Transfer learning for cross-lingual automatic speech recognition

Amit Das

***Abstract*—In this study, two instance based transfer learning phoneme modeling approaches are presented to mitigate the effects of limited data in a target language using data from richly resourced source languages. In the first approach, a maximum likelihood (ML) learning criterion is introduced to learn the model parameters of a given phoneme class using data from both the target and source languages. In the second approach, a hybrid learning criterion is introduced using the ML of the target data and the maximum mutual information (MMI) of the training data and the phoneme class labels. This not only takes into account increasing the ML estimates of the models using data from both target and source languages but also improves the discriminative ability of the estimated models using incorrect phoneme class labels.**

***Index Terms*—Transfer learning, maximum likelihood, maximum mutual information**

I. Introduction

WITH the widespread use of hands-free electronic gadgets, speech applications has been gaining more importance throughout the world. The utility of speech technologies like automatic speech recognition (ASR) in these gadgets is dependent on the versatility of ASR systems across users who speak different languages depending on which part of the world they belong to. Hidden Markov Models (HMMs) have gained the widest acceptance in building ASR systems. Ideally, language dependent or monolingual HMMs can be deployed in electronic gadgets where they are expected to be used by a majority of the population speaking the most common language. Although feasible, this is not commercially attractive for two reasons. Firstly, data collection of a specific language is a time consuming and expensive process. Secondly, experienced transcribers who can mark word or phoneme boundaries with a high degree of accuracy may be available only for a limited set of more popular languages like English. Hence, the need arises for building multilingual ASR systems and/or using them for rapid adaptation to a new target (desired) language. In this section, first a brief overview of several techniques used in building multilingual systems are explored followed by a brief explanation of some of the popular language adaptation techniques.

A multilingual ASR system is sometimes known as language independent system since it is versatile across multiple languages. This implies that acoustic-phonetic similarities across languages must be exploited. In [1], multilingual phone modeling was achieved using three approaches. In the first and the most obvious approach, given a set of corpora of multiple languages, language dependent phonemes can be mapped to a new mapping convention such as the WORLDBET [2] that has a wide phonetic symbol coverage across multiple languages. With this, all language dependent transcriptions can be converted to the WORLDBET convention. Therefore, this represents a sematic way of handling multilingual phoneme units. All the transcriptions and speech files from different language corpora are pooled together into one single global multilingual corpus. HMM training can be performed on this global corpus to form language independent acoustic models. The main disadvantage of this approach is that sometimes subtle language dependent variations might be lost during the mapping procedure. For example, monolingual phonemes for the alveolar “r” and palato-alveolar “r” sound differently but they might be represented with the same symbol in two different languages. After mapping to WORLDBET, both the phonemes will be mapped to the same symbol thereby blurring the distinct language properties.

The second approach is a data-driven approach as opposed to the sematic approach described earlier. Here, the phonemes are mapped to a multilingual set using a bottom-up clustering procedure based on log-likelihood distance measure [3] between two phoneme models. The models with least distances are merged together to form a new cluster. Because the estimation of the new phone models of the merged cluster is difficult to achieve, the distance between the two clusters is computed as the maximum of all distances found by pairing a phone model in the first cluster versus another phone model in the second cluster. This “furthest-neighbor” merging heuristic was used to encourage compact clusters and was known to work well empirically. The clustering process continues until all calculated cluster distances are higher than a pre-defined distance threshold or if a specified number of clusters have been formed. The disadvantage with a data-driven approach is that the phoneme models present in a single cluster lose their original phonetic symbol and use a symbol that is the best representation for the cluster. Hence, it is possible that models for the fricatives /s/ and /f/ might be fall in the same cluster whose phonetic symbol may simply be denoted by /f/. Thus, /s/ loses its original semantic representation by using /f/ as its identity which is misleading.

The third approach is a hybrid of the semantic and data driven approaches. Here, all monolingual triphone HMMs that have the same phonetic symbol for a given state (left, center, or right) are pooled together. For example, the Gaussian mixture densities of the phoneme /k/ in state 1 (left) of “cat”, “cut”, “kin”, may be pooled together to form a pool of mixture densities modeling the phoneme /k/. Clustering is performed by taking the a weighted L1-norm of the difference of all possible pairs of mean vectors present in this pool. The motivation behind this is that performing clustering at the level of mixture densities helps retain some distinctive

language dependent properties which are otherwise lost if the clustering were to be performed at the HMM level (as in the second approach). Experiments in [1] indicate that the highest multilingual recognition of isolated words was achieved using the third approach and very little degradation was observed compared to the recognition accuracies of monolingual models.

