GRAPH RELATIONAL FEATURES FOR SPEAKER RECOGNITION AND MINING

Zahi N. Karam[1,2], William M. Campbell[2], Najim Dehak[3]

[1]DSPG, RLE at MIT, Cambridge, MA , [2]MIT Lincoln Laboratory, Lexington, MA , [3]CSAIL at MIT, Cambridge, MA
zahi@mit.edu, wcampbell@ll.mit.edu, najim@csail.mit.edu

ABSTRACT

Recent advances in the field of speaker recognition have resulted in highly efficient speaker comparison algorithms [1] [2]. The advent of these algorithms allows for leveraging a background set, consisting a large numbers of unlabeled recordings, to improve recognition [3] [4]. In this work, a relational graph, where nodes represent utterances and links represent speaker similarity, is created from the background recordings in which the recordings of interest, train and test, are then embedded. Relational features computed from the embedding are then used to obtain a match score between the recordings of interest. We show the efficacy of these features in speaker verification and speaker mining tasks.

***Index Terms*—** Speaker Recognition, Speaker Mining, Relational Features, Graph Embedding

1. INTRODUCTION

Text-independent speaker comparison is the process of providing a match score between two speech recordings, we refer to the pair of speech recordings as a trial. In the past, the match score has been computed either by considering the two recordings in isolation, as in [1] [2], or using an impostor set of recordings and one of the two recordings to train a support vector machine (SVM) classifier and then test on the other [5]. Small sets of impostor recordings have also been used in score normalization [6] to calibrate scores across trials.

Recently, non-SVM algorithms that fall within the inner product discriminant functions (IPDFs) [2] or total variability (TV)[1] frameworks have provided state of the art comparison performance that is fast and efficient. In previous work we leveraged this efficiency to score each of the trial utterances against a large number, several thousands, of unlabeled background utterances. In [3] we used the scores between the trial and background utterances to embed the trial utterances as nodes in a graph, whose links correspond to speaker similarity and whose nodes are the recordings, and the distance along the graph between two utterances of a trial was used as a match score. In [4] the background set consisted of only impostor recordings and the scores were used to reduce false alarms by performing adaptive score normalization.

Motivated by the link prediction problem [7] this work embeds the trial recordings along with the background set in a graph and, in addition to using the shortest path as a match score, extracts several other features that capture the interconnection between the trial utterances and the background. This results in each trial being represented by a set of these graph relational features which can then be used with a classifier, e.g. linear SVM, that is trained to separate between true trials, where the trial recordings correspond to the same speaker, and false ones. The classifier is then used to classify whether an unseen pair of recordings corresponds to a true or false trial.

We will begin with a description of the total variability system which we will use both as a baseline and for graph construction. We then discuss the graph construction and embedding, followed by the relational features we'll extract from the graph. Next we present the classifier used along with the train and test setup. We conclude with results that show the efficacy of these features and suggestions for future work.

This work was sponsored by the Federal Bureau of Investigation under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions,and recommendations are those of the authors and are not necessarily endorsed by the United States Government. This work was also supported in part by the Texas Instruments Leadership University Program.

2. TOTAL VARIABILITY (TV)

The baseline system used in this work is the total variability (TV) system proposed in [1], a brief description is provided in this section, extended details may be found in the original work. The system begins by adapting the means of a universal background model (UBM), a Gaussian mixture model (GMM) representing the speech of the general population, to a specific utterance, utt. The vector of stacked means of the adapted GMM, m_{utt}, is called a GMM supervector and is used as a sufficient statistic representing the recording. The TV system is based on the TV space which is a linear subspace of the GMM supervector space that includes all variability, speaker and nuisance, observed in the GMM supervectors of the training data-set. The subspace is chosen such that for a given speech utterance its corresponding GMM supervector (m_{utt}) can be represented as

$$m_{utt} = m_{UBM} + T_{tel} w_{utt} \tag{1}$$

where m_{UBM} is the UBM mean supervector, T_{tel} is the low-rank matrix defining the TV subspace, and w_{utt} is the corresponding factor of the utterance in the space. In this work T_{tel} is trained on telephony data such that the w's are normally distributed with zero mean and unit variance.

