JAISCR, 2012, Vol. 2, No. 4, pp. 301 – 315

AN IMPROVED ANT COLONY OPTIMIZATION ALGORITHM AND ITS APPLICATION TO TEXT-INDEPENDENT SPEAKER VERIFICATION SYSTEM

Mehdi Hosseinzadeh Aghdam
Department of Computer Engineering and Information Technology, Payam Noor University
PO BOX 19395-3697, Tehran, IRAN
mhaghdam@yahoo.com

Abstract

With the growing trend toward remote security verification procedures for telephone banking, biometric security measures and similar applications, automatic speaker verification (ASV) has received a lot of attention in recent years. The complexity of ASV system and its verification time depends on the number of feature vectors, their dimensionality, the complexity of the speaker models and the number of speakers. In this paper, we concentrate on optimizing dimensionality of feature space by selecting relevant features. At present there are several methods for feature selection in ASV systems. To improve performance of ASV system we present another method that is based on ant colony optimization (ACO) algorithm. After feature selection phase, feature vectors are applied to a Gaussian mixture model universal background model (GMM-UBM) which is a text-independent speaker verification model. The performance of proposed algorithm is compared to the performance of genetic algorithm on the task of feature selection in TIMIT corpora. The results of experiments indicate that with the optimized feature set, the performance of the ASV system is improved. Moreover, the speed of verification is significantly increased since by use of ACO, number of features is reduced over 80% which consequently decrease the complexity of our ASV system.

1 Introduction

Automatic speaker recognition (ASR) systems are generally divided into two categories, namely: automatic speaker identification (ASI) systems which are designed to answer the question "who is the speaker?" or automatic speaker verification (ASV) systems that aim to answer the question "is the speaker who they claim to be?".

Automatic speaker verification refers to the task of verifying speaker's identity using speaker-specific information contained in speech signal. Speaker verification methods are totally divided into text-dependent and text-independent applications. When the same text is used for both training and testing, the system is called to be text-dependent while for text-independent operation; the text used to train and test of the ASV system is completely unconstrained. Text independent speaker verification requires no restriction on the type of input speech. In contrast, text independent speaker verification usually gives less performance than text dependent speaker verification, which requires test input to be the same sentence as training data [1].

Applications of speaker verification can be found in biometric person authentication such as an additional identity check during credit card payments over the Internet while, the potential applications of speaker identification can be found in multi-user systems. For instance, in speaker tracking the task is to locate the segments of given speaker(s) in an audio stream [2, 3]. It has also potential applications in automatic segmentation of teleconferences

and helping in the transcription of courtroom discussions.

Speech signals contain a huge amount of information and can be described as having a number of different levels of information. At the top level, we have lexical and syntactic features, below that are prosodic features, further below these are phonetic features, and at the most basic level we have low-level acoustic features, which generally give information on the system that creates the sound, such as the speakers' vocal tract. Information solely about how the sound is produced (from low level acoustic features) should give enough information to identify accurately a speaker as this is naturally speaker dependent and independent of text [4].

Low level acoustic features also contain some redundant features, which can be eliminated using feature selection (FS) techniques. The objective of FS is to simplify a dataset by reducing its dimensionality and identifying relevant underlying features without sacrificing predictive accuracy. By doing that, it also reduces redundancy in the information provided by the selected features [5]. In real world problems FS is a must due to the abundance of noisy, irrelevant or misleading features. Selected features should have high inter-class variance and low intra-class variability. Ideally they should also be as independent of each other as possible in order to minimize redundancy.

Feature selection is extensive and it spreads throughout many fields, including signal processing [6], face recognition [7], text categorization [8], data mining and pattern recognition [5, 7]. Feature selection has been rarely used in ASV systems. Day and Nandi [4] employed genetic programming for FS; also L plus-R minus feature selection algorithm is used by Pandit and Kittkr [9] for text-dependent speaker verification. Cohen and Zigel [10] employed Dynamic Programming for FS in speaker verification and Ganchev et al. [11] used information gain and gain ratio for FS in ASV task.

Among too many methods which are proposed for FS, population-based optimization algorithms such as genetic algorithm (GA)-based method and ant colony optimization (ACO)-based method have attracted a lot of attention. These methods attempt to achieve better solutions by application of knowledge from previous iterations. Genetic algorithms are optimization techniques based on the mechanism of natural selection. They used operations found in natural genetics to guide itself through the paths in the search space [12]. Because of their advantages, recently, GAs have been widely used as a tool for feature selection in data mining [13].

Meta-heuristic optimization algorithm based on behavior of ants was represented in the early 1990s by M. Dorigo and colleagues [14]. ACO is a branch of newly developed form of artificial intelligence called swarm intelligence. Formally, swarm intelligence refers to the problem-solving behavior that emerges from the interaction of cooperative agents, and computational swarm intelligence refers to algorithmic models of such behavior. ACO algorithm is inspired by social behavior of ant colonies. Although they have no sight, ants are capable of finding the shortest route between a food source and their nest by chemical materials called pheromone that they leave when moving [15].

ACO algorithm was firstly used for solving traveling salesman problem (TSP) [15] and then has been successfully applied to a large number of difficult problems like the quadratic assignment problem (QAP) [16], routing in telecommunication networks, graph coloring problems, scheduling, etc. This method is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal subset every time [7].

In this paper we propose an ACO algorithm for feature selection in ASV systems based on GMM-UBM and apply it to larger feature vectors containing Mel-frequency cepstral coefficients (MFCCs) and their delta coefficients, two energies and linear prediction cepstral coefficients (LPCCs) and their delta coefficients. Then, feature vectors are applied to a Gaussian mixture model universal background model (GMM-UBM) which is a text-independent speaker verification Model. Finally the verification quality and the length of selected feature vector are considered for performance evaluation.

The rest of this paper is organized as follows. Section 2 presents a brief overview of ASV systems. Feature selection methods are described in section 3. Ant colony optimization and proposed feature selection algorithm are explained in Section 4. Genetic algorithms are described in section 5. Section 6 reports computational experiments. It also includes a brief discussion of the results ob-

tained and finally the conclusion is offered in the last section.

2 An Overview of ASV Systems

The typical process in most proposed ASV systems involves: some form of preprocessing of the data (silence removal) and feature extraction, followed by some form of speaker modeling to estimate class dependent feature distributions (see Figure 1). A comprehensive overview can be found in [17]. Adopting this strategy the ASV problem can be further divided into the two problem domains of:

1 Preprocessing, feature extraction and selection

2 Speaker modeling and matching.

These two steps are described in following sections in more details.

