Capabilities-based force aggregation using random sets

Pontus Svenson
Dept of Data and Information Fusion
Swedish Defence Research Agency
SE 164 90 Stockholm
Sweden
ponsve@foi.se
http://www.foi.se/fusion

***Abstract*— Force aggregation is one of the most important functionalities of a situation analysis system. In order to reduce the amount of information displayed for an analyst, it is vitally important to cluster information that belongs together and display the aggregated information for the cluster instead of all the original objects.**

In order to do this, two different kinds of algorithms are necessary. First, we must have a method for grouping objects that belong together. This problem is often referred to as clustering or association; it is a variant of the NP-complete graph coloring problem.

Second, a group of objects that belong together must be classified. There have been some methods proposed for doing this. All of the present alternatives for aggregation rely on doctrinal information. However, in the new kind of situations that face us, it is increasingly likely that we will meet organizations that do not follow a strict doctrine in their organization. Instead, they will use task-forces or ad-hoc forces that are organized to solve a specific objective.

Here, we present a formalism for doing classification of task-forces based on less amount of doctrinal knowledge. The kind of doctrinal knowledge required by the approach suggested here is similar to the one needed to put together task-forces for solving a specific mission, ie, it is capabilities oriented. Using random set theory, we describe several different ways of force aggregation and present results from experiments performed with them. User interaction could be used to further enhance the method presented here.

I. Introduction

Information fusion [1] deals with ordering, fusing and classifying information in order to display it to a user in the best possible way. In order to reduce the amount of information, it is necessary to cluster or group data together, and present aggregated information. Traditionally, such aggregation is done using methods based on knowing the detailed doctrines of actors on a battle-field. However, in the new kinds of situations facing us in operations other than war (OOTW) applications, such doctrines are not available. In this paper, we present a possible way of aggregating information using less strict and detailed doctrinal information. The approach is based on generating lists of capabilities for observed entities and task-forces that the system recognizes, either because a user has input them or because an automatic plan-generation module has generated them. The method uses random sets, and is meant to be used in conjunction with both sensor and human intelligence.

This paper is outlined as follows. Section II introduces the problem and lists some previous approaches for force aggregation. In section III, we motivate the use of task-forces and capabilities. Section IV presents the mathematics of the presented method, while section V discusses how user interaction could be used to improve it. Finally, in section VI we present the results of a computer simulation of the method, and section VII contains our conclusions.

II. Force aggregation and operations other than war

In order to reduce the amount of information displayed for an analyst, it is vitally important to cluster information that belongs together and display the aggregated information for the cluster instead of all the original objects. Force aggregation is one of the most efficient ways of reducing/transforming the amount of information presented to decision-makers.

In order to do this, two different kinds of algorithms are necessary. First, we must have a method for grouping objects that belong together. This problem is most referred to as clustering or association (see, *e.g.*, [2], [3]); it is a variant of the NP-complete graph coloring problem. Second, a group of objects that belong together must be classified. There have been some methods proposed for doing this [4], [5], [6], [7]. All of the present alternatives for aggregation rely on doctrinal information.

The world, however, has changed. The likelihood of an armed attack by an adversary that follows strict doctrines has been reduced significantly. At the same time, the increased level of international engagement of Swedish armed forces in peace-enforcing or other operations other than war overseas has increased the importance of providing decision-support tools that are useful in the near future. In these kinds of situations it is increasingly likely that we will meet organizations that do not follow a strict doctrine in their organization. In many OOTW situations, we will be faced with groups of opponents that organize spontaneously, *e.g.*, legal demonstrations that degenerate into riots. A slightly more organized opponent will use task-forces or *ad hoc* forces that are specified to solve a specific objective.

