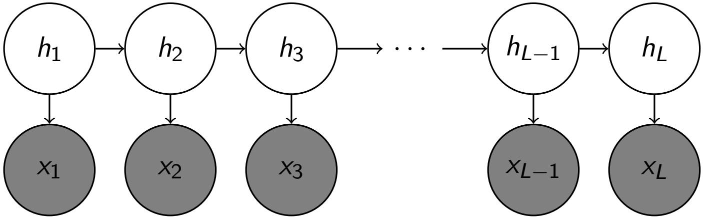
For today

- Inference in HMMs - factor graph view
- Parameter learning in HMMs
- Conditional Random Fields

HMMs specification recap



We specify an HMM by choosing:

- Transition probabilities $p(h_i|h_{i-1})$
- Emission probabilities $p(x_i|h_i)$

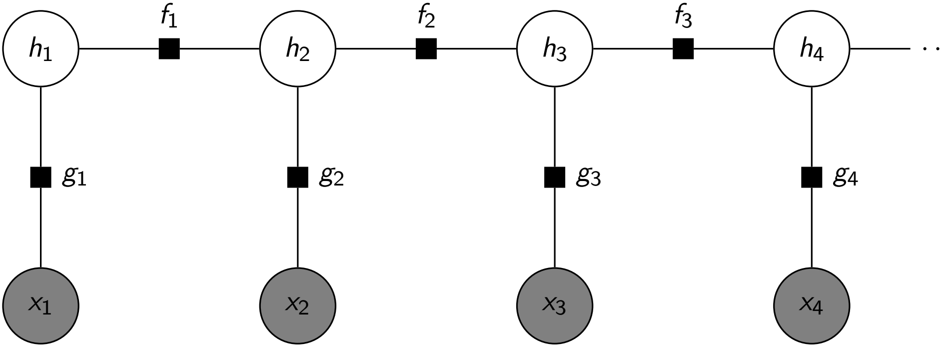
Typical inference tasks in HMMs

Typical tasks:

$p(h_i \mid \mathbf{x})$	Marginal posterior distribution of single latent variable
$p(h_i, h_{i-1} \mid \mathbf{x})$	Marginal posterior distribution of a latent variable pair
$\operatorname{argmax}_{h_i} p(h_i \mid \mathbf{x})$	Most-likely marginal assignment (Posterior decoding)
$\operatorname{argmax}_{\mathbf{h}} p(\mathbf{h} \mid \mathbf{x})$	Most-likely joint assignment (Viterbi decoding)

Posterior decoding and Viterbi decoding are not guaranteed to yield the same solutions.

Factor graph view of HMMs



With potentials being

$$\begin{aligned} f_i(h_i, h_{i+1}) &= p(h_{i+1}|h_i) \\ g(h_i, x_i) &= p(x_i|h_i) \end{aligned}$$

Message passing algorithms

Sum-product updates

$$\mu_{\phi_k \to x_i}(v) = \sum_{x_{C_k}, x_i = v} \phi_k(x_{C_k}) \prod_{j \in C_k, j \neq i} \mu_{x_j \to \phi_k}(x_j)$$

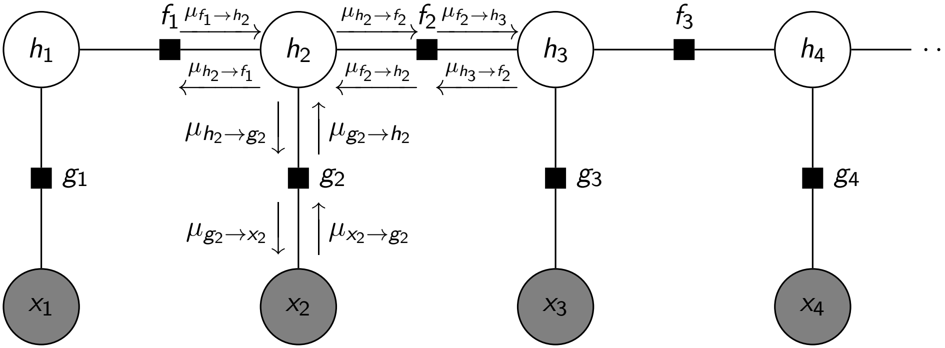
$$\mu_{x_i \to \phi_k}(v) = \prod_{\phi_l \in n(x_i), k \neq l} \mu_{\phi_l \to x_i}(v)$$

