# Voting Methods For Multiple Autonomous Agents

J.R. Parker
Laboratory for Computer Vision
Department of Computer Science
Univeristy of Calgary
Calgary, Alberta, Canada

Abstract

The use of many diverse algorithms simultaneously applied to a single problem improves the robustness of the solution. This is certainly true of handprinted character recognition. The problem addressed here is that of combining the results from many agents to give a single result that represents a synthesis of the component agents.

The recognition of hand printed characters has been the subject of a great deal of research due to the interesting nature of the problem, and to the enormous utility of a solution. The problem is a difficult one, at least partly because each individual writer produces a unique set of characters *each time they write*. To a computer the differences are large enough to pose a problem.

Given the number and variety of methods offered for hand printed character recognition, it may very well be that there is no single method that can be called the ‘best’. Many diverse algorithms each have strengths and weaknesses, good ideas and bad. One way to take advantage of this variety is to apply many methods to the same recognition task, and have a scheme to merge the results; this should be successful over a wider range of inputs than would any individual method[PARK94]. The weaknesses should, in an ideal situation, more or less cancel out rather than reinforcing each other giving high recognition rates under many sets of conditions.

This situation occurs generally in artificial intelligence, where combining the results from multiple autonomous agents has many applications. Here a collection of voting schemes are compared and used to join the results from five hand printed digit recognition algorithms, although the method is general enough to be applied to a variety of multiple agent problems. Each of the five methods have been used to classify a set of 1000 digits; the problem addressed here is that of computing a single classification in each case, given five individual classifications.

## 1 Merging multiple methods

In the general case a classifier can produce one of three kinds of classification. The simplest and probably the most common is a simple expression of the class determined for the data object. For a digit classification scheme this would mean that the classifier would simply state ‘this is a FIVE’, for example; this will be called a *type 1* response [XU92]. A classifier may also produce a ranking of the possible classes for a data object. In this case, the classifier may say: ‘this is most likely a FIVE, but could be a THREE, and is even less likely to be a FOUR.’ The number of classes ranked may well vary from case to case, and probabilities are not associated with the ranking. This will be called a *type 2* response. Finally, a classifier may give a probability or other such confidence rating to each of the possible classes. This is the most specific case of all, since either a ranking or a classification can be produced from it. In this case, each possible digit would be given a confidence number which can be normalized to any specific range. This will be called a *type 3* response.

Whatever the type of response, a multiple classifier must deal with three important problems:

1) The response of the multiple classifier must be the best one given the results of the individual classifiers. It should in some logical way represent the most likely true classification, even when presented with contradictory individual classifications.

2) The classifiers in the system may produce different types of response. These must be merged into a coherent single response.

3) The multiple classifier must yield the correct result more often than any of the individual classifiers, or there is no point.

The first problem above has various potential solutions for each of the possible type of response, and these will be dealt with first.

## 2 Merging type 1 responses

Given that the output of each of the classifiers is a single, simple classification value, the obvious way to combine them is by using a voting strategy. A majority voting scheme can be expressed as follows: let $C_i(x)$ be the result produced by classifier *i* for the digit image *x*, where there are *k* different classifiers in the system; then let *H(x,d)* be the number of classifiers giving a classification of *d* for the digit image *x*, where *d* is one of *{0,1,2,3,4,5,6,7,8,9}*. Then

$$E(x) = \begin{cases} j & \text{if } \max(H(x,i))=H(x,j) \text{ and } H(x,j)>\frac{k}{2} \\ 10 & \text{otherwise} \end{cases}$$

This is called a *simple majority vote* (SMV). An easy generalization of this scheme replaces the constant k/2 in the above expression with k*α for 0<=α<=1 [XU92]. This permits a degree of flexibility in deciding what degree of majority will be sufficient, and will be called a *weighted majority vote* (WMV). This scheme can be expressed as:

$$E(x) = \begin{cases} j & \text{if } \max(H(x,i))=H(x,j) \text{ and } H(x,j)>\alpha k \\ 10 & \text{otherwise} \end{cases}$$

Neither of these takes into account the possibility that all of the dissenting classifiers agree with each other. Consider the following cases: in case A there are ten classifiers, with six of them supporting a classification of ‘6’, one supporting ‘5’, one supporting ‘2’ and two classifiers rejecting the input digit. In case B, using the same ten classifiers, six of them support the classification ‘6’ and the other four all agree that it is a ‘5’. Do cases A and B both support a classification of ‘6’, and do they do so equally strongly?

