14.1 Pronunciation Modeling

The problem with using dictionaries for pronunciation modeling is that they tend to give base form pronunciations of words. PRONLEX gives a number of different pronunciations for a variety of word forms.

In speech synthesis, it is usually desirable to contain all possible pronunciations of words. This is not the case with speech recognition. If it were the case, words such as "banana" might be easily confused with a word such as "window". The hidden Markov chain model for "window" could score the word "banana" higher than the model for "banana". A certain level of discrimination between the words "window" and "banana" would reduce the number of times one word is confused with the other.

This poses a problem for recognition, since some words have multiple common pronunciations.

One method of increasing discriminativeness in the case of a "too general" model is to monitor how a specific person pronounces a word. By obtaining information such as rate of speech, mood, etc., the set of things to select from can be narrowed down based on how that person would pronounce it.

Decision trees are often times used for pronunciation modeling. Other kinds of trees are used for language models. Simpler models can usually work acceptably well with less training data. Aligning the base and surface form of a word can be done with Dynamic Time Warping, a topic covered earlier in the course.

A good model would be able to come up with pronunciations of words dynamically. The goal is to build a probabilistic mapping from base forms to surface forms[1]. This model would be able to accept a word which has never been seen before except in base form, and output the probabilistic translation of the word.

Typical problems encountered during construction of these models are computation complexity and discrimination. Table sizes reach $200 \times 50^{2r+1}$, where $r$ is the context size of the language. Use of a decision tree results in a significant reduction in size requirements, which consequentially reduces time needed to search the data.

A set of phonetic questions and rules determine the depth and breadth of the decision trees. Deciding which questions should come first, and which should come next, etc., is an empirical area of speech recognition in which work is still
being done. Computing the **entropy**, and eventually the **perplexity**, of the language can be used to help determine a solution.

**Entropy** \( H(p) = - \sum_i p_i \log p_i \)

**Perplexity** \( H_q(p) = - \sum_i p_i \log q_i \geq H(p) \)

The entropy is basically the amount of uncertainty. It is at a maximum when there is a maximal probability spread; i.e. uniform distribution. It is minimal when there is only one possible value. When designing a decision tree, the goal is to make decisions which minimize entropy, meaning that there is as little uncertainty about the present phone as possible. Before any decisions are made, all of the entropy is present. The first question to be asked should reduce the entropy by as much as possible. This entropy reduction is exactly equal to the mutual information. If the question reduces the answer to only one possible phone, this is the least amount of entropy and recognition has succeeded.

Perplexity is a measure of how complicated that a language is. Minimizing entropy also minimizes perplexity. Reduction of a vocabulary set by context decreases perplexity, for example, if the context of the conversation is known, it is easier to identify words. Some of these topic-limited perplexity values are given below[9].

- Domain: Radiology has a perplexity of about 20
- Domain: Emergency Medicine has a perplexity of about 60
- Domain: Journalism has a perplexity of about 105
- Domain: General English has a perplexity of about 247

### 14.2 Language Modeling

Most of the **language modeling** work which has been done has come out of speech recognition. When \( P(w) \) is a sentence instead of a word, maximum likelihood estimation using the counting method will not work, because many strings will result in zero probability. Here is an exercise that you can do to see how easy is is for a zero probability string to occur:

**Exercise**: The Google search engine can be used to search for an exact string of words. Find the shortest string of words for which Google does not find a match. The goal of this exercise is to demonstrate that when there is a large vocabulary, there may be strings of words which do not occur, but these strings should still be recognizable by a language model or the speech recognizer would output something other than the string which was said. **Solution**: It turns out that there are a large number of 2-word strings which have no exact match. "Avocado Transistor" is one such solution.

Perplexity is a measure of how "difficult" a language model things a language is. Sentences of the language are scored using the language model. If the language model is accurate and scores everything highly, it is a simple language model. If it scores everything lowly, then it is an inherently complex language model. The language model should score high on the training data.

