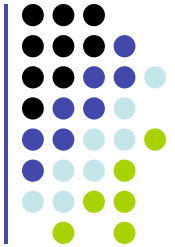
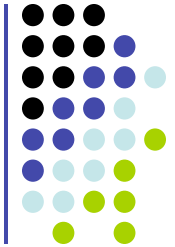


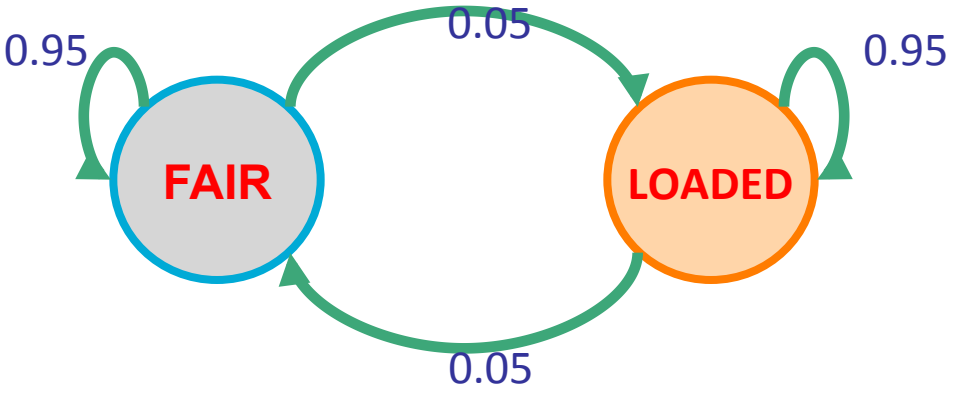
Pair-HMMs and CRFs

Slide Credits: Chuong B. Do

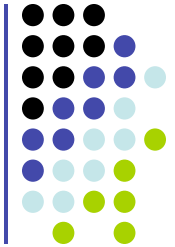
Quick recap of HMMs



- Formally, an HMM = (Σ, Q, A, a_0, e).
  - alphabet: $\Sigma = \{b_1, \ldots, b_M\}$
  - set of states: $Q = \{1, \ldots, K\}$
  - transition probabilities: $A = [a_{ij}]$
  - initial state probabilities: a_{0i}
  - emission probabilities: $e_i(b_k)$
- Example:

