A Pylonic Decision-Tree Language Model with Optimal Question Selection

Adrian Corduneanu
University of Toronto
73 Saint George St #299
Toronto, Ontario, M5S 2E5, Canada
g7adrian@cdf.toronto.edu

Abstract

This paper discusses a decision-tree approach to the problem of assigning probabilities to words following a given text. In contrast with previous decision-tree language model attempts, an algorithm for selecting nearly optimal questions is considered. The model is to be tested on a standard task, *The Wall Street Journal*, allowing a fair comparison with the well-known trigram model.

1 Introduction

In many applications such as automatic speech recognition, machine translation, spelling correction, *etc.*, a statistical language model (LM) is needed to assign probabilities to sentences. This probability assignment may be used, *e.g.*, to choose one of many transcriptions hypothesized by the recognizer or to make decisions about capitalization. Without any loss of generality, we consider models that operate left-to-right on the sentences, assigning a probability to the next word given its word history. Specifically, we consider statistical LM's which compute probabilities of the type $\mathrm{P}\{w_n \,|\, w_1, w_2, \ldots, w_{n-1}\}$, where w_i denotes the i-th word in the text.

Even for a small vocabulary, the space of word histories is so large that any attempt to estimate the conditional probabilities for each distinct history from raw frequencies is infeasible. To make the problem manageable, one partitions the word histories into some classes $C(w_1, w_2, \ldots, w_{n-1})$, and identifies the word probabilities with $\mathrm{P}\{w_n \,|\, C(w_1, w_2, \ldots, w_{n-1})\}$. Such probabilities are easier to estimate as each class gets significantly more counts from a training corpus. With this setup, building a language model becomes a classification problem: group the word histories into a small number of classes while preserving their predictive power.

Currently, popular N-gram models classify the word histories by their last $N-1$ words. N varies from 2 to 4 and the trigram model $\mathrm{P}\{w_n \,|\, w_{n-2}, w_{n-1}\}$ is commonly used. Although these simple models perform surprisingly well, there is much room for improvement. The approach used in this paper is to classify the histories by means of a decision tree: to cluster word histories $w_1, w_2, \ldots, w_{n-1}$ for which the distributions of the following word w_n in a training corpus are similar. The decision tree is pylonic in the sense that histories at different nodes in the tree may be recombined in a new node to increase the complexity of questions and avoid data fragmentation.

The method has been tried before (Bahl et al., 1989) and had promising results. In the work presented here we made two major changes to the previous attempts: we have used an optimal tree growing algorithm (Chou, 1991) not known at the time of publication of (Bahl et al., 1989), and we have replaced the *ad-hoc* clustering of vocabulary items used by Bahl with a data-driven clustering scheme proposed in (Lucassen and Mercer, 1984).

2 Description of the Model

2.1 The Decision-Tree Classifier

The purpose of the decision-tree classifier is to cluster the word history $w_1, w_2, \ldots, w_{n-1}$ into a manageable number of classes C_i, and to estimate for each class the next word conditional distribution $\mathrm{P}\{w_n \,|\, C_i\}$. The classifier, together with the collection of conditional probabilities, is the resultant LM.

The general methodology of decision tree construction is well known (*e.g.*, see (Jelinek, 1998)). The following issues need to be addressed for our specific application.

- A tree growing criterion, often called the measure of purity;
- A set of permitted questions (partitions) to be considered at each node;
- A stopping rule, which decides the number of distinct classes.

These are discussed below. Once the tree has been grown, we address one other issue: the estimation of the language model at each leaf of the resulting tree classifier.

2.1.1 The Tree Growing Criterion

We view the training corpus as a set of ordered pairs of the following word w_n and its word history $\langle w_1, w_2, \ldots, w_{n-1}\rangle$. We seek a classification of the space of all histories (not just those seen in the corpus) such that a good conditional probability $\mathrm{P}\{w_n \,|\, C(w_1, w_2, \ldots, w_{n-1})\}$ can be estimated for each class of histories. Since several vocabulary items may potentially follow any history, perfect "classification" or prediction of the word that follows a history is out of the question, and the classifier must partition the space of all word histories maximizing the probability $\mathrm{P}\{w_n \,|\, C(w_1, w_2, \ldots, w_{n-1})\}$ assigned to the pairs in the corpus.

We seek a history classification such that $C(w_1, w_2, \ldots, w_{n-1})$ is as informative as possible about the distribution of the next word. Thus, from an information theoretical point of view, a natural cost function for choosing questions is the empirical conditional entropy of the training data with respect to the tree:

$$H = -\sum_w \sum_i f(C_i) f(w \,|\, C_i) \log f(w \,|\, C_i).$$

Each question in the tree is chosen so as to minimize the conditional entropy, or, equivalently, to maximize the mutual information between the class of a history and the predicted word.

