Lecture 13:

Parameter learning in undirected models

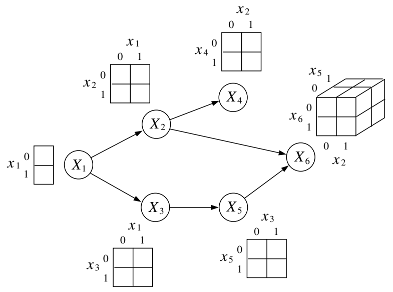
Kevin Murphy

1 November 2004

MLE for general Bayes nets (K+F 13.1–13.2, J 9.1–9.2)

- If we assume the parameters for each CPD are globally independent, then the log-likelihood function decomposes into a sum of local terms, one per node:

$$\log p(\mathcal{D}|\theta) = \log \prod_m \prod_i p(\mathbf{x}_i^m | x_{\pi_i}, \theta_i) = \sum_i \sum_m \log p(\mathbf{x}_i^m | x_{\pi_i}, \theta_i)$$

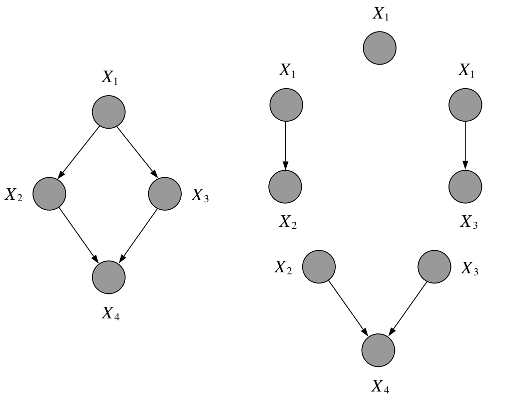


Example: A Directed Model

- Consider the distribution defined by the DAGM:

$$p(\mathbf{x}|\theta) = p(\mathbf{x}_1|\theta_1)p(\mathbf{x}_2|\mathbf{x}_1,\theta_2)p(\mathbf{x}_3|\mathbf{x}_1,\theta_3)p(\mathbf{x}_4|\mathbf{x}_2,\mathbf{x}_3,\theta_4)$$

- This is exactly like learning four separate small DAGMs, each of which consists of a node and its parents.



(a) (b)

Example: Multinomial

- We observe M iid die rolls (K-sided): $\mathcal{D}$=3,1,K,2,. . .
- Model: $p(k) = \theta_k \quad \sum_k \theta_k = 1$
- Likelihood (for binary indicators $[\mathbf{x}^m = k]$):

$$\ell(\theta; \mathcal{D}) = \log p(\mathcal{D}|\theta) = \sum_m \log \prod_k \theta_k^{[\mathbf{x}^m = k]}$$

$$= \sum_m \sum_k [\mathbf{x}^m = k] \log \theta_k = \sum_k N_k \log \theta_k$$

- The counts N_k are the sufficient statistics.
- We need to maximize this subject to the constraint $\sum_k \theta_k = 1$, so we use a Lagrange multiplier.

Lagrange multipliers

- Constrained cost function:

$$\tilde{l} = \sum_k N_k \log \theta_k + \lambda \left(1 - \sum_k \theta_k \right)$$

- Take derivatives wrt θ_k:

$$\begin{aligned} \frac{\partial \tilde{l}}{\partial \theta_k} &= \frac{N_k}{\theta_k} - \lambda = 0 \\ N_k &= \lambda \theta_k \\ \sum_k N_k &= M = \lambda \sum_k \theta_k = \lambda \\ \hat{\theta}_{k,ML} &= \frac{N_k}{M} \end{aligned}$$

- $\hat{\theta}_{k,ML}$ if the fraction of times k occurs.

