Odyssey 2014:
The Speaker and Language Recognition Workshop
16-19 June 2014, Joensuu, Finland

Unsupervised Clustering Approaches for Domain Adaptation in Speaker Recognition Systems

[1]*Stephen H. Shum,* [2]*Douglas A. Reynolds,* [3]*Daniel Garcia-Romero,* [3]*Alan McCree*

[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA
[2]MIT Lincoln Laboratory, Lexington, MA, USA
[3]JHU Human Language Technology Center of Excellence, Baltimore, MD, USA
sshum@csail.mit.edu, dar@ll.mit.edu, {dgromero, alan.mccree}@jhu.edu

10.21437/Odyssey.2014-40

Abstract

In this paper, we motivate and define the domain adaptation challenge task for speaker recognition. Using an i-vector system trained only on out-of-domain data as a starting point, we propose a framework that utilizes large-scale clustering algorithms and unlabeled in-domain data to adapt the system for evaluation. In presenting the results and analyses of an empirical exploration of this problem, our initial findings suggest that, while perfect clustering yields the best results, imperfect clustering can still provide recognition performance within 15% of the optimal. We further present a system that achieves recognition performance comparable to one that is provided all knowledge of the domain mismatch, and lastly, we outline throughout this paper some of the many directions for future work that this new task provides.

1. Introduction

With the ubiquity, connectivity, and expansive storage of data-recording devices (smart phones, embedded sensors, etc.), we are in an era with almost unlimited access to data. Nevertheless, we often struggle to make sense – much less make effective use – of most of it. Much of these data that we have such convenient access to are unlabeled and thus useless in many of the traditionally supervised machine learning scenarios that require explicit labeled examples. Because the process of using humans to tag and annotate is expensive and time-consuming, we'd like to develop methods that utilize existing, previously labeled examples in ways that can make use of the many unlabeled examples at hand. This is the problem of transfer learning and domain adaptation [1, 2].

In this paper, we consider such a scenario in the context of speaker recognition. Over the past 5 years, the i-vector approach to speaker recognition has proven to be the best performing system as demonstrated in NIST speaker recognition evaluations (SRE) [3]. One of the keys to this success is a framework that easily allows the use of large amounts of previously collected and labeled audio to characterize and exploit speaker and channel variability. In the SRE scenario, data from thousands of speakers each making over 10 calls from at least 2 different handsets, collected in a consistent manner, has been readily available from previous years. However, it is unrealistic to expect a large set of labeled data from matched conditions when applying such a system to a new domain. In this paper, we describe a challenge task using SRE data that demonstrates the effect of a subtle domain mismatch and design experiments that allow for an empirical exploration of unsupervised domain adaptation techniques on i-vector speaker recognition systems.

The rest of this paper is organized as follows. In the following section we discuss the use of prior data in an i-vector system and describe an experiment that demonstrates the effect of domain mismatch. We then outline a domain adaptation challenge task for exploring techniques to mitigate performance degradations due to such mismatch; this was one of the topics explored at the Johns Hopkins University (JHU), Center for Language and Speech Processing (CLSP) Summer Workshop 2013. Section 3 presents some initial experiments that attempt to quantify, at least to some extent, the difference between the two domains in question and, to first order, shed some insight on how the domain adaptation problem can be approached. In Section 4, we propose our general experimental framework and describe how it fits into an i-vector speaker recognition system. Section 5 outlines the various clustering algorithms we explore, while in Section 6, we present our initial experiments, results, and analysis. Finally, Section 7 concludes this paper.

2. Domain Adaptation Challenge Task

To distinguish our explorations from those of the NIST i-vector Challenge [4], we first explain the domain adaptation problem at hand. As mentioned previously, this task may also be referred to as the 2013 JHU CLSP Summer Workshop Challenge.

While a detailed description of the i-vector system and theory is beyond the scope of this paper (but can be found in [3, 5, 6]), it is worth providing a high-level overview to note where labeled and unlabeled data is required. In Figure 1 we show a simplified block diagram of i-vector extraction and scoring. A speech utterance (e.g., one side of a telephone call in SREs) is first represented by how its acoustic features (MFCC+deltas) are distributed relative to a Universal Background Model (UBM), which is a Gaussian mixture model (GMM) characterizing speaker-independent speech feature distributions. This representation consists of the zeroth-order (counts) and first-order (means) sufficient statistics of the speech utterance. These sufficient statistics are then transformed into an i-vector, typically of 600 dimensions, using a total variability matrix T. The i-vector is then whitened by subtracting a global mean, m, and scaling by the inverse square root of a global covariance matrix, W, and then length-normalized to unit length [5]. Finally, a scoring function between a model and test i-vector is computed; this requires a within-class (WC)

This work was sponsored by the Department of Defense 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.

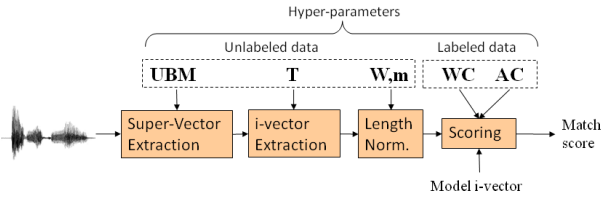

Figure 1: *High-level diagram of i-vector system showing all hyper-parameters and which, respectively, require labeled and unlabeled data to train.*

	SRE	SWB
# spkrs (m, f)	3790 (1115, 2675)	3114 (1461, 1653)
# calls	36470	33039
Avg. # calls/spkr	9.6	10.6
Avg. # phns/spkr	2.8	3.8

Table 1: *Summary statistics for the SRE and SWB training lists.*

matrix, characterizing how i-vectors from a single speaker vary, and an across class (AC) matrix, characterizing how i-vectors between different speakers vary. The scoring function most often used is called Probablistic Linear Discriminant Analysis (PLDA) [5, 7].

