Multiple Experts Voting: Two Rank Refinement Strategies

Seppo Puuronen*, Vagan Terziyan**, Boris Omelayenko**

(*) Department of Computer Science and Information Systems,

University of Jyvaskyla, Finland, e-mail: sepi@jytko.jyu.fi

(**) Department of Artificial Intelligence and Information Systems,

Kharkov State Technical University of Radioelectronics, Ukraine, e-mail: vagan@milab.kharkov.ua

Abstract

Often there is a need to collect expertise from more than one expert for decision making. Currently, the fast development for example in telecommunications, Internet, and WWW offers possibilities to collect even remote experts into groups. When opinions are collected from more than one expert for decision making, as in Group decision support systems (GDSS), one must either select the opinion of one expert or pool experts' opinions. Several different ways of pooling the expert opinions have been suggested, because it is assumed that final opinion has more validity if it forms some kind of consensus among the experts. One classical way of pooling is the use of voting. Our goal is to develop multiple expert voting system that is able to produce the final opinion of an expert group. In this paper we discuss a weighted voting system in which each voter express at most one opinion during each voting situation and the opinion receiving most support from voters wins. In our weighted approach each voter has in each vote as many votes as his weight indicates. These weights are changed between the voting situations taking into account the quality of decision. Our voting system includes four main parts implementing four so called strategies: strategy of deriving final opinion, strategy of quality evaluation of the derived opinion, strategy of voting, and strategy of expert ranking and rank refinement. In this paper we fix the first three strategies and investigate how different rank refinement strategies influence during voting process and to the final result. We have developed and experimentally evaluated two such strategies: the strategy of equal demands to leaders and losers and the strategy which sets higher demands to leaders than to losers. In an example we found, that the first strategy produces very fast clear subgroups of leaders and losers, but the quality of the final decision varies greatly from vote to vote. We also found that when the second strategy is used ranks change more slowly and the variation in quality is not so big, but rank updating requires more resources.

1. Introduction

Often there is a need to collect expertise from more than one expert for decision making. Group decision support systems (GDSS) have been developed to support the work of decision making groups. Nowadays, these systems offer several kinds of support for brainstorming, discussions, comments' exchange, document editing, and voting. Currently, the fast development for example in telecommunications, Internet, and WWW offers possibilities to collect even remote experts into groups. When in a decision making situation more than one expert is used, one must either select the opinion of the best expert or pool experts' opinions [7, 3].

In this paper we continue research presented in [5].

2. Model of the voting system and basic concepts

We define the multiple expert voting system as a six-tuple <*S,D,Q,V,P,T*>, where:

$S=\{S_1,S_2,\ldots,S_n\}$ is the set of n experts. We assign a numerical rank to each expert. These rank values form the set $r=\{r_1,r_2,\ldots,r_n\}$ where each r is the rank of the corresponding expert which presents his authority in the domain area. The rank values are changed during the voting process.

$D=\{D_1,D_2,\ldots,D_d\}$ is the set of d opinions. Domain may be structured, and then each opinion consists of m components $D_i=(C_1,C_2,\ldots,C_m)$. The value of each component is taken from the corresponding set of values E. There are restrictions concerning the valid combinations of the components. These restrictions are presented by predicate Di, as follows:

$$Di(C_1, C_2, ..., C_m) = \begin{cases} 1, & \text{if a combination } (C_1, C_2, ..., C_m) \text{ is valid in the domain,} \\ 0, & \text{otherwise.} \end{cases}$$

$Q =< Q_1, Q_2, ..., Q_q >$ is an ordered set of all the q questions, asked for experts during the voting process.

$V =< V_1, V_2, ..., V_q >$ is an ordered set of the q answers on corresponding questions of the set Q. Initially, it contains undefined answers, with the meaning "no answer given". Each answer $V_i \in V$ must be filled with only one opinion, constructed using the method to derive the final opinion on the basis of the n experts' opinions. The resulting opinion must belong to the set of all possible domain concepts: $V_i \in D$. The final opinions are derived one-by-one in the order of the questions.

