Stat 521A
Lecture 25

Outline

- MAP param estimation for UGMs (20.1-20.4)
- Learning using approximate inference (20.5)
- Alternative objectives (20.6)

Likelihood fn for UGMs

- Log-linear model

$$P(X_1, \ldots, X_n : \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left\{ \sum_{i=1}^{k} \theta_i f_i[\boldsymbol{D}_i] \right\}.$$

$$\ln Z(\boldsymbol{\theta}) = \ln \sum_{\xi} \exp\left\{ \sum_i \theta_i f_i[\xi] \right\}.$$

Concave

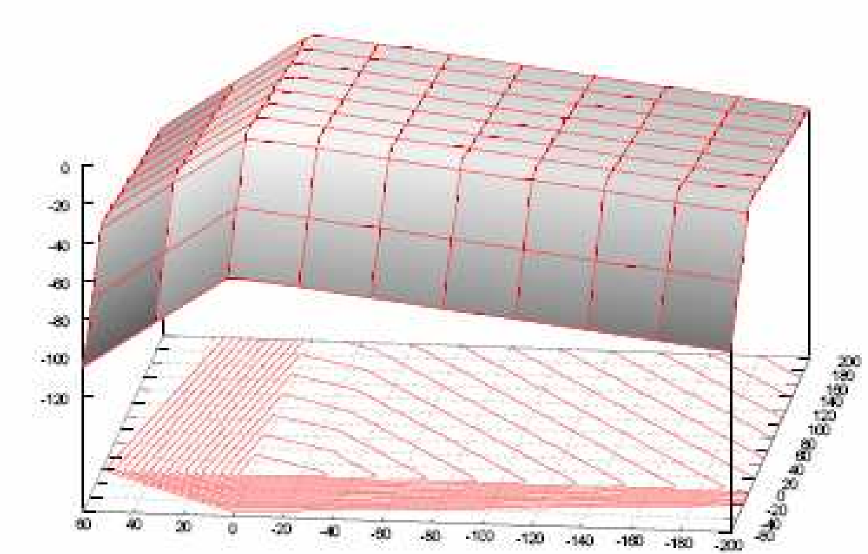


Figure 20.1 Log-likelihood surface for the Markov network A—B—C, as a function of $\ln \phi_1[a^1, b^1]$ (x-axis) and $\ln \phi_2[b^1, c^1]$ (y-axis); all other parameters in both potentials are set to 1. The data set $\mathcal{D}$ has $M = 100$ instances, for which $M[a^1, b^1] = 90$ and $M[b^1, c^1] = 15$. (The other sufficient statistics are irrelevant, as all of the other log-parameters are 0.)

LogZ: first deriv

Proposition 20.2.3: *Let Φ be a set of features. Then,*

$$
\begin{aligned}
\frac{\partial}{\partial \theta_i} \ln Z(\boldsymbol{\theta}) &= \boldsymbol{E}_{\boldsymbol{\theta}}[f_i] \\
\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln Z(\boldsymbol{\theta}) &= \boldsymbol{Cov}_{\boldsymbol{\theta}}[f_i; f_j],
\end{aligned}
$$

where $\boldsymbol{E}_{\boldsymbol{\theta}}[f_i]$ *is a shorthand for* $\boldsymbol{E}_{P(\mathcal{X}:\boldsymbol{\theta})}[f_i]$.

$$
\begin{aligned}
\frac{\partial}{\partial \theta_i} \ln Z(\boldsymbol{\theta}) &= \frac{1}{Z(\boldsymbol{\theta})} \sum_{\xi} \frac{\partial}{\partial \theta_i} \exp\left\{ \sum_j \theta_j f_j[\xi] \right\} \\
&= \frac{1}{Z(\boldsymbol{\theta})} \sum_{\xi} f_i[\xi] \exp\left\{ \sum_j \theta_j f_j[\xi] \right\} \\
&= \boldsymbol{E}_{\boldsymbol{\theta}}[f_i].
\end{aligned}
$$

