Discriminative Models

Miguel Ballesteros
7-11

Again some materials by Joakim Nivre.

Generative vs Discriminative

- A classic PCFG is a generative parsing model, a model of the joint probability P (x, y) of input and output.
- This is what we've seen so far.
- **Generative models** have many advantages:
  - possibility of deriving the related probabilities P(y|x) and P(x)
  - It is possible to use the same model for both parsing and language modeling.
  - the learning problem is normally a clean analytical solution: the relative frequency estimation used in treebank

Generative vs Discriminative

- Drawback of generative models:
  - they force us to make rigid independence assumptions, thereby severely restricting the range of dependencies that can be taken into account for disambiguation.
  - the search for more adequate independence assumptions has been an important driving force in research on statistical parsing

Generative vs **Discriminative**

- A discriminative model only makes use of the conditional probability P(y|x) of a candidate analysis *y* given the input sentence *x*.
- This means that it is no longer possible to derive the joint probability P(x, y)
- However, we have more freedom in defining features and in particular we can incorporate arbitrary features over the input.

Generative vs **Discriminative**

- We can train the model to maximize the probability of the output given the input
- We can minimize a loss function in mapping inputs to outputs.

Generative vs **Discriminative**

- Drawbacks of discriminative models:
  - Discriminative training methods normally require the use of numerical optimization techniques
    - computationally intensive
    - use of more complex features also makes the parsing problem harder.

Generative vs **Discriminative**

- Sometimes a distinction is made between two types of discriminative models:
  - Conditional models,
    - model the conditional distribution P(y|x) of outputs given inputs
  - (purely) discriminative models,
    - which try to optimize 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).

Log-Linear Models

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

$$P(y|x) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y') \cdot \mathbf{w}_i\right]}$$

- Every $\mathbf{f_i(x,y)}$ is a numerical feature encoding some property of the pair (x,y) and $\mathbf{w_i}$ is the real-valued weight associated with the feature function $\mathbf{f_i}$

  It is called log-linear model because the primary scoring function is a linear combination of weighted features in logarithmic space

$$\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i$$

Note that exp[log a] =a (exponentiation is the inverse of the logarithmic function)

Log-Linear Models

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

$$P(y|x) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y') \cdot \mathbf{w}_i\right]}$$

  - In general, $\mathbf{f_i(x,y)}$ can take any real number as its value.
  - In practice, it is normally a variable representing the presence (1) or absence (0) of some property.
    - Or an integer that counts some phenomena.

Log-Linear Models

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

$$P(y|x) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y') \cdot \mathbf{w}_i\right]}$$

  - $\mathbf{w}_i$ is tipically real-valued and can be either positive or negative (or zero), which means that $\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i$ can also be positive, negative or zero.
  - By doing exp [sum] we guarantee that the numerator is always positive.

Log-Linear Models

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

$$P(y|x) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x, y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x, y') \cdot \mathbf{w}_i\right]}$$

- $\mathbf{w}_i$ is tipically r
  negative (or z
  also be positi
- By doing exp
  always positiv

And by summing overall candidate outputs **y** in the denominator, we guarantee that all conditional probabilities are between 0 and 1 and sum to 1:

Log-Linear Models

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

$$P(y|x) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y') \cdot \mathbf{w}_i\right]}$$

  - $\mathbf{w}_i$ is tipically real-valued and can be either positive or negative (or zero), which means that $\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i$ can also be positive, negative or zero.
  - By doing exp [sum] we guarantee that the numerator is always positive.

And this is why we call this log-linear models

Log-Linear Models

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

$$P(y|x) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x, y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x, y') \cdot \mathbf{w}_i\right]}$$

  - This means that probabilities are normalized only relative to a given input *x* and the model does not give us any information about the input x itself.
    - It is not a generative model.... :-)

Log-Linear Models

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

$$P(y|x) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x, y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x, y') \cdot \mathbf{w}_i\right]}$$

  - We can use a PCFG to define a log-linear model by defining one feature function $\mathbf{f_i}$(x, y) for each context-free rule and by setting $\mathbf{f_i}$ (x, y) to count(i, y) and $\mathbf{w_i}$ to **log ($\mathbf{q_i}$) .**

Log-Linear Models

- 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)
- They are free to include arbitrary features over both the input and the output.
  - lexical co-occurrence features.
  - long-distance relations in parse trees features.

