PAPER Special Section on Emerging Technologies for Practical Ubiquitous and Sensor Networks

A GMM-Based Target Classification Scheme for a Node in Wireless Sensor Networks

Youngsoo KIM[†a], **Sangbae JEONG**[†], *Nonmembers*, *and* **Daeyoung KIM**[†], *Member*

SUMMARY In this paper, an efficient node-level target classification scheme in wireless sensor networks (WSNs) is proposed. It uses acoustic and seismic information, and its performance is verified by the classification accuracy of vehicles in a WSN. Because of the hard limitation in resources, parametric classifiers should be more preferable than non-parametric ones in WSN systems. As a parametric classifier, the Gaussian mixture model (GMM) algorithm not only shows good performances to classify targets in WSNs, but it also requires very few resources suitable to a sensor node. In addition, our sensor fusion method uses a decision tree, generated by the classification and regression tree (CART) algorithm, to improve the accuracy, so that the algorithm drives a considerable increase of the classification rate using less resources. Experimental results using a real dataset of WSN show that the proposed scheme shows a 94.10% classification rate and outperforms the k-nearest neighbors and the support vector machine.
key words: *target classification, sensor network, Gaussian mixture model (GMM), classification and regression tree (CART), decision tree*

1. Introduction

Target classification is one of the most important and demanding technologies used to achieve intelligent sensor network systems. Recently, various research on the performance improvement of classification have been conducted [1]–[5]. It is commonly known that the computation cost and performance of a classifier are in a trade-off relationship. However, hardware and software resources are tightly limited in wireless sensor networks (WSNs). Thus, the complexity of the classifier should be first considered.

In a WSN, acoustic, seismic, magnetic, and infrared sensors have been mainly used for target classification [5]. Among them, both acoustic and seismic sensors perform a central role for classifying targets in the WSN, while the others have assistant roles [1]–[4]. This is because the magnetic and infrared sensors should assume a strictly predefined path in a case of target classification, while the former do not have such a restriction. However, the signals of acoustic and seismic sensors are very complicated as well as hard to handle. Therefore, some technologies to classify objects using acoustic and seismic signals with low costs in WSNs are greatly required.

Most researches on the target classification in a WSN extract spectral features based on the fast Fourier transform (FFT) and classify objects with some type of frame-based classification algorithm such as the k-nearest neighbor (kNN) [11], the Maximum Likelihood (ML) [12], the support vector Machine (SVM) [13], etc. However, the computational complexity and required memory to run the classifiers have not yet been investigated. Also, the characteristics of sensed signals in a WSN have not been sufficiently studied.

This paper proposes the Gaussian Mixture Model (GMM) algorithm as a classifier [6], and a decision tree, estimated by the Classification & Regression Tree (CART) algorithm [10], for sensor fusion in a sensor node. To verify the feasibility of being applied to a node in a real sensor field, the complexities of the algorithms are analyzed. Since a parametric algorithm requires generally much less complexity than a nonparametric one, the GMM, as a parametric approach, is appropriate to WSN systems. In our experiments, The GMM also outperformed other algorithms, such as kNN and SVM. Also, the proposed fusion algorithm improved the classification rate better than the other sensor fusion methods, feature-level fusion and the Dempster-Shafer method [18], while it having a low complexity. The sampling rates, kinds of features, and feature dimensions were considered since they are closely related to the performance and the amount of resources. Our experiments showed that the best classification rate could be achieved at a 500 Hz sampling rate in the case of classifying some moving vehicles, and that the mel-frequency cepstral coefficients (MFCCs) [9] were superior to the spectral features [2] in classification rates. Moreover, the features comprising the 12th-order MFCC and its derivatives are the most excellent among several of their combined features. Consequently, we found that the proposed classification scheme is the optimal one for application to a node.

The structure of this paper is as follows. Section 2 contains a discussion of related works, and Sect. 3 describes our proposed classification scheme. In Sect. 4, our data collection and experimental setup are illustrated, and the performance of the proposed algorithm is experimentally compared and evaluated with those of other algorithms. Finally, Sect. 5 concludes our experiments and discusses future works.

2. Related Works

There have been many researches on the target classification in a WSN, and various algorithms and strategies have been proposed [1]–[5], [18]. To improve the accuracy with

Manuscript received March 1, 2008.
Manuscript revised May 30, 2008.
[†]The authors are with Information and Communications University, Daejeon, Republic of Korea.
a) E-mail: pineland@icu.ac.kr
DOI: 10.1093/ietcom/e91–b.11.3544

minimum costs, not only efficient algorithms but also decentralized processing and data fusion strategies have been recently proposed. Conspicuous researches on target classification that have used acoustic and seismic sensors thus far are as follows.

