CSC412 – Probabilistic Learning & Reasoning Sam Roweis

Lecture 23:

Features and Maximum Entropy Models

April 3, 2006

Constrained Clique Potentials

- So far we have discussed the most general form of a graphical model in which maximal cliques are parametrized by general potential functions $\psi_C(\mathbf{x}_C)$.
- But for large cliques these general potentials are exponentially costly for inference and have exponential numbers of parameters that we must learn from limited data.
- One solution: change the graphical model to make cliques smaller. But this changes the dependencies, and may force us to make more independence assumptions than we would like.
- Another solution: keep the same graphical model, but use a less general parameterization of the clique potentials.
- This is the idea behind *feature-based models*.
  It is also the same idea behind *factor graphs* which we already saw.

Features

- Consider a clique $\mathbf{x}_C$ of random variables in a graphical model, e.g. three consecutive characters $c_1c_2c_3$ in a string of English text.
- How would we build a model of $p(c_1c_2c_3)$?
- The full joint clique potential would be huge: $26^3 - 1$ parameters.
- However, we often know that some particular joint settings of the variables in a clique are quite likely or quite unlikely.
  e.g. `ing, ate, ion, ?ed, qu?, jkx, zzz,...`
- A "feature" is a function which is uniform over all joint settings except a few particulat ones on which it is high or low.
- For example, we might have $f_{\texttt{ing}}(c_1c_2c_3)$ which is 1 if the string is `'ing'` and 0 otherwise, and similar features for `'?ed'`, etc.
- We can also define features when the inputs are continuous. Then the idea of a cell on which it is active disappears, but we might still have a compact parameterization of the feature.

Features as Micropotentials

- By exponentiating them, each feature function can be made into a "micropotential". We can multiply these micropotentials together to get a clique potential.
- If we use a relatively small number of features, we can combine them into clique potentials using only a reasonable number of parameters but preserving the dependencies that we want.
- Example: a clique potential $\psi(c_1, c_2, c_3)$ could be expressed as

$$\psi(c_1, c_2, c_3) = e^{\theta_{\texttt{ing}} f_{\texttt{ing}}} e^{\theta_{\texttt{?ed}} f_{\texttt{?ed}}} \ldots$$
$$= \exp\left[\sum_{i=1}^{K} \theta_i f_i(c_1, c_2, c_3)\right]$$

- This is still a potential over 26^3 possible settings, but only uses K parameters if there are K features.

Combining Features

- Each feature has a weight which tells us how important it is and whether it increases or decreases the probability of the clique.
- This is a generalized exponential family distribution:

$$p(c_1c_2c_3) \propto \exp\{\ \theta_{\texttt{ing}}f_{\texttt{ing}}(c_1c_2c_3) + \theta_{\texttt{?ed}}f_{\texttt{?ed}}(c_1c_2c_3) + \theta_{\texttt{qu?}}f_{\texttt{qu?}}(c_1c_2c_3) + \theta_{\texttt{zzz}}f_{\texttt{zzz}}(c_1c_2c_3) + \ldots\}$$

- In general, the features may be overlapping, unconstrained indicators of any function of the clique variables:

$$\begin{aligned}\psi_c(\mathbf{x}_c) &\equiv \prod_{i \in I_C} \exp\{\theta_i f_i(\mathbf{x}_{Ci})\} \\ &= \exp\left\{\sum_{i \in I_C} \theta_i f_i(\mathbf{x}_{Ci})\right\}\end{aligned}$$

- How can we combine feature into a probability model?

Feature Based Model

- We can multiply these clique potentials as usual:

$$\begin{aligned}p(\mathbf{x}|\theta) &= \frac{1}{Z(\theta)} \prod_C \psi_C(\mathbf{x}_C) \\ &= \frac{1}{Z(\theta)} \prod_C \exp\left\{\sum_{i \in I_C} \theta_i f_i(\mathbf{x}_{Ci})\right\} \\ &= \frac{1}{Z(\theta)} \exp\left\{\sum_C \sum_{i \in I_C} \theta_i f_i(\mathbf{x}_{Ci})\right\}\end{aligned}$$

- However, in general we can forget about associating features with cliques and just use a simplified form:

$$p(\mathbf{x}|\theta) = \frac{1}{Z(\theta)} \exp\left\{\sum_i \theta_i f_i(\mathbf{x}_{Ci})\right\}$$

- This is just our friend the exponential model, with the features as sufficient statistics! We don't really need the graphical model at all.