One way to incorporate dissent into the decision is to let *max1* be the number of classifiers that support the majority classification *j* (max1 = *H(x,j)*), and to let *max2* be the number supporting the second most popular classification *h* (max2 = *H(x,h))*. The the classification becomes:

$$E(x) = \begin{cases} j & \text{if } \max(H(x,i))=H(x,j) \text{ and max1-max2 } \geq \alpha k \\ 10 & \text{Otherwise} \end{cases}$$

where α is between 0.0 and 1.0. This is called a *dissenting-weighted majority vote* (DWMV).

For the five classifier system being discussed, the SMV strategy gave the following resulkts:

**Correct:** 994 **Incorrect:** 2 **Rejected:** 4

This is in spite of poor results (76% recognition) from the neural net classifier (#5) on nines - indeed, 100% of the nines are recognized by the multiple classifier. This begs the question “what is the contribution of any one classifier to the overall result?” To determine this for the SMV case is simple. The multiple classifier can be run using any four of the five individual classifiers, and the results can be compared against the five classifier case above to determine whether the missing classifier assisted in the classification. The results are:

| | All | Omit #1 | Omit #2 | Omit #3 | Omit #4 | Omit #5 |
|---|---|---|---|---|---|---|
| **Correct** | 994 | 990 | 988 | 989 | **995** | 992 |
| **Incorrect** | 2 | 2 | 3 | 3 | 1 | 3 |
| **Rejected** | 4 | 8 | 9 | 8 | 4 | 5 |

In all but one case where a classifier is omitted the rejection rate increases, and sometimes the error rate increases as well. The decrease in the error rate that occurs when classifier 4 is omitted is evidence supporting its removal.

Evaluation of WMV is a little more difficult, requiring an assessment of the effect of the value of α on the results. A small program was written that varied α from 0.05 to 0.95, classifying all sample digits on each iteration. This process was then repeated five more times, omitting one of the classifiers each time to again test the relative effect of each classifier on the overall success. With this much data a numerical value is needed that can be used to assess the quality of the results. The recognition rate could be used alone, but this does not take into account that a rejection is much better than a misclassification; both would count against the recognition rate. A measure of *reliability* can be computed as:

$$\text{Reliability} = \frac{\text{Recognition}}{100\% - \text{Rejection}}$$

The reliability value will be low when few misclassifications occur. Unfortunately, it will be high if recognition is only 50%, with the other 50% being rejections. This would not normally be thought of as acceptable performance. A good classifier will combine high reliability with a high recognition rate; in that case, why not simply use the product *reliability*recognition* as a measure of performance? In the 50/50 example above this measure would have the value 0.5: reliability is 100% (1.0) and recognition is 50% (0.5). In a case where the recognition rate was 50%, with 25% rejections and 25% misclassifications, this measure will have the value 0.333, indicating that the performance is not as good. The value *reliability*recognition* will be called *acceptability.*The first thing that should be done is to determine which value of $\alpha$ gives the best results, and this is more accurately done when the data is presented in tabular form:

Table 1: Acceptability of the Multiple Classifier Using a Weighted Majority Vote

| Alpha | All 5 Used | Omit #1 | Omit #2 | Omit #3 | Omit #4 | Omit #5 |
|---|---|---|---|---|---|---|
| 0.05 | 0.994 | 0.982 | 0.978 | 0.984 | 0.992 | 0.986 |
| 0.25 | 0.994 | 0.984 | 0.979 | 0.985 | 0.993 | 0.987 |
| 0.50 | 0.992 | 0.967 | 0.963 | 0.973 | 0.978 | 0.971 |
| 0.75 | 0.956 | 0.817 | 0.819 | 0.816 | 0.823 | 0.817 |

Duplicate rows are not shown in the table. From this information it can be concluded that a should be between 0.45 and 0.5, for in this range the acceptability peaks without causing a drop in recognition rate. It can also be seen that the omission of classifier #4 causes the smallest decrease in acceptability, but that it does appear to make a positive contribution to the overall system in the WMV system.

