How to Select a Good Training-data Subset for Transcription: Submodular Active Selection for Sequences

Hui Lin, Jeff Bilmes

Department of Electrical Engineering, University of Washington, Seattle, WA 98195
{hlin,bilmes}@ee.washignton.edu

Abstract

Given a large un-transcribed corpus of speech utterances, we address the problem of how to select a good subset for word-level transcription under a given fixed transcription budget. We employ submodular active selection on a Fisher-kernel based graph over un-transcribed utterances. The selection is theoretically guaranteed to be near-optimal. Moreover, our approach is able to bootstrap without requiring *any* initial transcribed data, whereas traditional approaches rely heavily on the quality of an initial model trained on some labeled data. Our experiments on phone recognition show that our approach outperforms both average-case random selection and uncertainty sampling significantly.

Index Terms: Transcription, labeling, submodularity, submodular selection, active learning, sequence labeling, phone recognition, speech recognition

1. Introduction

In automatic speech recognition and many other language applications, unlabeled data are abundant but labels (e.g., transcriptions) are expensive and time-consuming to acquire. For example, large amounts of speech data can easily be obtained via telephone calls, and via modern voice-based applications such as Microsoft's Tellme and Google's voice search. Ideally, it would be possible to label all of this data for use as a training set in a speech recognition system, as aptly conveyed by the well known phrase "there is no data like more data." Unfortunately, this would be impractical given the ever increasing amount of available unlabeled data. Accurate phonetic transcription of speech utterances requires phonetic training and even then it may take a month to annotate 1 hour of speech [1], not to mention the difficulty of transcribing at the articulatory level. Partly due to this, such low-level transcription efforts have been sidelined by the community in favor of word-level transcriptions. But even word level transcriptions are time consuming (about 10 times real time), especially for conversational spontaneous speech. This problem is particularly acute for underrepresented languages or dialects with few speakers, where linguistic experts are even harder to find.

In this paper, we address the following question: given limited resources (time and/or budget), how can we optimally select a training data subset for transcription such that the resulting system has optimal performance. In fact, this is a well-known problem and goes by the name of *batch active learning*, where a subset of data that is most informative and representative of the whole is selected for labeling. Often, examples are queried in a greedy fashion according to an informativeness measure used to evaluate all examples in the pool. Two popular strategies for measuring informativeness include *uncertainty sampling* and the *query-by-committee* approach. Uncertainty sampling [2] is the simplest and most commonly used strategy. In this framework, an initial system is trained typically using a small set of labeled examples. Then, the system examines the rest of the unlabeled examples, and queries examples that it is most uncertain about. The measurement of uncertainty can either be entropy [3, 4, 5] or a confidence score [6, 7, 8, 3]. Query-by-committee [9, 10, 11] also starts with labeled data. A set of distinct models are trained as committee members. Each committee member is then allowed to vote on the labellings of the unlabeled examples. The most informative example is taken as the one the committee most disagrees about.

It has been shown that both uncertainty sampling and query-by-committee may fail when they tend to query outliers, which is the main motivating factor for other strategies like estimated error reduction [12]. The problem is that outliers might have high uncertainty (or a committee might find them controversial) but they are not good surrogates for "typical" samples. Indeed, an ideal selection strategy should choose a subset of samples that, when considered together, constitute in some form a good representation of the entire training data set. Methods such as [13, 14, 15, 3] address this problem, all of which have been shown to be superior to methods that do not consider representativeness measures. Our approach herein also belongs to this category. In particular, we use Fisher kernel (Section 4) to build a graph over the unlabeled sample sequences, and optimize submodular functions (to be defined) over the graph to find the most representative subset. Note that our Fisher kernel is over an unsupervised generative model, which enables us to bootstrap our active learning approach without needing *any* initial labeled data, yet we achieve good performance (see Section 5) perhaps because of the approximate optimality of our submodular procedures. This approach portends well to underrepresented languages for which an initial labeled set might be unavailable.