Often there are scenarios when despite having well trained multilingual phoneme models, the target language that needs to be recognized has no data or very limited data. Recognizing a target language with zero data training data of the target language in a multilingual ASR system is known as cross-language transfer. When limited data is available from the target language, language adaptation of multilingual ASR systems can be useful. This scenario is referred to as cross-lingual recognition or cross-lingual adaptation.

One of the earlier approaches in cross-lingual recognition was to bootstrap or seed acoustic models that were not trained using the target language [4]. In the bootstrapping process, the phoneme set of the target language is mapped to the multilingual phoneme set. Using a limited amount of training data from the target language, the acoustic multilingual acoustic model was retrained with the seed model. Later, [5] showed that such a procedure outperforms models using random seeds even with very few iterations (1-3). It is quite normal to expect that larger the amount of training data of the target language better will be its recognition accuracy. The lower the phonetic dissimilarity between phonemes of the source languages and those of the target language the greater is the recognition accuracy using bootstrapped models [6].

A second approach in cross-lingual adaptation was by using polyphone decision tree specialization (PDTS) [7]. The PDTS method is especially useful for context dependent models. In the PDTS approach, the clustered multilingual polyphone decision tree is adapted to the target language by restarting the decision tree growing process according to the limited amount of training data available from the target language. For example, the non-adapted polyphone decsion tree of a multilingual model may not capture finer variations of the rhotic phoneme "r" if the target language uses several of these variations. Hence, clustering the target language phonemes using the non-adapted tree would result in poorly estimated class models. It was shown in [7] that performance gain using the PDTS method exceeds the gain achieved by using larger adaptation data. Other cross-language adaptation methods include maximum aposteriori (MAP) adaptation [6] using the multilingual acoustic models as the prior model for MAP adaptation.

Recently, [8] proposed a cross-dialectal Gaussian mixture model training criteria to transfer knowledge from Modern Standard Arabic to Levantine Arabic by data sharing. Furthermore, such transfer learning criteria have been successfully implemented in [9] for semi-supervised learning for phone recognition, and prosody detection. This study extends the use of such transfer learning framework for cross-lingual recognition. The rest of the paper is organized as follows. In Section II, the problem definition for training phoneme class models is stated. In Section II-A, a transfer learning algorithm using generative models is explained. In Section II-B, another transfer learning algorithm using a hybrid generative-discriminative models is explained.

II. Algorithm

Let $\mathcal{X}^{(l)}$ comprise of a sequence of observed feature vectors generated from a language with language identity l. Hence, $\mathcal{X}^{(l)} = \left\{\mathbf{x}_t^{(l)}\right\}$ where each vector is subscripted with a time index $t = 1, \ldots, T$ and $\mathbf{x}_t^{(l)} \in \mathcal{R}^D$. Corresponding to $\mathcal{X}^{(l)}$, there are labels in $\mathcal{Y}^{(l)} = \left\{y_t^{(l)}\right\}$ where $y_t^{(l)} \in \left\{1, 2, \ldots, C^{(l)}\right\}$ where $C^{(l)}$ is the total number of phoneme classes in language l. Let $l \in \{1, 2\}$ where $l = 1$ is the language identity for target language and $l = 2$ is the language identity of all the other source languages. The target language is the language whose models are to be estimated. The set of source languages represent all the other languages whose data is shared with the target language in the model estimation process. It is to be noted that there might exist phoneme class labels that may be common across target and source languages. For a test feature $\mathbf{x}^{(1)}$, the Bayes classification rule $f : \mathcal{R}^D \rightarrow \{1, 2, \ldots, C\}$ assigns the class label $\hat{y}$ to $\mathbf{x}^{(1)}$ according to,

$$\begin{aligned} \hat{y} = f(\mathbf{x}^1) &= \underset{y \in \{1,2,\ldots,C\}}{\arg\max}\ p(y|\mathbf{x}^{(1)}) \\ &= \underset{y \in \{1,2,\ldots,C\}}{\arg\max}\ p(\mathbf{x}^{(1)}|y)p(y) \end{aligned} \tag{1}$$

