Deep Graphical Models I

Exponential Softmax

Sufficient Statistics

Belief Propagation
Consider Colorization

$x$ is a black and white image.

$y$ is a color image drawn from $\text{Pop}(y|x)$.

$\hat{y}$ is an arbitrary color image.

$Q_\Phi(\hat{y}|x)$ is the probability that model $\Phi$ assigns to the color image $y$ given black and white image $x$. 
Cross Entropy Training

\[ \Phi^* = \operatorname{argmin}_{\Phi} \mathbb{E}_{(x,y) \sim \text{Pop}} - \log Q_{\Phi}(y|x) \]
An auto-regressive model is locally normalized.

\[
Q_f(\hat{y}) = \prod_i Q_f(\hat{y}[i] \mid \hat{y}[\text{Parents}(i)])
\]

\[
Q_f(\hat{y}[i] \mid \hat{y}[\text{Parents}(i)]) = \text{softmax} f(\tilde{y} | \hat{y}[\text{Parents}(i)])
\]

There are exponentially many possible values for \(\hat{y}\) but each softmax is over a tractable-sized set.
General Markov Random Fields (MRFs) are More Challenging

We can run a CNN with parameters $\Phi$ on the black and white image $x$ to get a Markov random field (MRF) $f_{\Phi}(x)$ on possible color images.

The MRF $f_{\Phi}(x)$ will determine the probabilities $Q_{\Phi}(\hat{y}|x) = Q_{f_{\Phi}(x)}(\hat{y})$. 
Markov Random Fields (MRFs)

$\hat{y}[i]$ is the color value of pixel $i$ in image $\hat{y}$.

$\hat{y}[(i, j)]$ is the pair $(\hat{y}[i], \hat{y}[j])$ for neighboring pixels $i$ and $j$. 
Markov Random Fields (MRFs)

\[ f(\hat{y}) = \sum_{i \in \text{Nodes}} f[i, \hat{y}[i]] + \sum_{(i,j) \in \text{Edges}} f[(i,j), \hat{y}[(i,j)]] \]

Node Potentials  
Edge Potentials
Exponential Softmax

\[ Q_f(\hat{y}) = \text{softmax} \ f(\hat{y}) \]

\[ Q_f(\hat{y}) = \frac{1}{Z} e^{f(\hat{y})} \quad Z = \sum_{\hat{y}} e^{f(\hat{y})} \]

\[ f(\hat{y}) = \sum_{i \in \text{Nodes}} f[i, \hat{y}[i]] + \sum_{(i, j) \in \text{Edges}} f[(i, j), \hat{y}[(i, j)]] \]
Hyper-Graphs: More General and More Concise

A hyper-edge is a subset of nodes.

\[
f(\hat{y}) = \sum_{i \in \text{Nodes}} f[i, \hat{y}[i]] + \sum_{(i,j) \in \text{Edges}} f[(i,j), \hat{y}[(i,j)]]
\]

\[
f(\hat{y}) = \sum_{\alpha \in \text{HyperEdges}} f[\alpha, \hat{y}[\alpha]]
\]
Back-Propagation Through An Exponential Softmax

\[
\Phi^* = \arg\min_{\Phi} \ E(x,y) \sim_{\text{Pop}} - \log Q_{\Phi}(y|x) \\
= \arg\min_{\Phi} \ E(x,y) \sim_{\text{Pop}} - \log Q_{f\Phi}(x)(y)
\]

We need to back-propagate through the softmax to get \( f.\text{grad} \).

\( f \) is a tensor containing the numbers \( f[\alpha, \tilde{y}] \) where \( \tilde{y} \) is a possible value of \( \hat{y}[\alpha] \).

\[
f.\text{grad}[\alpha, \tilde{y}] = \frac{-\partial \log Q_f(y)}{\partial f[\alpha, \tilde{y}]}
\]
Back-Propagation Through An Exponential Softmax

\[ \text{loss}(f, y) = - \ln \left( \frac{1}{Z(f)} e^{f(y)} \right) \]
\[ = \ln Z(f) - f(y) \]

\[ f.\text{grad}[\alpha, \hat{y}] = \left( \frac{1}{Z} \sum_{\hat{y}} e^{f(\hat{y})} \left( \frac{\partial f(\hat{y})}{\partial f[\alpha, \hat{y}]} \right) \right) - \left( \frac{\partial f(y)}{\partial f[\alpha, \hat{y}]} \right) \]
Back-Propagation Through An Exponential Softmax

\[ f.\text{grad}[\alpha, \tilde{y}] = \left( \frac{1}{Z} \sum_{\hat{y}} e^{f(\hat{y})} \frac{\partial f(\hat{y})}{\partial f[\alpha, \tilde{y}]} \right) - \left( \frac{\partial f(y)}{\partial f[\alpha, \tilde{y}]} \right) \]