Meesookho, et al. proposed a collaborative signal processing and fusion algorithm in WSNs [1]. They used the normalized energies of 16 frequency bands of the acoustic and seismic signal spectra as feature parameters to classify three types of military vehicles, An Assault Amphibian Vehicle (AAV), Dragon Wagon (DW), and Light Armored Vehicle (LAV). Two classifiers, kNN and ML, were exploited in the experiments, and kNN outperformed ML. However, kNN remains hampered by its computational complexity in application to a real field of a WSN in spite of its simplicity. In addition, they simply combined both acoustic and seismic data by using higher dimensional feature vectors only for improving the accuracy in a node. However, the method needs more resources at the same time, so that it might be inappropriate to an indigent WSN system. Also, such a simple feature-level fusion was inferior to a general decision fusion across sensors in our experiments.

Marco et al. described a scheme of feature extraction, target classification, and region-level information fusion [2]. They extracted spectrum-based feature vectors based on the acoustic and seismic signals. The two features are a little different, however. For the acoustic modality, they chose the first 100 points and extracted 50-dimensional FFT-based features with a resolution of 19.375 Hz, while the seismic features were extracted through choosing the first 50 points. The kNN, ML and SVM are experimented to classify moving targets, AAV and DW, and noise, so that the SVM outperformed the others. However, the complexity of each algorithm was not investigated. Although the SVM showed the best performance, its complexity and amount of required memory are not adequate to general WSN systems that allow small resources. For fusing the information among nodes in a region, they made an experiment on some distance-based fusions, such as the nearest neighbor and MAP (Maximum A Posterior) Bayesian. However, they did not consider any fusion scheme within a sensor node.

In this paper, the GMM algorithm is used for the target classification in order to improve the classification rate in each sensor of a node. The sensor fusion using a decision tree, built by the CART algorithm, is then utilized to improve the classification rate. We then analyze the performance and the complexity of the proposed algorithms. In addition, the performance variations according to the kinds of features, the feature dimensions, and the sampling rates are analyzed since they are tightly coupled with the amount of required data to increase the efficiency.

3. Proposed Classification Scheme in a Node

3.1 Target Classification Using GMM Algorithm

The GMM algorithm has been successfully applied to text classification, image information retrieval, speech signal processing, etc. [6], [7]. GMM-based classifiers in WSNs can be employed in a node of a WSN in the same way as the applications mentioned above.

Let $\boldsymbol{x}$ be an N-dimensional vector and $p(\boldsymbol{x}|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a GMM with M component densities as Eq. (1).

$$p(\boldsymbol{x}|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{m=1}^{M} \alpha_m N(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \tag{1}$$

where $\alpha_m > 0$, $m = 1, \ldots, M$ are the mixture weights, with which the summation of α_m is 1, and $N(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$, $m = 1, \ldots, M$ are the N-variate Gaussian densities with mean vector $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$. GMMs can assume several different forms, depending on the type of covariance matrices.

The parameters $\boldsymbol{\Theta}_i = (\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ of the GMM should be trained to apply the above classification procedure. It is commonly performed by the EM algorithm [8] using training data. Let $w(i|\boldsymbol{x}_j)$ be the class conditional probability with which the jth data $\boldsymbol{x}_j$ is generated from the ith Gaussian component. Estimations of the mixture weight α_i, the mean vector $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$ is carried out through iterations of sequential parameter updating by from the following Eq. (2) to Eq. (5) for each component i. Through these operations, the likelihood of the model converges to a local maximum.

$$w(i|\boldsymbol{x}_j) = \frac{\hat{\alpha}_i N(\boldsymbol{x}_j|\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i)}{\sum_{m=1}^{M} \hat{\alpha}_m N(\boldsymbol{x}_j|\hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m)} \tag{2}$$

$$\hat{\alpha}_i = \frac{1}{N} \sum_{j=1}^{N} w(i|\boldsymbol{x}_j) \tag{3}$$

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{j=1}^{N} w(i|\boldsymbol{x}_j)\boldsymbol{x}_j}{\sum_{j=1}^{N} w(i|\boldsymbol{x}_j)} \tag{4}$$

$$\hat{\boldsymbol{\Sigma}}_i = \frac{\sum_{j=1}^{N} w(i|\boldsymbol{x}_j)(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}}_i)^T(\boldsymbol{x}_j - \hat{\boldsymbol{\mu}}_i)}{\sum_{j=1}^{N} w(i|\boldsymbol{x}_j)} \tag{5}$$

3.2 Decision Tree-Based Sensor Fusion

There are two levels, the feature-level and the decision-level, to fuse data of sensors in a node [15]. We compared those levels with each other. In our fusion algorithm, the decision-level fusion is done by using a decision tree constructed by the CART.