2.1 Feature Extraction

The original signal, the speech waveform, contains all information about the speaker, and each step in the extraction process can only reduce the mutual information or leave it unchanged. The objective of the feature extraction is to reduce the dimension of the input signal and thereby reduce the complexity of the system. The main task for the feature extraction process is to pack as much speaker-discriminating information as possible into as few features as possible.

The choice of features in any proposed ASV system is of primary concern, because if the feature set does not yield sufficient information then trying to estimate class dependent feature distributions is futile [18]. Most feature extraction techniques in speaker verification were originally used in speech recognition. However, the focus in using these techniques was shifted to extract features with high variability among people. Most commonly used features extraction techniques, such as Mel-frequency cepstral coefficients (MFCCs) and linear prediction cepstral coefficients (LPCCs) have been particularly popular for ASV systems in recent years. This transforms give a highly compact representation of the spectral envelope of a sound [19].

Delta-features, regardless on what features they are based, can be computed as a one-to-one function of the features themselves. Therefore, the delta-features do not contain more information than is already in the features, and from the theory, no gain can be achieved by using them together with the features. However, the delta-features can be used as a simplified way of exploiting inter-feature dependencies in suboptimal schemes [4].

2.2 Speaker Modeling

The speaker modeling stage of the process varies more in the literature. The purpose of speaker modeling is characterizing an individual which is enrolled into an ASV system with the aim of defining a model (usually feature distribution values). The three most popular methods in previous works are Gaussian mixture models (GMM) [19,20], Gaussian mixture models universal background model (GMM-UBM) [21, 22] and vector quantization [23]. Other techniques such as decision trees [24], support vector machine (SVM) [25] and artificial neural network (ANN) [26] have also been applied. In this paper GMM-UBM is used for speaker modeling.

2.3 GMM-UBM approach

GMM-UBM is the predominant approach used in speaker recognition systems, particularly for text-independent task [22]. Given a segment of speech Y and a speaker S, the speaker verification task consists in determining if Y was spoken by S or not. This task is often stated as basic hypothesis test between two hypotheses: Y is from the hypothesized speaker S (H_0), and Y is not from the hypothesized speaker S (H_1). A likelihood ratio (LR) between these two hypotheses is estimated and compared to a decision threshold Φ. The LR test is given by:

$$LR(Y,H_0,H_1) = \frac{p(Y|H_0)}{p(Y|H_1)} \tag{1}$$

where Y is the observed speech segment, $p(Y|H_0)$ is the likelihood function for the hypothesis H_0 evaluated for Y, $p(Y|H_1)$ is the likelihood function for H_1 and Φ is the decision threshold for accepting or rejecting H_0. If $LR(Y,H_0,H_1)>\Phi$, H_0 is accepted else H_0 is rejected. A model denotedλ_{hyp}represents H_0, it is learned using an extract of speaker S voice. The model λ_{UBM} represents the alternative hypothesis, H_1, and is usu-

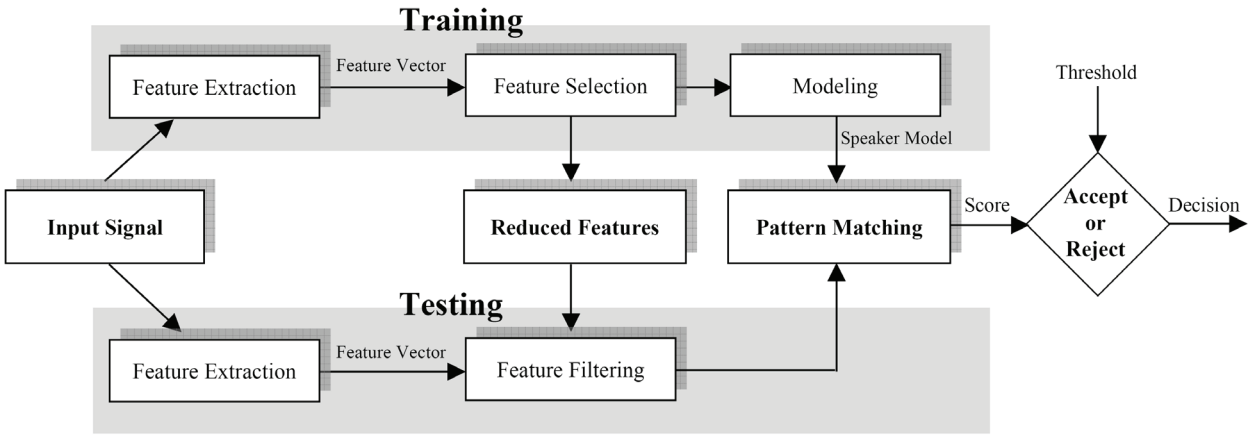

Figure 1. Overview of the speaker verification process

ally learned using data gathered from a large set of speakers.

The likelihood ratio statistic becomes $\frac{p(Y|\lambda_{hyp})}{p(Y|\lambda_{UBM})}$. Often, the logarithm of this statistic is used giving the Log *LR* (*LLR*):

$$LLR(Y) = \log p(Y|\lambda_{hyp}) - \log p(Y|\lambda_{UBM}) \quad (2)$$

In the presented approach, the models are Gaussian Mixture Models which estimate a probability density function by:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i N(x_t, \mu_i, \Sigma_i) \quad (3)$$

where $w_i, i = 1, \cdots .c$ are the mixture weights that satisfy $0 \leq w_i \leq 1, \sum_{i=1}^{c} w_i = 1$and $N(x_t, \mu_i, \Sigma_i)$are the d-variate Gaussian component densities with mean vectors μ_i and covariance matrices Σ_i

$$N(x_t, \mu_i, \Sigma_i) = \frac{\exp\left\{-\frac{1}{2}(x_t - \mu_i)' \Sigma_i^{-1} (x_t - \mu_i)\right\}}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \quad (4)$$

Usually a large number of components in the mixture and diagonal covariance matrices are used [21].

2.3.1 Universal Background Model

The UBM has been introduced and successfully applied by Reynolds [21] to speaker verification. It aims at representing the inverse hypothesis in the Bayesian test, i.e. it is designed to compute the data probability not to belong to the targeted speaker, i.e.λ_{hyp}. A UBM is learned with multiple audio files from different speakers, usually several hundreds. For speaker verification, some approaches consist in having specific UBM models, such as a UBM model per gender or per channel.

The UBM is trained with the EM algorithm on its training data. For the speaker verification process, it fulfills two main roles:

- It is the a priori model for all target speakers when applying Bayesian adaptation to derive speaker models.
- It helps to compute logarithm likelihood ratio much faster by selecting the best Gaussian for each frame on which likelihood is relevant [22].

