Discriminative Sequence Models and Conditional Random Fields

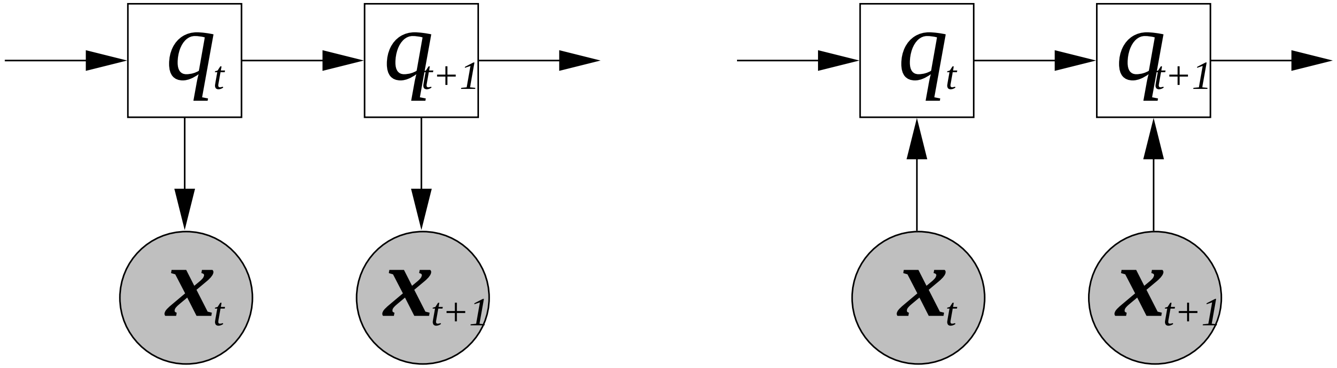
Stephen Clark
(based heavily on slides by Mark Gales)

Lent 2013

Machine Learning for Language Processing: Lecture 4

MPhil in Advanced Computer Science

Discriminative Sequence Models



Generative Model Discriminative Model

- Simple generative model (left) and discriminative model (right)
  - right BN a maximum entropy Markov model

$$P(q_0, \dots, q_T | \boldsymbol{x}_1, \dots, \boldsymbol{x}_T) = \prod_{t=1}^{T} P(q_t | q_{t-1}, \boldsymbol{x}_t)$$

state posterior probability given by (Z_t normalisation term at time t)

$$P(q_t | q_{t-1}, \boldsymbol{x}_t) = \frac{1}{Z_t} \exp\left(\sum_{i=1}^{D} \lambda_i f_i(q_t, q_{t-1}, \boldsymbol{x}_t) \right)$$

Sequence Maximum Entropy Models

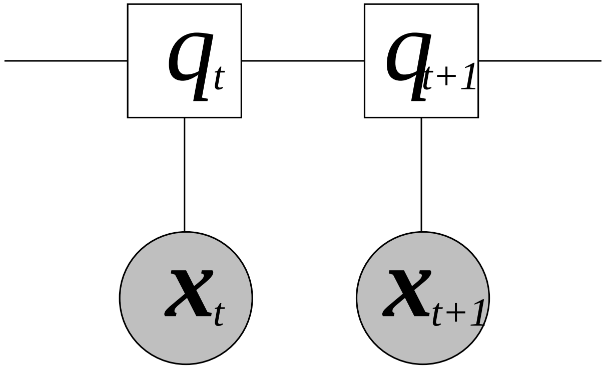
- State posteriors modelled in the Maximum Entropy Markov model
  - could extend to the complete sequence

$$P(q_0, \dots, q_T | \boldsymbol{x}_1, \dots, \boldsymbol{x}_T) = \frac{1}{Z} \exp\left(\sum_{i=1}^{D} \lambda_i f_i(q_0, \dots, q_T, \boldsymbol{x}_1, \dots, \boldsymbol{x}_T) \right)$$

- Problem is that there are a vast number of possible features

What features to extract from the state/observation sequence?

