Analysis of the Reliability and Behavior of Majority and Plurality Voting Systems

Sherif Yacoub, Xiaofan Lin, John Burns
Information Infrastructure Laboratory
HP Laboratories Palo Alto
HPL-2002-118
April 25th , 2002*

E-mail: {sherif_yacoub, xiaofan_lin, john_burns}@hp.com

majority voting, plurality voting, expert combination, fault-tolerance

Voting is a well-known technique used to combine decisions of peer experts. It has wide application in many domains. Voting is used in fault tolerant applications to mask errors from one or more experts using N-Modular Redundancy (NMR) and N-version Programming. It is also used in pattern recognition applications where decisions from several classifiers can lead to better recognition results.

There are several strategies for voting including: majority, weighted voting, plurality, instance runoff voting, threshold voting, and the more general weighted k-out-of-n systems. To use a voting schema in any application domain, we have to understand the various tradeoffs and parameters and how they impact the correctness, reliability, and confidence in the final decision made by the voting system. In this paper, we analyze the behavior of two voting schemas: majority voting and plurality voting. We conduct synthetic studies using a simulator that we developed to analyze results from each expert, apply a voting mechanism, and analyze the voting results. The simulator builds a decision tree and uses a depth-first traversal algorithm to obtain reliability of the system and other factors that describe the voting behavior. For this analysis, we define and study the following behaviors of a voting system: 1) the probability of reaching a consensus, "Pc"; 2) reliability of the voting system, "R"; 3) certainly index, "T"; and 4) the confidence index, "C". The parameters controlling the analysis are the number of participating experts (or units), the number of possible output symbols that can be produced by an expert, the probability distribution of each expert's output, and the voting schema. This study unleashes several behaviors of a voting system and introduces a synthetic approach to compute its reliability.

* Internal Accession Date Only
Approved for External Publication

1 INTRODUCTION

Voting is a general technique that finds application and acceptance in many domains including software systems. Voting systems are often used in distributed systems to control mutual exclusion [e.g. Paris 1986]. They are used to control the update procedure when clusters of workstations are isolated due to network problems and to vote on data replication [Ahammad *et.al.* 1987]. A later study of voting in distributed applications is discussed in [Kumar *et.al.* 1991]. Voting systems are also used in fault tolerant applications to achieve reliability [Nordmann *et.al.* 1999]. It is a widely used technique for combining classifiers in the pattern recognition field [Ho *et.al.* 1994, Xu *et.al.* 1992, Yacoub *et.al.* 2002].

Voting systems work for various purposes and in various domains. The analysis of such systems is useful in understanding their theory of operation and the underlying assumptions made by several researchers to simplify their deployment. Theoretical and experimental analysis of the behavior of voting schemas is thus an important research topic. There have been several investigations on the theoretical analysis of voting systems. For example, in the pattern recognition domain, theoretical analysis of the behavior of voting system is discussed by Lam and Suen [Lam *et.al.* 1997]. Experimental analysis of plurality and majority voting with application to pattern recognition is discussed in [Lin *et.al.* 2002]. In the reliability analysis domain, a general mathematical model for *k–out-of-n* systems is discussed in [Nordmann *et.al.* 1999, Parhami 1994b, and Parhami 1994a]. An investigation of the various types of voting systems is given in [Lorczak *et.al.* 1989]. Weighted *k-out-of-n* is a well-known generalized technique for voting. An investigation of reliability optimization for weighted voting systems is discussed by Levitin *et.al.* in [Levitin *et.al.* 2001, Levitin 2001].

The general setup for a voting system is illustrated in figure 1, where we have several experts producing data to be voted on by a voter component. As opposed to the traditional 3-mode decision models [as in Levitin *et.al.* 2001], where the expert produces a binary decision "1" or "0" or abstain from voting "x", we use a general framework where the output from each expert can take "K" values in response to an input and the expert has a predefined probability distribution function for the "K" symbols. We then study the behavior of voting schemas experimentally. Application of this technique to units with the tri-state model is discussed in [Yacoub 2002].

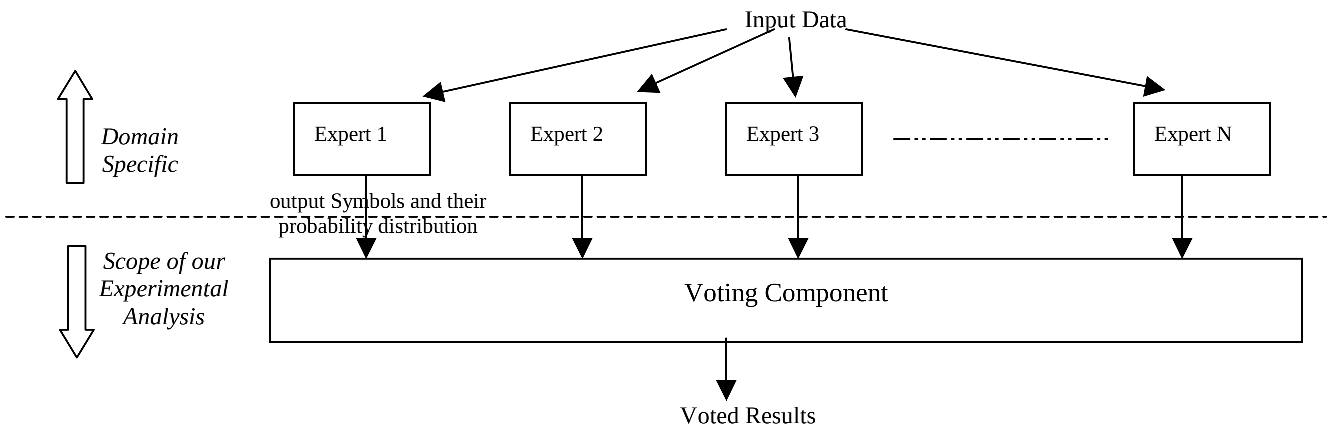

Figure 1 The scope of experimental analysis discussed in this paper.

In this paper, we take a synthetic approach to study the behavior of two well-known types of voting schemas: majority and plurality. The objective of this analysis is to experimentally find answers to the following questions:

1) What is the actual gain in terms of system reliability that we get when we use more experts?
2) How does the number of experts used in voting affect the likelihood of reaching a consensus between the experts?
3) How is the reliability of the voting system affected by changes in the number of experts or the number and distribution of the symbols produced by each expert? What is the effect of the number of possible symbols produced by an expert on the voting behavior?
4) What are the tradeoffs between confidence in results, chances of reaching a consensus, and reliability of the system?
5) Which voting schema to select? We would like to study what benefits and what penalties do we get from using different voting schemas.

To be able to answer these questions, we developed a simulator that creates a decision tree using the possible decisions of each expert and their probability distributions. The simulator uses a depth first traversal algorithm to obtain results for the selected voting schema. These results are then analyzed to obtain the behavior and the reliability of the voting system. We study the following behaviors: reliability, probability of reaching a consensus, certainty, and confidence. The definitions governing these terms are discussed in section 2. Section 3 describes decision tree, synthesis process, the simulator, and the experiment setup and procedure. Sections 4 and 5 describe the results of several studies. Section 5 is dedicated to reliability analysis. Finally we conclude the paper and summarize the findings in section 6.

2 DEFINITIONS AND ASSUMPTIONS

Definition 1: An Expert

Voting is used to combine data that is produced from multiple experts. The words "experts", "engines", "classifiers", and "units" are used interchangeably. The word "classifier" is often used in the pattern recognition domain, "expert" and "unit" are used in the reliability analysis domain, and "engine" and "algorithm" is used in the algorithm combination domain. Usually several experts are used and their results are combined to produce data that is more accurate (or of higher confidence) than data produced from one individual expert.

Definition 2: Number of Experts, N

We use "N" as the number of experts used in the voting system. We will study the behavior of the voting component as a function of N.

Definition 3: Symbols

Each expert produces data that belongs to a specific range of data symbols or data classes. We use the term "symbol" to refer to an individual output from the expert which belongs to a range of values (or set of symbols) that the expert output can take.

Definition 4: Number of Symbols, K

We define "K" as the total number of the possible symbols that can be produced by an expert. We will study the behavior of the voting system as a function of K. We use kj to refer to the jth symbol from the possible data set K.

Definition 5: Probability Mass Function of an Expert, PMFi

We define PMFi as the probability mass function that captures the probability distribution of the symbols produced by the i^{th} expert. Experts can have similar or different output data PMFs. PMFi is a discrete probability function. For instance, we can use uniform distribution (i.e. the expert is equally likely to produce any symbols) or Poisson distribution (i.e. the expert would favor certain symbols over others) or geometric (i.e. the expert would favor one symbol over the others).

Definition 6: Probability that an Expert Produces a Correct Output.

This is the probability that an expert will produce a correct output (irrespective of the voting context in which it is used). For a given input data, each expert (unit) will produce one or more symbols with a given PMF. One of these symbols is the correct one and the rest are wrong (the expert could make mistakes). If the expert always produces the correct symbol then its reliability Ri=1. However, in practical applications the expert makes mistakes and hence its reliability is not unity. The PMF models the probability distribution governing the experts output symbols and hence its reliability (when the most probable symbol is the correct one).