P is a semantic predicate, which defines piece of knowledge about relationship in the domain between the sets Q, D and S:

$$P(Q_i, D_j, S_k) = \begin{cases} 1, & \text{if the experts } S_k \text{ gives opinion } D_j \text{ as his answer to the question } Q_i; \\ 0, & \text{otherwise.} \end{cases}$$

The value of this predicate is updated just after experts' vote. Notice that the voting system of Figure 1 does not include any special domain knowledge outside the opinions of the experts.

$T = \{T_1, T_2, ..., T_t\}$ is the set of t techniques to process experts' opinions. In each voting process a group of four techniques are fixed. This group can be defined as a four-tuple $\langle MS, QS, RS, VS \rangle$, where: MS implements the strategy that is used to derive the final opinion, QS implements the strategy to measure the quality of the final opinion, RS implements experts' ranks recalculation process (rank refinement strategy), and VS implements the voting strategy that defines the ordering and grouping of the questions asked from the experts.

3. Strategies of techniques

Each group of techniques implements the four strategies: strategy to derive the final opinion, strategy to evaluate the quality of the final opinion, voting strategy, and strategy of rank refinement. Each strategy determines one aspect of the behaviour of the system and the first three of them will be shortly discussed in this chapter and their implementation is fixed for the rest of this paper. The fourth, rank refinement strategy will be discussed more deeply in the next chapter.

3.1. A strategy for deriving the final opinion

There exist several ways to derive the final opinion from the different opinions of experts. In this paper we select the most supported opinion as the final opinion. When the most supported opinion is determined we take into account the ranks of the experts so that higher ranked experts (later called as leaders) have more influence than lower ranked experts (later called losers).

The most supported opinion concerning question Q is derived in the following way: first each expert gives his votes about the usage of each component of the opinion, then ($n \times m$)-matrix SC^Q is formed. This defines relationships between the experts S and their opinions about components C_i as opinion V on the question Q, which can be presented formally as:

$$\forall S_i \in S,\ \forall D_i = (C_1, C_2, ..., C_m),\ D(C_1, C_2, ..., C_m) \& P(S, Q, D_i) \Rightarrow (SC^Q_{i,q} = C_q),\ q = \overline{1,m}.$$

The technique takes into account the rank of each expert which defines the weight of his vote among all the other votes. Let r^v_i be the rank of i-th expert before v-th voting. Let the vector $VOTE^Q$ contain the opinions of the experts in the current vote concerning question Q. It is derived from the matrix SC^Q as follows:

$$VOTE^Q_q = j^Q_q - y^Q_q, \forall q \in \overline{1,m}, \quad \text{where} \quad \varphi^Q_q = \sum_{\substack{i, \\ \forall i (SC^Q_{i,q} = 1)}}^{n} r^v_i, \quad \psi^Q_q = \sum_{\substack{i, \\ \forall i (SC^Q_{i,q} = 0)}}^{n} r^v_i.$$

In general case, after derivation the *VOTE* vector can include an impossible opinion due to inconsistency in the experts' knowledge expressed by components of opinion. If the domain area is such then some domain-specific algorithm is needed to fix the most supported opinion.

3.2. A voting strategy

There are several ways to order and group the questions given to experts for voting, i.e. expressing their opinions. We have discussed three voting strategies in [6]. In this paper we use batch voting strategy because it is not too sensitive on order and makes rank evaluation more flexible.

According to the batch voting strategy experts vote the same questions k times repeating their correct or wrong answers. Formally we define the sets Q^B (questions) and V^B (answers) as:

$$Q^B = \underbrace{Q \cdot Q \cdot \ldots \cdot Q}_{k\ times},\ V^B = \underbrace{V \cdot V \cdot \ldots \cdot V}_{k\ times},$$

where operation « · » denotes concatenation of two ordered sets.