An added difficulty in such situations is the fact that there is not just one opponent, but many. In peace-keeping operations, for instance, there will be at least two opposing sides. It is also important to be able to fuse and present to a user information on friendly or neutral units. There might be several entities belonging to several different types of organizations that need to be displayed for the user, both in order to avoid firing on neutral or friendly units, and in order to determine if a hostile group threatens a neutral group. Another possible extension of the work considered here is to use it to label/classify the health status of own forces. This approach would consider how the state of health of a soldier influences the capabilities both of the individual and of the groups to which they belong. By monitoring soldiers with for example sensors that measure heart-rate or perspiration, we could get a good view of the status of groups. This could be combined with systems for planning, to show how the physical state of the soldiers affects their ability to solve given tasks. For instance, in a riot situation such a display could be used to determine if a group of soldiers should try to reason with the rioters or call for backup: if their heart-rate and perspiration is high, it is likely that they are too excited to be able to convince the rioters to cease, and the commander must choose some other option.

In this paper, we assume that the clustering problem has already been solved, and that we are given a list of capabilities/features of groups. The task is to classify this group. For future work, it would be interesting to try to extend the method presented here and use it to aid in clustering, or for tracking groups of, *e.g.*, rioters.

III. Capabilities-based task-forces

Here, we present a formalism for doing classification of task-forces based on less detailed doctrinal knowledge as well as user interaction. The kind of doctrinal knowledge required by the approach suggested here is similar to the one needed to put together task-forces for solving a specific mission, *i.e.*, it is capabilities oriented.

In addition to the benefit of being able to classify and label units that don't use doctrines, capabilities-based force aggregation can also be interfaced to other parts of a fusion system, such as plan-recognition, in a natural way. Much work has recently been done on recognizing enemy behavior and building Bayesian nets that accurately represent a situation (*e.g.*, [8]). By using partial Bayesian nets to represent capabilities, this work could be connected to the present, giving us the possibility of generating task-forces based on the perceived actions of the enemy units. Such partial Bayesian nets have previously been used by some authors [9], [10] to represent situations.

Figure 1 shows a conceptual view of the relations between an enemy's goals and its plans, actions and capabilities. The maps from $\{goals\} \rightarrow \{plans\} \rightarrow \{capabilities\}$ and $\{capabilities\} \rightarrow \{plans\}$ are actually sort of inverse maps: They specify that in order to achieve these goals, I must do these things and have these capabilities. Both actions and capabilities can be observed by sensors.

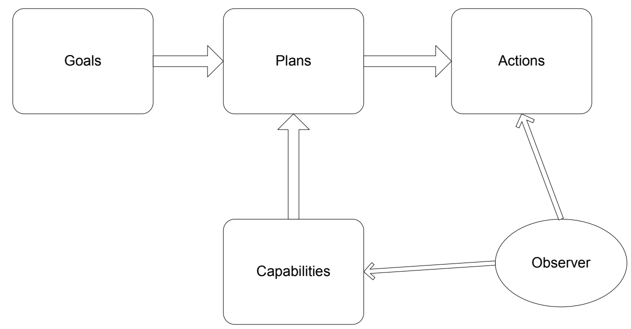

Fig. 1. Conceptual view of relations between observed capabilities and goals, actions and plans. Actions are determined by the plans of the enemy. The plan is developed in order to fulfill the enemy goals, but is constrained by the available resources. We can observe both enemy actions and their capabilities, and hence try to infer plans using two different sets of observations.

Methods for plan-recognition that use the movement patterns or actions of the enemy as input can give us suggested plans. Plans can also be inferred by observing capabilities and noting how the available capabilities and resources constrain the actions of the opponent. Determining plans from capabilities in this way would require the use of ontologies (*e.g.*, [11]) and some sort of simulation module that determines what the options of the enemy are; user interaction would then probably be required in order to make a reasonable choice between them. By having several different means of predicting plans, we get another opportunity to discover deception and misinformation: cases where plans determined using different methods don't match.

Note that a capability that comes from a specific entity (object or person) might be present in several different task-forces. This might occur, for example, in situations where we do not know the allegiances of all objects or persons.

