data equalisation

Let $\boldsymbol{x}$ be an input vector and N be the number of different classifiers, f^n, $n = 0,1,\ldots,N-1$. Assume that each classifier produces an output vector $\boldsymbol{y}^n \in \mathbb{R}^K$, $\boldsymbol{y}^n = f^n(\boldsymbol{x})$. Here K is the number of classes (in the case of digit recognition, $K = 10$). Let y_i^n be the activation value of output node i of the classifier n. So $\boldsymbol{y}^n = (y_0^n, y_1^n, \ldots, y_{K-1}^n)$. y_1^n is normalised such that it lies in the interval [0,1]. Let y be a general variable of y_1^n. Hence for any y in the interval [0,1], a transformation function can be applied to yield

$$z = T(y). \tag{8}$$

It is assumed that the transformation function given in Eq. (8) satisfies the conditions:

1. $T(y)$ is single-valued and
2. $0 \leqslant T(y) \leqslant 1$ for $0 \leqslant y \leqslant 1$.

The inverse transformation from z back to y is denoted by

$$y = T^{-1}(z), \quad 0 \leqslant z \leqslant 1, \tag{9}$$

where the assumption is that $T^{-1}(z)$ also satisfies condition 1 and 2 with respect to variable z.

From elementary probability theory, the probability density function of z is

$$p_r(z) = \left[p_r(y) \frac{\mathrm{d}y}{\mathrm{d}z} \right]_{z = T^{-1}(y)}. \tag{10}$$

In this work, a transformation function which is the cumulative distribution function of y is proposed:

$$z = T(y) = \int_0^r p_r(w)\, \mathrm{d}w, \quad 0 \leqslant y \leqslant 1, \tag{11}$$

where w is a dummy variable of integration and $p_r(w)$ is the probability density function of w.

From Eq. (11), the derivative of z with respect to y is

$$\frac{\mathrm{d}z}{\mathrm{d}y} = p_r(y). \tag{12}$$

Substituting $\mathrm{d}y/\mathrm{d}z$ into Eq. (10) yields

$$p_r(z) = \left[p_r(y) \frac{\mathrm{d}y}{\mathrm{d}z} \right]_{z=T^{-1}(y)} = 1, \tag{13}$$

which is uniform density in the interval [0,1].

The foregoing development indicates that using a transformation function equal to the cumulative distribution of y produces a uniform probability density function of z. In terms of improvement, this helps in higher contrast of output activation values before the evidence combination at the CC.

4. Evidences combination

4.1. The Dempster–Shafer theory

After the data equalisation, a new evidence combination method based on the Dempster–Shafer theory is used for combining the output activation values of various classifiers (Ng and Singh, 1997). Dempster–Shafer theory (Shafer, 1976) is motivated by the observation that Bayesian theory cannot represent ignorance. Like Bayesian probabilities and non-monotonic reasoning, which offers a methodology for reasoning under uncertainty, the Dempster–Shafer theory provides a framework for evaluating uncertainty information. In some cases, the theory can be usefully applied because of its unique and highly intuitive approach to addressing modeling tasks. The theory can combine evidence in a consistent and probabilistic manner to arrive at a more complete assessment of what the entire body of evidence implies.

Let θ be the quantity and Θ be a set of possible values of θ, i.e., $\Theta = \{\theta_0, \ldots, \theta_{K-1}\}$. Θ is called the *frame of discernment*. When a proposition corresponds to a subset of a frame of discernment, it means that the frame discerns that proposition. One reason to associate a proposition with a subset of Θ is that the logical notions of conjunction, disjunction, implication and negation can be translated into the more graphic set-theoretic notions of intersection, union, inclusion and complementation.

Let 2^{Θ} denote the set of all subsets of Θ. A function $\boldsymbol{m}$ is called a *basic probability assignment* if

$$\boldsymbol{m}: 2^{\Theta} \to [0,1], \quad \boldsymbol{m}(\varnothing) = 0 \text{ and } \sum_{A \subseteq \Theta} \boldsymbol{m}(A) = 1. \tag{14}$$

The quantity $\boldsymbol{m}(A)$ is called A's *basic probability number* and it is understood to be the measure of the belief that is committed exactly to A. To obtain the measure of the total belief committed to A, one must add to $\boldsymbol{m}(A)$ the quantity $\boldsymbol{m}(A)$ for all subsets B of A:

$$\mathbf{Bel}(A) = \sum_{B \subseteq A} \boldsymbol{m}(B). \tag{15}$$

A function $\mathbf{Bel}: 2^{\Theta} \to [0,1]$ is called a belief function over Θ. The belief function with the simplest structure is the one obtained by setting $\boldsymbol{m}(\Theta) = 1$ and $\boldsymbol{m}(A) = 0$ for all $A \neq \Theta$, and it has $\mathbf{Bel}(\Theta) = 1$ but $\mathbf{Bel}(A) = 0$ for all $A \neq \Theta$. Hence there is a one-to-one correspondence between the belief function and the basic probability assignment.