Collectively, the UBM, T, W, m, WC, and AC are known as the system's hyper-parameters and must be trained before a system can enroll and/or score any data. The UBM, T, W, and m represent general feature distributions and total variance of statistics and i-vectors, so they only require unlabeled data for training. The WC and AC matrices, however, require labeled data to learn within speaker (calls from the same speaker) and across speaker (calls from different speakers) variabilities. While all hyper-parameters are susceptible to mismatch, those requiring labeled data to train are more difficult to handle.

When porting a system to a new domain, we are faced with three options: (1) assume the new domain data is sufficiently close to the data used to train the hyper-parameters that the system will work well, (2) collect a large amount of unlabeled data from the new domain and adapt the hyper-parameters using unsupervised techniques, or (3) collect and label sufficient amounts of new domain data to allow re-training or supervised adaptation of the hyper-parameters. In this paper we will explore approaches to option (2).

Using Linguistic Data Consortium (LDC) telephone corpora, we have designed an experiment that demonstrates the effect of data mismatch on hyper-parameters and defines the challenge task on which we are working. In this experiment, SRE10 telephone data is used as enroll and test sets. Specifically, we are using the one conversation (1c) telephone data enroll and test lists from condition 5 (normal vocal effort) [8, 9]. We designate two datasets to be used for hyper-parameter training: the *in-domain* **SRE** set consists of all telephone calls from all speakers taken from the SRE 04, 05, 06, and 08 collections; this will serve as the "matched" hyper-parameter training list. The *out-of-domain* **SWB** set consists of all telephone calls from all speakers taken from the switchboard-I and switchboard-II (all phases) corpora; this will serve as the "mismatched" training list. Some key statistics of the two data sets are given in Table 1.

These two datasets appear very similar and the expectation is hyper-parameters trained from these should produce similar results. However, the resulting equal error rates (EER's) in

#	UBM & T	W & m	WC & AC	1c EER (%)
1	SWB	SWB	SWB	6.92%
2	SWB	SRE	SWB	5.54%
3	SWB	SRE	SRE	**2.30**%
4	SRE	SRE	SRE	2.43%

Table 2: *EER's on SRE10 from hyper-parameters trained using the SWB or SRE datasets, as specified.*

Table 2 clearly show a gap in performance on the SRE10 enroll/test set when hyper-parameters are trained with the different sets.[1] Similar performance gaps were observed by other sites using independent i-vector implementations, indicating that the performance gap is not a function of particular implementation details (features, speech activity detection, hyper-parameter training algorithms, etc.).

In this paper we are primarily focused on how to effectively train or adapt the hyper-parameters that depend on labeled data (WC, AC) when only unlabeled data is available in the target domain. Of the hyper-parameters which do not depend on labeled data – UBM, T, W, and m – it was found on this challenge set that the difference in using SWB or SRE for UBM and T training was insignificant, but using SRE (in-domain) data for training the whitening, W and m, gave a significant improvement (compare rows 1 and 2 in Table 2) [10]. We will use the system specified in row 2 of Table 2 as our starting out-of-domain baseline and the result in row 3 as our desired in-domain benchmark. To avoid making this a data engineering exercise, we restrict our system to only use the labeled SWB data and unlabeled SRE data. The ultimate goal is to develop a recipe that can be applied in future situations where only unlabeled data from a new domain is available.

3. Exploring the Domain Mismatch

Before attempting to compensate for the mismatch in performance between the SWB and SRE corpora, we attempt to explain, anecdotally, some of the difference between the two datasets. An analysis of respective age distributions and languages spoken – SWB includes only English, while SRE contains speech from over 20 different languages – yielded no fruitful insights. This came as a surprise; the work in [11] seemed to demonstrate that a discrepancy in languages spoken would introduce a dataset shift. However, we did notice that using different subsets of SWB produced different recognition results.

In particular, the entire SWB set can be broken down into six subsets that approximately correspond to their chronological release. Follow the labeling convention of [12], these subsets are: SWPH0 (1992), SWPH1 (1996), SWPH2 (1997), SWPH3 (1997), SWCELLP1 (1999), and SWCELLP2 (2000). We observed initially that, upon training a simple linear classifier to separate between the SRE and SWB i-vectors, the subsets of SWB that shared the most overlap with the SRE data were SWCELLP1 and SWCELLP2, while those that were easiest to separate were SWPH0, SWPH1, and SWPH2.[2] In light of this, we ran another set of baseline experiments using the various subsets of SWB in reverse chronological order. That is, we first use the labels from just SWCELLP2 and SWCELLP1, which are the two most recent subsets of SWB. Then we add in SWPH3, followed by SWPH2 and SWPH1, before finally in-

[1] We provide our implementation details in Section A.1 of the Supplementary Materials.

[2] Conversely, the 04, 05, 06, and 08 collections composing the SRE data seem to be more homogenous and do not exhibit similar trends.

cluding SWPH0. These results are shown in Table 3. For these experiments, we only varied the data used to train our WC & AC matrices; our UBM & T were always trained using the entirety of SWB (all subsets), while W & m were obtained using all of SRE. As such, row 4 of Table 3 displays exactly the same result as row 2 of Table 2.

#	WC & AC	1c EER (%)
1	SWCELLP1/2	4.67%
2	+ SWPH3	**3.51**%
3	+ SWPH1/2	4.85%
4	+ SWPH0	5.54%

Table 3: *EER's on SRE10 using various subsets of SWB to train the WC & AC hyper-parameters. Each of rows 2-4 implies the use of the SWB subset specified as well as all the data in the rows above. Per Section 3, the UBM & T were trained using all of SWB, while W & m were obtained from all of SRE.*

From these results, we can see that the SWB data is not homogenous and that there certainly exist subsets of our out-of-domain SWB data that are more suited to the in-domain SRE data. Similar findings were reported in [13], where the mismatch caused by different SWB subsets was compensated via a Nuisance Attribute Projection (NAP) before applying PLDA. These observations also seem to support a conjecture that the mismatch is, in part, driven by the progress in telephone technology moving from landline to cellular. A more detailed analysis of the meta-data, however, would be required before any more assertions can be made.

