Outline

- Recall: Markov Random Fields (MRFs)
- Conditional random fields (CRFs)
  - labeling examples in biology
Recall: Markov Random Fields

- We are interested in modeling the joint distribution over a set of $n$ random variables $Y = (Y_1, \ldots, Y_n)$.
- The variables may be discrete (e.g., specifying interactions), continuous (e.g., expression), or more complex discrete objects (e.g., graphs, alignments).
- We would like to estimate and use a parameterized joint distribution $P(Y = y | \theta)$ over these variables.
- We will specify the joint distribution on the basis of feature or potential functions that, individually, bias likely values for a subset of variables $Y_c = \{Y_i\}_{i \in c}$. 
Recall: Markov Random Fields

- For example, we may introduce a feature that encourages two variables to take the same value

\[ f_{ij}(y_i, y_j) = \delta(y_i, y_j) = \begin{cases} 1, & y_i = y_j \\ 0, & \text{otherwise} \end{cases} \]

- Based on such feature functions, the joint distribution can be written as

\[
P(Y = y|\theta) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{ij} \theta_{ij} f_{ij}(y_i, y_j) \right\} = \frac{1}{Z(\theta)} \exp \left\{ \theta \cdot f(y) \right\}
\]

- There may be multiple (simple) feature functions pertaining to the same or overlapping subset of variables.
Recall: Markov Random Fields

• We have previously used a protein-protein interaction graph to bias cluster assignments. Our feature functions were

\[ f_{ij}(y_i, y_j) = I_{ij} \cdot \delta(y_i, y_j) \]

where \( I_{ij} = 1 \) if \( i \) and \( j \) interact and zero otherwise. Based on these features, we specified a single parameter model

\[ P(Y = y|\eta) = \frac{1}{Z(\eta)} \exp \left\{ \sum_{(i,j)} \eta I_{ij} \delta(y_i, y_j) \right\} \]

(the same parameter is used for all the feature functions)

• Note that it is fine for the feature functions to overlap
Conditional Random Fields

- Feature functions may also depend on other information, giving rise to a conditional model

\[
P(Y = y|x, \theta) = \frac{1}{Z(\theta; x)} \exp \left\{ \theta \cdot f(y; x) \right\}
\]

- The advantage of this formulation is that the dependence on the “input” \( x \) can be arbitrary without complicating the modeling effort (in contrast to HMMs)

- CRFs have become particularly useful across natural language processing, computer vision, as well as computational biology (e.g., fold recognition)
Example: protein structure

- The goal is to map a sequence of amino acid residues \( \{x_1, x_2, \ldots\} \) into corresponding structural labels (\( \alpha \)-helix, turn, \( \beta \)-sheet) \( \{y_1, y_2, \ldots\} \)

\[
\begin{align*}
y &= \text{MLTAEKAAVTGFWGKVKVDEVGA} \\
x &= \text{Example: protein structure}
\end{align*}
\]
Example: protein structure

- The goal is to map a sequence of amino acid residues \( \{x_1, x_2, \ldots \} \) into corresponding structural labels (\( \alpha \)-helix, turn, \( \beta \)-sheet) \( \{y_1, y_2, \ldots \} \)

\[
\begin{align*}
y & = \\
x & = \text{MLTAEKAAVTGFWGKVDEVGA} \ldots
\end{align*}
\]

- The output labels \( \{y_1, y_2, \ldots \} \) depend on each other
Example: protein structure

- The goal is to map a sequence of amino acid residues \( \{x_1, x_2, \ldots \} \) into corresponding structural labels (\( \alpha \)-helix, turn, \( \beta \)-sheet) \( \{y_1, y_2, \ldots \} \)

\[
y = \text{MLTAEKAAVTGFWGKVKNDEVGA...}
\]

- The output labels \( \{y_1, y_2, \ldots \} \) depend on each other
- We don’t need to model the primary sequence \( \{x_1, x_2, \ldots \} \)
Example: protein structure

- The goal is to map a sequence of amino acid residues \( \{x_1, x_2, \ldots \} \) into corresponding structural labels (\( \alpha \)-helix, turn, \( \beta \)-sheet) \( \{y_1, y_2, \ldots \} \)

\[
y = \text{MLTAEKAAVTGFWGKVKVDVGA}\ldots
\]