MLE for Bayes nets with tabular CPDs

- Assume each CPD is represented as a table (multinomial) where

$$\theta_{ijk} \overset{\text{def}}{=} P(X_i = j | X_{\pi_i} = k)$$

- The sufficient statistics are just counts of family configurations

$$N_{ijk} \overset{\text{def}}{=} \sum_m I(X_i^m = j, X_{\pi_i}^m = k)$$

- The log-likelihood is

$$\begin{aligned} \ell &= \log \prod_m \prod_{ijk} \theta_{ijk}^{N_{ijk}} \\ &= \sum_m \sum_{ijk} N_{ijk} \log \theta_{ijk} \end{aligned}$$

- Using a Lagrange multiplier to enforce so $\sum_j \theta_{ijk} = 1$ we get

$$\hat{\theta}_{ijk}^{ML} = \frac{N_{ijk}}{\sum_{j'} N_{ij'k}}$$

MLE for undirected graphical models (J 9.3)

- For directed graphical models, the log-likelihood decomposes into a sum of terms, one per family (node plus parents).
- For undirected graphical models, the log-likelihood does not decompose, because the normalization constant Z is a function of all the parameters (c.f., EM)

$$\mathsf{P}(\mathbf{X}) = \frac{1}{Z} \prod_{\text{cliques } c} \psi_c(\mathbf{x}_c) \qquad Z = \sum_{\mathbf{X}} \prod_{\text{cliques } c} \psi_c(\mathbf{x}_c)$$

- In general, we will need to do inference to learn params for undirected model, even in the fully observed case.

Log Likelihood for undirected model with tabular clique potentials

- In terms of the counts, the log likelihood is given by:

$$\begin{aligned} p(\mathcal{D}|\theta) &= \prod_n \prod_{\mathbf{x}} p(\mathbf{x}|\theta)^{\delta(\mathbf{x}, \mathbf{x}^n)} \\ \log p(\mathcal{D}|\theta) &= \sum_n \sum_{\mathbf{x}} \delta(\mathbf{x}, \mathbf{x}^n) \log p(\mathbf{x}|\theta) \\ \ell &= \sum_{\mathbf{x}} n(\mathbf{x}) \log \left(\frac{1}{Z} \prod_c \psi_c(\mathbf{x}_c) \right) \\ &= \sum_c \sum_{\mathbf{x}_c} n(\mathbf{x}_c) \log \psi_c(\mathbf{x}_c) - N \log Z \end{aligned}$$

- So the clique counts $n(\mathbf{x}_c)$ are the sufficient statistics for our undirected model.
- But now there is a nasty $\log Z$ in the likelihood.

Derivative of log Likelihood

- Log-likelihood: $\ell = \sum_c \sum_{\mathbf{x}_c} n(\mathbf{x}_c) \log \psi_c(\mathbf{x}_c) - N \log Z$
- First term. $\frac{\partial \ell_1}{\partial \psi_c(x_c)} = n(x_c)/\psi_c(x_c)$
- Second term:

$$\begin{aligned}
\frac{\partial \log Z}{\partial \psi_c(x_c)} &= \frac{1}{Z}\frac{\partial}{\partial \psi_c(x_c)} \left(\sum_{\tilde{x}} \prod_d \psi_d(\tilde{x}_d) \right) \\
&= \frac{1}{Z} \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c) \frac{\partial}{\partial \psi_c(x_c)} \left(\prod_d \psi_d(\tilde{x}_d) \right) \\
&= \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c) \frac{1}{\psi_c(\tilde{x}_c)} \frac{1}{Z} \prod_d \psi_d(\tilde{x}_d) \\
&= \frac{1}{\psi_c(\tilde{x}_c)} \sum_{\tilde{x}} \delta(\tilde{x}_c, x_c) p(\tilde{x}) = \frac{p(x_c)}{\psi_c(x_c)}
\end{aligned}$$

Conditions on Clique Marginals

- Derivative of log-likelihood

$$\frac{\partial \ell}{\partial \psi_c(\mathbf{x}_c)} = \frac{n(\mathbf{x}_c)}{\psi_c(\mathbf{x}_c)} - N \frac{p(\mathbf{x}_c)}{\psi_c(\mathbf{x}_c)}$$

