An evolutionary confidence measurement for spoken term detection

Javier Tejedor[1], Alejandro Echeverría[2]
[1] Human Computer Technology Laboratory
[2] Machine Learning Group
Universidad Autónoma de Madrid
javier.tejedor@uam.es

Dong Wang
Multimedia Department, Eurecom
Sophia Antipolis, France
dong.wang@eurecom.fr

Abstract

We propose a new discriminative confidence measurement approach based on an evolution strategy for spoken term detection (STD). Our evolutionary algorithm, named evolutionary discriminant analysis (EDA), optimizes classification errors directly, which is a salient advantage compared with some conventional discriminative models which optimize objective functions based on certain class encoding, e.g. MLPs and SVMs. In addition, with the intrinsic randomness of the evolution strategy, EDA largely reduces the risk of converging to local minimums in model training. This is particularly valuable when the decision boundary is complex, which is the case when dealing with out-of-vocabulary (OOV) terms in STD. Experimental results on the meeting domain in English demonstrate considerable performance improvement with the EDA-based confidence for OOV terms compared with MLPs- and SVMs-based confidences; for in-vocabulary terms, however, no significant difference is observed with the three models. This confirms our conjecture that EDA exhibits more advantage for tasks with complex decision boundaries.

1. Introduction

The ever increasing volume of audio data available on the web or in huge multimedia repositories promotes the research on automatic indexing and retrieval methods for spoken documents. Spoken term detection (STD) is a fundamental task towards this direction [8], and was defined by the NIST as 'searching vast, heterogeneous audio archives for occurrences of spoken terms'. A multitude of research has been reported in this line [6, 12, 13, 7, 9].

The standard STD architecture, as depicted in Figure 1, consists of three main components: a speech recognition component that converts input speech to word or subword lattices; a term detector that searches the lattices for potential occurrences of search terms, and a decision maker which evaluates the found occurrences and hypothesizes reliable ones as output. In STD, a hypothesized occurrence is called a *detection*; if the detection corresponds to an actual occurrence, it is called a *hit*, otherwise it is a *false alarm (FA)*. If an actual occurrence is not detected, this is called a *miss*. The NIST tool is used to measure the STD performance in terms of average term weighted value (ATWV) and detection error tradeoff (DET) curves [8].

Within the STD subsystem, the decision maker plays an important role in determining eligible detections, which is usually based on certain confidence measures. Term-dependent confidence measures derived from discriminative models, e.g. multi-layer perceptron (MLP) or support vector machine (SVM), have been shown to outperform the commonly used lattice-based confidence [15]. Generally speaking, this discriminative approach treats the hit/FA decision as a two-class classification problem, and derives confidence measures from classification posterior probabilities. The classification posterior probabilities can be derived from any discriminative model, though MLPs and SVMs are the most commonly used. A possible disadvantage of the MLP and SVM, however, is that their cost functions are based on some intermediate distances instead of on the classification error rate itself. For example, MLPs take likehood on training data as their objective function, while SVMs maximize the minimum margin of training examples to the decision boundary. Another problem, mainly for MLPs, is that the training process heavily depends on initialization, and is much more likely to be trapped in a local minimum. This is particularly critical when the decision boundary is complex. For instance, when dealing with out-of-vocabulary (OOV) terms, the diverse properties (pronunciation variation, occurrence rate, confidence distribution, ASR error pattern) among OOV terms, compared with in-vocabulary (INV) terms, may lead to rather complicated decision boundary and hence to a high risk of local minimum.

We propose a new discriminative confidence estimation approach based on an evolutionary algorithm, named evolu-

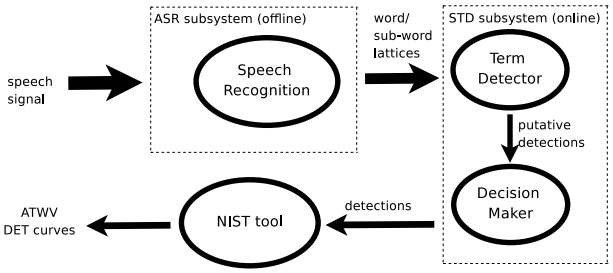

Figure 1. *The standard STD architecture: a speech recogniser converts speech into word/subword lattices; a term detector searches for potential occurrences of the search terms; a decision maker decides whether each detection is reliable. The NIST tool is used to evaluate detection performance.*

tionary discriminant analysis (EDA). Compared with MLPs and SVMs that optimize objective functions based on class encoding and intermediate distance, EDA takes the classification error rate as its objective function and therefore optimizes the evaluation metric directly. Moreover, the intrinsic randomness within the evolution strategy provides a rescue mechanism for models trapped in local minima. We argue that these advantages with EDA may lead to a better discriminative confidence estimation than standard MLPs and SVMs in STD, especially for OOV terms for which the decision boundary is complex. The authors note that EDA has been already used in other applications [11, 10]. The novelty of this paper from the EDA perspective is that EDA is extended to provide classification posterior probabilities instead of hard classification decisions. To our best knowledge, this is the first effort to apply evolutionary approaches in STD.