DWMV also uses the $\alpha$ parameter, and can be evaluated in a fashion identical to what has just been done for WMV. The optimal value of $\alpha$, obtained from Table 2, was found to be 0.25. The table also supports the removal of classifier #4, since the acceptability increases very slightly when that classifier is removed from the system. If the best overall multiple scheme also supports the removal of classifier #4 then it will be discarded.

Table 2: Acceptability of the Multiple Classifier Using a Dissenting Weighted Majority Vote

| Alpha | All 5 | Omit #1 | Omit #2 | Omit #3 | Omit #4 | Omit #5 |
|---|---|---|---|---|---|---|
| 0.05 | 0.993 | 0.988 | 0.985 | 0.986 | 0.994 | 0.989 |
| 0.25 | 0.985 | 0.988 | 0.985 | 0.986 | 0.994 | 0.989 |
| 0.30 | 0.985 | 0.969 | 0.965 | 0.973 | 0.983 | 0.975 |
| 0.45 | 0.960 | 0.969 | 0.965 | 0.973 | 0.983 | 0.975 |
| 0.55 | 0.960 | 0.857 | 0.859 | 0.842 | 0.877 | 0.868 |
| 0.65 | 0.832 | 0.857 | 0.859 | 0.842 | 0.877 | 0.868 |
| 0.80 | 0.832 | 0.817 | 0.819 | 0.816 | 0.823 | 0.817 |
| 0.85 | 0.784 | 0.817 | 0.819 | 0.816 | 0.823 | 0.817 |

## 3 Converting between response types

Before proceeding to analyze methods for merging type 2 responses (ranks) it would be appropriate to discuss means of converting one response type to another. In particular, not all of the classifiers yield a rank ordering, and this will be needed before merging the type 2 responses with those of types 1 and 3:

**Type 3 to Type 1:** Select the class having the maximum confidence rating as the response.

**Type 3 to Type 2:** Sort the confidence ratings in descending order. The corresponding classes are in rank order.

**Type 2 to Type 1**: Select the class having the highest rank as the type 1 response.

Converting a type 1 response to a type 3 cannot be done in a completely general and reliable fashion. However, an approximation can be had based on the measured past performance of the particular algorithm. Each row in the confusion matrix represents the classifications actually encountered for a particular digit with that classifier expressed as a probability, and the columns represent the other classifications possible for a specified classification; this latter could be used as the confidence rating. The conversions from type 1 can be expressed as:

**Type 1 to Type 3:** Compute the confusion matrix *K* for the classifier. If the classification in this case is *j*, then first compute:

$$s = \sum_{i=0}^{9} K(i, j)$$

Now compute the type 3 response as a vector V, where

$$V(i) = \frac{K(i, j)}{s}$$

**Type 1 to Type 2:** Convert from type 1 to type 3 as above, then convert to type 2 from type 3

## 4 Merging type 2 responses

The problem encountered when attempting to merge type 2 responses is as follows: given M rankings, each having N choices, which choice has the largest degree of support? For example, consider the following 3 voter/4 choice problem [STRA80]

**Voter 1**: a b c d **Voter 2:** c a b d **Voter 3:** b d c a

This case has no majority winner; a, b and c each get one first place vote. Intuitively, it seems reasonable to use the second place votes in this case to see if the situation resolves itself. In this case b receives two second place votes to a's one, which would tend to support b as the overall choice. In the general case there are a number of techniques for merging rank-ordered votes, four of which will be discussed here.