logZ: second deriv

$$
\begin{aligned}
\frac{\partial^2}{\partial\theta_j\partial\theta_i}\ln Z(\boldsymbol{\theta}) &= \frac{\partial}{\partial\theta_j}\left[\frac{1}{Z(\boldsymbol{\theta})}\sum_{\xi} f_i[\xi]\exp\left\{\sum_k \theta_k f_k[\xi]\right\}\right] \\
&= -\frac{1}{Z(\boldsymbol{\theta})^2}\left(\frac{\partial}{\partial\theta_j}Z(\boldsymbol{\theta})\right)\sum_{\xi} f_i[\xi]\exp\left\{\sum_k \theta_k f_k[\xi]\right\} \\
&\qquad +\frac{1}{Z(\boldsymbol{\theta})}\sum_{\xi} f_i[\xi]f_j[\xi]\exp\left\{\sum_k \theta_k f_k[\xi]\right\} \\
&= -\frac{1}{Z(\boldsymbol{\theta})^2}Z(\boldsymbol{\theta})\mathbf{E}_{\boldsymbol{\theta}}[f_i]\sum_{\xi} f_i[\xi]\tilde{P}(\xi:\boldsymbol{\theta}) \\
&\qquad +\frac{1}{Z(\boldsymbol{\theta})}\sum_{\xi} f_i[\xi]f_j[\xi]\tilde{P}(\xi:\boldsymbol{\theta}) \\
&= \mathbf{E}_{\boldsymbol{\theta}}[f_i]\sum_{\xi} f_i[\xi]P(\xi:\boldsymbol{\theta}) \\
&\qquad +\sum_{\xi} f_i[\xi]f_j[\xi]P(\xi:\boldsymbol{\theta}) \\
&= \mathbf{E}_{\boldsymbol{\theta}}[f_i f_j]-\mathbf{E}_{\boldsymbol{\theta}}[f_i]\mathbf{E}_{\boldsymbol{\theta}}[f_j] \\
&= \boldsymbol{Cov}_{\boldsymbol{\theta}}[f_i;f_j].
\end{aligned}
$$

Finding the MLE

At optimum, model moments = empirical moments

$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\boldsymbol{\theta} : \mathcal{D}) = \boldsymbol{E}_{\mathcal{D}}[f_i[\mathcal{X}]] - \boldsymbol{E}_{\boldsymbol{\theta}}[f_i]. \tag{20.4}$$

This analysis provides us with a precise characterization of the maximum likelihood parameters $\hat{\boldsymbol{\theta}}$:

Theorem 20.3.1: *Let Φ be a set of features. Then, $\boldsymbol{\theta}$ is a maximal likelihood parameter assignment if and only if $\boldsymbol{E}_{\mathcal{D}}[f_i[\mathcal{X}]] = \boldsymbol{E}_{\hat{\boldsymbol{\theta}}}[f_i]$ for all i.*

Must perform inference once per gradient

Just do gradient based optimization, eg stochastic gradient descent.
Expensive to compute Hessian explicitly, so use Quasi-Newton.

$$\frac{\partial}{\partial \theta_i \partial \theta_j} \ell(\boldsymbol{\theta} : \mathcal{D}) = -M \boldsymbol{C}ov_{\boldsymbol{\theta}}[f_i; f_j].$$

CRFs

- Conditional density models

$$\ell_{\boldsymbol{Y}|\mathbf{X}}(\boldsymbol{\theta}:\mathcal{D}) = \ln P(y[1,\ldots,M] \mid x[1,\ldots,M], \boldsymbol{\theta}) = \sum_{m=1}^{M} \ln P(\boldsymbol{y}[m] \mid \boldsymbol{x}[m], \boldsymbol{\theta}).$$

$$\frac{\partial}{\partial \theta_i} \ell_{\boldsymbol{Y}|\mathbf{X}}(\boldsymbol{\theta}:\mathcal{D}) = \sum_{m=1}^{M} \left[f_i[\boldsymbol{y}[m], \boldsymbol{x}[m]] - \boldsymbol{E}_{\boldsymbol{\theta}}[f_i \mid \boldsymbol{x}[m]] \right].$$