**Geometric mean** is another method of measuring perplexity. This method treats the probabilities as lengths, finds their averages, and takes the inverse to get the perplexity.

**The cross entropy** is always greater than or equal to the entropy.

English written text has a perplexity of about 68. This means that it is just as complicated as a language with only 68 words if there were no dependencies between words. In the sentence, "Yesterday I went to the store and bought a ??????", there is a limited but fairly large number of words which can sensibly fill in the question marks. It is substantially easier to guess the missing word in the sentence, "Today I cashed a ??????". These examples help to
demonstrate the concept of perplexity, and how it considers the prior context of the word. Most modern languages have around the same perplexity (50-100). If the plural and singular forms of a word are considered entirely separate, this adds a lot more to the perplexity than considering the singular word with a possibility of an added “’s”.

Infinitely complex language models are undesirable. Although languages are complex, they are not infinitely complex. The language model has to work together with the HMM of the acoustic model. Increasing the model complexity does not necessarily lead to a decrease in perplexity. This occurs when the data reduction involved with making a decision does not reduce the entropy by a sufficient amount. In short, there is less data available to make the following decision, and the following decision is just about as hard as the previous decision.

The maximum likelihood training method has zeroes in the language model. This is a problem. Another problem is having too big of a table; current memory technologies are insufficient to this taks. Some alternative language models have been developed. One such model is adaptive language modeling, where a cache of recent words is used. Words which have occurred recently and related words can have their probabilities boosted with respect to words which have not been used recently. Another method is backoff language modeling. When counts fall below a threshold, the probability is formed by a lower order model. A “’discount coefficient’” is used to determine how much probability weight to take from higher order maximum likelihood solutions and apply towards lower order solutions. This is the current most successful language model in use.

In-Class Questions and Answers

**Question:** What is a decision tree?
**Answer:** It’s a tree that expands as you go downwards; basically represents a set of questions. Some data item goes into the top of the tree. The first node answers some question about the data item, and passes the data item to the next node in the tree. Once a leaf is reached, you’ve gotten an answer pretty quickly. Like a binary search. Finally, when we get to the leaf node, we have a probability distribution on Y given that path to get to the leaf; a quick way of getting to the leaf.

**Question:** Isn’t the path from the root to a leaf unique? Not necessarily, but assuming it is: If it’s unique and the path taken is conditional; if there’s only one way to get there it seems like the conditioning is unnecessary.
**Answer:** It’s not the P(conditioning) that you’re looking for. Let’s say x is a set of integers; and we have a table XxY in size; P(Y—X). What a decision tree might do is if x¡110 then use P1(y), else if x is between 110 and 210 use P2(y), etc. It clusters the x values into regions. Most of the table entries are 0’s and we don’t want 0’s, so instead of P(Y—X) we say that it’s P(Y—f(x)). For many values of x we’re considering them equivocal to small values of x, i.e. breaking x up into regions with no distinction between elements of a region.

**Question:** I’m thinking about algorithms that will dynamically balance a tree as you build it. There’s all sorts of algorithms that go with trees that make them not skewed.
**Answer:** There’s notions in data structures that choose useful trees for different data sets. In some sense here, as you see when we actually look at this entropy minimizing criterium, we typically have things which tend to balance the trees, but there’s no guarantee and you’re not always trying to balance the trees. You could add some of those constraints on to the tree building procedure, but we’re not trying to optimize that. What are we trying to optimize? We’re trying to optimize the entropy.

**Question:** It’s going to be a Huffman tree, right?
**Answer:** You’re coding for length in a Huffman tree; what you end up getting in something where the expected length is the entropy. But you get a code where the expected length is exactly equal to the entropy. Essentially a compression for the source. This is a top-down building procedure; there are conditions where this is nonoptimal in a Huffman tree. Also you’re not after the length of the source; it’s the entropy of the target you want to be minimal.
Question: If you don’t stop when building the tree, does the entropy continue decreasing (on the training data)?
Answer: No, but it is guaranteed not to increase.