If $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are basic probability assignments, their combination, $\boldsymbol{m} = \boldsymbol{m}_1 \oplus \boldsymbol{m}_2$, is defined as

$$\boldsymbol{m}(A) = C^{-1} \sum_{D \cap B = A} \boldsymbol{m}_1(B)\boldsymbol{m}_2(D), \tag{16}$$

where

$$C^{-1} = \sum_{D \cap B \neq \varnothing} \boldsymbol{m}_1(B)\boldsymbol{m}_2(D), \quad \boldsymbol{m}(\varnothing) = 0 \text{ and } A \neq \varnothing. \tag{17}$$

Obviously, the combination rule may be generalized to combine multiple evidence. Because there is a one-to-one correspondence between **Bel** and $\boldsymbol{m}$, the orthogonal sum of belief functions is therefore $\mathbf{Bel} = \mathbf{Bel}_1 \oplus \mathbf{Bel}_2$.

Special kinds of **Bel** functions are very good at representing evidence. These functions are called *simple* and *separable support* functions. **Bel** is a *simple support* function if there exists an $F \subseteq \Theta$ called the focus of **Bel**, such that $\mathbf{Bel}(\Theta) = 1$ and

$$\mathbf{Bel}(A) = \begin{cases} \boldsymbol{s} & \text{if } F \subseteq A \text{ and } A \neq \Theta, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where $\boldsymbol{s}$ is called **Bel**'s *degree of support*.

A *separable support function* is either a simple support function or an orthogonal sum of simple support functions. Separable support functions are very useful when combining evidence from several sources. If **Bel** is a simple support function with focus $F \neq \Theta$, then $\boldsymbol{m}(F) = \boldsymbol{s}$, $\boldsymbol{m}(\Theta) = 1 - \boldsymbol{s}$, and $\boldsymbol{m}$ is 0 elsewhere.

Let F be a focus for two simple support functions with degrees of support $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, respectively. If $\mathbf{Bel} = \mathbf{Bel}_1 \oplus \mathbf{Bel}_2$ then $\boldsymbol{m}(F) = 1 - (1 - \boldsymbol{s}_1)(1 - \boldsymbol{s}_2)$, $\boldsymbol{m}(\Theta) = (1 - \boldsymbol{s}_1)(1 - \boldsymbol{s}_2)$ and $\boldsymbol{m}$ is 0 elsewhere.

Let $\boldsymbol{x}$ be an input vector and N be the number of different classifiers, f^n, $n = 0,1,\ldots,N-1$. It also assumes that each classifier produces an output vector $\boldsymbol{y}^n = \mathbb{R}^K$, $\boldsymbol{y}^n = f^n(\boldsymbol{x})$. Here K is the number of classes (in case of digit recognition, $K = 10$). Suppose that for each classifier f_n and each candidate class k, the computed value $e_k(\boldsymbol{y}^n)$ represents some measurement of evidence for the proposition "$\boldsymbol{y}^n$ belongs to class k". In terms of the Dempster–Shafer theory, these values could be combined according to the theory and the class with the highest evidence is chosen.

4.2. Proposed method of evidences combination

Let $\{\boldsymbol{t}_k\}$ be a subset of the training data corresponding to a class k. Let $\boldsymbol{r}_k^n$ be the mean vector for a set $\{f^n(\boldsymbol{t}_k)\}$ for each classifier f^n and each class k. $\boldsymbol{r}_k^n$ is a reference vector for each class k and $s_k^n = \phi(\boldsymbol{r}_k^n, \boldsymbol{y}^n)$ is the degree of support for class k from classifier n. The value of this function is between 1 and 0 with the maximum when the output vector coincides with a reference vector. A specific form for the function ϕ will be given later. Now the function ϕ is to be transformed into evidence $e^k(\boldsymbol{y}^n)$.