Review: Maximum Likelihood

$$\begin{aligned}\ell(\theta; \mathcal{D}) &= \sum_{\mathbf{x}} n(\mathbf{x}) \log p(\mathbf{x}|\theta) \\ &= \sum_{\mathbf{x}} n(\mathbf{x}) \left(\sum_i \theta_i f_i(\mathbf{x}) - \log Z(\theta)\right) \\ &= \sum_{\mathbf{x}} n(\mathbf{x}) \sum_i \theta_i f_i(\mathbf{x}) - N \log Z(\theta) \\ \frac{\partial \ell}{\partial \theta_i} &= \sum_{\mathbf{x}} n(\mathbf{x}) f_i(\mathbf{x}) - N \frac{\partial}{\partial \theta_i} \log Z(\theta) \\ &= \sum_{\mathbf{x}} n(\mathbf{x}) f_i(\mathbf{x}) - N \sum_{\mathbf{x}} p(\mathbf{x}|\theta) f_i(\mathbf{x}) \\ \Rightarrow \quad \sum_{\mathbf{x}} p(\mathbf{x}|\theta) f_i(\mathbf{x}) &= \sum_{\mathbf{x}} \frac{n(\mathbf{x})}{N} f_i(\mathbf{x}) = \sum_{\mathbf{x}} \bar{p}(\mathbf{x}) f_i(\mathbf{x})\end{aligned}$$

Derivative of log partition function is the expectation of the feature.
At ML estimate, model expectations match empirical feature counts.

Maximum Entropy

- We can approach the modeling problem from an entirely different point of view. *Begin* with some fixed feature expectations:

$$\sum_{\mathbf{x}} p(\mathbf{x}) f_i(\mathbf{x}) = \alpha_i$$

- Assuming expectations are consistent, there may exist many distributions which satisfy them. Which one should we select? The most uncertain or flexible one:
  i.e. the one with *maximum entropy*.
- This yields a new optimization problem:

$$\begin{aligned}\max\ \mathcal{H}[p(\mathbf{x})] &= -\sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) \\ \text{subject to } \sum_{\mathbf{x}} p(\mathbf{x}) f_i(\mathbf{x}) &= \alpha_i \\ \sum_{\mathbf{x}} p(\mathbf{x}) &= 1\end{aligned}$$

Solution to the MaxEnt Problem

- To solve the maxent problem, we use Lagrange multipliers:

$$L = -\sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) - \sum_i \theta_i \left(\sum_{\mathbf{x}} p(\mathbf{x}) f_i(\mathbf{x}) - \alpha_i \right) - \mu \left(\sum_{\mathbf{x}} p(\mathbf{x}) - 1 \right)$$

$$\frac{\partial L}{\partial p(\mathbf{x})} = 1 + \log p(\mathbf{x}) - \sum_i \theta_i f_i(\mathbf{x}) - \mu$$

$$p^*(\mathbf{x}) = e^{\mu - 1} \exp\left\{ \sum_i \theta_i f_i(\mathbf{x}) \right\}$$

$$Z(\theta) = e^{1-\mu} = \sum_{\mathbf{x}} \exp\left\{ \sum_i \theta_i f_i(\mathbf{x}) \right\}$$

$$p(\mathbf{x}|\theta) = \frac{1}{Z(\theta)} \exp\left\{ \sum_i \theta_i f_i(\mathbf{x}) \right\}$$

- So feature constraints + maxent implies exponential family.
- Problem is convex, so solution is unique.

Constraints from Data

- Where do the constraints α_i come from?
- Just as before, measure the empirical counts on the training data:

$$\alpha_i = \sum_{\mathbf{x}} \frac{n(\mathbf{x})}{N} f_i(\mathbf{x}) = \sum_{\mathbf{x}} \bar{p}(\mathbf{x}) f_i(\mathbf{x})$$

- This also ensures consistency automatically.
- Known as the "method of moments". (c.f. law of large numbers)
- We have seen a case of *convex duality*:
  In one case, we assume exponential family and show that ML implies feature expectations match observed counts.
  In the other case, we assume model expectations must match empirical feature counts and show that maxent implies exponential family distribution.