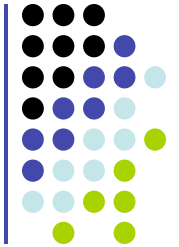


Pair-HMMs

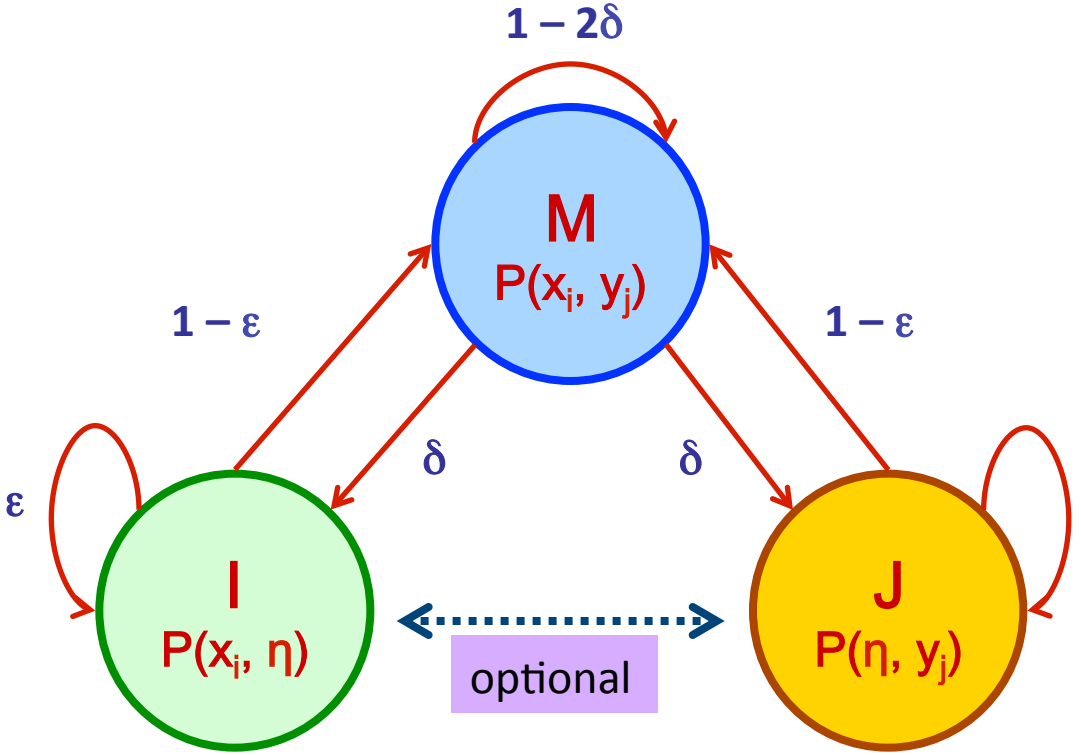


- Consider the HMM = $((\Sigma_1 \cup \{\eta\}) \times (\Sigma_2 \cup \{\eta\}), Q, A, a_0, e)$.
- Instead of emitting a pair of letters, in some states we may emit a letter paired with η (the empty string)
  - For simplicity, assume η is never emitted for both observation sequences simultaneously
  - Call the two observation sequences x and y

Application: sequence alignment



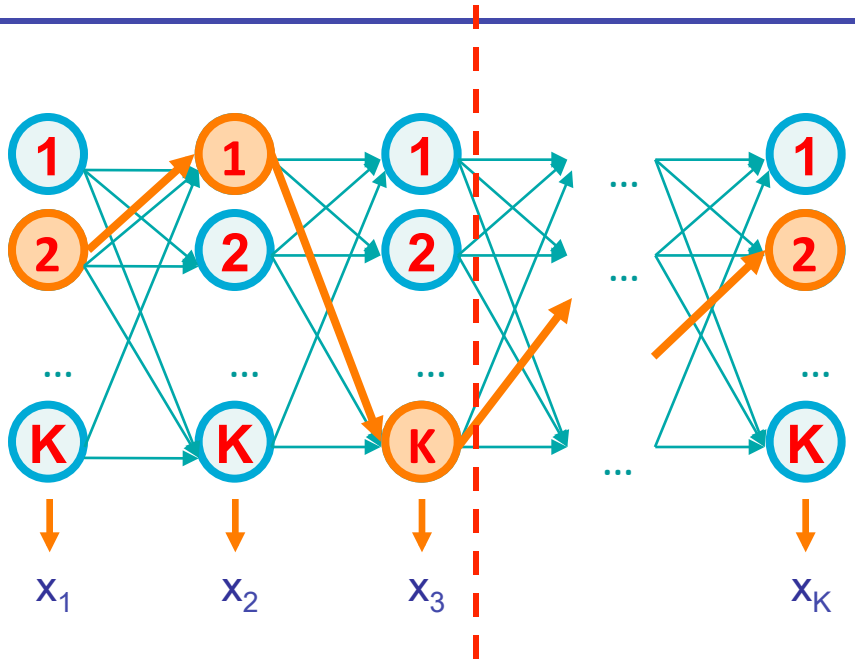
- Consider the following pair-HMM:



$\forall c \in \Sigma, P(\eta, c) = P(c, \eta) = Q(c)$

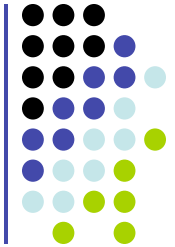
- **QUESTION**: What are the interpretations of P(c,d) and Q(c) for $c,d \in \Sigma$?
- **QUESTION**: What does this model have to do with alignments?
- **QUESTION**: What is the average length of a gapped region in alignments generated by this model? Average length of matched regions?

Recap: Viterbi for single-sequence HMMs



- Algorithm:
  - $V_k(i) = \max_{\pi 1 \ldots \pi i-1} P(x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-1}, x_i, \pi_i = k)$
  - Compute using dynamic programming!

