Unsupervised Model Adaptation using Information-Theoretic Criterion

Ariya Rastrow[1], **Frederick Jelinek**[1], **Abhinav Sethy**[2] **and Bhuvana Ramabhadran**[2]
[1]Human Language Technology Center of Excellence, and
Center for Language and Speech Processing, Johns Hopkins University
{ariya, jelinek}@jhu.edu
[2]IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
{asethy, bhuvana}@us.ibm.com

Abstract

In this paper we propose a novel general framework for unsupervised model adaptation. Our method is based on entropy which has been used previously as a regularizer in semi-supervised learning. This technique includes another term which measures the stability of posteriors w.r.t model parameters, in addition to conditional entropy. The idea is to use parameters which result in both low conditional entropy and also stable decision rules. As an application, we demonstrate how this framework can be used for adjusting language model interpolation weight for speech recognition task to adapt from Broadcast news data to MIT lecture data. We show how the new technique can obtain comparable performance to completely supervised estimation of interpolation parameters.

1 Introduction

All statistical and machine learning techniques for classification, in principle, work under the assumption that

1. A reasonable amount of training data is available.

2. Training data and test data are drawn from the same underlying distribution.

In fact, the success of statistical models is crucially dependent on training data. Unfortunately, the latter assumption is not fulfilled in many applications. Therefore, model adaptation is necessary when training data is not matched (not drawn from same distribution) with test data. It is often the case where we have plenty of labeled data for one specific domain/genre (source domain) and little amount of labeled data (or no labeled data at all) for the desired domain/genre (target domain). Model adaptation techniques are commonly used to address this problem. Model adaptation starts with trained models (trained on source domain with rich amount of labeled data) and then modify them using the available labeled data from target domain (or instead unlabeled data). A survey on different methods of model adaptation can be found in (Jiang, 2008).

Information regularization framework has been previously proposed in literature to control the label conditional probabilities via input distribution (Szummer and Jaakkola, 2003). The idea is that labels should not change too much in dense regions of the input distribution. The authors use the mutual information between input features and labels as a measure of label complexity. Another framework previously suggested is to use label entropy (conditional entropy) on unlabeled data as a regularizer to Maximum Likelihood (ML) training on labeled data (Grandvalet and Bengio, 2004).

Availability of resources for the target domain categorizes these techniques into either *supervised* or *unsupervised*. In this paper we propose a general framework for unsupervised adaptation using Shannon entropy and stability of entropy. The assumption is that in-domain and out-of-domain distributions are not too different such that one can improve the performance of initial models on in-domain data by little adjustment of initial decision boundaries (learned on out-of-domain data).

2 Conditional Entropy based Adaptation

In this section, conditional entropy and its relation to classifier performance are first described. Next, we introduce our proposed objective function for domain adaptation.

2.1 Conditional Entropy

Considering the classification problem where $\mathbf{X}$ and $\mathbf{Y}$ are the input features and the corresponding class labels respectively, the conditional entropy is a measure of the class overlap and is calculated as follows

$$H(\mathbf{Y}|\mathbf{X}) = E_{\mathbf{X}}[H(\mathbf{Y}|\mathbf{X}=x)] = -\int p(x)\left(\sum_y p(y|x)\log p(y|x)\right)dx \quad (1)$$

Through *Fano's Inequality* theorem, one can see how conditional entropy is related to classification performance.

Theorem 1 (Fano's Inequality) *Suppose* $P_e = P\{\hat{\mathbf{Y}} \neq \mathbf{Y}\}$ *where* $\hat{\mathbf{Y}} = g(X)$ *are the assigned labels for the data points, based on the classification rule. Then*

$$P_e \geq \frac{H(\mathbf{Y}|\mathbf{X})-1}{\log(|\mathcal{Y}|-1)}$$

where $\mathcal{Y}$ *is the number of possible classes and* $H(Y|X)$ *is the conditional entropy with respect to true distibution.*

The proof to this theorem can be found in (Cover and Thomas, 2006). This inequality indicates that $\mathbf{Y}$ can be estimated with low probability of error only if the conditional entropy $H(\mathbf{Y}|\mathbf{X})$ is small.