Consider a frame of discernment $\Theta = \{\theta_0,\ldots,\theta_{K-1}\}$, where θ_k is the hypothesis that "$\boldsymbol{y}^n$ belongs to class k". For any classifier f^n and each class k, s_k^n can represent evidence *pro*-hypothesis θ_k, and all s_i^n, with $i \neq k$, can represent evidence *pro* $\neg\,\theta_k$ or *contra* θ_k. s_k^n can be used as a degree of support for a simple support function with focus θ_k. This yields the basic probability assignment

$$\boldsymbol{m}_k(\theta_k) = s_k^n \quad \text{and} \quad \boldsymbol{m}_k(\Theta) = 1 - s_k^n. \tag{19}$$

In a similar manner, s_i^n are degrees of support for simple support functions with a common focus $\neg\,\theta_k$, if $i \neq k$. The combination of this simple support function with focus $\neg\,\theta_k$ is a separable support function with the degree of support $1 - \prod_{i \neq k}(1 - s_i^n)$. The corresponding basic probability assignment is

$$\boldsymbol{m}_{\neg\,k}(\neg\,\theta_k) = 1 - \prod_{i \neq k}(1 - s_i^n) \tag{20}$$

and

$$\boldsymbol{m}_{\neg\,k}(\Theta) = 1 - \boldsymbol{m}_{\neg\,k}(\neg\,\theta_k) = \prod_{i \neq k}(1 - s_i^n). \tag{21}$$

The derivation of evidence for class k is presented in the next section.

4.2.1. Computation of evidence for class k

Let e_k be the evidence for class k. The evidence can be computed by combining the evidence for class k from N classifiers:

$$e_k = \frac{\sum_{0 \leqslant n \leqslant N-1} e_k^n}{N}, \tag{22}$$

where e_k^n is the evidence for class k from classifier n. e_k^n can be computed as follows:

$$e_k^n = \frac{(\text{Support for class } k)(\text{No support for the rest of class})}{(\text{No conflict in classification})}$$

$$= \frac{(\text{Support for class } k)(\text{No support for the rest of class})}{1 - (\text{Support for class } k)[1 - (\text{No support for the rest of class})]}. \tag{23}$$

It is now easier to see that

$$(\text{Support for class } k) = s_k^n \tag{24}$$

and

$$(\text{No support for the rest of class}) = \prod_{i \neq k} (1 - s_i^n). \tag{25}$$

Substituting Eqs. (24) and (25) into Eq. (23) yields

$$e_k^n = \frac{s_k^n \prod_{i \neq k} (1 - s_i^n)}{1 - s_k^n \left[1 - \prod_{i \neq k} (1 - s_i^n)\right]}. \tag{26}$$

The evidence for class k can be computed by substituting Eq. (26) into Eq. (22). The support function for class k is obtained by using the Euclidean distance between $\boldsymbol{r}_k^n$ and $\boldsymbol{y}^n$:

$$s_k^n = 1 - \frac{\left(1 + \|\boldsymbol{r}_k^n - \boldsymbol{y}^n\|^2\right)}{\sum_{0 \leqslant i \leqslant K-1} \left(1 + \|\boldsymbol{r}_k^n - \boldsymbol{y}^n\|^2\right)}. \tag{27}$$

5. Results

Fifteen thousand hand-written digit images obtained from the database of National Institute of Standards and Technology, USA (Garris and Wilkinson, 1992) were used to conduct experiments. The original images are in grey level. These images are pre-processed using Karhunen Loeve (KL) transforms (Gonzalez and Woods, 1992) to produce KL features. These features are then used for the inputs of the classifiers. In the experiment, 4 classifiers and 10 classes (digits) were used. The 4 classifiers are Back-propagation (BP) (Rumelbart et al., 1986), Restricted Coulomb Energy (RCE) (Simpson, 1990), Linear Vector Quantization (LVQ) (Kohonen, 1982) and Contender's Network (CN) (Ng et al., 1995). BP was used because it is the most widely used neural network paradigm. RCE, LVQ and CN were used because they are the *k*-nearest neighbour classifiers normally used for clustering in pattern classification.

Each classifier has 10 output nodes which corresponds to 10 classes of digit. Each output node has a value ranges from 0 to 1. 1 and 0 means the output node is fully activated and not activated, respectively. The maximum output node of the classifier indicates the digit class that the current input belongs to.

Table 1
Performance of individual classifier

	Accuracy rate	Rejection rate
CN	79.96%	0%
RCE	65.38	0%
BP	78.56	0%
LVQ	78.3	0%

Table 2
Accuracy boost of the Combined Classifier by using the data equalisation on combining four classifiers

	Without data equalisation	With data equalisation
Rejection rate	0%	7.5%
Accuracy rate	86.67%	95.23%

Table 1 shows the accuracy rate of individual classifier at 0 rejection rate. There is no effect of data equalisation on the performance of each classifier. This is because data equalisation spreads out the output activation values of each classifier such that the average difference of the output activation value is smaller. It does not change the winning class of each classifier. That means the accuracy rate of each classifier is the same before and after the data equalisation.