Thus k series of q most supported opinions are produced. The last most supported opinion is the final opinion and thus the last q elements of the set V^B will form the resulting V as follows:

$$V_i = V^B_{q \times (k-1)+i},\ i = \overline{1,q}\ .$$

Expert ranks are changed during this iterative voting process according to the relationship between expert's opinion and the most supported one. Thus this strategy requires $k*q$ rank recalculations.

4. Rank refinement strategies

There are several different ways for rank refinement. In this paper we use rank values in the interval [0,n] where n is the number of experts. In this chapter we discuss two techniques to implement two different strategies. The first one is based on the idea that the changes in ranks do not depend on the expert belonging among those with ranks above the value n/2 or among those with ranks below the value n/2. Only the distance from the value n/2 has effect. Also the changes in both directions have the same size. We call this as a strategy with "equal requirements to leaders and losers". The second strategy is based on the idea that those who have higher previous ranks should have bigger responsibility than those who have smaller ranks. This means that an expert who has rank above n/2 in the case of having "wrong" opinion receives bigger negative change in rank than an expert who has already rank below n/2. Also when an expert has "right" opinion the positive change in his rank depends on his previous rank value so that those who have smaller ranks will receive bigger positive changes than those who have higher rank values. We call this as a strategy with "greater requirements to leaders than losers". These are discussed in their own subchapters but first we describe the common rank refinement formula where these strategies are implemented as part of the formula.

The main formula used to refine the rank of each expert is:

$$r_i^{v+1} = r_i^v + \Delta r_i^v,$$

where the value of Δr_i^v (the amount of punishment or prize), is calculated by the formula:

$$\Delta r_i^v = \delta_i^v \cdot \frac{\mu^v \cdot (\mu^v - con_i^v)}{m},$$

where:

$$\sigma_i^v = \frac{v}{v+n-1};\ con = \frac{2}{3} \cdot m;\ \mu^v = \frac{1}{n} \cdot \sum_j^n con_j^v \text{, and}$$

the value $\delta_i^v = \dfrac{r_i^v \cdot (n - r_i^v)}{n-1}$ depends on the rank refinement strategy.

The formulas, above, are based on the following basic assumptions:

- All the experts have the same initial rank which is equal to $\frac{n}{2}$.
- An expert's rank is always bigger than zero and less than the number of experts.
- After each vote the rank of each expert is recalculated.
- After each vote an expert improves his rank if his opinion has less conflicts with the most supported opinion, than the experts on an average. Otherwise, he loses part of his rank. In the main formula, above, this is achieved by the multiplier $\frac{\mu^v - con_i^v}{con}$, where con (maximum possible conflicts between opinions) is used to normalise the result.
- Expert's rank should not be changed if the expert does not participate voting.
- Expert's rank should not be changed if his opinion has as many conflicts with the final opinion as experts have on an average.
- The value of an expert responsibility grows from one vote to another. Thus an expert cannot lose or improve his rank essentially during the first vote. However, the maximum possible change in the ranks grows from vote to vote according to the multiplier σ_i.

4.1. The strategy "equal requirements to leaders and losers"

This strategy is based on the idea that the changes in ranks do not depend on the expert belonging among those with ranks above the value n/2 (leaders) or among those with ranks below the value n/2 (losers). There are several ways to implement this strategy but we have chosen to define the value of δ_i^v using the formula:

$$\delta_i^v = \frac{2 \cdot r_i^v \cdot (n - r_i^v)}{n}.$$

This formula has the following characteristics:

- The amount of change is biggest for an expert with the rank equal to $\frac{n}{2}$.
- The amount of change converges to zero when expert's rank approaches zero or n.

This strategy is very demanding to the experts. Even if an expert makes only a few mistakes in the very beginning of the voting process and falls into the group of "losers" he encounters quite big difficulties to restore his rank. On the other hand, if an expert manages in the beginning climb into the group of "leaders" he will quite probably stay there. Also, an expert with the smallest possible rank has an equal responsibility for a mistake as an expert with the highest possible rank. This gives no chance to a loser to catch up a leader. It is reasonable to use this strategy in applications where there are many experts in the beginning and goal is to select only some of them to continue voting after some initial questions.