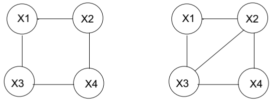
- Hence, for the maximum likelihood parameters, we know that:

$$p^*_{ML}(\mathbf{x}_c) = \frac{n(\mathbf{x}_c)}{N} \overset{\text{def}}{=} q(\mathbf{x}_c) \overset{\text{def}}{=} \tilde{p}(\mathbf{x}_c)$$

  In other words, at the maximum likelihood setting of the parameters, for each clique, *the model marginals must be equal to the observed marginals* (empirical counts).
- This doesn't tell us how to get the ML parameters, it just gives us a condition that must be satisfied when we have them.

MLE for undirected graphical models

- Is the graph *decomposable* (triangulated)?
- Are all the clique potentials defined on maximal cliques (not sub-cliques)? e.g., ψ_{123}, ψ_{234} not $\psi_{12}, \psi_{23}, \ldots$



- Are the clique potentials full tables (or Gaussians), or parameterized more compactly, e.g., $\psi_c(x_c) = exp(\sum_k w_k f_k(x_c))$?

Decomposable?	Max. Cliques	Tabular	Method
Yes	Yes	Yes	Direct
-	-	Yes	IPF
-	-	-	Gradient ascent
-	-	-	Iterative scaling

MLE for decomposable undirected models

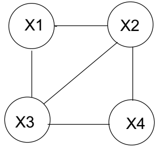
- Consider a chain $X_1 - X_2 - X_3$. The cliques are (X_1, X_2) and (X_2, X_3); the separator is X_2.
- The empirical marginals must equal the model marginals.
- Let us guess that $\hat{p}_{ML}(x_1, x_2, x_3) = \frac{\tilde{p}(x_1,x_2)\tilde{p}(x_2,x_3)}{\tilde{p}(x_2)}$
- We can verify this satisfies the conditions:

$$\hat{p}(x_1, x_2) = \sum_{x_3} \hat{p}(x_1, x_2, x_3) = \tilde{p}(x_1|x_2) \sum_{x_3} \tilde{p}(x_2, x_3) = \tilde{p}(x_1, x_2)$$

  and similarly $\hat{p}(x_2, x_3) = \tilde{p}(x_2, x_3)$.
- To compute the clique potentials, just equate them to the empirical marginals. Also, the separator must be divided into one of its neighbors. Then $Z = 1$.

$$\hat{\psi}^{ML}_{12}(x_1, x_2) = \tilde{p}(x_1, x_2), \quad \hat{\psi}^{ML}_{23}(x_2, x_3) = \frac{\tilde{p}(x_2, x_3)}{\tilde{p}(x_2)} = \tilde{p}(x_3|x_2)$$

MLE for decomposable undirected models



$$\hat{p}(x_{1:4}) = \frac{\tilde{p}(x_1, x_2, x_3)\tilde{p}(x_2, x_3, x_3)}{\tilde{p}(x_2, x_3)}$$

$$\hat{\psi}_{123} = \frac{\tilde{p}(x_1, x_2, x_3)}{\tilde{p}(x_2, x_3)}$$

$$\hat{\psi}_{234} = \tilde{p}(x_2, x_3, x_4)$$

If the potentials were defined on non-maximal cliques (e.g., ψ_{12}, ψ_{34}), we could not equate empirical marginals on max-cliques with model parameters.