Despite pre-existing extensive studies of active learning, there is relatively little work on active learning for sequence labeling. Several methods have been proposed, most of which are based either on uncertainty sampling or query-by-committee. In [11, 16, 6], confidence scores from a speech recognizer are used to indicate the informativeness of speech utterances. Active learning methods in [17] select the most uncertain examples based on an EM-style algorithm for learning HMMs from partially labeled data. In [18], several objective functions and algorithms are introduced for active learning in HMMs. Several new query strategies for probabilistic sequence models are introduced in [3] and an empirical analysis is conducted on a variety of benchmark datasets. Our approach can be distinguished from these methods in that we select the most representative

This work was supported by an ONR MURI grant (No. N000140510388).

subset in a *submodular* framework, where submodularity theoretically guarantees that the selection problem can be solved efficiently and near-optimally (see Section 2, Theorem 1 and Theorem 2). Submodularity has already been successfully used in active learning tasks. Robust submodular observation selection is explored in [19]. In [15], the authors relate Fisher information matrices to submodular functions so that the optimization can be done efficiently and effectively. To the best of our knowledge, our approach is the first work that incorporates submodularity for active learning in sequence labeling tasks such as speech recognition.

2. Background

2.1. Submodularity

Consider a set function $z : 2^V \rightarrow \mathbb{R}$, which maps subsets $S \subseteq V$ of a finite set V to real numbers. Intuitively, V is the set of all unlabeled utterances, and the function $z(\cdot)$ scores the quality of any chosen subset. $z(\cdot)$ is called *submodular*[20] if for any $S, T \subseteq V$,

$$z(S \cup T) + z(S \cap T) \leq z(S) + z(T) \tag{1}$$

An equivalent condition for submodularity is the property of diminishing returns. That is for any $R \subseteq S \subseteq V$ and $s \in V$,

$$z(S \cup \{s\}) - z(S) \leq z(R \cup \{s\}) - z(R) \tag{2}$$

Intuitively, this means that adding an element s helps at least as much as if we add it to a smaller set R than if we add it to the superset S. Submodularity is the discrete analog of convexity [20]. As convexity makes continuous functions more amenable to optimization, submodularity plays an essential role in combinatorial optimization. Common submodular functions appear in many important settings including graph-cut [21], set covering [22], and facility location problems [23].

2.2. Submodular Selection

We want to select a good subset S of training data V that maximizes some objective function, such that the size of S is no larger than K (our budget). That is, we wish to compute:

$$\max_{S \subseteq V} \{z(S) : |S| \leq K\} \tag{3}$$

While NP hard, this problem can be approximately solved using a simple greedy forward-selection algorithm. The algorithm starts with $S = \emptyset$, and iteratively adds the element $s \in V \setminus S$ that maximally increases the objective function value, i.e.,

$$s = \text{argmax}_{s \in V \setminus S} \, z(S \cup \{s\}) \tag{4}$$

until $|S| = K$. Actually, when $z(\cdot)$ is a nondecreasing and normalized submodular set function, this simple greedy algorithm performs near-optimally as guaranteed by the following theorems.

Theorem 1. Nemhauser et al. 1978 [24]. *If submodular function $z(\cdot)$ satisfies: i) nondecreasing: for all $S_1 \subseteq S_2 \subseteq V$, $z(S_1) \leq z(S_2)$; ii) normalized: $z(\emptyset) = 0$, then the set S^*_G obtained by the greedy algorithm is no worse than a constant fraction $(1 - 1/e)$ away from the optimal value, i.e.,*

$$z(S^*_G) \geq \left(1 - \frac{1}{e}\right) \max_{S \subseteq V : |S| \leq K} z(S)$$

The greedy algorithm, moreover, is likely to be the best we can do in polynomial time, unless $P = NP$.

Theorem 2. Feige 1998 [22] *Unless P=NP, there is no polynomial-time algorithm that guarantees a solution S^* with*

$$z(S^*) \geq (1 - 1/e + \epsilon) \max_{|S| \leq K} z(S), \epsilon > 0 \tag{5}$$

3. Submodular Selection

Batch active learning problems are often cast as a data subset selection, where the active learner can ask for the labels of the subset of data of size within budget, and that is most likely to yield the most accurate classifier. Problem (3) can also be viewed as a data selection problem. Suppose we have a set of unlabeled training examples $V = \{1, 2, ..., N\}$, where certain pairs (i, j) are similar and the similarity of i and j is measured by a nonnegative value $w_{i,j}$. We can represent the unlabeled data using a graph $G = (V, E)$, with nonnegative weights $w_{i,j}$ associated with each edge (i, j). The data selection problem is to find a subset S that is most representative of the whole set V, given the constraint $|S| \leq K$. To measure how "representative" S is of the whole set V, we introduce several submodular set functions.