4.2. The strategy "greater requirements to leaders than to losers"

This strategy is based on the idea that the changes in ranks depend on the expert belonging to those who have the rank above n/2 or below n/2. As the name of the strategy reveals the experts who have higher previous ranks will have bigger responsibility than those who have smaller ranks. When an expert belonging to leaders gives more than on an average conflicting opinion his rank will be changed more towards zero than the rank of an expert belonging to losers in the same situation. Changes also to the opposite direction, towards n are analogous so that leaders receive smaller change than losers. Also this strategy can be implemented in several ways, but we have chosen to define the value of δ_i^v using the formula:

$$\delta_i^v = \frac{r_i^v \cdot (n - r_i^v)}{n - 1}$$

This formula has the following characteristics:

- The positive amount of change is biggest for an expert with the rank close to zero.
- The negative amount of change is biggest for an expert with the rank close to *n*.
- The positive amount of change converges to zero when expert's rank approaches *n*.
- The negative amount of change converges to zero when expert's rank approaches zero.

This strategy is much more permitting to the mistakes of experts than the previous one. If an expert makes a few mistakes in the very beginning of the voting process and falls to the group of losers he will not be as responsible for new "mistakes" as a leader and he still have a possibility to climb back later. On the other hand, if an expert succeeds very well in the very beginning and becomes a leader then he has high responsibility for any mistakes in the future. It is reasonable to use this strategy in applications where opinions of all experts are always wanted and where experts are wanted to be motivated to learn.

5. Experiments

In this chapter we describe the domain area of temporal intervals and then in separate subchapters the experiments of the behaviour of the two strategies with an example taken from this domain area. The domain area of Allen's [1,2] relations between two temporal intervals is structured and has restrictions on the component combinations if these intervals are presented using the endpoints of the intervals and their relations. The temporal domain is defined according to Allen [1,2] as a set of 13 basic relations R_i for temporal intervals.

Let there be four experts voting on three tasks in Allen temporal domain. Each expert has expressed his three opinions on the three questions (q=3), as shown in Table 1.

Table 1. Expert opinions in the example

Expert	1st question	2nd question	3rd question
S_1	T_1 during T_2	T_3 after T_4	T_5 includes T_6
S_2	T_1 overlaps T_2	T_3 meets T_4	T_5 finished by T_6
S_3	T_1 starts T_2.	T_3 overlapped by T_4	T_5 after T_6
S_4	T_1 finished by T_2	T_3 before T_4	T_5 starts T_6

The most supported opinions, obtained after processing expert opinions with both strategies are presented in Figure 7. Figure 7 shows, that both strategies form similar consensus after 7 iterations. Both of them give the same most supported opinions on the 1st task (*overlaps*). Most supported opinions on the 2nd task differ only in one endpoint relation, as well as the 3rd most supported one. Figure 7 also shows that the most supported opinion obtained with the first strategy after 7 iterations is equal to one obtained by the second strategy after 21 iterations.