(Broken) Viterbi for pair-HMMs



- In the single sequence case, we defined

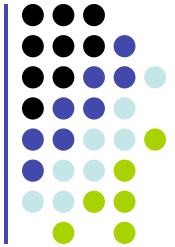
$$V_k(i) = \max_{\pi 1 \ldots \pi i-1} P(x_1 \ldots x_{i-1}, \pi_1 \ldots \pi_{i-1}, x_i, \pi_i = k)$$

$$= e_k(x_i) \cdot \max_j a_{jk} V_j(i - 1)$$

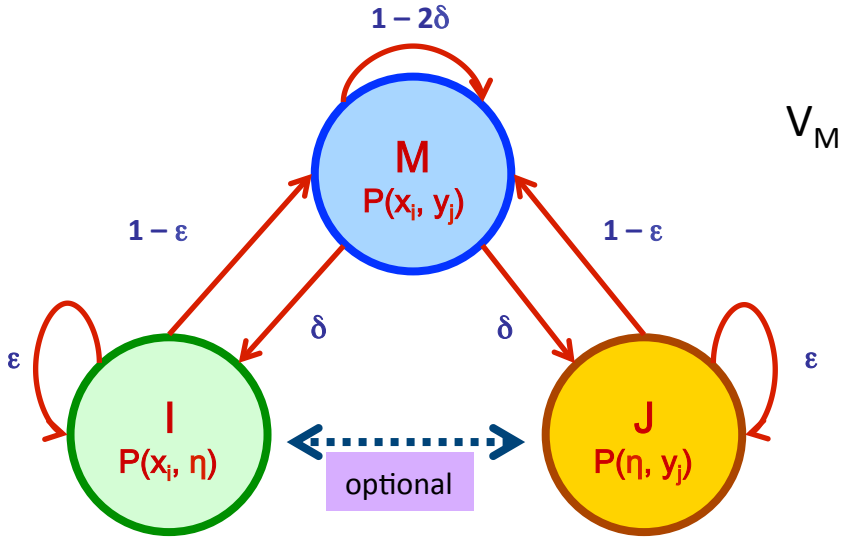
- In the pairwise case,

  $(x_1, y_1) \ldots (x_{i-1}, y_{i-1})$ no longer correspond to the first i – 1 letters of x and y

(Fixed) Viterbi for pair-HMMs



- Consider this special case:



$\forall c \in \Sigma, P(\eta, c) = P(c, \eta) = Q(c)$

$$V_M(i, j) = P(x_i, y_j) \max \begin{cases} (1 - 2\delta)\, V_M(i - 1, j - 1) \\ (1 - \varepsilon)\, V_I(i - 1, j - 1) \\ (1 - \varepsilon)\, V_J(i - 1, j - 1) \end{cases}$$

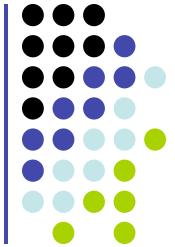
$$V_I(i, j) = Q(x_i) \max \begin{cases} \delta\, V_M(i - 1, j) \\ \varepsilon\, V_I(i - 1, j) \end{cases}$$

$$V_J(i, j) = Q(y_j) \max \begin{cases} \delta\, V_M(i, j - 1) \\ \varepsilon\, V_J(i, j - 1) \end{cases}$$

- Similar for **forward/backward** algorithms
  - (see Durbin et al for details)

QUESTION: What's the computational complexity of DP?

Connection to NW with affine gaps



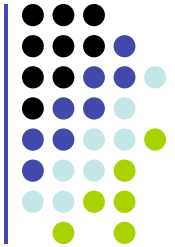
$$V_M(i, j) = P(x_i, y_j) \max \begin{cases} (1 - 2\delta)\, V_M(i - 1, j - 1) \\ (1 - \varepsilon)\, V_I(i - 1, j - 1) \\ (1 - \varepsilon)\, V_J(i - 1, j - 1) \end{cases}$$

$$V_I(i, j) = Q(x_i) \max \begin{cases} \delta\, V_M(i - 1, j) \\ \varepsilon\, V_I(i - 1, j) \end{cases}$$

$$V_J(i, j) = Q(y_j) \max \begin{cases} \delta\, V_M(i, j - 1) \\ \varepsilon\, V_J(i, j - 1) \end{cases}$$

- **QUESTION**: How would the optimal alignment change if we divided the probability for every single alignment by $\prod_{i=1 \ldots |x|} Q(x_i) \prod_{j = 1 \ldots |y|} Q(y_j)$?

