Chapter 8

Conditional Random Fields

8.1 Introduction

Discriminative models model the dependence of an unobserved variable y on an observed value x. We have already looked at a few, such as logistic regression and neural networks. We now turn to a new model, Conditional Random Fields, abbreviated as CRFs to again look at the task of Sequence Labeling. CRFs attempt to address some of the issues that arise in practice when using Hidden Markov Models.

8.2 Hidden Markov Models

Recall from earlier in the semester that we used Hidden Markov Models, or HMMs, to model sequences of unobserved variables. As you saw in earlier lectures and through your homework, HMMs can be very useful for numerous tasks, including ones where we are trying to fit a model to natural language data. We are now going to turn to a related model, Conditional Random Fields, but first it is worth discussing some inherent properties of HMMs from which we can motivate this new model.

8.2.1 Latent Variables

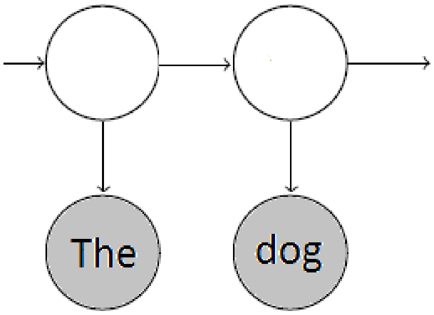
As we have discussed multiple repeatedly in the class, latent variables, also known as hidden variables, are parts of our model which we cannot directly observe. For instance, in our part-of-speech labeling task, we explicitly state that an observed word has a part-of-speech tag, but we do not know what it is. From our sequence of words, our observed variables, we fit a model of the most likely sequences of tags that generate these words. We made some assumptions for this model, such as the fact that each tag in the sequence generates the word associated with it.

8.2.2 Markov Property

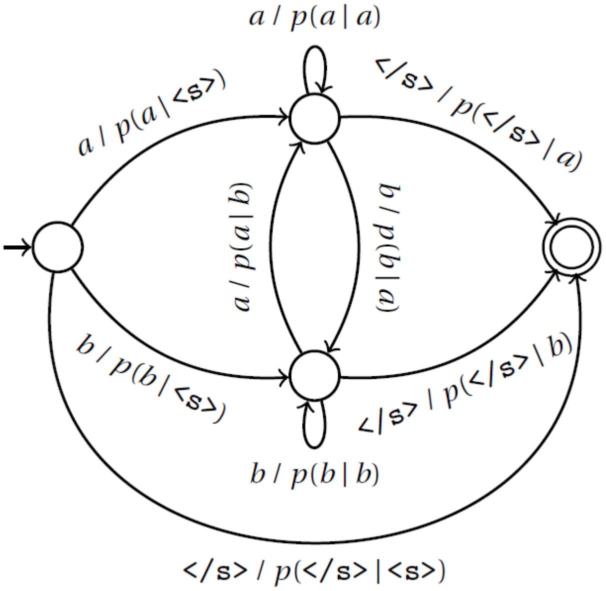
A key property of HMMs is the Markov Assumption. Recall that the Markov property is when the value of the next state only depends on the current state. It does not depend on the sequence of events that preceded it, but only on the state that the model is in at that point. Naturally, preceding states in a sequence have gotten the model to the current state it is in, but the next state is *only* dependent on the current state.

8.3 Graphical Models vs. Finite State Machines

So far this semester, we have talked about numerous learning models. However, we have not mentioned this yet, but many of them can be classified into a taxonomy of models called Probabilistic Graphical Models. These models use graph formalisms to represent dependencies between variables. In fact, we can represent HMMs as graphical models. Graphical models explicitly model the statistical dependencies between two variables, and within HMMs, you can see this as the observed variable being *generated* by an unobserved variable.



Looking at the figure, it can appear quite close to how we draw Finite State Automatas (FSAs). However, there is a difference between Graphical Models and Finite State Automatas. FSAs represent how we transition between different states. They can be weighted or even probabilistic, but inherently, they are modeling states. There is an inherent difference between modeling states and modeling the relationship between two variables.



Directed Graph

So far, most of the graphical models that we have discussed have been directed graphs. These models explicitly model the generative relationships between variables. In our part-of-speech tagging with HMMs, the assumption our model made was that a tag generated a word. The tags always generated the word and the dependency is explicitly modeled. Graphically, we represent this as our observed state, a word, being conditioned on an unobserved state that represents the part-of-speech tag. In your homework, when you implemented an HMM, you likely had an emission probability matrix that gave the probability of every word given every tag.