The conditional distribution $p(\mathbf{x}^{(1)}|y)$ is modeled using Gaussian mixture models (GMM) given by,

$$p(\mathbf{x}^{(1)}|y = j; \theta_j) = \sum_{m=1}^{M} \omega_{jm} \mathcal{N}(\mathbf{x}^{(1)}; \mu_{jm}, \mathbf{\Sigma}_{jm}) \tag{2}$$

where $\theta_j = \{\omega_{jm}, \mu_{jm}, \Sigma_{jm}\}_{m=1}^{M}$ represent the parameter set of the model and $\sum_{m=1}^{M} \omega_{jm} = 1$. Here, ω_{jm} represents the weight of the m^{th} Gaussian component density parametrized by the $D \times 1$ mean vector μ_{jm} and $D \times D$ covariance matrix $\mathbf{\Sigma}_{jm}$.

A. ML Based Transfer Learning

The objective is to learn the parameters θ of target language 1 by using *all* data from the distribution $(\mathcal{X}^{(1)}, \mathcal{Y}^{(1)})$ of the low resourced target language and selecting only *relevant* data from other richly resourced languages with distributions $(\mathcal{X}^{(2)}, \mathcal{Y}^{(2)})$. This is the case of *instance based inductive transfer learning* approach. In inductive transfer learning, a few labeled data in the target domain are required as the training data to induce the objective function. The term instance based learning comes from the fact that there are certain parts or instances of source data that can be reused together with the target data. Once we know a good model for the conditional distribution $p(\mathbf{x}^{(1)}|y = j; \theta_j)$, the Bayes rule in (1) can be applied for classification.

Usually, to learn the parameters of a GMM, the objective function to be maximized is the log-likelihood function of the training data. In this work, since the training data consists of both the target and source languages we regularize the

likelihood function of the target data with a regularization term involving the likelihood of the source data. Hence, the new objective function is,

$$\mathcal{J}(\theta_j) = \mathcal{L}(\mathcal{X}^{(1)}|\theta_j) + \alpha\mathcal{L}(\mathcal{X}^{(2)}|\theta_j),\ j = 1, \ldots, C \quad (3)$$

where,

$$\mathcal{L}(\mathcal{X}^{(1)}; \theta_j) = \sum_i \log\ p(\mathbf{x}_i^{(1)}|y_i^{(1)} = j; \theta_j) \quad (4)$$

$$\mathcal{L}(\mathcal{X}^{(2)}; \theta_j) = \sum_i \log\ p(\mathbf{x}_i^{(2)}|y_i^{(2)} \in V_j; \theta_j) \quad (5)$$

The optimal parameter set is given by,

$$\theta_j^\star = \arg\max_{\theta_j} \mathcal{J}(\theta_j)$$

The likelihood probabilities inside the logarithm can be obtained from (2) and α is a constant such that $\alpha < 1$. The auxiliary function for the new objective function becomes,

$$Q(\theta_j, \theta_j^0) = \frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} \sum_{m=1}^{M} p(m|\mathbf{x}_i^{(1)}, j; \theta_j^0) \log\ p(\mathbf{x}_i^{(1)}, m; \theta_j) + \frac{\alpha}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \sum_{m=1}^{M} p(m|\mathbf{x}_i^{(2)}, y_i^{(2)} \in V_j; \theta_j^0) \log\ p(\mathbf{x}_i^{(2)}, m; \theta_j). \quad (6)$$

The auxiliary function is iteratively maximized in an Expectation-Maximization (EM) framework to find the maximum likelihood (ML) parameters. In a given iteration, θ_j is the set of unknown parameters to be estimated and θ_j^0 is the set of known parameters estimated from a previous iteration. Before proceeding on to the next steps, a few notations need clarification here. In the first summand of (6), the term $y_i^{(1)} = j$ in $p(m|\mathbf{x}_i^{(1)}, y_i^{(1)} = j; \theta_j^0)$ is simply replaced by j. However, in the second summand, the labels of the source languages $y_i^{(2)}$ have not been explicitly assigned the class index j similar to the label assignment $y_i^{(1)} = j$ in the target language. A motivation behind doing this is that the semantic representation of a phoneme in the target language may not bear the same semantic represenation in the source language. However, acoustically the two phonemes in the target and source languages may be similar. Hence, such phonemes should not be ignored during training. Therefore, in (6), all phonemes in source languages which are acoustically similar to a phoneme j in the target language belong to the cluster V_j. A detailed discussion regarding the clustering procedure is given in [1].

