Gradient of the training objective

$$
\begin{aligned}
\nabla L(\theta) &= \sum_i \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \frac{\sum_{\mathbf{y}'} \mathbf{f}(\mathbf{y}', \mathbf{x}^i) \exp \theta \cdot \mathbf{f}(\mathbf{x}^i, \mathbf{y}')}{Z_\theta(\mathbf{x}^i)} - 2\theta/C \\
&= \sum_i \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \sum_{\mathbf{y}'} \mathbf{f}(\mathbf{x}^i, \mathbf{y}') \Pr(\mathbf{y}'|\theta, \mathbf{x}^i) - 2\theta/C \\
&= \sum_i \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - E_{\Pr(\mathbf{y}'|\theta, \mathbf{x}^i)} \mathbf{f}(\mathbf{x}^i, \mathbf{y}') - \cancel{2}\theta/C
\end{aligned}
$$

Expected value of features under the current parameters θ.

$\|y'\| = m^n$

$$
\begin{aligned}
E_{\Pr(\mathbf{y}'|\theta, \mathbf{x}^i)} f_k(\mathbf{x}^i, \mathbf{y}') &= \sum_{\mathbf{y}'} f_k(\mathbf{x}^i, \mathbf{y}') \Pr(\mathbf{y}'|\theta, \mathbf{x}^i) \\
&= \sum_{\mathbf{y}'} \sum_c f_k(\mathbf{x}^i, \mathbf{y}'_c, c) \Pr(\mathbf{y}'|\theta, \mathbf{x}^i) \\
&= \sum_c \sum_{\mathbf{y}'_c} f_k(\mathbf{x}^i, \mathbf{y}'_c, c) \underbrace{\Pr(\mathbf{y}'_c|\theta, \mathbf{x}^i)}_{\mu_c(y'_c|x^i)}
\end{aligned}
$$

Computing $E_{\Pr(\mathbf{y}|\theta^t,\mathbf{x}^i)} f_k(\mathbf{x}^i, \mathbf{y})$

Three steps:

1. $\Pr(\mathbf{y}|\theta^t, \mathbf{x}^i)$ is represented as an undirected model where nodes are the different components of $\mathbf{y}$, that is $y_1, \ldots, y_n$.
   The potential $\psi_c(\mathbf{y}_c, \mathbf{x}, \theta)$ on clique c is $\exp(\theta^t \cdot \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i_c, c))$
2. Run a sum-product inference algorithm on above UGM and compute for each $c, \mathbf{y}_c$ marginal probability $\mu(\mathbf{y}_c, c, \mathbf{x}^i)$.
3. Using these μs we compute
   $E_{\Pr(\mathbf{y}|\theta^t,\mathbf{x}^i)} f_k(\mathbf{x}^i, \mathbf{y}) = \sum_c \sum_{\mathbf{y}_c} \mu(\mathbf{y}'_c, c, \mathbf{x}^i) f_k(\mathbf{x}^i, c, \mathbf{y}'_c)$

Example

Consider a parameter learning task for an undirected graphical model on 3 variables $\mathbf{y} = [y_1 \; y_2 \; y_3]$ where each $y_j = +1$ or 0 and they form a chain. Let the following two features be defined for it.

$y_1 - y_2 - y_3$

$K = 2$

$y_i = 0$, $= 1$: $\begin{bmatrix} 0 \\ 0.3 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 2.1 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 0.9 \end{bmatrix}$

$f_1(\mathbf{x}, y_j, j) = x_j y_j$ (where x_j=intensity of pixel j)

$f_2(\mathbf{x}, (y_k, y_j), (k, j)) = [\![y_k \neq y_j]\!]$

where $[\![z]\!] = 1$ if $z =$ true and 0 otherwise.