We ran two additional experiments to test whether the domain mismatch can be overcome simply by selecting a subset of the out-of-domain SWB i-vectors for WC & AC training in some clever way. This sort of a strategy is known more formally in the literature as *covariate shift adaptation* and revolves around techniques such as *importance sampling* or *weighting* [14, 15]. Our initial experiments described below are not as sophisticated or well-developed, but we would like to point out that an initial attempt at the covariate shift problem in the context of closed-set speaker identification (i.e., as opposed to our problem of open-set speaker verification) was done in [15] and demonstrated some improvement using the techniques developed by [16, 17]. Some possible reasons why the work in [15] did not seem to achieve more significant improvements – despite demonstrating significant gains on the tasks evaluated in [16, 17] – may have been due to their evaluating on a closed-set speaker identification test set consisting of only ten speakers as well as their choice to identify speakers using a mere 1.5 s of observed speech, which is orders of magnitude less data than the 150 s of speech that we typically use to build speaker models for our task. In light of this, an investigation of these methods under a more appropriate context may be a fruitful avenue for further analysis.

Figure 2 shows the result of our first approach in blue, where SRE10 EER is plotted as a function of the proportion, x, of SWB i-vectors that were the closest in likelihood to the marginal distribution of the in-domain SRE i-vectors. When $x = 1.0$, we are using all of SWB, so the result is, correspondingly, the same as both row 4 of Table 3 and row 2 of Table 2. Similarly, in green is the set of results obtained using the proportion, x, of SWB i-vectors that were closest to the SRE i-vectors in a linear discriminant sense. That is, we trained a simple linear classifier between the SWB and SRE corpora and used the subset of SWB i-vectors whose scores were closest to those of the SRE i-vectors.

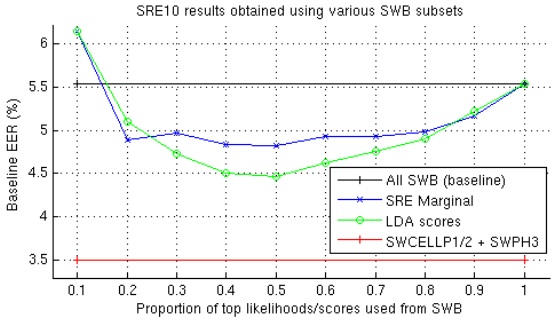

Figure 2: *SRE10 results obtained using various subsets of the SWB data for WC & AC.*

The results shown in Figure 2 suggest that there exist ways in which we can improve our baseline results by selecting, in unsupervised fashion, subsets of our out-of-domain data to match our in-domain data as closely as possible. However, our analysis is incomplete in two ways: (a) our methods for subset selection are still unable to attain performance comparable to the 3.51% EER obtained using SWCELLP1/2 + SWPH3 on row 2 of Table 3 and shown in red on Figure 2; and (b) we are still unable to explain why this aforementioned subset of SWB is able to obtain such an outstanding baseline result.

Upon implementing the domain adaptation framework outlined in Section 4, our experimental results still demonstrate that using *all* of the SWB data for WC & AC still provides the best speaker recognition performance. Indeed, despite the ability of subset selection to improve the initial baseline, there really is "nothing better than more data." Thus, the rest of this paper does not discern between different SWB subsets; we will use all of SWB as our labeled, out-of-domain data. We do, however, plan to return to a deeper analysis of domain mismatch as part of future work.

4. General Framework and Initial Setup

We begin our work assuming the existence of an initial set of hyper-parameters and PLDA scoring function [5, 10]; exact implementational details are consistent with our parallel work in [18] and can be found in the Supplementary Materials. For notational convenience, we will subsequently use Σ to refer to the WC matrix and Φ to refer to the AC matrix. As such, our initial setup begins with Σ_{SWB} and Φ_{SWB}, which we train using the labeled SWB data that are provided.

We propose the following approach to the domain adaptation problem and adhere to it throughout the rest of this work:

(a) Use Σ_{SWB} and Φ_{SWB} to compute a pairwise affinity matrix, A, on the unlabeled SRE data. Specifically, element A_{ij} is the log likelihood ratio between the hypothesis that i-vectors i and j are from the same speaker and the hypothesis that they come from different speakers.

(b) Use A to obtain a hypothesized speaker clustering of the SRE data. A discussion of different clustering algorithms will be covered in Section 5. These estimated speaker clusters can then be used to obtain Σ_{SRE} and Φ_{SRE}.

(c) The work in [10] found success in linearly interpolating between the prior (SWB) and new (SRE) covariance ma-

trices to obtain the final hyper-parameters:

$$\Sigma_{\text{F}} = \alpha_{\text{WC}} \cdot \Sigma_{\text{SRE}} + (1 - \alpha_{\text{WC}}) \cdot \Sigma_{\text{SWB}} \quad (1)$$

$$\Phi_{\text{F}} = \alpha_{\text{AC}} \cdot \Phi_{\text{SRE}} + (1 - \alpha_{\text{AC}}) \cdot \Phi_{\text{SWB}} \quad (2)$$

To simplify notation, we denote the set of parameters as $\alpha = \{\alpha_{\text{AC}}, \alpha_{\text{WC}}\}$. Note that setting $\alpha = \mathbf{1}$ corresponds to $\Sigma_{\text{F}} = \Sigma_{\text{SRE}}$ and $\Phi_{\text{F}} = \Phi_{\text{SRE}}$, or the hyper-parameters obtained using *only* the hypothesized speaker labels obtained from clustering the unlabeled SRE data. Conversely, setting $\alpha = \mathbf{0}$ is equivalent to not using any of the in-domain data; this yields the 5.54% EER shown on row 2 of Table 2.