3 Feature Selection Approaches

Feature selection is included in discrete optimization problems. The whole search space for optimization contains all possible subsets of features, meaning that its size is:

$$\sum_{s=0}^{n} \binom{n}{s} = \binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{n} = 2^n \quad (5)$$

where, n is the dimensionality (the number of features) and s is the size of the current feature subset [27]. Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity. However, the degree of optimality of the final feature subset is often reduced [5].

The objectives of feature selection are manifold, the most important ones being:

- To avoid over fitting and improve prediction performance.
- To provide faster and more cost-effective models.
- To gain a deeper insight into the underlying processes that generated the data.

In the context of classification, feature selection techniques can be organized into three categories, depending on their evaluation procedure: filter methods, wrapper methods and embedded methods [28]. If an algorithm performs FS independent of any learning algorithm (i.e. it is a completely separate preprocessor), then it is included in filter approach category. This approach is mostly includes selecting features based on inter-class separability criterion [28]. If the evaluation procedure is tied to the task (e.g. classification) of the learning algorithm, the FS algorithm is a sort of wrapper approach. This method searches through the feature subset space using the estimated accuracy from an induction algorithm as a measure of subset suitability. If the feature selection and learning algorithm are interleaved then the FS algorithm is a kind of embedded approach [27]. A common disadvantage of filter methods is that they ignore the interaction with the classifier. The wrapper method is computationally more involved, but takes the dependency of the learning algorithm on the feature subset into account [5].

In the wrapper approach the evaluation function calculates the suitability of a feature subset produced by the generation procedure and it also compares that with the previous best candidate, replacing it if found to be better. A stopping criterion is tested in each of iterations to determine whether or not the FS process should continue.

Other famous FS approaches are based on the genetic algorithm [13], simulated annealing, particle swarm optimization [29] and ant colony optimization [30, 7]. Reference [30] proposes a subset search procedure based on ACO for speech classification problem. The hybrid of ACO and mutual information has been used for feature selection in the forecaster [31]. Furthermore, ACO is used for finding rough set reducts [5]. Also, in [7] an ACO-based method has been used in the application of face recognition systems and some surveys of feature selection algorithms are given in [32, 13].

4 Ant Colony Optimization

In the early 1990s, ant colony optimization was introduced by M. Dorigo and colleagues as a novel nature-inspired meta-heuristic for the solution of hard combinatorial optimization problems. An ant colony optimization algorithm is essentially a system based on agents which simulate the natural behavior of ants, including mechanisms of cooperation and adaptation. The inspiring source of ACO is the foraging behavior of real ants [33].

The first ACO algorithm developed was the ant system [34], and since then several improvement of the AS have been devised [35, 36]. The ACO algorithm is based on a computational paradigm inspired by real ant colonies and the way they function. The underlying idea was to use several constructive computational agents (simulating real ants). A dynamic memory structure incorporating information on the effectiveness of previous choices based on the obtained results, guides the construction process of each agent.

The paradigm is based on the observation made by ethologists about the medium used by ants to communicate information regarding shortest paths to food by means of pheromone trails. A moving ant lays some pheromone on the ground, thus making a path by a trail of this substance. While an isolated ant moves practically at random, exploration, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, exploitation, and consequently reinforces the trail with its own pheromone. What emerges is a form of autocatalytic process through which the more the ants follow a trail, the more attractive that trail becomes to be followed. The process is thus characterized by a positive feedback loop, during which the probability of choosing a path increases with the number of ants that previously chose the same path. The mechanism above is the inspiration for the algorithms of the ACO family [37].

4.1 Proposed ACO Algorithm for Feature Selection

As mentioned earlier given a feature set of size n, the FS problem is to find a minimal feature subset of size s ($s<n$) while retaining a suitably high accuracy in representing the original features. Therefore, there is no concept of path. A partial solution does not define any ordering among the components of the solution, and the next component to be selected is not necessarily influenced by the last component added to the partial solution [38, 39]. Furthermore, solutions to an FS problem are not necessarily of the same size. To apply an ACO algorithm to solve a feature selection problem, these aspects need to be addressed. The first problem is addressed by redefining the way that the representation graph is used.

4.1.1 Graph Representation

The feature selection problem may be reformulated into an ACO-suitable problem. The main idea of ACO is to model a problem as the search for a minimum cost path in a graph. Here nodes represent features, with the edges between them denoting the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion. Figure 2 illustrates this setup. Nodes are fully connected to allow any feature to be selected next. The ant is currently at node a and has a choice of which feature to add next to its path (dotted lines). It chooses feature b next based on the transition rule, then c and then d. Upon arrival at d, the current subset $\{a, b, c, d\}$ is determined to satisfy the traversal stopping criterion (e.g. suitably high verification quality has been achieved with this subset). The ant terminates its traversal and outputs this feature subset as a candidate for data reduction [8].

On the basis of this reformulation of the graph representation, the transition rules and pheromone update rules of standard ACO algorithms can be applied. In this case, pheromone and heuristic value are not associated with links. Instead, each feature has its own pheromone value and heuristic value.

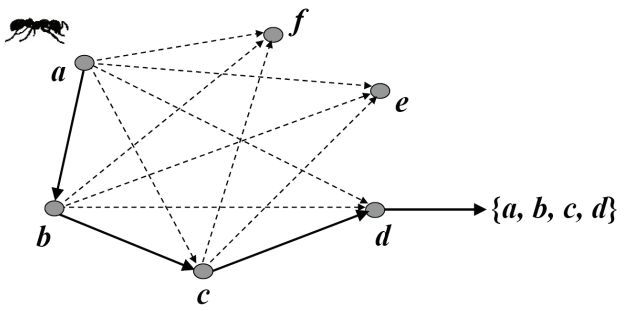

Figure 2. ACO problem representation for feature selection

4.1.2 Heuristic Desirability

The basic ingredient of any ACO algorithm is a constructive heuristic for probabilistically constructing solutions. A constructive heuristic assembles solutions as sequences of elements from the finite set of solution components. A solution construction starts with an empty partial solution. Then, at each construction step the current partial solution is extended by adding a feasible solution component from the set of solution components [33]. A suitable heuristic desirability of traversing between features could be any subset evaluation function for example, an entropy-based measure [5] or rough set dependency measure [40]. In proposed algorithm verification quality is mentioned as heuristic information for feature selection. The heuristic desirability of traversal and node pheromone levels are combined to form the so-called ***probabilistic transition rule*****,** denoting the probability that ant k will include feature i in its solution at time step t:

$$P_i^k(t) = \begin{cases} \frac{[\tau_i(t)]^\alpha . [\eta_i]^\beta}{\sum_{u \in J^k} [\tau_u(t)]^\alpha . [\eta_u]^\beta} & if\ i \in J^k \\ 0 & otherwise \end{cases} \quad (6)$$

where, J^k is the set of feasible features that can be added to the partial solution; τ_i and η_i are respectively the pheromone value and heuristic desirability associated with feature i. α and β are two parameters that determine the relative importance of the pheromone value and heuristic information.