Initial parameters $\theta = [\theta_1, \theta_2] = [3, -2]$

Examples: $\mathbf{x}^1 = [0.1, 0.7, 0.3], \mathbf{y}^1 = [1, 1, 0]$ label

$D = \{(x^1, y^1), \ldots (x^N, y^N)\}$

$N = 1$

Using these we can calculate:

$F_\theta(y_2, c = \{2\}, x) = \theta_1 \cdot f_1(x, y_2, j) = \theta_1 x_2 y_2$

1. $F_\theta(y_j, c = \{j\}, \mathbf{x})$ = Log Node potentials for $y_j = \theta.\mathbf{f}(\mathbf{x}, y_j, j) = \theta_1 x_j y_j$. For e.g. for y_1 it is $[0, 3 \times 0.1]$.
2. $F_\theta((y_1, y_2), c = (1, 2), \mathbf{x})$ = log edge potentials $= \log \psi_{12}(y_1, y_2)$ $= \theta_2 f_2(\mathbf{x}, (y_1, y_2), (1, 2)) = [0, -2, 0, -2]$

$$\begin{array}{c|cc} & y_2 = 0 & 1 \\ \hline y_1 = 0 & 0 & -2 \\ 1 & -2 & 0 \end{array}$$

$= \log \psi_{23}(y_2, y_3)$

Example (continued)

1. Use above potentials to run sum-product inference on a junction tree to calculate marginals $\mu(y_j, j)$ and $\mu(y_k, y_j, (k, j))$ DOUBT
2. Using these we calculate expected value of features as: $\sum_j \left(\sum_{y_j'} \mu_j(y_j', j)\right) x_j y_j'$

$$E[f_1(\mathbf{x}^1, \mathbf{y})] = \sum_{j=1}^{3} x_j \boxed{\mu_j(1, j)} = 0.1\mu(1, 1) + 0.7\mu(1, 2) + 0.3\mu(1, 3)$$

$j=1 \qquad j=1 \qquad j=2 \qquad j=3$

$\sum_c \sum_{y_c'} \mu_j(y_c', c) f_2(y_c', c, x) \qquad y_c' \in \{(0,0), (0,1), (1,0), (1,1)\}$

$$E[f_2(\mathbf{x}^1, \mathbf{y})] = \underbrace{\mu(1, 0, (1, 2)) + \mu(0, 1, (1, 2))}_{c=(1,2)} + \underbrace{\mu(1, 0, (2, 3)) + \mu(0, 1, (2, 3))}_{c=(2,3)}$$

$f_2 = 0$

3. The value of $\mathbf{f}(\mathbf{x}^1, \mathbf{y}^1)$ for each feature is (Note value of $\mathbf{y}^1 = [1, 1, 0]$):

$= \sum_c f_1(x', y_c', c) = \sum_{j=1}^{3} f_1(x', y_j', j) = \sum_{j=1}^{3} x_j y_j$

$$f_1(\mathbf{x}^1, \mathbf{y}^1) = \underbrace{0.1 * 1}_{j=1} + \underbrace{0.7 * 1}_{j=2} + \underbrace{0.3 * 0}_{j=3} = 0.8$$

$$f_2(\mathbf{x}^1, \mathbf{y}^1) = \underbrace{[\![y_1^1 \neq y_2^1]\!]}_{c=(1,2)} + \underbrace{[\![y_2^1 \neq y_3^1]\!]}_{c=(2,3)} = 1$$

$0 \qquad 1$

4. The gradient of each parameter is then.

$$\nabla L(\theta_1) = 0.8 - E[f_1(\mathbf{x}^1, \mathbf{y})] - 2 * 3/C$$

$-\frac{20}{c}$

$$\nabla L(\theta_2) = 1 - E[f_2(\mathbf{x}^1, \mathbf{y})] + 2 * 2/C$$

$\theta_2 = -2$

Another Example

$p(y_1, y_2, \ldots y_6)$

Consider a parameter learning task for an undirected graphical model on six variables $\mathbf{y} = [y_1\ y_2\ y_3\ y_4\ y_5\ y_6]$ where each $y_j = +1$ or -1.
Let the following eight features be defined for it. $K = 8$