Iterative Proportional Fitting (IPF)

- Let's go back to the derivative of the likelihood:

$$\frac{\partial \ell}{\partial \psi_c(\mathbf{x}_c)} = \frac{n(\mathbf{x}_c)}{\psi_c(\mathbf{x}_c)} - N\frac{p(\mathbf{x}_c)}{\psi_c(\mathbf{x}_c)}$$

- From this we can derive another relationship:

$$\frac{q(\mathbf{x}_c)}{\psi_c(\mathbf{x}_c)} = \frac{p(\mathbf{x}_c|\theta)}{\psi_c(\mathbf{x}_c)}$$

  in which ψ_c appears implicitly in the model marginal $p(\mathbf{x}_c|\theta)$.
- To solve for ψ_c is hard, because it appears on both sides of this implicit nonlinear equation.
- The idea of IPF is to hold ψ_c fixed on the right hand side (both in the numberator and denominator) and solve for it on the left hand side. We cycle through all cliques, then iterate:

$$\psi_c^{(t+1)}(\mathbf{x}_c) = \psi_c^{(t)}(\mathbf{x}_c)\frac{q(\mathbf{x}_c)}{p^{(t)}(\mathbf{x}_c)}$$

IPF for pairwise MRFs

```
while not converged
        for each node $i$
            for each neighbor $j \in N_i$
                $m_{ij}^t = P(X_i, X_j|\theta^t)$ (inference)
                $c_{ij}$ = normalize(empirical counts($X_i, X_j$))
                $\psi_{ij}^{t+1} = \psi_{ij}^t \times \frac{c_{ij}}{m_{ij}^t}$
                $\theta^{t+1} = \theta^t \setminus \psi_{ij}^t \cup \psi_{ij}^{t+1}$
```

If the graph is decomposable, we will converge after updating each potential once.

Properties of IPF Updates

- IPF iterates a set of fixed-point equations.
- However, we can prove it is also a coordinate ascent algorithm (coordinates = parameters of clique potentials).
- Hence at each step, it will increase the log-likelihood, and it will converge to a global maximum.

KL Divergence View

- IPF can also be seen to be coordinate ascent in the likelihood using the way of expressing likelihoods using KL divergences.
- First, we observe that maximizing the log likelihood is equivalent to minimizing the KL divergence (cross entropy) from the observed distribution to the model distribution:

$$\max \ell \Leftrightarrow \min KL[q(\mathbf{x}) \| p(\mathbf{x}|\theta)] = \sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\theta)}$$

- Next, we use a property of KL divergence based on the conditional chain rule: $p(\mathbf{x}) = p(\mathbf{x}_a)p(\mathbf{x}_b|\mathbf{x}_a)$:

$$KL[q(\mathbf{x}_a, \mathbf{x}_b) \| p(\mathbf{x}_a, \mathbf{x}_b)] = KL[q(\mathbf{x}_a) \| p(\mathbf{x}_a)] + \sum_{\mathbf{x}_a} q(\mathbf{x}_a) KL[q(\mathbf{x}_b|\mathbf{x}_a) \| p(\mathbf{x}_b|\mathbf{x}_a)]$$

IPF minimizes KL divergence

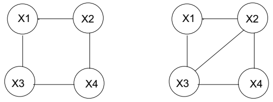
- Putting these two together, we see that:

$$KL[q(\mathbf{x}) \| p(\mathbf{x}|\theta)] = KL[q(\mathbf{x}_c) \| p(\mathbf{x}_c|\theta)] + \sum_{\mathbf{x}_c} q(\mathbf{x}_c) KL[q(\mathbf{x}_{\tilde{c}}|\mathbf{x}_c) \| p(\mathbf{x}_{\tilde{c}}|\mathbf{x}_c, \theta)]$$

  But changing the clique potential has no effect on the conditional distribution, so the second term in unaffected. To minimize the first term, we set the marginal to the observed marginal, just as in IPF.
- In fact, we can interpret IPF updates as retaining the "old" conditional probabilities $p^{(t)}(\mathbf{x}_{\tilde{c}}|\mathbf{x}_c)$ while replacing the "old" marginal probability $p^{(t)}(\mathbf{x}_c)$ with the observed marginal $q(\mathbf{x}_c)$.

