Performance analysis of pattern classifier combination by plurality voting

Xiaofan Lin, Sherif Yacoub, John Burns, Steven Simske
Intelligent Enterprise Technologies Laboratory
HP Laboratories Palo Alto
HPL-2002-150 (R.1)
February 20th , 2003*

E-mail: {xiaofan.lin, sherif.yacoub, john.burns, steven.simske}@hp.com

plurality voting, majority voting, classifier combination, stochastic simulation, classifier evaluation

Plurality voting is widely used in pattern recognition. However, there is little theoretical analysis of plurality voting. In this paper, we attempt to explore the rationale behind plurality voting. The recognition/error/rejection rates of plurality voting are compared with those of majority voting under different conditions. It is demonstrated that plurality voting is more efficient in achieving the tradeoff between rejection rate and error rate. We also discuss some practical problems when applying plurality voting to real-world applications.

* Internal Accession Date Only
Approved for External Publication
To be published in Pattern Recognition Letters, volume 24, 2003

1. Introduction

Originating from the social sciences (Grofman et al., 1983), voting has become a popular system combination technique in various engineering disciplines, especially in pattern recognition (Xu et al., 1992; Ho et al., 1994; Lam and Suen, 1997). The appeal of voting arises from its generality, simplicity, and effectiveness. For most problems with several solutions, voting can be used to improve the system's reliability or accuracy. In the most naive form, voting treats individual systems as black boxes and needs no additional internal information for the implementation. More importantly, in many real-world applications, there is only marginal, if any, performance difference between voting and more advanced combination schemes, which usually require more detailed information from individual systems, greater development efforts, and customization to a specific domain (Lee et al., 1993). Voting is especially advantageous for the combination of some commercial-of-the-shelf (COTS) classifiers, from which we cannot get information beyond the top candidates.

On the other hand, most work is experimental in nature and does not answer why voting is effective and what the theoretical limit of a voting system is. In this respect, Lam and Suen (1997) conducted an in-depth research on the behavior of majority voting, which requires the agreement of more than *half* of the participants to reach a decision. In reality, another common variant of voting is plurality voting, which selects the candidate with the *most* votes. Many people do not distinguish between plurality voting and majority voting, and are more accustomed to the term of "majority voting" even if the underlying criterion is plurality voting. This paper analyzes the performance of classifier combination by plurality voting in terms of recognition/error/rejection rates, and emphasizes the relationship as well as the difference between majority voting and plurality voting.

This paper is organized as follows. In Section 2, the rationale of plurality voting is investigated theoretically. Section 3 introduces the analysis methodology based on stochastic simulation. Section 4 analyzes how the combination system's recognition rate is affected by various parameters such as the number of classifiers, the number of classes, and the recognition rates of individual classifiers. Section 5 discusses how to control the reliability in plurality voting and shows the advantage of plurality voting over majority voting. In Section 6, some practical considerations are addressed to fill the gap between theory and reality. Section 7 gives a summary and suggests future research directions.

2. Rationale behind plurality voting

Plurality voting means that the candidate with the most votes is chosen. But what is the theoretic foundation for this claim? To answer this question, we first model the problem in terms of pattern recognition:

N classifiers $\{E_1, E_2, ..., E_N\}$ are available for a M-class pattern recognition task, in which input object X is classified into one of the M classes $\{C_1, C_2, ..., C_M\}$. Classifier E_i keeps a constant recognition rate p_i for any input object X, that is:

$$P(E_i(X) = C(X)) = p_i \quad (1)$$

where $C(X)$ is the true class that X belongs to, and $E_i(X)$ is the class selected by E_i.

In addition, all other classes have the same probability to be chosen in case of incorrect recognition:

$$P(E_i(X) = C_j) = (1 - p_i)/(M - 1) = e_i \tag{2}$$

where j=1,2,...,M and $C_j \neq C(X)$.

We also assume that each classifier makes its decision independently:

$$P(E_1(X), E_2(X), ..., E_N(X) \mid C(X) = C_j) = \prod_{i=1}^{N} P(E_i(X) \mid C(X) = C_j) \tag{3}$$

where j=1,2,...,M.

With the above formulation, the problem can be analyzed: In order to minimize the error rate of the combination system, the class C_j with the largest *a posteriori* probability should be selected according to the Bayes' rule:

$$P(C(X) = C_j \mid E_1(X), E_2(X), ..., E_N(X))$$

$$= P(E_1(X), E_2(X), ..., E_N(X) \mid C(X) = C_j) P(C(X) = C_j) / P(E_1(X), E_2(X), ..., E_N(X))$$

$$= [\prod_{i=1}^{N} P(E_i(X) \mid C(X) = C_j)] P(C(X) = C_j) / P(E_1(X), E_2(X), ..., E_N(X)) \tag{4}$$

$$P(E_i(X) \mid C(X) = C_j) = \begin{cases} p_i (\text{if } E_i(X) = C_j) \\ e_i (\text{if } E_i(X) \neq C_j) \end{cases} = e_i (p_i / e_i)^{\delta_{ij}(X)} \tag{5}$$

where $\delta_{ij}(X) = \begin{cases} 1 (\text{if } E_i(X) = C_j) \\ 0 (\text{if } E_i(X) \neq C_j) \end{cases}$

$$P(C(X) = C_j \mid E_1(X), E_2(X), ..., E_N(X))$$

$$= [\prod_{i=1}^{N} e_i (p_i / e_i)^{\delta_{ij}(X)}] \, P(C(X) = C_j) / P(E_1(X), E_2(X), ..., E_N(X))$$