IV. Method

The most natural way of representing the capabilities associated to a given task-force is as a random set [12], [13]. A random set is simply a set-valued stochastic variable.

In the following, let Θ denote the set of possible classifications or capabilities. 2^Θ is the set of subsets of Θ. Let A be a representation of capabilities needed to reach some goal or describing some known task-force. In general, we will assume that a type of task-force can be modeled as a random set $A : \Omega \rightarrow 2^\Theta$. This just means that we can list a number of focal sets of A and give the probability assigned to each. Each probability is a measure of the likelihood that such a task-force will have the capabilities listed. Example:

$$\begin{aligned} A : P(\{a, f, g\}) &= 0.5 \\ P(\{a, b\}) &= 0.2 \\ P(\{a, c\}) &= 0.25 \\ P(\{a, d\}) &= 0.05 \end{aligned}$$

Here and in the rest of this paper, $a, b, \ldots$ represent arbitrary capabilities. We will use A to indicate task-force capability-lists that are present in the fusion system and used to classify and label discovered task-forces. Possible sources of such A's could be: automatically generated from plan-generators, user input, comparison to old scenarios/using organizational memory to recognize situations/capabilities.

It is likely that most such "task-force templates" stored in the system will include only a few focal-sets, in order to make them comprehensible to the human operator of the system.

We also have an object of interest, whose capabilities have been determined. This too we represent as a random set

$$\begin{aligned} Z : P(\{a, b, c\}) &= p_1 \\ P(\{a, b\}) &= p_2 \\ P(\{a, c, d\}) &= p_3 \\ P(\{a, d, e\}) &= p_4 \end{aligned}$$

and is assumed to represent some clustered object or a group of actors in a crowd that either the system or the human operator have determined belong together.

Note that in order to represent the capabilities as random sets, it is necessary to remove duplications of capabilities. For the purpose of presenting information to a user, this is not a severe restriction. However, if a particular capability vanishes when used (for example, bombs), it might be necessary to add a numerical marker to it, indicating how large the capacity is. Another possible way of dealing with such situations is to use random multi-sets instead of random sets; this would allow repetitions of capabilities.

Note that the use of random sets here is essential and cannot be replaced with Dempster-Shafer mass functions. This is because we here have a different interpretation of the probability mass given to a certain set: by allocating a probability p to a set $\{a, b, c\}$, we mean that the observed object has all the capabilities a, b, and c with probability p. In the Dempster-Shafer interpretation, on the other hand, such an allocation would mean that our belief that one of a, b, or c is correct is p, but we cannot make any further statement regarding which one is the correct capability. The difference is the same as the difference between conjunctions and disjunctions. It is possible to extend the representation of capabilities and use Dempster-Shafer theory to represent an additional layer of uncertainty in the observations by considering sets of sets, *i.e.*, using 2^{2^Θ}. This could be done in two separate ways, analogous to the disjunctive or conjunctive normal forms of logical expressions, but here we do not consider either of those possibilities.

Now we seek to combine the task-force template and the clustered object in order to get a matching or fitness value for how well the observed task-force matches the stored ones. By calculating this matching value for a number of different task-forces, the system can indicate to the user what type of task-force that it believes the clustered object represents.

There are a number of different ways of calculating this. One simple is

$$\sum_{x,y \in \Theta} \sum_{t \in T} f(x, t) g(y, t), \tag{1}$$

where T is some set to which we can compare the elements of A and Z. The simplest would be to simply let $T = \Theta$. Another alternative is to allow T to depend on A and Z. The functions f and g measure the similarity of their arguments; they could be identical or different to take into account the added uncertainty in observations.