Max-product updates

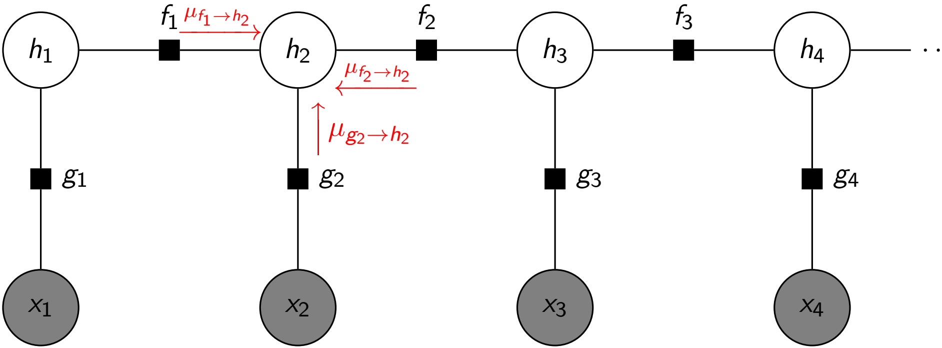
$$\mu_{\phi_k \to x_i}(v) = \max_{x_{C_k}, x_i = v} \phi_k(x_{C_k}) \prod_{j \in C_k, j \neq i} \mu_{x_j \to \phi_k}(x_j)$$

$$\mu_{x_i \to \phi_k}(v) = \prod_{\phi_l \in n(x_i), k \neq l} \mu_{\phi_l \to x_i}(v)$$

Factor graph view of HMMs



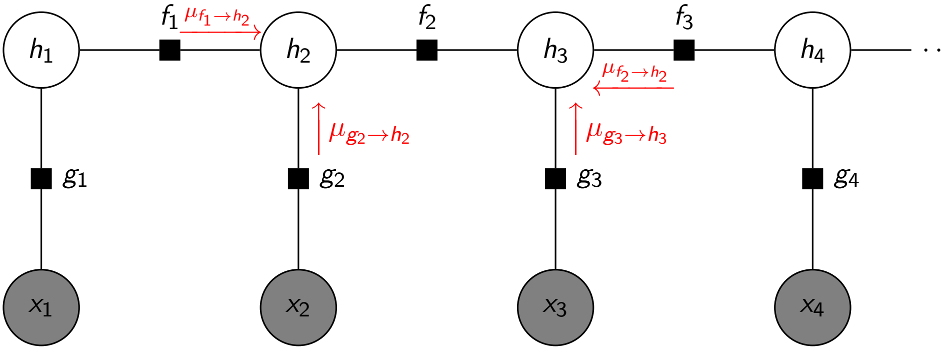
Computing marginals



Univariate marginals are computed by:

$$p(h_i = v) \quad \propto \quad \mu_{f_{i-1} \to h_i}(v)\mu_{f_i \to h_i}(v)\mu_{g_i \to h_i}(v)$$

Computing marginals



Pairwise marginals are computed by:

$$
\begin{aligned}
p(h_i = a, h_{i+1} = b) \quad \propto \quad & \mu_{f_{i-1} \to h_i}(a) \mu_{g_i \to h_i}(b) \times \\
& f_i(h_i = a, h_{i+1} = b) \times \\
& \mu_{f_i \to h_{i+1}}(b) \mu_{g_{i+1} \to h_{i+1}}(b)
\end{aligned}
$$