3.1. Submodular Set Functions

Our first objective is the uncapacitated facility location function [23]:

$$\text{Facility location: } z_1(S) = \sum_{i \in V} \max_{j \in S} w_{i,j} \tag{6}$$

It measures the similarity of S to the whole set V. We can also measure the similarity of S to the remainder, i.e., the graph cut function:

$$\text{Graph cut: } z_2(S) = \sum_{i \in V \setminus S} \sum_{j \in S} w_{i,j} \tag{7}$$

Both of these functions are submodular as seen by verifying inequality 2 (proof omitted due to space limitations).

In order to apply Theorem 1, the objective function should also satisfy the nondecreasing property. Obviously, the facility location objective function is nondecreasing. For the graph cut objective, the increment of adding k into S is

$$z_2(S \cup \{k\}) - z_2(S) = \sum_{i \in V \setminus S} w_{i,k} - \sum_{j \in S \cup \{k\}} w_{k,j}$$

which is not always nonnegative. Fortunately, the proof of Theorem 1 does not use the monotone property for all possible sets [24][19, page 58]. The graph cut can also meet the conditions for Theorem 1 if $|S| \ll |V|$, which is usually the case in applications where we have a large amount of data but only limited resources for labeling.

With the above objectives, we can use the greedy algorithm to solve the data selection problem efficiently and near-optimally. The greedy algorithm for submodular data selection with the facility location objective is described in Algorithm 1, where $\rho_i = \max_{j \in S} w_{i,j}$ is updated to optimize the running of the algorithm. The graph-cut objective algorithm is similar and is omitted to conserve space.

Algorithm 1 Greedy algorithm for facility location objective

Input: $G = (V, E)$ with weights $w_{i,j}$ on edge (i, j); K: the number of examples to be selected
Initialization: $S = \emptyset$, $\rho_i = 0, i = 1, ..., N$ where $N = |V|$
while $|S| \leq K$ **do**
  $k^* = \arg\max_{k \in V \setminus S} \sum_{i \in V, (i,k) \in E} (\max\{\rho_i, w_{i,k}\} - \rho_i)$
  $S = S \cup \{k^*\}$
  for all $i \in V$ **do**
    $\rho_i = \max\{\rho_i, w_{i,k^*}\}$
  end for
end while

4. Fisher Kernel

We express the pairwise "similarity" between the utterances i and j in terms of kernel function $\kappa(i, j)$ so that $w_{i,j} = \kappa(i, j)$. Since the examples are sequences with possibly different lengths, we use the Fisher kernel [25], which is applicable to variable length sequences. Consider a generative model (e.g., a hidden Markov models, or more generally, a dynamic Bayesian network (DBN)) with parameters θ that models the generation process of the sequence. Denote $X_i = (x_{i,1}, \ldots, x_{i,T_i})$ as the i^{th} feature sequence with length T_i. Then a fixed length vector, known as the Fisher score, can be extracted as:

$$U_i = \frac{\partial}{\partial \theta} \log p(X_i|\theta) \tag{8}$$

Each component of U_i is a derivative of the log-likelihood score for the sequence X_i with respect to a particular parameter — the Fisher score is thus a vector having the same length as the number of parameters θ. The computation of gradients in Eq. 8 in the context of DBNs is described in detail in [26].

Given Fisher scores, different sequences with different lengths may be represented by fixed-length vectors, so we can easily define several Fisher kernel functions to measure pairwise similarity, e.g., cosine similarity, radial-basis function (RBF) kernel similarity, or as shown below, the negative ℓ_1 similarity:

$$\text{Negative } \ell_1 \text{ norm: } \kappa(i, j) = -||U_i - U_j||_1 \tag{9}$$

The generative model that is used to generate the Fisher score may contain several types of parameters (i.e., discrete conditional probability tables and continuous Gaussian parameters), and the values associated with different types of parameters may have quite different numeric dynamic ranges. In order to reduce the heterogeneity within the Fisher score vector, all our experiments apply the following global variance normalization to produce the final Fisher score vectors U_i':

$$U_i' = (\mathbf{diag}(\Sigma))^{-\frac{1}{2}} \cdot (U_i - \bar{U}) \tag{10}$$

where $\bar{U} = \frac{1}{N} \sum_{i=1}^{N} U_i$ and $\Sigma = \frac{1}{N} \sum_{i=1}^{N} (U_i - \bar{U})^T (U_i - \bar{U})$

5. Experiments

We evaluated our methods on a phone recognition task using the TIMIT corpus. Random selection was used as a baseline. Specifically, we randomly take $p\%$ of the TIMIT training set, where $p = 2.5, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90$. For each subset, a 3-state context-independent (CI) hidden Markov model (HMM) (implemented as a DBN) was trained for each of the 48 phones. The number of Gaussian components in the