Geometric Interpretation

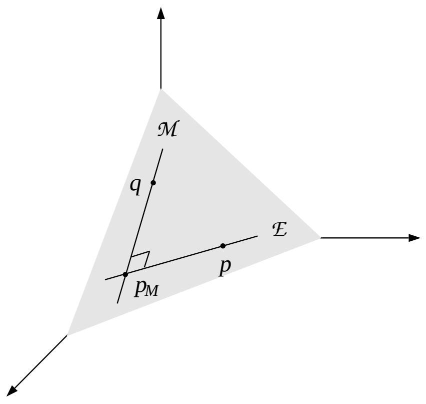
Define two submanifolds on the probability simplex $p(\mathbf{x})$.

The first is $\mathcal{E}$, the set of all exponential family distributions based on a particular set of features $f_i(\mathbf{x})$.

The second is $\mathcal{M}$, the set of all distributions that satisfy the feature expectation constraints.

They intersect at a single distribution p_M, the maxent, maximum likelihood distribution.

A generalized Pythagorean theorem holds for cross entropies.



$$\mathcal{E} = \left\{ p(\mathbf{x}) \ : \ \frac{1}{Z(\theta)} \exp\left\{ \sum_i \theta_i f_i(\mathbf{x}) \right\} \right\}$$

$$\mathcal{M} = \left\{ p(\mathbf{x}) \ : \ \sum_{\mathbf{x}} p(\mathbf{x}|\theta) f_i(\mathbf{x}) = \sum_{\mathbf{x}} \frac{n(\mathbf{x})}{N} f_i(\mathbf{x}) = \sum_{\mathbf{x}} \bar{p}(\mathbf{x}) f_i(\mathbf{x}) \right\}$$

KL$[q \| p]$ = KL$[q \| p_M]$ + KL$[\ p_M \| p\ ]$

KL Divergence Interpretations

- The generalized Pythagorean theorem allows us to interpret maxent and ML models in a very succinct way:
  KL$[q \| p]$ = KL$[q \| p_M]$ + KL$[\ p_M \| p\ ]$
- The maxent problem (actually the generalization of it to any reference distribution $h(x)$ beyond the uniform) is:

$$\min_q \mathrm{KL}[q \| h] \qquad \text{subj.to.} \quad q \in \mathcal{M}$$

  in other words, find the distribution q to which h is closest, which still respects the moment constraints.
- The ML problem, given empirical counts $\tilde{p}$ is:

$$\min_p \mathrm{KL}[\tilde{p} \| p] \qquad \text{subj.to.} \quad p \in \mathcal{E}$$

  in other words, find the distribution closest to $\tilde{p}$ which is still in the exponential family.

Conditional MaxEnt Models

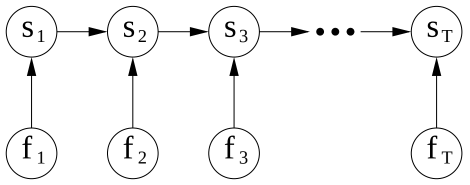
- So far we have focussed on maxent models for density estimation (unsupervised learning).
- We can also formulate such models for classification and regression (conditional density estimation).
- For classification, the simplest model is:

$$p(c|\mathbf{x}) = \frac{\exp \sum_i \theta_{ci} f_i(\mathbf{x})}{\sum_{c'} \exp \sum_i \theta_{c'i} f_i(\mathbf{x})}$$

  where each class gets its own set of weights θ_{ci} over the features and we do the classification using softmax.
- If we use the "identity features" $f_i(\mathbf{x}) = x_i$ then this is exactly equivalent to the *logistic regression* model we saw before.
- The model above is like doing logistic regression on the features. Now features can be very complex, nonlinear functions of the data.

More Complex Models

- We can extend this modeling idea to more complex conditional maxent models, for example the *maximum entropy markov model*.



- The joint distribution is now a conditional model:

$$p(s_1^T | x_1^T) = \prod_t p(s_t | s_{t-1}, f_t(x_1^T))$$

- The features f_t can be very nonlocal functions of the underlying input sequence, for example they can consult things in the past and in the future.

Generative Model Interpretation

- MaxEnt models are a generalized version of expoential family models, and so they can be thought of as generative models which assign probability distribution to joint settings of the features $f_i(\mathbf{x})$.
- But they *are not* generative models of the original inputs $\mathbf{x}$, because the features may be very complicated, nonlinear functions.
- Futhermore, it may be possible to generate joint feature settings which do not correspond to *any* possible input $\mathbf{x}$.
- For example, what if our generative model gives $f_{\texttt{ing}}(c_1, c_2, c_3) = 1$ and $f_{\texttt{?ed}}(c_1, c_2, c_3) = 1$?