Under the constraints $\sum_{m=1}^{M} \omega_{jm} = 1$ and $\mathbf{\Sigma}_{jm} \succ 0$, differentiating $Q(\theta_j, \theta_j^0)$ with respect to $\mu_{jm}, \mathbf{\Sigma}_{jm}, \omega_{jm}$, the reestimation equations to find the optimal ML parameters are given as,

$$\omega_{jm} = \frac{\frac{1}{N^{(1)}} n_{jm}^{(1)}(1) + \frac{\alpha}{N^{(2)}} n_{jm}^{(2)}(1)}{1 + \alpha} \quad (7)$$

$$\mu_{jm} = \frac{\frac{1}{N^{(1)}} n_{jm}^{(1)}(\mathbf{x}) + \frac{\alpha}{N^{(2)}} n_{jm}^{(2)}(\mathbf{x})}{\frac{1}{N^{(1)}} n_{jm}^{(1)}(1) + \frac{\alpha}{N^{(2)}} n_{jm}^{(2)}(1)} \quad (8)$$

$$\mathbf{\Sigma}_{jm} = \frac{\frac{1}{N^{(1)}} n_{jm}^{(1)}(\mathbf{x}^2) + \frac{\alpha}{N^{(2)}} n_{jm}^{(2)}(\mathbf{x}^2)}{\frac{1}{N^{(1)}} n_{jm}^{(1)}(1) + \frac{\alpha}{N^{(2)}} n_{jm}^{(2)}(1)} \quad (9)$$

where,

$$n_{jm}^{(l)}(1) = \sum_{i=1}^{N^{(l)}} \gamma_{i,j,m}^{(l)},\ l = 1, 2 \quad (10)$$

$$n_{jm}^{(l)}(\mathbf{x}) = \sum_{i=1}^{N^{(l)}} \gamma_{i,j,m}^{(l)} \mathbf{x}_i^{(l)},\ l = 1, 2 \quad (11)$$

$$n_{jm}^{(l)}(\mathbf{x}^2) = \sum_{i=1}^{N^{(l)}} \gamma_{i,j,m}^{(l)} \Delta_{i,j,m}^{(l)} \Delta_{i,j,m}^{(l)^T}\ l = 1, 2 \quad (12)$$

and,

$$\gamma_{i,j,m}^{(l)} = p(m|\mathbf{x}_i^{(l)}, j; \theta_j^0),\ l = 1, 2 \quad (13)$$

$$\Delta_{i,j,m}^{(l)} = (\mathbf{x}_i^{(l)} - \mu_{jm}),\ l = 1, 2 \quad (14)$$

are the necessary sufficient statistics required for computing the reestimation equations.

Ignoring the superscript in parenthesis for the language identity momentarily, we represent the conditional distribution $p(\mathbf{x}_i^{(1)}|y_i = j; \theta_j)$ as $p(\mathbf{x}_i|y_i; \theta)$. There are three inherent problems with the estimation of the conditional distribution $p(\mathbf{x}_i|y_i; \theta)$. Firstly, the choice of the distribution for real world problems is mostly governed by how well it is mathematical tractable rather than how well it fits the real world data. Even though a GMM can model arbitrary distributions, ambiguities still remain in its prototype design. For example, there exists no well defined procedure to determine the optimal choice of the number of mixtures or the type of covariance matrix (diagonal, full) to be used. Secondly, the estimation method may not produce consistent estimated parameters. Finally, if the amount of training data is limited the quality of the estimated parameters cannot be guaranteed to be reliable. The third point is the most relevant in the current work.