Must perform inference M times per gradient

MRFs with hidden variables

- Must perform inference M times per gradient

$$\frac{1}{M}\ln P(\mathcal{D}\mid\boldsymbol{\theta}) = \frac{1}{M}\ln\left(\sum_{m=1}^{M}\sum_{\boldsymbol{h}[m]} P(\boldsymbol{o}[m],\boldsymbol{h}[m]\mid\boldsymbol{\theta})\right)$$

$$= \frac{1}{M}\ln\left(\sum_{m=1}^{M}\sum_{\boldsymbol{h}[m]} \tilde{P}(\boldsymbol{o}[m],\boldsymbol{h}[m]\mid\boldsymbol{\theta})\right) - \ln Z.$$

$$\frac{\partial}{\partial\theta_i}\ln\sum_{\boldsymbol{h}[m]}\tilde{P}(\boldsymbol{o}[m],\boldsymbol{h}[m]\mid\boldsymbol{\theta}) = \boldsymbol{E}_{\boldsymbol{h}[m]\sim P(\mathcal{H}[m]\mid\boldsymbol{o}[m],\boldsymbol{\theta})}[f_i],$$

Proposition 20.3.3: *For a data set* $\mathcal{D}$

$$\frac{\partial}{\partial\theta_i}\frac{1}{M}\ell(\boldsymbol{\theta}:\mathcal{D}) = \frac{1}{M}\left[\sum_{m=1}^{M}\boldsymbol{E}_{\boldsymbol{h}[m]\sim P(\mathcal{H}[m]\mid\boldsymbol{o}[m],\boldsymbol{\theta})}[f_i]\right] - \boldsymbol{E}_{\boldsymbol{\theta}}[f_i].$$

clamped unclamped

CRFs with hidden variables

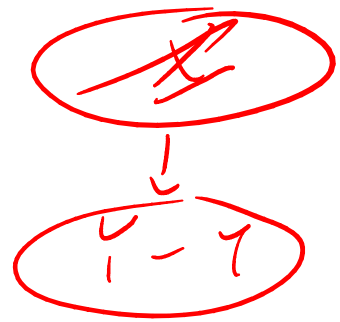
- Training is similar to MRFs with hidden variables, except expectations condition on x_n, so need to be redone for each case

Summary

MRF

$$\nabla = \sum_i f(x_i) - M E_X[f(X)]$$

CRF

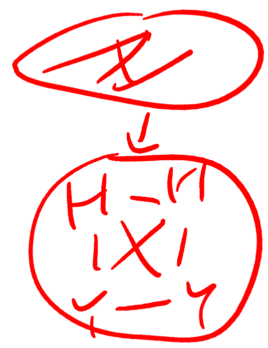


$$\nabla = \sum_i f(x_i, y_i) - \sum_i E_Y[f(x_i, Y)]$$

MRF + H

$$\nabla = \sum_i E_H f(H_i, x_i) - M E_{H,X}[f(H, X)]$$

CRF + H



$$\nabla = \sum_i E_H f(x_i, y_i, H) - \sum_i E_{H,Y}[f(x_i, Y, H)]$$

ML and MaxEnt

- MLE in the expfam is equivalent to MaxEnt subject to moment constraints

Maximum-Entropy

Find $Q(\mathcal{X})$
that maximize $H_Q(\mathcal{X})$
subject to

$$E_Q[f_i] = E_{\mathcal{D}}[f_i] \quad i = 1, \ldots, k$$

Theorem 20.3.4: *The distribution Q^* is the maximum entropy distribution satisfying Eq. (20.10) if and only if $Q^* = P_{\hat{\boldsymbol{\theta}}}$, where*

$$P_{\hat{\boldsymbol{\theta}}}(\mathcal{X}) = \frac{1}{Z(\hat{\boldsymbol{\theta}})} \exp\left\{\sum_i \hat{\theta}_i f_i[\mathcal{X}]\right\}$$

and $\hat{\boldsymbol{\theta}}$ is the maximum likelihood parameterization relative to $\mathcal{D}$.

Proof

PROOF For notational simplicity, let $P = P_{\hat{\theta}}$. From Theorem 20.3.1, it follows that $\mathbb{E}_P[f_i] = \mathbb{E}_{\mathcal{D}}[f_i[\mathcal{X}]]$ for $i = 1, \ldots, k$, and hence that P satisfies the constraints of Eq. (20.10). Therefore, to prove that $P = Q^*$, we need only show that $\mathbb{H}_P(\mathcal{X}) \geq \mathbb{H}_Q(\mathcal{X})$ for all other distributions Q that satisfy these constraints. Consider any such distribution Q.

