IMPROVING SEMI-SUPERVISED CLASSIFICATION FOR LOW-RESOURCE SPEECH INTERACTION APPLICATIONS

Manoj Kumar, Pavlos Papadopoulos, Ruchir Travadi, Daniel Bone, Shrikanth Narayanan

Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles, CA

ABSTRACT

We propose a semi-supervised learning method to improve classification performance in scenarios with limited labeled data. We employ adaptation strategies such as entropy-filtering and self-training, and show that our method achieves up to 17.2% relative improvement in UAR for a multi-class problem. We apply our method to two different tasks: speaker clustering for adult-child interactions during autism assessment sessions, and a variation of the language identification task (LID). We show that in both tasks our method improves classification accuracy while using lesser training data than the baseline and demonstrate the robustness of our setup to the degree of adaptation by controlling the threshold on uncertainty of classification.

***Index Terms*—** Semi-supervised learning, i-vectors, speaker clustering, language identification

1. INTRODUCTION AND PREVIOUS WORK

Semi-supervised algorithms often utilize a small amount of labeled data in combination with a (usually) larger set of unlabeled data to improve learning. They are useful in scenarios where obtaining labeled data can be expensive, time-consuming and requires skilled expertise. Various strategies for semi-supervised methods have been developed, including the use of generative models [1, 2], self-training [3] and graph based [4] methods. Each variant of semi-supervised learning places different assumptions on the distribution of unlabeled data.

In this paper, we propose adaptation strategies in a semi-supervised manner to improve classification performance in speech processing scenarios with limited labeled data. We illustrate the importance of adapting the classifier to variabilities localized within a recording session of audio conditions, a commonly encountered scenario in real-life interaction sessions of diagnostic or therapeutic nature. To demonstrate the validity of our method, we choose two different applications: speaker clustering and spoken language identification (LID).

Speaker clustering, which is the process of identifying speaker identities corresponding to a set of speaker homogeneous segments, forms a vital component of many speech processing applications (e.g. speaker diarization). We perform speaker clustering on ADOS (Autism Diagnostic Observation Schedule) sessions [5], which are semi-structured interactions of diagnostic nature between a trained clinician and a child suspected to be on the autism spectrum. Next, we look at a variation of the spoken language identification task [6, 7] on the CALLFRIEND corpus. Our task differs from traditional LID as follows: instead of using separate training and validation corpora which are typically larger than the evaluation data, we label one utterance in each session for training purposes. Both tasks include session-specific variabilities arising from factors such as speaker's gender, age, linguistic capabilities and background noise.

In both tasks, we implement semi-supervised classification as a two-step process. First, a 'global' model is built using per-speaker 'enrollment' (henceforth referred to as 'labeled') data from all available sessions. Next, classification is performed within a session by adapting the global model to the session's data. We employ different adaptation strategies during this step. First we introduce an entropy-based filtering step to identify sessions with 'similar' labeled segments, which are then used to train a session specific classifier. Next, we classify the unlabeled segments using self-training in an iterative manner by classifying those with high confidence scores and retraining our classifier. This method (bootstrapping) where the classifier trains on its own predictions has been used in natural language processing and computer vision [8, 9]. Finally, we use a distance-based assignment for the remaining segments deemed *uncertain* by self-training. This way, we adapt the 'global' classifier to each session's variability during the classification process. We analyze the effect of each of these strategies, both individually and in combination, and propose a system that enhances classification performance by incorporating all of them. Finally, we study the robustness of our framework with respect to the uncertainty threshold hyper-parameter.

The rest of the paper is organized as follows: Section 2 defines the limited labeled data scenario and discusses the components of our method. Section 3 describes the datasets we used, the experimental setup, and presents the results obtained using the proposed adaptation strategies. Conclusions and future work directions are discussed in Section 4.

2. METHODOLOGY

In speech processing applications we often encounter scenar-

Thanks to DARPA, NSF and Simons Foundation for funding.

ios where the amount of labeled data is limited and annotations are expensive. We designed the proposed method to address such cases by carefully selecting "high confidence" examples, and apply self-training. In our method, each recording session is divided into speaker homogeneous segments, and we assume that a single data sample (segment in our case) per class is labeled in every session. We perform classification at the segment level using i-vectors [10] in all our experiments.