- The output labels \( \{y_1, y_2, \ldots \} \) depend on each other
- We don’t need to model the primary sequence \( \{x_1, x_2, \ldots \} \)

\[
P(y|x, \theta) = \frac{1}{Z(\theta, x)} \exp \left( \theta \cdot f(y, x) \right)
\]
Example: protein structure

\[ y = \]

\[ x = \text{MLTAEKAAVTGFWGKVDEVGA...} \]
Example: protein structure

\[ y = \]

\[ x = \text{MLTAEKAAVTGFWGKVDEVGA} \ldots \]

\[ y_1 \quad y_2 \quad y_3 \quad y_4 \]

\[ \text{label transition features} \]

\[ f_{kl}(y_i, y_{i+1}) = \begin{cases} 
1, & y_i = k, \ y_{i+1} = l \\
0, & \text{otherwise} 
\end{cases} \]

(do not depend on the position)
Example: protein structure

\[ y = \]

\[ x = \text{MLTAEKAAVTGFWGKVKEVGA} \ldots \]

\[ \theta \cdot f(y, x) = \sum_{kl} \theta_{kl} \sum_{i=1}^{n-1} f_{kl}(y_i, y_{i+1}) + \ldots \]
Example: protein structure

\[ y = \]

\[ x = \text{MLTAEKAAVTGFWGKVDEVG...} \]

**input sequence features**

\[ f_{i,k,j,r}(y_i, x) = \begin{cases} 
1, & y_i = k, x_{i+j} = r \\
0, & \text{otherwise}
\end{cases}, \quad j = -5, \ldots, 5 \]  

**label transition features**

\[ f_{kl}(y_i, y_{i+1}) = \begin{cases} 
1, & y_i = k, y_{i+1} = l \\
0, & \text{otherwise}
\end{cases} \]

(do not depend on the position)

\[ \theta \cdot f(y, x) = \sum_{kl} \theta_{kl} \sum_{i=1}^{n-1} f_{kl}(y_i, y_{i+1}) + \ldots \]
Example: protein structure

\[ y_i = \text{helix} \]

\[ x = \text{MLTAEKAAVTGFWGKVDEVGA} \ldots \]

\[ j = -4 \quad i \]

\[ x_{i+j} = K \]

Input sequence features

\[ f_{i,k,j,r}(y_i, x) = \begin{cases} 
1, & y_i = k, x_{i+j} = r \\
0, & \text{otherwise} 
\end{cases}, \quad j = -5, \ldots, 5 \text{ (e.g.)} \]
Example: protein structure

\[ y = \]

\[ x = \text{MLTAEKAAVTGFHWKVKVDEVGA...} \]

input sequence features

\[ f_{i,k,j,r}(y_i, x) = \begin{cases} 1, & y_i = k, x_{i+j} = r \\ 0, & \text{otherwise} \end{cases}, \quad j = -5, \ldots, 5 \text{ (e.g.)} \]

label transition features

\[ f_{kl}(y_i, y_{i+1}) = \begin{cases} 1, & y_i = k, y_{i+1} = l \\ 0, & \text{otherwise} \end{cases} \]

(do not depend on the position)

\[ \theta \cdot f(y, x) = \sum_{kl} \theta_{kl} \sum_{i=1}^{n-1} f_{kl}(y_i, y_{i+1}) + \sum_{kj} \theta_{kj} \sum_{i=1}^{n} f_{kj}(y_i, x) \]
Example: protein structure

\[ y = \]  

\[ x = \text{MLTAEKAAVTGFWGKVKVDVGA...} \]

- Given a set of primary sequences \( x^t, t = 1, \ldots, m \), and the corresponding secondary structure labelings, \( y^t, t = 1, \ldots, m \), we find parameters \( \theta \) that maximize the conditional log-likelihood

\[
l(D; \theta) = \sum_{t=1}^{m} \log P(y^t|x^t, \theta)\]
Example: protein structure

\[
y = \begin{array}{c}
  y_1 \\
  \circ \\
  y_2 \\
  \circ \\
  y_3 \\
  \circ \\
  y_4 \\
  \circ \\
  \ldots
\end{array}
\]

\[
x = \text{MLTAEKAAVTGFWGKVKVDEVGA}\ldots
\]