MLE for undirected graphical models

- Is the graph *decomposable* (triangulated)?
- Are all the clique potentials defined on maximal cliques (not sub-cliques)? e.g., ψ_{123}, ψ_{234} not $\psi_{12}, \psi_{23}, \ldots$.



- Are the clique potentials full tables (or Gaussians), or parameterized more compactly, e.g., $\psi_c(x_c) = exp(\sum_k w_k f_k(x_c))$?

Decomposable?	Max. Cliques	Tabular	Method
Yes	Yes	Yes	Direct
-	-	Yes	IPF
-	-	-	Gradient ascent
-	-	-	Iterative scaling

Constrained Clique Potentials (J ch 19)

- 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 $c_1 c_2 c_3$ in a string of English text.
- How would we build a model of $p(c_1 c_2 c_3)$?
- 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 $f_{\texttt{ing}}(c_1 c_2 c_3)$ 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.
- Example: a clique potential $\psi(c_1, c_2, c_3)$ could be expressed as

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

- This is still a potential over 26^3 possible settings, but only uses K parameters if there are K features.
- By having one indicator function per combination of $\mathbf{x}_C$, we recover the standard tabular potential.

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:

$$\begin{aligned} p(c_1 c_2 c_3) \propto \exp\{ \quad & \theta_{\texttt{ing}} f_{\texttt{ing}}(c_1 c_2 c_3) + \theta_{\texttt{?ed}} f_{\texttt{?ed}}(c_1 c_2 c_3) + \\ & \theta_{\texttt{qu?}} f_{\texttt{qu?}}(c_1 c_2 c_3) + \theta_{\texttt{zzz}} f_{\texttt{zzz}}(c_1 c_2 c_3) + \ldots\} \end{aligned}$$

- 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 for exponential family

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

i.e., At ML estimate, model expectations match empirical feature counts.
(*) $\frac{\partial \log Z}{\partial \theta_i} = E f_i(X)$ (Jordan eqn 8.40).

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:

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

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

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

$$Z(\theta) \overset{\text{def}}{=} e^{1-\mu} = \sum_{\mathbf{x}} \exp\left\{ \sum_i \theta_i f_i(\mathbf{x}) \right\} \quad \text{since } \sum_x p^*(x) = 1$$

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

- 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.

Conditional MaxEnt Models

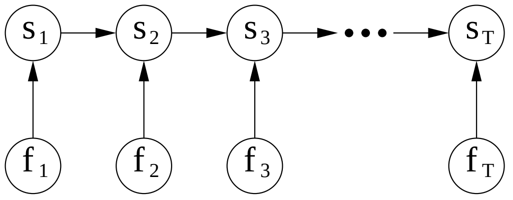
- So far we have focussed on maxent models for density estimation (unsupervised learning).
- We can also formulate such models for classification and regression (conditional density estimation).
- For classification, the simplest model is:

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

  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.

Maximum entropy Markov model (MEMM)

- We can combine local probabilistic classifiers together into a Markov chain, to get a *maximum entropy markov model*.



- A MEMM encodes a *conditional* density:

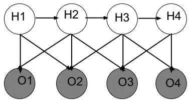
$$p(h_1^T | o_1^T) = \prod_t p(h_t | h_{t-1}, f_t(o_1^T))$$

  whereas an HMM encodes a *joint* density

$$p(h_1^T, o_1^T) = \prod_t p(h_t | h_{t-1}) p(o_t | h_t)$$

Generative vs discriminative models

- An HMM is a model of how to *generate* observations o_t from hidden states h_t.
- Hence an HMM defines a *joint distribution over hidden and observed variables*, $p(o_{1:t}|h_{1:t})p(h_{1:t})$.
- An MEMM is a *conditional model of hidden states given observations*, $p(h_{1:t}|o_{1:t})$.
- Advantages of conditional models:
  - Do not need to waste parameters modeling observed inputs $o_{1:t}$.
  - Can incorporate arbitrary, nonlocal features of the input, without increasing complexity.


H1
H2
H3
H4
O1
O2
O3
O4