A classic PCFG is a generative parsing model, a model of the joint probability $P(x, y)$ of input and output. In this lecture, I will first review the distinction between generative and discriminative models (§1) and introduce the widely used log-linear models (§2). I will then discuss three different ways in which these models are used in current parsing systems: local discriminative models (§3), global discriminative models (§4), and discriminative rerankers (§5).

1. Generative and Discriminative Models

The PCFG model considered in previous lectures is a generative model in the sense that is models the joint probability $P(x, y)$ of the input $x$ and output $y$ (which in a PCFG is equivalent to $P(y)$ when restricted to the subspace GEN($x$)). Generative models have many advantages, such as the possibility of deriving the related probabilities $P(y|x)$ and $P(x)$ through conditionalization and marginalization, which makes it possible to use the same model for both parsing and language modeling. Another attractive property is the fact that the learning problem for these models often has a clean analytical solution, such as the relative frequency estimation used in treebank grammars, which makes learning both simple and efficient.

The main drawback with generative models is that they force us to make rigid independence assumptions, thereby severely restricting the range of dependencies that can be taken into account for disambiguation. As we have seen in the previous section, the search for more adequate independence assumptions has been an important driving force in research on statistical parsing, but we have also seen that more complex models inevitably makes parsing computationally harder and that we must therefore often resort to approximate algorithms. Finally, it has been pointed out that the usual approach to training a generative statistical parser maximizes a quantity – usually the joint probability of inputs and outputs in the training set – that is only indirectly related to the goal of parsing, that is, to maximize the accuracy of the parser on unseen sentences.

A discriminative model only makes use of the conditional probability $P(y|x)$ of a candidate analysis $y$ given the input sentence $x$. Although this means that it is no longer possible to derive the joint probability $P(x, y)$, it has the distinct advantage that we have more freedom in defining features and in particular can incorporate arbitrary features over the input. It also

Date: 2013-02-15.
means that we can train the model to maximize the probability of the output given the input or even to minimize a loss function in mapping inputs to outputs. On the downside, discriminative training methods normally require the use of numerical optimization techniques, which can be computationally intensive, and the use of more complex features also makes the parsing problem harder.

Sometimes a distinction is made between two types of discriminative models: conditional models, which explicitly model the conditional distribution \( P(y|x) \) of outputs given inputs, and (purely) discriminative models, which try to optimise the mapping from inputs to outputs without explicitly modeling a conditional distribution (for example, by directly trying to minimize the error rate of the parser). For now, we are going to concentrate on conditional models, which are widely used in phrase structure parsing and related frameworks. Later in the course we will return to other kinds of discriminative models, which are more common in dependency parsing.

2. Log-Linear Models

A log-linear model of the distribution \( P(y|x) \) has the following form:

\[
P(y|x) = \frac{\exp \left[ \sum_{i=1}^{k} f_i(x, y) \cdot w_i \right]}{\sum_{y' \in \text{GEN}(x)} \exp \left[ \sum_{i=1}^{k} f_i(x, y') \cdot w_i \right]}
\]

where every \( f_i(x, y) \) is a numerical feature encoding some property of the pair \((x, y)\) and \( w_i \) is the real-valued weight associated with the feature function \( f_i \). This is called a log-linear model because the primary scoring function \( \sum_{i=1}^{k} f_i(x, y) \cdot w_i \) is a linear combination of weighted features in logarithmic space.\(^1\) In general, \( f_i(x, y) \) can take any real number as its value, but in practice it is normally either a binary variable representing the presence (1) or absence (0) of some property, or an integer representing a count of some phenomenon. By contrast, the weight \( w_i \) is typically real-valued and can be either positive or negative (or zero), which means that the sum \( \sum_{i=1}^{k} f_i(x, y) \cdot w_i \) can also be either positive or negative (or zero). So by exponentiating this sum, we guarantee that the numerator in Equation 1 is always positive.\(^2\) And by summing over all candidate outputs \( y' \) in the denominator, we guarantee that all conditional probabilities are between 0 and 1 and sum to 1:

\[
0 \leq P(y|x) \leq 1
\]

\[
\sum_{y'} P(y'|x) = 1
\]

This means that probabilities are normalized only relative to a given input \( x \), and the model does not give us any information about the probability of the input \( x \) itself, which is why it is not a generative model.