- Given a set of primary sequences \(x^t, t = 1, \ldots, m\), and the corresponding secondary structure labelings, \(y^t, t = 1, \ldots, m\), we find parameters \(\theta\) that maximize the conditional log-likelihood

\[
l(D; \theta) = \sum_{t=1}^{m} \log P(y^t|x^t, \theta)
\]

- Sparsity problem!
Dealing with sparsity

\[
\begin{align*}
V \text{ DGDQCE} & \quad \text{SNPCLNGGSCKDD} \quad \text{INSYECWCPFGFEGBKNCES} \\
K \text{ DGDQCE} & \quad \text{GHPCLNQGCHCKDG} \quad \text{IGDYTCTCAEFGFEGBKNCER} \\
N \text{ SYPGCPSSYDGYCLNGGVCNIESLDSYTCNCVIGYSGDRCQTRI} & \\
V\text{ VSHFNDCPDSTQFCFH} & \quad \text{GTCRFLVQEDKPACVCHSGYV GARCE} \\
\end{align*}
\]

(Eddy, 1995)

- residue in position \( i \), i.e., \( x_i \), is replaced by frequency of residues that are aligned with \( x_i \) in the multiple alignment
- as a result, we base label predictions on “local context” rather than individual residue identities
Example

- Gene identification by segmenting DNA sequence into states that represent gene structure (Bernal et al., 2007)

- The output $y$ is a segmentation of the DNA sequence $x$
- The feature functions in $P(y|x, \theta)$ depend on the segment, previous state, as well as the input sequence $x$
- The features score how well each segment matches the corresponding “state” label
Example

- Protein fold recognition (Liu et al., 2006) by modeling how the residue sequence can be segmented into a structural graph describing the fold

- The output $y$ is a segmentation of the input sequence $x$ into states defined by the structural graph (above)

- The feature functions in the conditional distribution $P(y|x, \theta)$ assess how well segments match the states they are assigned to
Example

- We can specify distributions over regulatory networks by scoring individual features present in the networks (cf. Yeang et al., 2004)
- Feature functions bias the presence of interactions, pathways

\[
P(\text{graph}|\text{data}) = \frac{1}{Z} \exp \left\{ \theta \cdot f(\text{graph}, \text{data}) \right\}
\]
Example: disulfide bridges

- Cysteine residues in proteins may form co-valent bonds (disulfide bridges) which are important for structural stability, folding
- We can try to learn to predict which of the possible bonds are actually "active" in a particular protein molecule

(Taskar et al. 2005)
Example: disulfide bridges

- We need to quantify the “potential” for cysteines to form di-sulphide bridges. We will learn these potentials based on sequence features.
Example: disulfide bridges

- We need to quantify the “potential” for cysteines to form di-sulphide bridges. We will learn these potentials based on sequence features.
- Once the “weight” of each match is quantified, the predicted pattern of di-sulphide bonds can be found via maximum weight imperfect matching.
Example: disulfide bridges

$x_i =$ sequence window around the $i^{th}$ cysteine

$y_{ij} = 1$ if cysteine $i$ is matched to cysteine $j$ (zero otherwise)

$f_{ij}(x_i, x_j)$ feature vector for matching $i$ with $j$

$\theta$ parameters for learning how to match
Example: disulfide bridges

\[ x_5 = QNCYP \]
\[ x_6 = EGCSG \]

\[ f_{56}(x_5, x_6) = \]

symmetric comparison
Example: disulfide bridges

\[ x_i = \text{sequence window around the } i^{th} \text{ cysteine} \]
\[ y_{ij} = 1 \text{ if cysteine } i \text{ is matched to cysteine } j \text{ (zero otherwise)} \]
\[ f_{ij}(x_i, x_j) \text{ feature vector for matching } i \text{ with } j \]
\[ \theta \text{ parameters for learning how to match} \]

\[ \theta \cdot f(y, x) = \sum_{ij} y_{ij} (\theta \cdot f_{ij}(x_i, x_j)) \]

"weight" of i-j match
Example: disulfide bridges

- Given $m$ protein sequences $x^t$, $t = 1, \ldots, m$, and the corresponding known “cysteine matches” $y^t$, $t = 1, \ldots, m$, we find parameters $\theta$ that maximize the conditional log-likelihood

$$l(D; \theta) = \sum_{t=1}^{m} \log P(y^t|x^t, \theta)$$

(sparisity problem dealt with in the same way as before)
Recall: learning MRFs

- Suppose (initially) that we have already chosen which features to include in the model.