Another possibility is to iterate this procedure, where the Σ_{F} and Φ_{F} obtained in step (c) respectively replace the Σ_{SWB} and Φ_{SWB} of step (a) and the process is repeated until some form of convergence criterion is met, after which we proceed to the final recognition task. Extensive coverage of these experiments is beyond the scope of this paper, but we did observe from some pilot experiments that, assuming a reasonable choice of clustering algorithm in step (b), iterating can have a positive effect on both clustering and recognition performance. These improvements were also relatively inexpensive to obtain, as the results tended to converge within just a few (≤ 5) iterations of this algorithm, which suggests that further investigation along these lines may be a fruitful avenue for future work.

5. Clustering Algorithms

Our experiments focus on a subset of the algorithms from previous work on speaker clustering [19], which were chosen to work given only a (potentially sparse) pairwise affinity matrix. That is, we need not go back to the raw data (i-vectors); simply knowing the relationships between them will suffice.

The well-known and widely-used *agglomerative hierarchical clustering (AHC)* is a simple and greedy algorithm that works in bottom-up fashion, initializing each i-vector as its own cluster and iteratively merging two clusters at a time via some cluster-similarity metric – we use the unweighted group average score [20] – until some stopping criterion (e.g., BIC [21], maximum distance [22], number of clusters [19], etc.) is met [23]. In our implementation, the number of clusters is provided as an input; Section 6.2 discusses how that stopping criterion can be estimated automatically.

We also evaluate the performance of various random walk algorithms explored in earlier work [19]. Both *Infomap* and *Markov Clustering (MCL)* are explained in [24] and [25], respectively. Here, each i-vector can be thought of as a node on a large graph, and each edge is weighted according to the affinity between the two associated i-vectors. In agreement with our previous observations, both methods worked well assuming some reasonable choice of a sparse graph (i.e., affinity matrix). As such, we implemented the local node pruning algorithm outlined in both the Supplementary Materials and in Section 5.2 of [19] to automatically sparsify the affinity matrix.[3]

6. Experiments and Results

6.1. Initial Results and Observations

For each clustering algorithm, we report the following measures of clustering performance: number of speakers estimated ($\hat{K}$), cluster confusion error as detailed in Section 4.2 of [19] and implemented in the evaluation of "speaker confusion error" the NIST Speaker Diarization scoring script [26],[4], average cluster purity, and average speaker fragmentation, which we define as the average number of clusters used to represent all utterances of a single speaker.

For recognition performance, we present the EER's obtained using various values of α. In addition to both $\alpha = \mathbf{0}$ EER and $\alpha = \mathbf{1}$ EER introduced in Section 4, we also consider α^* EER, which is the SRE10 performance of hyper-parameters trained using both the hypothesized speaker labels obtained from clustering on SRE *and* combined (via some α) with the hyper-parameters from the labeled SWB data, as is represented in Eqns. (1) and (2). In this scenario, we report the best result obtained across all values $\alpha \in [0, 1] \times [0, 1]$, though for simplicity, our experiments sample α in intervals of 0.2.

In addition to reporting the results from our hypothesized (Hyp.) clusters, we also show the results of a perfect clustering, which is the SRE10 performance using hyper-parameters obtained from the use of exact speaker labels *and* the selection of the optimal values for α^*. Since we only use sampled subsets of the SRE data in this experiment, this result represents the best we can do. Admittedly, the "α^* EER" is an oracle-based experiment that assumes knowledge of some best-case scenario. We report our results this way so as to establish performance bounds in a controlled environment.

The results from our initial experiments are shown in rows 1-3 of Table 4. Instead of simply obtaining Σ_{SRE} and Φ_{SRE} by clustering just once on the entire SRE dataset, we strove to attain some form of statistical significance by sampling just a subset of the SRE data ten different times. For each sample, we randomly select all utterances from $K = 1000$ out of the original 3790 speakers, cluster all their utterances using the various algorithms described in Section 5, and then obtain the corresponding hyper-parameters for the speaker recognition task.

We can see that AHC provides the best clustering and recognition results by a significant margin. Yet, despite a rather wide range of clustering performances, the resulting range in speaker recognition performance is not nearly as dramatic. This could be specific to the SWB and SRE datasets; we know that the EER is upper-bounded at 5.5% using just the SWB hyper-parameters. When $\alpha = \mathbf{1}$, a better clustering algorithm yields better recognition results; however, the impact of clustering on recognition performance is attenuated by the presence of adaptation (i.e., when $\alpha \in (0, 1) \times (0, 1)$).

Figure 3 shows a summary of AHC clustering and subsequent recognition results as we vary the number of speakers sampled from the SRE data. The top plot shows, in red and green lines respectively, the α^* EER and $\alpha = \mathbf{1}$ EER, while the results obtained via hypothesized versus perfect clusters are denoted by dash-dotted and solid lines, respectively. In black is the $\alpha = \mathbf{0}$ EER line, while the blue line, whose corresponding y-axis is on the right side of the plot, denotes cluster confusion error. Despite cluster error increasing as we sample larger and larger speaker subsets, we can see that recognition EER's continue to decrease, though the rate of decline seems to slow after 2500 speakers.

The values of α^* given perfect clusters may be different from the values of α^* under the hypothesized clusters. In our experiments, we observed that α^*_{WC} for hypothesized clusters was consistently similar to α^*_{WC} for perfect clusters. However, the bottom plot of Figure 3 shows that the difference between α^*_{AC} for hypothesized (red) and perfect (blue) clusters increases

[3] Sections A.2, A.3 of the Supplementary Materials also provide an outline of these methods.