The transition probability used by ACO is a balance between pheromone intensity (i.e. history of previous successful moves), τ_i, and heuristic information (expressing desirability of the move),

η_i. This effectively balances the exploitation-exploration trade-off. The best balance between exploitation and exploration is achieved through proper selection of the parameters α and β. If α=0, no pheromone information is used, i.e. previous search experience is neglected. The search then degrades to a stochastic greedy search. If β=0, the attractiveness (or potential benefit) of moves is neglected.

4.1.3 Pheromone Update Rule

After all ants have completed their solutions, pheromone evaporation on all nodes is triggered, and then according to equation (7) each ant k deposits a quantity of pheromone,$\Delta\tau_i^k(t)$, on each node that it has used.

$$\Delta\tau_i^k(t) = \begin{cases} \omega.\gamma(S^k(t)) + \varphi.(n/|S^k(t)|) & if\ i \in S^k(t) \\ 0 & otherwise \end{cases} \quad (7)$$

where $S^k(t)$ is the feature subset found by ant k at iteration t, and $|S^k(t)|$is its length. The pheromone is updated according to both the measure of the verification quality,$\gamma(S^k(t))$, and feature subset length. ω and φ are two parameters that control the relative weight of verification quality and feature subset length, $\omega \in$[0,1] and $\varphi = 1 - \omega$. This formula means that the verification quality and feature subset length have different significance for feature selection task. In our experiment we assume that verification quality is more important than subset length, so they were set as ω =0.7, φ=0.3.

In practice, the addition of new pheromone by ants and pheromone evaporation are implemented by the following rule applied to all the nodes:

$$\tau_i(t+1) = (1-\rho)\tau_i(t) + \Delta\tau_i^g(t) + \sum_{k=1}^{m} \Delta\tau_i^k(t) \quad (8)$$

where, m is the number of ants at each iteration and $\rho \in$(0,1) is the pheromone trail decay coefficient. The main role of pheromone evaporation is to avoid stagnation, that is, the situation in which all ants constructing the same solution. g indicates the best ant at each iteration. All ants can update the pheromone according to equation (8) and the best ant deposits additional pheromone on nodes of the best solution. This leads to the exploration of ants around the optimal solution in next iterations.

4.1.4 Solution Construction

The overall process of ACO feature selection can be seen in Figure 3. The process begins by generating a number of ants which are then placed randomly on the graph i.e. each ant starts with one random feature. Alternatively, the number of ants to place on the graph may be set equal to the number of features within the data; each ant starts path construction at a different feature. From these initial positions, they traverse nodes probabilistically until a traversal stopping criterion is satisfied. The resulting subsets are gathered and then evaluated. If an optimal subset has been found or the algorithm has executed a certain number of times, then the process halts and outputs the best feature subset encountered. If none of these conditions hold, then the pheromone is updated, a new set of ants are created and the process iterates once more.

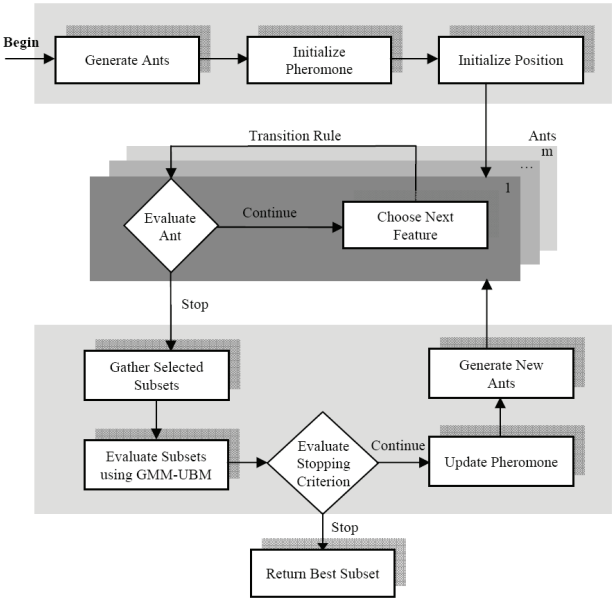

Figure 3. Overall process of ACO for feature selection in ASV

Typically an ASV system consists of several essential parts including feature extraction and feature selection. After preprocessing of speech signals, feature extraction is used to transform the input signals into a feature set (feature vector). Feature selection is applied to the feature set to reduce the dimensionality of it. ACO is used to explore the space of all subsets of given feature set. The performance of selected feature subsets is measured by invok-

ing an evaluation function with the corresponding reduced feature space and measuring the specified verification result. The best feature subset found is then output as the recommended set of features to be used in the actual design of the ASV system.

The main steps of proposed feature selection algorithm are as follows:

1. Initialization
   - Determine the population of ants.
   - Set the intensity of pheromone trial associated with any feature.
   - Determine the maximum of allowed iterations.
2. Solution generation and evaluation of ants
   - Assign any ant randomly to one feature and visiting features, each ant builds solutions completely.
   - In this step, the evaluation criterion is Equal Error Rate (EER). If an ant is not able to decrease the EER in ten successive steps, it will finish its work and exit.
3. Evaluation of the selected subsets
   - Sort selected subsets according to verification quality and their length. Then, select the best subset.
4. Check the stop criterion
   - Exit, if the number of iterations is more than the maximum allowed iteration, otherwise continue.
5. Pheromone updating
   - Decrease pheromone concentrations of nodes then, all ants deposit the quantity of pheromone on graph.
   - Finally, allow the best ant to deposit additional pheromone on nodes.
6. Generation of new ants
   - In this step previous ants are removed and new ants are generated.
7. Go to 2 and continue

5 Genetic Algorithms

The GAs are stochastic global search methods that mimic the metaphor of natural biological evolution [12]. These algorithms are general purpose optimization algorithms with a probabilistic component that provide a means to search poorly understood, irregular spaces. Instead of working with a single point, GAs work with a population of points. Each *point* is a vector in hyperspace representing one potential (or candidate) solution to the optimization problem. A population is, thus, just an ensemble or set of hyperspace vectors. Each vector is called a chromosome in the population. The number of elements in each vector (chromosome) depends on the number of parameters in the optimization problem and the way to represent the problem. How to represent the problem as a string of elements is one of the critical factors in successfully applying a GA (or other evolutionary algorithm) to a problem.