Let $Y(X) = (\prod_{i=1}^{N} e_i) / P(E_1(X), E_2(X), ..., E_N(X))$

$$P(C(X) = C_j \mid E_1(X), E_2(X), \ldots, E_N(X))$$

$$= \mathrm{Y(X)}[\, P(C(X) = C_j) \prod_{i=1}^{N} (p_i / e_i)^{\delta_{ij}(X)} \,] \qquad (6)$$

Since Y(X) is the same for every class, the effective decision function is only the second part, whose logarithmic form is:

$$D_j(X) = \ln P(C(X) = C_j) + \sum_{i=1}^{N} \ln(p_i / e_i)\delta_{ij}(X) \qquad (7)$$

The class C_j that maximizes $D_j(X)$ is selected. If each class has the same *a priori* probability and every classifier has the same recognition rate *p*, the decision function becomes:

$$D_j(X) = \ln(1/M) + \ln[(M-1)p/(1-p)] \sum_{i=1}^{N} \delta_{ij}(X) \qquad (8)$$

By removing the class-independent part, we can reduce it to:

$$D_j^{'}(X) = \sum_{i=1}^{N} \delta_{ij}(X) \qquad (9)$$

Eq. (9) is the commonly cited plurality voting rule and Eq. (7) reflects a more generic form: Each classifier can have a different weight and each class has a constant representing its *a priori* probability. From the above analysis, we know that plurality voting as in Eq. (9) is equivalent to the Bayesian criterion under the following conditions:

1) The classifiers are independent of each other as defined in Eq. (3).
2) The misclassifications are evenly distributed among the M-1 residual classes.
3) All of the classifiers have the same recognition rate.
4) The input objects are evenly distributed among all of the classes.

If the last two conditions are not satisfied, the weighted version of Eq. (7) can be used.

In case of a tie using Eq. (7) or (9), we can arbitrarily select one of the classes with the maximum support (we will propose other alternatives in Section 4). Besides, when the recognition rate p is below 1/M, $\ln[(M-1)p/(1-p)]$ is negative in Eq. (8) and thus the "reverse plurality" rule should be used in Eq. (9): The class with the least votes should be selected. This seemingly weird judgment is not difficult to understand. When p is less than 1/M, it means that X is more likely to belong to any other class rather than the class chosen by the classifier. Fortunately, this rarely happens in reality because well-developed classifiers can easily pass that threshold.

The independence assumption is not easy to meet in practical pattern recognition applications. More commonly, all of the classifiers are prone to make mistakes simultaneously on some very difficult samples. Taking this factor into account, we propose the modified model, which is composed of both the independent and dependent situations:

a) The N classifiers will simultaneously misrecognize a sample with a probability of α;

b) Otherwise (with a probability is 1-α), the N classifiers will perform independently according to Eq. (2) and Eq. (3).

Under this model, the overall recognition rate of Classifier E_i is (1-α) p_i.

In the dependent situation, all combination schemes, including using Eq. (7), will achieve the same recognition rate of zero because none of the classifiers selects the correct class. On the other hand, the optimal decision function is still Eq. (7) in the independent situation. In conclusion, Eq. (7) is the overall optimal decision function even under this mixed model. For simplicity, we will concentrate on the independent situation in the following analysis.

3. Stochastic simulation

Unlike majority voting (Lam and Suen, 1997), it is difficult to derive a closed formula applicable to different M's and N's for calculating the plurality voting system's accuracy, because plurality can be obtained through many different patterns. So a stochastic simulation is used to quantitatively analyze the performance.

A real-world pattern classification system consists of two major steps: feature extraction and classification. In the simulation, we treat each classifier E_j as a black box with a single parameter: the recognition rate p_j, according to the model introduced in Section 2. Fig. 1 shows how the individual classifiers are simulated. For example, to simulate the event "Correctly recognize the input with a probability of 0.8", the computer generates a random float number f in the range of [0,1], and if $f \leq 0.8$ the input is considered correctly recognized and otherwise it is misrecognized.

Fig. 2 shows how the experiments are conducted. In each test, a large number of samples (for example, MAXCOUNT=10,000,000) are randomly generated and fed into the individual classifiers. Plurality voting is then executed on their outputs. With a large amount of samples, the simulation results are expected to be very close to those strictly calculated from probability theory. In this way, we can conveniently study how different parameters influence the voting performance without an explicit formula. Besides, this kind of simulation can also be applied to other combination strategies such as majority voting and strict plurality voting, which will be introduced in Section 5.

Table 1 displays 18 examples obtained from the simulation. The three classifiers have recognition rates of 0.8, 0.9, and 0.8 respectively. The three classes are ‘a’, ‘b’, and ‘c’. As shown in the table, the three classifiers make 3, 2, and 4 errors respectively. The plurality voting makes one error.

4. Recognition rate of plurality voting

This section focuses on the recognition rate of plurality voting, especially in contrast to majority voting. Table 2, Table 3, and Table 4 show the simulation results for M=50, 3, and 2 respectively. In the shaded areas, “reverse plurality” voting is adopted due to each individual classifier’s poor recognition rate.

First, three observations can be made under the condition that all of the classifiers have the same recognition rate *p*:

1) The voting system’s recognition rate keeps rising or at least stays the same with more classifiers.