Decoding of Log-Linear Models

- The decoding or parsing problem for a log-linear model given input x is as usual an argmax search for the optimal output y^* :

$$
\begin{aligned}
y^* &= \text{argmax}_y \, P(y|x) \\
&= \text{argmax}_y \, \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \text{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y') \cdot \mathbf{w}_i\right]} \\
&= \text{argmax}_y \, \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right] \\
&= \text{argmax}_y \, \sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i
\end{aligned}
$$

Decoding of Log-Linear Models

We can simplify the computation at parsing time by disregarding the denominator 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.

$$
\begin{aligned}
y^* &= \text{argmax}_y \, P(y|x) \\
&= \text{argmax}_y \, \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \text{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y') \cdot \mathbf{w}_i\right]} \\
&= \text{argmax}_y \, \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right] \\
&= \text{argmax}_y \, \sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i
\end{aligned}
$$

Log-Linear Models

- Learning the weights of a log-linear model can be done in many different ways,
- The standard approach is based on maximum conditional likelihood estimation
  - we try to learn weights that maximize the probability of outputs given inputs in our training set
- In order to use these learning methods, we again need to be able to efficiently compute statistics over exponentially large search spaces.

Global Discriminative Models

- In a global discriminative model, we maintain a conditional model over the **entire output structure** (given the input)
  - we can use exact inference both in learning and training:
    - to induce the model that truly maximizes the conditional likelihood of outputs given inputs in the training set (training).
    - to find the parse that truly maximizes the conditional probability given the input under the current model (decoding).

$$P(y|x) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y) \cdot \mathbf{w}_i\right]}{\sum_{y' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(x,y') \cdot \mathbf{w}_i\right]}$$

Global Discriminative Models

- 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. (**CKY**)
- The limited scope of features is the major drawback of this approach.

Local Discriminative Models

- We give up the idea of having a conditional model of complete output structures (given inputs)
- We build a model of smaller local structures or derivation steps
- We assume that we can arrive at a globally optimal solution by making locally optimal choices.

Local Discriminative Models

- We can use arbitrarily complex features over the history and lookahead over the future.

- We can perform very efficient parsing, often with linear time complexity.

Local Discriminative Models

- Local discriminative models normally take the form of history-based models
- Derivation of a candidate analysis *y* is modeled as a sequence of decisions with each decision conditioned on relevant parts of the derivation history.

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

$$P(d_i|\Phi(d_1, \ldots, d_{i-1}, x)) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(\Phi(d_1, \ldots, d_{i-1}), d_i) \cdot \mathbf{w}_i\right]}{\sum_{d' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(\Phi(d_1, \ldots, d_{i-1}), d') \cdot \mathbf{w}_i\right]}$$

Local Discriminative Models

- Local discriminative models normally take the form of history-based models
- Derivation of a can eled as a sequence of d sion conditioned on rele on history.

Φ is the feature function

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

$$P(d_i|\Phi(d_1, \ldots, d_{i-1}, x)) = \frac{\exp\left[\sum_{i=1}^{k} \mathbf{f}_i(\Phi(d_1, \ldots, d_{i-1}), d_i) \cdot \mathbf{w}_i\right]}{\sum_{d' \in \mathrm{GEN}(x)} \exp\left[\sum_{i=1}^{k} \mathbf{f}_i(\Phi(d_1, \ldots, d_{i-1}), d') \cdot \mathbf{w}_i\right]}$$

Local Discriminative Models

- 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

Local Discriminative Models

- 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

You already know about a local discriminative model... the Shift-Reduce Algorithm for CFGs that we saw a couple of lectures back.

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.

Reranking

- The global discriminative model is used to rerank the n top candidates already ranked by a base parser, which is often a generative PCFG parser.

- 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.