It is worth noting that we can use a PCFG to define a log-linear model by defining one feature function \( f_i(x, y) \) for each context-free rule and by setting \( f_i(x, y) \) to \( \text{COUNT}(i, y) \) and \( w_i \) to \( \log q_i \). However, the real power of log-linear models for syntactic parsing comes from the fact that we are no longer limited to features that are assumed to be statistically independent (like the PCFG productions used to derive a parse tree) but are free to include arbitrary features over both the input and the output. This includes, for example, lexical co-occurrence features and features over long-distance relations in parse trees. Such features are now unproblematic from a statistical modeling perspective, but they can still cause problems with respect to computational efficiency in both decoding and learning.

---

\(^1\)Recall that exponentiation is the inverse of the logarithmic function so that \( \exp[\log a] = a \).

\(^2\)If \( a < 0 \) then \( 0 < \exp[a] < 1 \); if \( a = 0 \) then \( \exp[a] = 1 \); if \( a > 0 \) then \( \exp[a] > 1 \).
The decoding or parsing problem for a log-linear model given input $x$ is as usual an argmax search for the optimal output $y^*$:

$$
\begin{align*}
    y^* &= \arg\max_y P(y|x) \\
    &= \arg\max_y \frac{\exp[\sum_{i=1}^{k} f_i(x, y) \cdot w_i]}{\sum_{y' \in \text{GEN}(x)} \exp[\sum_{i=1}^{k} f_i(x, y') \cdot w_i]} \\
    &= \arg\max_y \exp \left[ \sum_{i=1}^{k} f_i(x, y) \cdot w_i \right] \\
    &= \arg\max_y \sum_{i=1}^{k} f_i(x, y) \cdot w_i
\end{align*}
$$

Although we can simplify the computation at parsing time by disregarding the denominator (the so-called partition function) and even the exponentiation step, we may nevertheless have to search over an exponentially large set of candidate outputs, and whether there is an efficient algorithm for doing this depends on the complexity of our feature representations. For example, if we want to use a standard CKY-style algorithm, features have to be local to elementary productions, just as in the old PCFG model.

Learning the weights of a log-linear model can be done in many different ways, but the standard approach is based on maximum conditional likelihood estimation, where we try to learn weights that maximise the probability of outputs given inputs in our training set. The bad news is that, unlike the case for maximum joint likelihood estimation that we used for treebank PCFGs, there is no closed form solution to this problem, so we have to use numerical methods. The good news is that our objective function is convex, so that we can use gradient-based methods to derive an exact solution.

However, in order to use these learning methods, we again need to be able to efficiently compute statistics over exponentially large search spaces. In particular, we need to be able to compute the so-called partition function appearing in the denominator of Equation 1. And whether there is an efficient dynamic programming algorithm for doing this again depends on our choice of feature representations.

Summing up, we can say that conditional probability models in the form of log-linear models give us more freedom in designing complex and potentially informative features for disambiguation, without having to worry about statistical independence. However, in practice, we always have to make compromises in order to guarantee efficient decoding and parameter estimation. In what follows, we will look at three different ways of balancing these conflicting requirements.

3. **Local Discriminative Models**

In a local discriminative model, we give up the idea of having a conditional model of complete output structures (given inputs) in favor of a model of smaller local structures or derivation steps and assume that we can arrive at a globally optimal solution by making locally optimal choices. This is an approximation compared to the global models that we will consider later, but it has the advantage that we can use arbitrarily complex features and that we can perform very efficient parsing, often with linear time complexity.