[4] Section A.4 of the Supplementary Materials reiterates the specifics.

#		# Spkrs K	# Clstrs $\hat{K}$	Clustering Performance			α^* EER (%)			$\alpha = \mathbf{1}$ EER (%)		
				Confusion	Purity	Frag.	Perfect	Hyp.	Gap	Perfect	Hyp.	Gap
1	AHC	1000	1000*	7.4%	94.9%	1.20	2.37	**2.55**	**7.8%**	2.77	**3.16**	**14%**
2	Infomap	—	918	18.2%	85.9%	1.44	—	2.71	14%	—	3.45	25%
3	MCL	—	997	15.1%	90.3%	1.45	—	2.68	13%	—	3.40	23%
4												
5	Infomap+AHC	1000	918	9.0%	92.6%	1.19	2.37	2.56	8.2%	2.77	3.18	15%
6	MCL+AHC	—	997	7.5%	94.9%	1.20	—	**2.56**	**8.0%**	—	**3.16**	**14%**

Table 4: *Results from initial experiments in domain adaptation. Clustering performance was evaluated using labels from the SRE data; recognition performance (EER's) is reported for the 1c task in SRE10. Section 6.1 explains rows 1-3; Section 6.2 discusses rows 5-6.*

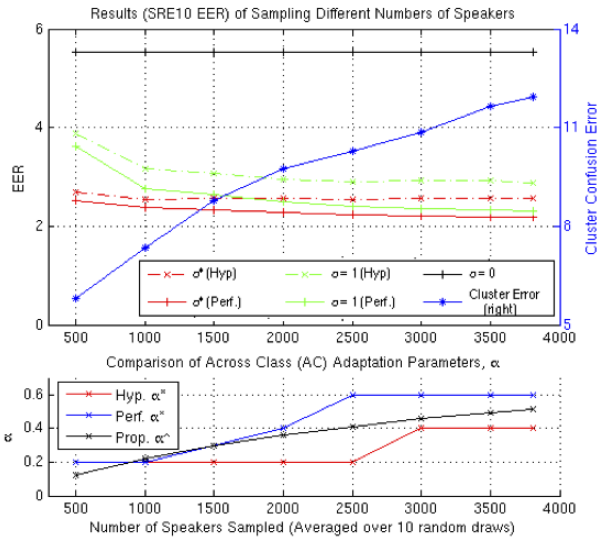

Figure 3: *Summary of clustering (AHC) and recognition (SRE10) results as a function of the number of speakers sampled from the SRE data. A detailed explanation can be found in Section 6.1.*

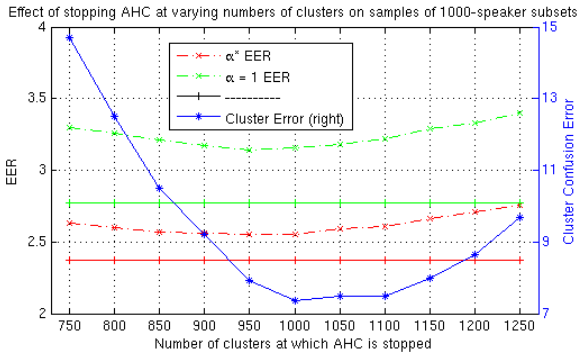

Figure 4: *Effect of stopping AHC at varying numbers of clusters. Dash-dotted and solid lines correspond to results using hypothesized and perfect clusters, respectively. A more detailed explanation can be found in Section 6.2.*

with the number of speakers sampled. For reference, we also plot the value of $\hat{\alpha}_{\text{AC}}$ (black), which is the value that reflects the relative proportion of the number of speakers between the SRE data used and the SWB data. While values for $\alpha^*_{\text{WC}} \in [0.4, 0.8]$ for the most part, we see that $\alpha^*_{\text{AC}} \in [0.2, 0.6]$. This may suggest that adapting to the SRE WC matrix is of higher relative importance than adapting to the SRE AC matrix.

From rows 1-3 of Table 4, it is clear that AHC, when given the number of speakers, provides the best clustering and recognition results. Nevertheless, Infomap and MCL are able to do a reasonable job in detecting the number of speakers, which we did not explicitly explore with AHC. Instead, we could simply consider running MCL or Infomap to obtain an estimate of the number of speakers, $\hat{K}$, and use that estimate as an input to the AHC algorithm. This brings to bear the question of how robust AHC is to error in estimating the number of speakers. In particular, if MCL/Infomap do not provide an exact estimate of the number of speakers as an input to AHC, how much does that affect subsequent recognition results?

6.2. Effect of Cluster Number on Recognition Performance

Figure 4 shows the result of stopping AHC at varying numbers of clusters. These results are averaged over ten random draws of 1000 speakers, and α^* is optimized as previously discussed. The plot of cluster confusion error, in blue, is scaled according to the y-axis on the right and shows that clustering performance is best when AHC is provided a number of clusters equal to the number of speakers present. Yet, considering recognition performance alone, we can see that the resulting EER is relatively robust to stopping AHC at incorrect numbers of clusters. We can actually provide AHC with a significant underestimate of the number of speakers and still do fairly well on the SRE10. Additional experiments are necessary to better understand this phenomenon; in particular, an underestimate seems more forgiving than an overestimate, which implies the somewhat counterintuitive idea that modeling multiple speakers as one cluster is acceptable. One hypothetical explanation is that the resulting WC matrix accounts for additional uncertainty that is somehow beneficial to our task, but we leave this as an open thread for further analysis.

