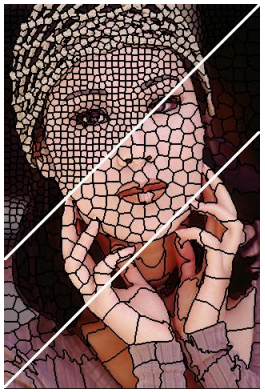
TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2018

Deep Graphical Models

Labeling Pixels

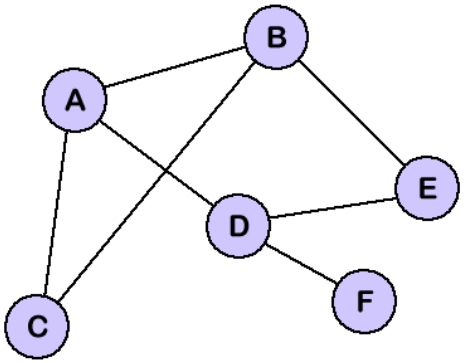


[Achanta et al.]

$$\Phi^* = \operatorname*{argmin}_{\Phi} \; E_{(x,y)\sim \mathrm{Pop}} - \log Q_\Phi(y|x)$$

$$\Phi^* = \operatorname*{argmin}_{\Phi} \; E_{y\sim \mathrm{Pop}} - \log Q_\Phi(y)$$

Deep Graphical Models



We are interested in an assignment $\hat{y}$ of a value to each node.

$$f_\Phi(x)(\hat{y}) = \sum_{i \in \text{Nodes}} f_\Phi(x)[i, \hat{y}[i]] + \sum_{(i,j) \in \text{Edges}} f_\Phi(x)[(i,j), \hat{y}[(i,j)]]$$

Node Potentials Edge Potentials

Notation

$\hat{y}$ denotes an assignment of a value to each node.

y is a node assignment drawn from the population.

$\tilde{y}$ is a value that a single node can have, or a pair of values that a pair of nodes can have.

$\hat{y}[i]$ is the node value that $\hat{y}$ assigns to node i.

$\hat{y}[(i,j)]$ is the pair of values that $\hat{y}$ assigns to i and j.

The Tensor $f_\Phi(x)$

$$f_\Phi(x)(\hat{y}) = \sum_{i \in \text{Nodes}} f_\Phi(x)[i, \hat{y}[i]] + \sum_{(i,j) \in \text{Edges}} f_\Phi(x)[(i,j), \hat{y}[(i,j)]]$$

$f_\Phi(x)$ consists of the tables (tensors) $f_\Phi(x)[i, \tilde{y}]$ and $f_\Phi(x)[(i,j), \tilde{y}]$.

Exponential Softmax

$$Q_{f_\Phi(x)}(\hat{y}) = \underset{\hat{y}}{\text{softmax}}\ f_\Phi(x)(\hat{y})$$

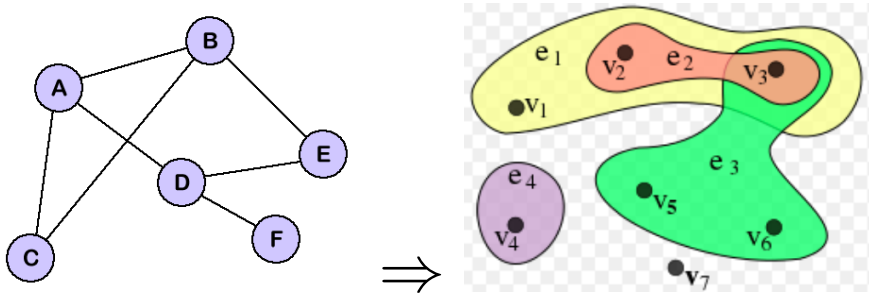
$$Q_f(\hat{y}) = \frac{1}{Z}\, e^{f(\hat{y})} \qquad Z = \sum_{\hat{y}} e^{f(\hat{y})}$$

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

Exponential Softmax

How can we back-propagate through an exponential softmax to get $f_\Phi(x)$.grad?