General Formulation

Consider N sessions in the corpus with up to K classes; i.e., a session i will contain k_i classes such that $k_i \in [2, K]$. Let $\{x_{ij}\}$ $j \in J_i$ denote the set of all segments within session i, i.e, the cardinality of J_i is the number of segments in session i. Also let $\{x_{ij'}\}$ $j' \in J'_i$ represent the set of labeled segments in session i (one labeled segment for each class). The objective now becomes to classify the unlabeled data $\{x_{ij}\}$ $\forall j \notin J'_i$. In all our experiments, we first build a 'global' supervised classifier (referred to as S_0) using only the labeled data across all sessions, and perform unsupervised adaptation to create another classifier (S_i) specific to session i.

2.1. Entropy-based filtering

The global classifier S_0 is trained using the labeled utterances from all sessions. In order to avoid over-fitting, we do not aim for perfect classification accuracy on the training set. Instead, if the labeled segments of a session are incorrectly classified by S_0, we consider this as an indicator that the classification performance of S_0 on unlabeled segments of that session is also likely to be poor.

In other words, confidence of classification for the labeled example $\{x_i^C\}, C \in [1, K_i]$ from session i can indicate the suitability of S_0 for class C within the i^{th} session. We control the number of labeled examples to retain from each class while classifying within session i. Using the class posteriors for labeled examples obtained using S_0, we define an entropy-inspired score (after Shannon's entropy [11]) for each labeled example as follows:

$$e_i^C = \begin{cases} \displaystyle\sum_{x \in (p_i^C, 1-p_i^C)} x \log(x) & \text{if } p_i^C >= 0.5 \\ 2log(0.5) - \displaystyle\sum_{x \in (p_i^C, 1-p_i^C)} x \log(x) & \text{if } p_i^C < 0.5 \end{cases} \tag{1}$$

where p_i^C is the posterior probability from S_0 for x_i^C. The number of labeled segments belonging to class C from other sessions to retain while classifying session i is:

$$N_i^C = Ne^{e_i^C}$$

The closest N_i^C segments are selected using a distance measure. Note that the entropy score in (1) is asymmetric about p_i^C = 0.5, unlike regular entropy. This ensures that e_i^C is monotonically increasing with p_i^C. This behavior is preferable since a confident and correct classification of x_i^C should favor retaining most of the data used in S_0, and vice-versa. Further, the formulation in (1) ensures that the minimum fraction of labeled data that will be retained is 0.25. This natural regularization guarantees that we do not remove all the data from any class before classification.

2.2. Self-training (Bootstrapping)

Since S_0 may not necessarily capture the session characteristics from all sessions, we classify in an iterative manner while adapting to the session's data. At every iteration, we augment the set of labeled examples by selecting the most confident unlabeled segment (using the posterior probability) along with the label predicted using the current model. The model is then re-trained at the end of each iteration. The algorithm is presented as Algorithm 1.

Algorithm 1 Iterative bootstrapping

```
Input
  • $S_0$: Global classifier
  • $X_{train}$: Labeled features, $Y_{train}$: Labels
  • $\{x_{ij}\}, j \notin J'_i$: Unlabeled segments from session $i$
  • $T$: Uncertainty threshold
while All segments not classified & max($p_j^C$) $> T$ do
    Obtain posterior probabilities $p_j^C$ using $S_0$ for unlabeled data
    $[j'', C''] = \arg\max_{j,C}(p_j^C)$
    $X_{train}.append(x_{ij''})$
    $Y_{train}.append(C'')$
end while
```

2.3. Classification of uncertain segments

Self-training is known to suffer in the later iterations, as confidence of the most certain unlabeled segment keeps decreasing. To avoid this, we use the posterior probabilities from S_i to help identify uncertain segments and fall back to a simpler classification method for such examples. While it is straightforward to use the posterior probability as an uncertainty metric for two classes, we use the entropy of the posterior in the case of multiple classes. Cosine-distance has been used for score computation with i-vectors for many applications [10, 12, 13]. Hence, we classify the uncertain segments by assigning them to the nearest labeled segment based on cosine similarity.

