CMSC 35000 Introduction to Artificial Intelligence **Winter 2005**

Lecture 16: March 1, 2005

Lecturer: Partha Niyogi *Scribe: Paolo Codenotti*

1 Vector based models

We will look at vector based models for

1. Part of speech tagging
2. Word sense disambiguation
3. Document classification and information retrieval
4. Segmentation (phonological/morphological)

2 Part of speech tagging

Given a sentence, we want to assign to each word a tag, such as noun, verb, adjective. Such an assignment corresponds to a parsing of the sentence. We want to find the most likely tagging given a particular sequence of words. Possible solutions to this problem are:

1. *Dictionary.*
   Look up dictionary, and see what tag corresponds to what word. The problem is that some words have different possible tags: for example, in the sentence "time flies like an arrow", time could be a noun, or a verb, and the phrase would be grammatically correct.
2. *Probabilistic dictionary.*
   Word by word choose independently the most likely part of speech tag. Actual implementations of this give the correct answer about 90% of the time.
3. *Hidden Markov model.*
   We can construct a hidden Markov chain, which corresponds to a hidden context free grammar. The hidden Markov model will have as many states as parts of speech, and there are given probabilities of moving from one part of speech to the next. When in a particular state, the process will output a terminal (a word) according to a probability

distribution which depends on the state (part of speech) it's in. Now given a string of words, the problem reduces to finding the most likely state sequence, which we have already seen how to solve.

3 Word sense disambiguation

Words in any language can have several different meanings. For example, in English the word 'plant' can mean a kind of living organism, or a factory (a nuclear plant); similarly, the word 'bank' can mean the place where money is stored, or the bank of a river.

A possible approach to this problem is to look at a text, and for each word, look at the different contexts in which it appears. Thus for each word we create a vector $w_i = [c_{i,1}, c_{i,2} \cdots c_{i,N}]$, where $c_{i,j}$ = number of times that word j occurs in the context of word i, and N = number of words in the language. At this point we have represented our data as a set of vectors, and we can use any of the algorithms we have studied for classification of vectors, using any possible norm.

This approach has several problems:

1. The number of words N will be very large ($\sim$ 60000 in English), therefore the classification algorithms will be inefficient.
2. The context of a word will be no more than 100 words around it, so each of the vectors w_i will be mostly $0's$.
3. Because of 2, most vectors will be orthogonal, and thus the distance between vectors will not make much sense.

The possible solutions to these problems include:

1. *Dimensionality reduction.* Embed the space of $w_i's$ in a space of much lower dimension. This speeds up the classification algorithms, and solves the problem of many of the coordinates being 0 and thus the vectors being orthogonal.
2. *Clustering words into terms.* This is a specific method of dimensionality reduction. Do vector classification with vectors not of words, but of *terms.* A term is a natural group, or cluster of words. To cluster we can:
   (a) Look a dictionary
   (b) Cluster automatically

So we have seen that to solve this problem, we need to look at a combination of solutions to the problems of:

1. dimensionality reduction
2. clustering
3. classification

4 Document classification and retrieval

We are given documents $d_1, \cdots d_n$, represented as vectors $v_1, \cdots v_n$. We want to classify these, and then be able to retrieve documents which are interesting to us. An example of an application of this is distinguishing spam from actual emails. We want some classifier function (f, τ) such that we accept for $f(x) > \tau$. Such a classifier might run into trouble because of the high dimensionality of the space we're in (n might be very large). So we can, as above, apply a combination of dimensionality reduction, clustering and classification to solve the problem.

In general, in the analysis of the classification, we want to look at the rate of error. However, among errors we can distinguish between false positive and false negative. For example, in the case of spam, we would rather classify as relevant a message which is really spam, than erasing a relevant email thinking it was spam. This gives rise to a Receiver Operating Curve (ROC), which is a curve of false positive vs. false negative errors. In any specific problem, we will want to minimize the total number of errors, given that we are somewhere on the ROC.

5 Segmentation

The motivation for this problem comes from speech learning. When a child hear a sentence, he/she doesn't hear breaks between words. However, he/she is somehow able to learn words. Formally, let us assume that we already learned a set of phonimes $P = \{P_1, \cdots P_k\}$. We want to learn a lexicon $W = \{w_1, \cdots w_N\}$, where $w_i \in P^* \, \forall i$. We are given sentences $s_1, s_2, \cdots$, where $s_i \in W^*$. However, we only see the representation of s_i as $s_i \in P^*$. A learning algorithm is a function $\mathcal{A} : \bigcup_{k \geq 1} P^k \to \mathcal{L}$, where $P^k = (P^*)^k$, and $\mathcal{L}$ = collection of all finite subsets of P^*.

For example, if $P = \{a, b\}$, if the target is $W = \{a, b\}$, then we get all possible sentences. If the target is $W = \{aa, bb\}$, then we get all sentences where a's and b's appear in pairs.

We note that classifying every phonime as a word, or classifying every sentence as a word are both consistent with the information received. Therefore we cannot learn a lexicon if we don't have any a priori knowledge about $\mathcal{L}$. This suggests there is an equivalent of Gold's theorem for this setting. What is the analogue of an infinite language in Gold's theorem? What is the analogue of a locking sequence?