B. Hybrid ML-MMI Based Transfer Learning

One approach to mitigate this problem is to design the classifier directly based on posterior distribution $p(y_i|\mathbf{x}_i; \theta)$ (instead of the conditional distribution $p(\mathbf{x}_i|y_i; \theta)$) since the former is used as the optimal rule for classification (1) . A popular approach for training the posterior distribution is the Maximum Mutual Information Estimation (MMIE) based training originally proposed in [10]. A brief explanation of why the MMIE training is equivalent to training the posterior distribution $p(y_i|\mathbf{x}_i; \theta)$ is presented here. For a sequence of feature vectors and their corresponding labels, the joint posterior distribution is given as,

$$\prod_i p(y_i|\mathbf{x}_i; \theta) = \prod_i \frac{p(y_i, \mathbf{x}_i; \theta)}{p(\mathbf{x}_i; \theta)} \quad (15)$$

$$= \prod_i \frac{p(\mathbf{x}_i|y_i; \theta) p(y_i)}{\sum_{y_i} p(\mathbf{x}_i|y_i; \theta) p(y_i)} \quad (16)$$

The MMI between $\{\mathbf{x}_i, y_i\}_{i=1}^N$ is given by,

$$I(\mathbf{x}_1, \ldots, \mathbf{x}_N, y_1, \ldots, y_N; \theta) \propto \prod_i \frac{p(y_i, \mathbf{x}_i; \theta)}{p(\mathbf{x}_i; \theta) p(y_i)} \quad (17)$$

If $p(y_i)$ is treated as a constant, then (15) is equivalent to (17). In (16), the numerator term contains the ML term of $\mathbf{x}_i$ given its true phoneme class label y_i while the denominator term contains the sum of likelihoods all phoneme class labels (both correct and incorrect labels). During training, the true phoneme class label for $\mathbf{x}_i$ is known. Therefore, maximizing (16) is equivalent to maximizing the ML term in the numerator while simultaneously minimizing the denominator term. The denominator term can be simultaneosuly minimized if the ML terms of $\mathbf{x}_i$ given the *incorrect* phoneme class labels can be minimized. This implies θ_{y_i} is modeled in such a way that it attempts to increase the ML score of the true phoneme class while decreasing the ML score of the incorrect phoneme classes. Hence, it is capable of incorporating discriminating ability.

In this work, the motivation behind incorporating MMIE for learning model θ_j is that since the target language data is limited it might benefit to make use of those data points of source languages that are likely to have the same phoneme class j. This is similar to the ML learning procedure described in Section II-A. However, MMIE additionally can even make use of other data points (of source languages) that are unlikely to have the same phoneme class j to incorporate discrimination against models of classes other than the j^{th} class. Keeping these in mind, the new objective function is designed as,

$$\mathcal{J}(\theta) = \log\, p(\mathcal{X}^{(1)}|\mathcal{Y}^{(1)};\theta) + \alpha \log\, p(\mathcal{Y}^{(2)}|\mathcal{X}^{(2)};\theta). \quad (18)$$

The optimal parameter set is given by,

$$\theta^{\star} = \arg\max_{\theta} \mathcal{J}(\theta)$$

The corresponding weak sense auxiliary function [11] is,

$$\begin{aligned} Q(\theta,\theta^0) &= Q^{(1)}_{num}(\theta,\theta^0) + \alpha Q^{(2)}_{num}(\theta,\theta^0) \\ &\quad - \alpha Q^{(2)}_{den}(\theta,\theta^0) + Q_{sm}(\theta,\theta^0) \end{aligned} \quad (19)$$

where the term $Q^{(1)}_{num}$ is the strong sense auxiliary function of the target language. The terms $Q^{(2)}_{num}$ and $Q^{(2)}_{den}$ correspond to the strong sense auxiliary functions of the source languages. The term Q_{sm} is a strong sense auxiliary function to increase the concavity of overall auxiliary function $Q(\theta,\theta^0)$ around the local optimum. Expanding the first auxiliary function, we get,

$$\begin{aligned} Q^{(1)}_{num} &= \frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} \sum_{m=1}^{M} p(m|\mathbf{x}^{(1)}_i, y_i; \theta^0_j) \log\, p(\mathbf{x}^{(1)}_i, m; \theta_j) \\ &= \frac{1}{N^{(1)}} \sum_{i=1}^{N^{(1)}} \sum_{m=1}^{M} \gamma^{(1)}_{i,y_i,m} \log\, (\omega_{y_i,m} \mathcal{N}(\mathbf{x}^{(1)}; \mu_{y_i,m}, \mathbf{\Sigma}_{y_i,m})) \end{aligned} \quad (20)$$