Learning parameters of HMM

We will use exact EM:

$$\begin{aligned}
E: q^{\text{new}} &= \operatorname*{argmax}_q \sum_t \sum_{\mathbf{h}^t} q(\mathbf{h}^t) \log p(\mathbf{x}^t, \mathbf{h}^t|\theta) - \sum_h q(\mathbf{h}^t) \log q(\mathbf{h}^t) \\
M: \theta^{new} &= \operatorname*{argmax}_\theta \sum_t \sum_{\mathbf{h}^t} q^{\text{new}}(\mathbf{h}^t) \log p(\mathbf{x}^t, \mathbf{h}^t|\theta)
\end{aligned}$$

Which parameters are we learning

$$
\begin{aligned}
p(h_1 = m|\pi) &= \pi_m \\
p(h_i|h_{i-1}, T) &= T(h_i, |h_{i-1}) \\
p(x_i|h_i, \nu) &= g(x_i; \nu_{h_i})
\end{aligned}
$$

and in the case of MoG $\nu_k = (\mu_k, \Sigma_k)$ mean and covariance matrix of the k^{th} class.

So the M-step

$$
M : \theta^{\mathrm{n}ew} = \operatorname*{argmax}_{\theta} \sum_t \sum_{h^t} q^{\mathrm{new}}(h^t) \log p(x^t, h^t|\theta)
$$

operates on $\theta = \{\pi, T, \nu_1, \ldots, \nu_K\}$.

M-step derivation

$$T^{\text{new}} = \operatorname*{argmax}_T \sum_t \sum_{\mathbf{h}^t} q(\mathbf{h}^t) \log \left\{ p(h_1^t) p(x_1^t|h_1^t) \prod_{l=2}^{L} T(h_l^t|h_{l-1}^t) p(x_l^t|h_l^t) \right\}$$

Let us simplify the expression under argmax

$$\operatorname*{argmax}_T \sum_t \sum_{\mathbf{h}^t} q(\mathbf{h}^t) \log \left\{ p(h_1^t) p(x_1^t|h1) \prod_{l=2}^{L} T(h_l^t|h_{l-1}^t) p(x_l^t|h_l^t) \right\} =$$

$$\operatorname*{argmax}_T \sum_t \sum_{\mathbf{h}^t} q(\mathbf{h}^t) \left(\log \left\{ \prod_{l=2}^{L} T(h_l^t|h_{l-1}^t) \right\} + \underbrace{\log \left\{ p(h_1^t) p(x_1^t|h_1^t) \prod_{l=2}^{L} p(x_l^t|h_l^t) \right\}}_{\text{no occurrence of T}} \right)$$

$$\operatorname*{argmax}_T \sum_t \sum_{l=2}^{L} \sum_{\mathbf{h}^t} q(\mathbf{h}^t) \underbrace{\log \left\{ T(h_l^t|h_{l-1}^t) \right\}}_{\text{function of } h_l^t, h_{l-1}^t} =$$

$$\operatorname*{argmax}_T \sum_t \sum_{l=2}^{L} \sum_{h_l^t, h_{l-1}^t} q(h_l^t, h_{l-1}^t) \log \left\{ T(h_l^t|h_{l-1}^t) \right\}$$

M-step derivation

Again as with mixing proportions we are learning categorical distributions

$$\sum_a T(a|b) = 1$$

So we need to solve a constrained optimization problem

$$\begin{aligned}
\underset{T}{\text{maximize}} \; & \sum_t \sum_{l=2}^{L} \sum_{h_l^t, h_{l-1}^t} q(h_l^t, h_{l-1}^t) \log \left\{ T(h_l^t | h_{l-1}^t) \right\} \\
\text{subject to} \; & \sum_a T(a|b) = 1, \forall b
\end{aligned}$$

and the Lagrangian for this problem is

$$\begin{aligned}
L(T, \lambda) &= \sum_t \sum_{l=2}^{L} \sum_{h_l^t, h_{l-1}^t} q(h_l^t, h_{l-1}^t) \log \left\{ T(h_l^t | h_{l-1}^t) \right\} \\
&+ \sum_b \lambda_b \left(\sum_a T(a|b) - 1 \right)
\end{aligned}$$