From Theorem 8.4.1, it follows that:

$$\mathbb{H}_P(\mathcal{X}) = -\sum_i \hat{\theta}_i \mathbb{E}_P[f_i] + \ln Z(\theta). \tag{20.11}$$

Thus,

$$\begin{aligned}
\mathbb{H}_P(\mathcal{X}) - \mathbb{H}_Q(\mathcal{X}) &= -\left[\sum_i \theta_i \mathbb{E}_P[f_i[\mathcal{X}]]\right] + \ln Z_P - \mathbb{E}_Q[-\ln Q(\mathcal{X})] \\
(i) \quad &= -\left[\sum_i \theta_i \mathbb{E}_Q[f_i[\mathcal{X}]]\right] + \ln Z_P + \mathbb{E}_Q[\ln Q(\mathcal{X})] \\
&= \mathbb{E}_Q[-\ln P(\mathcal{X})] + \mathbb{E}_Q[\ln Q(\mathcal{X})] \\
&= \mathbb{D}(Q|P) \geq 0,
\end{aligned}$$

where (i) follows from the fact that both $P_{\hat{\theta}}$ and Q satisfy the constraints, so that $\mathbb{E}_{P_{\hat{\theta}}}[f_i] = \mathbb{E}_Q[f_i]$ for all i.

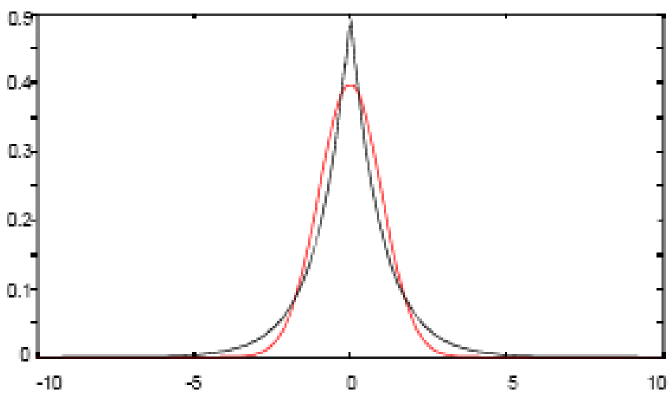
We conclude that $\mathbb{H}_{P_{\hat{\theta}}}(\mathcal{X}) \geq \mathbb{H}_Q(\mathcal{X})$ with equality if and only if $P_{\hat{\theta}} = Q$. Thus, the maximum entropy distribution Q^* is necessarily equal to $P_{\hat{\theta}}$, proving the result. ■

MAP estimation

- Convex prior + convex likelihood makes objective strictly convex (unique soln)
- Also helps prevent overfitting
- L2 and L1

$$P(\theta \mid \sigma^2) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{\theta_i^2}{2\sigma^2}\right\},$$

$$P_{Laplacian}(\theta \mid \beta) = \frac{1}{2\beta} \exp\left\{-\frac{|\theta|}{\beta}\right\}.$$



$$\begin{aligned} \ln \frac{P(\xi)}{P(\xi')} &= \sum_{i=1}^{k} \theta_i f_i[\xi] - \sum_{i=1}^{k} \theta_i f_i[\xi'] \\ &= \sum_{i=1}^{k} \theta_i (f_i[\xi] - f_i[\xi']). \end{aligned}$$

Learning with approximate inference

- Recall that the gradient requires model expectation over the features

$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta : \mathcal{D}) = \mathbb{E}_{\mathcal{D}}[f_i[\mathcal{X}]] - \mathbb{E}_{\boldsymbol{\theta}}[f_i]. \tag{20.4}$$

- We can use approximate inference to approximate the expectation, but approximate gradients can cause learning to diverge

Pseudo moment matching

- At the optimum, the pseudo marginals must satisfy

$$E_{\beta_i[C_i]}[f_{C_i}] = E_{\mathcal{D}}[f_i[C_i]].$$

- Suppose we use tabular features. Then

$$\beta_i[c_i^j] = \hat{P}(c_i^j).$$

- Hence we don't need to run inference. There are multiple potentials that can generate these beliefs. We can uniquely recover one set using (for any ordering i<j)

$$\phi_i \leftarrow \frac{\beta_i}{\mu_{i,j}}.$$

Unified inference and learning

- Pseudo moment matching only works for unconditional, tabular potentials with no tying and no regularizer
- To combine BP with param optimization, we can optimize