Definition 7: Majority Voting

Majority voting is a voting schema that has the following consensus rule:

$$\begin{cases} \textit{Consensus} & \textit{iff } \lfloor N/2 \rfloor +1 \text{ experts agree (produce the same output symbol)} \\ \textit{No Consensus} & \text{otherwise.} \end{cases} \qquad \text{. . . . . } \textit{Eq. 1}$$

Definition 8: Plurality Voting

Plurality voting is a voting schema that has the following consensus rule:

$$\begin{cases} \textit{Consensus} & \textit{iff } \text{ (L experts agree) \&\& !(M experts agree where M>=L)} \\ \textit{No Consensus} & \text{otherwise.} \end{cases} \qquad \text{. . . . . } \textit{Eq. 2}$$

Definition 9: Reliability of the Voting System "R"

We define "R" as the probability that the output from the voting system is correct. To measure whether the output is correct or not it has to be known a priori that a specific output symbol is the correct one and hence we are able to judge that the system output if correct. In our experiments, we assume that for a given input there is one correct symbol that each expert should produce. When the voting result among the expert outputs is that correct symbol, we have a correct output for the voting system and R is calculated using the probability of reaching consensus on that correct symbol.

Definition 10: Probability that a Consensus is reached "Pc"

We define "P_C" as the probability of reaching a consensus. We note here that the probability of reaching a consensus is not the same as the probability that the output is correct. For example, assume that we are using a majority voting of three experts A, B, C. Experts A and B produce the same results say "x" and expert C produces a different result say "y". The output from the majority voting in this case is the "x" symbol. However, if the correct result is "y" (as measured against a ground truth) then the result of the voting system is not the same as the correct result. Hence, Pc and R are different random variables. Therefore, we study the behavior of the voting system in terms of both "Pc" and "R". In the case where all symbols are equally probable, we can posit one of the symbols to be the correct one, and then study what percentage of the consensus cases map to the consensus on correct symbol. If we use uniform probability distributions, for example, each of the symbols equally likely to be the consensus—and thus (1/K)x100% of the consensus cases will be the actually "correct" output, hence R and Pc are related. However, for other distributions, the relationship between Pc and R is not that obvious and they have to be studied separately.

Definition 11: Certainty Index (Weighted Consensus), T

We define "T" as the certainty index. T is a function of the following factors:

- Probability that a specific consensus case is reached. A consensus case is defined as a case where we have a set of votes sufficient to make a consensus according to the voting strategy and its consensus rule (see equations 1 and 2). For instance, with three experts A, B, C and a majority voting, the case where A and B agree and differ from C is a consensus case.
- Number of experts participating in making a specific consensus. An expert is said to participate in making a consensus if the output of the expert is the same as the consensus results.

For example, assume that we have three experts, A, B, and C. Two experts produce the same result and the third produced a different result. The result of the majority voting for this case will be the symbol agreed upon by the two experts. Then the number of experts participating in reaching the consensus is 2 out of 3. If we expand the decision tree as shown in the following section using the PMFi for all experts, we calculate the probability of this case, assume 6/9. Now consider another case where the three experts produce the same result then we have 3 out of 3 experts producing the consensus. The probability of the case where the three experts agree can be calculated from the decision tree with uniform PMF, which is 1/9. For several consensus cases, the general formula for T is given by:

$$T = \sum_{s \in \{\text{Consensus Cases}\}} \mathrm{P(s/c)} \times \frac{\mathrm{NEs}}{\mathrm{N}} = \sum_{s \in \{\text{Consensus Cases}\}} \frac{\mathrm{P(s,c)}}{\mathrm{Pc}} \times \frac{\mathrm{NEs}}{\mathrm{N}} \quad \ldots\ldots\ldots\ldots \textit{Eq. 3}$$

where P(s/c) is the probability of the case "s" given that consensus is reached, "NEs" is the number of experts making a consensus in the case "s", and Pc is the consensus probability. Using the previous example, certainty T is calculated as (6/9 * 2/3 + 1/9 * 3/3)/(6/9 + 1/9).

Note that the certainty index that we calculate means how certain we are about the consensus reached but it does not say whether we are certain it is "correct" or "wrong". If T is calculated for those cases only where a consensus is reached on the correct symbol, then T measures the certainty in the correct result as opposed to certainty in the consensus. We will use the two variables T(Pc) and T(R) to distinguish the two cases.

Definition 12: Confidence Index, C

The confidence index "C" represents how confident we are in the results produced by the voting system. It takes into consideration an important factor, which is how many experts are participating in reaching a consensus. In this case it is similar to the certainty index but it is NOT normalized to the total number of experts. The formula for calculating C is:

$$C = \sum_{s \in \{\text{Consensus Cases}\}} \text{P(s/c) x NEs} = N \times T \quad \text{. . . . .} \textit{Eq. 4}$$

Since this factor is not normalized to the number of experts, then it will be bigger than unity. The number is significant in comparing confidence produced using several N-experts systems. Note that C is calculated for all consensus case, which does not distinguish between the correct or wrong symbols. To calculate the confidence in the correct symbol we consider the consensus cases in which the system reaches consensus in the correct symbol. We use the two variables C(Pc) and C(R) to distinguish confidence in the voting system consensus and confidence in the voting system reliability respectively.

3 SYNTHETIC EXPERIMENT SETUP

We built a simulation system for the purpose of conducting various experiments to study the behavior and reliability of the majority and plurality systems as a function of several parameters. In this section we describe the inputs, outputs, and the operation of the simulator.

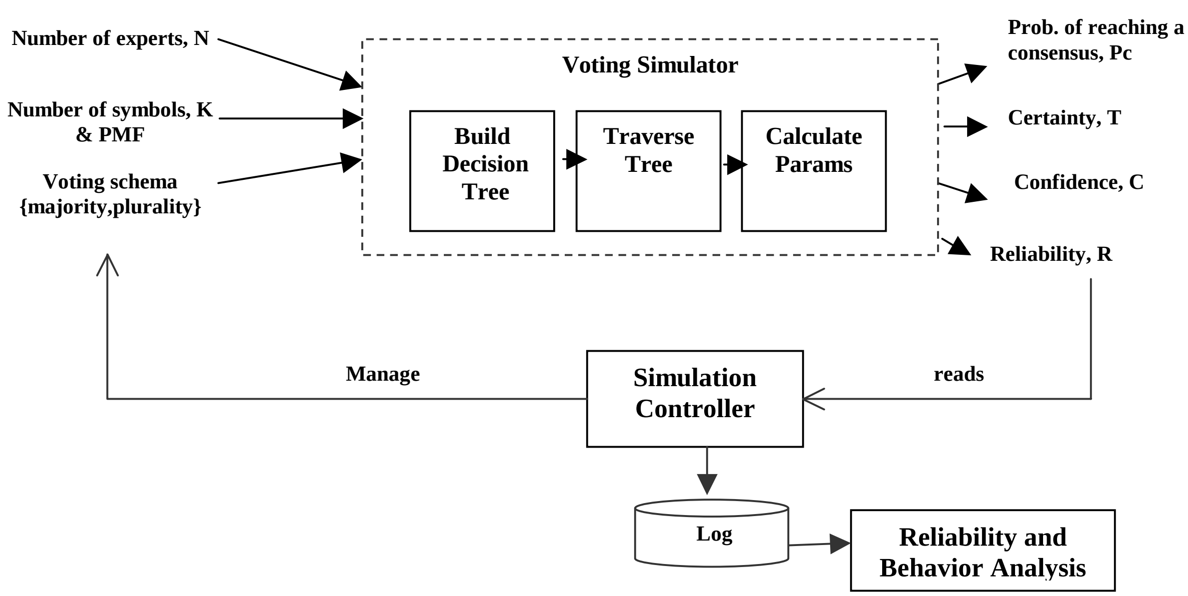

Figure 2 The simulation system

The simulator accepts three inputs:

a) the number of experts participating in the experiment (previously defined as N),
b) the number of possible symbols that an expert can produce (previously defined as K), and the PMF for the K values for each expert (defaulted to uniform if not provided)
c) the type of voting system (currently, the simulator supports only two types: majority and plurality).

For every input tuple (N, K & PMF per expert , Schema), the output of the simulation is:

a) the reliability of the voting system, R (see definition 9)
b) the probability of reaching a consensus, Pc (see definition 10)
c) the certainty index, T(Pc) and T(R) (see definition 11)
d) the confidence index, C(Pc) and C(R) (see definition 12)

The simulator is executed for each experiment using the tuple: (N,K,Schema). For example, if we want to study the effect of the parameter N on Pc, T, R, and C, then the simulator will run several experiments fixing K and the Schema and varying N for some predefined range.