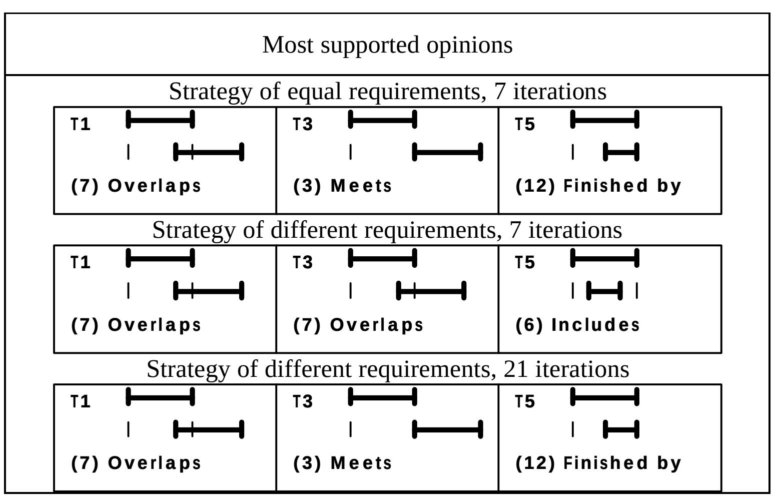

Figure 7. Comparison of both ranking strategies

6. Conclusions

When opinions are collected from more than one expert for decision making one must either select the opinion of one expert or pool experts' opinions. Several different ways of pooling the expert opinions have been suggested, because it is assumed that final opinion has more validity if it forms some kind of consensus among the experts. One classical way of pooling is the use of voting.

In this paper we have discussed a weighted voting system in which each voter express at most one opinion during each voting situation and the opinion receiving most support from voters wins. In our weighted approach each voter has in each vote as many votes as his weight indicates. These weights are changed between the voting situations taking into account the quality of decision.

Our voting system includes four main parts implementing four so called strategies: strategy of deriving final opinion, strategy of quality evaluation of the derived opinion, strategy of voting, and strategy of expert ranking and rank refinement. Our focus in this paper is on the rank refinement strategy and we have developed and experimentally evaluated two such strategies.

We found, that the strategy of equal demands to leaders (weight over middle value) and losers (weight under middle value) produce very fast clear subgroups of leaders and losers. But quality of the final decision varies greatly from vote to vote. This strategy demands less updating resources, but gives rough and varying results. It seems that it might be useful in time-critical applications where the lasting quality is not the main goal.

The other strategy which sets higher demands for leaders than losers changes ranks more slowly. Also the variation in quality is not so big, but rank updating requires more resources. It seems that this strategy might be useful in applications where steady quality is important.

In our test example both strategies gave same final results but there were clear difference in the number of iterated voting situations needed. This difference is expected still increase when the number of experts is raised. It seems that domain related context-dependent methods are needed to select appropriate strategies for voting systems. Our approach can be further developed towards multi-level system. This approach might be extremely important when there exist many distributed experts and limited resources in time-critical domain area.

Acknowledgement: This research is partly supported by the grant of the Academy of Finland.

References

[1] Allen, J.F., Maintaining Knowledge About Temporal Intervals, *Communications of the ACM*, V.26, No.11, 1983, pp. 832-843.

[2] Allen, J.F., Temporal Reasoning and Planning, In: Allen J.F. (ed.), *Reasoning About Plans*, Morgan Kaufmann, San Mateo, 1991, pp. 8-33.

[3] Barrett, A.R. and Edwards, J.S., Knowledge Elicitation and Knowledge Representation in a Large Domain with Multiple Experts, *Expert Systems with Applications*, Pergamon, V.8, No.1, 1995, pp. 169-176.

[4] Becker, K. and Bacelo, T., Integrating Voting Techniques to a Discussion Rationale Model-based GDSS, In: *Proceedings of the 4th Conference of the International Society for Decision Support Systems, ISDSS'97*, July 21-22 1997, Lausanne, Switzerland, http: //isdss97.unil.ch/isdss97/papers/71.htm.

[5] Kaikova, H. and Terziyan, V., Temporal Knowledge Acquisition From Multiple Experts, In: Shoval P. & Silberschatz A. (Eds.), *Proceedings of NGITS'97*, Neve Ilan, June 30 - July 3, 1997, pp. 44-55.

[6] Omelaenko B., Terziyan V., Puuronen S., Multiple Expert Knowledge Acquisition: Experimental Investigation of Three Voting Strategies, In: *STeP-98 - Human and Artificial Information Processing*, Finnish Conference on Artificial Intelligence, 7-9 September 1998, Jyvaskyla, Finland, Publications of the Finnish AI Society, 1998, pp. 88-97.

[7] Taylor, W.A., Weimann, D.H. and Martin, P.J., Knowledge Acquisition and Synthesis in a Multiple Source Multiple Domain Process Context, *Expert Systems with Applications*, Pergamon, V.8, No.2, 1995, pp. 295-302.