Although the above theorem is useful in a sense that it connects the classification problem to Shannon entropy, the true distributions are almost never known to us[1]. In most classification methods, a specific model structure for the distributions is assumed and the task is to estimate the model parameters within the assumed model space. Given the model structure and parameters, one can modify *Fano's Inequality* as follows,

Corollary 1

$$P_e(\theta) = P\{\hat{\mathbf{Y}} \neq \mathbf{Y}\ |\theta\} \geq \frac{H_\theta(\mathbf{Y}|\mathbf{X})-1}{\log(|\mathcal{Y}|-1)} \quad (2)$$

where $P_e(\theta)$ *is the classifier probability of error given model parameters,* θ *and*

$$H_\theta(\mathbf{Y}|\mathbf{X}) = -\int p(x)\left(\sum_y p_\theta(y|x)\log p_\theta(y|x)\right)dx$$

Here, $H_\theta(\mathbf{Y}|\mathbf{X})$ *is the conditional entropy imposed by model parameters.*

Eqn. 2 indicates the fact that models with low conditional entropy are preferable. However, a low entropy model does not necessarily have good performance (this will be reviewed later on) [2]

2.2 Objective Function

Minimization of conditional entropy as a framework in the classification task is not a new concept and has been tried by researchers. In fact, (Grandvalet and Bengio, 2004) use this along with the maximum likelihood criterion in a semi-supervised set up such that parameters with both maximum likelihood on labeled data and minimum conditional entropy on unlabeled data are chosen. By minimizing the entropy, the method assumes a prior which prefers minimal class overlap. Entropy minimization is used in (Li et al., 2004) as an unsupervised non-parametric clustering method and is shown to result in significant improvement over k-mean, hierarchical clustering and etc.

These methods are all based on the fact that models with low conditional entropy have their decision boundaries passing through low-density regions of the input distribution, $P(X)$. This is consistent with the assumption that classes are well separated so that one can expect to take advantage of unlabeled examples (Grandvalet and Bengio, 2004).

In many cases shifting from one domain to another domain, initial trained decision boundaries (on

[1] In fact, Theorem 1 shows how relevant the input features are for the classification task by putting a lower bound on the best possible classifier performance. As the overlap between features from different classes increases, conditional entropy increases as well, thus lowering the performance of the best possible classifier.

[2] Imagine a model which classifies any input as class 1. Clearly for this model $H_\theta(\mathbf{Y}|\mathbf{X}) = 0$.

out-of-domain data) result in high conditional entropy for the new domain, due to mismatch between distributions. Therefore, there is a need to adjust model parameters such that decision boundaries goes through low-density regions of the distribution. This motivates the idea of using minimum conditional entropy criterion for adapting to a new domain. At the same time, two domains are often close enough that one would expect that the optimal parameters for the new domain should not deviate too much from initial parameters. In order to formulate the technique mentioned in the above paragraph, let us define Θ_{init} to be the initial model parameters estimated on out-of-domain data (using labeled data). Assuming the availability of enough amount of unlabeled data for in-domain task, we try to minimize the following objective function w.r.t the parameters,

$$\theta_{\mathbf{new}} = \underset{\theta}{\operatorname{argmin}}\, H_\theta(\mathbf{Y}|\mathbf{X}) + \lambda \left|\left|\theta - \theta_{\mathbf{init}}\right|\right|_p \tag{3}$$

where $\left|\left|\theta - \theta_{\mathbf{init}}\right|\right|_p$ is an L_p regularizer and tries to prevent parameters from deviating too much from their initial values[3].

Once again the idea here is to adjust the parameters (using unlabeled data) such that low-density separation between the classes is achieved. In the following section we will discuss the drawback of this objective function for adaptation in realistic scenarios.