The results from the simulator are exactly the same results that we obtain if we do the mathematical calculations by hand with the advantage of having an automated algorithm to perform the task on the analyst's behalf. This is because the simulator expands all possible combinations (100% coverage) and will not miss any combination case. To illustrate this, the following procedure explains the operation performed by the simulator to calculate Pc, T, R, and C.

a) Creating a Decision Tree

First, the simulator uses the number of experts N, the number of possible values that an expert can produce K, the set of PMFi for each expert, to build a decision tree. The tree explores all possible combinations of the experts' output and is thus equivalent to any analytical (mathematical) result.

The number of experts defines the depth of the tree. Each node at one level will be expanded to K nodes in the lower level. Going from one level to another is the same as exploring all possible outputs from an expert. At the lowest level of the tree, we have all possible combinations that can be produced by N experts for K possible symbols. The nodes at the lowest level are called *terminal nodes*. The arcs in the decision tree is labeled with the probability that the expert produces the next level output symbol. This is usually specified in the PMF distribution for each expert.

As an example, consider the case where we have three experts (N=3) and two possible output symbols for an expert (K=2). We have two possible symbols, lets assume they are "x" and "y". The PMF distribution is uniform for the first and third experts and one symbol "x" is favorable for the second expert compared to “y”, PMF2(x)=2/3, and PMF2(y)=1/3. We have three experts; hence the depth of the tree is three. The following figure illustrates the expansion of the tree.

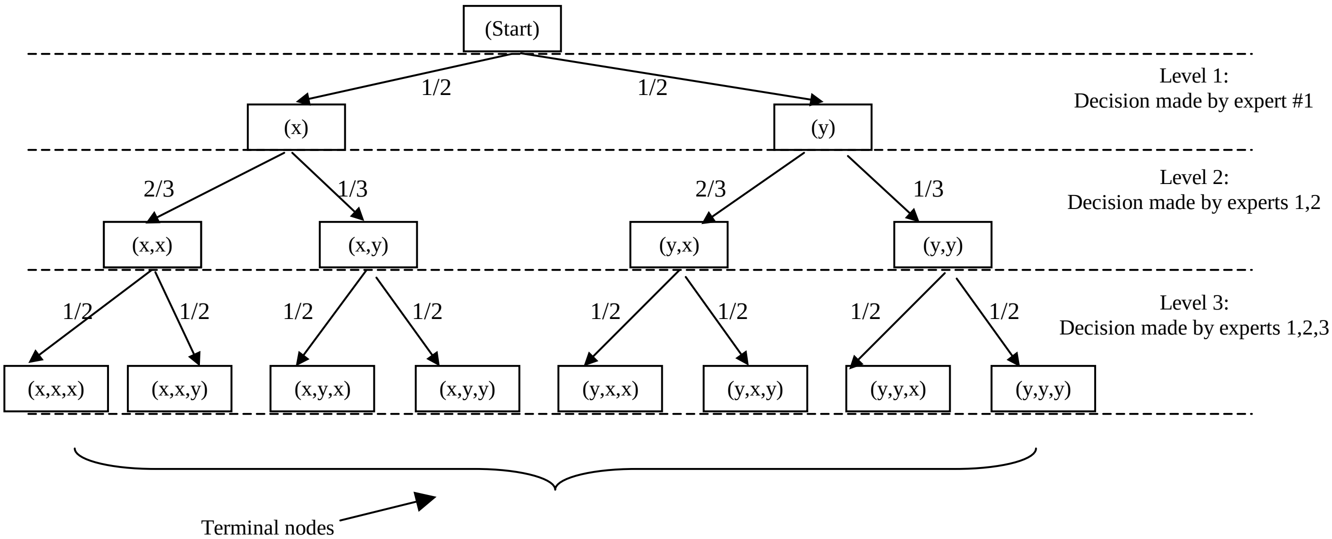

Figure 3 Decision tree for three experts with two possible decisions (x or y)

Note: Order is Irrelevant

The order of the experts in the tree is not relevant. This is because: a) there is no explicit dependency between experts, b) the decision made by one expert does not affect the decision made by the other (experts are not collaborative), and c) the order of symbols at the terminal nodes is not the one affecting the decision, instead it is the count (score in case of weighted experts) of the symbols that matters.

Note: Number of Terminal Nodes

The number of terminal nodes obtained from the decision tree is function of the number of experts N and the number of possible symbols produced by each expert K. The total number of terminal nodes will be given by:

$$NumberOfTer\min alNodes = M = K^{N} \qquad \text{Eq. 5}$$

b) Assigning probability to various combinations.

For each terminal node, we calculate the probability of reaching that node by back propagating the tree and accumulating the probability of each branch. Since independence between experts is assumed, the probability of reaching a terminal node is the *product* of the probability of all branches traversed to reach that node. Hence:

$$PT(m) = \prod_{i=1}^{N} PMF_i(S_m(i)) \qquad \text{. . . . . } Eq.\ 6$$

where $S_m(i)$ is the i^{th} symbol in the m^{th} terminal node result. For instance if the results in the second terminal node is (x,x,y), then $S_2(1)=S_2(2)=$"x" and $S_2(3)=$"y".

For the above decision tree, the probability of the terminal node (x,x,y) is: ½*2/3*½.

Special case: Uniform Distribution

In a simple situation, all symbols might be equally probable for all experts. Hence, the probability that the i^{th} expert produces a symbol k_j from the set of symbols R is given by:

$$PMF_i(k_j / UniformDistribution) = \frac{1}{K} \qquad \text{. . . . . } Eq.\ 7$$

This means that all the terminal nodes are equally probable and hence each decision has a probability of 1/8 for the case shown in figure 2. Note that the number of terminal nodes according to equation 5 is : K^N , hence the probability of the m^{th} terminal node PT(m) will be 1/ K^N:

$$PT(m) = \frac{1}{K^N} \qquad \text{. . . . . } Eq.\ 8$$

Note: Practical Considerations

The PMF distribution depends on the domain in which the system is used. For instance, for pattern recognition applications, a uniform distribution is not a practical assumption since all symbols are *not* normally equally probable (experts are not really doing any recognition). For elections, one might assume uniform distribution between candidates. For reliability analysis, one might start with equal probability of success and failures for a particular expert before building history for its operation.

c) Evaluating Probability of reaching a consensus (Pc).

Each one of the terminal nodes is checked against the consensus rule for the required voting schema. Consensus rules for majority and plurality voting are given in equations 1 and 2. The probability of reaching a consensus is then calculated as follows:

$$Pc = \sum_{m=1}^{M} \mathrm{PT(m)} \times \delta(\mathrm{m}) \qquad \text{. . . . . } Eq.\ 9$$

where PT(m) is calculated using equation 6 and δ(m) is given by:

$$\delta(\mathrm{m}) = \begin{cases} 1 \ \ \textit{iff}\ \text{m is a consensus terminal node} \\ 0 \ \ \text{otherwise} \end{cases}$$

Special case: Uniform Distribution

In a simple situation when all symbols might be equally probable, we evaluate the number of nodes reaching a consensus and the probability of reaching a consensus is simplified to (and, under these special assumptions, it can be deduced from the binomial distribution):

$$Pc = \frac{\text{Number of terminal nodes reaching a consensus}}{\text{Total number of terminal nodes}} = \frac{N_C}{K^N} \qquad \text{. . .. . . . } \textit{Eq. 10}$$

d) Evaluating the certainty index T.

To calculate the certainty index, we only consider the terminal nodes labeled in procedure (c) above as reached a consensus. For each terminal node we calculate the percentage of experts participating in the consensus relative to the total number of experts. As an example, consider the terminal node (x,x,y). The consensus result in this case is “x”. Hence the certainty in this particular consensus decision is 2/3. Certainty index is then calculated by summing all consensus cases and weighing it with the probability that the consensus case is reached (i.e. weighted or normalized consensus).

$$T = \sum_{m=1}^{M} \frac{NEm}{N} \times \frac{PT(m)}{Pc} \times \delta(m) \qquad \text{. . .. . } \textit{Eq. 11}$$

where: NEm is the number of experts making a consensus for the m^{th} consensus case, PT(m) is calculated from equation 6 and δ(m) is given from equation 9.

Special case: Uniform Distribution

In a simple situation when all symbols might be equally probable, we evaluate the number of nodes reaching a consensus and the certainty index is simplified to:

$$T = \sum_{m=1}^{M} \frac{NE_m}{N} \times \frac{1}{K^N} \times \delta(m) \times \frac{1}{\text{Pc}} = \frac{1}{N \times Nc} \times \sum_{m=1}^{M} NE_m \times \delta(m) \qquad \text{. . . . . } \textit{Eq.12}$$

Note equations 11 and 12 are calculated for the all consensus cases, hence the values obtained there are T(Pc). To calculate T(R) we consider a subset of the consensus cases in which the consensus is reached on a correct symbol only.

e) Evaluating the confidence index C

To calculate the confidence index, we only consider the terminal nodes labeled in procedure (c) above as reached a consensus. For each consensus case, we count the number of experts participating in the consensus. As an example, consider the terminal node (x,x,y). The consensus result in this case is “x”.