In rows 5-6 of Table 4, we show the results of using Infomap and MCL to estimate the number of speakers and taking that estimate as an input to AHC for clustering and recognition. We can see that both random walk algorithms are able to provide a reasonable estimate of the number of speakers, and the resulting recognition performance is just about as good as the case in which AHC is given the exact number of speakers (row 1). This is expected, as subsequent partitions produced at each step of AHC differ only by a single cluster merge and thus yield only small changes in cluster error. But more significantly, the gap in recognition performance between knowing and not knowing *a priori* the number of speakers in the unlabeled SRE is effectively nil. Indeed, when final recognition performance is the main priority, obtaining an exact estimate of the number of speakers may, in fact, be unnecessary.

As a final experiment, we run our proposed adaptation procedure on the full SRE data, using Infomap and MCL to esti-

	$\hat{K}$	Perfect	Hypothesized	Gap (%)
AHC	3790*	2.23	2.58	16%
Infomap+AHC	3196	—	**2.53**	**13%**
MCL+AHC	3971	—	2.61	17%

Table 5: *SRE10 results obtained using the entire unlabeled SRE dataset and optimal hyper-parameter adaptation, with* $\alpha^*_{AC} = 0.4$ *and* $\alpha^*_{WC} = 0.8$. *It should be noted that the 2.23% EER given a perfect clustering is different from the 1c EER of 2.30% shown in row 3 of Table 2 because of the adaptation with SWB hyper-parameters. The latter result is obtained with no adaptation, or* $\alpha^*_{AC} = \alpha^*_{WC} = 1$.

mate the number of speakers for input to AHC. Table 5 shows our final results, which were obtained with $\alpha^*_{\text{AC}} = 0.4$ and $\alpha^*_{\text{WC}} = 0.8$. Note how Infomap+AHC severely underestimates the number of speakers – thus obtaining the worst clustering performance of the three algorithms – but manages to attain recognition performance that is even better than when AHC is given the correct number of clusters. We hope to better understand this phenomenon in future work.

6.3. Automatic Estimation of Adaptation Parameters

We have not yet addressed a way to automatically determine the optimal values for $\alpha = \{\alpha_{\text{AC}}, \alpha_{\text{AC}}\}$. While a complete undertaking of the problem is beyond the scope of this paper, Figure 5 shows the result of independently optimizing both α_{AC} and α_{WC}, averaged over ten sampled subsets of 1000 speakers; the color scaling is shown to the right of each subplot, and blue indicates a relatively low EER, while red indicates a relatively high EER. The plot on the left suggests that there is a reasonably wide range of possible values for α that yield EER's less than 3%; this fact is consistent for sampled subsets that contain different numbers of speakers as well (e.g., 500, 1500, 2000, etc.). The heatmap on the right, in which the color scaling is limited to only the values that are within 10% of the optimal EER of 2.55% shown on the left, further confirms this notion. It seems as though we can obtain sufficiently good results simply by erring on the low side (i.e., $[0, 0.4]$) in our estimate of α_{AC} and using a moderate value of α_{WC} (i.e., $[0.4, 0.8]$), but more experiments are needed to better understand this phenomenon and how it might generalize to other datasets.

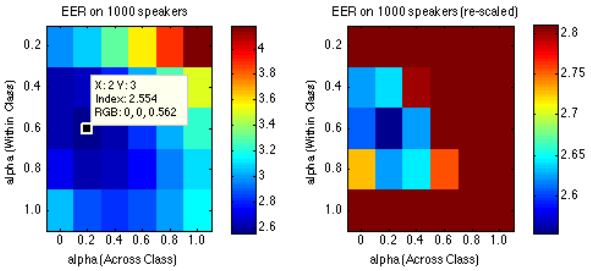

Figure 5: *Heatmaps showing the result of independently optimizing the adaptation parameters,* α. *Both plots involve the same raw data but different color scalings to illustrate the range of* α *that is appropriate for domain adaptation.*

7. Conclusion

In this paper, we motivate and define the domain adaptation challenge task for speaker recognition. Using the proposed framework and various unsupervised clustering algorithms, we present the results of initial experiments and highlight avenues for further analysis. We have seen that both an imperfect clustering and an imprecise estimate of the number of speakers are forgivable in the presence of adaptation with out-of-domain hyper-parameters. And although the optimal selection of their values remains an open question, we observe that a range of adaptation parameter values yields decent results. Finally, our best system so far obtains recognition performance that is within 15% of a system that has access to all speaker labels.

Acknowledgments We would like to thank Najim Dehak and Ekapol Chuangsuwanich for their many insights and helpful discussions in the development of this work.

8. References

[1] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, October 2010.

[2] Hal Daume III and Daniel Marcu, "Domain adaptation for statistical classifiers," *Journal of Artificial Intelligence Research*, vol. 26, pp. 101–126, May 2006.

[3] Najim Dehak, Patrick Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[4] NIST, "The 2013-2014 speaker recognition i-vector machine learning challenge," 2013, http://www.nist.gov/itl/iad/mig/upload/sre-ivectorchallenge_2013-11-18_r0.pdf.

[5] Daniel Garcia-Romero and Carol Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proceedings of Interspeech*, 2011.

[6] Bengt J. Borgstrom and Alan McCree, "Discriminatively trained bayesian speaker comparison of i-vectors," in *Proceedings of ICASSP*, 2013.

[7] Simon J.D. Prince and James H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proceedings of ICCV*, 2007.

[8] NIST, "Speaker recognition evaluation 2010," 2010, http://www.nist.gov/itl/iad/mig/sre10.cfm.

[9] Alvin Martin and Craig Greenberg, "The 2010 nist speaker recognition evaluation (sre10)," 2010, http://www.nist.gov/itl/iad/mig/upload/SRE10_maineval_workshop_public_brief.pdf.