3 Issues with Minimum Entropy Criterion

It is discussed in Section 2.2 that the model parameters are adapted such that a minimum conditional entropy is achieved. It was also discussed how this is related to finding decision boundaries through low-density regions of input distribution. However, the obvious assumption here is that the classes are well separated and there in fact exists low-density regions between classes which can be treated as boundaries. Although this is a suitable/ideal assumption for classification, in most practical problems this assumption is not satisfied and often classes overlap. Therefore, we can not expect the conditional entropy to be convex in this situation and to achieve minimization w.r.t parameters (other than the trivial solutions).

Let us clarify this through an example. Consider X to be generated by mixture of two 2-D Gaussians (each with a particular mean and covariance matrix) where each Gaussian corresponds to a particular class (binary class situation) . Also in order to have linear decision boundaries, let the Gaussians have same covariance matrix and let the parameter being estimated be the prior for class 1, $P(Y = 1)$. Fig. 1 shows two different situations with overlapping classes and non-overlapping classes. The left panel shows a distribution in which classes are well separated whereas the right panel corresponds to the situation where there is considerable overlap between classes. Clearly, in the later case there is no low-density region separating the classes. Therefore, as we change the parameter (here, the prior on the class $Y = 1$), there will not be any well defined point with minimum entropy. This can be seen from Fig. 2 where model conditional entropy is plotted vs. class prior parameter for both cases. In the case of no-overlap between classes, entropy is a convex function w.r.t the parameter (excluding trivial solutions which happens at $P(Y = 1) = 0, 1$) and is minimum at $P(Y = 1) = 0.7$ which is the true prior with which the data was generated.

We summarize issues with minimum entropy criterion and our proposed solutions as follows:

- Trivial solution: this happens when we put decision boundaries such that both classes are considered as one class (this can be avoided using the regularizer in Eqn. 3 and the assumption that initial models have a reasonable solution, e.g. close to the optimal solution for new domain)

- Overlapped Classes: As it was discussed in this section, if the overlap is considerable then the entropy will not be convex w.r.t to model parameters. We will address this issue in the next section by introducing the entropy-stability concept.

4 Entropy-Stability

It was discussed in the previous section that a minimum entropy criterion can not be used (by itself) in

[3]The other reason for using a regularizer is to prevent trivial solutions of minimum entropy criterion