$f_1(y_j, y_{j+1}) = [\![y_j + y_{j+1} > 1]\!], 1 \leq j < 5$ $\quad$ $f_2(y_1, y_3) = -2y_1y_3$

$f_3(y_2, y_3) = y_2y_3$ $\quad$ $f_4(y_3, y_4) = y_3y_4$

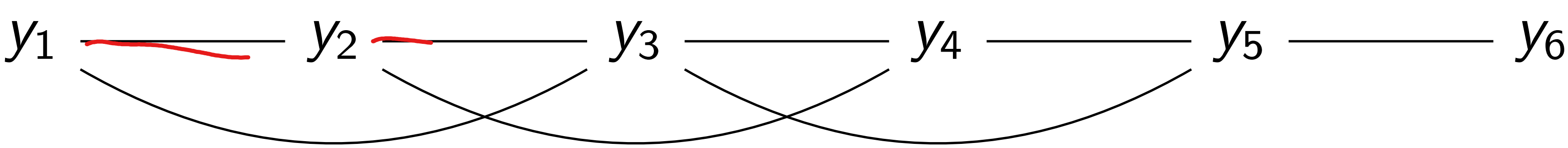
$f_5(y_2, y_4) = [\![y_2y_4 < 0]\!]$ $\quad$ $f_6(y_4, y_5) = 2y_4y_5$

$f_7(y_3, y_5) = -y_3y_5$ $\quad$ $f_8(y_5, y_6) = [\![y_5 + y_6 > 0]\!]$.

where $[\![z]\!] = 1$ if $z =$ true and 0 otherwise. That is, $\mathbf{f}(\mathbf{y}) = [f_1\ f_2\ f_3\ f_4\ f_5\ f_6\ f_7\ f_8]^T$. Assume the corresponding weight vector to be $\theta = [1\ 1\ 1\ 2\ 2\ 1\ \ -1\ 1]^T$

Example

Draw the underlying graphical model corresponding to the 6 variables.



Draw an arc between any two y which appear together in any of the 8 features.

Example

Draw the junction tree corresponding to the graph above and assign potentials to each node of your junction tree so that you can run message passing on it to find $Z = \sum_{\mathbf{y}} \exp(\theta^T \mathbf{f}(\mathbf{x}, \mathbf{y}))$ that is, define $\psi_c(\mathbf{y}_c)$ in terms of the above quantities for each clique node c in the JT.

For clique c, $\psi_c(\mathbf{y}_c) = \exp(\theta_c \cdot \mathbf{f}_c(\mathbf{x}, \mathbf{y}_c))$. *log* of the potentials are shown below

y_1, y_2, y_3	y_2, y_3, y_4	y_3, y_4, y_5	y_5, y_6
$1.f_1(y_1, y_2) + 1.f_1(y_2, y_3) + 1.f_2(y_1, y_3) + 1.f_3(y_2, y_3)$	$2.f_5(y_2, y_4) + 1.f_1(y_3, y_4) + 2.f_4(y_3, y_4)$	$-1.f_7(y_3, y_5) + 1.f_1(y_4, y_5) + 1.f_6(y_4, y_5)$	$1.f_1(y_5, y_6) + 1.f_8(y_5, y_6)$

DOUBT

Example

Suppose you use the junction tree above to compute the marginal probability for each pair of adjacent variables in the graph of part (a). Let $\mu_{ij}(-1,1), \mu_{ij}(1,1), \mu_{ij}(-1,-1), \mu_{ij}(1,-1)$ denote the marginal probability of variable pairs y_i, y_j taking values (-1,1), (1,1), (-1,-1) and (1,-1) respectively. Express the expected value of the following features in terms of the μ values.