The CART algorithm, proposed by Breiman et al. in 1984 [10], is a tree-building technique that could be ideally

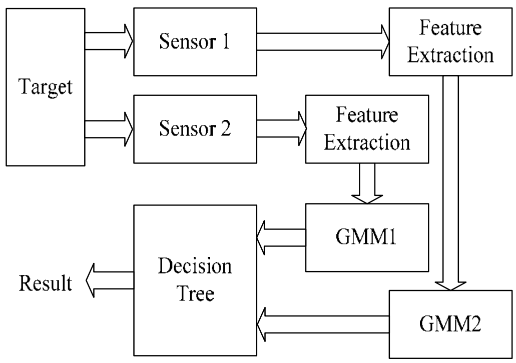

Fig. 1 Process of sensor fusion using a decision tree.

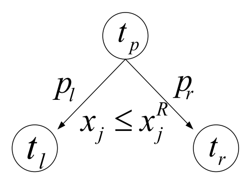

Fig. 2 Splitting algorithm of CART.

suited to the generation of information fusion rules among sensors. Log likelihoods for all candidates scored by the GMM algorithm are used as the input vector of the decision tree, made by the CART, as shown in Fig. 1. In other words, after the evaluation of log likelihoods of all candidates for each sensor, they are normalized by the length of their own detected event, respectively, concatenated with each other, and inputted to the decision tree. Finally, the best candidate is produced as a result.

The CART generally consists of three basic steps: maximum tree building, tree pruning, and optimal tree selection. A detailed description is as follows.

3.2.1 Construction of Maximum Tree

The first step is to build the maximum tree through splitting the learning samples up to the last observations, i.e., when terminal nodes contain observations only of one class. Let t_p be a parent node and t_l and t_r be the left and right child nodes of parent node t_p, respectively. Consider the learning samples with variable matrix X with M number of variables x_j and N observations. Let class vector Y comprise N observations with the total amount of K classes.

The classification tree is built in accordance with the splitting rule, the rule that performs the splitting of learning samples into smaller parts, as shown in Fig. 2.

Each data have to be divided into two parts with maximum homogeneity which is defined by the so-called impurity function $i(t)$. Let x_j be a variable j and x_j^R be the best splitting value of variable x_j. Since the impurity of parent node t_p is constant for any of the possible splits, $x_j \leq x_j^R$, $j = 1, \ldots, M$, the maximum homogeneity of the left and right child nodes will be equivalent to the maximization of the change of impurity function $\Delta i(t)$ in Eq. (6).

$$\Delta i(t) = i(t_p) - E[i(t_c)] \tag{6}$$

where, t_c is the left and right child nodes of the parent node t_p. Assuming that P_l and P_r are the probability of the right and the left nodes, Eq. (7) is obtained.

$$\Delta i(t) = i(t_p) - P_l i(t_l) - P_r i(t_r) \tag{7}$$

Therefore, at each node, the CART solves the maximization problem given in Eq. (8).

$$\underset{x_j \leq x_j^R, j=1,\ldots,M}{\arg\max} \left[i(t_p) - P_l i(t_l) - P_r i(t_r) \right] \tag{8}$$

Equation (6) implies that the CART will search through all possible values of all variables in matrix X for the best split question $x_j < x_j^R$ that will maximize the change of impurity measure $\Delta i(t)$.

The Gini splitting rule is one of the most broadly used algorithms. It uses the following impurity function $i(t)$ in Eq. (9).

$$i(t) = \sum_{k \neq l} p(k|t) p(l|t) \tag{9}$$

where, k and l are the class indices and $p(k|t)$ is the conditional probability of class k if we are in node t. Applying Eq. (9) to Eq. (8), we will get the following change of impurity measure $\Delta i(t)$ in Eq. (10).

$$\Delta i(t) = -\sum_{k=1}^{K} p^2(k|t_p) + P_l \sum_{k=1}^{K} p^2(k|t_l) + P_r \sum_{k=1}^{K} p^2(k|t_r) \tag{10}$$

Therefore, the Gini algorithm will solve the problem given in Eq. (11).

$$\underset{x_j \leq x_j^R, j=1,\ldots,M}{\arg\max} \left[-\sum_{k=1}^{K} p^2(k|t_p) + P_l \sum_{k=1}^{K} p^2(k|t_l) + P_r \sum_{k=1}^{K} p^2(k|t_r) \right] \tag{11}$$

The Gini algorithm will search in the learning samples for the largest class and isolate it from the rest of the data. The Gini works well for noisy data.