In the TV space, the match score ($s(utt_a, utt_b)$) between two utterances utt_a and utt_b is computed as a weighted inner-product where the weighting effectively performs channel compensation:

$$s(utt_a, utt_b) = \frac{w^t_{utt_a} A W^{-1} A^t w_{utt_b}}{\sqrt{w^t_{utt_a} A W^{-1} A^t w_{utt_a}}\sqrt{w^t_{utt_b} A W^{-1} A^t w_{utt_b}}}.$$

A corresponds to a linear discriminant analysis (LDA) projection matrix, trained to project into a space that captures inter-speaker variability while avoiding within speaker variability, and W is the within speaker covariance matrix computed in the LDA space. It is of importance to note the symmetry in the scoring function and that due to the normalization, in the denominator, $s(utt_a, utt_a) = 1$.

The system operates on cepstral features, extracted using a 25 ms Hamming window. 19 Mel frequency cepstral coefficients together with log energy are calculated every 10 ms. Delta and double delta coefficients were then calculated using a 5 frame window to produce 60-dimensional feature vectors. This 60-dimensional feature vector was subjected to feature warping using a 3 second sliding window. The UBMs used are gender dependent Gaussian mixture models containing 2048 Gaussians. The UBM and the LDA projection are trained on data from the Switchboard II, Switchboard cellular, and telephone utterances from the 2004/05/06 NIST SRE [8]. The TV subspace is trained on the these corpora as well as the Fisher English corpus. The WCCN matrix is computed using only the telephone utterances from the 2004/05/06 NIST SRE data sets [8].

2.1. Symmetric Normalization (SNORM)

It is common for speaker verification systems to be followed by a score normalization technique, the goal of which is to reduce within trial variability leading to improved performance, better calibration, and more reliable threshold setting. In this work the baseline used consists of the TV system followed by symmetric score normalization (SNorm) [6]. For every score $s(utt_a, utt_b)$ between two utterances, the corresponding SNorm score $\hat{s}(utt_a, utt_b)$ is

$$\hat{s}(utt_a, utt_b) = \frac{s(utt_a, utt_b) - \mu_a}{\sigma_a} + \frac{s(utt_a, utt_b) - \mu_b}{\sigma_b} \qquad (2)$$

where μ_i and σ_i are the mean and standard deviation of the scores of utt_i scored against an impostor list. Gender dependent impostor lists are used that consist of 614 female and 406 male English telephone utterances drawn from the 2005/06 NIST SRE data-sets.

3. GRAPH EMBEDDING

The symmetric scoring function $s(utt_a, utt_b)$ in Section 2 is used to compute a pair-wise match score between each two recordings in the set consisting of the background and trial utterances, resulting in a square and symmetric match-score matrix. The score matrix encodes not only the direct comparison between the trial utterances but also how they interact with the background set. This information can be leveraged to improve on the direct match score. Motivated by the link prediction problem [7] we generate a relational graph that summarizes the score matrix and extracts relational features from the graph. These features in addition to the direct match score can be combined to label a trial as true or false.

The relational graph consists of nodes representing recordings and undirected edges between the nodes. Two nodes are connected by an edge if they are "similar" enough. This can be decided in one of two ways: the first connects two nodes if their pair-wise match score is above a set threshold ϵ, the second includes an edge between two nodes if one is among the K-nearest neighbors (KNNs) of the other. The KNNs of a particular node is the set of K recordings whose pair-wise score with the recording is the highest. The choice of which graph construction method and the parameters K and ϵ will result in very different graphs. These differences allow us to examine the match-score matrix from different perspectives which we speculate would yield somewhat complementary graph relational features. We therefore include both construction methods and several parameter choices in the feature extraction process.

Another choice in graph construction is whether the edges of the graph are weighted or not. Weighted graphs use the pair-wise score between two recordings for the weight of the edge connecting them. Binary graphs on the other hand have all their edge weights set to unity, therefore all the information is encoded in whether an edge exists between two nodes or not. In the next section, we propose several relational features, some applicable to both binary and weighted graphs, others to only one.