   If $p \neq 1/M$, the plurality voting system’s recognition rate will approach 1 with sufficiently large N. Of course, the “reverse plurality” rule is used when $p < 1/M$. When $p = 1/M$, the classifier completely randomly selects a class without providing any information. Otherwise, with more classifiers, more information about the input object is obtained and the recognition rate $P_C(N)$ increases. When M is larger than 2, $P_C(N)$ is monotonically increasing with N. When M is 2, $P_C(N)$ follows an interesting stepwise pattern: $P_C(2N) < P_C(2N+1) = P_C(2N+2)$. In contrast, the zigzag-shaped recognition rate curve is characteristic of majority voting: $P_C(2N+1) > P_C(2N-1) > P_C(2N)$, which means that adding an extra classifier to a pool of odd number of classifiers actually decreases the recognition rate (Lam and Suen, 1997). The zigzag pattern in majority voting is caused by more rejections with an even number of classifiers. Fig. 3 compares the different patterns.

2) When individual classifiers perform decently (p>1/M) and with the same p and N, the recognition rate increases with increasing M (see the curves for the simple plurality voting in Fig. 5).

   With more classes, errors will be more scattered among the M-1 incorrect classes, and consequently the chance for the correct class to stand out in plurality voting will be greater. This characteristic does not apply to majority voting, in which the number of classes is irrelevant to the recognition rate. Majority voting accepts the results if and only if one class receives more than half of the total votes. Once the threshold is exceeded, any other class will receive fewer votes no matter how the remaining votes are distributed.

3) Combining three comparable classifiers provides a good start. If all of the classifiers have the same recognition rate, we can see the largest gain in recognition rate by using three classifiers compared with using only one classifier. The improvement becomes more gradual with more classifiers. So it is wise to start with combining three classifiers and to incorporate more classifiers only if the three-classifier system cannot achieve the required recognition rate.

Second, if the classifiers have different recognition rates, the situation can be more complicated because the classifiers should be weighted based on Eq. (7). In the extreme, if the best classifier is much more accurate than the others and thus has a weight larger than the sum of the other classifiers' weights, the optimal decision can be dominated by only the best classifier — the addition of other classifiers does not improve the accuracy. Let us look at the following example. Three classifiers with $p_1 = 0.99, p_2 = 0.8, p_3 = 0.9$,respectively, are used. There are five classes with an equal *a priori* probability of 0.2. The optimal weights for the three classifiers are 4.595, 1.386 and 2.303 respectively. The simulation shows that using Eq. (9) results in a recognition rate of about 0.983, which is lower than 0.99 achieved by the best individual classifier.

5. Reliability of plurality voting

The above analysis is concentrated on the absolute recognition rate of plurality voting without any rejections. In case of a tie using Eq. (7) or (9), we arbitrarily make the selection. However, it is often desirable to measure the system's reliability — the probability of the decision to be correct on a given input object. A direct use of the reliability metric is to reduce the error rate by rejecting suspicious results. In theory, Eq. (6) is the optimal measurement of reliability in defining the *a posteriori* probability of the winning class given the results of N classifiers. Because the sum of the *a posteriori* probabilities of all of the M classes is 1, it can be given as:

$$P(C(X) = C_j \mid E_1(X), E_2(X), \ldots, E_N(X))$$

$$= P(C(X) = C_j) \prod_{i=1}^{N} (p_i / e_i)^{\delta_{ij}(X)} / \sum_{k=1}^{M} [P(C(X) = C_k) \prod_{i=1}^{N} (p_i / e_i)^{\delta_{ik}(X)}] \qquad (10)$$

If all of the classes have the same *a priori* probabilities and all of the classifiers have the same recognition rate p, we can get:

$$P(C(X)=C_j \mid E_1(X), E_2(X), ..., E_N(X))$$

$$= (p/e)^{\sum_{i=1}^{M} \delta_{ij}(X)} / \sum_{k=1}^{M} (p/e)^{\sum_{i=1}^{N} \delta_{ik}(X)}$$

$$= 1/[1+\sum_{k=1}^{M,k\neq j} (p/e)^{\sum_{i=1}^{N} \delta_{ik}(X) - \sum_{i=1}^{N} \delta_{ij}(X)}] \quad (11)$$

It can be seen that the classification reliability is decided by the vote difference between the winning class and the other classes. If the result is accepted only when the winning class receives at least r more votes than the closest competitor, the reliability is then bounded by:

$$P(C(X)=C_j \mid E_1(X), E_2(X), ..., E_N(X))$$

$$\geq 1/[1+(M-1)(p/e)^{-r}] \quad (12)$$

So the desired reliability can be achieved through enforcing different r. In contrast, such kind of flexibility is not available in majority voting. We call plurality voting with r=0 “simple plurality” voting and that with r>0 “strict plurality” voting.

A significant advantage of plurality voting over majority voting is its higher rejection efficiency, that is, making fewer rejections when reducing the error rate to the same level. Table 7 gives several examples under the condition that p=0.85/0.90/0.95, N=7, and M=12. When r is 2, plurality voting has a lower or the same rejection rate (defined as the percentage of samples rejected by the combination system), but still achieves a lower error rate than majority voting. The experimental results in handwriting recognition also support this conclusion (Xu et al., 1992).