Rewriting (20) where it contains the parameters of only the j^{th} class,

$$Q^{(1)}_{num} = \frac{1}{N^{(1)}} \sum_{\substack{i=1 \\ i:y_i \in V_j}}^{N^{(1)}} \sum_{m=1}^{M} \gamma^{(1)}_{i,j,m} \log\, (\omega_{j,m} \mathcal{N}(\mathbf{x}^{(1)}; \mu_{j,m}, \mathbf{\Sigma}_{j,m})) \quad (21)$$

Differentiating (21) with respect to $\omega_{jm}, \mu_{jm}, \mathbf{\Sigma}_{jm}$, we get,

$$\frac{\partial Q^{(1)}_{num}}{\partial \omega_{jm}} = \frac{1}{N^{(1)}} \sum_{\substack{i=1 \\ i:y_i \in V_j}}^{N^{(1)}} \gamma^{(1)}_{i,j,m} \omega^{-1}_{jm} \quad (22)$$

$$\frac{\partial Q^{(1)}_{num}}{\partial \mu_{jm}} = \frac{1}{N^{(1)}} \sum_{\substack{i=1 \\ i:y_i \in V_j}}^{N^{(1)}} \gamma^{(1)}_{i,j,m} \mathbf{\Sigma}^{-1}_{jm} \Delta^{(1)}_{i,j,m} \quad (23)$$

$$\frac{\partial Q^{(1)}_{num}}{\partial \mathbf{\Sigma}_{jm}} = \frac{1}{N^{(1)}} \sum_{\substack{i=1 \\ i:y_i \in V_j}}^{N^{(1)}} \gamma^{(1)}_{i,j,m} (1 - \Delta^{(1)}_{i,j,m} \Delta^{(1)^T}_{i,j,m} \mathbf{\Sigma}^{-1}_{jm}) \quad (24)$$

where, $\gamma^{(1)}_{i,j,m}, \Delta^{(1)}_{i,j,m}$ are as defined in (13) and (14) respectively. For the case of $Q^{(2)}_{num}$, an identical set of equations can be genrated by replacing the superscript (1) with (2) to indicate the use of the source languages instead of the target language. Next, expanding $Q^{(2)}_{den}$, we get,

$$Q^{(2)}_{den} = \frac{1}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \sum_{y=1}^{C} \sum_{m=1}^{M} p(m, y|\mathbf{x}^{(2)}_i; \theta^0_j) \log\, p(\mathbf{x}^{(2)}_i, y, m; \theta_j). \quad (25)$$

It may be noted that in (25) every data point $\mathbf{x}^{(2)}_i$ is evaluated across all classes and not just its own class y_i. Rewriting (25) in terms of j^{th} class, we get,

$$\begin{aligned} Q^{(2)}_{den} &= \frac{1}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \sum_{m=1}^{M} p(m, j|\mathbf{x}^{(2)}_i; \theta^0) \log\, p(\mathbf{x}^{(2)}_i, j, m; \theta) \\ &= \frac{1}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \sum_{m=1}^{M} p(m, j|\mathbf{x}^{(2)}_i; \theta^0) \log\, p(m|j;\theta) p(\mathbf{x}^{(2)}_i|j, m; \theta) \\ &= \frac{1}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \sum_{m=1}^{M} p(j|\mathbf{x}^{(2)}_i; \theta^0) p(m|\mathbf{x}^{(2)}_i, j; \theta^0) \times \\ &\qquad \log\, p(m|j;\theta) p(\mathbf{x}^{(2)}_i|j, m; \theta) \\ &= \frac{1}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \sum_{m=1}^{M} \xi^{(2)}_{i,j} \gamma^{(2)}_{i,j,m} \log\, (\omega_{j,m} \mathcal{N}(\mathbf{x}^{(2)}; \mu_{j,m}, \mathbf{\Sigma}_{j,m})) \end{aligned} \quad (26)$$

where in going from the first step to the second step the term $p(j;\theta)$ inside the logarithm has been ignored since it is a constant and disappears during differentiation of $Q^{(2)}_{den}$. Furthermore, the term $\xi^{(2)}_{i,j} = p(j|\mathbf{x}^{(2)}_i; \theta^0)$ is defined as,