3.2.2 Tree Pruning

Maximum trees may turn out to be of very high complexity and consist of hundreds of levels. Therefore, they have to be optimized by pruning insignificant nodes. There are two punning algorithms: optimization by number of points in each node and cross-validation. In the first algorithm, we say that the splitting is stopped when the number of observations in a node is less than a predefined required minimum N_{min}. The second pruning algorithm, cross validation, is to find the optimal proportion between the complexity of the tree and misclassification error. It is achieved through the cost-complexity function of Eq. (12)

$$R_\alpha(T) = R(T) + \alpha\left(\hat{T}\right) \rightarrow \min_T \qquad (12)$$

where, $R(T)$ is a misclassification error of tree T and $\alpha(\hat{T})$ which is the complexity measure that depends on $\hat{T}$, which is the total sum of terminal nodes in the tree. The value of α is found through the sequence of in-sample testing when a part of the learning sample is used to build the tree; the other part of the data is taken as a testing sample. The process is repeated several times for randomly selected learning and testing samples.

3.2.3 Optimal Tree Selection

The maximum tree tends to overfit the learning dataset. The goal in selecting the optimal tree is thus to find the correct complexity parameter α such that the information in the learning dataset is fit but not overfit. The problem of finding this value for α can be solved by using either an independent set of data or the technique of cross validation.

4. Experiments & Evaluations

4.1 Dataset & System Environments

In our experiments, the dataset, recorded in a real world WSN environment for the DARPA SensIT (sensor information technology) program [2], was utilized. The dataset contains signals from three classes, two types of vehicles, and background noise. The vehicles were a heavy wheeled truck (DW) and a tracked vehicle (AAV). All signals were sampled at 4,960 Hz and 16 bits per sample using the WINS NG 2.0 nodes with acoustic and seismic sensors deployed near a junction of three roads. The event signals, created by ten runs for AAV and DW, and noise signals, collected manually from background signals, were used toward the experiments. An example of LPC (linear predictive coefficient) power spectra of acoustic signals characterizing those objects is depicted in Fig. 3, where the LPC order was 30. In estimating the LPC, the autocorrelation method and the Levinson-Durbin algorithm were utilized [19]. Since the constant false alarm rate (CFAR) detection algorithm given by [2] is a kind of adaptive detection algorithm which detects an event by using the energy information of signals, some noise signals in a run could be higher than the events' signals in another run, as shown in the low frequency range of Fig. 3.

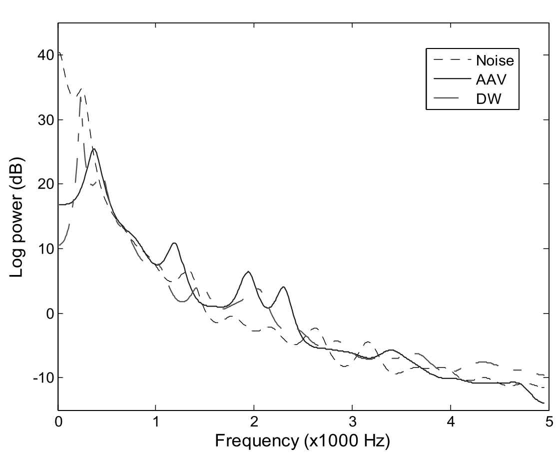

Fig. 3 Spectral characteristics of acoustic signal for each target.

Basically, the MFCCs were extracted and used as features because they are not only well-matched to GMM-based algorithms, but because their performances are also verified already. The dimension of the feature should be considered in terms of the computational resources and will be discussed in Sect. 4.3.

Figure 4 shows the structure of the node-level classification process based on our proposed scheme. Firstly, target intervals are detected by using the energy values of sensed signals. For the intervals, the acoustic/seismic signals are inputted to extract MFCCs. Then, length-normalized log-likelihoods for AAV, DW, and noise are evaluated using the corresponding GMM and the feature vectors. To compare the classification performance of the proposed algorithm, kNN and SVM were evaluated. In kNN, the number of neighbors, k, was 100 and evaluated based on the L2-norm distance, while the radial basis function (RBF) was used as a kernel function of SVM [11], [13]. Given a training set of instance-label pairs, SVM finds a separating hyperplane with the maximal margin through the training process, and then improves the performance by using the plane. In evaluating the performance of the GMM, the number of mixtures was four, and their parameters were trained by the expectation maximization algorithm. The final step of the classification process is the decision tree. It is generated by the CART and improves the classification rate by using the log-likelihoods themselves from the GMM as its inputs.

4.2 Optimal Sampling Rate and Gaussian Mixtures

In the viewpoint of resource management, it is indispensable to analyze the sampling rate in WSNs since it is closely related to the memory size and the amount of processing [16]. An optimal sampling rate should be obtained and applied to a WSN system to reduce some unnecessary redundancy. To estimate the optimal sampling rate, we re-sampled our data at 100, 300, 500, 700, 1,000, 2,000, 3,000, 4,000 and 4,960 Hz, and evaluated the classification performance for each frequency. Since even a small difference in sampling rates may cause a significant variation in classification accuracies at somewhat low sampling rates below 1 kHz compared with the variation in relatively high sampling rates, experimental intervals below 1 kHz are narrower than those above the sampling rate.