The rest of the paper is organized as follows: we first introduce the discriminative confidence estimation in Section 2, and then present the evolutionary algorithm in Section 3. The experimental settings and results are presented in Section 4, and some conclusions in Section 5.

2. Discriminative confidence estimation

The discriminative confidence measurement was presented in [15] to deal with the highly diverse properties among OOV terms. The basic idea is to treat the hit/FA decision as a binary classification task and derive confidence measures from the classification posterior probabilities. Any discriminative model can be employed to derive the posterior probabilities, such as MLPs and SVMs studied in the original paper [15]. A particular advantage along with the discriminative approach is that term-dependent factors can be involved in model inputs and hence being taken into account in measuring confidence scores. As it has been demonstrated [15], the term-dependent discriminative confidence estimation is highly effective and substantially outperforms the widely used lattice-based confidence, especially for OOV terms.

Following the notations in [15], we denote a detection as d, and its discriminative confidence as $c_p(d)$. The discriminative approach can be formally represented as a non-linear mapping f from a set of informative features to $c_p(d)$:

$$f : (c_f(d), A, L, T, R_0(K), R_1(K)) \longrightarrow c_p(d) \quad (1)$$

where $c_f(d)$ is the lattice-based confidence (i.e., c_{lat}) derived from lattice posterior probabilities [16], given by

$$c_{lat} = \frac{\sum_{\pi_\alpha,\pi_\beta} p(O|\pi_\alpha, K_{t_s}^{t_e}, \pi_\beta) P(\pi_\alpha, K_{t_s}^{t_e}, \pi_\beta)}{\sum_\xi p(O|\xi) P(\xi)} \quad (2)$$

where π_α and π_β denote any path before and after the term detected K, with π_α starting from the beginning of the speech and π_β finishing at the end; ξ denotes any complete path through the lattice. t_s is the start time of the term detected K, t_e is the end time of the term detected K and O represents the speech. The rest of input features include the acoustic likelihood (A), the language model score (L), the duration of the detection T, and two term-dependent features $R_0(K)$ and $R_1(K)$ defined as follows:

$$R_0(K) = \frac{\sum_i c_f(d_i^K)}{T_0} \quad (3)$$

and

$$R_1(K) = \frac{\sum_i (1 - c_f(d_i^K))}{T_0} \quad (4)$$

where $c_f(d_i^K)$ represents the lattice-based confidence of the i-detection of the term K, and T_0 is the length of the audio. Note that R_0 and R_1 are designed to introduce term-dependency (occurrence rates here) in the modeling, and are motivated by the definition of the evaluation metric ATWV [8]. The mapping function f can be implemented as any discriminative model, e.g. an MLP or an SVM studied in [15], or an EDA presented in the next section.

3. Evolutionary Discriminant Analysis for confidence estimation

A potential problem of MLPs and SVMs is that they are not optimized with respect to the evaluation metric, i.e., classification error rate. For MLPs, the objective is maximum likelihood while for SVMs the objective is maximum margin. An ideal approach, of course, is to optimize the classification error rate *directly*, i.e.,

$$\hat{\theta} = arg\min_{\theta} \sum_{d} \delta\{H_{g_\theta}(d), t(d)\} \quad (5)$$

where d is a training exemplar, and $t(d)$ is its class label; g_θ is a projection function depending on parameters θ, H_{g_θ} is a classification function in the projected space determined by g_θ, and $\delta(a, b)$ is an indication function which is equal to 0 if $a = b$ and equal to 1 otherwise. The main obstacle with such an objective, however, is that the objective function will never be continuous, failing the conventional gradient-based training approach. A possible solution is to exploit the evolution strategy [1] to 'breed' some solutions and then choose the optimum. Specifically, an evolution strategy maintains a group of possible solutions (called *chromosomes*) and allows them to evolve in a random way, called *evolution*. The evolution process in fact implements a random-directed search and can hence be used to optimize non-linear discrete functions. This leads to the evolutionary discriminant analysis (EDA) [10, 11].