Hence the confidence in this particular consensus decision is the absolute number 2. The confidence index is then calculated by summing all consensus cases and weighing it with the probability that the consensus case is reached.

$$C = \sum_{m=1}^{M} NEm \times \frac{1}{Pc} \times PT(m) \times \delta(m) \qquad \text{. . . . . } Eq.13$$

Special case: Uniform Distribution

In a simple situation when all symbols are equally probable, we evaluate the number of nodes reaching a consensus and the confidence index is simplified to:

$$C = \sum_{m=1}^{M} NE_m \times \frac{1}{K^N} \times \frac{K^N}{Nc} \times \delta(m) \quad = \quad \frac{1}{Nc} \times \sum_{m=1}^{M} NE_m \times \delta(m) \qquad \text{. . . . . } Eq.\ 14$$

From equations 12 and 14, then

$$C = N \times T \qquad \text{. . .. . } Eq.\ 15$$

Hence C and T are tightly related. The certainty index T measures the confidence we have in the consensus given that we know the best result will be produced by all experts agreeing on the output. The confidence index C is an absolute measure for confidence used to compare confidence in the results produced by multiple N-expert systems.

Note equations 14 and 15 are calculated for the all consensus cases, hence the values obtained there are C(Pc). To calculate C(R) we consider a subset of the consensus cases in which the consensus is reached on a correct symbol only.

f) Evaluating Reliability, R

Each one of the terminal nodes is checked against the consensus rule for the required voting schema. For those nodes that reached a consensus, we determine the consensus symbol and compare with the correct symbol. R is then calculated using equation 9 for those cases where the consensus symbol is the correct one.

Note: Since the number of terminal nodes is exponentially related to the number of experts (N) given to the simulator, the expansion of all possible branches in the tree is impractical due to the extensive memory requirements. To enable the simulator to work with any value of N, we have taken a traverse-while-build approach by keeping track of indexing while traversing the tree. In this case we use a depth first traversal algorithm and at one point in time only N-1 nodes of the tree are kept in memory.

4 Synthetic Experimental Results

The results produced in this section are obtained for uniform distribution of symbols produced by each expert. Results from other PMF functions are not shown. These results illustrate the types of behaviors and reliability analysis that we can perform using the synthetic analysis approach and simulator.

4.1 Prob. of a reaching consensus for Plurality and Majority voting as function of N

4.1.1 Purpose

The purpose of this experiment is to study the effect of changing the number of experts "N" on the probability of reaching a consensus "Pc". We conduct the study for both majority and plurality voting. For a given number of possible output symbols "K" (i.e. the output from each expert could be any of K-symbols), the simulator expands the decision tree and assesses the consensus rule at the terminal nodes. Assuming uniform PMF, the probability of reaching consensus is calculated using equation 10. We have conducted the experiments for values of K=2,3,4,5 and for various number of experts N. Results are shown in the following figure.

4.1.2 Results

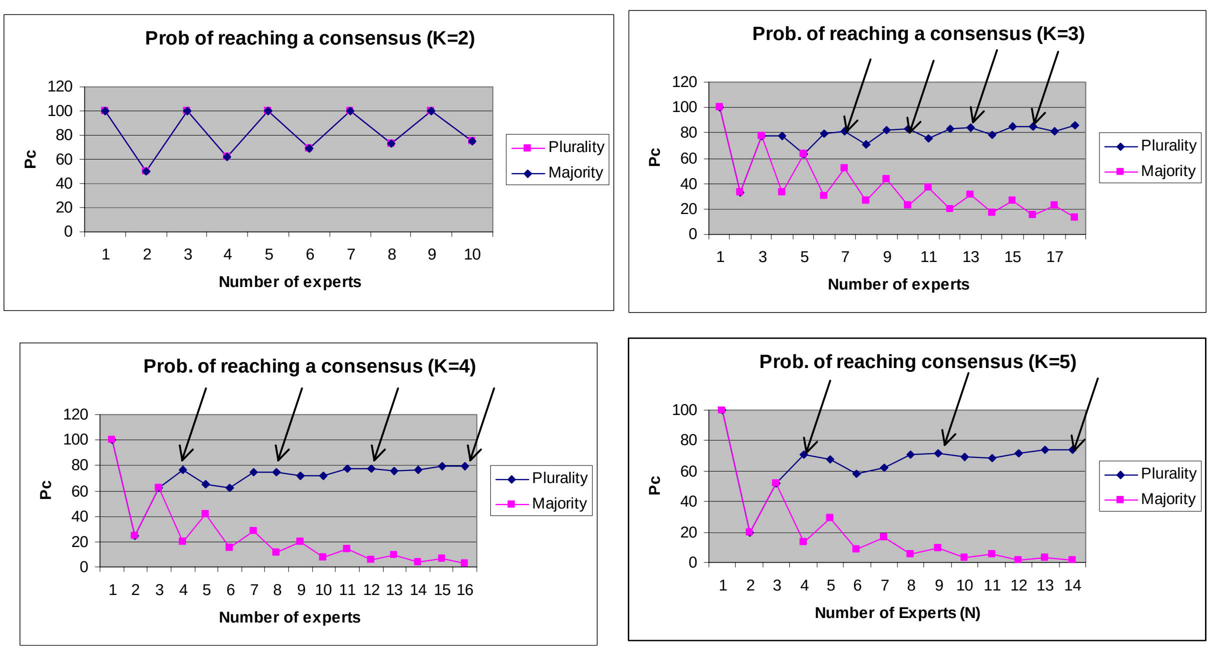

Figure 4 Pc for Majority and Plurality voting for various K values.

For the simple case where we only have one expert, we will always (100%) reach a consensus since there is only one expert and hence its vote is the consensus.

For two and three experts, the results for the majority and plurality voting are the same irrespective of the number of possible output symbols. This is because for two and three experts, the consensus rules for the two voting schemas are equivalent.

4.1.3 Analysis

a) For a majority voting system, the even number of experts (assume L) decreases the chance of reaching a consensus when compared to the probability of reaching a consensus using L-1 (the smaller odd number) experts. This experimental result confirms the theoretical conclusions found by Lam and Suen [Lam *et.al.* 1997]. Whereas Lam *et.al.* study the probability of reaching a correct result "R" as a function of the probability that an expert is correct, we study the probability of reaching a consensus "Pc". We come with the same conclusion that for majority voting the even number of experts can be substituted with a lower odd number of experts and as a result we get higher probability of reaching a consensus (or higher probability of getting a correct output according to Lam)

b) The probability of reaching a consensus using plurality voting is always greater than or equal to the probability of reaching a consensus using majority voting. This is because the consensus condition for plurality voting is more relaxed than the consensus condition for the majority voting. However, this does not mean that we should always use plurality voting instead of majority because confidence in the voting results is another important factor to consider (as discussed later).

c) The oscillation in the values of "Pc" as a function of the number of experts is very obvious for the majority voting. We can define another term that we call "***oscillation period of Pc***", which is the number of experts between two consecutive peak values of "Pc" where only peak values for $N>3$ are considered. For majority voting, the oscillation period of Pc is "2", since the even expert would usually produce a lower Pc value than the Pc values produced by the enclosing odd number of experts.

d) The "Pc" value for majority voting is asymptotically decreasing as the number of experts increases. This also confirms the theoretical analysis by Lam *et.al.* where because of uniform distribution the probability of each symbols would be < 0.5 ($p<0.5$ in [Lam *et.al.* 1997]).

e) The "Pc" value for plurality voting is asymptotically *increasing* as the number of experts increases.

f) Plurality voting also experiences a similar oscillation. From experimental results shown in the previous figure, the oscillation period for a plurality voting is dependent on the number of symbols K. Our simulation shows that the oscillation period for plurality vote systems is: **"K".** This is a note-worthy behavior that should be further studied theoretically.

4.2 Certainty Index (T) for Plurality and Majority voting as function of N

4.2.1 Purpose

The purpose of this experiment is to study the certainty in the consensus reached by a voting schema as a function of the number of experts "N". We conduct the study for both majority and plurality voting. For a given number of possible output symbols "K", the simulator expands the decision tree and assesses the certainty index for all consensus cases as discussed in equations 1 and 2. We can also calculate certainty for consensus case were the consensus is correct T(R), however we limit the discussion here to T(Pc). T is then calculated using the equation 12. We have conducted the experiments for values of K=2,3,4,5 and various numbers of experts N. Results are shown in figure 5.