In Section 2, we have examined how different N and M affect the performance of simple plurality voting. Now a similar study is furnished for strict plurality voting with r=1 in Table 6 and Table 7 for M=3 and 50 respectively. When M=2 and r=1, plurality voting is equivalent to majority voting and the results can be directly obtained from the paper by Lam and Suen (1997). Several characteristics of strict plurality voting can be noticed from the Fig. 4 and Fig. 5:

1) When M is large (for example, 50), $P_C(N)$ is monotonically increasing with N.
2) When M is 2, it follows the zigzag pattern.
3) When M is in the middle of the spectrum, there is no fixed relationship between $P_C(2N)$ and $P_C(2N-1)$. For example, when M=3, $P_C(4) = P_C(3)$ but $P_C(6) > P_C(5)$.
4) When other conditions hold the same, the recognition rate of strict plurality voting is lower than that of the simple plurality voting because of the introduction of rejections. On the other hand, it is higher than (when M>2) or the same as (when M=2) that of majority voting.
5) With the same *p* and N, the recognition rate of strict plurality voting is monotonically increasing with M when N>3. When N=3, the recognition rate is independent of M.

6. Reality check

In this section, we examine plurality voting in the context of several typical applications such as Optical Character Recognition (OCR), Part-of-speech (POS) tagging, Automatic Speech Recognition (ASR), and Handwriting Recognition. The first two applications come from our previous research, and the last two are based on the work by other researchers. Table 8 compares the actual results (cited from the literature) against the theoretical limits (calculated with the simulation described above). The gap between theory and practice is attributed to the fact that some assumptions made in Section 2 do not completely reflect the reality:

First, in the foregoing analysis, it is assumed that a classifier randomly selects a wrong class among the M-1 residual classes in case of an error. In practice, for a given object, the classifiers usually concentrate the decision within a few choices, instead of all of the classes. For example, in OCR, if the input character is “3” and the classifier recognizes it incorrectly, the wrong choice is often within the scope of the subset {“5”, “s” “S”} rather than the set of 80-100 possible characters. Halteren et al. (2001) also noticed that in Part-of-speech (POS) tagging there are many confusing POS tag pairs. In consequence, a modified M should be used in place of the number of all possible classes. That is why small M’s are chosen in Table 8. As mentioned earlier, a smaller M means a lower recognition rate in combination (see Fig. 5). Thus, the actual error rate reduction will be less than the ideal situation in which the errors are evenly distributed among all residual classes. The confusion matrices of the classifiers can be used to estimate the effective M.

Second, for most pattern recognition tasks, it is a challenge to design independent classifiers each with excellent accuracy. The classifiers available are usually statistically correlated even though they are developed by different researchers, extract different features, and follow different classification schemes. The direct outcome is a far less spectacular error rate reduction than that achievable with independent classifiers. At the end of Section 2, we discussed a mixed model. Under that model, the lowest error rate of the combination system will be α instead of zero. In practice, we define the following Independence Factor (IF) to evaluate how independent the classifiers are:

$$IF = \frac{P_{comb} - \max_{1 \le i \le N} P_i}{P_{theory} - \max_{1 \le i \le N} P_i}$$

where P_{comb} is the voting system's actual recognition rate, P_{theory} is the theoretical limit on the recognition rate under the independence model, and $\max_{1 \le i \le N} P_i$ represents the best recognition rate of individual classifiers.

The Independence Factor captures how close the real improvement is to the theoretical limit. Table 8 shows that it varies for different domains and is always less than 1, which means that the individual classifiers are not really independent. Here we see the reason behind the popular wisdom in the pattern recognition community: "Design classifiers that are as independent as possible, and then combine them." Another requirement is that the individual classifiers should have comparable accuracies, as otherwise some classifiers will play a limited or even no role in the combination (see Eq. (7)). The two requirements together pose a great challenge for researchers. In many applications only one strategy proves effective and individual methods are just variants along the same general direction. For example, most successful speech recognition systems are along the line of Hidden Markov Models (HMM) and only differ in technical details, and accordingly the Independence Factor is very low. Furthermore, as a domain becomes mature, each participating classifier "borrows" successful techniques from others or simply integrates more information internally. Although the individual classifiers achieve higher accuracy in this way, they are more correlated and drive the

Independence Factor lower. This trend has been observed in various areas such as OCR and ASR. For example, it can be seen from Table 8 that the actual recognition rate achieved by the plurality voting of three high-performance commercial OCR engines is 0.998, which is far less than the theoretical limit of 0.99988 by one order of magnitude. Similarly, the Independence Factor is only 0.20 for the speech recognition application cited in Table 8.

Third, some classifiers do not actually recognize input objects separately. In the above discussion, we assume that each input object is sent into individual classifiers and recognized. For some real-world problems, however, we can only send a whole collection of objects into each classifier. For example, a whole page image containing many characters is passed to each OCR engine and the speech signal of at least a whole sentence is input to each ASR engine. The benefit of doing so is that the recognition rate can be improved by utilizing the context information in neighboring objects, which is extremely critical for tasks like POS tagging and ASR. The downside is that significant effort is needed on object alignment even before reaching the stage of combining the results of individual objects (Lin, 2002a). In addition, the model introduced in Section 2 does not consider such contextual relationship among neighboring objects.

7. Conclusions

This paper addresses the performance analysis of classifier combination using plurality voting. We have achieved the following goals:

1) Theoretically justify the plurality voting decision criteria and explain the implicit assumptions.

2) Analyze how different parameters can affect the asymptotic behavior of plurality voting.
3) Offer an insight into the reliability control issue for plurality voting and establish the strict plurality voting as a more effective way to reduce error rate than majority voting.
4) Define the Independence Factor to measure differences between practice and theory, and address practical issues in plurality voting.