Connection to NW with affine gaps



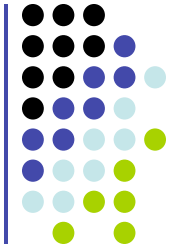
$$V_M(i, j) = \frac{P(x_i, y_j)}{Q(x_i)\,Q(y_j)} \max \begin{cases} (1 - 2\delta)\, V_M(i - 1, j - 1) \\ (1 - \varepsilon)\, V_I(i - 1, j - 1) \\ (1 - \varepsilon)\, V_J(i - 1, j - 1) \end{cases}$$

$$V_I(i, j) = \max \begin{cases} \delta\, V_M(i - 1, j) \\ \varepsilon\, V_I(i - 1, j) \end{cases}$$

$$V_J(i, j) = \max \begin{cases} \delta\, V_M(i, j - 1) \\ \varepsilon\, V_J(i, j - 1) \end{cases}$$

- Account for the extra terms “along the way.”

Connection to NW with affine gaps



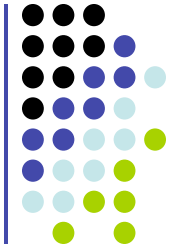
$$\log V_M(i, j) = \log \frac{P(x_i, y_i)}{Q(x_i)\, Q(y_j)} + \max \begin{cases} \log (1 - 2\delta) + \log V_M(i - 1, j - 1) \\ \log (1 - \varepsilon) + \log V_I(i - 1, j - 1) \\ \log (1 - \varepsilon) + \log V_J(i - 1, j - 1) \end{cases}$$

$$\log V_I(i, j) = \max \begin{cases} \log \delta + \log V_M(i - 1, j) \\ \log \varepsilon + \log V_I(i - 1, j) \end{cases}$$

$$\log V_J(i, j) = \max \begin{cases} \log \delta + \log V_M(i, j - 1) \\ \log \varepsilon + \log V_J(i, j - 1) \end{cases}$$

- Take logs, and ignore a couple terms.

Connection to NW with affine gaps

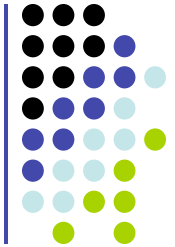


$$M(i, j) = S(x_i, y_j) + \max \begin{cases} M(i-1, j-1) \\ I(i-1, j-1) \\ J(i-1, j-1) \end{cases}$$

$$I(i, j) = \max \begin{cases} d + M(i-1, j) \\ e + I(i-1, j) \end{cases}$$

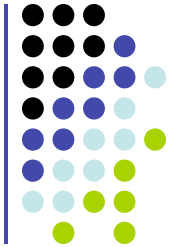
$$J(i, j) = \max \begin{cases} d + M(i, j-1) \\ e + J(i, j-1) \end{cases}$$

- Rename!



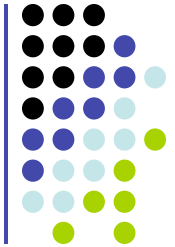
Conditional random fields

Motivation



- HMMs: probabilistic models for labeling sequences
  - For an input sequence x and a parse π, an HMM defines P(π, x)
  - When labeling sequences, what we really care about is P(π | x)
    - In the case of Viterbi, $\arg\max_{\pi} P(\pi \mid x) = \arg\max_{\pi} P(x, \pi)$
- Why do need to parameterize P(x, π)? Can't we model P(π | x) directly?
  - Intuition: $P(x, \pi) = P(x)\, P(\pi \mid x)$ ← why waste effort modeling P(x)?

Conditional random fields



- **Definition**

$$P(\pi \mid x) = \frac{\exp(\sum_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i))}{\sum_{\pi'} \exp(\sum_{i=1 \ldots |x|} w^T F(\pi'_i, \pi'_{i-1}, x, i))}$$

partition coefficient

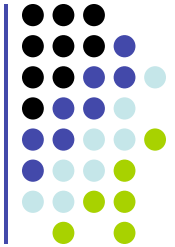
where

F : (state, state, observations, index) $\rightarrow \mathbf{R}^n$ "local feature mapping"

$w \in \mathbf{R}^n$ "parameter vector"

  - Summation over all possible state sequences $\pi'_1 \ldots \pi'_{|x|}$
  - $a^T b$ for vectors $a, b \in \mathbf{R}^n$ denotes inner product, $\sum_{i=1 \ldots n} a_i b_i$

Interpretation of weights



- Define "global feature mapping"

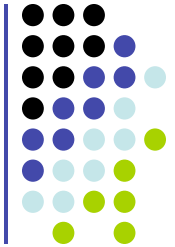
$$F(x, \pi) = \sum_{i=1 \ldots |x|} F(\pi_i, \pi_{i-1}, x, i)$$

Then,

$$P(\pi \mid x) = \frac{\exp(w^T F(x, \pi))}{\sum_{\pi'} \exp(w^T F(x, \pi'))}$$

- **QUESTION**: If all features are nonnegative, what is the effect of increasing some component w_j?

