2004 IEEE International Conference on Multimedia and Expo (ICME)

Optimal Decision Fusion With Applications to Target Detection in Wireless Ad Hoc Sensor Networks

Marco F. Duarte *IEEE Student Member*, and Yu Hen Hu *IEEE Fellow*

Abstract—**Decision fusion is a decentralized decision making process where local decisions are combined to reach a global decision. In this work, we propose a complementary optimal decision fusion (CODF) method to the target detection task that arises in wireless ad hoc sensor network signal processing. We conduct extensive comparative study using real world sensor signal data, and observe superior performance of CODF when compared with state-of-the-art decision fusion methods. In addition to distributed sensor network applications, the proposed CODF algorithm can be applied to numerous multi-modality, multi-agent, multi-media signal processing problems.**

I. Introduction

Decision fusion data fusion method that has found applications in multi-modal multimedia signal processing [13], [3], [6], decentralized detection [1], collaborative sensor network signal processing [5], and the like. With decision fusion, individual component decision makers (pattern classifiers) report their own local decisions (classification results) to a common fusion center where a final consensus decision will be made. In doing so, only the local decisions, rather than the raw data, need to be transmitted to the fusion center. If a local decision can be represented by an integer $\{n; 1 \leq n \leq N\}$, then it can be encoded using $\log_2 N$ bits. Thus, transmitting a decision to the fusion center, rather than the raw data sample, often represents a significant saving in communication bandwidth. For applications where communication cost is high, such as a wireless sensor network, decision fusion is advantageous.

Previously, in [1], optimal data fusion is presented under the constraint of a fixed k out of n weighted threshold fusion architecture. In [2], a hierarchical model is used and Bayesian Gibbs sampling method is used to design the fusion rule. Data fusion has also been studied in the context of combining multiple classifiers. In [14], three types of classifier combination methods, namely, *averaged Bayes classifiers*, *voting principals*, and *Dempster-Shafer fuzzy combinations* have been reviewed. Some experiments have been conducted but no conclusive comparison results are available. In [12], the accuracy of individual classifiers are estimated, and classifiers are selected dynamically based on which classifier will yield best performance in specific local region. Ji and Ma [9] proposed to use a structure consisting of randomly generated linear local classifiers with a voting fusion mechanism to perform pattern classification tasks. Petrakos et al. [11] discussed the effect of correlations between classifiers and their impacts on fusion performance. In [8], an optimal behavior knowledge space (BKS) method has been proposed that directly combined classifier output vectors into different classes. It has been argued that this BKS method produces optimal decision fusion results but no further development has been found in the literature.

Recently [4], we developed an optimal decision fusion (ODF) method. Assuming that the set of local decision makers are fixed, and that the number of training samples are sufficiently large, we show that a look-up-table based ODF method is capable of producing decision fusion results that are no worse than any other decision fusion algorithms. In effect, we derived a tight theoretical performance upper bound of any decision fusion algorithm. The ODF method is basically the same as the BKS method proposed in [8]. However, when there are only finite number of training data samples, either LUT or BKS method may exhibit inferior performance.

In this paper, we focus on the analysis and enhancement of the ODF method with an application to target detection in a wireless ad hoc sensor network environment. Specifically, we developed a complementary ODF (CODF) method that uses ODF in conjunction with a non-ODF decision fusion method to improve the overall performance while conserving storage space. For most of feature vectors that the non-ODF method yields correct results, the ODF method will stay silent. When the ODF method issues a opinion, it will then overwrite the opinion given by the non-ODF decision maker. These two decision fusion methods collaborate to complement each other, and hence the name complementary ODF. We further analyze the performance of the CODF method and discussed the potential impacts on its performance when the component decision making units give correlated local decisions.

This paper is organized as follows: Section II presents the theoretical framework for this problem. Section III specifies our approach for Complementary Optimal Decision Fusion. Section IV contains the experiments we performed with our method and their results. Finally, Section V presents some conclusions to the paper.

II. Problem Formulation

A. Decision Fusion Framework

We assume a decision fusion architecture that consists of a *fusion center* and K distributed sensors as local decision makers. The k^{th} sensor observes a feature vector $\mathbf{x}_k$. According

The authors are with the University of Wisconsin - Madison, Electrical and Computer Engineering, Madison WI 53706, USA. This project is supported by DARPA under grant no. F 30602-00-2-0555

to a local *decision rule*, $\mathbf{x}_k$ will be assigned to a class label among a set of N possible labels $\mathbf{C} = \{C_1, C_2, \cdots, C_N\}$. That is,

$$\ell_k(\mathbf{x}_k) = d_k \in \mathbf{C}$$

d_k is called a *decision*. We use the set membership notation $\mathbf{x}_k \in C_n$ to denote that $d_k = C_n$.