5.1 Genetic Algorithm for Feature Selection

Several approaches exist for using GAs for feature subset selection. The two main methods that have been widely used in the past are as follow. First is due to [13], of finding an optimal binary vector in which each bit corresponds to a feature (binary vector optimization (BVO) method). A ‘1’ or ‘0’ suggests that the feature is selected or dropped, respectively. The aim is to find the binary vector with the smallest number of 1’s such that the verification quality is maximized. This criterion is often modified to reduce the dimensionality of the feature vector at the same time [41]. The second and more refined technique uses an m-ary vector to assign weights to features instead of abruptly dropping or including them as in the binary case [42]. This gives a better search resolution in the multidimensional space [43].

6 Experimental Results

A series of experiments was conducted to show the utility of proposed feature selection algorithm. All experiments have been run on a machine with 3.0GHz CPU and 512 MB of RAM. We implement proposed ACO algorithm and GA-based feature selection algorithm in Matlab R2006a. The operating system was Windows XP Professional. The following sections describe TIMIT dataset and implementation results.

6.1 TIMIT Dataset

The TIMIT corpus [44] is used in this paper. This corpus contains 630 speakers (438 male and 192 female) representing 8 major dialect regions of the country-regionplaceUnited States, each speaking ten sentences. There are two sentences that are spoken by all speakers and the remaining eight are selected randomly from a large database.

The speech signal is recorded through a high quality microphone with a sampling frequency of 16 kHz in a quiet environment, with no session interval between recordings. Eight sentences (SX, SI) were used to develop each speaker model, and the remaining 2 SA sentences were used to test each speaker. The 40 speakers included in both the test and train directories were used during the TIMIT (40) trials.

6.2 Evaluation Measure

The evaluation of the speaker verification system is based on detection error tradeoff (DET) curves, which show the tradeoff between false alarm (FA) and false rejection (FR) errors. Typically equal error rate (EER), which is the point on the curve where *FA* = *FR*, is chosen as evaluation measure. We also used detection cost function (DCF) defined as [45]:

$$DCF = C_{miss}.FRR.P_{target} + C_{FA}.FAR.(1 - P_{target}) \quad (9)$$

where P_{target} is the priori probability of target tests with$P_{target} = 0.01$, *FRR* and *FAR*are false rejection rate and false acceptance rate respectively at an operating point and the specific cost factors $C_{miss} = 10$and$C_{FA} = 1$. Hence, the point of interest is shifted towards low *FA* rates.

6.3 Experimental Setup

Various values were tested for the parameters of proposed algorithm. The results show that the highest performance is achieved by setting the parameters to values shown in Table 1.

Experiments were conducted on a subset of TIMIT corpora consist of 24 male and 16 female speakers of different accent that were selected randomly. Data were processed in 20 ms frames (320 samples) with 50% overlaps. Frames were segmented by Hamming window and pre-emphasized with $\alpha = 0.97$to compensate the effect of microphone's low pass filter. At first, feature vector was created by extracting MFCCs from silence removed data for each frame. In the next step, delta coefficients were calculated based on the MFCCs and appended to existing feature vector. Furthermore, two energies were applied to input vectors as described earlier. Then we consider the LPCCs and their delta coefficients respectively and append them to the feature vector. The final feature set contains F = 50 features. Table 2 shows the overall set of features.

Finally, verification process was performed using the GMM-UBM approach. The performance criterion is due to EER and DCF according to an adopted decision threshold strategy.

6.4 Results and discussion

The feature subset length and verification quality are two criteria which are considered to assess the performance of algorithms. Comparing the first criterion, we noted that both ACO-based and GA-based algorithms reduce the dimensionality of feature space. Furthermore, the ACO-based algorithm selected a smaller subset of features than the GA-based algorithm. Table 3 shows the number of selected features by ACO-based and GA-based approaches. As we can see in Table 3, ACO can degrade dimensionality of features over 80%.

The second performance criterion is due to EER and DCF according to an adopted decision threshold strategy. The EER and DCF for GMM-UBM, GA-based and ACO-based algorithms with different number of Gaussian (16, 32 and 64) were shown in Table 4.

Ideally the number of mixtures, *M*, should ap-

	Population	Iteration	Crossover probability	Mutation probability	Initial pheromone	α	β	ρ
GA	30	50	0.7	0.005	-	-	-	-
ACO	30	50	-	-	1	1	0.1	0.2

Table 1. GA and ACO parameter settings

Feature Name	Order
Mel Frequency Cepstral Coefficient (MFCC)	12
Linear Prediction Cepstral Coefficient (LPCC)	12
First diff of MFCC ($\Delta - MFCC$)	12
First diff of LPCC($\Delta - LPCC$)	12
Energy	2
Total	50

Table 2. The overall feature set

Selection Method Selection Method	Number of Selected Features	Percentages of Selected Features
GA- GMM-UBM(16)	16	32%
GA-GMM-UBM(32)	17	34%
GA-GMM-UBM(64)	16	32%
ACO-GMM-UBM(16)	10	20%
ACO-GMM-UBM(32)	9	18%
ACO-GMM-UBM(64)	10	20%

Table 3. Selected features of GA and ACO

Number of Gaussians	GMM-UBM		GA		ACO	
	EER	DCF	EER	DCF	EER	DCF
16	5.08	0.0563	4.966	0.0554	4.318	0.0401
32	4.56	0.0505	3.679	0.0389	2.634	0.0309
64	6.667	0.0707	4.961	0.0532	4.23	0.0386

Table 4. EER and DCF for GMM-UBM, GA-based and ACO-based algorithms with different number of Gaussians

proximate the number of natural classes in data. If M is less than the number of natural classes, closely placed clusters will fuse to form larger clusters whose variance is higher. This results in a larger percentage of frames, both the true speaker and imposter, getting average scores. The performance goes down because these frames cannot be discriminated. This is a case of trying to under-fit training data. When M is more than the number of natural classes, larger clusters are broken up into smaller sub-clusters or spurious frames get represented as separate clusters. The new clusters will have lower variance. This would lead to increase in average and low scoring frames and hence would lead to lower performance. This is a case of trying to over-fit training data. As could be seen in Table 4, speaker verification performance is found to increase as M is increased but begins to drop after a point at which over-fitting starts taking place.