- We wish to use the available complete observations $D = \{y^t\}_{t=1,...,m}$ to estimate $\theta$.

- The log-likelihood of the data is given by

$$l(D; \theta) = \sum_{t=1}^{m} \log P(Y = y^t|\theta)$$

$$= \sum_{t=1}^{m} [\theta \cdot f(y^t) - \log Z(\theta)]$$

- The parameters are set optimally according to this criterion when empirical expectations of the features agree with those evaluated from the model.
Recall: learning MRFs

- The optimality conditions reduce to matching expectations

\[
\frac{dl(D; \theta)}{d\theta} = m \left[ \frac{1}{m} \sum_{t=1}^{m} f(y^t) - E_{y \sim P_\theta} \{ f(y) \} \right] = 0
\]

where

\[
E_{y \sim P_\theta} \{ f(y) \} = \sum_y P(Y = y | \theta) f(y) = \frac{d}{d\theta} \log Z(\theta)
\]

- The more features are included in the model, the more expectations are “matched” during maximum likelihood estimation
- We can introduce features into the model sequentially
Learning CRFs

- In CRFs, if we have a set of \( m \) input/output pairs \( D_m = \{(x^t, y^t)\}_{t=1,...,m} \), we can set the parameters by maximizing the conditional log-likelihood

\[
l(D; \theta) = \sum_{t=1}^{m} \log P(Y = y^t | x^t, \theta) = \sum_{t=1}^{m} \left[ \theta \cdot f(y^t; x^t) - \log Z(\theta; x^t) \right]
\]

- The “matching” condition for expectations is now expressed in terms of conditional predictions from the model:

\[
\frac{1}{m} \sum_{t=1}^{m} f(y^t; x^t) = \frac{1}{m} \sum_{t=1}^{m} \sum_{y} P(Y = y | x^t, \theta) f(y; x^t)
\]
Learning CRFs... differently

- Instead of maximizing log-likelihood, we could alternatively just require that the correct labeling has the highest probability

$$\log P(y^t|x^t, \theta) \geq \log P(y|x^t, \theta) + 1, \ y \neq y^t$$
Learning CRFs... differently

- Instead of maximizing log-likelihood, we could alternatively just require that the correct labeling has the highest probability

\[
\log P(y^t | x^t, \theta) \geq \log P(y | x^t, \theta) + 1, \quad y \neq y^t
\]

or

\[
\theta \cdot f(y^t, x^t) \geq \theta \cdot f(y, x^t) + 1, \quad y \neq y^t
\]
Learning CRFs... differently

- Instead of maximizing log-likelihood, we could alternatively just require that the correct labeling has the highest probability
  \[
  \log P(y^t|x^t, \theta) \geq \log P(y|x^t, \theta) + 1, \ y \neq y^t
  \]
  or
  \[
  \theta \cdot f(y^t, x^t) \geq \theta \cdot f(y, x^t) + 1, \ y \neq y^t
  \]

- We can find the smallest parameter values (smallest $\|\theta\|^2$) that satisfy these constraints. This is often easier / more feasible in comparison to maximizing the log-likelihood.

  (the training method only works for CRFs, not MRFs)
Example: disulfide bridges

RSCCPCYWGGCPWGQNCYPEGCSGPKV

<table>
<thead>
<tr>
<th>K</th>
<th>SVM</th>
<th>PROFILE</th>
<th>DAG-RNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.63/0.63</td>
<td>0.77/0.77</td>
<td>0.74/0.74</td>
</tr>
<tr>
<td>3</td>
<td>0.51/0.38</td>
<td>0.62/0.52</td>
<td>0.61/0.51</td>
</tr>
<tr>
<td>4</td>
<td>0.34/0.12</td>
<td>0.51/0.36</td>
<td>0.44/0.27</td>
</tr>
<tr>
<td>5</td>
<td>0.31/0.07</td>
<td>0.43/0.13</td>
<td>0.41/0.11</td>
</tr>
</tbody>
</table>

(Taskar et al. 2005)