From Graphs to Hypergraphs: A More Compact Notation



$$f_\Phi(x)(\hat{y}) = \sum_{i \in \text{Nodes}} f_\Phi(x)[i, \hat{y}[i]] + \sum_{(i,j) \in \text{Edges}} f_\Phi(x)[(i,j), \hat{y}[(i,j)]]$$

Node Potentials Edge Potentials

$$f_\Phi(x)(\hat{y}) = \sum_{\alpha \in \text{HyperEdges}} f_\Phi(x)[\alpha, \hat{y}[\alpha]]$$

An expression for $f.\mathrm{grad}[\alpha, \tilde{y}]$

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

$$Z = \sum_{\hat{y}} e^{f(\hat{y})} \qquad \mathrm{loss}(y) = -\ln\left(\frac{1}{Z}\, e^{f(y)}\right)$$

$$\mathrm{loss}(y) = \ln Z - f(y)$$

$$f.\mathrm{grad}[\alpha, \tilde{y}] = \frac{1}{Z}\left(\partial Z / \partial f[\alpha, \tilde{y}]\right) - \partial f(y) / \partial f[\alpha, \tilde{y}]$$

An expression for $f.\mathrm{grad}[\alpha, \tilde{y}]$

$$f(\hat{y}) = \sum_{\alpha} f[\alpha, \hat{y}[\alpha]] \qquad Z = \sum_{\hat{y}} e^{f(\hat{y})}$$

$$\begin{aligned} f.\mathrm{grad}[\alpha, \tilde{y}] &= \frac{1}{Z}\left(\partial Z / \partial f[\alpha, \hat{y}]\right) - \partial f(y)/\partial f[\alpha, \tilde{y}] \\ &= \left(\frac{1}{Z}\sum_{\hat{y}} e^{f(\hat{y})}\left(\partial f(\hat{y})/\partial f[\alpha, \hat{y}]\right)\right) - \partial f(y)/\partial f[\alpha, \tilde{y}] \\ &= \left(E_{\hat{y}\sim Q_f} \mathbb{1}[\hat{y}[\alpha] = \tilde{y}]\right) - \mathbb{1}[y[\alpha] = \tilde{y}] \end{aligned}$$

An expression for $f.\mathrm{grad}$

$$f.\mathrm{grad}[\alpha, \tilde{y}] \;=\; E_{\hat{y} \sim Q_f} \mathbb{1}[\hat{y}[\alpha] = \tilde{y}] - \mathbb{1}[y[\alpha] = \tilde{y}]$$

$$E_{y \sim \mathrm{Pop}}\, f.\mathrm{grad}[\alpha, \tilde{y}] \;=\; P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y}) - P_{y \sim \mathrm{Pop}}(y[\alpha] = \tilde{y})$$

Features and Weights

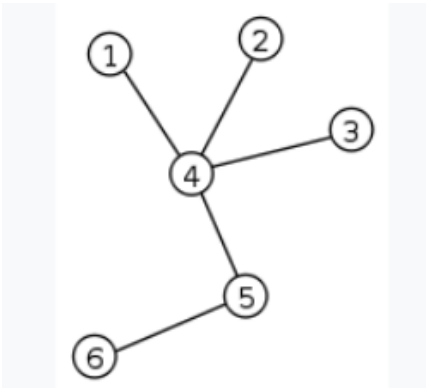
The indicators $\mathbb{1}[\alpha, \hat{y}[\alpha] = \tilde{y}]$ form a feature vector $\Psi(\hat{y})[\alpha, \tilde{y}]$.

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

$$
\begin{aligned}
f(\hat{y}) &= \sum_{\alpha} f[\alpha, \hat{y}[\alpha]] \\
&= \sum_{\alpha,\tilde{y}} f[\alpha, \tilde{y}] \mathbb{1}[\alpha, \hat{y}[\alpha] = \tilde{y}] \\
&= \sum_{\alpha,\tilde{y}} f[\alpha, \tilde{y}] \Psi(\hat{y})[\alpha, \tilde{y}] \\
&= f^{\top} \Psi(\hat{y})
\end{aligned}
$$

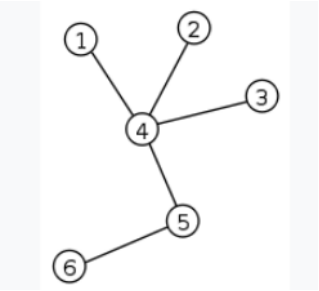
Tree Graphs are Tractable

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



For trees we can compute $P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y})$ by message passing.