The *Borda count* [BORD81,BLAC58] is a well-known scheme for resolving this kind of situation. Each alternative is given a number of points depending on where in the ranking it has been placed. A selection is given no points for placing last, one point for placing next to last, and so on up to N-1 points for placing first. In other words, the number of points given to a selection is the number of classes below it in the ranking. For the 3 voter/4 choice problem described above the situation is:

**Voter 1:** a (3) b (2) c (1) d (0) **Voter 2:** c (3) a (2) b (1) d (0) **Voter 3:** b (3)d (2)c (1)a (0)

where the points received by each selection appears in parentheses behind the choice. The overall winner is the choice receiving the largest total number of points:

a = 3 + 2 + 0 = 5

b = 2 + 1 + 3 = 6

c = 1 + 3 + 1 = 5

d = 0 + 0 + 2 = 2

This gives choice b as the 'Borda winner'. However, the Borda count does have a problem that might be considered serious. Consider the following 5 voter/3 choice problem:

**Voter 1:** a b c **Voter 2:** a b c **Voter 3:** a b c **Voter 4 :** b c a **Voter 5:** b c a

The Borda counts are a=6, b=7, c=2, which selects b as the winner. However, a simple majority of the first place votes would have selected a! This violates the so-called *majority criterion* [STRA80]:

***If a majority of voters have an alternative X as their first choice, a voting rule should choose X.***

This is a weaker version of the *Condorcet Winner Criterion* [COND85]:

***If there is an alternative X which could obtain a majority of votes in pair-wise contests against every other alternative, a voting rule should choose X as the winner.***

This problem may have to be taken into account when assessing performance of the methods.

A procedure suggested by Thomas Hare [STRA80] falls into the category of an *elimination* process. The idea is to repeatedly eliminate undesirable choices until a clear majority support one of the remaining choices. Hare's method is as follows: if a majority of the voters rank choice X in first place, then X is the winner; otherwise, the choice with the *smallest number of first place votes* is removed from consideration, and the first place votes are re-counted. This elimination process continues until a clear majority supports one of the choices.

The Hare procedure satisfies the majority criterion, but fails the Condorcet winner criterion as well as the *monotonicity criterion*:

***If X is a winner under a voting rule, and one or more voters change their preferences in a way favorable to X without changing to order in which they prefer any other alternative, then X should still be the winner.***

No rule that violates the monotonicity criterion will be considered as an option for the multiple classifier. This decision will eliminate the Hare procedure, but not the Borda count. With the monotonicity criterion in mind, two relatively simple rank merging strategies become interesting. The first is by Black[BLAC58], and chooses the winner by the Condorcet criterion if such a winner exists; if not, the Borda winner is chosen. This is appealing in its simplicity, and can be shown to be monotonic. Another strategy is the so-called Copeland rule [STRA80]: for each option compute the number of pair-wise wins of that option with all other options, and subtract from that the number of pair-wise losses. The overall winner is the class for which this difference is the greatest. In theory this rule is superior to the others discussed so far, but it has a drawback in that it tends to produce a relatively large number of tie votes in general.

The Borda, Black, and Copeland rules were implemented as described and applied to the five classifier problem, and the results are summarized in Table 3. All methods supported the removal of classifier #4.

Table 3: Results of the Voting Rules for Rank Ordering (Omit #4)

| Rule | Recognition | Error | Rejection | Reliability | Acceptability |
|---|---|---|---|---|---|
| Borda | 99.9 | 0.1 | 0.0 | 0.999 | 0.998 |
| Black | 99.9 | 0.1 | 0.0 | 0.999 | 0.998 |
| Copeland | 99.6 | 0.2 | 0.2 | 0.998 | 0.994 |

From this table it would appear that the Borda scheme is tied with Black, followed by Copeland. It is important to temper this view with the fact that this result was obtained from basically one observation. Confirmation would come from applying these schemes to a large number of sets of characters. Another consideration is that a voting scheme may err in favor of the correct classification when it should, in fact, be rejected. Upon careful analysis this was found to have happened for the Borda method applied to digit #267. The rankings were:

Classifier 1:2

Classifier 2:1 7 4 2 9

Classifier 3:2

Classifier 4:1 9 6 3 2 7 8 5

Classifier 5:1 2 9

The Borda count for the one digit is 27, and for the two digit is 37, giving a classification of two even though the majority winner and the Condorcet winner is one! Thus, the Black scheme classifies this digit (correctly according to the votes, in my opinion) as a one. Given this problem, and the fact that Black and Borda are otherwise equally acceptable, my conclusion is that the Black classifier is slightly superior to the others.

## 5 Merging type 3 responses

The five classifier system under discussion has no single classifier that gives a proper type 3 response, and only one that yields a reliable set of weights for each digit (#5, the neural net). Because of this, the problem of merging type 3 responses was not pursued with as much vigor as were the type 1 and 2 problems. Indeed, the solution may be quite simple. Suen [XU92] decides that any set of type 3 classifiers can be combined using an averaging technique. That is,

$$P_E(x \in C_i | x) = \frac{1}{k} \sum_{j=1}^{k} P_j(x \in C_i | x), \text{ i=1,...,M}$$

where PE is the probability associated with a given classification for the multiple classifier, and Pk is the probability associated with a given classification for each individual classifier k. The overall classification is the value j for which $P_E(x \in C_j | x)$ is a maximum.

There is little actual type 3 data, but it could be approximated by using the *a posteriori* method described previously, where it is used to convert type 1 responses to type 3 responses. Using this approximate data set, the result obtained by merging type 3 responses using averaging is given by:

**Correct:** 997 **Incorrect:** 3 **Rejected:** 0

Acceptability is 0.994.

# 6 Results from the multiple classifier

Using the acceptability measure to assess each of the merging methods discussed, we need to look only at the best method in each of the three groups; that is, the best multiple type 1 classifier, the best type 2, and the best type three. The best three are:

| Name | Type | Acceptability |
|---|---|---|
| SMV | 1 | 0.994 |
| Black | 2 | 0.998 |
| Average | 3 | 0.994 |

From the table above it can be seen that the best classifier uses the Black scheme for merging rank ordered responses, and which omits classifier #4.

This work was supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

# 7 References

[BLAC58] Black, D., The Theory of Committees and Elections, Cambridge University Press, 1958.

[BORD81] Borda, Jean-Charles de., Memoire sur les Elections au Scrutin, Histoire de l'Academie Royale des Sciences, Paris, 1781.

[BRAM83] Brams, S.J and Fishburn, P.C., Approval Voting, Birkhauser, Boston, 1983.

[COND85] Condorcet, Marquis de., Essai sur l'application de l'analyse a la probabilite des decisions rendues a la pluralite des voix, Paris, 1785.

[ENEL84] Enelow, J.M. and Hinich, M.J., The Spatial Theory of Voting: An Introduction, Cambridge University Press, Cambridge, 1984.

[FARQ69] Farquharson, R., Theory of Voting, Yale University Press, New Haven, 1969.

[PARK94] Parker, J.R., Recognition of Hand Printed Digits Using Multiple/Parallel Methods, Third Golden West International Conference on Intelligent Systems, Las Vegas, June 6-9/94.

[HO94] Ho, T.K., Hull, J.J., and Srihari, S.N., Decision Combination In Multiple Classifier Systems, IEEE-PAMI, Vol. 16 No. 1, Jan. 1994.

[KIMU91] Kimura, F. and Shridhar, M., Handwritten Numeral Recognition Based On Multiple Algorithms, Pattern Recognition, Vol. 24 No 10, 1991.

[STRA80] Straffin, P.D. Jr., Topics in the Theory of Voting, Birkhauser, Boston, 1980.

[XU92] Xu, L., Krzyzak, A., and Suen, C.Y., Methods Of Combining Multiple Classifiers And Their Application To Handwriting Recognition, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 22 No. 3, May/June 1992.