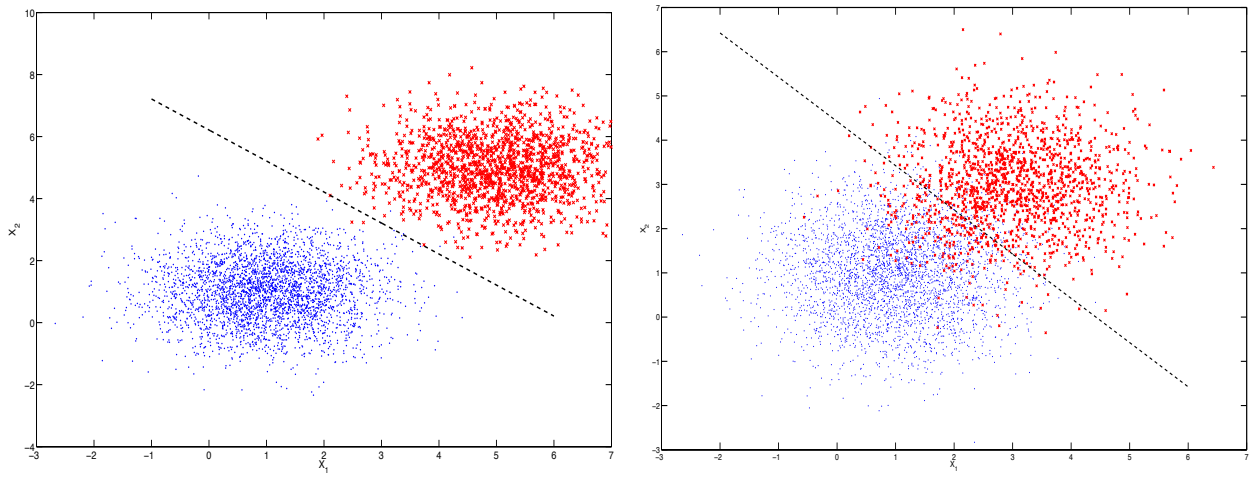

Figure 1: Mixture of two Gaussians and the corresponding Bayes decision boundary: (left) with no class overlap (right) with class overlap

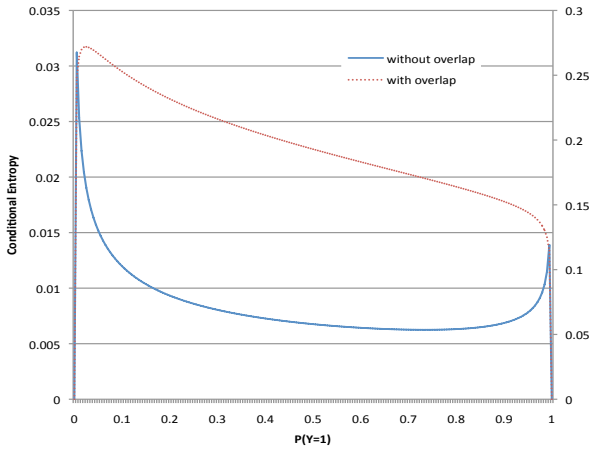

Figure 2: Condtional entropy vs. prior parameter, $P(Y = 1)$

situations where there is a considerable amount of overlap among classes. Assuming that class boundaries happen in the regions close to the tail of class distributions, we introduce the concept of *Entropy-Stability* and show how it can be used to detect boundary regions. Define *Entropy-Stability* to be the reciprocal of the following

$$\left|\left|\frac{\partial H_\theta(\mathbf{Y}|\mathbf{X})}{\partial \theta}\right|\right|_p = \left|\left|\int p(x) \frac{\partial \left(\sum_y p_\theta(y|x) \log p_\theta(y|x)\right)}{\partial \theta} dx\right|\right|_p \quad (4)$$

Recall: since θ is a vector of parameters, $\frac{\partial H_\theta(\mathbf{Y}|\mathbf{X})}{\partial \theta}$ will be a vector and by using L_p norm Entropy-stability will be a scalar.

The introduced concept basically measures the stability of label entropies w.r.t the model parameters. The idea is that we prefer models which not only have low-conditional entropy but also have stable decision rules imposed by the model. Next, we show through the following theorem how Entropy-Stability measures the stability over posterior probabilities (decision rules) of the model.

Theorem 2

$$\left|\left|\frac{\partial H_\theta(\mathbf{Y}|\mathbf{X})}{\partial \theta}\right|\right|_p = \left|\left|\int p(x) \left(\sum_y \frac{\partial p_\theta(y|x)}{\partial \theta} \log p_\theta(y|x)\right) dx\right|\right|_p$$

where the term inside the parenthesis is the weighted sum (by log-likelihood) over the gradient of posterior probabilities of labels for a given sample x

Proof The proof is straight forward and uses the fact that $\sum \frac{\partial p_\theta(y|x)}{\partial \theta} = \frac{\partial (\sum p_\theta(y|x))}{\partial \theta} = 0$.

Using Theorem 2 and Eqn. 4, it should be clear how Entropy-Stability measures the expected stability over the posterior probabilities of the model. A high value of $\left|\left|\frac{\partial H_\theta(\mathbf{Y}|\mathbf{X})}{\partial \theta}\right|\right|_p$ implies models with less stable decision rules. In order to explain how this is used for detecting boundaries (overlapped

regions) we once again refer back to our mixture of Gaussians' example. As the decision boundary moves from class specific regions to overlapped regions (by changing the parameter which is here class prior probability) we expect the entropy to continuously decrease (due to the assumption that the overlaps occur at the tail of class distributions). However, as we get close to the overlapping regions the added data points from other class(es) will resist changes in the entropy. resulting in stability over the entropy until we enter the regions specific to other class(es).