Compared with MLPs and SVMs, EDA possesses several advantages. First, the classification error minimization suggests better performance in classification tasks; second, the evolution strategy reduces the risk of local minimum in model training. Finally, the continuous projection function g and discrete classification function H can be chosen freely, leading to a highly flexible decision strategy. In this paper we make use of EDA for STD in decision making. Different from previous studies where EDA is used to predict class labels, we extend EDA in this paper to derive classification posterior probabilities which are then used as discriminative confidence in decision making.

In this section, we first present how the classification task is cast to an evolutionary treatment using a certain coding scheme, and then describe how to represent EDA errors with a fitness function. This is followed by a presentation of the evolution procedure and the extension to a posterior probability estimation.

3.1. Coding scheme

In the EDA objective function represented in Equation (5), the projection g_θ can be chosen freely. In our implementation, we take advantage of an MLP's non-linear approximation and reuse the MLP structure to implement the projection function. It must be emphasized, however, that the MLP-alike EDA is fundamentally different from an MLP: the layer structure in EDA is just a projection and the output does not necessarily relate to classification posterior probabilities; the objective function is never maximum likelihood but minimum classification errors; and the optimization is not based on gradient but evolution.

We start from a 3-layer MLP structure (one hidden layer) that consists of $e + 1$ input units, $h + 1$ hidden units and n output units. The notation $+1$ here denotes the bias unit in both the input and hidden layers. Note that n is not necessarily equal to the number of classes as in the case of an MLP. We denote the weights of the first layer as w_{ij}, where $i = 0, \ldots, e$ and $j = 1, \ldots, h$, and the weights of the second layer as v_{jk}, where $j = 0, \ldots, h$ and $k = 1, \ldots, n$. A sigmoidal active function $\varphi(z) = 1/(1 + \exp^{-z})$ is applied to the output of the hidden units. The k-th output of the function represented by this MLP is then formulated as follows:

$$y_k(d) = \sum_{j=0}^{h} v_{jk}\varphi(\sum_{i=0}^{e} d_i w_{ij}). \quad (6)$$

From the EDA perspective, the weights of the MLP comprise the EDA model parameters θ, and need to be optimized using the evolutionary approach. From the perspective of evolutionary computing, the weights are alleles that comprise chromosomes. We choose a simple way to map the parameters θ in EDA to the chromosomes in evolution, where all the alleles are listed one by one as follows:

$$\theta = (\ldots, w_{i1}, w_{i2}, \ldots, w_{ih}, \ldots, v_{j1}, v_{j2}, \ldots, v_{jn}, \ldots). \quad (7)$$

where $i = 0, \ldots, e$, and $j = 0, \ldots, h$.

3.2. Fitness function

In order to minimize classification errors, the fitness value assigned to a chromosome in Equation (7) should be the number of detections misclassified with the EDA whose parameters correspond to this chromosome. The classification function H is implemented by first projecting a detection d to the n-dimensional space represented by the output of the MLP-based projection function, and then assigning it to the class whose projected mean is closer to the projection of d. This n-dimensional projection algorithm is sketched as follows:

- The mean components of the training set (m_r^i), where $i = 1, \ldots, e$ and $r \in [FA, hit]$ are projected to the n-dimensional space by g_θ where θ represents the parameters corresponding to the current chromosome:

$$\hat{m}_r^k = g_\theta(m_r) = \sum_{j=0}^{h} v_{jk}\varphi(\sum_{i=0}^{e} m_r^i w_{ij}) \qquad k = 1, \ldots, n. \quad (8)$$

- Each detection d in the training data is projected accordingly:

$$\hat{d}_k = g_\theta(d) = \sum_{j=0}^{h} v_{jk}\varphi(\sum_{i=0}^{e} d_i w_{ij}) \qquad k = 1, \ldots, n. \tag{9}$$

- Each detection d is assigned to the class of the closest training mean in the projected space:

$$H_{g_\theta}(d) = \arg \min_{r \in [FA, hit]} \sum_{k=1}^{n} (\hat{d}_k - \hat{m}_r^k)^2 \tag{10}$$

- The classification error rate is assigned to the chromosome corresponding to θ as its fitness:

$$J(\theta) = \frac{1}{N} \sum_{d} \delta(t(d), H_{g_\theta}(d)) \tag{11}$$

where N is the number of detections.