M-step derivation

The following first order conditions have to hold for an optimum

$$\begin{aligned}
\frac{\partial L(T,\lambda)}{\partial T(a|b)} L(T^*,\lambda^*) &= 0 \\
\frac{\partial L(T,\lambda)}{\partial \lambda_b} L(T^*,\lambda^*) &= 0
\end{aligned}$$

and more explicitly

$$\begin{aligned}
\sum_t \sum_{l=2}^{L} q(h_l^t = a, h_{l-1}^t = b) \frac{1}{T(a|b)} + \lambda_b &= 0 \\
\sum_a T(a|b) - 1 = 0
\end{aligned}$$

this last part you can push through yourselves to get

$$T^{\mathrm{new}}(a|b) = \frac{\sum_t \sum_{l=2}^{L} q(h_l^t = a, h_{l-1}^t = b)}{\sum_t \sum_{l=2}^{L} q(h_{l-1}^t = b)}$$

M-step derivation

Similar gymnastics lead to

$$\pi_m^{\mathrm{new}} = \frac{\sum_t q(h_1^t = m)}{\sum_t \sum_{h_1^t} q(h_1^t)} = \frac{\sum_t q(h_1^t = m)}{N}$$

M-step derivation

Which marginals of $q(\mathbf{h}^t)$ do we need for the update of ν

$$\nu^{\text{new}} = \underset{\nu}{\operatorname{argmax}} \sum_t \sum_{\mathbf{h}^t} q(\mathbf{h}^t) \log \left\{ p(h_1^t) p(x_1^t | h_1^t) \prod_{l=2}^{L} T(h_l^t | h_{l-1}^t) p(x_l^t | h_l^t) \right\}$$

and we can (and you should!) push through simplification of the update to obtain

$$\begin{aligned} \nu^{\text{new}} &= \underset{\nu}{\operatorname{argmax}} \sum_t \sum_l \sum_{h_l^t} q(h_l^t) \log \left\{ p(x_l^t | h_l^t) \right\} \\ &= \underset{\nu}{\operatorname{argmax}} \sum_t \sum_l \sum_{h_l^t} q(h_l^t) \log \left\{ g(x_l^t | \nu_{h_l^t}) \right\} \end{aligned}$$

Equating the derivative of the expression under argmax with respect to ν to zero yields the updates.

M-step derivation

In the case of the gaussian distribution specified by μ_k and Σ_k

$$
\begin{aligned}
\mu_k^{\mathrm{new}} &= \frac{\sum_t \sum_{l=1}^{L} q(h_l^t = k) x_l^t}{\sum_t \sum_{l=1}^{L} q(h_l^t = k)} \\
\Sigma_k^{\mathrm{new}} &= \frac{\sum_t \sum_{l=1}^{L} q(h_l^t = k) x_l^t (x_l^t)'}{\sum_t \sum_{l=1}^{L} q(h_l^t = k)} - \mu_k^{\mathrm{new}} (\mu_k^{\mathrm{new}})'
\end{aligned}
$$

Marginals

Recall that E-step is

$$E: q^{\mathrm{new}} = \underset{q}{\operatorname{argmax}} \sum_t \sum_{\mathbf{h}^t} q(\mathbf{h}^t) \log p(\mathbf{x}^t, \mathbf{h}^t | \theta) - \sum_{\mathbf{h}^t} q(\mathbf{h}^t) \log q(\mathbf{h}^t)$$

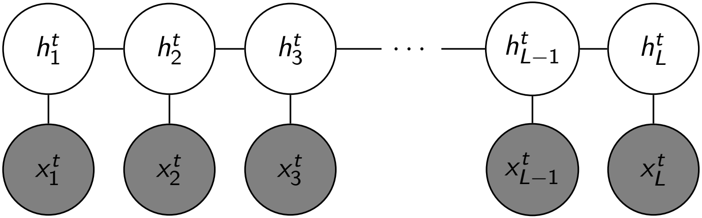
alternatively

$$\underset{q}{\operatorname{argmin}} \, \mathrm{KL}(q(\mathbf{h}^t) || p(\mathbf{x}^t, \mathbf{h}^t | \theta))$$