(Simple) Linear Chain Conditional Random Fields



- Extract features based on undirected graph
  - conditional independence assumptions similar to HMM (though undirected)

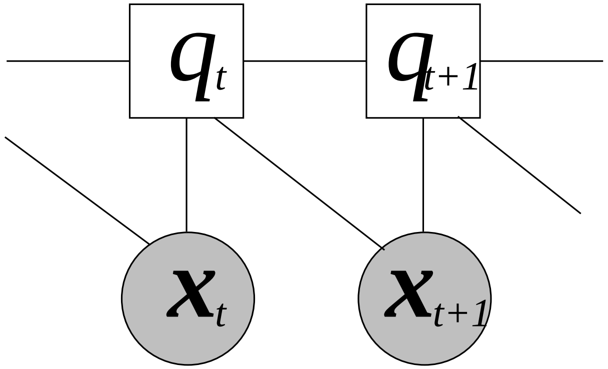
- Posterior model becomes

$$P(q_0, \ldots, q_T | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T) = \frac{1}{Z} \exp\left(\sum_{t=1}^{T}\left(\sum_{i=1}^{D_{\mathsf{t}}} \lambda_i^{\mathsf{t}} f_i(q_t, q_{t-1}) + \sum_{i=1}^{D_{\mathsf{a}}} \lambda_i^{\mathsf{a}} f_i(q_t, \boldsymbol{x}_t)\right)\right)$$

  - D_{t} number of transition style features with parameters $\boldsymbol{\lambda}^{\mathsf{t}}$
  - D_{a} number of word style features with parameters $\boldsymbol{\lambda}^{\mathsf{a}}$

- This has some relationships to HMMs for particular forms of features (though training different)

Linear Chain Conditional Random Fields



- Extract features based on undirected graph
  - conditional independence assumptions extended to previous state

- Posterior model becomes

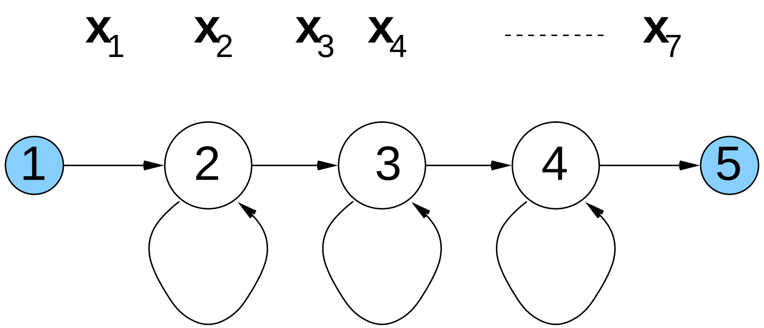
$$P(q_0, \ldots, q_T | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T) = \frac{1}{Z} \exp\left(\sum_{t=1}^{T}\left(\sum_{i=1}^{D} \lambda_i f_i(q_t, q_{t-1}, \boldsymbol{x}_t)\right)\right)$$

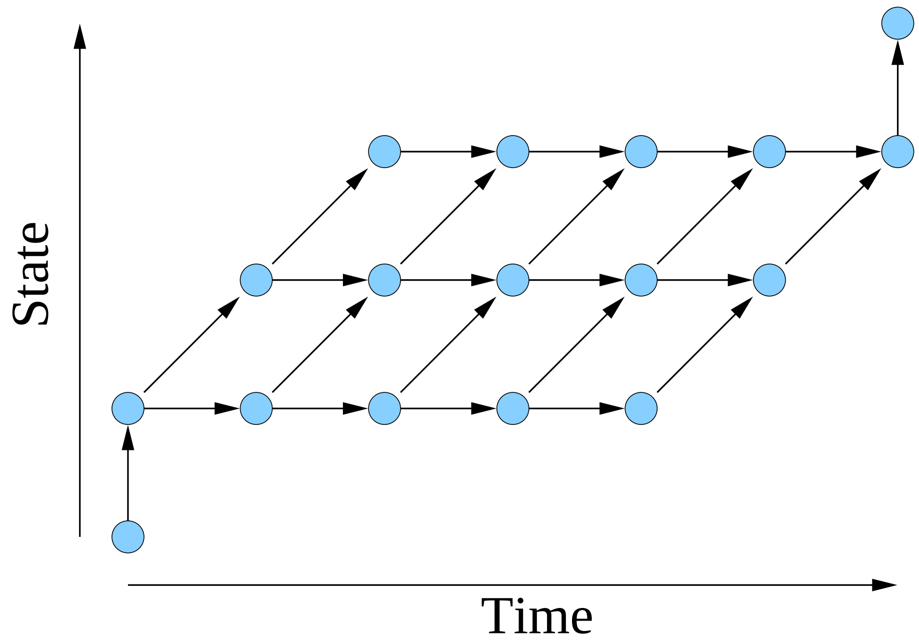
- More interesting than HMM-like features
  - features the same as MaxEnt Markov model
  - BUT normalised globally not locally