A global feature vector $\mathbf{x}$ is the concatenation of all local feature vectors. That is,

$$\mathbf{x} = \left[\ \mathbf{x}_1^T \quad \mathbf{x}_2^T \quad \cdots \quad \mathbf{x}_K^T\ \right]^T$$

The k^{th} sensor will evaluate $\mathbf{x}_k$ and make a local decision $d_k \in \mathbf{C}$. In other applications, it is possible that all sensors receive the same feature vector, that is, $\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_K$. In such a case, we simply use $\mathbf{x} = \mathbf{x}_1$ without concatenation. The feature space is the space spanned by the global feature vector. For each feature vector $\mathbf{x}$, a decision rule maps it into a particular class label. Equivalently, the decision rule *partitions* the feature space N into disjointed regions. Feature vectors within each region are assigned to the same label.

For decision fusion, each of the K sensors will forward its local decision d_k to a common fusion center, where a decision fusion algorithm will compute a final decision $\ell(\mathbf{d})$ based **only** on a decision vector that consists of the set of local decisions:

$$\mathbf{d} = \left[\ d_1 \quad d_2 \quad \cdots \quad d_K\ \right]$$

The assumption that the fusion center does not have the global feature vector $\mathbf{x}$ to make a decision is important, and appropriate for wireless communication channels where the communication cost is very high.

III. Complementary Optimal Decision Fusion (CODF)

A. Optimal Decision Fusion

The decision fusion is based on the $K \times 1$ feature vector $\mathbf{d}$. There will be at most N^K different decision vectors. Feature vectors $\mathbf{x}$ mapped to the same decision vector will be assigned to the same class label. As such the N^K different decision vectors will partition the feature space into N^K disjoint regions. Moreover, each of these regions will be assigned to a specific class label by a decision fusion algorithm. As such, if the probability of correct decision assignment is maximized for each individual decision vector, the resulting decision fusion method will be optimal in the sense that it maximize the probability of making correct decisions given only the decision vectors. Therefore, the optimal decision fusion (ODF) amounts to a look-up table (LUT): In each entry of this table is a different decision vector and its corresponding decision assignment. In [4], we have shown that such an ODF decision fusion scheme does not necessarily reach the performance of a Bayesian decision. ODF is optimal in that it maximizes the probability of correct decision under the constraint that only the decision vector $\mathbf{d}$ is used for the purpose of decision fusion.

B. Complementary ODF (CODF)

This LUT-based ODF method has the same formulation as the BKS method proposed by Huang and Suen [8] in 1993. The difference is that the ODF method uses a training set that has finite number of feature vectors. The impact of finite number training data is two-fold: First, there may be fewer training samples than the number of different decision vectors $\mathbf{d}$, N^K. Second, there may be too few training samples falling within each of the r_m regions. In either of these two cases, there is no sufficient information to infer the proper class label assignment to that corresponding decision vector $\mathbf{d}$. Furthermore, depending on the specific structure of individual component decision makers, it is possible that certain decision vectors will not occur regardless how many training samples are available. Another potential drawback of the ODF method is that even when there is a sufficient amount of training data, it is possible that there are too many different entries in the ODF table. As such, the storage cost of such a decision fusion classifier will be very high.

To alleviate this problem, we propose a hybrid approach. We will use a simple and effective non-ODF decision fusion method to team up with the ODF method. For decision vectors that the ODF method fail to yield reliable decisions due to lack of sufficient training samples, we resort to these complementary decision fusion methods. For decision vectors that both ODF and these methods yield identically correct results, we choose either ODF or these methods based on trade-offs between storage cost versus computation cost. For decision vectors that ODF yield correct results while these non-ODF decision fusion classifiers yield incorrect result, we use ODF. As such, the ODF table can be significantly reduced and the overall performance can be improved. This non-ODF decision fusion method is used to *complement* the ODF performance, and hence this hybrid method will be called complementary ODF (CODF) method.