Relationship with HMMs



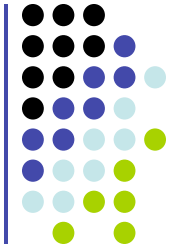
- Recall that in an HMM,

$$\log P(x, \pi) = \log P(\pi_0) + \sum_{i=1 \ldots |x|} [\log P(\pi_i \mid \pi_{i-1}) + \log P(x_i \mid \pi_i)]$$
$$= \log a_0(\pi_0) + \sum_{i=1 \ldots |x|} [\log a(\pi_{i-1}, \pi_i) + \log e_{\pi i}(x_i)]$$

- List the logarithms of all n parameters of our model in a long vector w

$$w = \begin{bmatrix} \log a_0(1) \\ \ldots \\ \log a_0(K) \\ \log a_{11} \\ \ldots \\ \log a_{KK} \\ \log e_1(b_1) \\ \ldots \\ \log e_K(b_M) \end{bmatrix} \in \mathbf{R}^n$$

Relationship with HMMs



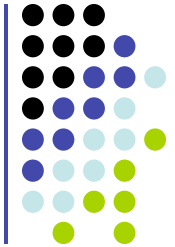
$$\log P(x, \pi) = \log a_0(\pi_0) + \sum_{i=1 \ldots |x|} [\, \log a(\pi_{i-1}, \pi_i) + \log e_{\pi i}(x_i) \,] \qquad (*)$$

- For each component w_j, define F_j to be a 0/1 indicator variable of whether the j^{th} parameter should be included in scoring x, π at position i:

$$w = \begin{bmatrix} \log a_0(1) \\ \ldots \\ \log a_0(K) \\ \log a_{11} \\ \ldots \\ \log a_{KK} \\ \log e_1(b_1) \\ \ldots \\ \log e_K(b_M) \end{bmatrix} \in \mathbf{R}^n \qquad F(\pi_i, \pi_{i-1}, x, i) = \begin{bmatrix} 1\{i = 1 \wedge \pi_{i-1} = 1\} \\ \ldots \\ 1\{i = 1 \wedge \pi_{i-1} = K\} \\ 1\{\pi_{i-1} = 1 \wedge \pi_i = 1\} \\ \ldots \\ 1\{\pi_{i-1} = K \wedge \pi_i = K\} \\ 1\{x_i = b_1 \wedge \pi_i = 1\} \\ \ldots \\ 1\{x_i = b_M \wedge \pi_i = K\} \end{bmatrix} \in \mathbf{R}^n$$

- Then, $\log P(x, \pi) = \sum_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i)$

Relationship with HMMs



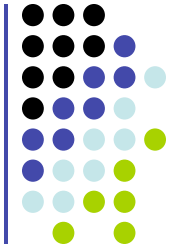
$$\log P(x, \pi) = \sum_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i)$$

- Equivalently,

$$P(\pi \mid x) = \frac{P(x, \pi)}{\sum_{\pi} P(x, \pi)} = \frac{\exp(\sum_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i))}{\sum_{\pi'} \exp(\sum_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i))}$$

- Therefore, an HMM can be converted to an equivalent CRF

CRFs ≥ HMMs

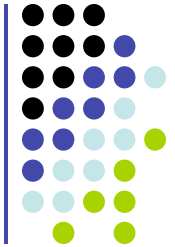


- **A**ll HMMs can be converted to CRFs. But, the reverse does not hold true

- **Reason #1**: HMM parameter vectors obey certain constraints (e.g., $\sum_i a_{0i} = 1$) but CRF parameter vectors (w) are unconstrained

- **Reason #2:** CRFs can contain features that are **NOT** possible to include in HMMs

CRFS ≥ HMMs (continued)



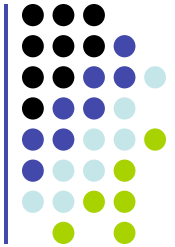
- In an HMM, our features were of the form

$$F(\pi_i, \pi_{i-1}, x, i) = F(\pi_i, \pi_{i-1}, x_i, i)$$

  - I.e., when scoring position i in the sequence, feature only considered the emission x_i at position i.
  - Cannot look at other positions (e.g., x_{i-1}, x_{i+1}) since that would involve "emitting" a character more than once – double-counting of probability

- CRFs don't have this restriction
  - Why? Because CRFs don't attempt to model the observations x!

