2004 IEEE 6th Workshop on Multimedia Signal Processing

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 standard datasets, 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 [1], [2], [3], decentralized detection [4], 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.

Recently [6], 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 [7]. 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 shows some experiments to evaluate the performance of the proposed methods. Finally, Section V presents some conclusions to the paper.

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

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 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,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_K^T \end{bmatrix}^T$$

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

0-7803-8578-0/04/$20.00 ©2004 IEEE

fusion algorithm will compute a final decision $\ell(\mathbf{d})$ based **only** on a decision vector that consists of the set of local decisions:

$$\mathbf{d} = [\begin{array}{cccc} d_1 & d_2 & \cdots & d_K \end{array}]$$

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 [6], we have shown that this method is optimal on the training set, and 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 [7] 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. Proposed Complementary Methods

Some rules are proposed below for use as complementary methods in the CODF algorithm. These rules may be adapted for problems with more than two classes identified.

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

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

where $d_i(x) = 1$ if $x_i \in C_1$ and $d_i(x) = 0$ if $x_i \in C_2$ and $w_i = 1$ for non-weighted voting. 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:

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

Here, $e(k)$ is the number of errors that occur when threshold k is selected. This method is the easiest to implement, but may yield low performance, since all nodes are being weighted equally.

2) Weighted Least Square Thresholding: The weighted least square thresholding assumes that each of the samples observed will be assigned labels by classifier i as follows: $d_i(x) = 1$ if $x \in C_1$ and $d_i(x) = -1$ if $x \in C_2$. It is based on a linear least-squares filter [8], where each decision $d_i(x)$ will be weighted by a value w_i, and our observation $\mathbf{o}$ for a set of samples $\mathbf{x} = x_0, x_1, \ldots, x_m$ will be:

$$\mathbf{o}(\mathbf{x}) = \mathbf{D}\mathbf{w} \tag{3}$$

where $\mathbf{D} = [\mathbf{d}^T(x_0), \mathbf{d}^T(x_1), \mathbf{d}^T(x_2), \ldots \mathbf{d}^T(x_m)]^T$ is the matrix of the decision vectors for the m training samples, $\mathbf{d}(x) = [d_1(x), d_2(x), \ldots d_K(x)]$ is the decision vector for sample x and $\mathbf{w} = [w_0, w_1, \ldots w_K]^T$ is the weight vector, where w_i is the weight for classifier i. The solution to this filter is expressed as

$$\mathbf{w} = \mathbf{D}^{+}\mathbf{l} \tag{4}$$

where $\mathbf{l}$ is the label vector for the training samples, $\mathbf{l} = [l_1, l_2, \ldots, l_m]^T$. For each sample, $l_i = 1$ if $x_m \in C_1$ and

$l_i = -1$ if $x_m \in C_2$. Also, $\mathbf{D}^+$ is the pseudoinverse of $\mathbf{D}$, defined as

$$\mathbf{X}^+ = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{5}$$

In this case, the error is defined as

$$\mathbf{e} = \mathbf{l} - \mathbf{D}\mathbf{w} \tag{6}$$

The sample x will be assigned C_1 as the label if the observation is positive, and C_2 if the observation is negative.

3) Optimal Linear Threshold: The optimal linear threshold method uses the method of steepest descent [8] to obtain a progressively accurate estimate of an optimal weighting vector:

$$\mathbf{w}(a+1) = \mathbf{w}(a) - \eta\mathbf{g}(a) \tag{7}$$

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 $e(a)$, defined as

$$\mathbf{g}(a) = \left[\frac{\partial e(a)}{\partial w_1(a)}, \frac{\partial e(a)}{\partial w_2(a)}, ..., \frac{\partial e(a)}{\partial w_n(a)}\right]^T \tag{8}$$

where the error is defined as

$$e(a) = \mathbf{1}_m^T(\mathbf{l} - \mathbf{D}\mathbf{w}(a)) \tag{9}$$

with respect to $\mathbf{w}(a)$. In equation 9, $\mathbf{1}_m = [1, 1, ..., 1]^T$ is an $m \times 1$ vector, which effectively sums the error in the output label for all training samples. This method uses the same label-assigning criterion as the previous one.

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.

4) 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:

$$w_i = \frac{r_i}{\sum_{i=1}^{n} r_i} \tag{10}$$

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

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

$$w_i = \begin{cases} 1 & \text{if } i = \arg\max_{1 \le i \le n} r_i \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

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

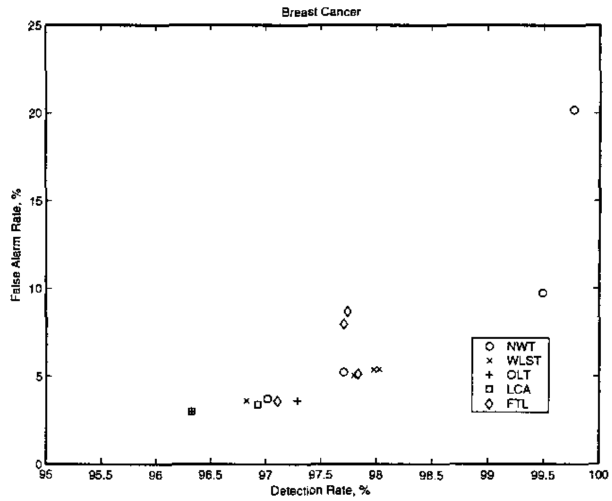

Fig. 1. Complementary method performance for the Breast Cancer Dataset - Detection metrics

IV. Experiments

To select the best suited complementary method for the CODF scheme, we test the different possibilities using some standard datasets from a repository from the University of California at Irvine [9]. We chose some sets that feature large training feature dimension, large number of samples and well-defined features for all samples. 10% of the samples in each set are used to train Maximum Likelihood classifiers on disjoint subsets of the feature vector's dimensions, leaving the remaining 90% for use in the fusion method training and testing. We used the proposed complementary methods to develop a fusion rule to fuse the decision set. The size of the features, the number of classifiers and the size of the training samples are shown on table I. We use three-way cross validation to evaluate the performance of each complementary method for each of the available datasets. Each test is executed 10 times and the results for the occurrences averaged for these datasets as well. The classification rate of each method is recorded.