Figure 5 shows the classification accuracies accord-

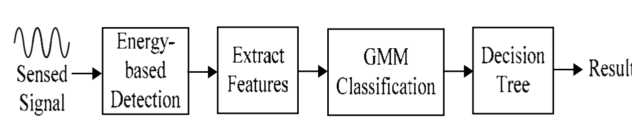

Fig. 4 Proposed node-level classification system.

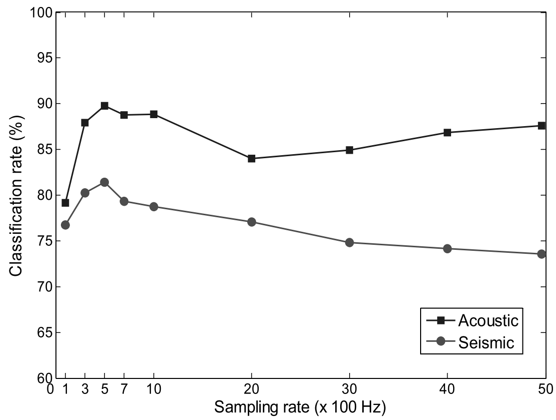

Fig. 5 Accuracies according to the sampling rate.

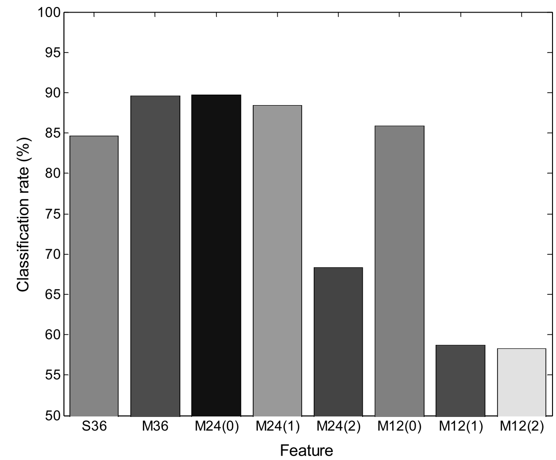

Fig. 6 Classification rates for each feature.

Table 1 Classification accuracies according to the number of Gaussian mixtures (unit:%).

Sensor \ Mixture#	1	2	4	8	16
Acoustic	83.28	87.44	**89.76**	88.70	88.46
Seismic	74.17	76.05	77.57	79.24	**81.44**

ing to the sampling rates for acoustic and seismic signals. For each sampling rate, the number of Gaussian mixtures was equally four, and the dimension of feature vectors was 24 consisting of the 12th-order MFCC and its derivatives. It can be said that the optimal accuracy can be obtained from approximately a 500 Hz in the case of classifying AAV, DW and noise with acoustic or seismic signals. In other words, some redundant information starts to be included above 500 Hz sampling rate. This is because significant discriminating characteristics are shown mainly below 500 Hz, as depicted in Fig. 5. If we consider the power consumption problem of WSN systems, it seems unnecessary to increase the sampling rate more than 500 Hz as in our experiment. Interestingly, the performances below 300 Hz were dramatically decreased regardless of the kind of sensors, which seems that it is becoming more difficult that any significant spectral information can be found below the frequency. Some decreasing classification rates in higher sampling rates occur because the number of Gaussian distributions is not optimal for each sampling rate. Finally, since the performance of acoustic signals is superior to that of seismic signals, it can be said that the quality of acoustic signals is better than seismic signals for target classification in WSNs.

Because a Gaussian function basically consists of mean and variance vectors, 1,728 bytes are additionally needed per mixture if the 24th-order feature vector, three target classes, and 4 bytes per parameter are used. Thus, we should minimize the number of mixtures as much as possible. The accuracies according to each of the number of mixtures are shown in Table 1. When four mixtures are used, the best classification rate of acoustic signals is obtained. The appropriate number of mixtures improves the accuracy, but too many mixtures can cause an overfitting problem which aggravates the accuracy [14]. We can see such a situation by comparing the accuracies with each other when four mixtures, eight mixtures, and sixteen mixtures are used. For seismic signals, the best classification rate is produced with 16 mixtures.

4.3 Comparison of Performance According to Features

As mentioned before, the issues in terms of the kinds and the dimensions of features should be analyzed in order to reduce the amount of required resources [16].