3. EXPERIMENTS AND RESULTS

We applied our method on two different tasks: speaker clustering on the ADOS dataset and LID–like task on CALLFRIEND. In the latter task, we demonstrate the ability of our methods to generalize to multiple classes and replicate our findings using a publicly available corpora [1]. Following, we

[1] https://catalog.ldc.upenn.edu/

describe the datasets and experimental setup.

3.1. Datasets

We use 269 sessions from Module 3 of the ADOS which is a spoken diagnostic interaction between a clinician and a child. The sessions include the *Emotions* subtask, where the child is asked to identify the causes and effects of various emotions in them; and the *Social Difficulties & Annoyance* subtask, where various social problems at home and school are discussed. The sessions cover children from ages 4 to 13 years (Duration: μ = 219s, σ = 89s). We define speaker homogeneous segments within an utterance boundary, which are obtained using the ground truth speech transcripts. For both the child and adult, we choose the longest utterance (Duration: μ = 7.01s, σ = 1.98s) from each session as the labeled data so as to ensure each speaker is represented sufficiently. In total, we have 538 labeled segments and 10,284 unlabeled segments. We use a leave-one-session out cross validation scheme where one session is used for evaluation and the rest for training the GMM-UBM (Gaussian mixture model-universal background model) and i-vector extractor. This is repeated so as to cover every session.

For the next task, we use 13 languages from the CALLFRIEND corpus, which consists of unscripted telephone conversations between native speakers of the particular language. We use Japanese, Korean, Mandarin (Mainland & Taiwan dialects), Spanish (Carribean & Non-Carribean), Tamil and Vietnamese to train the GMM-UBM and i-vector extractor, and Arabic, Farsi, French, German and Hindi for evaluation purposes. For each language, we pool data from the train, dev and eval subsets resulting on an average of 118 speakers per language. We use an energy based voiced activity detector from Kaldi [14] to remove silence regions since the conversations were recorded with low levels of background noise. A segment is defined as a contiguous speech utterance of 2 seconds in duration. We formulate the task similar to spoken language identification by creating synthetic sessions which includes speech segments from speakers of different languages, while ensuring that each language within a session is represented by only one speaker. For example, a session can include segments from speaker 1 of French, speaker 10 of German and speaker 100 of Farsi. The number of languages per session is varied from 2 to 5, and 200 unique sessions are created for each of them. The segment to be labeled from each language is randomly chosen from within that conversation.

3.2. System selection using baseline performance

We define a baseline classifier that does not perform session-level adaptation., we use the global classifier S_0 to classify all the sessions in the corpus. We use support vector machines for both S_0 and session specific classifiers in this work, since they are a popular choice for supervised classification algorithms. We use the baseline performance to optimize the number of Gaussian mixtures while estimating the GMM-UBM, and the i-vector dimension. In the case of LID, we also decide our front-end feature representation between MFCC and SDC (Shifted Delta Cepstra) since the latter has been used recently due to its ability to capture temporal information [15]. While we use the five evaluation languages for parameter optimization in CALLFRIEND, in ADOS we resort to 20-fold cross validation since a leave-one-session out would be computationally expensive. We also experiment with smaller number of GMM mixtures and i-vector dimensions in addition to commonly used values in the case of ADOS considering the size of the corpus.

We use the unweighted average recall (UAR) as our performance metric for each session, which takes into account class imbalances [16]. We report the results as UAR averaged across sessions. The optimal combinations of i-vector dimension and number of UBM mixtures were found to be (400 and 2048) and (20 and 256) for the case of CALLFRIEND and ADOS corpora respectively. These parameter combinations are used in the rest of this work.

3.3. Session-level adaptation strategies

We measure the effect of each of the adaptation strategies on classification performance. We also present all possible combinations of the strategies in Table 1, and study their contributions.