Local discriminative models normally take the form of history-based models, where the derivation of a candidate analysis $y$ is modeled as a sequence of decisions with each decision conditioned on relevant parts of the derivation history. Because they are conditional models, they also include the input sentence $x$ as a conditioning variable:

$$
P(y|x) = \prod_{i=1}^{m} P(d_i | \Phi(d_1, \ldots, d_{i-1}, x))
$$

This means that we are guaranteed to find a **global** maximum, and do not risk getting trapped in a **local** maximum, as was the case when using EM to perform maximum marginal likelihood estimation.
This makes it possible to condition decisions on properties of the input, for example by using a lookahead such that the next $k$ tokens of the input sentence can influence the probability of a given decision. Therefore, conditional history-based models have often been used to construct incremental and near-deterministic parsers that parse a sentence in a single left-to-right pass over the input, using beam search or some other pruning strategy to efficiently compute an approximation of the most probable analysis $y$ given the input sentence $x$. This guarantees that parsing can be performed efficiently even with exponentially many derivations and a model structure that is often unsuited for dynamic programming. Because the conditional model is only over local decisions, these models can also be trained very efficiently using logistic regression or similar methods.

Conditional history-based models were first proposed in phrase structure parsing, as a way of introducing more structural context for disambiguation compared to standard grammar rules (Briscoe & Carroll, 1993; Carroll & Briscoe, 1996; Jelinek et al., 1994; Magerman, 1995). Today it is generally considered that, although parsers based on such models can be implemented very efficiently to run in linear time (Ratnaparkhi, 1997, 1999), their accuracy lags a bit behind the best-performing generative models and global discriminative models.

Summing up, in syntactic parsers based on local, discriminative models, the generative component $\text{GEN}(x)$ is typically defined by a derivational process, such as a history-based model or a bottom-up parsing algorithm, while the evaluative component $\text{EVAL}(x, y)$ is a probability model of complete parse trees given inputs and parts of the derivation history, which can be combined to derive a conditional probability for the entire parse tree. Because these parsers use heuristic search algorithms, such as beam search, they are not guaranteed to find the most probably parse tree for a given sentence.

4. Global Discriminative Models

In a global discriminative model, we maintain a conditional model over the entire output structure (given the input), which means that we can use exact inference both in learning – to induce the model that truly maximizes the conditional likelihood of outputs given inputs in the training set – and in decoding – to find the parse that truly maximizes the conditional probability given the input under the current model. However, in order to make learning and decoding tractable computationally, this requires that our feature model factors into reasonably local, non-overlapping structures, so that we can use dynamic programming in essentially the same way as for standard PCFG parsing. The limited scope of features is the major drawback of this approach.

Global discriminative models have been used as an alternative to the generative PCFG model, but are especially popular in grammar-based parsers for formalisms like LFG (Johnson et al., 1999; Riezler et al., 2002), HPSG (Toutanova et al., 2002; Miyao et al., 2003), and CCG (Clark & Curran, 2004). In these systems, the generative component $\text{GEN}(x)$ is typically defined by a grammar with hard constraints, while the evaluative component $\text{EVAL}(x, y)$ is a conditional probability model of complete parse trees given inputs. Although these systems can in principle use exact inference, it is often necessary to apply pruning for efficiency reasons.

5. Reranking

A third way of managing the complexity of conditional models is to limit their applicability to a small finite set of candidate structures generated by a different (more efficient model).
These parsers are known as reranking parsers, because the global discriminative model is used to rerank the n top candidates already ranked by the base parser, which is often a generative PCFG parser. Since the set of candidate parses is small enough to be enumerated efficiently without dynamic programming, this approach enables truly global features without imposing any particular factorization. Applying a discriminative reranker on top of a generative base parser therefore often leads to a significant improvement in parsing accuracy (Collins, 2000; Collins & Duffy, 2002; Collins & Koo, 2005; Charniak & Johnson, 2005). However, it is worth noting that the single most important feature in the global discriminative model is normally the probability assigned to an analysis by the generative base parser.

References