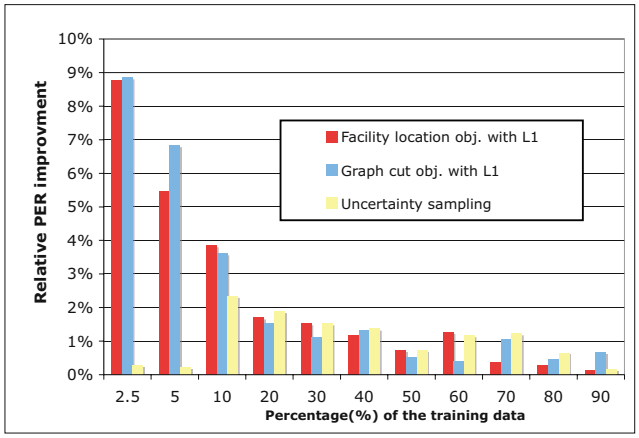

Figure 1: Relative improvements over the average phone error rate of random selection. No initial model scenario.

Gaussian mixture model (GMM) was optimized according to the amount of training data available. The 48 phones were then mapped down to 39 phones for scoring purposes following standard practice [27]. Recognition was performed using standard Viterbi search without a phonetic language model (a language model was not used here to emphasize the acoustic modeling performance, and since this speeds up experimental turnaround time by avoiding tedious language model scaling and penalty parameter tuning when large random selection experiments are performed). 100 trials of random selection experiments were performed for each of the percentage numbers above. The average phone error rates (PER) were calculated and used as baseline. The standard deviation was around 0.01 for small p and about 0.005 for larger p. Apart from the data selection strategy, experiments on uncertainty sampling and submodular selection followed exactly the same setups as random selection.

Uncertainty sampling and submodular selection were evaluated under two scenarios. The first scenario we considered is when there is no initial model available. In this scenario, uncertainty sampling would typically randomly select a small portion of the unlabeled data to label, and then train an initial model using these randomly selected data. We did the following: a) randomly select $\alpha\%$ of the training data, acquire the labels and train an initial model; b) use the learned model to predict the unlabeled data, select the M most uncertain samples for labelling; c) retrain the model using all labeled data. If the number of labeled samples reaches the target amount, stop, else go to step b). We used $\alpha = 1$ and $M = 100$ in the experiments, and the average per-frame log-likelihood was used as the uncertainty measurement.

For our submodular selection method, HMMs with 16-component GMMs were obtained by unsupervised training using all the unlabeled data. This model was used as the generative model for the Fisher score using `gmtkKernel`, a GMTK [28] DBN implementation of Fisher kernels. The negative ℓ_1 norm was used to construct the graph (we also tested other measures which had similar results). The relative PER improvements over the average of the 100 random experiments are shown in Figure 1. As we can see, uncertainty sampling achieves improvements over random sampling in general, but when the target percentage number is small (i.e., 2.5% and 5%), which is usually the case in real-world applications, it performs similarly to random selection since the model used for the uncertainty measurement is of low quality. On the other hand, submodular data selection outperforms both random selection and

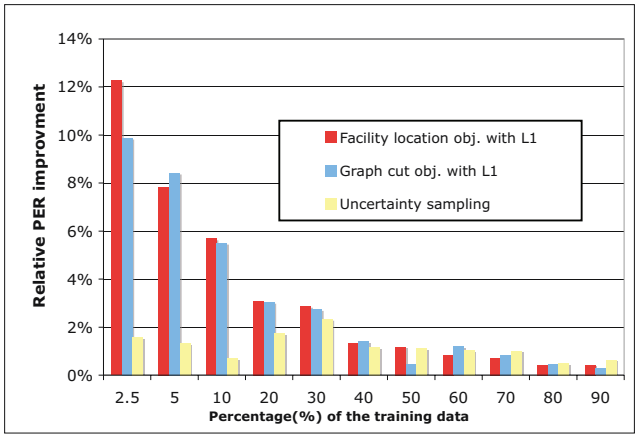

Figure 2: Relative improvements over the average phone error rate of random selection. With initial model scenario.

uncertainty sampling, especially when the percentage is small. This implies that even a model trained without any labeling information works quite well for our approach. In other words, the submodular data selection approach proposed here is quite robust to the scenario where no initial "boot" model is available.