2.1.2 The Set of Questions and Decision Pylons

Although a tree with general questions can represent any classification of the histories, some restrictions must be made in order to make the selection of an optimal question computationally feasible. We consider elementary questions of the type $w_{-k} \in S$, where w_{-k} refers to the k-th position before the word to be predicted, and S is a subset of the vocabulary. However, this kind of elementary question is rather simplistic, as one node in the tree cannot refer to two different history positions. A conjunction of elementary questions can still be implemented over a few nodes, but similar histories become unnecessarily fragmented. Therefore a node in the tree is not implemented as a single elementary question, but as a modified decision tree in itself, called a *pylon* (Bahl et al., 1989). The topology of the pylon as in Figure 1 allows us to combine answers from elementary questions without increasing the number of classes. A pylon may be of any size, and it is grown as a standard decision tree.

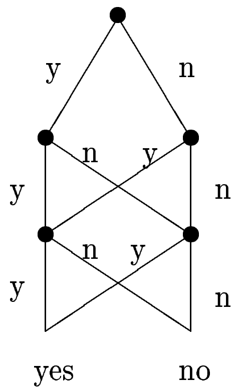

Figure 1: The structure of a pylon

2.1.3 Question Selection Within the Pylon

For each leaf node and position k the problem is to find the subset S of the vocabulary that minimizes the entropy of the split $w_{-k} \in S$. The best question over all k's will eventually be selected. We will use a greedy optimization algorithm developed by Chou (1991). Given a partition $P = \{\beta_1, \beta_2, \ldots, \beta_k\}$ of the vocabulary, the method finds a subset S of P for which the reduction of entropy after the split is nearly optimal.

The algorithm is initialized with a random partition $S \cup \bar{S}$ of P. At each iteration every atom β is examined and redistributed into a new partition $S' \cup \bar{S}'$, according to the following rule: place β into S' when

$$\sum_w f(w|w_{-k} \in \beta) \log \frac{f(w|w_{-k} \in \beta)}{f(w|w_{-k} \in S)} \leq \sum_w f(w|w_{-k} \in \beta) \log \frac{f(w|w_{-k} \in \beta)}{f(w|w_{-k} \in \bar{S})}$$

where the f's are word frequencies computed relative to the given leaf. This selection criterion ensures a decreasing empirical entropy of the tree. The iteration stops when $S = S'$ and $\bar{S} = \bar{S}'$.

If questions on the same level in the pylon are constructed independently with the Chou algoritm, the overall entropy may increase. That is why nodes whose children are merged must be jointly optimized. In order to reduce complexity, questions on the same level in the pylon are asked with respect to the same position in the history.

The Chou algorithm is not accurate when the training data is sparse. For instance, when no history at the leaf has $w_{-k} \in \beta$, the atom is invariantly placed in S'. Because such a choice of a question is not based on evidence, it is not expected to generalize to unseen data. As the tree is growing, data is fragmented among the leaves, and this issue becomes unavoidable. To deal with this problem, we choose the atomic partition P so that each atom gets a history count above a threshold.

The choice of such an atomic partition is a complex problem, as words composing an atom must have similar predictive power. Our approach is to consider a hierarchical classification of the words, and prune it to a level at which each atom gets sufficient history counts. The word hierarchy is generated from training data with an information theoretical algorithm (Lucassen and Mercer, 1984) detailed in section 2.2.

2.1.4 The Stopping Rule

A common problem of all decision trees is the lack of a clear rule for when to stop growing new nodes. The split of a node always brings a reduction in the estimated entropy, but that might not hold for the true entropy. We use a simplified version of *cross-validation* (Breiman et al., 1984), to test for the significance of the reduction in entropy. If the entropy on a held out data set is not reduced, or the reduction on the held out text is less than 10% of the entropy reduction on the training text, the leaf is not split, because the reduction in entropy has failed to generalize to the unseen data.

2.1.5 Estimating the Language Model at Each Leaf

Once an equivalence classification of all histories is constructed, additional training data is used to estimate the conditional probabilities required for each node, as described in (Bahl et al., 1989). Smoothing as well as interpolation with a standard trigram model eliminates the zero probabilities.

2.2 The Hierarchical Classification of Words

The goal is to build a binary tree with the words of the vocabulary as leaves, such that similar words correspond to closely related leaves. A partition of the vocabulary can be derived from such a hierarchy by taking a cut through the tree to obtain a set of subtrees. The reason for keeping a hierarchy instead of a fixed partition of the vocabulary is to be able to dynamically adjust the partition to accommodate for training data fragmentation.