Examples of non-local features for CRFs



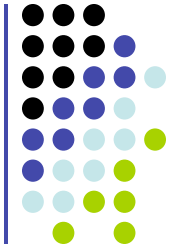
- Casino:
  - Dealer looks at previous 100 positions, and determines whether at least 50 over them had 6's

    $F_j(\text{LOADED, FAIR, x, i}) = 1\{ x_{i-100} \ldots x_i \text{ has} > 50 \text{ 6s} \}$

- CpG islands:
  - Gene occurs near a CpG island

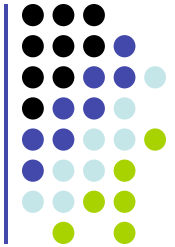
    $F_j(*, \text{EXON, x, i}) = 1\{ x_{i-1000} \ldots x_{i+1000} \text{ has} > 1/16 \text{ CpGs} \}$

Three basic questions for CRFs



- **Evaluation:** Given a sequence of observations x and a sequence of states π, compute $P(\pi \mid x)$
- **Decoding:** Given a sequence of observations x, compute the maximum probability sequence of states $\pi_{ML} = \arg\max_{\pi} P(\pi \mid x)$
- **Learning:** Given a CRF with unspecified parameters w, compute the parameters that maximize the likelihood of π given x, i.e., $w_{ML} = \arg\max_{w} P(\pi \mid x, w)$

Viterbi for CRFs



- Note that:

$$\text{argmax}_\pi P(\pi \mid x) = \text{argmax}_\pi \frac{\exp(\sum_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i))}{\sum_{\pi'} \exp(\sum_{i=1 \ldots |x|} w^T F(\pi'_i, \pi'_{i-1}, x, i))}$$

$$= \arg\max_\pi \exp(\sum_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i))$$

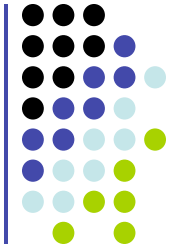
$$= \arg\max_\pi \sum_{i=1 \ldots |x|} w^T F(\pi_i, \pi_{i-1}, x, i)$$

- We can derive the following recurrence:

$$V_k(i) = \max_j [\, w^T F(k, j, x, i) + V_j(i-1) \,]$$

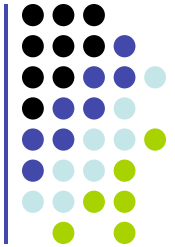
- **Notes:**
  - Even though the features may depend on arbitrary positions in x, x is constant. DP depends only on knowing the previous state
  - Computing the partition function (denominator) can be done by a similar adaptation of the forward/backward algorithms