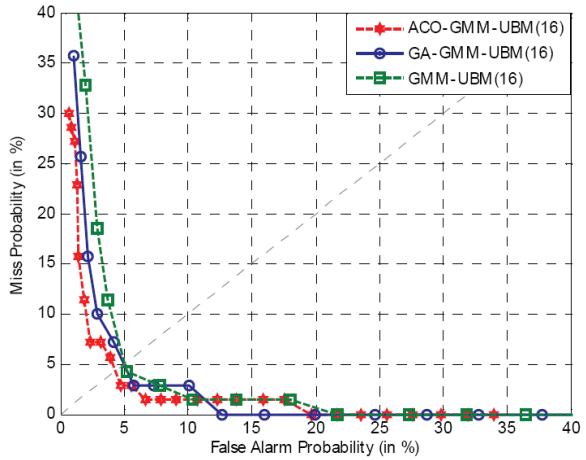

Figure 4. DET curves for CityplaceGMM-UBM, StateGA and ACO with 16 Gaussians

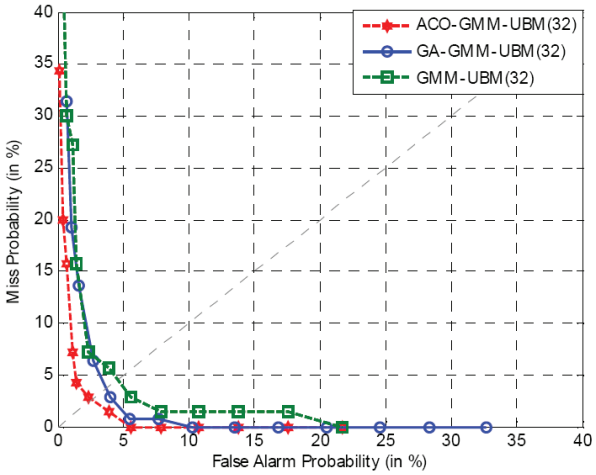

Figure 5. DET curves for CityplaceGMM-UBM, StateGA and ACO with 32 Gaussians

DET curves for GA-based and ACO-based algorithms with 16, 32 and 64 Gaussians are shown in Figure 4 through 6. From the results, it can be seen that ACO-GMM-UBM yields a significant improvement in speed than the baseline GMM-UBM approach. The improvement is due to the selection of optimal feature set by ACO algorithm.

To further highlight the search process, we graph percent selected features of every ant's current iteration (horizontal coordinate) against verification performance (vertical coordinate). Each point in the figure is an ant. The process of the ant colony searching for optimal solutions for TIMIT dataset is given in Figures 7(a) through 7(f).

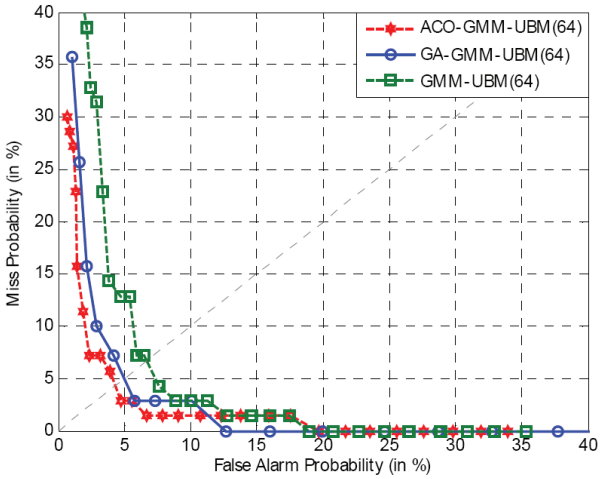

Figure 6. DET curves for CityplaceGMM-UBM, StateGA and ACO with 64 Gaussians

From the results and figures, we can see that, compared with GA, ACO is quicker in locating the optimal solution. In general, it can find the optimal solution within tens of iterations. If exhaustive search is used to find the optimal feature subset in the TIMIT dataset, there will be tens of billions of candidate subsets, which is impossible to execute. But with ACO, at the 50th iteration the optimal solution is found.

Ant colony optimization has powerful exploration ability; it is a gradual searching process that approaches optimal solutions. The running time of ACO is affected more by the problem dimension (feature numbers), and the size of data. For some datasets with more features, after finding a suboptimal solution, the GA cannot find a better one.