4. GRAPH RELATIONAL FEATURES

Once the trial and background utterances are embedded in a graph we can extract several features that capture the interaction between the trial utterances via the graph. These features are split into two main classes: those that examine only the immediate neighborhood of the trial utterances and those that extend beyond that. To simplify the presentation of the features we first present some notation:

- The nodes in the graph, representing trial and background utterances, are indexed from 1 to M, where M is the total number of nodes in the graph.
- Each trial consists of an enrollment and test utterance E and T respectively.
- N_x is the set of neighbors of node x, i.e. the nodes connected to x by an edge, e.g. N_E is the set of neighbors of E.
- $|X|$ is the cardinality of the set X .
- $||x||$ is the 2-norm of the vector x .
- The vector v_x is a typically sparse vector, of size Mx1, that captures the interaction of x with the remaining graph nodes:
  - Zero valued entries in the vector indicate the lack of an edge between the utterance x and the nodes corresponding to the zero locations.
  - For weighted graphs, the value of the non-zero vector entries indicates the weight of the edge between x and the corresponding graph node.
  - For binary graphs, all non-zero entries have a value of one and indicate edges between x and the graph nodes.

4.1. Neighborhood Features

The premise of these features is that if E and T are utterances of the same speaker then their match scores with the background utterances should be similar indicating they lie within the same neighborhood of the graph.

4.1.1. Binary graph

We adopt the following features from [7] where they are used for link prediction:

- *Common neighbors*=$|N_E \cap N_T|$ counts the number of common neighbors between E and T.
- *Jaccard's coefficient*=$\frac{|N_E \cap N_T|}{|N_E \cup N_T|}$ normalizes the common neighbor score by the total number of nodes connected to both E and T. An example scenario where the normalization would be useful is in the case where a particular enrollment utterance E shares the same number of common neighbors with two separate test utterances T_1 and T_2, however $|N_{T_2}| \gg |N_{T_1}|$ and thus the Jaccard coefficient would penalize T_2.
- *Adamic*=$\Sigma_{z \in N_E \cap N_T} \frac{1}{log|N_z|}$ a measure that combines the size of the intersection set with how highly connected the nodes in the intersection are. This could be thought of as another form normalized common neighbors.

4.1.2. Weighted graph

The features in this section are inspired by those of the binary graph.

- *Inner product*=$v_E^t.v_T$ is based on the common neighbors measure.
- *Normalized inner products*=$\frac{v_E^t.v_T}{||v_E||.||v_T||}$ and $\frac{v_E^t.v_T}{||v_E||+||v_T||}$ which are inspired by Jaccard's coefficient.
- *Adamic Weighted*=$\Sigma_{z \in N_E \cap N_T} \frac{1}{log||v_z||}$, based on the binary Adamic feature.
- *Landmark Euclidean distance*=$||v_E - v_T||$, a measure that considers the recordings in the graph as landmarks and that the vectors v_E and v_T represent the coordinates of E and T in the space defined by the landmarks.

4.2. Paths Features

In the previous sections our discussion has focused on graphs constructed based on match scores. One can also create graphs based on the Euclidean distance between the TV representation of the recordings. In the K-NN version of the distance based graphs the NN are selected to be the closest ones to a recording in the Euclidean space. And in the epsilon version of the graphs, edges exist between nodes that are less than ϵ apart from one another. Given the normalization of the match score presented in Section 2 the euclidean distance between two recordings is just

$$e(utt_a, utt_b) \quad = \quad \sqrt{2 - 2s(utt_a, utt_b)}. \tag{3}$$

These distance graphs allow for extracting paths based features that go beyond the immediate neighborhoods of the trial utterances:

4.2.1. Shortest path

- *Shortest path*=$2^{-SP(E,T)}$, where $SP(E,T)$ is the value of the shortest from node E to T, which we compute using a Matlab implementation of the Dijkstra algorithm [9].
- *Number of hops*=$2^{-NH(E,T)}$, where $NH(E,T)$ is the number of edges traversed along the shortest path from E to T.

4.2.2. N-Step Markov (NSM):

NSM is a feature used to quantify the relative importance of E to T [10] by computing the probability that a random walk started at E will visit T after N steps are taken. Which can be computed as the value at the Tth index of the vector:

$$NSM(E,.) = \mathbf{A}i_E + \mathbf{A}^2 i_E + \mathbf{A}^3 i_E + ... + \mathbf{A}^N i_E, \tag{4}$$

where i_E is a vector of size Mx1 of all zeros except for 1 at the Eth index, and $\mathbf{A}$ is an MxM matrix representing transition probabilities from one node to another. We obtain $\mathbf{A}$ from the distance graph by dividing each outward edge from a node by the sum of all outward edges from that node. In this paper we choose to set $N = 15$ since beyond that the contribution of $\mathbf{A}^N i_E$ to the NSM score is minimal.