In most cases, the analysis is conducted with majority voting as a reference. Table 9 illustrates the relationship and difference between them.

In this paper, we have demonstrated that the combination of independent classifiers can result in dramatic accuracy improvement. In theory, however, independent classifiers may not be the best choice. Kuncheva and Duin (2000) pointed out that the combination of “negatively dependent” classifiers could deliver an even better recognition rate than that of independent classifiers. The “negatively dependent” classifiers are intelligently complementary to each other. When a classifier has difficulty recognizing an object, the other classifiers are more likely to recognize it correctly. Negatively dependent classifiers actually perform sub-clustering by dividing the overall domain into several regions, one for each classifier to excel in. That is an interesting area for future work.

References

Fiscus, J.G., 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop, Santa Barbara, USA, 347-352.

Grofman, B., G. Owen, S. Field, 1983. Thirteen theorems in search of the truth. Theory Decision 15, 261-278.

Halteren, H.V., J. Zavrel, W. Daelemans, 2001. Improving wordclass tagging through combination of machine learning systems. Computational Linguistics 27 (2), 199-230.

Ho, T.K., J. J. Hull, S. N. Srihari, 1994. Decision combination in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(1), 66-75.

Kuncheva, L.I., C.J. Whitaker, C.A. Shipp, 2000. Is independence good for combining classifiers?. Proceedings of International Conference on Pattern Recognition, Barcelona, Spain.

Lam, L., C.Y. Suen, 1997. Application of majority voting to pattern recognition: An analysis of the behavior and performance. IEEE Transactions on Systems, Man, and Cybernetics 27(5), 553-567.

Lee, D.S., S. N. Srihari, 1993. Handprinted digit recognition: A comparison of algorithms. Proceedings of 3rd International Workshop on Frontiers in Handwriting Recognition, Buffalo, USA, 153-162.

Lin, X., 2002a. Reliable OCR solution for digital content re-mastering. Document Recognition and Retrieval IX, Proceedings of SPIE 4670, San Jose, USA, 223-231.

Lin, X., 2002b. Part-of-speech tagging of imperfect text. HP Labs Technical Report HPL-2002-7.

Xu, L., A. Krzyzak, Ching Y. Suen, 1992. Method of combining multiple classifiers and their application to handwriting recognition. IEEE Transactions on Systems, Man, and Cybernetics 22(5), 418-435.

Table 1
Examples of stochastic simulation (Incorrect results are in bold font)

No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Ground Truth	a	b	a	a	b	c	a	c	c	a	b	b	c	a	b	b	c	a
$E_1(p_1 = 0.8)$	a	b	a	a	b	c	a	c	c	**b**	b	b	c	a	b	b	**b**	**b**
$E_2(p_2 = 0.9)$	a	b	a	a	b	c	a	c	**a**	a	b	b	**b**	a	b	b	c	a
$E_3(p_3 = 0.8)$	a	b	a	a	b	**b**	a	**a**	c	a	b	**c**	c	a	b	b	c	**b**
Voting	a	b	a	a	b	c	a	c	c	a	b	b	c	a	b	b	c	**b**

Table 2
Recognition rates of a simple plurality voting system with different *p*'s and N's (M=50)

p	N								
	2	3	4	5	6	7	8	9	10
0.3	0.30114	0.35862	0.4428	0.52841	0.60387	0.66585	0.71789	0.76138	0.79693
0.6	0.60218	0.73979	0.85286	0.91931	0.95532	0.97636	0.98634	0.99185	0.99553
0.7	0.70314	0.84647	0.93112	0.97075	0.98691	0.99383	0.99717	0.99862	0.9994
0.8	0.80101	0.92933	0.97684	0.99253	0.99764	0.99928	0.99981	0.99988	0.99996
0.9	0.90138	0.98117	0.99666	0.99936	0.99981	0.99997	1	1	1
0.95	0.94976	0.99502	0.99949	0.99995	0.99999	1	1	1	1

Table 3
Recognition rates of a simple plurality voting system with different *p*'s and N's (M=3)

p	N								
	2	3	4	5	6	7	8	9	10
0.3	0.36769	0.37346	0.37953	0.387	0.39277	0.3974	0.40265	0.40713	0.41045
0.6	0.60042	0.69609	0.73449	0.76892	0.80298	0.83144	0.85027	0.87176	0.8888
0.7	0.69984	0.81497	0.84992	0.88677	0.91334	0.93516	0.94795	0.96071	0.96938
0.8	0.79994	0.91222	0.93374	0.96145	0.97468	0.98481	0.9898	0.99355	0.99581
0.9	0.90007	0.9763	0.98423	0.99454	0.99715	0.99891	0.99937	0.99974	0.99988
0.95	0.95008	0.99389	0.99615	0.99928	0.99968	0.99991	0.99997	0.99999	1

Table 4
Recognition rates of a simple plurality voting system with different *p*'s and N's (M=2)

p	N								
	2	3	4	5	6	7	8	9	10
0.3	0.70216	0.78255	0.78397	0.83567	0.83596	0.87296	0.87314	0.90017	0.90251
0.6	0.59957	0.64833	0.6485	0.6826	0.68306	0.71065	0.71158	0.73251	0.7345
0.7	0.69817	0.78448	0.78386	0.83771	0.836	0.87404	0.87328	0.9009	0.90233
0.8	0.80113	0.89708	0.89734	0.94247	0.94226	0.9671	0.96743	0.9804	0.98085
0.9	0.89947	0.97193	0.97127	0.99191	0.99149	0.99714	0.99741	0.99902	0.99914
0.95	0.94994	0.99272	0.99264	0.99882	0.99982	0.99982	0.99981	0.99997	0.99997