For the two class problems, the performance is measured using the detection and false alarm probabilities for the global detector, as well as the overall classification rate when the problem is treated as a classification problem. The results for the different datasets and minimum number of samples are shown in figures 1 to 3. These figures compare the detection and false alarm rates for the different complementary methods on each dataset.

As can be seen from these plots, using the Weighted Least Square Voting method as complementary method generally yields the best performance - the closest performance to the 100-0 corner in most cases - and is the least dependent on the minimum number of samples established in the BKS table. Thus, we choose this complementary method for our proposed CODF fusion scheme.

TABLE I

DESCRIPTION OF FEATURE SELECTION AND CLASSIFIER TRAINING FOR DATASETS FROM UC-IRVINE'S REPOSITORY

Dataset Name	Feature Dimension	Number of classifiers	Feature Dimension per classifier	Number of samples
Breast Cancer - Wisconsin	9	3	3	683
Ionosphere	34	17	2	351
Tic-Tac-Toe	9	3	3	958

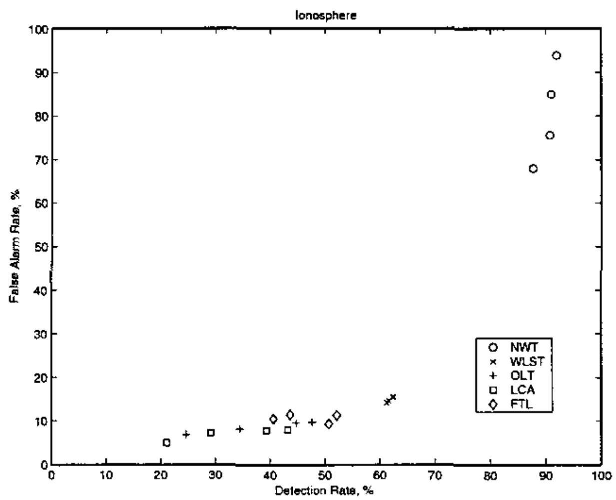

Fig. 2. Complementary method performance for the Ionosphere Dataset - Detection metrics

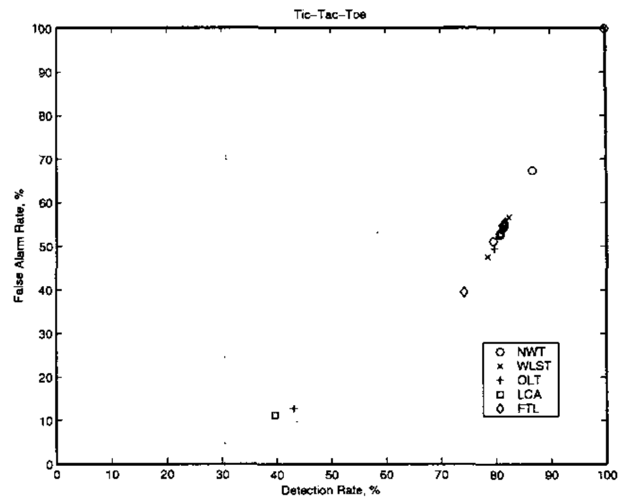

Fig. 3. Complementary method performance for the Tic-Tac-Toe Dataset - Detection metrics

V. CONCLUSIONS

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 hold 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. Experimental results show that the ODF method has the lowest training error, the CODF method performs better than simple ODF under most conditions, and the size of the CODF table is smaller than the size of the ODF table for all available complementary rules.

Due to the reduced size allowed for the paper, no results were presented regarding real-world sensor data. The reader is invited to visit our website, http://www.ece.wisc.edu/~sensit, to review our further work on applications of CODF on sensor networks.

REFERENCES

[1] H. Wu, M. Siegel, and P. Khosla, "Vehicle sound signature recognition by frequency vector principal component analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 48, no. 5, pp. 1005–1009, Oct. 1999.

[2] T. Chen, H. Graf, B. Haskell, E. Petajan, Y. Wang, H. Chen, and W. Chou, "Speech-assisted lip synchronization in audio-visual communications," in *Proceedings of the International Conference on Image Processing*, vol. 2. Washington, DC: IEEE, Oct. 1995, pp. 579–582.

[3] L. Girin, E. Foucher, and G. Feng, "An audio-visual distance for audio-visual speech vector quantization," in *Proceedings of the Second Workshop on Multimedia Signal Processing*. Grenoble, France: IEEE, Dec. 1998, pp. 523–528.

[4] Z. Chair and P. Varshney, "Optimal data fusion in multiple sensor detection systems," *Multidimensional Systems and Signal Processing*, vol. AES-22, no. 1, pp. 9–31, 1986.

[5] M. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing*, 2004, to be published.

[6] ——, "Optimal decision fusion with applications to target detection in wireless ad hoc sensor networks," in *Proceedings of the International Conference on Multimedia and Expo*. Taipei, Taiwan: IEEE, June 2004, to be published.

[7] Y. Huang and C. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90–94, Jan. 1995.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice Hall PTR, 1999.

[9] C. Blake and C. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html