We compared the performance of the MFCC with that of the 36th-order spectral feature, named S36, modified from [2]. In order to analyze the performance for the various dimensions of the MFCC, 12th-order MFCCs, named M12(0) in our experiments, were basically extracted per 10 msec, and their 1st-derivatives, named M12(1), and 2nd-derivatives, named M12(2), were properly concatenated as feature parameters. In Fig. 6, M36, M24(0), M24(1), and M24(2) represent static + 1st-derivatives + 2nd-derivatives, static + 1st-derivatives, static + 2nd-derivatives, and 1st-derivatives + 2nd-derivatives, respectively. No energy features were considered in order to avoid false alarms caused by noises with a high energy. In fact, since the complexity of the MFCC extraction largely depends on the FFT ($N \log_2 N$), the complexity is similar to that of the existing spectral features. The MFCCs are calculated using the inverse discrete cosine transform of the logarithm of the mel-scaled filter bank signal energies. Generally, they result in high performances while using a low-dimensional signal representation.

Experimental results reveal that M24(0) shows the best classification rate and outperforms the conventional spectral features, although it uses a smaller feature dimension. Among the 24th-order feature vectors based on the MFCC, feature vectors containing static MFCC vectors show relatively good performances.

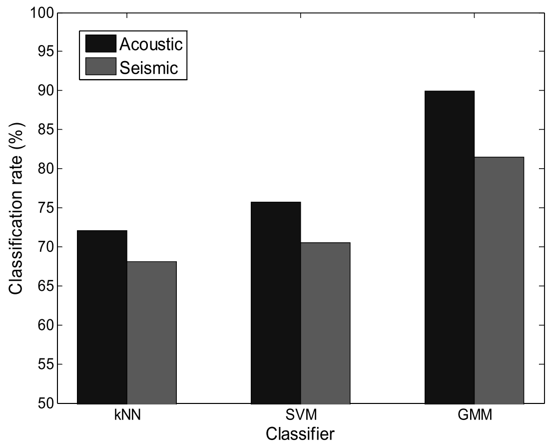

Fig. 7 Comparison of classification rates of classifiers.

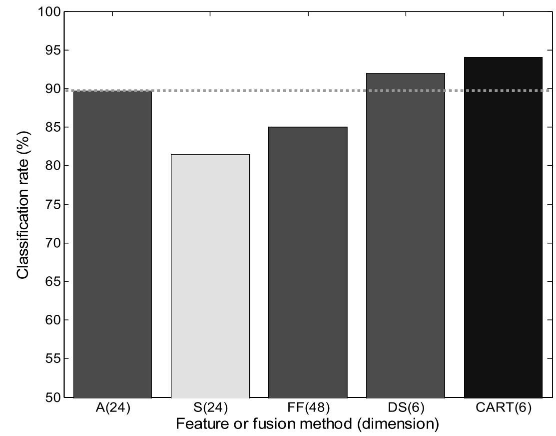

Fig. 8 Accuracies of fusion methods in a sensor node. (A: Acoustic, S: Seismic, FF: Feature fusion, DS: Dempster-Shafer)

4.4 Comparative Analysis of Classifier Performances

The performance of the GMM algorithm was evaluated with those of other classifiers, such as kNN and SVM. The performance of the GMM algorithm was produced using the 24th-order MFCC (M24(0)) and four mixtures per GMM state. All of the experiments were performed using the 5-fold cross validation method.

Figure 7 shows the comparative performances of the vehicle classification according to the classifiers; The GMM algorithm outperforms the others. The average classification rates using acoustic/seismic signals at the sampling rate of 500 Hz were 72.03/68.09%, 75.32/70.48% and 89.76/81.44% for kNN, SVM and GMM, respectively. We can see that the GMM, as a parametric method, produces a satisfactory performance compared to the other algorithms for classifying moving vehicles in a WSN using acoustic and seismic sensors.

4.5 Sensor Fusion Using a Decision Tree Generated by CART Algorithm

Since targets have different signatures corresponding to multiple modalities, e.g., acoustic and seismic, multimodal fusion aims to aggregate such data optimally in order to improve the overall classification performance. The rationale is that individual modalities provide complementary information. We focus on a node-level sensor fusion in this paper. There are two methods to fuse heterogeneous sensors in a node. One is to merge multiple features themselves, and the other is to fuse various classification results. We experimented on and compared those methods with each other.

In our study, the CART algorithm was used for the sensor fusion since it can generate a binary decision tree which produces the minimum classification rate for given training data. The decision tree can utilize directly the log-likelihoods themselves scored by the GMM, while the famous Dempster-shafer fusion algorithm needs some additional pre-processing steps [18]. To make a classification tree by CART, the Gini algorithm was used to split a node of the tree. Then, the 10-fold cross-validation method was used to identify the subtree with the lowest misclassification rate. The minimal size below which a node will not be split is ten. A total of 28,080 log-likelihood vectors, which were evaluated using the three target class GMMs for each sensor, were used for the experiment.

Table 2 Variable importance for sensor fusion in CART.