so

$$q(\mathbf{h}^t) \propto p(\mathbf{x}^t, \mathbf{h}^t | \theta)$$

Clique tree

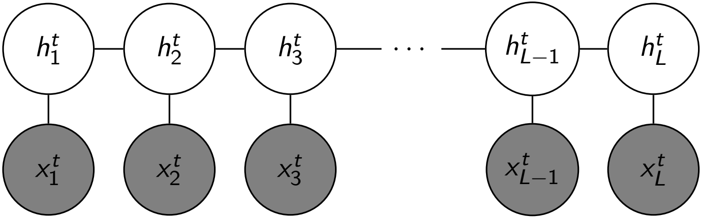


$$q(\mathbf{h}^t) = \frac{1}{Z^t} p(h_1^t) p(x_1^t | h_1) \prod_{l=2}^{L-1} p(h_l^t | h_{l-1}^t) p(x_l^t | h_l^t)$$

Sum product instantiation

$$m_{h_{i-1},h_i}(v) = \sum_{h_{i-1}} p(h_i|h_{i-1})p(x_i|h_i)m_{h_{i-2},h_{i-1}}(h_{i-1})$$

$$m_{h_{i+1},h_i}(v) = \sum_{h_{i+1}} p(h_{i+1}|h_i)p(x_{i+1}|h_{i+1})m_{h_{i+2},h_{i+1}}(h_{i+1})$$



Computing marginals from messages

Once both forward and backward pass are done

$$\begin{aligned} q(h_l = v) &= m_{h_{l-1},h_l}(v) m_{h_{l+1},h_l}(v) \\ q(h_l = v_1, h_{l+1} = v_2) &= m_{h_{l-1},h_l}(v_1) p(h_{l+1}|h_l) p(x_l|h_l) m_{h_{l+2},h_{l+1}}(v_2) \end{aligned}$$

Looking at code ...

Conditional Random Fields

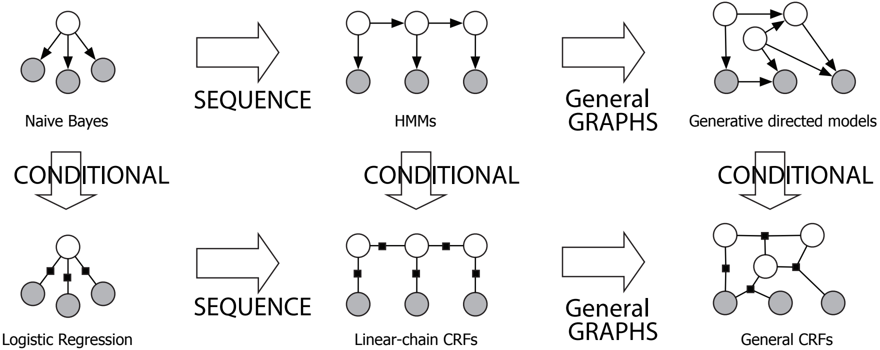
CRFs are the discriminative analog of MRFs.

In particular, a linear CRF is a discriminative analog of HMM.

This is the same relationship that held between Naive Bayes and Logistic Regression.

CRFs can be seen as generalization of Logistic Regression to a structured set of labels.

CRFs and Generative models



Source: Charles Sutton

From HMM to CRF

A joint probability of an HMM (dropping the instance index t for simplicity)

$$p(\mathbf{x}, \mathbf{h}) = \prod_l p(x_l|h_l)p(h_l|h_{l-1})$$

can be rewritten as

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp\left\{ \sum_l \log\{p(h_l|h_{l-1})\} + \log\{p(h_l|x_l)\} \right\}$$

and using indicator functions

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp\left\{ \sum_l \sum_{a,b} \lambda_{ab}[h_l = a][h_{l-1} = b] + \sum_l \sum_a \sum_o \xi_{ao}[h_l = a][x_l = o] \right\}$$

in the case of the parametrization we used earlier $\lambda_{ab} = \log T(a|b)$ and $\xi_{ao} = \log g(o; \nu_a)$