To avoid over-fitting, we divide the training data into a training set and a validation set. The training set is used to calculate the class means m_r with $r \in [FA, hit]$, and the projected class means $\hat{m}_r$ will be used in the classification function H to compute the errors on both the training and validation sets. Then, the errors on the training ($J_{tr}(\theta)$) and validation sets ($J_{va}(\theta)$) are summed into a single figure $J(\theta) = J_{tr}(\theta) + J_{va}(\theta)$ to compose the fitness value.

3.3. The evolution algorithm

We have mapped the EDA parameters to chromosomes and the EDA error function to the fitness function in the evolutionary approach. The evolutionary process that searches for the optimal chromosome will then correspond to EDA parameter optimization. This leads to the EDA algorithm given as follows:

- Initialize an MLP structure.
- Choose an evolution strategy $(\mu, \lambda|\rho)$, where μ is the size of the parent population (group of solutions in the current step of the algorithm), λ is the size of the offspring population (group of solutions obtained from the parent population by means of evolutionary operators), and ρ is the size of the family (parents whose recombination leads to offsprings).
- Choose a mutation step σ, which is the noise level on alleles when reproducing a new generation.
- Generate the initial population by creating μ MLPs whose weights are randomly selected in $[-0.5, 0.5]$.
- The following steps are repeated until a prescribed number of generations, within which no fitness improvement is got, is reached:
  - Generate λ offspring MLPs by recombining ρ networks randomly selected from the current generation. These ρ networks form a parent set, and each allele (weight of the MLP) of the offspring is inherited from one of its parents that is selected randomly.
  - Mutate the generated offsprings by adding independent random noise that is drawn from a normal distribution $N(0, \sigma)$ to the alleles.
  - Form a new generation by selecting μ best individuals among the λ offsprings. This is called *Comma replacement* in the evolution strategy field and is generally recommended for optimizing continuous parameters [1].

3.4. Confidence measure

We now employ the trained EDA model to predict discriminative confidence for STD, or posterior probabilities of a detection belonging to hit or FA classes. This can be achieved by measuring the relative distance of the projected detection $\hat{d}$ to the projected mean of hit and FA classes, i.e., $\hat{m}_{hit}$ and $\hat{m}_{FA}$ respectively. Figure 2 shows the approach: first draw a vector between projected class means ($\overrightarrow{\hat{m}_{FA}\hat{m}_{hit}}$), and then project the detection image $\hat{d}$ on to the vector, obtaining the new image $\ddot{d}$. The posterior probability of d belonging to the hit class, or the confidence of d, is then derived by

$$c_p(d) = \begin{cases} 0 & \text{if } \ddot{d} \leq \ddot{m}_{FA} \\ \frac{\ddot{d} - \ddot{m}_{FA}}{\ddot{m}_{hit} - \ddot{m}_{FA}} & \text{if } \ddot{m}_{FA} < \ddot{d} < \ddot{m}_{hit} \\ 1 & \text{if } \ddot{m}_{hit} \leq \ddot{d} \end{cases} \tag{12}$$

where $\ddot{d}$, $\ddot{m}_{hit}$ and $\ddot{m}_{FA}$ are the projections of $\hat{d}$, $\hat{m}_{hit}$ and $\hat{m}_{FA}$ in the space $\overrightarrow{\hat{m}_{FA}\hat{m}_{hit}}$ respectively.

4. Experiments

Experiments were conducted on the English meeting domain, recorded using individual headset microphones (IHM). The same training speech and text data used for building the AMI RT05s LVCSR system [5] were used to train the acoustic models (AM) and the language model (LM). The NIST RT04s dev set was used for parameter tuning, and the evaluation corpus comprised three sub-sets: the

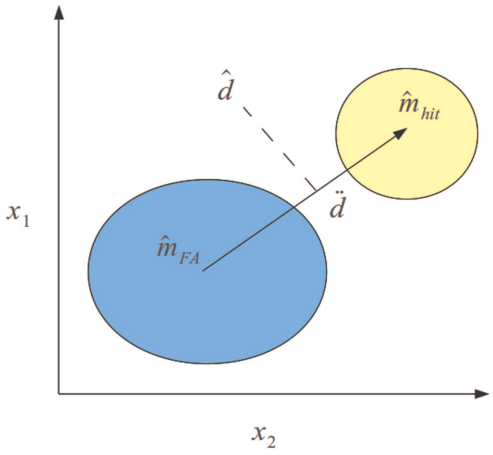

Figure 2. *EDA output space with two dimensions (x_1 and x_2), and projected class means $\hat{m}_{FA}$ and $\hat{m}_{hit}$. A projected detection $\hat{d}$ is again projected (to derive $\ddot{d}$) over the vector drawn between projected class means $\hat{m}_{FA}$ and $\hat{m}_{hit}$, to further compute the confidence measure.*

NIST RT04s and RT05s eval sets, and a new meeting corpus recorded at the University of Edinburgh as part of the AMIDA project (*http://www.amiproject.org/*).