Another possibility is to use the Dempster-Shafer [14] conflict

$$\sum_{x \cap y = \emptyset} A(x) Z(y) = 1 - \sum_{x \cap y \neq \emptyset} A(x) Z(y). \tag{2}$$

There are however a number of reasons why the Dempster-Shafer conflict is not good as a measure of similarity of random sets. The DS conflict measures the amount of contradiction of the combined beliefs and assumes that there is a correct alternative in Θ, whereas we here have one in 2^Θ. It would be correct to use the DS conflict if the beliefs to be combined were elements of 2^{2^Θ} instead of 2^Θ. Since we here are not interested in the degree of internal conflict of a report on a task-force it is better to use a function that explicitly measures the similarity of the random sets.

An alternative that partially overcomes the limitation of the DS conflict is the modified Dempster-Shafer combination introduced by Mahler [15] and Fixsen-Mahler [16]. This rule integrates *a priori* knowledge in the rule

$$\sum_{x \cap y = \emptyset} A(x) Z(y) \frac{\Gamma(x \cap y)}{\Gamma(x)\Gamma(u)}, \tag{3}$$

where Γ is a measure that can be taken to represent prior knowledge. In our context, Γ could be the prior probability that the object of interest possesses the capabilities.

The measure introduced below in equation (7) can be seen as an application of this rule, but with a different interpretation and implementation of Γ.

The first fitness measure used in the experiments presented in section VI is

$$P(A = Z). \tag{4}$$

This equation corresponds to taking the trace of the combined tensor constructed by taking the outer product of Z and A represented as vectors, and seems at first glance to be the most natural measure. It turns out, however, to be worse than the next two measures.

Recall that Z denotes the clustered object or group of actions that we wish to label, while A is the templates or automatically generated capabilities lists that we wish to compare to. Note that it is also possible to compare the task-force to fragments of templates. Such fragments would probably be easier to generate in a real system.

When having such partially or fragmentary task-forces, the most natural measure seems to be

$$P(A \subseteq Z). \tag{5}$$

This equation simply checks to what extent the template fragment is present in the observed entity. If we define A_i as fragments of templates, then we would compare

$$P(A_i \subseteq Z) \tag{6}$$

and output a list of A_i and their probabilities to the user.

Motivated by the Fixsen-Mahler rule above, we also tried to determine the match or fitness between the observed object and the task-forces by using a "fuzzy" measure

$$\sum_{x,y \in \Theta} P(A = x)P(Z = y)S(x, y), \tag{7}$$

where the function S measures the similarity between its arguments. This equation is the one to use when we are not sure if the capabilities in Z are determined completely accurately. If there is a finite probability that a capability could be mistaken for another in the observations, this could be compensated for in the S function.

Simply by playing around with the order of A and Z and choosing different operators to put in-between, it is possible to come up with a long list of other alternative measures. Using

$$P(Z \subseteq A) \tag{8}$$

could also be useful. A here should be interpreted as the capabilities required to reach a major goal. In the case where Z is a proper subset of A, then taking $A \setminus Z$ could be used to determine what capabilities the enemy might need to obtain in the future. It could be used in order to prioritize what supply lines or resources that should be destroyed or protected in order to prevent them from being captured by the opponents.

Another possibly representation of the task-forces is to try to model the problem with true Dempster-Shafer structures. As mentioned above, this would entail adding a level of abstraction by using sets of sets of capabilities. In this case, we could use

$$\sum_{x} P(Z \subseteq x)P(A \subseteq x) \tag{9}$$

to compare them. This equation is inspired by the approach used in [4]. For the application presented in this paper, however, it is not as useful. This is because here we do not have the Dempster-Shafer interpretation of Z, as explained above.

V. User interaction

As mentioned above, user interaction could be used to improve the method in many different ways.

Interaction could be used to control the task-force generation. A user could describe possible goals for the enemy, the system could generate plans needed to reach that objective and another module could determine the capabilities needed for that. If the user distrusts the results of the computer system, it should be possible to manually change them. In such cases, however, it would be useful if the system also stored the change, and possibly used both its original result and the one changed by the user. There should be an option for displaying both of them, and if they differ by too much, the user should be alerted to it. The kind of interaction suggested here is very similar in spirit to level 5 fusion for sensor management [17].