C. Complementary Decision Fusion Methods

We list several choices of decision fusion methods that fit into the description of being simple and effective:

Non-Weighted Threshold voting: For our two-class problem, the simples method is to perform non-weighted voting:

$$\sum_{i=1}^{K} w_i d_i(\mathbf{x}) \underset{\ell=C_2}{\overset{\ell=C_1}{\gtrless}} t \tag{1}$$

where $d_i(\mathbf{x}) = 1$ if $\mathbf{x} \in C_1$ and $d_i(\mathbf{x}) = 0$ if $\mathbf{x} \in C_2$. For such a voting scheme fusing K nodes, there will be $K - 1$ distinct threshold possibilities, since the result of such voting scheme will yield an integer number result for each available class. Therefore, an optimal threshold can be calculated that will minimize the error:

$$t = \arg \min_{0 \le k \le K-1} e(k + 1/2) \tag{2}$$

This method is the easiest to implement, but will yield low performance, since all nodes are being weighted equally.

Weighted Least Square Thresholding: The weighted least square thresholding assumes that the two classes observed will be assigned labels as follows: $d_i(\mathbf{x}) = 1$ if $\mathbf{x} \in C_1$ and $d_i(\mathbf{x}) = -1$ if $\mathbf{x} \in C_2$. It is based on a linear least-squares filter [7], where each decision $d_i(\mathbf{x})$ will be weighted by a value w_i, and our observation o for the training set will be $o = \mathbf{D}\mathbf{w}$, where $\mathbf{D} = [\mathbf{d}(\mathbf{x}_1), \mathbf{d}(\mathbf{x}_2), ...\mathbf{d}(\mathbf{x}_m)]^T$ is the matrix of the decision vectors for the m training samples, $\mathbf{d}(\mathbf{x}) = [d_1)(\mathbf{x}), d_2(\mathbf{x}), ...d_K(\mathbf{x})]$ is the decision vector for sample x, and $\mathbf{w} = [w_1, w_2, ...w_K]^T$ is the weight vector. The solution to this filter is expressed as

$$\mathbf{w} = \mathbf{D}^{+}\mathbf{l} \tag{3}$$

where $\mathbf{l}$ is the label vector for the training samples, $\mathbf{D}^{+}$ is the pseudoinverse of $\mathbf{D}$, defined as $\mathbf{X}^{+} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$, and the error is defined as

$$\mathbf{e} = \mathbf{l} - \mathbf{D}\mathbf{W} \tag{4}$$

Optimal Linear Threshold: The optimal linear threshold method uses the method of steepest descent [7] to obtain a progressively accurate estimate of an optimal weighting vector:

$$\mathbf{w}(a+1) = \mathbf{w}(a) - \eta\mathbf{g}(a) \tag{5}$$

where a is the order of the iteration, η is the learning-rate parameter, $\mathbf{w}(a)$ is the weight vector for the current iteration, and $\mathbf{g}(a)$ is the gradient vector of the error $\mathbf{e}(a)$, as defined previously, with respect to $\mathbf{w}(a)$:

$$\mathbf{g}(a) = \left[\frac{\partial \mathbf{e}(a)}{\partial w_1(a)}, \frac{\partial \mathbf{e}(a)}{\partial w_2(a)}, ..., \frac{\partial \mathbf{e}(a)}{\partial w_K(a)}\right]^T \tag{6}$$

For this method, we need to set an initial set of weights and a convergence criterion. For our case, we have chosen a random initial set of weights and used both minimum convergence error and maximum number of iteration criterions for convergence.

Local Classifier Accuracy Weighting: This method intuitively will assign weights to the different decisions proportional to their accuracy level; i.e., a classifier that is more likely to be correct will be assigned a larger weight. The weights are then normalized:

$$w_i = \frac{r_i}{\sum_{i=1}^{K} r_i} \tag{7}$$

where r_i is the classification rate for classifier i [5]. This method will yield acceptable results if the accuracy of the classifiers remains constant among different sets of samples.

Following the Leader: This heuristic method will assign as decision result the label assigned by the classifier most likely to be correct:

$$w_i = \begin{cases} 1 & \text{if } i = \arg\max_{1 \le i \le K} r_i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Thus its performance will depend on whether the behavior of the classifiers remains constant among different sets of samples.