Table 5
Rejection efficiency of plurality voting vs. majority voting (M=12, N=7)

	p=0.85		p=0.9		p=0.95	
R	Rejection Rate (%)	Error Rate (%)	Rejection Rate (%)	Error Rate (%)	Rejection Rate (%)	Error Rate (%)
0	0	0.2460	0	0.0131	0	0.0037
1	0.356	0.0534	0.020	0.0058	0.0008	0.0032
2	**0.668**	**0.0043**	**0.154**	**0.0032**	**0.012**	**0.0031**
3	2.777	0.0034	0.862	0.0030	0.107	0.0028
Majority Voting	1.203	0.0043	0.273	0.0032	0.020	0.0032

Table 6
Recognition rates of a strict plurality voting system with different *p*'s and N's (M=50, r=1)

p	N								
	2	3	4	5	6	7	8	9	10
0.3	0.09059	0.21775	0.34066	0.4533	0.54194	0.61199	0.66099	0.70627	0.74422
0.6	0.35704	0.64725	0.81405	0.89939	0.94267	0.9659	0.97988	0.98736	0.99353
0.7	0.49264	0.78087	0.91009	0.96066	0.9823	0.99145	0.9956	0.99812	0.99906
0.8	0.64116	0.89613	0.96963	0.99003	0.99659	0.99881	0.99955	0.99993	0.99992
0.9	0.81089	0.97284	0.99533	0.99894	0.99987	0.99998	0.99999	1	1
0.95	0.90242	0.99265	0.99923	0.99991	0.99998	1	1	1	1

Table 7
Recognition rates of a strict plurality voting system with different *p*'s and N's (M=3, r=1)

p	N								
	2	3	4	5	6	7	8	9	10
0.3	0.24527	0.25684	0.20922	0.29233	0.30272	0.27434	0.32226	0.33131	0.31073
0.6	0.35991	0.64715	0.64815	0.68285	0.75265	0.78346	0.7966	0.83848	0.85639
0.7	0.49036	0.78382	0.78266	0.83752	0.88279	0.91015	0.92511	0.94707	0.95795
0.8	0.64012	0.89592	0.89612	0.94201	0.96239	0.97735	0.98356	0.99086	0.99376
0.9	0.80907	0.9721	0.97182	0.99135	0.99496	0.99823	0.99901	0.99966	0.99981
0.95	0.90276	0.99268	0.99270	0.99887	0.99934	0.99986	0.99994	0.99999	0.99999

Table 8
Applications using plurality voting

No	1	2	3	4
Source	Lin, 2002a	Lin, 2002b	Fiscus, 1997	Xu et al., 1992
Domain	OCR	POS Tagging	ASR	Handwriting Recognition
Recognition Rates of Individual Classifiers (Best Result in Bold)	0.9887 0.9945 **0.9971**	0.9340 **0.9345** 0.8935	**0.551** 0.549 0.513 0.511 0.498	0.8605 0.9310 0.9295 **0.9390**
Plurality Voting (Actual)	0.9983	0.9595	0.603	0.9890
Plurality Voting (Theoretical)	0.99988 (M=3)	0.9827 (M=2)	0.8068 (M=10)	0.9938 (M=3)
Independence Factor	0.4317	0.5187	0.203	0.9124

Table 9
Plurality voting vs. majority voting

	Simple Plurality Voting	Strict Plurality Voting	Majority Voting
Decision Criterion	Most Votes	Most Votes with at Least Margin of r (r>0)	At Least Half of Total Votes
Analysis Method	Monte Carlo Method	Monte Carlo Method	Closed Formula
Effect of M on Recognition Rate	Monotonically Increasing	N>3: Monotonically Increasing N=3, Not Affected	Not Affected
Effect of N on Recognition Rate	Stepwise Pattern	No Fixed Pattern	Zigzag Pattern
Reliability Control	Not Available	Flexible	Fixed
Relationship		Equivalent When M=2 and r=1	