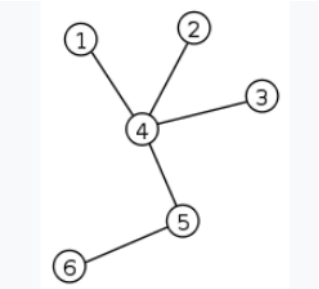
Message Passing



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

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

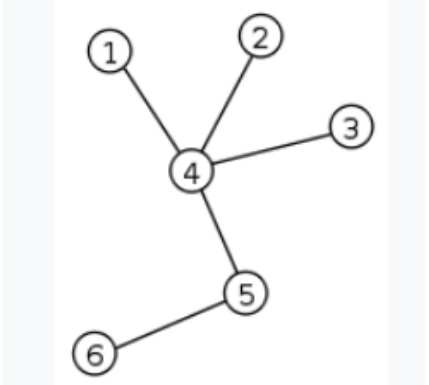
Message Passing



For any node i we will have

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

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



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

$$P_{\hat{y}\sim Q_f}(\hat{y}[i] = \tilde{y}) = \frac{1}{Z} \left(\prod_{j\in N(i)} Z_{j\to i}(\tilde{y}) \right) e^{f[i,\tilde{y}]}$$

Estimation by Sampling

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

We can estimate $P_{\hat{y} \sim Q_f}(\hat{y}[\alpha] = \tilde{y})$ by sampling $\hat{y}$

Monte Carlo Markov Chain (MCMC) Sampling

Metropolis Algorithm

Pick an initial graph label $\hat{y}$ and then repeat:

1. Pick a "neighbor" $\hat{y}'$ of $\hat{y}$ uniformly at random. The neighbor relation must be symmetric. Perhaps Hamming distance one.

2. If $f(\hat{y}') > f(\hat{y})$ update $\hat{y} = \hat{y}'$

3. If $f(\hat{y}') \leq f(\hat{y})$ then update $\hat{y} = \hat{y}'$ with probability $e^{-(f(\hat{y})-f(\hat{y}'))}$

Markov Processes and Stationary Distributions

A Markov process is a process defined by a fixed state transition probability $P(\hat{y}'|\hat{y}) = M_{\hat{y}',\hat{y}}$.

Let P^t the probability distribution for time t.

$$P^{t+1} = MP^t$$

If every state can be reached form every state (ergodic process) then P^t converges to a unique **stationary distribution** P^∞

$$P^\infty = MP^\infty$$

Metropolis Correctness

To verify that the Metropolis process has the correct stationary distribution we simply verify that $MP = P$ where P is the desired distribution.

This can be done by checking that under the desired distribution the flow from $\hat{y}$ to $\hat{y}'$ equals the flow from $\hat{y}'$ to $\hat{y}$ (**detailed balance**).

Metropolis Correctness

For $f(\hat{y}) \geq f(\hat{y}')$

$$\text{flow}(\hat{y}' \to \hat{y}) = \frac{1}{Z} e^{f(\hat{y}')} \frac{1}{N}$$

$$\text{flow}(\hat{y} \to \hat{y}') = \frac{1}{Z} e^{f(\hat{y})} \frac{1}{N} e^{-\Delta f} = \frac{1}{Z} e^{f(\hat{y}')} \frac{1}{N}$$

But detailed balance is not required in general (see Hamiltonian MCMC).

Gibbs Sampling

The Metropolis algorithm wastes time by rejecting proposed moves.

Gibbs sampling avoids this move rejection.

In Gibbs sampling we select a node i at random and change that node by drawing a new node value conditioned on the current values of the other nodes.

Gibbs Sampling

$$Q_f(i = \tilde{y} \mid \hat{y}) \doteq Q_f(\hat{y}[i] = \tilde{y} \mid \hat{y}[1], \ldots, \hat{y}[i-1], \hat{y}[i+1], \ldots, \hat{y}[I])$$

Markov Blanket Property:

$$Q_f(i = \tilde{y} \mid \hat{y}) = Q_f(i = \tilde{y} \mid \hat{y}[N(i)])$$

Gibbs Sampling, Repeat:

- Select i at random
- draw $\tilde{y}$ from $Q_f(i = \tilde{y} \mid \hat{y})$
- $\hat{y}[i] = \tilde{y}$

Gibbs Sampling

Let $\hat{y}[i = \tilde{y}]$ be the assignment $\hat{y}'$ equal to $\hat{y}$ except $\hat{y}'[i] = \tilde{y}$.

$$Q_f(i = \tilde{y} \mid \hat{y}) = \frac{Q_f(\hat{y}[i] = \tilde{y})}{\sum_{\tilde{y}} Q_f(\hat{y}[i] = \tilde{y})}$$

$$= \frac{e^{f(\hat{y}[i=\tilde{y}])}}{\sum_{\tilde{y}} e^{f(\hat{y}[i=\tilde{y}])}}$$

Correctness Proof

$Q_f(\hat{y})$ is a stationary distribution of Gibbs Sampling.