4.2.2 Results

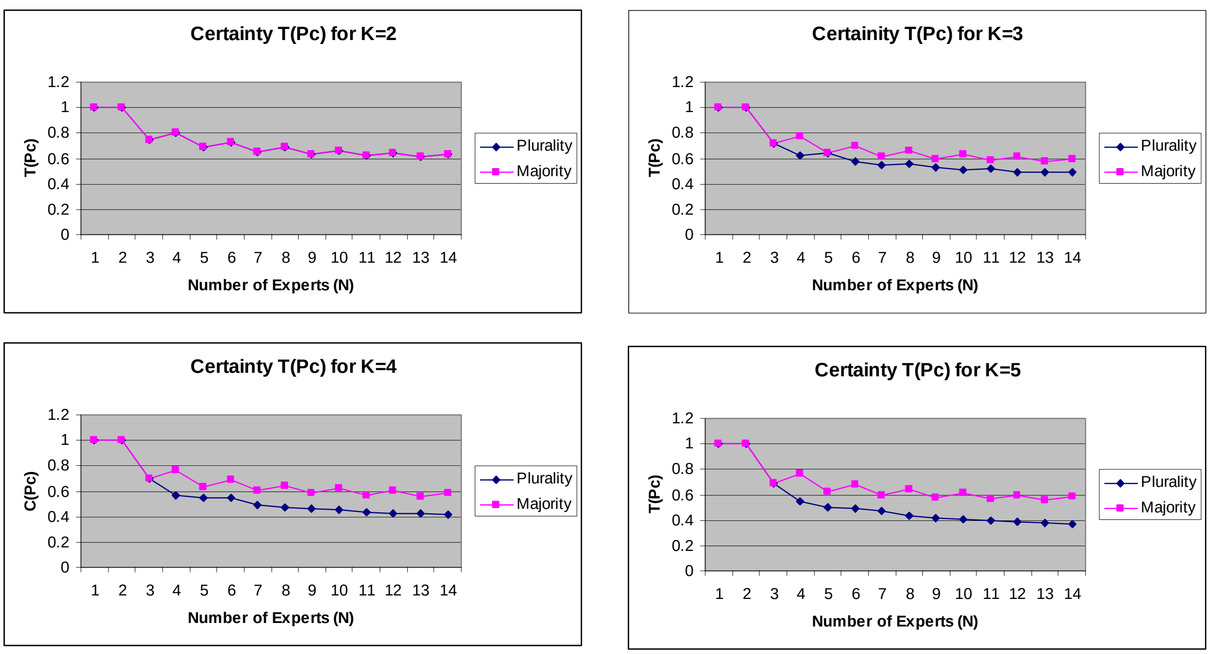

Figure 5 Certainty T for Plurality and Majority Voting for various K values.

4.2.3 Analysis

From figure 5 we find that:

a) The certainty index in the consensus produced by a majority voting schema is higher than or equal to the certainty index that we have in the results of a plurality voting. The logical interpretation of this result is that the consensus rule of a majority voting is more strict in terms of the requirements on the number of experts participating in reaching a consensus as compared to the plurality voting.

b) As the number of experts increases, the certainty in the result decreases! This is true because the addition of another expert is interpreted by the voting system as another source of information and another factor against consensus because all symbols are equally likely. The certainty index is a measure of the mean number of experts voting towards a consensus normalized to the total number of experts participating in the voting and hence, the decrease in the certainty index.

c) For the majority voting case, the certainty index for an even number of experts is higher than the certainty index for the previous and following odd number of experts. For instance T4 is greater than T3 and T5 for any values of K.

d) Trade-off between T and Pc in majority voting. From section 4.1, we concluded that probability of reaching a consensus Pc for "odd" number of experts is higher than the probability of reaching a consensus for the previous and the following "even" number of experts, for example: Pc(4 experts) < Pc(3 experts) and Pc(4 experts) < Pc(5 experts) for all values of K. However as we concluded in "c" above, T4 > T3 and T4 > T5. Thus we conclude that with the transition from odd number of experts to even number of experts the probability of reaching a consensus decreases but the certainty in the consensus reached by the system increases.

e) The plurality voting does not have the same tradeoff between probability of reaching consensus and the certainty index as the case for majority voting. T for plurality voting is asymptotically decreasing as function of N (mean number to have a simple plurality drops as we make it more analog as N>>K).

4.3 Confidence Index (C) for Plurality and Majority voting as function of N

4.3.1 Purpose

The purpose of this experiment is to study the confidence we have in the results of a voting schema with "N" number of experts. We would like to study how the confidence changes as we add new experts to the voting system. We conduct the study for both majority and plurality voting. For a given number of possible output symbols "K", the simulator expands the decision tree and assesses the confidence index "C" for all consensus situations as discussed in equations 1 and 2. C is calculated according to equation 14 in the previous section. We have conducted the experiments for values of K=2,3,4,5 and for various values of number of experts N. Results are shown in figure 5.

4.3.2 Results

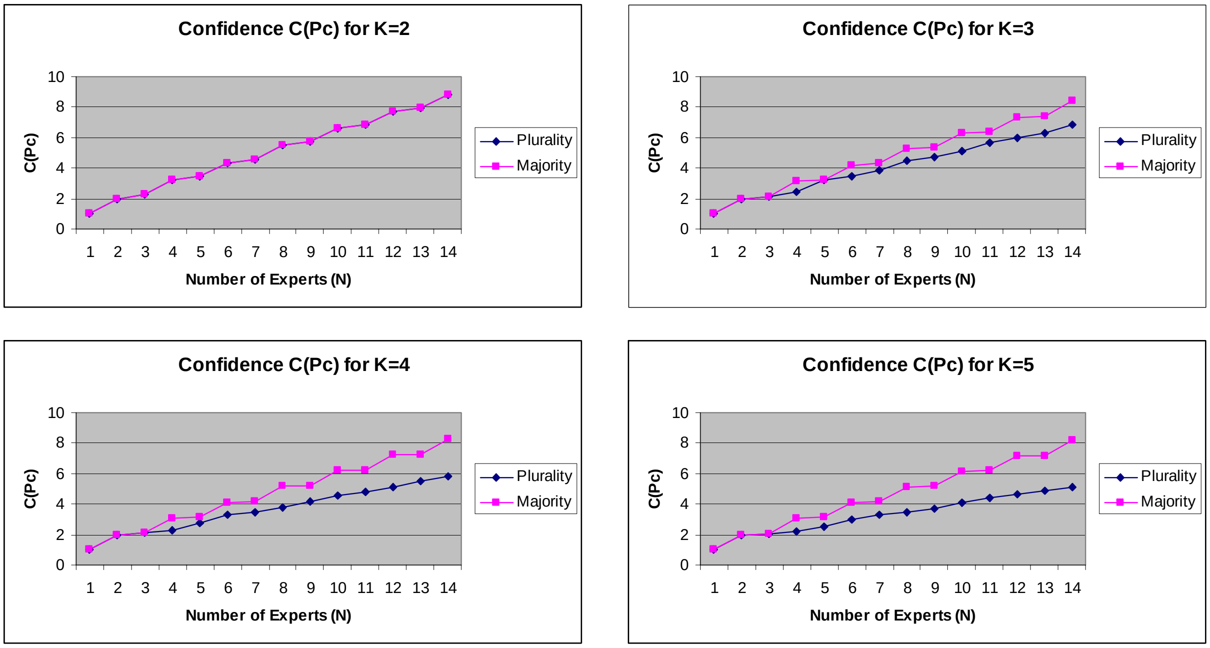

Figure 6 Confidence index C for Plurality and Majority voting for various K values.

4.3.3 Analysis

From figure 6, we find that:

a) The confidence index C(Pc) that we have in the results produced by a majority voting schema is higher than or equal to the confidence index that we have in the results of a plurality voting. Same result and explanation concluded for the certainty index T.

b) The ratio between C for majority voting and C for plurality voting is proportional to the number of possible output symbols K. For instance assume N=7, then for K =3, C(Pc)(majority)/C(Pc)(plurality) = 4.3/3.84 = ~1.12 . While in case of K=5, C7(majority)/C7(plurality) = 4.2/3.4 = ~ 1.24. Hence, we conclude that as the number of possible outcome symbols increases, the majority voting will have higher confidence in the results than the plurality voting.

c) The confidence index is linearly increasing. This means that as we add more experts to the voting system, whether it is "majority" or "plurality" voting, we gain more confidence in the output. This is because when a consensus is reached, we will probably have more experts participating in reaching the consensus than the case of fewer experts. Note that confidence index C(Pc) is not normalized to the number of experts participating in the voting and that is why it provides a good measure for the user's confidence in the results of a particular voting schema

4.4 Probability of reaching a Consensus as function of N and K

4.4.1 Purpose

The purpose of this experiment is to study the probability of reaching a consensus as function of the number of experts N and the number of possible output symbols that an expert can produce K. We conduct the study for both majority and plurality voting. For a given number of possible output symbols "K", the simulator expands the decision tree and assesses the probability of reaching a consensus (based on the consensus rule for a voting schema) as discussed in section 4.1. We have conducted the experiments for values of K=2 through 9 where the number of experts N varies from 1 to 6. Results are shown in figures 7 and 8.