5. CLASSIFIER

In Section 3 we presented two graph embedding techniques, K-NN and epsilon graphs, each with a parameter that can be varied to obtain different resultant graphs. These graphs are then used in Section 4 to extract three categories of features: binary graph neighborhood

Table 1. The graph relational features used in classification

	K used in K-NN	ϵ used in Epsilon Graph
BGN	5, 10, 20, 50, 100, 250, 500, 750, 1000	.35, .4, .45
WGN	5, 10, 20, 50, 100, 250, 500, 750, 1000	-.4, -.3, -.2, -.1, 0, .1, .2, .3, .4
Paths		1.1, 1.2, 1.3

(BGN), weighted graph neighborhood (WGN) and paths features. Combining the different graph construction with the different feature extraction techniques results in a large set of features to represent each trial. We narrow the set down to 135 features according to the efficacy of each individual feature on the development set. Table 1 lists the resulting set.

These relational features are combined with the baseline match-score to obtain a 136 dimensional feature vector that represents each trial of interest, consisting of a train and test utterance. The features are then individually normalized to have zero mean and unit variance across the training set. A linear SVM classifier is then trained per gender on the development set to separate between true and false trials. This is done using the LibSVM toolbox [11] with five fold cross-validation to set the regularization parameter c. Once trained, the SVM is used to classify test trials as true or false. The next section presents the results of our approach on the speaker recognition and speaker mining tasks.

6. RESULTS

The focus of this work will be on the one conversation train one conversation test scenario and all results will use the 2008 NIST SRE English telephony data as a training/development set. The final performance will be measured on condition 5 of the 2010 NIST SRE which consists of normal vocal effort English telephony speech.

6.1. Speaker Recognition Task

The speaker recognition task follows the standard NIST SRE task requiring that each trial be considered independently of all other trials in the evaluation, therefore when extracting the graph relational features for a given trial it was embedded in an impostor background set. The background sets used are of size 6932 for males and 9281 for females and consist of utterances from the 2004/05/06 NIST SREs.The regularization parameter c was set via cross-validation to 5 for males and 15 for females. Figure 1 shows the detection error trade-off (DET) curves of the baseline, in blue, and our proposed algorithm, in red, on the NIST SRE 08 data which was used to train the SVM classifier. Keeping in mind that we are testing on the training data, it is worthwhile to note the potential of the graph relational features.

Figure 2 shows the DET curves of the baseline, in blue, and our proposed algorithm, in red, on the held out test set, and we note that our algorithm yields moderate improvement over the baseline.

6.2. Speaker Mining Task

In the speaker mining task, we relax the constraint requiring each trial to be considered independently and include all the trials of the particular evaluation in the graph background set along with utterances from the 2004/05/06 NIST SREs. This yielded background

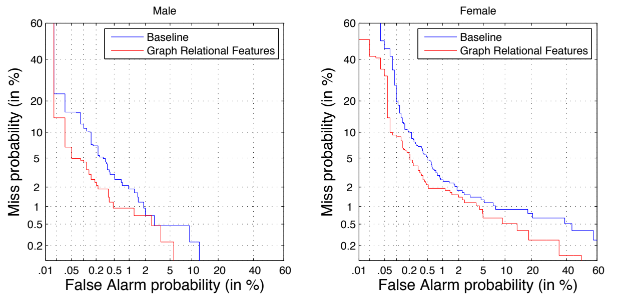

Fig. 1. Speaker recognition DET plots of the baseline and proposed system on the training set (NIST SRE 08).