From HMM to CRF

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp \left\{ \sum_{l} \sum_{a,b} \lambda_{ab} [h_l = a][h_{l-1} = b] + \sum_{l} \sum_{a} \sum_{o} \xi_{ao} [h_l = a][x_l = o] \right\}$$

and we can construct features

$$f_{ab}(h^1, h^2, x) = [h^1 = a][h^2 = b]$$
$$f_{ao}(h^1, h^2, x) = [h^1 = a][x = o]$$

Using this set of features we can rewrite the probability as

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \exp \left\{ \sum_{l} \sum_{r} \beta_r f_r(h_l, h_{l-1}, x_l) \right\}$$

Finally we obtain the conditional

$$p(\mathbf{h}|\mathbf{x}) = \frac{p(\mathbf{h}, \mathbf{x})}{p(\mathbf{x})} = \frac{\exp\left\{\sum_l \sum_r \beta_r f_r(h_l, h_{l-1}, x_l)\right\}}{\sum_{\mathbf{h}} \exp\left\{\sum_l \sum_r \beta_r f_r(h_l, h_{l-1}, x_l)\right\}}$$

Linear chain CRF

The distribution of sequential labels $\mathbf{h}$ given sequential data $\mathbf{x}$

$$p(\mathbf{h}|\mathbf{x}) = \frac{p(\mathbf{h},\mathbf{x})}{p(\mathbf{x})} = \frac{\exp\left\{\sum_l \sum_r \beta_r f_r(h_l, h_{l-1}, x_l)\right\}}{\sum_{\mathbf{h}} \exp\left\{\sum_l \sum_r \beta_r f_r(h_l, h_{l-1}, x_l)\right\}}$$

is called linear-chain Conditional Random Field.

In the context of CRFs both labels $\mathbf{h}$ and $\mathbf{x}$ are observed on the training set and the objective is to maximize the (conditional) log likelihood

$$\begin{aligned} \mathrm{LL}(\beta) &= \sum_t \log p(\mathbf{h}^t|\mathbf{x}^t) \\ &= \sum_t \sum_l \sum_r \beta_r f_r(h_l^t, h_{l-1}^t, x_l) \\ &- \sum_t \underbrace{\log\left\{\sum_{\mathbf{h}^t} \exp\left\{\sum_l \sum_r \beta_r f_r(h_l^t, h_{l-1}^t, x_l^t)\right\}\right\}}_{\text{log partition function } \log Z(\beta, \mathbf{x}^t)} \end{aligned}$$

Optimizing the conditional log likelihood for CRF

We maximize $\mathrm{LL}(\beta)$ with respect to β and we can use the same regularization terms as before (ridge/lasso).

The good news is that $\log Z$ is convex in β ($\log \sum \exp$ of a linear combination of βs).

Gradient computation is non-trivial though due to the $\log Z$ term.

$$
\begin{aligned}
\frac{\partial}{\partial \beta_r} \mathrm{LL}(\beta) &= \sum_t \sum_l f_r(h_l^t, h_{l-1}^t, x_l) \\
&- \sum_t \frac{\sum_{\mathbf{h}^t} \exp\left\{\sum_l \sum_r \beta_r f_r(h_l, h_{l-1}, x_l)\right\} \sum_l f_r(h_l, h_{l-1}, x_l)}{\sum_{\mathbf{h}^t} \exp\left\{\sum_l \sum_r \beta_r f_r(h_l^t, h_{l-1}^t, x_l^t)\right\}} \\
&= \underbrace{\sum_t \sum_l f_r(h_l^t, h_{l-1}^t, x_l)}_{\text{feature count in the data}} - \underbrace{\sum_t E_p\left[\sum_l f_r(h_l, h_{l-1}, x_l)\right]}_{\text{expected feature count}}
\end{aligned}
$$

Computing the expectations

We have two types of features

$$\sum_t E_p \left[\sum_l f_{ab}(h_l^t, h_{l-1}^t, x_l^t) \right] = \sum_t \sum_l p(h_l^t, h_{l-1}^t | \mathbf{x}^t, \beta)[h_l = a][h_{l-1}^t = b]$$

$$\sum_t E_p \left[\sum_l f_{ao}(h_l^t, h_{l-1}^t, x_l^t) \right] = \sum_t \sum_l p(h_l^t | \mathbf{x}^t, \beta)[h_l^t = a][x_l^t = o]$$

we can see which marginals we need to compute.