In the following subsection we use this idea to propose a new objective function which can be used as an unsupervised adaptation method even for the case of input distribution with overlapping classes.

4.1 Better Objective Function

The idea here is to use the Entropy-Stability concept to accept only regions which are close to the overlapped parts of the distribution (based on our assumption, these are valid regions for decision boundaries) and then using the minimum entropy criterion we find optimum solutions for our parameters inside these regions. Therefore, we modify Eqn. 3 such that it also includes the Entropy-Stability term

$$\theta_{\mathbf{new}} = \underset{\theta}{\operatorname{argmin}} \left(H_\theta(\mathbf{Y}|\mathbf{X}) + \gamma \left|\left| \frac{\partial H_\theta(\mathbf{Y}|\mathbf{X})}{\partial \theta} \right|\right|_{p'} + \lambda \left|\left|\theta - \theta_{\mathbf{init}}\right|\right|_p \right) \tag{5}$$

The parameter γ and λ can be tuned using small amount of labeled data (Dev set).

5 Speech Recognition Task

In this section we will discuss how the proposed framework can be used in a speech recognition task. In the speech recognition task, Y is the sequence of words and X is the input speech signal. For a given speech signal, almost every word sequence is a possible output and therefore there is a need for a compact representation of output labels (words). For this, word graphs (Lattices) are generated during the recognition process. In fact, each lattice is an *acyclic directed graph* whose nodes correspond to particular instants of time, and arcs (edges connecting nodes) represent possible word hypotheses. Associated with each arc is an acoustic likelihood and language model likelihood scores. Fig. 3 shows an example of recognition lattice [4] (for the purpose of demonstration likelihood scores are not shown).

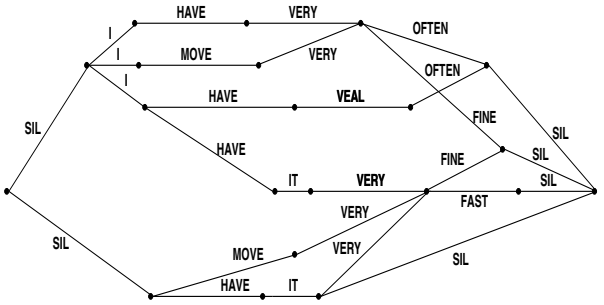

Figure 3: Lattice Example

Since lattices contain all the likely hypotheses (unlikely hypotheses are pruned during recognition and will not be included in the lattice), conditional entropy for any given input speech signal, x, can be approximated by the conditional entropy of the lattice. That is,

$$H_\theta(\mathbf{Y}|\mathbf{X} = x_i) = H_\theta(\mathbf{Y}|\mathcal{L}_i)$$

where $\mathcal{L}_i$ is the corresponding decoded lattice (given speech recognizer parameters) of utterance x_i.

For the calculation of entropy we need to know the distribution of X because $H_\theta(\mathbf{Y}|\mathbf{X}) = E_X[H_\theta(\mathbf{Y}|\mathbf{X} = x)]$ and since this distribution is not known to us, we use *Law of Large Numbers* to approximate it by the empirical average

$$H_\theta(\mathbf{Y}|\mathbf{X}) \approx -\frac{1}{N} \sum_{i=1}^{N} \sum_{y \in \mathcal{L}_i} p_\theta(y|\mathcal{L}_i) \log p_\theta(y|\mathcal{L}_i) \tag{6}$$

Here N indicates the number of unlabeled utterances for which we calculate the empirical value of conditional entropy. Similarly, expectation w.r.t input distribution in entropy-stability term is also approximated by the empirical average of samples.