Table 1: Mean UAR for speaker clustering on ADOS and LID on CALLFRIEND. Results are reported separately for each number of languages (2-5) on CALLFRIEND. (*E*: Entropy-based filtering, *B*: Boostrapping, *D*: Cosine-Distance based assignment for uncertain segments)

Method	ADOS	CALLFRIEND			
		2	3	4	5
Baseline	87.62	68.68	55.35	47.07	41.41
E	89.36	73.13	60.58	52.56	47.94
B	88.23	69.83	56.18	47.64	42.38
D	92.25	71.23	55.11	44.99	38.66
E+D	92.90	74.51	**61.68**	**53.09**	**48.62**
E+B	89.87	75.28	60.57	52.30	46.64
B+D	92.25	70.58	56.34	47.51	42.06
E+D+B	**92.91**	**76.81**	61.32	52.69	46.98

We observe a consistent increase in mean UAR across the corpora and across different number of languages in the case of LID. Furthermore, the baseline performance decreases as the number of classes grows.

Classifying uncertain segments using cosine distance (*D*) enhances classification accuracy when the number of classes in small, e.g. ADOS and up to 3 languages in LID, but the performance drops otherwise. This happens since labeled examples from the same class have different representations across different sessions. The entropy-based filtering of labeled data (*E*) gives the largest gains in performance out of all the strategies we employed. In contrast to distance-based classification (*D*), performance gains are proportional to the number of classes., in the 2 language scenario we observe a performance boost of 6.48% while we have a 15.77% increase on the 5

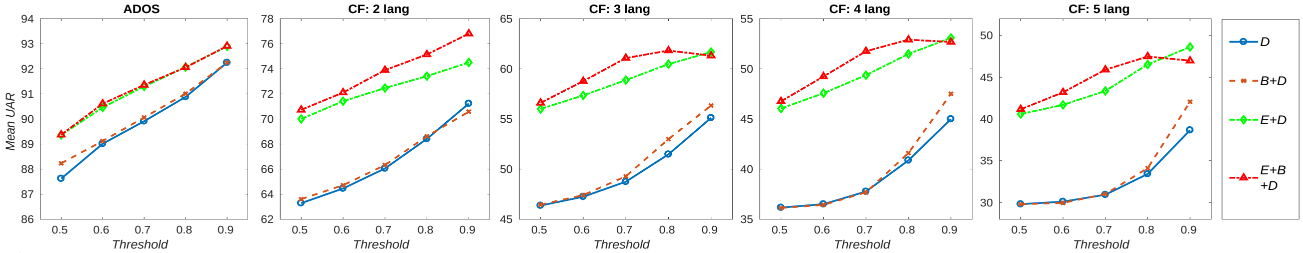

Fig. 1: Effect of threshold for determining uncertain segments on Mean UAR (in %). Presented in combination with entropy-filtering and bootstrapping for each corpora.

language case. Entropy-filtering reduces the amount of labeled segments that are used within a session, which suggests that simply adding more data does not necessarily imply better classifiers. It is more beneficial to retain labeled data from other sessions that are *similar* (in terms of cosine distance) to the labeled data from the current session.

Bootstrapping by itself provides only moderate gains in performance. The performance drops marginally for larger number of classes when bootstrapping is combined with distance-based classification, which could be explained in the same way as distance-based classification (D) alone. However, combining bootstrapping with entropy-filtering decreases performance for larger number of classes when compared to entropy-filtering alone. We expect that the baseline performance (which serves as the initial step) influences the bootstrap strategy, since the latter is of iterative nature. Overall, the best clustering performance is obtained using a combination of all adaptation methods (E+B+D) for lower number of classes, while entropy-filtering followed by distance-based classification (E+D) is the best configuration in the case of large number of classes.

3.4. Dependence on uncertainty threshold

In the last set of our experiments, we look at how the parameter for uncertainty threshold influences the overall performance of our scheme. The threshold determines the amount of data that will be classified using the distance measure, as well as the amount of labeled data used while adapting S_0.