Three basic questions for CRFs

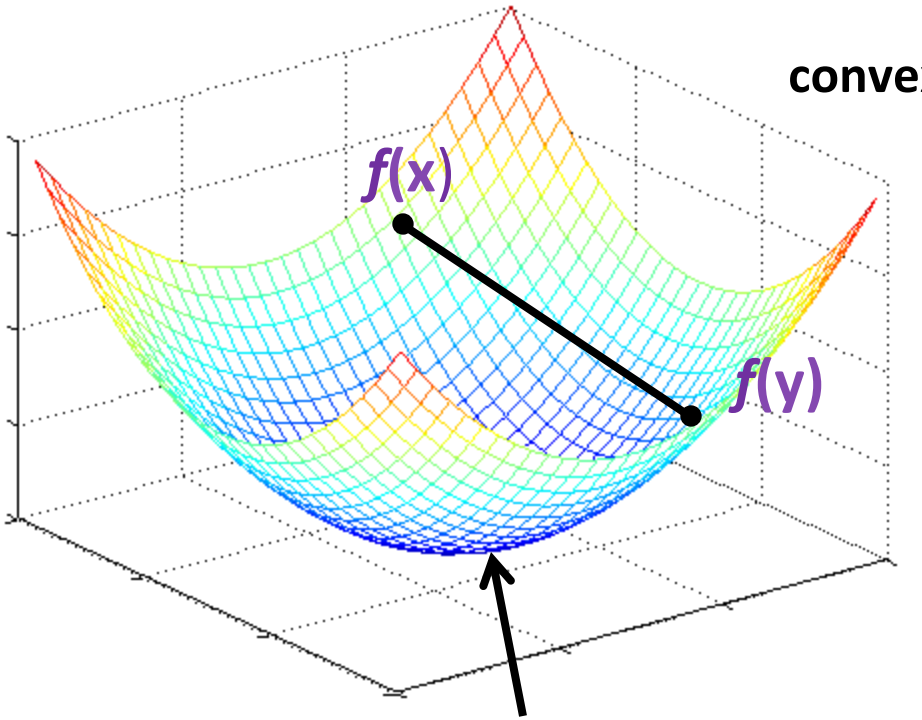


- **Evaluation:** Given a sequence of observations x and a sequence of states π, compute $P(\pi \mid x)$
- **Decoding:** Given a sequence of observations x, compute the maximum probability sequence of states $\pi_{ML} = \arg\max_{\pi} P(\pi \mid x)$
- **Learning:** Given a CRF with unspecified parameters w, compute the parameters that maximize the likelihood of π given x, i.e., $w_{ML} = \arg\max_{w} P(\pi \mid x, w)$

Learning CRFs

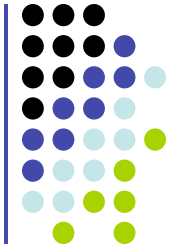


- Key observation: $-\log P(\pi \mid x, w)$ is a differentiable, **convex** function of w



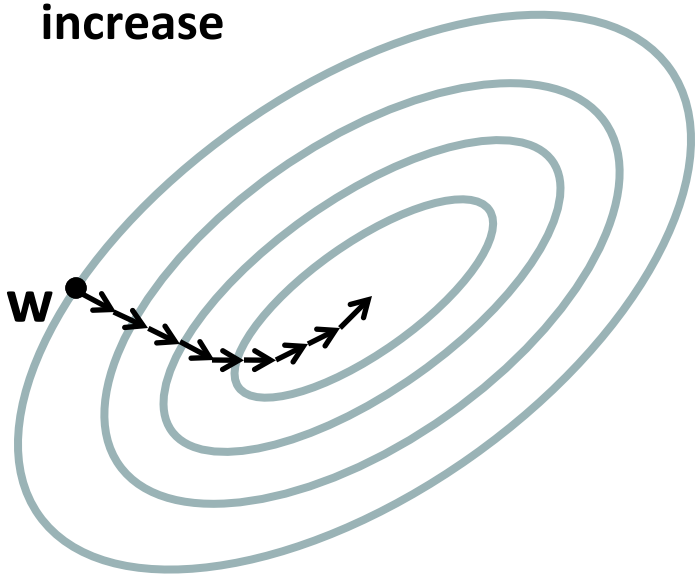
Any local minimum is a global minimum.

Learning CRFs (continued)

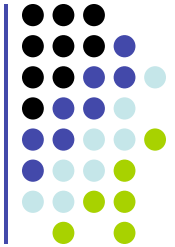


- Compute partial derivative of log $P(\pi \mid x, w)$ with respect to each parameter w_j, and use the gradient ascent learning rule:

Gradient points in the direction of greatest function increase



The CRF gradient



- It turns out that

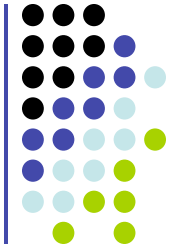
$$(\partial/\partial w_j) \log P(\pi \mid x, w) = F_j(x, \pi) - E_{\pi' \sim P(\pi' \mid x, w)} [\, F_j(x, \pi') \,]$$

correct value for jth feature

expected value for jth feature (given the current parameters)

- This has a very nice interpretation:
  - We increase parameters for which the correct feature values are greater than the predicted feature values
  - We decrease parameters for which the correct feature values are less than the predicted feature values
- This moves probability mass from incorrect parses to correct parses

Summary of CRFs



- We described a probabilistic model known as a CRF
- We showed how to convert any HMM into an equivalent CRF
- We mentioned briefly that inference in CRFs is very similar to inference in HMMs
- We described a gradient ascent approach to training CRFs when true parses are available