Normalisation term

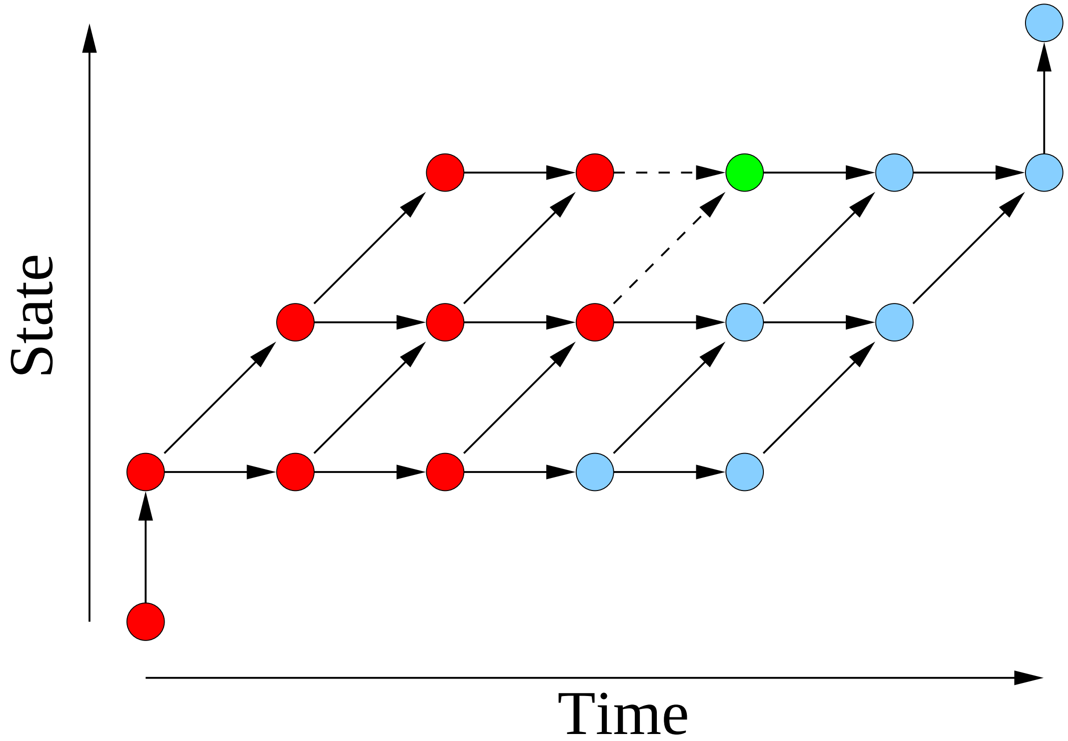
- Need to be able to compute the normalisation term efficiently
  - initially consider the simple linear chain case

$$Z = \sum_{\boldsymbol{q} \in \boldsymbol{Q}_T} \exp\left(\sum_{t=1}^{T}\left(\sum_{i=1}^{D_{\mathtt{t}}} \lambda_i^{\mathtt{t}} f_i(q_t, q_{t-1}) + \sum_{i=1}^{D_{\mathtt{a}}} \lambda_i^{\mathtt{a}} f_i(q_t, \boldsymbol{x}_t)\right)\right)$$





Total Path Cost to a State/Time



- Red possible partial paths
- Green state of interest

$$\text{LAdd}(a,b) = \log\left(\exp(a) + \exp(b)\right)$$
$$\exp(\text{LAdd}(a,b)) = \exp(a) + \exp(b)$$

- Total path cost to state $\mathtt{s}_i$ at time t is $\alpha_i(t)$
  - total path cost to state $\mathtt{s}_4$ at time 5 given by (compare to Viterbi)

$$\alpha_4(5) = \text{LAdd}\left(\alpha_3(4) + \sum_{i=1}^{D_\mathtt{t}} \lambda_i^\mathtt{t} f_i(\mathtt{s}_4, \mathtt{s}_3), \alpha_4(4) + \sum_{i=1}^{D_\mathtt{t}} \lambda_i^\mathtt{t} f_i(\mathtt{s}_4, \mathtt{s}_4)\right) + \sum_{i=1}^{D_\mathtt{a}} \lambda_i^\mathtt{a} f_i(\mathtt{s}_4, \boldsymbol{x}_5)$$