\[ = \left( \sum_{\hat{y}} Q_f(\hat{y}) \frac{\partial f(\hat{y})}{\partial f[\alpha, \tilde{y}]} \right) - \left( \frac{\partial f(y)}{\partial f[\alpha, \tilde{y}]} \right) \]

\[ = E_{\hat{y} \sim Q_f} 1[\hat{y}[\alpha] = \hat{y}] - 1[y[\alpha] = \tilde{y}] \]

\[ = P_{\hat{y} \sim Q_f} (\hat{y}[\alpha] = \tilde{y}) - 1[y[\alpha] = \tilde{y}] \]
Sufficient Statistics

\[ f.\text{grad}[\alpha, \tilde{y}] = P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y}) - 1[y[\alpha] = \tilde{y}] \]

To compute \( f.\text{grad} \) it suffices to compute \( P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y}) \).

By (minor) abuse of terminology we will call the quantities \( P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y}) \) the **sufficient statistics** for \( f \).

We now focus on computing the sufficient statistics for a given MRF \( f \).
An Aside: Features and Weights

The indicators $1[\hat{y}[\alpha] = \tilde{y}]$ form a 0-1 feature vector $\Psi(\hat{y})$.

The tensor $f[\alpha, \tilde{y}]$ forms a weight vector.

$$f(\hat{y}) = \sum_{\alpha} f[\alpha, \hat{y}[\alpha]]$$

$$= \sum_{\alpha, \tilde{y}} f[\alpha, \tilde{y}] 1[\alpha, \hat{y}[\alpha] = \tilde{y}]$$

$$= f^\top \Psi(\hat{y})$$
An Aside: Features and Weights

The sufficient statistics $P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \hat{y})$ are just the expected value of the features under the distribution defined by the MRF.
Belief Propagation

\[ f \cdot \text{grad}[\alpha, \tilde{y}] = P_{\hat{y} \sim Q_f} (\hat{y}[\alpha] = \tilde{y}) - 1[y[\alpha] = \tilde{y}] \]

For trees we can compute \( P_{\hat{y} \sim Q_f} (\hat{y}[\alpha] = \tilde{y}) \) exactly by message passing, aka, belief propagation.
Message Passing

For each edge \((i, j)\) there is a message \(Z_{i \rightarrow j}\) and a message \(Z_{j \rightarrow i}\).

Each message is assigns a weight to each node value of the target node.

\(Z_{j \rightarrow i}[\tilde{y}]\) is the partition function for the subtree attached to \(i\) through \(j\) and with \(\hat{y}[i]\) restricted to \(\tilde{y}\).
Message Passing

\[
Z(i, \tilde{y}) = \sum_{\hat{y}: \hat{y}[i] = \tilde{y}} e^{f(\hat{y})} = e^{f[i, \tilde{y}]} \left( \prod_{j \in N(i)} Z_{j \rightarrow i}[\tilde{y}] \right)
\]

\[
P_{\hat{y} \sim Q_f}(\hat{y}[i] = \tilde{y}) = \frac{Z(i, \tilde{y})}{Z}, \quad Z = \sum_{\tilde{y}} Z(i, \tilde{y})
\]
Message Passing

\[
Z_{j \rightarrow i}[\tilde{y}] = \sum_{\tilde{y}'} e^{f[j,\tilde{y}']+f[j,i],\{\tilde{y}',\tilde{y}\}} \left( \prod_{k \in N(j), k \neq i} Z_{k \rightarrow j}[\tilde{y}'] \right)
\]
Message Passing

\[ Z(\{i, j\}, \tilde{y}) \doteq \sum_{\hat{y}} e^{f(\hat{y})} \]

\[ = e^{f[i, \tilde{y}[i]] + f[j, \tilde{y}[j]] + f[\{i, j\}, \tilde{y}]} \]

\[ \prod_{k \in N(i), k \neq j} Z_{k \to i}[\tilde{y}[i]] \]

\[ \prod_{k \in N(j), k \neq i} Z_{k \to j}[\tilde{y}[j]] \]

\[ P_{\hat{y} \sim Q_f}(\hat{y}[\{i, j\}] = \tilde{y}) = \frac{Z(\{i, j\}, \tilde{y})}{Z} \]
Loopy BP

Message passing is also called belief propagation (BP).

In a graph with cycles it is common to do **Loopy BP**.

This is done by initializing all message $Z_{i \rightarrow j}[\tilde{y}] = 1$ and then repeating (until convergence) the updates

$$Z_{j \rightarrow i}[\tilde{y}] = \sum_{\tilde{y}'} e^{f[j,\tilde{y}']+f[j,i],\{\tilde{y}',\tilde{y}\}} \left( \prod_{k \in N(j), k \neq i} Z_{k \rightarrow j}[\tilde{y}'] \right)$$
END