Variable	Score
Log-likelihood of DW by an acoustic sensor	100.00
Log-likelihood of AAV by an acoustic sensor	93.20
Log-likelihood of noise by a seismic sensor	83.64
Log-likelihood of noise by an acoustic sensor	69.44
Log-likelihood of DW by a seismic sensor	38.24
Log-likelihood of AAV by a seismic sensor	13.31

Figure 8 shows the classification accuracy of the proposed fusion scheme, and it is compared with that of the Dempster-Shafer algorithm. Also, the accuracies before the fusion are included. It is a commonly understood that any fusion method should improve the classification rate of each sensor itself. It is shown that the accuracy of an acoustic sensor is rather higher than that of feature-level fusion, which means that feature-level fusion is invalid in the case that the discriminativeness of a sensor's feature is significantly different from that of others. This causes even the superiority of the best sensor to be aggravated. On the other hand, the sensor fusion based on a decision tree, generated by CART, produces a considerable improvement compared with the accuracy of an acoustic sensor itself. In addition, the proposed fusion algorithm outperformed the Dempster-Shafer algorithm which is one of widely applied fusion schemes in WSNs. Therefore, the proposed fusion algorithm can be said to be a very effective method for node-level sensor fusion.

Table 2 shows the importances of all variables used in CART, which are relative values and are calculated based on the contribution of variables in constructing the deci-

Table 3 Classification accuracy & Confusion matrix.

Class	Number of cases	Accuracy (%)	Confusion Matrix		
			AAV	DW	Noise
AAV	8,160	95.23	7,771	228	161
DW	7,680	90.04	488	6,915	277
Noise	12,240	95.89	90	413	11,737

Table 4 Data size of trained results of classifiers. (unit: kByte)

kNN	SVM	GMM
1,333,858.21	264.28	2.50

Table 5 Average computing time per feature vector. (unit: msec)

kNN	SVM	GMM
103.965	30.011	0.009

sion tree [17]. AAV and DW can be identified by acoustic sensors better than seismic sensors, while noise can be better classified by seismic sensors than acoustic sensors. This means that acoustic sensors can receive significant help from seismic sensors through sensor fusion in identifying noise. The final classification accuracy of each target class and the confusion matrix are shown in Table 3. Each target has a satisfactory accuracy through our node-level classification scheme. Finally, we can obtain 94.10% accuracy, a 4.34% improvement compared with the accuracy of acoustic sensors before fusion, which means 42.38% in an error reduction sense.

4.6 Complexity Analysis

To apply a classifier to a sensor node with a hard limitation of resources, its complexity should be as low as the allowable resources. Thus, this paper analyzes the complexity of the proposed classification scheme experimentally.

In kNN, a typical nonparametric algorithm, the amount of computation for classification increases in proportion to the size of the training set. Such a nonparametric algorithm usually has all instances as its trained set so that the size of the trained set should be very huge and unable to be applied to a sensor node directly. In the case of SVM, the information of its support vectors retrieved as a result of training increase in proportion to the size of the training set as well, even though the size is smaller than that of kNN. In contrast to them, GMM, a parametric-based classifier, has the parameters of its model as a trained result. Therefore, it can have a constant number of parameters for classification regardless of the size of the training set. Table 4 shows the data size of each classifier produced by the training process in our experiment. While kNN roughly needs 1.33 GBytes and SVM 264 kBytes, GMM requires 2.5 kBytes which is a very small amount of memory. This amount of resources is appropriate for application toward general sensor nodes in WSNs.

Table 5 shows the average computing time per feature vector for each classifier measured by a Pentium-IV computer equipped with a 3.2 GHz CPU and 4 GBytes of memory. We can see that GMM complete the computation in a very short time, while kNN and SVM require a huge amount of computing time. We can conclude that the GMM has no problem in being applied to a sensor node in the aspect of computational speed.

Let us consider the complexity of the decision tree generated by the CART algorithm. It is usually very quick to fuse some decision information based on a decision tree since the tree algorithm is very simple [10]. The computational complexity in searching the tree is almost ignorable compared with those of the above classification algorithms. Also, only parameters for node indexes, rules and class labels need to be stored. In addition, by controlling the number of tree nodes, the required memory size can be controlled within an allowable range easily. In our experiments, we used 49 splitting nodes and 50 terminal nodes in order to obtain our results.

5. Conclusion & Future Works

In this paper, a node-level target classification scheme based on the GMM algorithm was proposed, experimented on, and evaluated using acoustic and seismic data in a WSN. As a node-level classification performance, we can completely obtain 94.10% accuracy, a very high rate, through the proposed scheme.

The GMM algorithm is suggested as a classifier of WSN systems, and we analyzed its complexity in order to consider the feasibility of its being applied to a real node in WSNs. We found that the GMM, as a parametric method, not only outperforms the other classifiers, the kNN and SVM, but that it also needs a relatively small resource compared with them. The sensor fusion using a decision tree, estimated by CART, shows a satisfactory improvement compared with the feature-level and Dempster-Shafer fusion method even though the algorithm is simple and needs a small amount of resources. According to our analyses, the acoustic sensors are more useful for identifying vehicles, AAV and DW, while the seismic sensors are more useful to identify noise. In addition, this paper gives a clue toward how high the sampling rate should become in order to guarantee the classification accuracy in WSN applications. Also, the kind of feature and its dimensions should be analyzed since they are closely related to the amount of required resources. Thus, we can find the optimal classification scheme which can be applied to a sensor node.