Sum product again

We've done this several times over so I will just write out the potentials

$$\begin{aligned} p(\mathbf{h}|\mathbf{x}) &\propto \exp\left\{\sum_l \sum_r \beta_r f_r(h_l, h_{l-1}, x_l)\right\} \\ &= \prod_l \exp\left\{\sum_{ab} \beta_{ab}[h_l = a, h_{l-1} = b]\right\} \exp\left\{\sum_{ai} \beta_{ao}[h_l = a, x_l = o]\right\} \\ &= \prod_l \phi_l(h_l, h_{l-1}) \psi_l(h_l, x_l) \end{aligned}$$

So computation of the marginals $p(h_l^t, h_{l-1}^t|\mathbf{x}^t, \beta)$ and $p(h_l^t|\mathbf{x}^t, \beta)$ should be a walk in the arboretum[1]

[1]full of new sights and smells but ultimately just as easy as a walk in the park

Optimization options

$$\frac{\partial}{\partial \beta_r} \mathrm{LL}(\beta) = \underbrace{\sum_t \sum_l f_r(h_l^t, h_{l-1}^t, x_l)}_{\text{feature count in the data}} - \underbrace{\sum_t E_p \left[\sum_l f_r(h_l, h_{l-1}, x_l) \right]}_{\text{expected feature count}} - \sum_r \frac{\gamma}{2} \beta_r$$

You can compute the gradients so options are

- gradient descent
- conjugate gradients
- L-BFGS

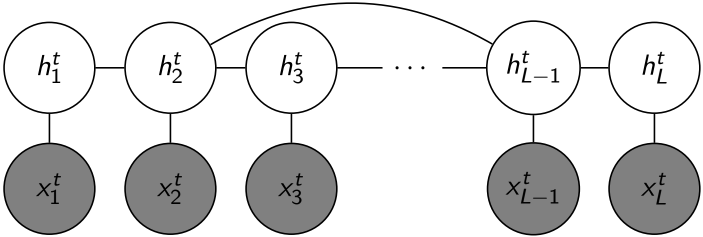
Prediction

Unsurprisingly, you can run max-product on the same distribution

$$
\begin{aligned}
p(\mathbf{h}|\mathbf{x}) &\propto \exp\left\{\sum_l \sum_r \beta_r f_r(h_l, h_{l-1}, x_l)\right\} \\
&= \prod_l \exp\left\{\sum_{ab} \beta_{ab}[h_l = a, h_{l-1} = b]\right\} \exp\left\{\sum_{ai} \beta_{ao}[h_l = a, x_l = o]\right\} \\
&= \prod_l \phi_l(h_l, h_{l-1}) \psi_l(h_l, x_l)
\end{aligned}
$$

to obtain $\mathbf{h}^*$ for a given $\mathbf{x}$, the most likely annotation of the sequence $\mathbf{x}$.

Skip-chain CRF



Features that operate on non-local parts of sequence, e.g.

$$f_r(x_1, x_{100}, h_1, h_{100})$$

Modifying HMM or CRF to include these features can make exact inference intractable.

CRFs are not inherently easier to train than HMMs.

The standard approximation is to run forward-backward without forming the full cliques (loopy belief prop).

Advantages of CRFs

An NLP and CV workhorse, CRFs empirically perform better than HMMs in sequence annotation.

If you are doing sequence annotation or segmentation CRF should be your first choice.

If you are doing unsupervised learning you will have to resort to HMMs.

Either way parameterization and state space structure is the key.

We did ...

- Max-product
- Hidden Markov Models (inference, learning)
- Conditional Random Fields (inference, learning)

We did ...

- Short sequence model
- Conditional Random Fields