[10] Daniel Garcia-Romero and Alan McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proceedings of ICASSP*, 2014.

[11] Carlos Vaquero, "Dataset shift in plda-based speaker verification," in *Proceedings of Odyssey*, 2012.

[12] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and Others, "Promoting robustness for speaker modeling in the community: the prism evaluation set," 2012, http://code.google.com/p/prism-set/.

[13] Hagai Aronowitz, "Adaptation of plda to new domains," in *Results from JHU CLSP Summer Workshop*, 2013.

[14] Hidetoshi Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, October 2000.

[15] Makoto Yamada, Masashi Sugiyama, and Tomoko Matsui, "Covariate shift adaptation for semi-supervised speaker identification," in *Proceedings of ICASSP*, 2009.

[16] M. Sugiyama, M. Krauledat, and K.-R. Muller, "Covariate shift adaptation by importance weighted cross validation," *Journal of Machine Learning Research*, 2007.

[17] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bunau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," *Annals of the Institute of Statistical Mathematics*, 2008.

[18] Daniel Garcia-Romero, Alan McCree, Stephen Shum, Niko Brummer, and Carlos Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[19] Stephen H. Shum, William M. Campbell, and Douglas A. Reynolds, "Large-scale community detection on speaker content graphs," in *Proceedings of ICASSP*, 2013.

[20] R. Sokal and C. Michener, "A statistical method for evaluating systematic relationships," *University of Kansas Science Bulletin*, 1958.

[21] David A. van Leeuwen, "Speaker linking in large data sets," in *Proceedings of Odyssey*, 2010.

[22] Marijn Huijbregts and David van Leeuwen, "Large scale speaker diarization for long recordings and small collections," *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

[23] Sue E. Tranter and Douglas A. Reynolds, "An overview of automatic speaker diarisation systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1557–1565, September 2006.

[24] Martin Rosvall and Carl T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, 2008.

[25] Stijn van Dongen, *Graph Clustering by Flow Simulation*, Ph.D. thesis, University of Utrecht, May 2000.

[26] NIST, "Diarization error rate (DER) scoring code," 2006, www.nist.gov/speech/tests/rt/2006-spring/code/md-eval-v21.pl.

[27] Douglas Reynolds, Thomas Quatieri, and Robert Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[28] Andrea Lancichinetti and Santo Fortunato, "Community detection algorithms: a comparative analysis," *Physical Review E*, 2009.

[29] Enrique Amigo, Julio Gonzalo, Javier Artiles, and Felisa Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information Retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

A. Supplementary Materials

In this section, we provide more extended explanations that are relevant to our discussion in the main paper. With the exception of Part A.1, in which we provide the details of our i-vector system implementation, much of this section is a reiteration of our previous work [19]. Although this is mostly a reiteration of previously published research, we feel that this would be a convenient way for the interested reader to obtain a necessary understanding of the essentials. To that end, Part A.2 refers to the clustering algorithms discussed in Section 5, while Part A.3 explains the local node pruning algorithm also mentioned in Section 5. And lastly, Part A.4 provides more detail on our choice of clustering performance metric.

A.1. System Implementation

In this section, we give an overview of the i-vector system implementation used in our experiments. Both the UBM, which is a Gaussian mixture model (GMM) characterizing speaker-independent speech feature distributions [27], and the total variability matrix, T [3], were trained using just SWB. Before obtaining any PLDA hyperparameters for either SWB or SRE, we obtain a whitening transform (global mean subtraction and scaling by the inverse square root of the global covariance matrix, W) from just the unlabeled SRE data;[5] note that whitening is an unsupervised procedure and does not require any speaker labels [5]. This whitening transform is applied to both the i-vectors from the SWB and SRE, and finally, all the i-vectors are length-normalized to unit length. The initial PLDA hyperparameters, Φ_{SWB} and Σ_{SWB}, are then obtained using the speaker labels from SWB and their respective pre-processed i-vectors.

A.2. Graph Clustering Algorithms

In this section, we provide a summary of the random walk algorithms we explored in [19] and utilized in this paper.

Markov Clustering (MCL) [25] As summarized in [28], this algorithm converts a graph affinity matrix to a stochastic matrix by dividing the elements of each row by their sum and then iterates between two steps. During *expansion*, we compute an integer power of this matrix (usually a square), which yields the probability matrix of a random walk after that number of steps (e.g., 2). During *inflation*, each element of the matrix is raised to some power, α, artificially enhancing the probability of a random walker being trapped within a community. These steps are iterated until we obtain the stochastic matrix of a forest (i.e., disconnected clusters), whose components are the communities. By solely iterating on the stochastic matrix, this method satisfies the Markov property, and we obtain clusters of separated communities upon convergence. We run this algorithm according to the implementation provided by [25] using the default settings for the parameter $\alpha = 2$.

Infomap [24] The problem of finding the best cluster structure of a graph can be seen as the problem of optimally compressing its associated random walk sequence. The goal of Infomap is to arrive at a two-level description that exploits both the network's structure and the fact that a random walker is statistically likely to spend long periods of time within certain clusters of nodes. More specifically, we look for a module partition $\mathbf{M}$ (i.e., set of cluster assignments) of N nodes into m clusters

[5] The only time we used SWB data for whitening was to obtain the results in row 1 of Table 2.

that minimizes the following expected description length of a single step in a random walk on the graph:

$$L(\mathbf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p_{\circlearrowright}^{i} H(\mathcal{P}^{i}). \quad (3)$$

This equation comprises two terms: first is the entropy of the movement between clusters, and second is the entropy of movements within clusters, both of which are weighted respectively by the frequency with which it occurs in the particular partitioning. Here, $q_{\curvearrowright}$ is the probability that the random walk switches clusters on any given step, and $H(\mathcal{Q})$ is the entropy of the top-level clusters. Similarly, $H(\mathcal{P}^{i})$ is the entropy of the within-cluster movements and $p_{\circlearrowright}^{i}$ is the fraction of within-cluster movements that occur in cluster i.