- Select i at random
- draw $\tilde{y}$ from $Q_f(i = \tilde{y} \mid \hat{y})$
- $\hat{y}[i] = \tilde{y}$

The distribution before the update equals the distribution after the update.

Pseudolikelihood

In Pseudolikelihood we replace the objective $-\log Q_f(\hat{y})$ with the objective $-\log \tilde{Q}_f(\hat{y})$ where

$$\tilde{Q}_f(\hat{y}) \doteq \prod_i Q_f(i = \hat{y}[i] \mid \hat{y})$$

$$\text{loss}(f) \doteq -\log \tilde{Q}(y)$$

$$f.\text{grad}[\alpha, \tilde{y}] = \sum_i -\partial \log Q_f[i = \hat{y}[i] \mid \hat{y}]/\partial f[\alpha, \tilde{y}]$$

Pseudolikelihood Consistency

$$\operatorname*{argmin}_{Q} \; E_{y \sim \mathrm{Pop}} \; -\log \tilde{Q}(y) = \mathrm{Pop}$$

Proof of Consistency I

We have

$$\min_{Q} \; E_{y\sim \mathrm{Pop}} \; -\log \tilde{Q}(y) \;\; \leq \;\; E_{y\sim \mathrm{Pop}} \; -\log \widetilde{\mathrm{Pop}}(y)$$

If we can show

$$\min_{Q} \; E_{y\sim \mathrm{Pop}} \; -\log \tilde{Q}(y) \;\; \geq \;\; E_{y\sim \mathrm{Pop}} \; -\log \widetilde{\mathrm{Pop}}(y)$$

Then the minimizer (the argmin) is Pop as desired.

Proof of Consistency II

We will prove the case of two nodes.

$$
\begin{aligned}
& \min_{Q} \; E_{y \sim \mathrm{Pop}} - \log Q(y[1]|y[2]) \; Q(y[2]|y[1]) \\
\geq \; & \min_{Q_1, Q_2} E_{y \sim \mathrm{Pop}} - \log Q_1(y[1]|y[2]) \; Q_2(y[2]|y[1]) \\
= \; & \min_{Q_1} E_{y \sim \mathrm{Pop}} - \log Q_1(y[1]|y[2]) + \min_{Q_2} E_{y \sim \mathrm{Pop}} - \log Q_2(y[2]|y[1]) \\
= \; & E_{y \sim \mathrm{Pop}} - \log \mathrm{Pop}(y[1]|y[2]) + E_{y \sim \mathrm{Pop}} - \log \mathrm{Pop}(y[2]|y[1]) \\
= \; & E_{y \sim \mathrm{Pop}} - \log \widetilde{\mathrm{Pop}}(y|x)
\end{aligned}
$$

Contrastive Divergence

Algorithm (CDk): Run k steps of MCMC for $Q_f(\hat{y})$ **starting from** y to get $\hat{y}$.

Then set

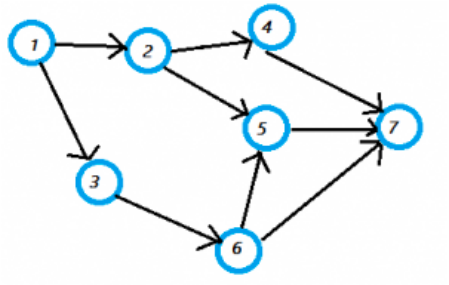
$$f.\text{grad}[\alpha, \tilde{y}] = \mathbb{1}[\hat{y}[\alpha] = \tilde{y}] - \mathbb{1}[y[\alpha] = \tilde{y}]$$

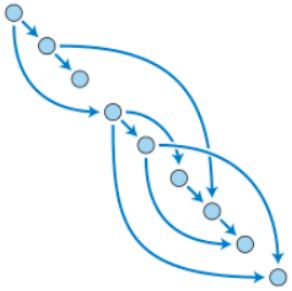
Theorem: If $Q_f(\hat{y}) = \text{Pop}$ then

$$E_{y \sim \text{Pop}}\, \mathbb{1}[\hat{y}[\alpha] = \tilde{y}] - \mathbb{1}[y[\alpha] = \tilde{y}] = 0$$

Here we can take $k = 1$ — no mixing time required.