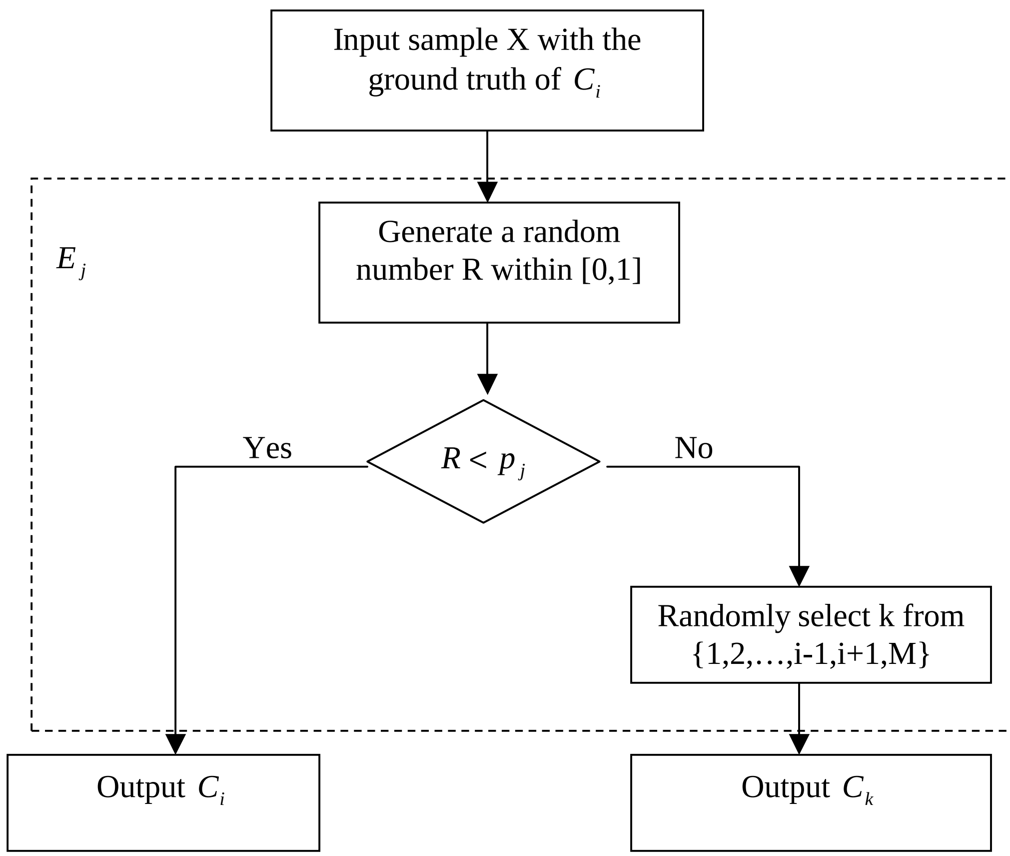

Fig. 1. Simulation of E_j with a recognition rate of p_j

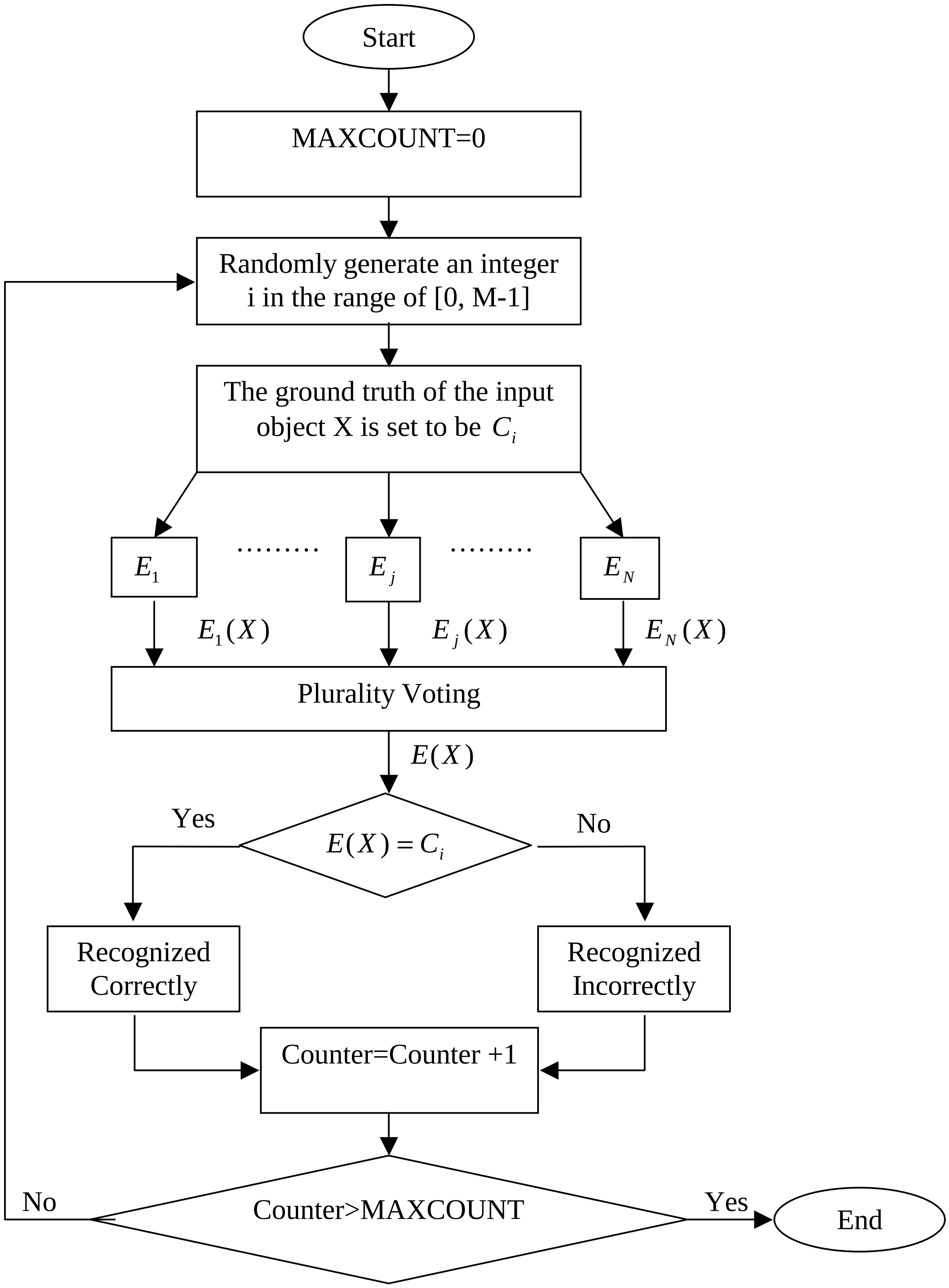

Fig. 2. Workflow of the experiment

(MAXCOUNT is the total number of tests, and please refer to Fig. 1 for the configuration of E_j)