$$\begin{aligned} \xi^{(2)}_{i,j} &= \frac{p(\mathbf{x}^{(2)}_i|j;\theta^0) p(j;\theta^0)}{\sum_j p(\mathbf{x}^{(2)}_i|j;\theta^0) p(j;\theta^0)} \\ &= \frac{p(\mathbf{x}^{(2)}_i; \theta^0_j)}{\sum_k p(\mathbf{x}^{(2)}_i; \theta^0_k)} \end{aligned} \quad (27)$$

is simply the ratio of maximum likelihood score of $\mathbf{x}^{(2)}_i$ with respect to θ^0_j and sum of maximum likelihood scores of $\mathbf{x}^{(2)}_i$ with respect to the model of all classes. It is assumed that classes have uniform priors $p(j;\theta^0)$. Differentiating (21) with

respect to $\omega_{jm}, \mu_{jm}, \mathbf{\Sigma}_{jm}$, we get,

$$\frac{\partial Q_{den}^{(2)}}{\partial \omega_{jm}} = \frac{1}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \xi_{i,j}^{(2)} \gamma_{i,j,m}^{(2)} \omega_{jm}^{-1} \tag{28}$$

$$\frac{\partial Q_{den}^{(2)}}{\partial \mu_{jm}} = \frac{1}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \xi_{i,j}^{(2)} \gamma_{i,j,m}^{(2)} \mathbf{\Sigma}_{jm}^{-1} \Delta_{i,j,m}^{(2)} \tag{29}$$

$$\frac{\partial Q_{den}^{(2)}}{\partial \mathbf{\Sigma}_{jm}} = \frac{1}{N^{(2)}} \sum_{i=1}^{N^{(2)}} \xi_{i,j}^{(2)} \gamma_{i,j,m}^{(2)} (1 - \Delta_{i,j,m}^{(2)} \Delta_{i,j,m}^{(2)^T} \mathbf{\Sigma}_{jm}^{-1}) \tag{30}$$

Collecting all the partial derivatives for ω_{jm} ((22), (28)), μ_{jm} ((23), (29)), and $\mathbf{\Sigma}_{jm}$ ((24), (30)) and equating them to 0, we get the reestimation equations as,

$$\omega_{jm} = \frac{\frac{1}{N^{(1)}} n_{jm}^{\prime(1)}(1) + \frac{\alpha}{N^{(2)}} n_{jm}^{\prime(2)}(1) - \frac{\alpha}{N^{(2)}} n_{jm}^{\prime\prime(2)}(1) + D_{jm}\omega_{jm}^{0}}{\frac{1}{N^{(1)}} n_{j}^{\prime(1)}(1) + \frac{\alpha}{N^{(2)}} n_{j}^{\prime(2)}(1) - \frac{\alpha}{N^{(2)}} n_{j}^{\prime\prime(2)}(1) + D_{jm}\omega_{jm}^{0}} \tag{31}$$

$$\mu_{jm} = \frac{\frac{1}{N^{(1)}} n_{jm}^{\prime(1)}(\mathbf{x}) + \frac{\alpha}{N^{(2)}} n_{jm}^{\prime(2)}(\mathbf{x}) - \frac{\alpha}{N^{(2)}} n_{jm}^{\prime\prime(2)}(\mathbf{x}) + D_{jm}\mu_{jm}^{0}}{\frac{1}{N^{(1)}} n_{jm}^{\prime(1)}(1) + \frac{\alpha}{N^{(2)}} n_{jm}^{\prime(2)}(1) - \frac{\alpha}{N^{(2)}} n_{jm}^{\prime\prime(2)}(1) + D_{jm}\mu_{jm}^{0}} \tag{32}$$

$$\mathbf{\Sigma}_{jm} = \frac{\frac{1}{N^{(1)}} n_{jm}^{\prime(1)}(\mathbf{x}^2) + \frac{\alpha}{N^{(2)}} n_{jm}^{\prime(2)}(\mathbf{x}^2) - \frac{\alpha}{N^{(2)}} n_{jm}^{\prime\prime(2)}(\mathbf{x}^2) + D_{jm}\mathbf{\Sigma}_{jm}^{0}}{\frac{1}{N^{(1)}} n_{jm}^{\prime(1)}(1) + \frac{\alpha}{N^{(2)}} n_{jm}^{\prime(2)}(1) - \frac{\alpha}{N^{(2)}} n_{jm}^{\prime\prime(2)}(1) + D_{jm}\mathbf{\Sigma}_{jm}^{0}} \tag{33}$$