4.4.2 Results

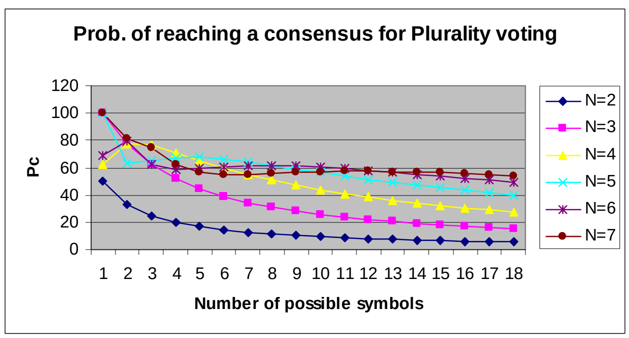

Figure 7 Pc as function of N and K for plurality voting

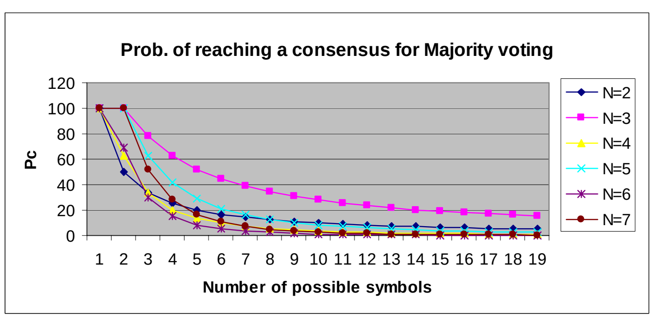

Figure 8 Pc as function of N and K for majority voting

4.4.3 Analysis

a) For majority voting, the probability of reaching a consensus Pc for an odd number of experts is usually higher than Pc for an even number of experts for a give value of “K”. For example, in figure7, the results for N=3 and N=5 are always larger than the Pc values for N=2, N=4, or N=6.

b) For majority voting, a voting system with three experts always outperforms any number of experts in terms of the probability of reaching a consensus.

c) For plurality voting, Pc is proportional to the number of experts used. However, these results are correct for comparing N for number of symbols K, where K>=N. Hence, for large values of “K” the more experts N>K the higher the chance of reaching a consensus.

4.5 *Confidence Index C as function of N and K.*

4.5.1 Purpose

The purpose of this experiment is to study the confidence index C(Pc) as function of the number of experts used in the voting N and the number of possible output symbols that an expert can produce. We conduct the study for both majority and plurality voting. For a given number of possible output symbols “K”, the simulator expands the decision tree and assesses confidence index as discussed in section 4.2. We have conducted the experiments for values of K=2 through 14 where the number of experts N varies from 2 to 6. Results are shown in figures 9 and 10. We note that the values of C evaluated here are for C(Pc), C(R) is discussed in section 5.

4.5.2 Results

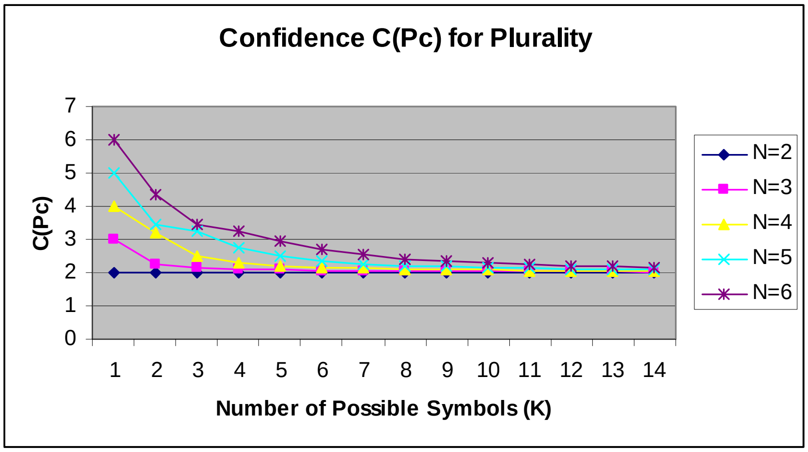

Figure 9 C as function of N and K for plurality voting

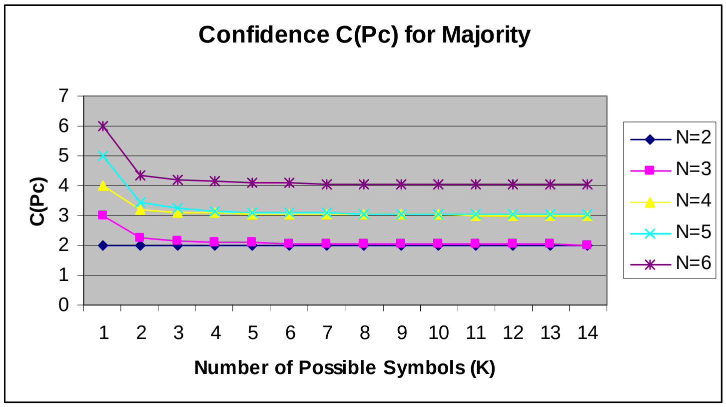

Figure 10 C as function of K and N for Majority voting

4.5.3 Analysis

a) For both majority and plurality voting, for a given value of K, as the number of experts increase, the level of confidence that we have in the results also increases. This is a logical result and it is the reason voting is used as opposed to a single expert.

b) For majority voting, as the number of possible symbols increases, K>>N, the confidence index becomes constant. It is also clear that consecutive even and odd experts converge. For example, from figure 10, we note that for N=2 and N=3, for values of K > 3, the two systems (one with 2 experts and another with 3 experts) will have the same confidence index C(2 experts)=C(3 experts) =2. The same phenomenon is noticeable for N=4 and N=5 systems, they converge to a confidence index C(4 experts) =C(5experts) =3. Based on other experiments with higher values of N, we find that for majority voting, as K >> N, the confidence index converge to $\lfloor N/2 \rfloor + 1$

c) For plurality voting, as the number of possible symbols increases, K>>N, the confidence index for all experts converges to a constant value =2.

4.6 Probability of reaching a consensus for majority and plurality voting as function of K

4.6.1 Purpose

The purpose of this experiment is to study the probability of reaching a consensus as function of the number of symbols that an expert can produce given a specific number of experts. We conduct the study for both majority and plurality voting. For a given number of experts "N", the simulator expands the decision tree and assesses the probability of reaching a consensus. We have conducted the experiments for values of N=2 through 5 where the number of symbols K varies from 1 to 10. Results are shown in figure 10.

4.6.2 Results

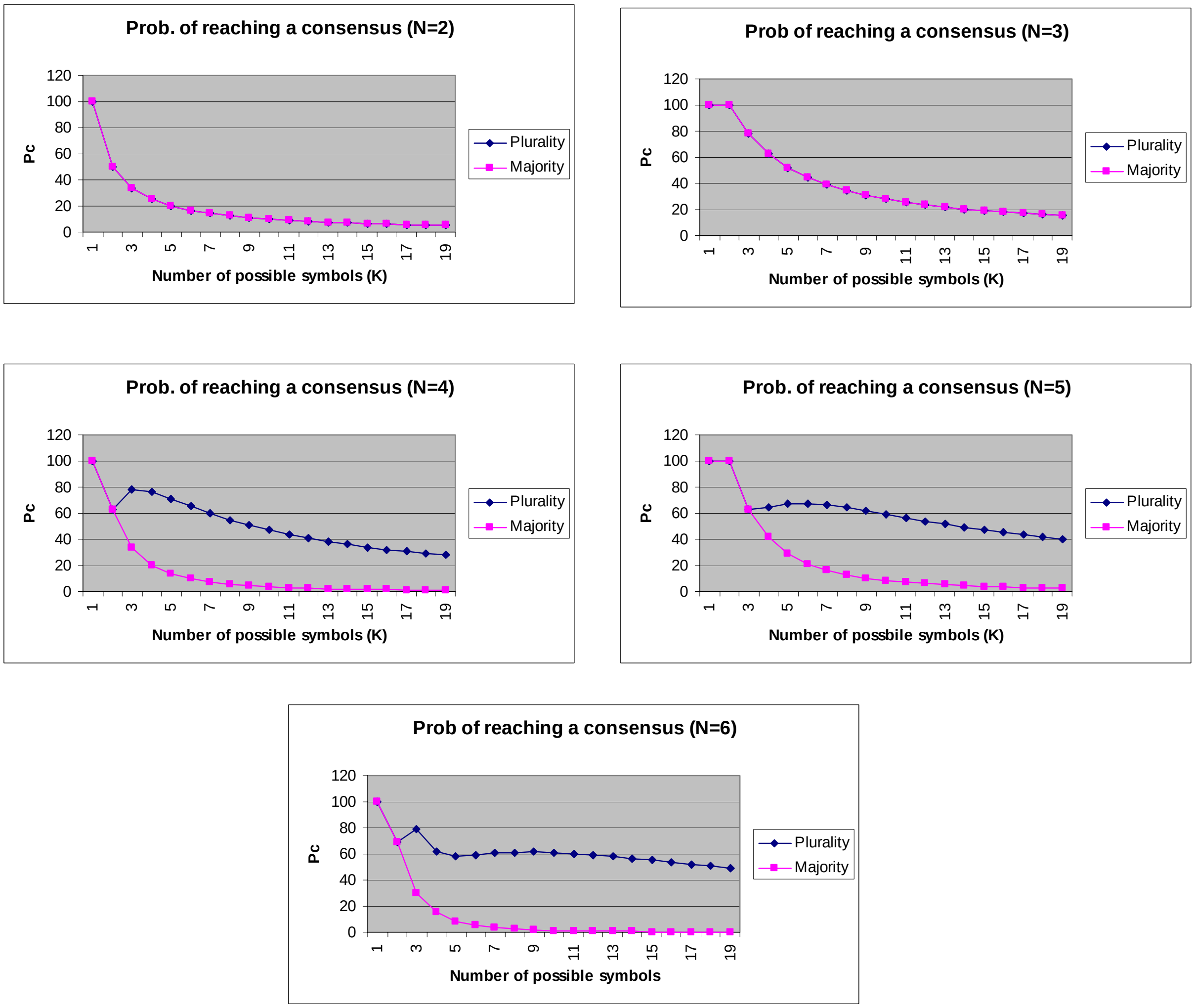

Figure 11 Pc for plurality and majority voting as function of K

4.6.3 Analysis

a) For a particular value of N, plurality voting has higher probability of reaching a consensus than majority voting for all K values.

b) For majority voting, as the number of possible symbols "K" increases, Pc is decreasing monotonically.