Figure 2 shows an example of how information could be shown to a user. Several groups of rioters (red objects) with different capabilities have been seen. A threat analysis module has suggested that the hospital (green star) is the most likely goal. The commander of our forces has access to a company in the south of the city as well as a small group of MP's that are positioned between the rioters and their goal. However, the health status of the MP's is not good (as shown by the ligher color on the map): they are too excited to be able to reason rationally with the rioters and convince them to stop. By combining this kind of information with planning tools that take into account the capabilities of both own and opposing groups, it should be possible to suggest good courses of action to the commander.

VI. Simulations

In this section we discuss how to determine capabilities from observations for two different types of scenarios, and present results from a computer simulation of the method.

A. Traditional war

In traditional war scenarios we do not get capabilities directly from the sensors. Instead, we are given mass functions of object types. Using force aggregation based on doctrines and force templates, we can also get platoons, companies, and battalions. By providing a translation table listing the capabilities of various recognized enemy units, we can get instead a list of capabilities for each recognized force unit. Even though doctrinal-based approaches are clearly best when the enemy follows a doctrine, complementing that system with one based on task-forces and capabilities could be beneficial in some cases. For example, if we manage to destroy several of the enemy vehicles, they will be forced to re-organize in an *ad hoc* manner, leading to capabilities-based task-forces. Using the methods presented here could also enable us to discover situations where the enemy tries to surprise us.

B. Operations other than war

The example scenario used in the computer experiments presented below is imagined to take place in international operations such as peace-keeping or enforcing. In such situations, the other actors do not necessarily have a clear doctrine or even clear goals for their actions. We will assume that the sensor system looks at an area where there are a lot of actors. There could be both organized crowds and spontaneously formed mobs. There could also be present regular or semi-regular troops, such as a guerilla or militia force. As mentioned above, in some cases it might be difficult to determine

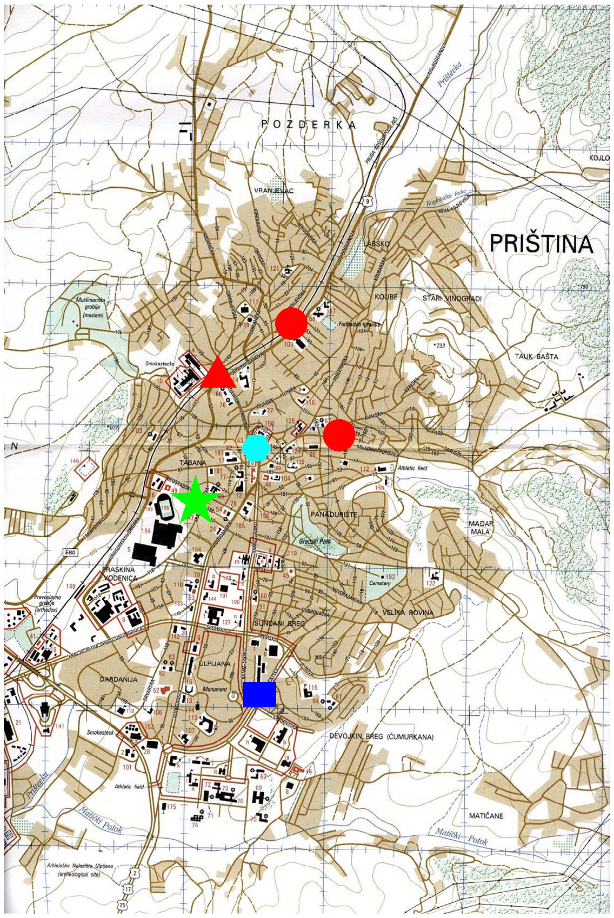

Fig. 2. This figure shows an example of how the capabilities-oriented aggregation of opposing forces (red objects) could be combined with health-monitoring of own forces (blue and cyan objects). In the picture, three groups of rioters are converging on a hospital (green star). We have a company of rested soldiers in the southern part of the city that need to come to the aid of the small group of fatigued soldiers in the middle of the map. Two of the groups of rioters have been determined to consist of people armed with side-arms and stones to throw (red circles), while the third group also has access to machine-guns (red triangle). Note that the symbols used are examples only: a real system would have to take usability and man-machine interaction aspects into account.