The hierarchical classification of words was built with an entirely data-driven method. The motivation is that even though an expert could exhibit some strong classes by looking at parts of speech and synonyms, it is hard to produce a full hierarchy of a large vocabulary. Perhaps a combination of the expert and data-driven approaches would give the best result. Nevertheless, the algorithm that has been used in deriving the hierarchy can be initialized with classes based on parts of speech or meaning, thus taking account of prior expert information.

The approach is to construct the tree backwards. Starting with single-word classes, each iteration consists of merging the two classes most similar in predicting the word that follows them. The process continues until the entire vocabulary is in one class. The binary tree is then obtained from the sequence of merge operations.

To quantify the predictive power of a partition $P = \{\beta_1, \beta_2, \ldots, \beta_k\}$ of the vocabulary we look at the conditional entropy of the vocabulary with respect to class of the previous word:

$$H(w \mid P) = \textstyle\sum_{\beta \in P} p(\beta) H(w \mid w_{-1} \in \beta) \quad = \\ - \textstyle\sum_{\beta \in P} p(\beta) \sum_{w \in V} p(w \mid \beta) \log p(w \mid \beta)$$

At each iteration we merge the two classes that minimize $H(w \mid P') - H(w \mid P)$, where P' is the partition after the merge. In information-theoretical terms we seek the merge that brings the least reduction in the information provided by P about the distribution of the current word.

IRAN'S
UNION'S
IRAQ'S
INVESTORS'
BANKS'
PEOPLE'S

FARMER
TEACHER
WORKER
DRIVER
WRITER
SPECIALIST
EXPERT
TRADER

PLUMMETED
PLUNGED
SOARED
TUMBLED
SURGED
RALLIED
FALLING
FALLS
RISEN
FALLEN

MYSELF
HIMSELF
OURSELVES
THEMSELVES

CONSIDERABLY
SIGNIFICANTLY
SUBSTANTIALLY
SOMEWHAT
SLIGHTLY

Figure 2: Sample classes from a 1000-element partition of a 5000-word vocabulary (each column is a different class)

The algorithm produced satisfactory results on a 5000-word vocabulary. One can see from the sample classes that the automatic building of the hierarchy accounts both for similarity in meaning and of parts of speech.

3 Evaluation of the Model

The decision tree is being trained and tested on the *Wall Street Journal* corpus from 1987 to 1989 containing 45 million words. The data is divided into 15 million words for growing the nodes, 15 million for cross-validation, 10 million for estimating probabilities, and 5 million for testing. To compare the results with other similar attempts (Bahl et al., 1989), the vocabulary consists of only the 5000 most frequent words and a special "unknown" word that replaces all the others. The model tries to predict the word following a 20-word history.

At the time this paper was written, the implementation of the presented algorithms was nearly complete and preliminary results on the performance of the decision tree were expected soon. The evaluation criterion to be used is the perplexity of the test data with respect to the tree. A comparison with the perplexity of a standard back-off trigram model will indicate which model performs better. Although decision-tree letter language models are inferior to their N-gram counterparts (Potamianos and Jelinek, 1998), the situation should be reversed for word language models. In the case of words the vocabulary is significantly larger, making impossible the estimation of N-gram models for $N > 3$. However, we expect that due to the good smoothing of the trigram probabilities a combination of the decision-tree and N-gram models will give the best results.

4 Summary

In this paper we have developed a decision-tree method for building a language model that predicts words given their previous history. We have described a powerful question search algorithm, that guarantees the local optimality of the selection, and which has not been applied before to word language models. We expect that the model will perform significantly better than the standard N-gram approach.

5 Acknowledgments

I would like to thank Prof.Frederick Jelinek and Sanjeev Khudampur from Center for Language and Speech Processing, Johns Hopkins University, for their help related to this work and for providing the computer resources. I also wish to thank Prof.Graeme Hirst from University of Toronto for his useful advice in all the stages of this project.

References

L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer. 1989. A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:1001–1008.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. 1984. *Classification and regression trees.* Wadsworth and Brooks, Pacific Grove.

P. A. Chou. 1991. Optimal partitioning for classification and regression trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:340–354.

F. Jelinek. 1998. *Statistical methods for speech recognition.* The MIT Press, Cambridge.

J. M. Lucassen and R. L. Mercer. 1984. An information theoretic approach to the automatic determination of phonemic baseforms. In *Proceedings of the 1984 International Conference on Acoustics, Speech, and Signal Processing*, volume III, pages 42.5.1–42.5.4.

G. Potamianos and F. Jelinek. 1998. A study of n-gram and decision tree letter language modeling methods. *Speech Communication*, 24:171–192.