Our future works are to implement the proposed scheme on a node in a real field of WSN systems, and to employ a collaboration scheme among sensor nodes if necessary.

Acknowledgements

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2008-C1090-0801-0015), and the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MOST) (No. R0A-2008-000-10038-0).

References

[1] C. Meesookho, S. Narayanan, and C.S. Raghavendra, "Collaborative classification applications in sensor networks," Second IEEE Sensor Array and Multichannel Signal Processing Workshop, pp.370–374, 2002.
[2] M. Duarte and Y.-H. Hu, "Vehicle classification in distributed sensor networks," J. Parallel Distrib. Comput., vol.64, no.7, pp.826–838, 2004.
[3] D. Li, K.D. Wong, Y.H. Hu, and A.M. Sayeed, "Detection, classification and tracking of targets in distributed sensor networks," IEEE Signal Process. Mag., vol.19, no.2, pp.17–29, 2002.
[4] R.R. Brooks, C. Griffin, and D. Friedlander, "Distributed target classification and tracking in sensor networks," Proc. IEEE, pp.1163–1171, 2003.
[5] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," Comput. Netw. J., vol.46, no.5, pp.605–634, 2004.
[6] D.A. Reynolds, A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification, Ph.D. Thesis, Georgia Institute of Technology, 1992.
[7] T.H. Falk and W.-Y. Chan, "A sequential feature selection algorithm for GMM-based speech quality estimation," European Signal Processing Conference, 2005.
[8] G. Xuan, W. Zhang, and P. Chai, "EM algorithms of Gaussian mixture model and hidden Markov model," Proc. International Conference on Image Processing, pp.145–148, 2001.
[9] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," J. Computer Science & Technology, vol.16, no.6, pp.582–589, 2001.
[10] L. Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees, Wadsworth, 1984.
[11] B.V. Dasarathy, Nearest Neighbor: Pattern Classification Techniques, IEEE Computer Society, 1990.
[12] S.M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice Hall, 1993.
[13] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, vol.2, pp.121–167, 1998.
[14] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, 2nd ed., John-Wiley, 2000.
[15] L.A. Klein, Sensor and data fusion, SPIE, 2004.
[16] M. Basu and T.K. Ho, Data Complexity in Pattern Recognition (Advanced Information and Knowledge Processing), Springer, 2006.
[17] W.-Y. Loh and N. Vanichsetakul, "Tree-structured classification via generalized discriminant analysis," J. American Statistical Association, vol.83, no.403, pp.715–725, 1988.
[18] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang, "Sensor fusion using Dempster-Shafer theory," Proc. 19th IEEE Instrumentation and Measurement Technology Conference, pp.7–12, USA, 2002.
[19] M.H. Hayes, "Signal modeling," in Statistical Digital Signal Processing and Modeling, pp.129–214, John Wiley & Sons, 1996.

Youngsoo Kim received the B.S. degree in computer science from Korea Air Force Academy in 1994, and the M.S. degree in computer science from Sogang University, Seoul, Korea, in 2001. He is currently a Ph.D. candidate in the Department of Engineering from Information and Communications University (ICU), Daejeon, Korea. His main research areas are in signal processing and intelligent system technologies for wireless sensor networks.

Sangbae Jeong received the B.S. degree in electronics engineering from Pusan National University, Pusan, Korea, in 1997, the M.S. degree in electrical and electronics engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1999, and the Ph.D. degree in electronics engineering from ICU, Daejeon, Korea, 2002. From 2002 to 2005, he was a research staff member in Samsung Advanced Institute of Technology, Yongin, Korea. Since 2006, he has been a research assistant professor in Digital Media Laboratory of ICU. His main research areas are in embedded speech recognition systems, microphone array-based beamforming technologies, and biomedical signal processing.

Daeyoung Kim received the B.S. and the M.S. degrees in electrical engineering from Pusan National University, Pusan, Korea, in 1990 and 1992, and the Ph.D. degree in computer science from University of Florida, USA, in 2001. From 1992 to 1997, and he was with Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. From 2001 to 2002, he was a research professor in Arizona University, USA. Beginning in 2002, he has been a faculty member of the School of Engineering, ICU. Currently, he is an associate professor in the Department of Engineering, ICU, and a Director in Global Sensor Network Research Laboratory of Korea, ICU. His research interests include real-time embedded systems, RFID, wireless sensor networks, and data processing.