exactly to which faction a specific entity belongs, making it necessary to include its capabilities in several different task-forces belonging to different groups of actors. For example, an unmarked truck might belong to task-force Z_1 with probability p_1 and task-force Z_2 with probability p_2. Then the capability represented by that truck should be represented in both random sets corresponding to the different task-forces. Explicitly, this would mean that the one-point cover function of Z_1 and Z_2 would have the values p_1 and p_2 for that capability.

Capabilities can be generated in several different ways.

- As in the scenario with traditional warfare, sensor reports could be one source. This would necessitate translating from object or unit type to capabilities. Such translations would require having an ontology that is appropriate to the situation and that links types to capabilities. An ontology for this would also allow the translation to depend on external factors, such as weather.
- Capabilities could also be determined directly from observations. It can be argued that most intelligence data in OOTW arise from human observation. Such HUMINT reports could come either from operators immersed in the area of interest, or from analysts viewing surveillance cameras or UAV video sensors. Text mining could be used to extract relevant information on observed capabilities from such reports.
- Processing sensor output might also give rise to reports on capabilities. This could be the case, for example, if an algorithm that is capable of recognizing guns runs over the output of a video sensor.

Some examples of capabilities that might be of interest are shown in figure 3.

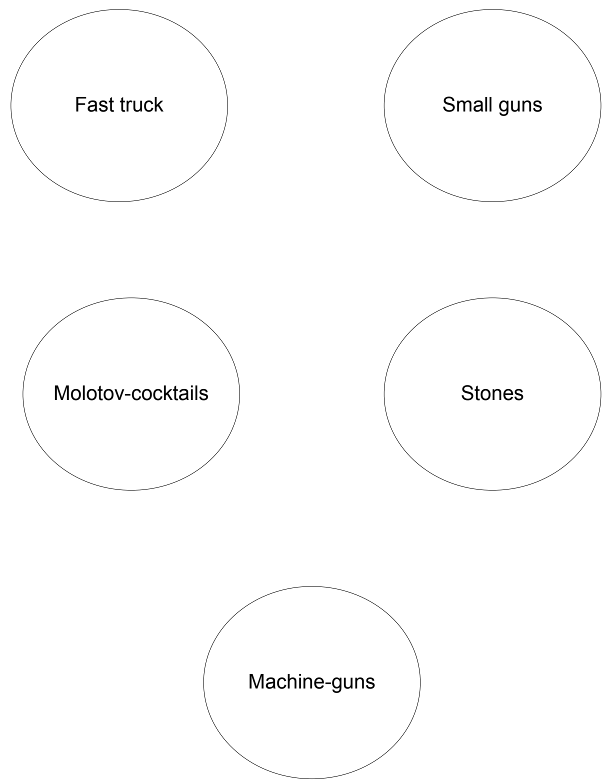

Fig. 3. Some possible capabilities that a group of actors in an OOTW scenario might possess.

C. Experiments

Here we present results from a computer simulation using the rules in equations 4, 6, and 7. For this simulation, we used a set of 8 possible capabilities, giving a total of 255 possible non-empty subsets of capabilities. 4 task-forces were generated, making sure that they were reasonably different. Each task-force consists of a number of focal sets and associated probabilities, and are assumed to represent either known structures or the output of a planning system. For equation 7, we used a similarity function S that was loosely based on cardinality of the symmetric difference between its two arguments.