Our second scenario is when an initial model is available to help the data selection. Such a model should have reasonable quality. In our experiments, we assume a very high quality initial model to strongly contrast with our first scenario – an initial model with 16-component GMM-HMMs was trained on *all* the labeled TIMIT data, which was then used in the uncertainty sampling approach, and also in the submodular selection method as the generative model. The results are shown in Figure 2 — with a better quality initial model, uncertainty sampling performs better when selecting small percentages of the data but not necessarily with more data (presumably due to its selection of unrepresentative outliers). Submodular data selection also performs better in general with a better quality initial model. In particular, more than 12% relative improvement over random selection is achieved when selecting 2.5% of the data. And again, submodular selection outperforms both random sampling and uncertainty sampling. Also, notice that there are only relatively minor performance drops in our approach when shifting from a supervised trained initial model to an unsupervised trained initial model, illustrating yet again that submodular selection seems robust to the quality of the initial model.

6. References

[1] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Speech Input/Output Assessment and Speech Databases*. ISCA, 1989.

[2] D. Lewis and W. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc. New York, NY, USA, 1994, pp. 3–12.

[3] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *EMNLP*, 2008. [Online]. Available: http://pages.cs.wisc.edu/~bsettles/pub/settles.emnlp08.pdf

[4] B. Varadarajan, D. Yu, L. Deng, and A. Acero, "Maximizing global entropy reduction for active learning in speech recognition," in *ICASSP*, 2009.

[5] Y. Wu, R. Zhang, and A. Rudnicky, "Data selection for speech recognition," in *ASRU*, Dec. 2007, pp. 562–565.

[6] D. Hakkani-tür and A. Gorin, "Active learning for automatic speech recognition," in *in Proceedings of the ICASSP*, 2002, pp. 3904–3907.

[7] A. Culotta and A. McCallum, "Reducing labeling effort for structured prediction tasks," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2005.

[8] D. Hakkani-Tür, G. Riccardi, and G. Tur, "An active approach to spoken language processing," *ACM Trans. Speech Lang. Process.*, vol. 3, no. 3, pp. 1–31, 2006.

[9] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.

[10] I. Dagan and S. Engelson, "Committee-based sampling for training probabilistic classifiers," in *ICML*. Morgan Kaufmann, 1995, pp. 150–157.

[11] G. Tur, R. Schapire, and D. Hakkani-Tur, "Active learning for spoken language understanding," in *ICASSP*, vol. 1, 2003.

[12] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *ICML*, 2001, pp. 441–448.

[13] A. Fujii, T. Tokunaga, K. Inui, and H. Tanaka, "Selective sampling for example-based word sense disambiguation," *Computational Linguistics*, vol. 24, no. 4, pp. 573–597, 1998.

[14] H. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *ICML*. ACM New York, NY, USA, 2004.

[15] S. Hoi, R. Jin, J. Zhu, and M. Lyu, "Batch mode active learning and its application to medical image classification," in *ICML*. ACM New York, NY, USA, 2006, pp. 417–424.

[16] G. Tur, D. Hakkani-Tür, and R. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Communication*, vol. 45, no. 2, pp. 171–186, 2005.

[17] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*. Springer-Verlag London, UK, 2001, pp. 309–318.

[18] B. Anderson and A. Moore, "Active learning for hidden markov models: Objective functions and algorithms," in *Machine Learning-International workshop*, vol. 22, 2005, p. 9.

[19] A. R. Krause, "Optimizing sensing: Theory and applications," Ph.D. dissertation, Carnegie Mellon University, 2008.

[20] L. Lovasz, "Submodular functions and convexity," *Mathematical programming-The state of the art,(eds. A. Bachem, M. Grotschel and B. Korte) Springer*, pp. 235–257, 1983.

[21] M. Goemans and D. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.

[22] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

[23] G. Cornuejols, M. FISHER, and G. Nemhauser, "On the uncapacitated location problem," in *Studies in Integer Programming: Proceedings of the Institute of Operations Research Workshop, Sponsored by IBM, University of Bonn, Germany, Sept. 8-12, 1975*, vol. 1. North Holland, 1977, pp. 163–177.

[24] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[25] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *Advances in neural information processing systems*, pp. 487–493, 1999.

[26] J. Bilmes, "Fisher kernels for DBNs," University of Washington, Tech. Rep., 2008.

[27] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[28] J. Bilmes and C. Bartels, "Graphical model architectures for speech recognition," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 89–100, September 2005.