1. DOUBT

$$f_1 = \sum_j \big(f_1(-1,-1)\mu_{j,j+1}(-1,-1) + f_1(-1,1)\mu_{j,j+1}(-1,1) + f_1(1,-1)\mu_{j,j+1}(1,-1) + f_1(1,1)\mu_{j,j+1}(1,1)\big)$$

2. $f_2 = 2\big(-\mu_{1,3}(-1,-1) + \mu_{1,3}(-1,1) + \mu_{1,3}(1,-1) - \mu_{1,3}(1,1)\big)$
3. $f_8 = \mu_{56}(1,1)$

Training algorithm

Input: $D = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{\ell=1}^N$, $\mathbf{f} : f_1 \dots f_K$

Output: $\theta = \text{argmax} \sum_{\ell=1}^N (\theta \cdot \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \log Z_\theta(\mathbf{x}^i)) - ||\theta||^2/C$

Initialize $\theta^0 = \mathbf{0}$ random values.

for $t = 1 \dots T$ **do** Training iteration.

  for $i = 1 \dots N$ **do**

    $g_{k,i} = f_k(\mathbf{x}^i, \mathbf{y}^i) - E_{\Pr(\mathbf{y}'|\theta^t, \mathbf{x}^i)} f_k(\mathbf{x}^i, \mathbf{y}') \quad k = 1 \dots K$

  end for

expensive sum-product inference in graphical

  $g_k = \sum_i g_{k,i} \quad k = 1 \dots K$

  $\theta_k^t = \theta_k^{t-1} + \gamma_t (g_k - 2\theta_k^{t-1}/C)$

  Exit if $||\mathbf{g}|| \approx zero$

end for

what is m?

Running time of the algorithm is $O(INn(m^2 + K))$ where I is the total number of iterations.

chain graphical model.

m^{w+1}

Local conditional probability for BN

$$\Pr(y_1, \ldots, y_n | \mathbf{x}, \theta) = \prod_j \Pr(y_j | \mathbf{y}_{\mathrm{Pa}(j)}, \mathbf{x}, \theta)$$

$c = (j, pa(j))$

$$= \prod_j \frac{\exp(F_\theta(\mathbf{y}_{\mathrm{Pa}(j)}, y_j, j, \mathbf{x}))}{\sum_{y'_j=1}^{m} \exp(F_\theta(\mathbf{y}_{\mathrm{Pa}(j)}, y'_j, j, \mathbf{x}))}$$

locally normalized

$$\log P(y_1 \ldots y_n | x, \theta) = \sum_{j=1}^{n} F_\theta(\underbrace{\vec{y}_{Pa(j)}, y_j}_{Y_c}, j, x) - \log \sum_{y'_j=1}^{m} \exp(\quad)$$

Training for BN

$D = \{(x^1, y^1), \ldots (x^N, y^N)\}$: Goal learn θ.

$$LL(\theta, D) = \sum_{i=1}^{N} \log \Pr(\mathbf{y}^i | \mathbf{x}^i, \theta)$$

$$= \sum_{i=1}^{N} \log \prod_j \Pr(y^i_j | \mathbf{y}^i_{\text{Pa}}(j), \mathbf{x}^i, \theta)$$

$$= \sum_i \sum_j \log \Pr(y^i_j | \mathbf{y}^i_{\text{Pa}}(j), \mathbf{x}^i, \theta)$$

$$= \sum_i \sum_{j=1}^{n} F_\theta(\mathbf{y}^i_{\text{Pa}(j)}, y^i_j, j, \mathbf{x}^i)) - \log \sum_{y'=1}^{m} \exp(F_\theta(\mathbf{y}^i_{\text{Pa}(j)}, y', j, \mathbf{x}^i))$$

softmax over $F_\theta(\ )$

Like normal classification task. No challenge arising during training because of graphical model. Normalizer is easy to compute.

Explains the popularity of BNs in training deep networks.

Table Potentials in the feature framework.