Forward-Backward Algorithm

- α is related to the forward-probability that is used to train HMMs (in the hidden data case)
  - recursion for this form of model can be expressed as

$$\alpha_j(t) = \log\left(\sum_{k=1}^{N}\exp\left(\alpha_k(t-1) + \sum_{i=1}^{D_{\tt t}}\lambda_i^{\tt t} f_i(\mathbf{s}_j, \mathbf{s}_k)\right)\right) + \sum_{i=1}^{D_{\tt a}}\lambda_i^{\tt a} f_i(\mathbf{s}_j, \boldsymbol{x}_t)$$

  - normalisation term can then be expressed as $Z = \exp(\alpha_N(T))$

Forward-Backward Algorithm

- There's also a term related to the backward-probability
  - consider observation at time t given state $\mathtt{s}_j$, $\beta_j(t)$

$$\beta_j(t) = \log\left(\sum_{k=1}^{N} \exp\left(\beta_k(t+1) + \sum_{i=1}^{D_{\mathtt{t}}} \lambda_i^{\mathtt{t}} f_i(\mathtt{s}_k, \mathtt{s}_j) + \sum_{i=1}^{D_{\mathtt{a}}} \lambda_i^{\mathtt{a}} f_i(\mathtt{s}_k, \boldsymbol{x}_{t+1})\right)\right)$$

  - designed so that $Z = \sum_{i=1}^{N} \exp\left(\alpha_i(t) + \beta_i(t)\right)$

Training CRFs

- Training for CRFs is normally fully observed

training observation sequence	$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$
training label sequence	$y_1, \ldots, y_T$

  – where $y_\tau \in \{\omega_1, \ldots, \omega_K\}$

- Need to find the model parameters $\boldsymbol{\lambda}$ so that

$$\begin{aligned}\hat{\boldsymbol{\lambda}} &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ P(y_1, \ldots, y_T | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T, \boldsymbol{\lambda}) \right\} \\ &= \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ \frac{1}{Z} \exp\left(\sum_{i=1}^{D} \lambda_i f_i(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T, y_1, \ldots, y_T) \right) \right\}\end{aligned}$$

Generalised Iterative Scaling for CRFs

- CRF (also MaxEnt model) training is a convex optimisation problem
  - one solution to train parameters is generalised iterative scaling

$$\lambda_i^{[k+1]} = \lambda_i^{[k]} + \frac{1}{C} \log \left(\frac{f_i(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T, y_1, \ldots, y_T)}{\sum_{\boldsymbol{q} \in \boldsymbol{Q}_T} P(\boldsymbol{q}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T, \boldsymbol{\lambda}^{[k]}) f_i(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T, \boldsymbol{q})} \right)$$

  - iterative approach (parameters at iteration k are $\boldsymbol{\lambda}^{[k]}$)

- Numerator is the empirical feature count (as for MaxEnt models)

- Calculation of the feature expectations (denominator) uses forward-backward

Inference with CRFs

- Recognition with CRFs involves finding the most probable label sequence $\hat{\boldsymbol{q}}$

$$\begin{aligned}\hat{\boldsymbol{q}} &= \operatorname*{argmax}_{\boldsymbol{q}\in\boldsymbol{Q}_T} \{P(\boldsymbol{q}|\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T)\} \\ &= \operatorname*{argmax}_{\boldsymbol{q}\in\boldsymbol{Q}_T} \left\{\sum_{i=1}^{D} \lambda_i f_i(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T,\boldsymbol{q})\right\}\end{aligned}$$

  - normalisation term Z not used as it is the same for all label sequences

- The Viterbi algorithm is often used to perform recognition
  - for the simple linear chain CRF relationship to HMM Viterbi clear:

$$\hat{\boldsymbol{q}} = \operatorname*{argmax}_{\boldsymbol{q}\in\boldsymbol{Q}_T} \left\{\sum_{t=1}^{T}\left(\sum_{i=1}^{D_{\mathtt{t}}} \lambda_i^{\mathtt{t}} f_i(q_t,q_{t-1}) + \sum_{i=1}^{D_{\mathtt{a}}} \lambda_i^{\mathtt{a}} f_i(q_t,\boldsymbol{x}_t)\right)\right\}$$