We first selected 256 terms from the AMI dictionary as INV terms, which have 2329 occurrences in the evaluation data. Then we compared the AMI dictionary (in active use and assumed to represent current usage) and the COMLEX Syntax dictionary v3.1 (published by LDC in 1996 and therefore historical from an STD perspective), and selected 412 terms as OOV terms from the AMI dictionary that do not occur in the COMLEX dictionary. These terms simulate the evolution of novel terms over time. Additionally, we selected 70 *artificial OOV terms* that have more occurrences and are plausible as search terms. In total we have 482 OOV terms and 2736 occurrences in the evaluation data. To ensure the OOV terms in the experiment represent truly novel terms, we purged all of them from the training speech and text.

We built a phoneme-based system. It used state-clustered triphone models, 39-dimensional MFCC features and a 6-gram phoneme LM. Cambridge University's HTK was used to train the acoustic models and perform lattice generation, and the SRI LM toolkit was used to train the LM. An enhanced Joint-Multigram model [4] trained with the AMI dictionary was applied to predict pronunciations for the OOV terms. The *Lattice2Multigram* tool from *Speech@FIT* (Brno University) was used to hypothesize the detections from the phoneme lattices. More information about these setups can be found in [14].

We trained an MLP, an SVM and an EDA to estimate the discriminative confidence. STD experiments were first conducted on the development set, and then detections were collected with hits and false alarms labelled, which were employed to train the MLP, SVM and the EDA. A 3-layer MLP, whose structure is comprised of an input layer with 6 inputs according to Equation 1, a hidden layer with a sigmoid activation and an output layer with a soft-max activation which contains 2 output units according to hit/FA classification, was trained using the standard error back-propagation algorithm [2]. The number of hidden units, which was chosen to minimize the number of classification errors on the development set by cross-validation, is 30 in our experiments. The SVM was trained with the LIBSVM toolkit [3] with a radial basis kernel function.The parameters, including the error penalty C for classification and the radius scale γ for the kernel, were again optimized by cross-validation, giving $C = 32$ and $\gamma = 0.5$ in our experiments. Both MLP and SVM were used as complete hit/FA classifiers, as in the previous work [15].

For EDA, the MLP structure was chosen to minimize the fitness value ($h = 12$ for the hidden units and $n = 5$ for the output units). For the number of input units, $e = 6$ according to Equation 1. We pragmatically chose the evolution strategy $(\mu = 15, \lambda = 100|\rho = 2)$, and the mutation step σ is fixed to 0.15. The evolution process stops when there is not a fitness improvement within the last 100 generations.

4.1. Results and discussion

The experimental results are shown in Table 1 in terms of ATWV [8]. We observe that the EDA outperforms the MLP- and SVM-based discriminative confidence for both INV and OOV terms. Paired t-tests show that this improvement is statistically significant ($p < 0.001$) for OOV terms compared with the SVM and weakly significant ($p < 0.09$) compared with the MLP. For INV terms, the improvement achieved by EDA is insignificant compared with the MLP ($p \approx 0.4$) and hardly significant compared with the SVM ($p \approx 0.1$). DET curves in Figure 3 show that EDA outperforms both MLP and SVM considerably for OOV terms when the FA is low. For INV terms, EDA does not show obvious advantage over the other two models.

The experimental results suggest that an evolution strategy is more powerful than MLPs and SVMs when dealing with the OOV terms. This can be explained by the high diversity in terms of ASR error pattern, occurrence rate and confidence distribution of OOV terms. This diversity introduces comprehensive decision boundaries for classification, leading to a high risk of converging to a local minimum. The EDA approach, with the classification error rate as its objective function and the evolution strategy as its rescue mechanism, is able to ameliorate the local minimum prob-

	ATWV	
Confidence estimator	INV terms	OOV terms
MLP	0.5466	0.2952
SVM	0.5434	0.2920
EDA	**0.5500**	**0.2994**

Table 1. *STD system performance with discriminative confidence estimated by the MLP, SVM and EDA for INV and OOV terms.*

lem and therefore provides with a better treatment for the OOV diversity.