IV. Experiments

We run experiments on data collected from a sensor network deployed in an outdoor sensor field. Each sensor node consists of an on-board computer, power source (battery), one or more sensors with different modalities, and wireless transceivers. This sensor node features acoustic sensing using a microphone, creating a signal that is sampled at 5 kHz at 12 bit resolution. The on-board computer is a 32-bit RISC processor running the Linux operating system. The sensor field is an area of approximately 900×300 meters in a California Marine training ground. The sensors are laid out along side the road. The separation of adjacent sensors ranges from 20-40 meters. We group the sensors into a single *region*. Sensors within each region will be able to communicate freely. One sensor within each region is designated as a *manager node*. The manager node will be given the authority to communicate with manager nodes of surrounding regions. This hierarchy of communication ensures that only local wireless traffic will be engaged, and hence contributes to the goal of energy conservation. Military vehicles are driving passing through the roads. The objective is to detect the vehicles when they pass through each region. During the experimentation in November 2001, multi-gigabyte data samples [5] have been recorded and are used in this paper. We will call these data Sitex02 data set.

For each of the 0.75 second duration, the energy of the acoustic signal will be computed. This single energy reading then will be fed into a *constant false alarm rate* (CFAR) energy detector [10] to determine whether the current energy reading has a magnitude that exceeds a computed threshold. If so, a *node-detection* event will be declared for this duration. Otherwise, the energy reading is considered as contributions from the background noise. The optimal decision fusion can be applied to the detection problem, since this task requires a single decision to be derived from the data gathered by all the nodes contained in the region. We compare the results of our proposed method with the decision fusion rules mentioned in Section III, and we use these rules as complementary rules for our CODF method.

We combined the AAV and DW data and partition the data set into a training data set by the 20 experiment runs performed, and we select one run for testing while grouping data from all other runs for training; this is commonly known as Leave One Out testing. Then we use the proposed CODF method and other methods to develop a fusion rule and then use the rule to fuse the detection decision set. The classification rate of both the CODF and the alternative methods are recorded. The training and testing error rates of the different available methods are depicted in figure 1 and 2. We tested the alternate methods plus the CODF method using each of the proposed complementary rules or ODF alone. The effect of using a backup rule in the size of the CODF training table is shown in Figure 3.

From the plot it can be seen that the Optimal Decision Fusion gives the lowest training error among the different methods, and that its performance varies along different runs and backup methods. This can be explained by the fact that for most of the runs, some of the ensemble results from the

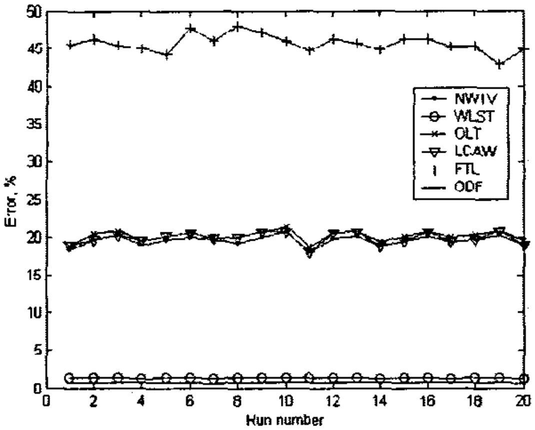

Fig. 1. Training detection error rates for several decision fusion rules and different runs

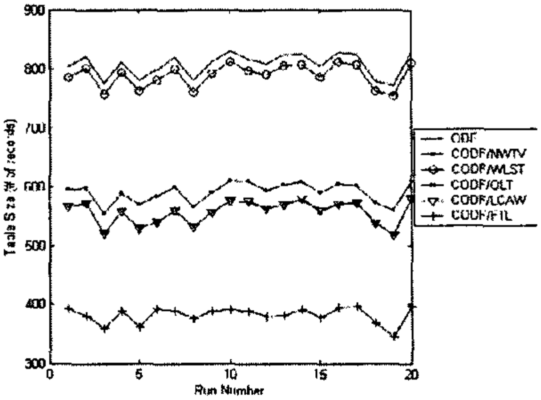

Fig. 3. Effect of backup rule implementation on the size of the Optimal Decision Fusion Training Table for different runs