We randomly generated $N = 100$ samples of observations. Each observation was generated by taking a focal set of one of the task-force structures and assigning mass to both it and several mutations of it. These mutations correspond to errors in sensor reports. The error probability of the sensors was varied between between 0 and 1.

As can be seen in figure 4, the fuzzy similarity measure (equation (7)) gives best results for small error probabilities. For larger error probabilities, the subset measure (equation (6)) gives the best results. Note, however, that the fraction of cases where the subset measure was unable to determine a best fit (shown in figure 5) increases considerably for these cases.

For realistic sensors, given that the errors in determining correct capabilities are smaller than those corresponding to about 0.1 error probability in the simple model given here, we can conclude that the fuzzy measure seems to be the most reliable. In a real system, the user should be presented with all of the measures. It would be interesting to see if some combination of the measures presented here would give better results.

VII. Summary and Future Work

In conclusion, we presented a formalism for determining appropriate labels or classifications of task-forces based on what capabilities they were determined to possess. The kind of force aggregation method presented here is useful in situations where we do not have detailed doctrines or other domain knowledge on enemy organization. Examples of such situations are military operations other than war, such as peacekeeping or peace-enforcing operations.

In such situations, we might face several different opponents who are not organized in standard military ways using platoons, companies, and battalions. Instead, they will form *ad hoc* task-forces that are organized to solve a specific task. By connecting the method presented here with improved task-recognition and planning methods, it might be possible to reach a higher level of situation awareness in such operations.

As shown in section VI, the measure based on fuzzy similarity (equation (7)) gave very good results for small observation error probabilities. For larger error probabilities, the measure based on subsets (equation (6)) seemed better.

There are many opportunities for future work in this area. First, the simulation presented here is purposefully very simplistic. The measures presented here need to be tested in a more realistic scenario, with a more realistic way of generating observations. It would be very interesting to see if data collected in missions overseas could be used to test the method. Second, work needs to be done on how to present the aggregations formed here to a user. In particular, how should a user interact with the system, both in order to input fragmentary templates or to request additional information on a task-force that is deemed interesting. Should just one of the classifications be presented to the user or several of them?

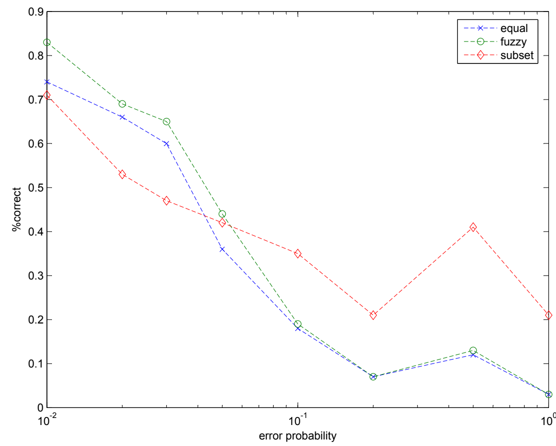

Fig. 4. This figure shows the results from a computer simulation using simulated sensors to determine capabilities and a list of four manually constructed task-force templates. The fraction of correct classification using either equality (equation (4)), subset (equation (6)) or the fuzzy similarity measure (equation (7)) as a function of the error probability of the simulated sensors are shown.

Other possible future improvements include using boosting to get better classification, or designing better measures of similarity. In particular, it would be very interesting to see if methods inspired by conditional event algebra (*e.g.*, [12]) could be used.

Acknowledgement: I thank Christian Mårtenson for commenting on a draft version of this manuscript and Johan Schubert for valuable discussions on force aggregation.

References

[1] D. L. Hall and J. Llinas, editors. *Handbook of Multisensor Data Fusion*. CRC Press, Boca Raton, FL, USA, 2001.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[3] Johan Schubert. Clustering belief functions based on attracting and conflicting metalevel evidence using Potts spin mean field theory. *Information Fusion*, 5(4):309, 2004.