Since the number of paths (hypotheses) in the lattice is very large, it would be computationally infeasible to compute the conditional entropy by enumerating all possible paths in the lattice and calculating

[4] The figure is adopted from (Mangu et al., 1999)

Element	$\langle p, r \rangle$
$\langle p_1, r_1 \rangle \otimes \langle p_2, r_2 \rangle$	$\langle p_1 p_2,\ p_1 r_2 + p_2 r_1 \rangle$
$\langle p_1, r_1 \rangle \oplus \langle p_2, r_2 \rangle$	$\langle p_1 + p_2,\ r_1 + r_2 \rangle$
0	$\langle 0, 0 \rangle$
1	$\langle 1, 0 \rangle$

Table 1: **First-Order (Expectation) semiring**: Defining *multiplication* and *sum* operations for first-order semirings.

their corresponding posterior probabilities. Instead we use Finite-State Transducers (FST) to represent the hypothesis space (lattice). To calculate entropy and the gradient of entropy, the weights for the FST are defined to be First- and Second-Order *semirings* (Li and Eisner, 2009). The idea is to use *semirings* and their corresponding operations along with the forward-backward algorithm to calculate first- and second-order statistics to compute entropy and the gradient of entropy respectively. Assume we are interested in calculating the entropy of the lattice,

$$\begin{aligned} H(p) &= -\sum_{d \in \mathcal{L}_i} \frac{p(d)}{Z} \log(\frac{p(d)}{Z}) \\ &= \log Z - \frac{1}{Z} \sum_{d \in \mathcal{L}_i} p(d) \log p(d) \\ &= \log Z - \frac{\bar{r}}{Z} \end{aligned} \tag{7}$$

where Z is the total probability of all the paths in the lattice (normalization factor). In order to do so, we need to compute $\langle Z, \bar{r} \rangle$ on the lattice. It can be proved that if we define the first-order semiring $\langle p_e, p_e \log p_e \rangle$ (p_e is the non-normalized score of each arc in the lattice) as our FST weights and define semiring operations as in Table. 1, then applying the forward algorithm will result in the calculation of $\langle Z, \bar{r} \rangle$ as the weight (semiring weight) for the final node.

The details for using Second-Order *semirings* for calculating the gradient of entropy can be found in (Li and Eisner, 2009). The same paper describes how to use the forward-backward algorithm to speed-up the this procedure.

6 Language Model Adaptation

Language Model Adaptation is crucial when the training data does not match the test data being decoded. This is a frequent scenario for all Automatic Speech Recognition (ASR) systems. The application domain very often contains named entities and N-gram sequences that are unique to the domain of interest. For example, conversational speech has a very different structure than class-room lectures. Linear Interpolation based methods are most commonly used to adapt LMs to a new domain. As explained in (Bacchiani et al., 2003), linear interpolation is a special case of Maximum A Posterior (MAP) estimation, where an N-gram LM is built on the adaptation data from the new domain and the two LMs are combined using:

$$p(w_i|h) = \lambda p_B(w_i|h) + (1-\lambda) p_A(w_i|h) \qquad 0 \leq \lambda \leq 1$$

where p_B refers to out-of-domain (background) models and p_A is the adaptation (in-domain) models. Here λ is the interpolation weight.

Conventionally, λ is calculated by optimizing perplexity (PPL) or Word Error Rate (WER) on some held-out data from target domain. Instead using our proposed framework, we estimate λ on enough amount of unlabeled data from target domain. The idea is that resources on the new domain have already been used to build domain specific models and it does not make sense to again use in-domain resources for estimating the interpolation weight. Since we are trying to just estimate one parameter and the performance of the interpolated model is bound by in-domain/out-of-domain models, there is no need to include a regularization term in Eqn. 5. Also $\left|\left|\frac{\partial H_\theta(\mathbf{Y}|\mathbf{X})}{\partial \theta}\right|\right|_p = |\frac{\partial H_\lambda(\mathbf{Y}|\mathbf{X})}{\partial \lambda}|$ because we only have one parameter. Therefore, interpolation weight will be chosen by the following criterion

$$\hat{\lambda} = \underset{0 \leq \lambda \leq 1}{\operatorname{argmin}} H_\lambda(\mathbf{Y}|\mathbf{X}) + \gamma |\frac{\partial H_\lambda(\mathbf{Y}|\mathbf{X})}{\partial \lambda}| \tag{8}$$