4.7 Confidence index for majority and plurality voting as function of K

4.7.1 Purpose

The purpose of this experiment is to study the confidence index as function of the number of symbols that an expert can produce given a specific number of experts. We conduct the study for both majority and plurality voting. For a given number of experts "N", the simulator expands the decision tree and assesses the confidence index C. We have conducted the experiments for values of N=2 through 4 where the number of symbols K varies from 1 to 10. Results are shown in figure 12.

4.7.2 Results

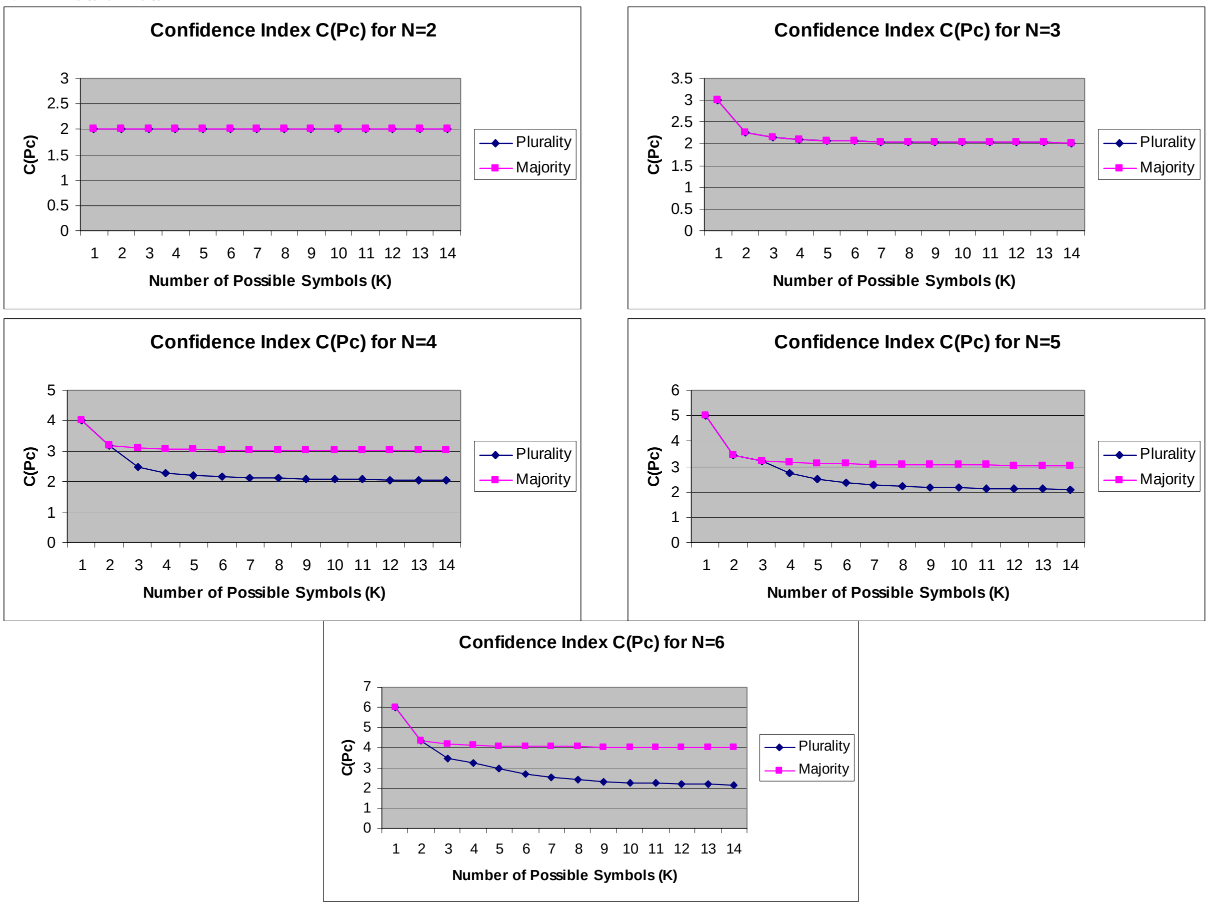

Figure 12 Confidence index for plurality and majority voting as function of K

4.7.3 Analysis

a) For a given number of experts N, majority voting always produces high confidence results than plurality voting for all values of K.

b) For majority voting, C for is asymptotic and they converge to a value of to $\lfloor N/2 \rfloor + 1$

c) For plurality voting, C is asymptotic to a constant value = 2.

5 RELIABILITY ANALYSIS

In the previous section, we studied the behavior of the voting system irrespective of whether it produces the correct output or not. In this section we study the reliability of the voting system R, and the confidences in the correct output C(R) as a function of N, K.

5.1 Reliability (R) as function of N and K

5.1.1 Purpose

The purpose of this experiment is to study the reliability of a voting system as function of the number of symbols that an expert can produce as well as the number of experts used.

5.1.2 Results

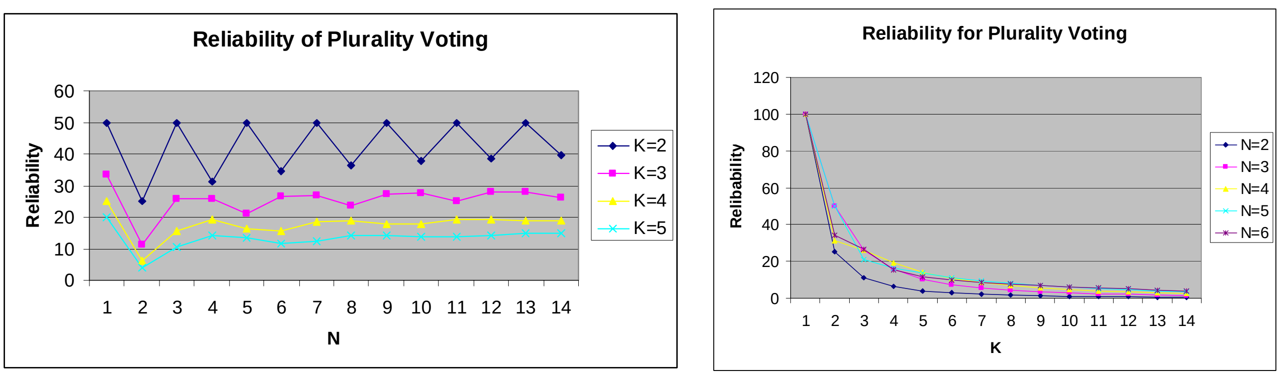

Figure 13 a) R of the voting system as function of N for various K, and b) R as function of K for various N

5.1.3 Analysis

a) From figure 13-a, we find that for a given number of experts, as the number of possible outcomes increases "K", the reliability of the system degrades. This is because the possibility of another outcome decreases the chance of obtaining the consensus and hence the chance that the consensus is correct.

b) For a given K, the reliability of the voting system is oscillating and the period is K. This means that for some given number of symbols the maximum boost in the reliability of the system is achieved in increments of K-experts.

c) Figure 13-b illustrates the severe degradation in the reliability of the voting system as the number of possible symbols increases given a uniform distribution of those symbols. This is because the distribution we simulated here is uniform distribution and hence the addition of new symbols is addition of unreliability factor to the system.

d) From figure 13-a, three experts can constitute a voting system whose reliability is competitive to the reliability produced by a larger number of experts.

5.2 Correctness Confidence C(R) as function of N and K

5.2.1 Purpose

The purpose of this experiment is to study the confidence index that we have in the correctness of the voting system output, C(R). We study C(R) as a function of the number of symbols that an expert can produce as well as the number of experts used.