Undirected Graph

There is no particular reason why we must limit ourselves to directed graphs, and we can use undirected graphs within graphcial models as well. This has a few implications. Values are conditioned off of each other and dependencies go both ways, rather than just from one variable to another.

These models allow for greater flexibility and can even represent cyclic dependencies. However, it must be stressed that they cannot represent induced dependencies. When designing a learning algorithm, you have to make a choice of how you want to model dependencies and then choose either a directed or an undirected graphical model. There are benefits and downsides to both classes.

8.4 Generative and Discriminative Models

A generative model is a joint distribution that models $p(t, w) = p(t)p(w|t)$ or $p(w)p(t|w)$. The key part of the definition of a generative model is the joint distribution. From that, it describes how t *generates* w. A discriminative model is a conditional distribution where the classification is modeled directly. A discriminative model just cares about the conditional distribution $p(t|w)$. Note that this is a subtle difference and that you can get from one to the other through Bayes Rule. The main difference between these models is that the discriminative model does not model $p(w)$.

In practice, estimating $p(w)p(t|w)$ and then finding $p(t|w)$ yields a different answer than just finding $p(t|w)$ directly. This is due to the fact that we never actually have the true distribution in our training data.

Already in this class, we have seen two similar models with one being generative and the other discriminiative. The first learning algorithm we looked at, Naive Bayes, is a generative model. In your homework, you implemented this and the discriminative model, Logisitic Regression. The goal of both of these methods was to classify data, but they approached it from different perspectives.

8.5 Conditional Random Fields

Conditional Random Fields are a type of model that explicitly models conditional distributions. They are an undirected graphical model that is discriminative. Most often, they are used to model latent variables in sequences - just like HMMs. However, they relax strong independence assumptions and focus directly on modeling the conditional dependencies.

We define a graph, G, to consist of vertices $\boldsymbol{t}$ (tags) and $\boldsymbol{w}$ (observed words) as well as edges E. The variables $\boldsymbol{t}$ conditioned on $\boldsymbol{w}$ observe the Markov Property with respect to the graph. This means that the nodes can be divided into two separate sets of observed and unobserved variables. Formally:

$$P(\boldsymbol{t_v}|\boldsymbol{w}, \boldsymbol{t_z}, z \neq v) = p(\boldsymbol{t_v}|\boldsymbol{w}, \boldsymbol{t_z}, z \sim v)$$

where $z \sim v$ means that z and v are neighbors in G.

Based on this definition and the Markov Property Assumption for the tags, we can reformulate CRFs as:

$$p(t|w) = \frac{1}{Z(w)} \prod_{n=1}^{N} \Psi_n(t_n, t_{n-1}, w_n)$$

where $\frac{1}{Z(w)}$ is a normalization factor in order to assure that we have a proper probability distribution. Though nuanced, note that Z is now a function of w which makes our computation easier. We define:

$$\Psi_n(t_n, t_{n-1}, w_n) = exp(\sum_k \theta_k f_k(t_n, t_{n-1}, w_n))$$

This is similar to what we did when we went from Naive Bayes to Logistic Regression and introduced an exponentiation of sums. As before, we define k to be our features. f_k is a feature function that maps information from the current tag, the previous tag and the current word into a score. θ_k is a weight parameter for that feature function. This allows for easy generalization of features into our CRFs.

8.6 Training on CRFs

In most use cases, a CRF graph is modeled as a tree or chain, as we have for part-of-speech tagging. Fortunately training these models also use the forward-backward algorithm we have discussed in this class. There are some minor changes. We define our forward values (α) as:

$$\alpha_n(j) = \sum_{t<1...n-1>} \Psi_t(j, t_{n-1}, w_n) \prod_{n'=1}^{n-1} \Psi_{n'}(t_{n'}, t_{n'-1}, w_{n'})$$

And likewise define our backwards values as:

$$\beta_n(i) = \sum_{t<n+1...N>} \prod_{n'=n+1}^{N} \Psi_{n'}(t_{n'}, t_{n'-1}, w_{n'})$$

The results give us $Z(w)$ which is $\beta_0(t_0) = \sum \alpha_T$. Our marginal distribution is defined as:

$$p(t_n|w) = \frac{1}{Z(w)} \alpha_n(t_n) \beta_n(y_n)$$

References

Lafferty, John, Andrew McCallum, and Fernando CN Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In *Proceedings of the 18th International Conference on Machine Learning*, pp. 282-289.

Sutton, Charles and Andrew McCallum (2011). "An Introduction to Conditional Random Fields". In *Foundations and Trends in Machine Learning*, pp. 267-373.