Directed Graphical Models are Tractable





A directed graphical model is locally normalized.

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

$$Q_f(\hat{y}[\alpha] \mid \hat{y}[\text{Parents}(i)]) = \frac{1}{Z[\alpha]} e^{f[\alpha, \hat{y}[\alpha], \hat{y}[\text{Parents}(i)]]}$$

$$Z[\alpha] = \sum_{\tilde{y}} e^{f[\alpha, \tilde{y}, \hat{y}[\text{Parents}(i)]]}$$

Directed Graphical Models are Tractable

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

$$Q_f(\hat{y}[\alpha] \mid \hat{y}[\text{Parents}(i)]) = \frac{1}{Z[\alpha]} e^{f[\alpha, \hat{y}[\alpha], \hat{y}[\text{Parents}(i)]]}$$

$$Z[\alpha] = \sum_{\tilde{y}} e^{f[\alpha, \tilde{y}, \hat{y}[\text{Parents}(i)]]}$$

$$f.\text{grad}[\alpha, \tilde{y}, \tilde{y}_{\text{Parents}}] = Q_f(i = \tilde{y} \mid \hat{y}) \mathbb{1}[\hat{y}[\text{Parents}] = \tilde{y}_{\text{Parents}}]$$

$$-\mathbb{1}[\hat{y}[\alpha] = \tilde{y},\ \hat{y}[\text{Parents}] = \tilde{y}_{\text{Parents}}]$$

Optimizing Error

Systems are often evaluated by error rather than loss (log loss).

Should we train directly to minimize error?

Should translation be trained directly on BLEU score?

Should segmentation be trained on intersection over union?

Should cancer screening be trained for on recall?

If the model provides probabilities we can do Bayesian inference. But the model might not be sufficiently expressive.

Label Adjustment

We can consider an arbitrary error function $\mathrm{Err}(y, \hat{y})$ assigning an error value the true label is y and the system guesses $\hat{y}$.

$$
\begin{aligned}
f(\hat{y}) &= \sum_{\alpha} f[\alpha, \hat{y}[\alpha]] \\
\hat{y}^*(f) &= \operatorname*{argmax}_{\hat{y}} f(\hat{y}) \\
\breve{y}^*(f, y)) &= \operatorname*{argmax}_{\breve{y}} f(\breve{y}) - \epsilon \mathrm{Err}(y, \breve{y}) \quad \textbf{(adjusted label)} \\
\mathrm{error}(y, f) &= \mathrm{Err}(y, \hat{y}^*(f))
\end{aligned}
$$

$$
f_\Phi(x).\mathrm{grad}[\alpha, \tilde{y}] \texttt{ -= } \eta \left(\mathbb{1}[\hat{y}^*_{f_\Phi(x)}[\alpha] = \tilde{y}] - \mathbb{1}[\breve{y}^*_{f_\Phi(x)}[\alpha] = \tilde{y}] \right)
$$

Label Adjustment Theorem

$$f_\Phi(x).\text{grad}[\alpha, \tilde{y}] \texttt{ -= } \eta \left(\mathbb{1}[\hat{y}^*_{f_\Phi(x)}[\alpha] = \tilde{y}] - \mathbb{1}[\breve{y}^*_{f_\Phi(x)}[\alpha] = \tilde{y}] \right)$$

Theorem: For a continuous and smooth population distribution

$$\nabla_\Phi \; E_{(x,y)\sim\text{Pop}} \; \text{Err}(y, \hat{y}^*(f_\Phi(x)))$$

$$= \lim_{\epsilon \to 0} \; \frac{1}{\epsilon} \; E_{(x,y)\sim\text{Pop}}$$

$$\sum_{\alpha,\tilde{y}} \left(\; \mathbb{1}[\hat{y}^*_{f_\Phi(x)}[\alpha] = \tilde{y}] - \mathbb{1}[\breve{y}^*_{f_\Phi(x)}[\alpha] = \tilde{y}] \right) \nabla_\Phi f_\Phi(x)[\alpha, \tilde{y}]$$

END