Assume $\mathbf{x}^i$ does not exist..(As in HMMs)

- $F_\theta(\mathbf{y}^i_{\text{Pa}(j)}, y^i_j, j)) = \log P(y^i_j | \mathbf{y}^i_{\text{Pa}(j)})$, normalizer vanishes.
- $\Pr(y_j | \mathbf{y}_{\text{Pa}(j)})$ = Table of real values denoting the probability of each value of x_j corresponding to each combination of values of the parents (θ^j).
- If each variables takes m possible values, and has k parents, then each $\Pr(y_j | \mathbf{y}_{\text{Pa}(j)})$ will require $m^k(\overset{-1}{m})$ parameters in θ^j.

$$\theta^j_{v u_1, \ldots, u_k} = \Pr(y_j = v | \mathbf{y}_{pa(j)} = [u_1, \ldots, u_k])$$

Maximum Likelihood estimation of parameters

$$
\begin{aligned}
& \max_\theta \sum_i \sum_j \log P(y^i_j | \mathbf{y}^i_{\mathrm{Pa}(j)}) \\
= \quad & \max_\theta \sum_i \sum_j \log \theta^j_{y^i_j \mathbf{y}^i_{\mathrm{Pa}(j)}} \quad s.t. \sum_v \boxed{\theta^j_{vu_1,\ldots,u_k}} = 1 \ \forall j, u_1, \ldots, u_k \\
= \quad & \max_\theta \sum_i \sum_j \log \theta^j_{y^i_j \mathbf{y}^i_{\mathrm{Pa}(j)}} - \sum_j \sum_{u_1,\ldots,u_k} \lambda^j_{u_1,\ldots,u_k} \left(\sum_v \theta^j_{vu_1,\ldots,u_k} - 1\right)
\end{aligned}
$$

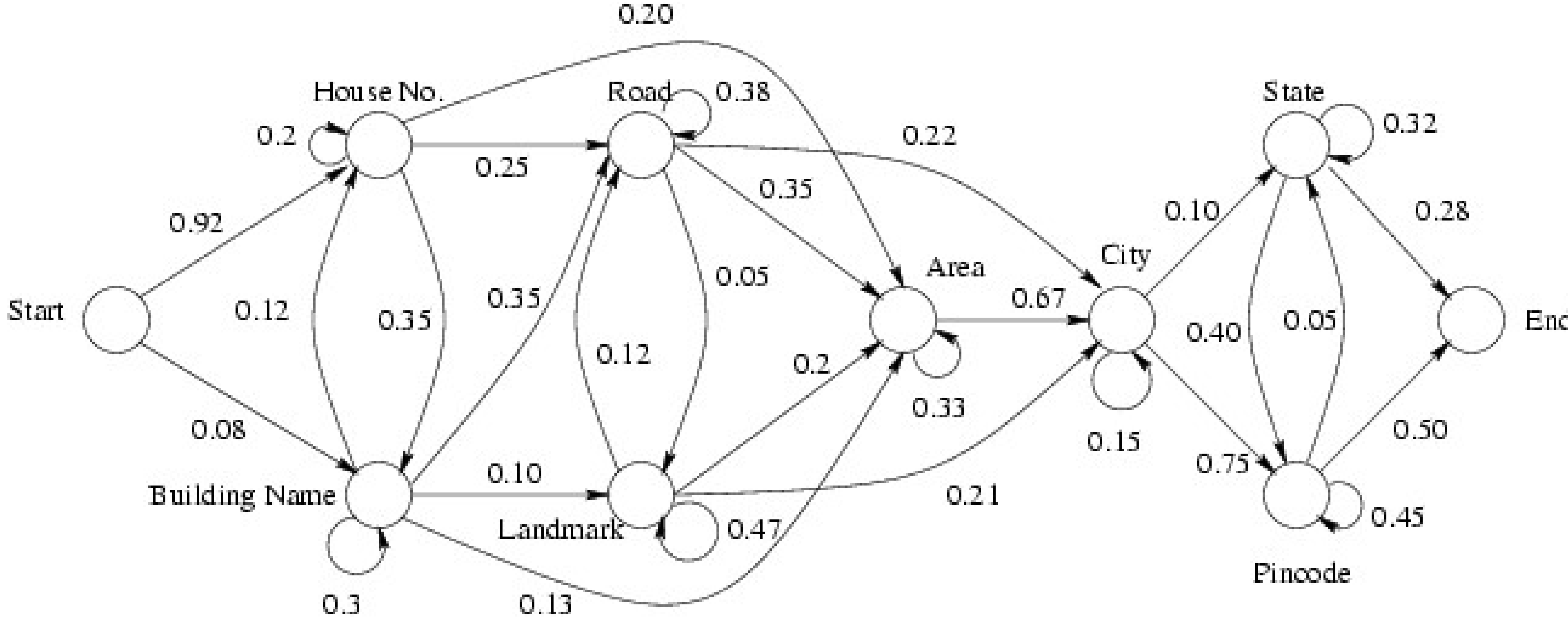
Solve above using gradient descent to get