For the purpose of estimating one parameter λ, we use $\gamma = 1$ in the above equation

7 Experimental Setup

The large vocabulary continuous speech recognition (LVCSR) system used throughout this paper is based on the 2007 IBM Speech transcription system for GALE Distillation Go/No-go Evaluation (Chen et al., 2006). The acoustic models used in this system

are state-of-the-art discriminatively trained models and are the same ones used for all experiments presented in this paper.

For LM adaptation experiments, the out-of-domain LM (p_B, Broadcast News LM) training text consists of 335M words from the following *broadcast news* (BN) data sources (Chen et al., 2006): 1996 CSR Hub4 Language Model data, EARS BN03 closed captions, GALE Phase 2 Distillation GNG Evaluation Supplemental Multilingual data, Hub4 acoustic model training transcripts, TDT4 closed captions, TDT4 newswire, and GALE Broadcast Conversations and GALE Broadcast News. This language model is of order 4-gram with Kneser-Ney smoothing and contains $4.6M$ n-grams based on a lexicon size of $84K$.

The second source of data is the MIT lectures data set (J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, 2007) . This serves as the target domain (in-domain) set for language model adaptation experiments. This set is split into 8 hours for in-domain LM building, another 8 hours served as unlabeled data for interpolation weight estimation using criterion in Eqn. 8 (we refer to this as unsupervised training data) and finally 2.5 hours Dev set for estimating the interpolation weight w.r.t WER (supervised tuning) . The lattice entropy and gradient of entropy w.r.t λ are calculated on the unsupervised training data set. The results are discussed in the next section.

8 Results

In order to optimize the interpolation weight λ based on criterion in Eqn. 8, we devide $[0, 1]$ to 20 different points and evaluate the objective function (Eqn. 8) on those points. For this, we need to calculate entropy and gradient of the entropy on the decoded lattices of the ASR system on 8 hours of MIT lecture set which is used as an unlabeled training data. Fig. 4 shows the value of the objective function against different values of model parameters (interpolation weight λ). As it can be seen from this figure just considering the conditional entropy will result in a non-convex objective function whereas adding the entropy-stability term will make the objective function convex. For the purpose of the evaluation, we show the results for estimating λ directly on the transcription of the 8 hour MIT lecture data and compare it to estimated value using our framework. The results are shown in Fig. 5. Using $\lambda = 0$ and $\lambda = 1$ the WERs are 24.7% and 21.1% respectively. Using the new proposed objective function, the optimal λ is estimated to be 0.6 with WER of 20.1% (Red circle on the figure). Estimating λ w.r.t 8 hour training data transcription (supervised adaptation) will result in $\lambda = 0.7$ (green circle) and WER of 20.0%. Instead $\lambda = 0.8$ will be chosen by tuning the interpolation weight on 2.5 hour Dev set with comparable WER of 20.1%. Also it is clear from the figure that the new objective function can be used to predict the WER trend w.r.t the interpolation weight parameter.