We experimented with values between 0.5 (all examples are deemed certain by the classifier) and 0.9 (most examples are considered uncertain) with a step of 0.1, and present the results in Figure 1. We observe that in most cases the performance increases as the threshold value is increased towards 0.9. This is expected, since we classify only a small, but confident subset of the data with the supervised classifier and hence minimize the errors from uncertain examples. However, the dependence is not uniform across different combinations involving entropy-filtering and bootstrapping. Combining cosine-distance based classification with bootstrapping (B+D) makes the system highly dependent on the threshold, suggesting that bootstrapping accumulates errors at each iteration as we continue classifying segments with lower confidence scores. This effect is somewhat ameliorated with entropy filtering (E+B+D, E+D) especially as the number of languages increases. Since the performance is monotonically increasing with the threshold in most cases, we further fine tuned the threshold between 0.9 and 1.0 (all examples considered uncertain) and present the results for two cases - ADOS and CALLFRIEND with 3 languages. From Figure 2, we observe that there exists a clear optimal threshold for ADOS while the performance saturates for most cases in the case of CALLFRIEND. Hence, while a large threshold favors better performance in general, further inferences might be corpus-specific.

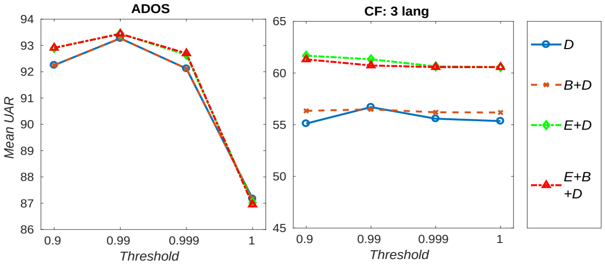

Fig. 2: Classification performance against fine tuned uncertainty threshold for ADOS and CALLFRIEND: 3 languages. The abscissa has been scaled non-linearly for better visualization

4. CONCLUSIONS

In this work, we propose adaptation strategies to improve clustering performance in a semi-supervised manner for speech processing applications. Specifically, we build a global classifier across different recording sessions and adapt it to session-specific variabilities using entropy-filtering and bootstrapping in an iterative manner. We use the reliability of classification to terminate the adaptation and switch to a simple distance-based assignment. We find that entropy-filtering provides the largest gains as a standalone method while a combination of methods provides the best classification performance. Further, selecting a small but confident subset of labeled data using the uncertainty threshold generally favors classification over using a large number of uncertain examples. In the next step, we would like to automatically select labeled segments within an active learning setup, in order to make this methodology fully unsupervised. We would also like to investigate different non-linear functional forms in place of the entropy for selecting the labeled examples in the global model, and analyze the robustness of adaptation strategies, especially entropy-filtering to noise conditions.

5. REFERENCES

[1] X. Zhu, Z. Ghahramani, and J.D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning*, 2003, pp. 912–919.

[2] D.P. Kingma, S. Mohamed, D.J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[3] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, no. 3, pp. 4, 2006.

[4] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005, pp. 57–64.

[5] C. Lord, M. Rutter, P.C. DiLavore, S. Risi, K. Gotham, and S. Bishop, "Autism diagnostic observation schedule: Ados-2," 2012.

[6] Y. K. Muthusamy, E. Barnard, and R. A. Cole, "Reviewing automatic language identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 33–41, Oct 1994.

[7] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: A tutorial," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.

[8] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *Application of Computer Vision. Seventh IEEE Workshops on*, Jan 2005, vol. 1, pp. 29–36.

[9] B. Maeireizo, D. Litman, and R. Hwa, "Co-training for predicting emotions with spoken dialogue data," in *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, 2004.

[10] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. 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.

[11] C. E. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, Jan. 2001.

[12] J. Silovsky and J. Prazak, "Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 4193–4196.

[13] M. Li, C. Lu, A. Wang, and S. Narayanan, "Speaker verification using lasso based sparse total variability supervector with PLDA modeling," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2012, pp. 1–4.

[14] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, and P. Schwarz, "The kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*. dec 2011, IEEE Signal Processing Society.

[15] M. A. Kohler and M. Kennedy, "Language identification using shifted delta cepstra," in *The 45th Midwest Symposium on Circuits and Systems*, Aug 2002, vol. 3, pp. 69–72.

[16] B.W. Schuller, S. Steidl, and A. Batliner, "The INTERSPEECH 2009 emotion challenge," in *10th Annual Conference of the International Speech Communication Association*, Sep 2009, pp. 312–315.