Table 2 shows the considerable boost in the CC's accuracy by using the data equalisation on combining four classifiers, i.e., BP, RCE, LVQ and CN. The data equalisation helps in improving contrast among the output activation values of the classifier. The effect is to increase the competitiveness among output nodes of the classifier. When these outputs are sent to CC, it causes higher rejection rate of the CC. The increase in the rejection rate means more unwanted patterns are rejected by the CC. Fig. 2 shows the effect of data equalisation on combining a subset of these four classifiers in the *R-A* plane. As the number of classifier increases, CC gains in accuracy rate at the price of increased rejection rate.

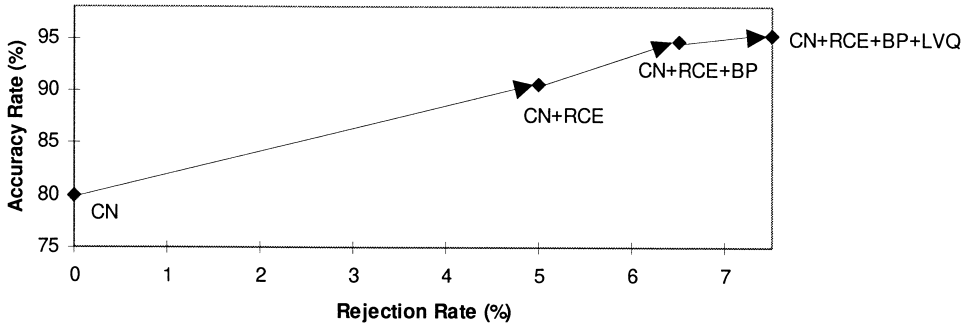

Fig. 2. Position shift of Combined Classifier in *R-A* plane.

6. Conclusion

Part of the motivation for combining the outputs of different classifiers can be derived from the usual statistical technique of lowering the variance of estimates by averaging. Decisions taken by teams can be much better than decisions taken by individuals, provided that suitable methods for combining the outputs are provided. In this paper, a data equalisation and an evidence combination method based on the Dempster–Shafer theory are proposed for the combination. From the experimental results, it is concluded that by introducing the data equalisation before the evidence combination, the accuracy rate of the Combined Classifier is improved. The boost in accuracy rate is because of the rejection which rejects the noisy patterns.

References

Breiman, L., 1996. Bagging predictors. Machine Learning 24 (2), 123–140.

Freund, Y., Schapire, R.E., 1996. Game theory, on-line prediction and boosting. In: Proc. 9th Ann. Conf. on Computational Learning Theory. Desenzano del Garda, Italy, pp. 325–332.

Garris, M.D., Wilkinson, R.A., 1992. Hand-written segmented character database. Technical Report, Special Database 3, HWSC, National Institute of Standards and Technology, 1992.

Gonzalez, R.C., Woods, R.E., 1992. Digital Image Processing. Addison-Wesley, Reading, MA.

Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. Biological Cybernet. 43, 59–69.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation applied to hand-written zip code recognition. Neural Comput. 1, 541–551.

Ng, G.S., Singh, H., 1997. New evidence combination method for multiple pattern classifiers. In: Proc. Joint Pacific Asian Conf. on Expert Systems/Singapore Internat. Conf. on Intelligent Systems, Singapore, pp. 393–400.

Ng, G.S., Erdogan, S.S., Ng, P.W., 1995. Contender's network, a new competitive-learning scheme. Pattern Recognition Letters 16, 1111–1118.

Quinlan, J.R., 1996. Bagging, boosting, and C4.5. In: Proc. Nat. Conf. on Artificial Intelligence, Vol. 1. AAAI, Menlo Park, CA, pp. 725–730.

Rumelbart, D.E., Hinton, G.E., Williams, R.H., 1986. Parallel Distributed Processing Explorations in the Microstructures of Cognition. MIT Press, Cambridge, MA.

Shafer, G., 1976. A Mathematical Theory of Evidence. Princeton University Press, Princeton, NJ, 1976.

Simpson, P.K., 1990. Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations. Pergamon, Oxford.

Xu, L., Krzyzak, A., Suen, C.Y., 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. Systems Man Cybernet. 22 (3), 418–435.