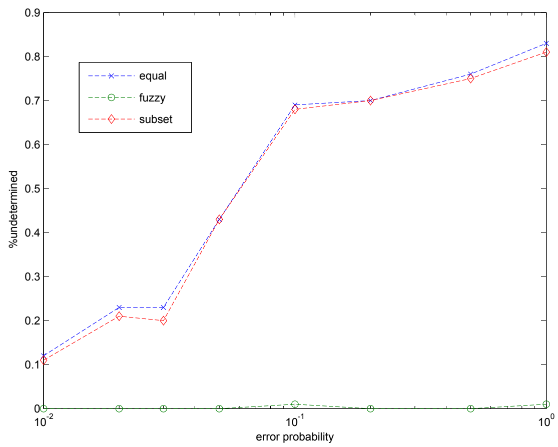

Fig. 5. This figure shows the fraction of cases where the methods could not determine a best fit as a function of the error probability of the simulated sensors.

[4] J. Schubert. Evidential force aggregation. In *Proceedings of the Sixth International Conference on Information Fusion, Cairns, Australia*, pages 1223–1229, Sunnyvale, CA, USA, 2003. International Society of Information Fusion.

[5] F. P. Lorenz and J. Biermann. Knowledge-based fusion of formets: discussion of an example. In *Proceedings of the Fifth International Conference on Information Fusion, Annapolis, MD, USA*, pages 374–379, Mountain View, CA, USA, 2002. International Society of Information Fusion.

[6] J. K. Johnson and R. D. Chaney. Recursive composition inference for force aggregation. In *Proceedings of the Second International Conference on Information Fusion, Sunnyvale, CA, USA*, pages 1187–1195, Mountain View, CA, USA, 1999. International Society of Information Fusion.

[7] J. Biermann. HADES – A knowledge-based system for message interpretation and situation determination. In A. Pasqual del Pobil, J. Mira, and M. Ali, editors, *Tasks and methods in Applied Artificial Intelligence, Proceedings of the Eleventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, LNCS 1416, pages 707–716, Berlin, Germany, 1998. Springer-Verlag.

[8] R. Suzić. Representation and recognition of uncertain enemy policies using statistical models. In *Proceedings of the NATO/RTO Conference on Military Data and Information Fusion, Prague, Czech Republic,* in press, 2003.

[9] Ed Wright, Suzanne Mahoney, and Kathryn Blackmond Laskey. Use of domain knowledge models to recognize cooperative force activities. In *Proceedings of the 2001 MSS National Symposium on Sensor and Data Fusion*, 2001.

[10] Kathryn Blackmond Laskey, Suzanne Mahoney, and Ed Wright. Hypothesis management in situation-specific network construction. In *Seventeenth Conference on Uncertainty in Artificial Intelligence*, 2001.

[11] Robert Suzić. Generic representation of military organisation and military behaviour: UML and Bayesian networks. In *Proceedings of the NATO RTO Symposium on C3I and M&S Interoperability*, October 2003.

[12] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1997.

[13] R. Mahler. *An Introduction to Multisource-Multitarget Statistics and its Applications*. Lockheed Martin Technical Monograph, 2000.

[14] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, USA, 1976.

[15] Ronald P. S. Mahler. Combining ambiguous evidence with respect to ambiguous a priori knowledge, i: Boolean logic. *IEEE Transactions on Systems, Man and Cybernetics*, A26(1):27, 1996.

[16] Dale Fixsen and Ronald P. S. Mahler. The modified dempster-shafer approach to classification. *IEEE Transactions on Systems, Man and Cybernetics*, 27(1):96, 1997.

[17] Erik P. Blasch and Susan Plano. JDL level 5 fusion model ”user refinement” issues and applications in group tracking. In *Signal Processing, Sensor Fusion, and Target Recognition X!*, volume 4729 of *SPIE*, page 270, 2002.