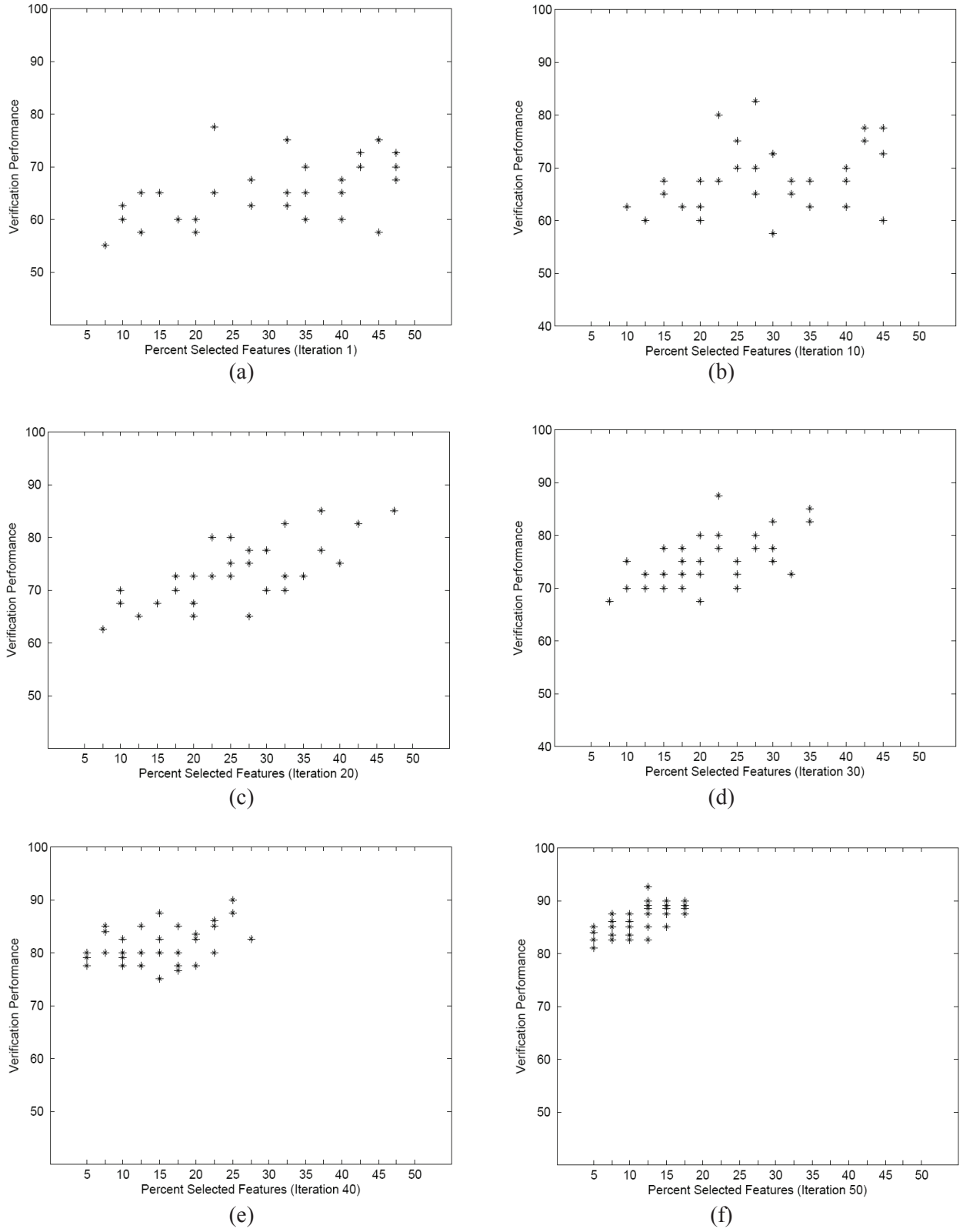

Figure 7. (a) Iteration 1 of ACO on TIMIT dataset; (b) Iteration 10 of ACO on TIMIT dataset; (c) Iteration 20 of ACO on TIMIT dataset; (d) Iteration 30 of ACO on TIMIT dataset; (e) Iteration 40 of ACO on TIMIT dataset; (f) Iteration 50 of ACO on TIMIT dataset

However, ACO can search in the feature space until the optimal solution is found. The GA is affected greatly by the number of features.

Ant colony optimization comprises a very simple concept, and the ideas can be implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. This optimization technique does not suffer, however, from some of the difficulties of GAs; interaction in the colony enhances rather than detracts from progress toward the solution. Further, an ant colony system has memory, which the genetic algorithm does not have. Changes in genetic populations result in the destruction of previous knowledge of the problem. In ant colony optimization, ants that past optima are tugged to return towards them; knowledge of good solutions is retained by all ants.

7 Conclusion

In this paper, we have addressed the problem of optimizing the acoustic feature set by ACO technique for text-independent speaker verification system based on GMM-UBM. Ant colony optimization selected the relevant features among all Mel-cepstrum coefficients in order to increase the performance of our ASV system. We compare its performance with another prominent population-based feature selection method, genetic algorithm. The experimental results on subset of TIMIT database showed that ACO is able to select the more informative features without loosing the performance than GA. The feature vectors size reduced over 80% which led to a less complexity of our ASV system. Moreover, verification process in the test phase speeds up because less complexity is achieved by the proposed system in comparison with current ASV systems.

Ant colony optimization has the ability to converge quickly; it has a strong search capability in the problem space and can efficiently find minimal feature subset. Experimental results demonstrate competitive performance. More experimentation and further investigation into this technique may be required. The pheromone trail decay coefficient (ρ) and pheromone amount ($\Delta\tau_i^k(t)$) have an important impact on the performance of ACO. The selection of the parameters may be problem-dependent. The deposited pheromone,$\Delta\tau_i^k(t)$, calculated using equation (7), expresses the quality of the corresponding solution. ρ simulates the pheromone evaporation. Evaporation becomes more important for more complex problems. If ρ=0, i.e. no evaporation, the algorithm does not converge. If pheromone evaporates too much (a large ρ is used), the algorithm often converged to sub-optimal solutions for complex problem. In many practical problems, it's difficult to select the best ρ without trial-and-error. α and β are also key factors in ACO for feature selection.

8 Acknowledgment

The authors wish to thank the Office of Graduate studies of the placePlaceTypeUniversity of PlaceNameIsfahan for their support.

References

[1] B. Xiang, and T. Berger, *Efficient Text-Independent Speaker Verification with Structural Gaussian Mixture Models and Neural Network*. IEEE Transactions on Speech and Audio Processing, 11(5), 2003.

[2] I. Lapidot, H. Guterman, and A. Cohen, *Unsupervised speaker recognition based on competition between self-organizing maps*. IEEE Transactions on Neural Networks, 13(2), 2002, 877–887.

[3] A. Martin, and M. Przybocki, *Speaker recognition in a multi-speaker environment*. Proc. of 7th European Conf. Speech Communication and Technology, CityplaceAalborg, countryregionDenmark, 2001, 787–790.

[4] P. Day, and A.K. Nandi, *Robust Text-Independent Speaker Verification Using Genetic Programming*. IEEE Transactions on Audio, Speech, and Language Processing, 15(1), 2007, 285-295.

[5] R. Jensen, *Combining rough and fuzzy sets for feature selection*. Ph.D. thesis, PlaceTypeplaceUniversity of PlaceNameEdinburgh, 2005.

[6] S. Nemati, R. Boostani, and M.D. Jazi, *A Novel Text-Independent Speaker Verification System using Ant Colony Optimization Algorithm*. Proc. Internat. Conf. on Image and Signal Processing ICISP2008, LNCS 5099, Springer-Verlag, Berlin Heidelberg, 2008, 421–429.

[7] H.R. Kanan, K. Faez, and M. Hosseinzadeh Aghdam, *Face Recognition System Using Ant Colony*

Optimization-Based Selected Features. Proc. of the First IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA), placecountry-regionUSA, 2007, 57-62.

[8] M. Hosseinzadeh Aghdam, N. Ghasem-aghaee, and M.E. Basiri, *Application of Ant Colony Optimization for Feature Selection in Text Categorization*. Proc. of the IEEE Congress on Evolutionary Computation, 2008.

[9] M. Pandit, and J. Kittkr, *Feature Selection for a DTW-Based Speaker Verification System*. Proc. of the 1998 IEEE Internat. Conf. on Acoustics, Speech and Signal Processing, CitySeattle, WA, placecountry-regionUSA, 1998, 769-772.

[10] A. Cohen, and Y. Zigel, *On Feature Selection for Speaker Verification*. Proc. of COST 275 workshop on The Advent of Biometrics on the Internet, 2002, 89–92.