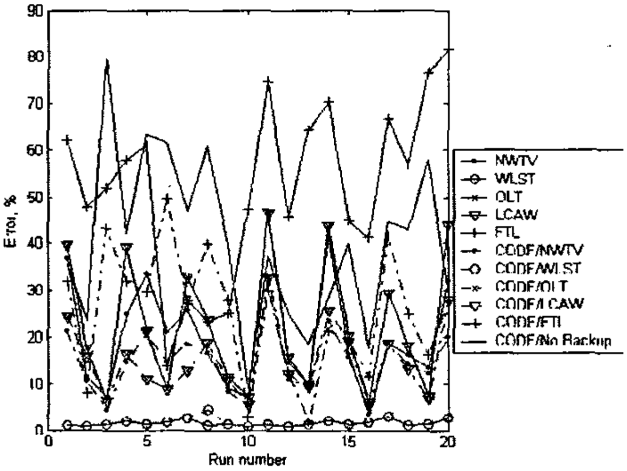

Fig. 2. Testing detection error rates for several decision fusion rules and different runs

different classifiers did not appear in other runs and therefore were not considered during the training; in these cases the performance of the CODF rule will be contingent on the performance of the backup rule. As seen in figure 2, for almost all cases the testing performance of the CODF with a complementary rule is better than that of the rule without any enhancement. We also show that for all methods, the size of the CODF training table is reduced by removing those records whose result match that of the backup rule; the number of the records removed is proportional to the accuracy of the backup decision rules.

V. Conclusions and Further Work

In this paper we have shown that using a simple algorithm, the optimal decision fusion method can be defined even without explicitly knowing the classification rates of the different sensors, or for dependent sensors in a region. We also have shown experimental results that holds our argument. It is important, however, to have enough features available during training so that the statistics for each one of the different regions based on the decision vector results are statistically defined. In the future, we will report results of application of this method in other real-world problems, such as collaborative signal processing and handwritten character recognition. We will also report on any patterns observed for sensor network applications.

References

[1] Chair, Z., Varshney, P.K. Optimal Data Fusion in Multiple Sensor Detection Systems. IEEE Transactions on Aerospace and Electronic Systems **22**(1) (1986) 98–101

[2] Chen, B., Varshney, P.K. A Bayesian Sampling Approach to Decision Fusion Using Hierarchical Models. IEEE Transactions on Signal Processing **50** (1992) 1809–1818

[3] Chen, T., Graf, H.P., Haskell, B., Petajan, E., Wang, Y., Chen, H., Chou, W.: Speech-assisted lip synchronization in audio-visual communications. Proceedings of the Internatioaln Conference on Image Processing **2** (1995) 579–582

[4] Duarte, M., Hu, Y.H.: Optimal Decision Fusion in a Distributed Wireless Sensor Network. Submitted to the 3rd International Processing Sensor Networks Conference IPSN'04. Berkeley, CA (2004)

[5] Duarte, M., Hu, Y.H.: Vehicle Classification in Distributed Sensor Networks. To be published in Journal of Parallel and Distributed Computing (2004)

[6] Girin, L., Foucher, E., Feng, G.: An audio-visual distance for audio-visual speech vector quantization. IEEE Second Workshop on Multimedia Signal Processing (1998) 523–528

[7] Haykin, S. Neural Networks: A Comprehensive Foundation. Upper Saddle River, NJ: Prentice Hall 2 ed. (1999) 117–133

[8] Huang, Y.S., Suen, C.Y.: A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals. IEEE Trans. Pattern Analysis and Machine Intelligence **17** (1995) 90–94

[9] Ji, C., Ma, S.: Combinations of weak classifiers. IEEE Trans. Neural Networks **8** (1997) 32–42

[10] Li, D., Wong, K.D., Hu, Y.H., Sayeed, A.M.: Detection, classification and tracking of targets. IEEE Signal Processing Magazine **19** (2002) 17–29

[11] Petrakos, M., Benediktsson, J. A., Kanellopoulos, I.: The effect of classifier agreement on the accuracy of the combined classifier in decision level fusion. IEEE Trans. on Geoscience and Remote Sensing **39** (2001) 2539–2546

[12] Woods, K., Kegelmeyer Jr., W. P., Bowyer, K.: Combination of multiple classifier using local accuracy estimates. IEEE Trans. Pattern Analysis and Machine Intelligence **19** (1997) 405–410

[13] Wu, L., Oviatt, S.L., Cohen, P.R.: Multimodal integration - A statistical view. IEEE Trans. Multimedia **1** (1999) 334–341

[14] Xu, L., Krzyzak, A., Suen, C. Y.: Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. Systems, Man, and Cybernetics **22** (1992) 418–435