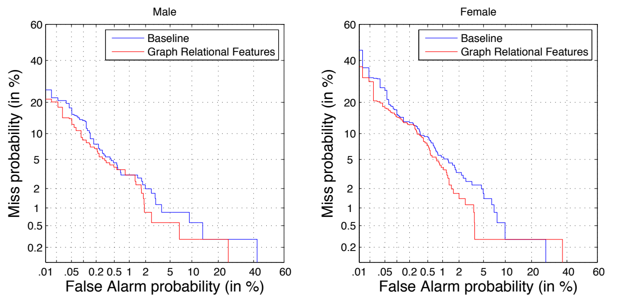

Fig. 2. Speaker recognition DET plots of the baseline and proposed system on the held out test set (NIST SRE 10).

sets of size 8475 for males and 12099 for females on the development set and 9868 and 13209 for males and females on the held out test set. We note that in this task the background set is not only comprised of impostor utterances and may have speaker overlap with the trial of interest. During SVM training the regularization parameter c was set via cross-validation to 3 for males and 2 for females. Figure 3 shows the DET curves of the baseline, in blue, and our proposed algorithm, in red, on the NIST SRE 08 data used to train the SVM classifier. As in the recognition task, keeping in mind that we are testing on the training data, it is worthwhile to note the potential of the graph relational features.

Figure 4 shows the DET curves of the baseline, in blue, and our

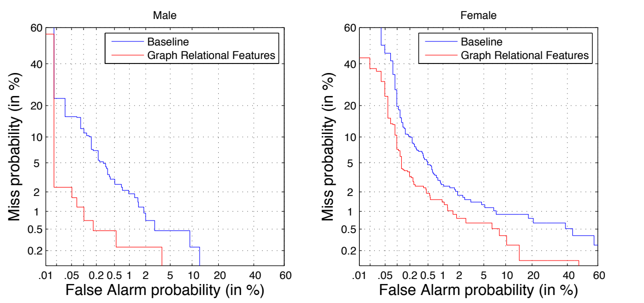

Fig. 3. Speaker mining DET plots of the baseline and proposed system on the training set (NIST SRE 08).

proposed algorithm, in red, on the held out test set and clearly shows the improvement of our algorithm over the baseline.

7. CONCLUSION AND FUTURE WORK

In this paper we have presented a framework to use relational features extracted from speaker similarity graphs for improved speaker

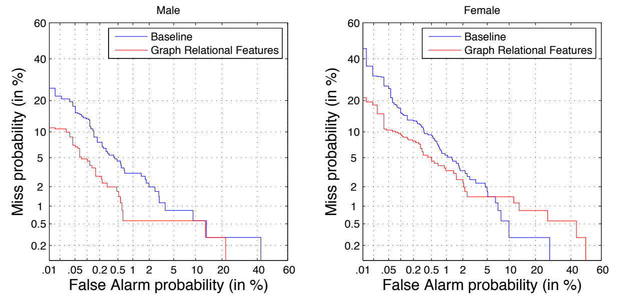

Fig. 4. Speaker mining DET plots of the baseline and proposed system on the held out test set (NIST SRE 10).

comparison. We applied this framework to two speaker comparison tasks, speaker recognition and mining. In both tasks, our proposed system outperformed the baseline, with significant improvement observed in the speaker mining task. We also present results from test on train scenarios to highlight the potential of the features.

Future work will focus on improving generalization to the test set by applying impostor driven feature normalization to the individual relational features, as well as subspace methods to handle correlation and noise in the feature vectors.

8. REFERENCES

[1] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, To Appear.

[2] W. M. Campbell, Z. Karam, and D. E. Sturim, "Speaker comparison with inner product discriminant functions," in *Advances in NIPS 22*, 2009, pp. 207–215.

[3] Z. N. Karam and W. M. Campbell, "Graph-embedding for speaker recognition," in *Proc. Interspeech*, 2010.

[4] Z. N. Karam, W. M. Campbell, and N. Dehak, "Towards reduced false-alarms using cohorts," in *Proc. ICASSP*, 2011.

[5] W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a gmm supervector kernel and nap variability compensation," in *Proc. ICASSP*, 2006.

[6] S. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised speaker adaptation based on the cosine similarity for text-independent speaker verification," in *Proc. Odyssey*, 2010.

[7] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proc. 12th International Conference on Information and Knowledge Management*, 2003.

[8] "The NIST year 2010 speaker recognition evaluation plan," http://www.itl.nist.gov/iad/mig/tests/sre/2010/index.html, 2010.

[9] J. B. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, 2000.

[10] S. White and P. Smyth, "Algorithms for estimating relative importance in networks," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003.

[11] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: a library for support vector machines," http://www.csie.ntu.edu.tw/ cjlin/libsvm, 2001.