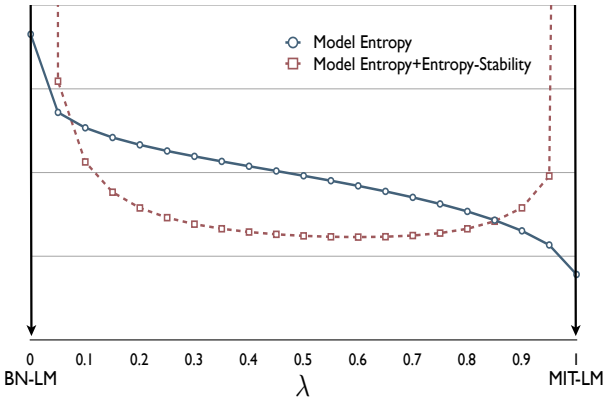

Figure 4: Objective function with and without including Entropy-Stability term vs. interpolation weight λ on 8 hours MIT lecture unlabeled data

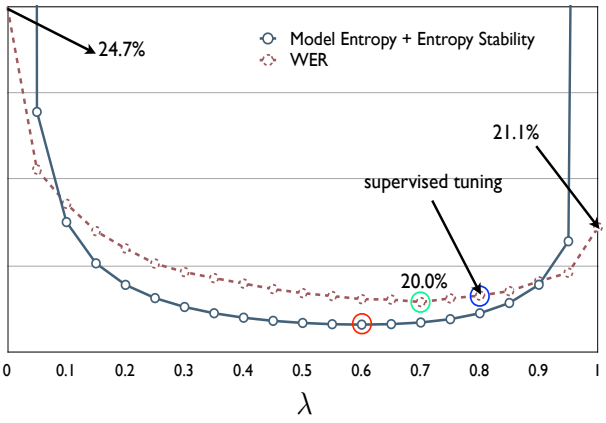

Figure 5: Estimating λ based on WER vs. the information-theoretic criterion

Therefore, it can be seen that the new unsuper-

vised method results in the same performance as supervised adaptation in speech recognition task.

9 Conclusion and Future Work

In this paper we introduced the notion of entropy stability and presented a new criterion for unsupervised adaptation which combines conditional entropy minimization with entropy stability. The entropy stability criterion helps in selecting parameter settings which correspond to stable decision boundaries. Entropy minimization on the other hand tends to push decision boundaries into sparse regions of the input distributions. We show that combining the two criterion helps to improve unsupervised parameter adaptation in real world scenario where class conditional distributions show significant overlap. Although conditional entropy has been previously proposed as a regularizer, to our knowledge, the gradient of entropy (entropy-stability) has not been used previously in the literature. We presented experimental results where the proposed criterion clearly outperforms entropy minimization. For the speech recognition task presented in this paper, the proposed unsupervised scheme results in the same performance as the supervised technique.

As a future work, we plan to use the proposed criterion for adapting log-linear models used in Machine Translation, Conditional Random Fields (CRF) and other applications. We also plan to expand linear interpolation Language Model scheme to include history specific (context dependent) weights.

Acknowledgments

The Authors want to thank Markus Dreyer and Zhifei Li for their insightful discussions and suggestions.

References

M. Bacchiani, B. Roark, and M. Saraclar. 2003. Unsupervised language model adaptation. In *Proc. ICASSP*, pages 224–227.

S. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig. 2006. Advances in speech transcription at IBM under the DARPA EARS program. *IEEE Transactions on Audio, Speech and Language Processing*, pages 1596–1608.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of information theory*. Wiley-Interscience, 3rd edition.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems (NIPS)*, volume 17, pages 529–536.

J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay. 2007. Recent progress in MIT spoken lecture processing project. In *Proc. Interspeech*.

Jing Jiang. 2008. A literature survey on domain adaptation of statistical classifiers, March.

Zhifei Li and Jason Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *EMNLP*.

Haifeng Li, Keshu Zhang, and Tao Jiang. 2004. Minimum entropy clustering and applications to gene expression analysis. In *Proceedings of IEEE Computational Systems Bioinformatics Conference*, pages 142–151.

Lidia Mangu, Eric Brill, and Andreas Stolcke. 1999. Finding consensus among words: Lattice-based word error minimization. In *Sixth European Conference on Speech Communication and Technology*.

M. Szummer and T. Jaakkola. 2003. Information regularization with partially labeled data. In *Advances in Neural Information Processing Systems*, pages 1049–1056.