where,

$$\begin{aligned}
n_{jm}^{\prime(l)}(1) &= \sum_{\substack{i=1 \\ i:y_i \in V_j}}^{N^{(l)}} \gamma_{i,j,m}^{(l)}, \ l = 1,2 \\
n_{jm}^{\prime\prime(2)}(1) &= \sum_{i=1}^{N^{(2)}} \xi_{i,j}^{(2)} \gamma_{i,j,m}^{(2)} \\
n_{j}^{\prime(l)}(1) &= \sum_{m=1}^{M} n_{jm}^{\prime(l)}(1), \ l = 1,2 \\
n_{j}^{\prime\prime(2)}(1) &= \sum_{m=1}^{M} n_{jm}^{\prime\prime(2)}(1) \\
n_{jm}^{\prime(l)}(\mathbf{x}) &= \sum_{\substack{i=1 \\ i:y_i \in V_j}}^{N^{(l)}} \gamma_{i,j,m}^{(l)} \mathbf{x}_i^{(l)}, \ l = 1,2 \\
n_{jm}^{\prime\prime(2)}(\mathbf{x}) &= \sum_{i=1}^{N^{(2)}} \xi_{i,j}^{(2)} \gamma_{i,j,m}^{(2)} \mathbf{x}_i^{(2)}, \\
n_{jm}^{\prime(l)}(\mathbf{x}^2) &= \sum_{\substack{i=1 \\ i:y_i \in V_j}}^{N^{(l)}} \gamma_{i,j,m}^{(l)} \Delta_{i,j,m}^{(l)} \Delta_{i,j,m}^{(l)^T}, \ l = 1,2 \\
n_{jm}^{\prime\prime(2)}(\mathbf{x}^2) &= \sum_{i=1}^{N^{(2)}} \xi_{i,j}^{(2)} \gamma_{i,j,m}^{(2)} \Delta_{i,j,m}^{(2)} \Delta_{i,j,m}^{(2)^T},
\end{aligned} \tag{34}$$

Selection of the D_{jm} is critical in that $D \geq D_{min}$ guarantees $p(\mathcal{Y}^{(2)}|\mathcal{X}^{(2)};\theta) \geq p(\mathcal{Y}^{(2)}|\mathcal{X}^{(2)};\theta^0)$. A discussion on the selection of D_{jm} is given in [12].

REFERENCES

[1] J. Kohler, "Multilingual phone models for vocabulary-independent speech recognition tasks," *Speech Communication.*, vol. 35, no. 1-2, pp. 21–30, Aug. 2001.

[2] J. L. Hieronymus, "Ascii phonetic symbols for the world's languages: Worldbet," Bell Labs Technical Memorandum, Tech. Rep.

[3] B. H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden markov models," AT&T Technical Journal, Tech. Rep. 2.

[4] B. Wheatley, K. Kondo, W. Anderson, and Y. Muthusamy, "An evaluation of cross-language adaptation for rapid hmm development in a new language," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*

[5] T. Schultz and A. Waibel, "Fast bootstrapping of lvcsr systems with multilingual phoneme sets," in *Eurospeech.*, 1997.

[6] J. Kohler, "Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1998, vol. 1, pp. 417–420.

[7] T. Schultz and A. Waibel, "Polyphone decision tree specialization for language adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2000, vol. 3, pp. 1707–1710.

[8] P. Huang and M. Hasegawa-Johnson, "Cross-dialectal data transferring for gaussian mixture model training in arabic speech recognition," *4th International Conference on Arabic Language Processing*, pp. 119–123, 2012.

[9] J.-T. Huang, "Semi-supervised learning for acoustic and prosodic modeling in speech applications," Ph.D. dissertation, University of Illinois at Urbana-Champaign.

[10] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1986, pp. 49–52.

[11] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge University.

[12] P. Woodland and D. Povey, "Large scale discriminative training of hidden markov models for speech recognition," *Computer Speech and Language.*, vol. 16.