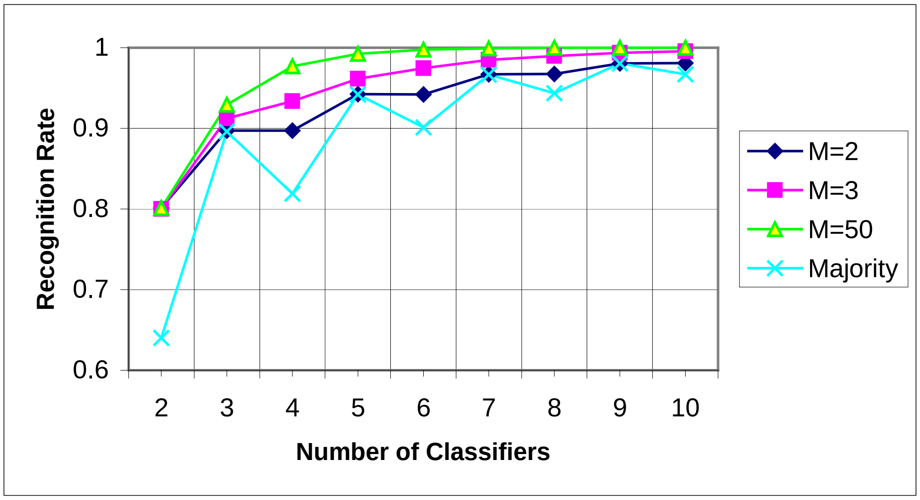

Fig. 3. Relationship between recognition rates and numbers of classifiers
(Simple plurality voting with M=2,3,50 vs. majority voting, p=0.8)

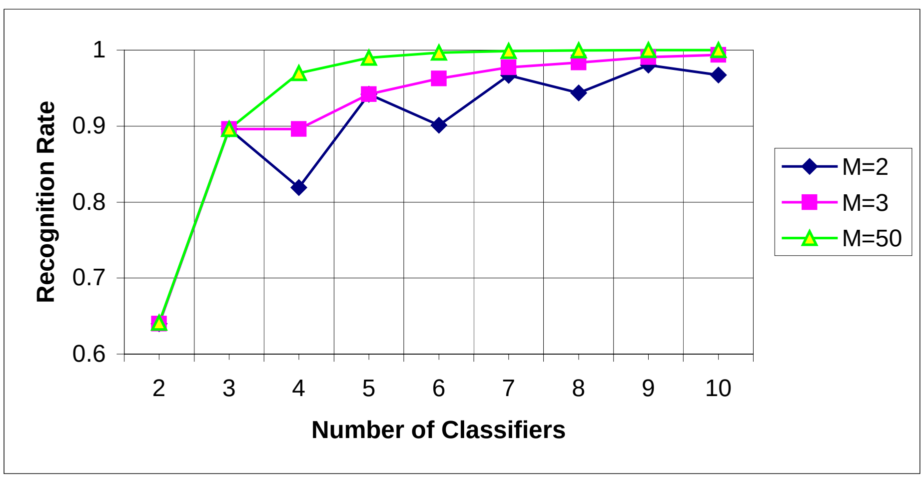

Fig. 4. Relationship betweens recognition rates and numbers of classifiers
(Strict plurality voting with M=2,3,50 and r=1, p=0.8)

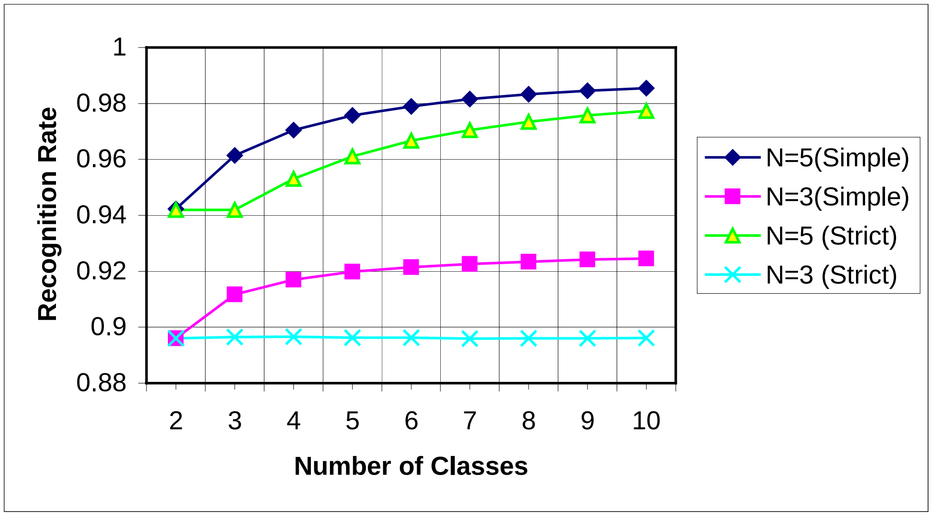

Fig. 5. Relationship between recognition rates and numbers of classes
(Simple plurality voting with N=3 and 5, strict plurality voting (r=1) with N=3 and 5, p=0.8)