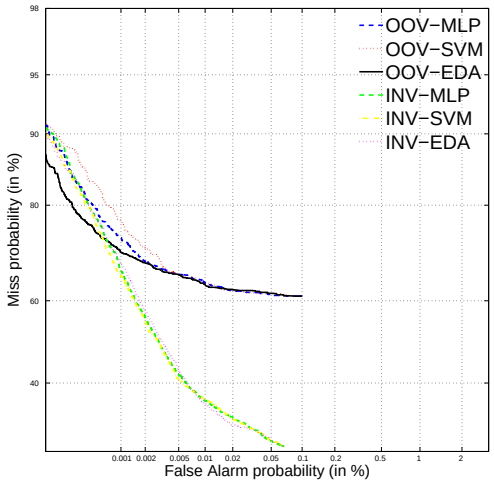

Figure 3. *DET curves of the STD system with the MLP, SVM and EDA confidence estimators for INV and OOV terms.*

5. Conclusions

This paper has proposed an evolutionary algorithm for confidence estimation in an STD system. We have shown the potential of the evolutionary algorithms in confidence estimation for putative detections. For INV terms our evolution strategy provides with similar performance to that of other classification techniques such as MLP and SVM. For OOV terms, significant performance improvement is obtained, confirming our conjecture that the evolutionary approach is powerful in handling complex decision boundaries introduced by the high diversity of OOV terms.

Future work will investigate new features for EDA and other classifiers to enhance the STD performance. In addition, new fitness functions will be investigated, particularly the one that optimizes the ATWV metric directly.

References

[1] H.-G. Beyer and H.-P. Schwefel. Evolution strategies - a comprehensive introduction. *Journal of Natural Computing*, 1(1):3–52, March 2002.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: A library for support vector machines*, 2001.

[4] S. Deligne, F. Yvon, and F. Bimbot. Variable-length sequence matching for phonetic transcription using joint multigrams. In *Proc. Eurospeech'95*, pages 2243–2246, September 1995.

[5] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, I. McCowan, D. Moore, V. Wan, R. Ordelman, and S. Renals. The 2005 AMI system for the transcription of speech in meetings. In *Machine Learning for Multimodal Interaction*, volume 3869/2006, pages 450–462. Springer Berlin /Heidelberg, 2006.

[6] J. Mamou, B. Ramabhadran, and O. Siohan. Vocabulary independent spoken term detection. In *Proc. ACM-SIGIR'07*, pages 615–622, Amsterdam, The Netherlands, July 2007.

[7] S. Meng, P. Yu, J. Liu, and F. Seide. Fusing multiple systems into a compact lattice index for Chinese spoken term detection. In *Proc. ICASSP'08*, pages 4345–4348, March 2008.

[8] NIST. *The spoken term detection (STD) 2006 evaluation plan*. National Institute of Standards and Technology (NIST), 10 edition, September 2006.

[9] C. Parada, A. Sethy, and B. Ramabhadran. Balancing false alarms and hits in spoken term detection. In *Proc. ICASSP'10*, volume 1, pages 5286–5289, March 2010.

[10] A. Sierra and A. Echeverría. Neural networks trained by distance to means. *WSEAS Transactions on Information Science and Applications*, 2(9):1446–1453, September 2005.

[11] A. Sierra and A. Echeverría. Evolutionary discriminant analysis. *IEEE Transactions on Evolutionary Computation*, 10(1):81–92, February 2006.

[12] I. Szöke, M. Fapšo, M. Karafiát, L. Burget, F. Grézl, P. Schwarz, O. Glembek, P. Matějka, S. Kontár, and J. Černocký. BUT system for NIST STD 2006 - English. In *Proc. NIST Spoken Term Detection Evaluation workshop (STD'06)*. NIST, December 2006.

[13] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang. The SRI/OGI 2006 spoken term detection system. In *Proc. Interspeech'07*, pages 2393–2396, August 2007.

[14] D. Wang. *Out-of-vocabulary spoken term detection*. PhD thesis, The Center for Speech Technology Research, Edinburgh University, December 2009.

[15] D. Wang, S. King, J. Frankel, and P. Bell. Term-dependent confidence for out-of-vocabulary term detection. In *Proc. Interspeech'09*, pages 2139–2142, September 2009.

[16] F. Wessel, K. Macherey, and R. Schlüter. Using word probabilities as confidence measures. In *Proc. ICASSP'98*, volume 1, pages 225–228, May 1998.