(1) Derivation of IPF

Parameter learning in Undirected Models

D: dataset of m i.i.d data-points

$$l(D;\theta) = \log \prod_{j=1}^{m} \frac{1}{Z} \prod_{i} \psi_i(C_i = c_i^{(j)})$$

$$= \sum_{j=1}^{m} \sum_{i} \log \psi_i(C_i = c_i^{(j)}) - m \log Z$$

swap

$$= \sum_{i} \sum_{j=1}^{m} \log \psi_i(C_i = c_i^{(j)}) - m \log Z$$

count

$$= \sum_{i} \sum_{c_i} \underbrace{\text{count}(C_i = c_i)}_{\hat{P}} \log \psi_i(C_i = c_i) - m \log Z$$

$$= m \sum_{i} \underbrace{\sum_{c_i} \hat{P}(C_i = c_i) \log \psi_i(C_i = c_i)}_{\text{Does this look like something familiar?}} - m \log Z$$

Hint: Expectation

Want to learn

Parameter : $\boxed{\psi_k(c_k)}$

What does this mean for HW 5?

$$\theta = \begin{bmatrix} \theta_1 & \theta_2 \\ \theta_3 & \theta_4 \end{bmatrix}$$

θ_1 ↔ $\psi_k(c_k)$

Take derivative:

$$\frac{\partial \ell(D:\theta)}{\partial \psi_k(c_k)} = \frac{\partial}{\partial \psi_k(c_k)} \left[m \sum_i \sum_{c_i} \hat{P}(c_i) \log \psi_i(c_i) \right] - \frac{\partial}{\partial \psi_i(c_i)} [m \log Z]$$

First term $\frac{\partial}{\partial \psi_k(c_k)} \left[m \sum_i \sum_{c_i} \hat{P}(c_i) \log \psi_i(c_i) \right]$

only when they agree on i, i.e. $k=i$

does a term survive

Parameter independance assumption

$$= m \hat{P}(c_i) \frac{\partial}{\partial \psi_k(c_k)} \log \psi_k(c_k)$$

$$= m \frac{\tilde{P}(c_i)}{\psi_k(c_k)}$$

Now $\frac{\partial}{\psi_k(C_k)} \log Z$

$$= \frac{\partial}{\partial \psi_k(C_k)} \log \sum_x \prod_i \psi_i(x_i)$$

$$= \frac{1}{Z} \frac{\partial}{\partial \psi_k(C_k)} \left[\sum_x \prod_i \psi_i(x_i) \right]$$

$$= \frac{1}{Z} \left[\sum_x \left\{ \prod_{i \neq k} \psi_i(x_i) \right\} \underbrace{\frac{\partial}{\partial \psi_k(C_k)} \psi_k(x_k)}_{= I(x_k = C_k)} \right]$$

$$= \frac{1}{\psi_k(C_k)} \left[\frac{1}{Z} \sum_x \prod_i \psi_i(x_i) \, I(x_k = C_k) \right]$$

$$= \frac{1}{\psi_k(C_k)} \left[\sum_x \underbrace{\frac{1}{Z} \prod_i \psi_i(x_i)}_{P(x)} I(x_k = C_k) \right]$$

$$= \frac{E[I(x_k = C_k)]}{\psi_k(C_k)} = \frac{P_k(x_k = C_k)}{\psi_k(C_k)}$$

② Parameter Learning for Log-Linear Models

$$P(x) = \frac{1}{Z} \exp\left\{ \sum_i w_i \phi_i(x_i) \right\}$$

Again D: dataset of m i.i.d samples

$$l(D:\theta) = \log \prod_{j=1}^{N} \frac{1}{Z} \exp\left\{ \sum_i w_i^T \phi_i(x_i) \right\}$$

$$= \sum_j \sum_i w_i^T \phi_i(x_i) - m \log Z$$

(swap)

$$= \sum_i \sum_j w_i^T \phi_i(x_i) - m \log Z$$

(count)

$$= m \sum_i \sum_{c_i} \hat{P}(x_i = c_i)\, w_i^T \phi_i(x_i = c_i) - m \log Z$$

Want to learn: $\vec{w}_k$

Take derivative:

$$\frac{\partial}{\partial \vec{w}_k} l(D:\theta) = \frac{\partial}{\partial \vec{w}_k}\left[m \sum_i \sum_{c_i} \hat{P}_R(x_i=c_i)\, w_i^T \phi_i(x_i=c_i)\right] - \frac{\partial}{\partial \vec{w}_k} m \log Z$$

(non-zero only when $k=i$)

$$= m \sum_{c_i} \hat{P}(x_i=c_i)\, \phi_i(x_i=c_i) - \frac{\partial}{\partial w_i} \log Z$$

$$= m\, E_{\hat{P}}[\phi_i(x_i)] - m \frac{\partial}{\partial w_i} \log Z$$

$$\frac{\partial}{\partial \vec{w}_k} \log Z = \frac{\partial}{\partial w_k} \log \sum_x \exp\left\{ \sum_i w_i^T \phi_i(x_i)\right\}$$

$$= \frac{1}{Z}\left[\frac{\partial}{\partial \vec{w}_k} \sum_x \exp\left\{ \sum_i w_i^T \phi_i(x_i)\right\}\right]$$

$$= \frac{1}{Z} \sum_x \exp\left\{ \sum_i w_i^T \phi_i(x_i)\right\} \frac{\partial}{\partial \vec{w}_i} \sum_i w_i^T \phi_i(x_i)$$

$$= \sum_x P_R(x)\, \phi_i(x_i)$$

$$= E_P[\phi_i(x_i)]$$

Thus gradient

$$= m\left\{ \underbrace{E_{\hat{p}}[\phi_i(x_i)]}_{\text{what data believes / knows about feature distribution}} - \underbrace{E_p[\phi_i(x_i)]}_{\text{what our model believes about feature distribution}} \right\}$$

Algorithm

1) Initialize weights

2) Compute gradient

3) take small step / whatever heuristic you like

4) Repeat

Again notice: Learning requires inference