Ultimately, Eqn. (3) serves as a criterion for a bottom-up agglomerative clustering search. The implementation provided by [24] uses Eqn. (3) to repeatedly merge the two clusters that give the largest decrease in description length until further merging gives an increase. Results are further refined using a simulated annealing approach, the specifics of which can be found in [24]. Our work in this paper, however, did not use this algorithm according to the exact implementation from [24]; rather, we used the modified version detailed below.

Infomap-λ Although the original formulation of Infomap in [24] involves no tuneable parameters, the minimization criterion presented in Eqn. (3) implicitly assigns equal weight to the between-cluster and within-cluster entropies. As such, our previous work introduced a parameter, λ, into the equation as follows [19]:

$$L(\mathbf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \lambda \sum_{i=1}^{m} p_{\circlearrowright}^{i} H(\mathcal{P}^{i}). \quad (4)$$

The original Infomap corresponds to $\lambda = 1$. Letting $\lambda \to \infty$ increases our relative sensitivity to within-cluster entropy and yields more clusters that are smaller in size. Conversely, letting $\lambda \to 0$ favors larger and fewer clusters. We ran this algorithm with $\lambda = 1.5$, as that was the value that gave us the most consistent results in our previous experiments [19].

A.3. Local Node Refinements [19]

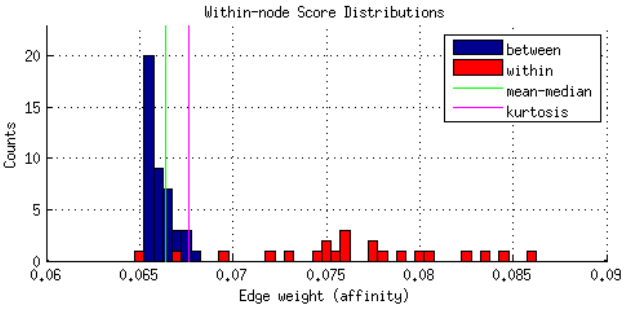

Figure 6: *Example histogram of within- and between-speaker score distributions for one particular node, as well as the cutoff thresholds discussed in Section A.3.*

Figure 6 shows the distribution of the top 100 PLDA log likelihoods between an arbitrary utterance A produced by speaker s_A and the rest of the utterances in the data. These scores are separated into two different histograms: red "within-speaker" scores and blue "between-speaker" scores. The combined score distribution, including both within- and between-speaker scores, which is what the clustering algorithms see, has a right skew. Assuming, per the speaker recognition literature, that both within- and between-speaker scores can be modeled using respective Gaussian distributions [27], we can use simple measures of symmetry and kurtosis to arrive at the following heuristic to prune away between-speaker edges.

Let Z_A denote the combined distribution of scores for some node A. We keep the subset of scores Z_A^+, or edges, that are greater than some threshold θ_{mm} (i.e., $Z_A^+ = \{z \in Z_A | z > \theta_{\text{mm}}\}$), where θ_{mm} is the largest value such that for the subset of scores $Z_A^- = \{z \in Z_A | z \leq \theta_{\text{mm}}\}$, $\text{mean}(Z_A^-) \leq \text{median}(Z_A^-)$. This method assumes that the mean should be greater than the median in a combined score distribution with a right skew, but without the tail of within-speaker scores, the remaining between-speaker score distribution should be symmetric.

Taking the assumption of between-speaker score Gaussianity a step further, we introduce kurtosis into our local-node pruning. In this case, we choose θ_{kurt} to be the largest score value such that $\text{kurtosis}(Z_A^-) \leq 3$, where 3 is the kurtosis of a normal distribution. Figure 6 shows the cutoff found by kurtosis in magenta, as well as the cutoff, in green, found by the mean-median method above.

In our implementation of this heuristic, we take our full affinity matrix, consisting of all the pairwise PLDA log likelihoods between all of the i-vectors in the data, and sparsify it such that only the top 100 scores for each row (i.e., utterance) have non-zero entries, thus turning it into a 100-NN graph. Then, for each row of the sparse matrix, we use the threshold $\tilde{\theta} = \max\{\theta_{\text{mm}}, \theta_{\text{kurt}}\}$. An edge was pruned away if either node in the edge-pair deemed the connection unnecessary.

A.4. Evaluating Cluster Error

Section 6.1 mentions the use of "speaker confusion error" as a metric to measure clustering performance in addition to the more standard measures of average cluster purity and average speaker fragmentation. We also show plots of the speaker confusion error in Figures 3 and 4. Admittedly, there exist a number of different metrics for evaluating cluster quality, including Precision and Recall, Normalized Mutual Information, F-score, B-cubed, et cetera [29]. We describe the one we chose below, which met our desire for a single number that summarizes the essential aspects of precision, recall, cluster confusion and purity and allows us to seamlessly compare performance across all algorithms and their parameters.

Let our r hypothesized clusters be indexed by i and our s true clusters be indexed by j. We evaluate our clustering output by considering all possible alignments that assign each hypothesized cluster i to exactly one true cluster j. Given such an alignment, say $i \leftrightarrow j$, we define cluster error as the number of elements in hypothesized cluster i whose true cluster is not actually j. Furthermore, any of the $|r - s|$ extra hypothesized or true clusters that did not receive an assignment are also counted as errors. Every possible alignment is considered, and the alignment that provides the smallest clustering error is used. In enforcing a one-to-one assignment of hypothesized-to-true clusters, we are able to summarize both the precision and recall of our clustering output. The procedure described above is equivalent to the evaluation procedure of "speaker confusion error" in the NIST Speaker Diarization task [26].