Question: What finds the best split, and best in what sense?
Answer: It’s best in terms of maximum mutual information (greatest reduction in entropy H). That’s the criterion. Let’s say that you’ve got 2 questions: one question doesn’t decrease the average H; it’d be a terrible question to ask because we would be no close to an answer but would have less data (doesn’t tell us anything more specific about the y, but reduces information). It’s really an empirical science determining when you want to stop splitting the tree.

Question: Are the questions categorical like that, or is there more variety?
Answer: There’s both. Got to look at the data to determine the cluster types. Sometimes there’s things which make sense (voicing/nonvoicing, place/manner of articulating). It’s mostly data driven. It’s very difficult to put a ”meaning” to a very large machine learning system.

Question: When you say perplexity is notoriously difficult, is it difficult in a procedural or subjective way?
Answer: More in a procedural way, i.e. forgot to score something, etc. The devil is in the details (a surprising number of details.

Question: What was the size of the vocabulary for the Wall Street Journal example?
Answer: I think 60000, can’t remember exactly. Much bigger than the 1386 for the unigram. I’ll look it up and have it for next lecture.

Question: Shouldn’t the unigram be the same size as the vocabulary?
Answer: You’re not saying all words in the vocabulary are equally probable. i.e. ”’the’” is more probably than Constantinople

Question: Won’t this still give zero probabilities for words that you haven’t seen before?
Answer: Yes, but you can add a 4th term times a uniform distribution over the language to make it non-zero.

Key Terms

- **BASE FORM** (A.K.A. phoneme) - Transcription of a word as given by a dictionary[2]
- **CORPUS** - Large database; in speech recognition, generally refers to a large collection of speech samples
- **CROSS ENTROPY** - A well known measure of ”information”, which has been successfully employed in diverse fields of engineering and science, and in particular in neural computation, for about half a century[4].
- **DECISION TREE** - A decision tree takes as input an object or situation described by a set of properties, and outputs a yes/no decision. Decision trees therefore represent Boolean functions[5].
- **DYNAMIC TIME WARPING** - The cleverness of dynamic time warping lies in the computation of the distance between input streams and templates. Rather than comparing the value of the input stream at time to the template stream at time , an algorithm is used that searches the space of mappings from the time sequence of the input stream to that of the template stream, so that the total distance is minimized[8].
- **ENTROPY** - In physics, the word entropy has important physical implications as the amount of ”disorder” of a system[3].
- **GEOMETRIC MEAN** - The Geometric Mean of a sequence \( \{a_i\}_{i=1}^n \) is defined by \( G(a_1, ..., a_n) = (\Pi_{i=1}^n a_i)^{1/n} \)[3]
• **Hidden Markov Chain** - A Markov Chain where the value of the random variables is not known directly. The value of the random variables is postulated indirectly by knowledge of observation sequences.

• **Isolated Word Speech Recognizer** - An algorithm or device designed to recognize a single word of speech out of context.

• **Language Modeling** - Formal specification of the permissible structures for the language, and parsing of a sentence to see if its structure is compliant with the specification[6].

• **Levinstein Distance** - A measure of the similarity between two strings

• **Markov Chain** - A collection of random variables (where the index \( t \) runs through 0, 1, ...) having the property that, given the present, the future is conditionally independent of the past[3].

• **Perplexity** - Defined as \( 2^{\text{Entropy}} \); a measure of how complicated a language is

• **PRONLEX** - The CALLHOME American English Lexicon was originally distributed under the name COMLEX Pronouncing Lexicon (PRONLEX). The latest version of PRONLEX contains 90,988 lexical entries and includes coverage of WSJ30, WSJ64, Switchboard and CallHome English[7].

• **Pronunciation Modeling** - Lexical model; vocabulary definition and word pronunciation[6].

• **Surface Form** (A.K.A. phone) - Transcription of a word as given by a human labeler[2]

### References