$$
\underbrace{\theta^j_{vu_1,\ldots,u_k}}_{} = \frac{\sum_{i=1}^{N} [[y^i_j = v, \mathbf{y}^i_{Pa(j)} = u_1, \ldots, u_k]]}{\sum_{i=1}^{N} [[\mathbf{y}^i_{Pa(j)} = \underline{u_1, \ldots, u_k}]]} \tag{1}
$$

HMM parameters

Three types of potentials:

1. Transition probabilities
   $\Pr(y_t = v | y_{t-1} = u) = \frac{\text{Number of transitions from u to v}}{\text{Total transitions out of state u}}$ Example:



2. Emission probabilities, Probability of emitting symbol v from state u
   $\Pr(x_t = v | y_t = u) = \frac{\text{Number of times v generated from u}}{\text{number of transition from u}}$

Example: HMM parameter learning

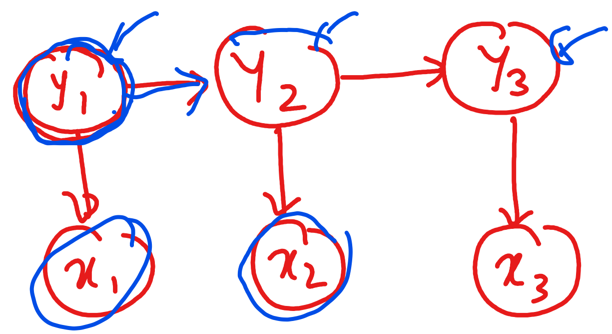
$y_j \in \{1,2,3\}$

$x_j \in \{A,B,C,D\}$

$D = (N = 3, n = 4)$

(y_1, x_1)	(y_2, x_2)	(y_3, x_3)	(y_4, x_4)
1, A	1, B	2, A	3, C
2, B	1, A	3, A	3, D
1, B	1, B	2, C	3, D

$$\theta^1_1 = \frac{\sum_{i=1}^{N} [[y^i_1 = 1]]}{\sum_{i=1}^{N} [[1]]}$$



$P(y) =$

1	2	3
2/3	1/3	0

$\theta^1_{v_1 \ldots v_n}$ $[\theta^1_1$ θ^1_2 $\theta^1_3]$

$P(y|y') =$

y'	y=1	y=2	y=3
1	2/5	2/5	1/5
2	1/3	0	2/3
3	0	0	1

$\theta^2_{v:u} = P(y_2 = v \mid y_1 = u) \leftarrow$ 9 values to them

$\theta^{y_2}_{v:u} = \theta^{y_3}_{v:u} = \theta^{y_4}_{v:u}$

$P(x|y) =$

	x=A	x=B	x=C	x=D
1	2/5	3/5	0	0
2				
3				

$\theta^{x_1}_{u:v} = P(x_1 = u \mid y_1 = v) \leftarrow$ 12 values.

$\theta^{x_1}_{A1}$ $\theta^{x_1}_{B1}$ - - -

$\theta^{x_1} = \theta^{x_2} = \theta^{x_3} = \theta^{x_4}$