5.2.2 Results

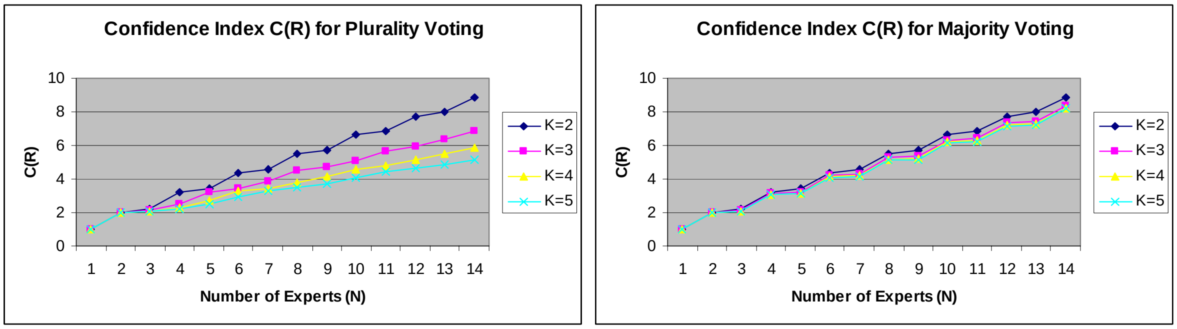

Figure 14 C(R) as function of N for various K for (a) plurality and (b) majority voting

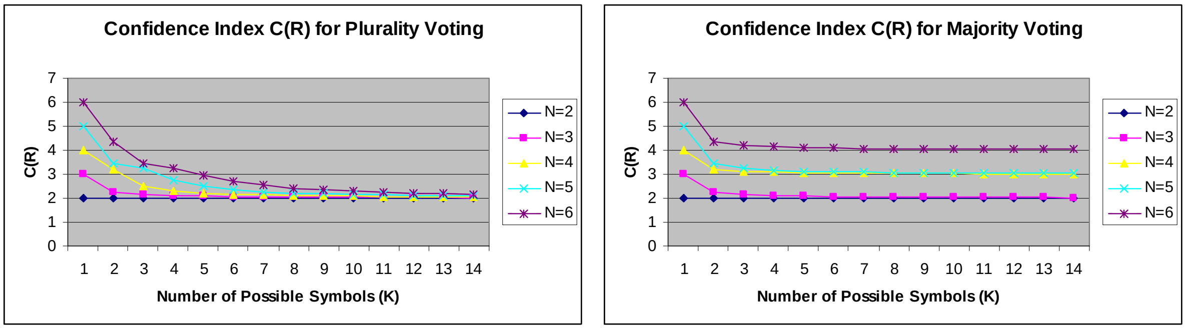

Figure 15 C(R) as function of K for various N, for (a) plurality and (b) majority voting

5.2.3 Analysis

a) From Figure 14, we find that for a given number of symbols, the confidence in the correctness of the system can be increased by increasing the number of experts.

b) From figure 14, it is notable that plurality voting is more sensitive (decreases) to number of symbols than majority voting. Hence, for majority voting, confidence is not much affected by increasing the number of symbols.

c) As noted before, from figure 15 it is clear that for a given number of experts as the number of symbols increases confidence in the correctness decreases. The asymptotic curves are previously discussed.

6 CONCLUSION

In this paper, we analyze the reliability and behavior of two voting schemas: majority and plurality voting. We have taken a synthetic experimental approach in which a simulator is used to expand all possible decisions using a decision tree. The decision tree is function of the number of experts and the number of possible symbols that each expert can produce. We have conducted a series of studies to analyze the behavior of these two voting techniques in terms of the reliability of the voting system, the probability of reaching a consensus, certainty in the consensus, confidence in the output, and confidence in the correctness of the output. The results of these studies are discussed in details together with their logical interpretation (section 4 and 5). We compare plurality and majority voting, and study the effect of the number of experts and the number of possible expert outputs on the behavior and reliability of the voting schema. We summarize some results that we discern from this experiment, using *uniform PMF* for each expert, as follows:

- For the same number of experts and number of possible output symbols, plurality voting has a higher probability of reaching a consensus than majority voting and the certainty in the consensus and the confidence in voting output is higher as well.
- We notice an oscillating behavior in the probability of reaching a consensus for majority and plurality voting. For majority the period is 2 and for plurality the period is K. Therefore, if we want to increment the number of units to use we would rather use increments of K-units to achieve maximum reliability increase.
- For majority voting, there is a trade off between reaching a consensus and obtaining a higher confidence in the result when increasing the number of experts from odd to even numbers.
- The probability of reaching a consensus is asymptotically decreasing for majority voting while it is asymptotically increasing for plurality voting.
- The ratio between the confidence in the results of majority voting and the confidence in the results of plurality voting is directly proportional to the number of possible output symbols K.

- For majority voting, as K >> N, the confidence index converges to $\lfloor N/2 \rfloor +1$. For plurality voting, as the number of possible symbols increases, K>>N, the confidence index for all experts converges to a constant value =2.

There are still several problems that can be analyzed using the same approach that we used in this study. For example, we can study the behavior of other voting schemas such as *k-out-of-n* or instance runoff voting. We can also study various PMF distributions and their effect on the voting behavior. The results that we obtained in this study reveals the behavior of the majority and plurality voting schemas and enable us to make better choices in terms of the schema to use and the number of experts given some domain requirements such as accuracy or consensus constraints.

A long-term objective of this research is to create reliability lookup tables for voting systems. Ultimately by studying different probability distributions of the output of the experts and running synthetic experiments like the ones we developed in this paper, we can construct lookup tables that the analyst uses to make decisions about the number of experts to use in implementing a voting system. Alternatively, a tool could be developed to study what-if scenarios by submitting parameters about the experts and obtaining results in terms of the reliability, confidence, and probability of reaching a consensus.

REFERENCES

[Ahammad *et.al.* 1987] M. Ahammad and M. Ammar, "Performance Characterization of Quorum-Consensus Algorithms for Replicated Data," *Proceedings of the 7th Symposium on Reliability in Distributed Software and Database Systems*, 1987, pp161-167.

[Ho *et.al.* 1994] T.K. Ho, J. J. Hull, and S. N. Srihari, "Decision Combination in Multiple Classifier Systems", IEEE Trans. on PAMI, 16(1), pp.66-75, 1994.

[Kumar *et.al.* 1991] Akhil Kumar and Kavindra Malik, "Voting Mechanisms if Distributed Systems", *IEEE Transactions on Reliability*, Vol 40, No. 5, 1991, pp593-600

[Lam *et.al.* 1994] L. Lam and C.Y. Suen, "A Theoretical Analysis of the Application of Majority Voting to Pattern Recognition", *Proceedings of IEEE,* pp. 418-420, 1994.

[Lam *et.al.* 1997] L. Lam and C. Suen, "Application of Majority Voting to Pattern Recognition: An Analysis of Its Behavior and Performance," IEEE Transaction on Systems, Man., and Cybernetics, Vol 27, No 5, September 1997.

[Levitin 2001] G. Levitin "Analysis and Optimization of Weighted Voting Systems consisting of Voting Units with Limited Availability", Reliability Engineering and System Safety, 73(2001), pp 91-100.

[Levitin *et.al.* 2001] G. Levitin and A. Lisnianski, "Reliability Optimization for Weighted Voting System", Reliability Engineering and System Safety, 71(2001), pp 131-138.

[Lin *et.al.* 2002]. X. Lin, S. Yacoub, and J. Burns, "Performance Analysis of Classifier Combination by Plurality Voting and their application in Pattern Recognition" Hewlett-Packard Laboratories, Technical Report, March 2002.

[Lorczak *et.al.* 1989] P. Lorczak, A. Caglayan, and D. Eckhardt, "A Theoretical Investigation of Generalized Voters for Redundant Systems", IEEE Computer Soc. Press, 21-23 June 1989, pp 444-451

[Nordmanna *et.al.* 1999] L. Nordmann and H. Pham, "Weighted Voting Systems," *IEEE Transactions on Reliability*, Vol 48, No 1, 1999.

[Parhami 1994a] B. Parhami, "Threshold voting is fundamentally simpler than plurality voting" *International Journal on Reliability, Quality, and Safety Engineering*, Vol 1(1), 1994, pp95-102.

[Parhami 1994b] B. Parhami, "Voting Algorithms" *IEEE Transaction on Relaibility* , Vol 43, 1994, pp617-629.

[Paris 1986] J. Paris, "Voting with WiTesses: A Consistency Scheme for Replicated Files," *Proceedings of the 6th International Conference on Distributed Computing Systems*, May 1986, pp606-616.

[Xu *et.al.* 1992] L. Xu, A. Krzyzak, and C. Suen, "Method of Combining Multiple Classifiers and Their Application to Handwriting Recognition", *IEEE Transactions on Syst. Man Cybernet*, Vol. 22(5), pp. 418-435, 1992.

[Yacoub 2002] Sherif Yacoub, "Analysis of the Behavior and Reliability of Voting Systems Comprising Tri-State Units" submitted to the 21st *IEEE Symposium on Reliable Distributed Systems* (SRDS), Osaka University, Suita, Japan 2002.

[Yacoub *et.al.* 2002] S. Yacoub and S. Simske, "Meta-Algorithmic Patterns", Hewlett-Packard Laboratories, Technical Report, April 2002.