[11] T. Ganchev, P. Zervas, N. Fakotakis, and G. Kokkinakis, *Benchmarking Feature Selection Techniques on the Speaker Verification Task, 2006*.

[12] M. Srinivas, and L.M. Patnik, *Genetic Algorithms: A Survey*. IEEE Computer Society Press, Los lamitos, 1994.

[13] A. Haydar, M. Demirekler, and M.K. Yurtseven, Feature selection using genetic algorithm and its application to speaker verification, Electron. Lett., vol. 34, no. 15,(July 1998), 1457–1459.

[14] M. Dorigo, and G.D. Caro, *Ant Colony Optimization: A New Meta-heuristic*. Proc. of the Congress on Evolutionary Computing, 1999.

[15] M. Dorigo, V. Maniezzo, and A. Colorni, *Ant System: Optimization by a colony of cooperating agents*. IEEE Transaction on Systems, Man, and Cybernetics-Part B, 26(1), 1996, 29–41.

[16] V. Maniezzo, and A. Colorni, *the Ant System Applied to the Quadratic Assignment Problem*. IEEE Transaction on Knowledge and Data Engineering, 11(5), 1999, 769–778.

[17] F. Bimbot, J.F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D.A. Reynolds, *A tutorial on text-independent speaker verification*. Eurasip Journal on Applied Signal Processing, 2004, 430-451.

[18] M.E. Basiri, N. Ghasem-Aghaee, and M.H. Aghdam, *Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins*. Proc. of 6th European Conf. on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, LNCS 4973, Springer-Verlag Berlin Heidelberg, 2008, 12–23.

[19] L. Cheung-chi, GMM-*Based Speaker Recognition for placeMobile Embedded Systems*. Ph.D. thesis, placePlaceTypeUniversity of PlaceName-Hong Kong, 2004.

[20] D.A. Reynolds, and R.C. Rose, *Robust text-independent speaker identification using Gaussian mixture speaker models*. IEEE Transactions on Speech and Audio Processing, 3(1), 1995, 72-83.

[21] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, *Speaker verification using adapted Gaussian mixture models*. Digital Signal Processing, 10, 2000, 19–41.

[22] D. Neiberg, *Text Independent speaker verification using adapted Gaussian mixture models*. Ph.D. thesis, Centre for Speech Technology (CTT) Department of Speech, Music and Hearing KTH, Stockholm, Sweden, 2001.

[23] Y. Linde, A. Buzo, and R. Gray, *An Algorithm for Vector Quantizer Design*. IEEE Transactions on Communications, 28(1), 1980, 84-95.

[24] J. Navratil, Q. Jin, W.D. Andrews, & J.P. Campbell, *Phonetic speaker recognition using maximum-likelihood binary-decision tree models*. Proc. IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), place-Hong Kong, 2003.

[25] V. Wan, *Speaker Verification Using Support Vector Machines*. Ph.D. thesis, University Sheffield, U.K, 2003.

[26] R. Wouhaybi, and M.A. Al-Alaou, *Comparison of neural networks for speaker recognition*. Proc. 6th IEEE Internat. Conf. on Electronics, Circuits Systems (ICECS), 1999, 125-128.

[27] D. Mladeni, *Feature Selection for Dimensionality Reduction*. Subspace, Latent Structure and Feature Selection, Statistical and Optimization, Perspectives Workshop, SLSFS 2005, Bohinj, Slovenia, LNCS 3940 Springer, 2006, 84–102.

[28] R.O. Duda, and P.E. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons, placeChichester, 1973.

[29] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, *Feature selection based on rough sets and particle swarm optimization*. Pattern Recognition Letters, 28(4), 2007, 459–471.

[30] A.A. Ani, *Ant Colony Optimization for Feature Subset Selection*. Transaction on Engineering, Computing and Technology, 4, 2005, 35–38.

[31] C.K. Zhang, and H. Hu, *Feature Selection Using the Hybrid of Ant Colony Optimization and Mutual Information for the Forecaster*. Proc. of the 4th Internat. Conf. on Machine Learning and Cybernetics, 2005, 1728–1732.

[32] J. Bins, *Feature Selection from Huge Feature Sets in the Context of Computer Vision*. Ph.D. thesis, Department Computer Science, PlaceNameplaceColorado PlaceTypeState PlaceTypeUniversity, 2000.

[33] M. Dorigo, and C. Blum, *Ant colony optimization theory: A survey*. Theoretical Computer Science, 2005, 243–278.

[34] M. Dorigo, *Optimization, Learning and Natural Algorithms*. Ph.D. thesis, Dipartimento di Electtronica, Politecnico di Milano. Italy, 1992.

[35] L.M. Gambardella, and M. Dorigo, *Ant-Q: A Reinforcement Learning Approach to the TSP*. Proc. of the Twelfth Internat. Conf. on Machine Learning, 1995, 252–260.

[36] T. Sttzle, and H.H. Hoos, *MAX-MIN Ant System and Local Search for the Traveling Salesman Problem*. Proc. of IEEE Internat. Conf. on Evolutionary Computation, 1997, 309–314.

[37] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati, *A new algorithm for a Dynamic Vehicle Routing Problem based on Ant Colony System*. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, Technical Report IDSIA-23-02, 2002.

[38] C. Blum, and M. Dorigo, *The hyper-cube framework for ant colony optimization*. IEEE Transaction on Systems, Man, and Cybernetics -Part B, 34(2), 2004, 1161–1172.

[39] G. Leguizamon, and Z. Michalewicz, *A New Version of Ant System for Subset Problems*. Proc. of IEEE Congress on Evolutionary Computation 1999.

[40] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishing, placeCityDordrecht, 1991.

[41] J. Yang, and V. Honavar, *Feature Subset Selection Using a Genetic Algorithm*. IEEE Intelligent Systems. 13, 1998, 44–49.

[42] W.F. Punch, E.D. Goodman, L.C.S.M. Pei, P. Hovland, and R. Enbody, *Further research on Feature Selection and Classification using Genetic Algorithms*. Proc. Internat. Conf. on Genetic Algorithms, 1993, 557–564.

[43] M. Raymer, W. Punch, E. Goodman, L. Kuhn, and A.K. Jain, *Dimensionality Reduction Using Genetic Algorithms*. IEEE Transactions on Evolutionary Computing, 4, 2000, 164–171.

[44] J. Garofolo, et al. DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. National PlaceTypeplaceInstitute of PlaceName-Standards and Technology, 1990.

[45] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, *the DET Curve in Assessment of Detection Task Performance*. Proc. Eurospeech, 4, 1997, 1895–1898.