Approx-Maximum-Entropy

$$
\begin{array}{lll}
\textbf{Find} & Q & \\
\textbf{that maximize} & \sum_{C_i \in \mathcal{U}} H_{\beta_i}(C_i) - \sum_{(C_i - C_j) \in \mathcal{U}} H_{\mu_{i,j}}(S_{i,j}) & \\
\textbf{subject to} & E_{\beta_i}[f_i] = E_{\mathcal{D}}[f_i] & i = 1, \ldots, k \\
 & Q \in Local[\mathcal{U}] &
\end{array}
$$

The model parameters theta are the Lagrange multipliers for E[f]
And the messages are the Lagrange multipliers for the local consistency

Example

A	B	C
0	0	0
0	1	0
1	0	0

Tied features

$f_{00}(x,y) = 1$ iff $x=y=0$

$f_{11}(x,y) = 1$ iff $x=y=1$

$E_D\, f_{00} = I(A_1 = B_1) + I(A_1 = C_1) + I(B_1 = C_1) + \ldots = 5$

Find Q

that maximize $H_{\beta_1}(A,B) + H_{\beta_2}(B,C) + H_{\beta_3}(A,C) - H_{\mu_{1,2}}(B) - H_{\mu_{2,3}}(C) - H_{\mu_{2,3}}(A$

subject to

$$\sum_i E_{\beta_i}[f_{00}] = 2 \quad (20.17)$$

$$\sum_i E_{\beta_i}[f_{11}] = 0 \quad (20.18)$$

$$\sum_a [\beta_1[a,b]] - \sum_c [\beta_2[b,c]] = 0 \quad (\forall b) \quad (20.19)$$

$$\sum_b [\beta_2[b,c]] - \sum_a [\beta_3[a,c]] = 0 \quad (\forall c) \quad (20.20)$$

$$\sum_c [\beta_3[a,c]] - \sum_b [\beta_1[a,b]] = 0 \quad (\forall a) \quad (20.21)$$

$$\sum_{c_i} \beta_i[c_i] = 1 \quad i = 1,2,3 \quad (20.22)$$

$$\beta_i \geq 0 \quad i = 1,2,3 \quad (20.23)$$

Double loop algorithm

- Inner loop optimizes δ_{ij} by iterating the fixed point eqns
- Outer loop optimizes θ eg using gradient descent

Approximating Z

- Loglik

$$\ell(\theta : \xi) = \ln \tilde{P}(\xi \mid \theta) - \ln Z(\theta)$$

$$\ln \tilde{P}(\xi \mid \theta) - \ln \left(\sum_{\xi'} \tilde{P}(\xi' \mid \theta) \right).$$

- We can approximate the sum in different ways

Pseudolikeliood

- Define

$$P(\xi) = \prod_{j=1}^{n} P(x_j \mid x_1, \ldots, x_{j-1}) \qquad P(\xi) \approx \prod_j P(x_j \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$$

$$\ell_{\text{pseudo}}(\boldsymbol{\theta} : \mathcal{D}) = \frac{1}{M} \sum_m \sum_j \ln P(x_j[m] \mid \boldsymbol{x}_{-j}[m], \boldsymbol{\theta})$$

$$\begin{aligned} P(x_j \mid \boldsymbol{x}_{-j}) = \frac{P(x_j, \boldsymbol{x}_{-j})}{P(\boldsymbol{x}_{-j})} &= \frac{\tilde{P}(x_j, \boldsymbol{x}_{-j})}{\tilde{P}(\boldsymbol{x}_{-j})} \\ &= \frac{\tilde{P}(x_j, \boldsymbol{x}_{-j})}{\sum_{x'_j} \tilde{P}(x'_j, \boldsymbol{x}_{-j})}. \end{aligned}$$

Gradient of PL

Convex

$$\ln P(x_j \mid x_{-j}) = \left(\sum_{i \,:\, Scope[f_i] \ni X_j} \theta_i f_i[x_j, u_j] \right) - \ln \left(\sum_{x'_j} \exp \left\{ \sum_{i \,:\, Scope[f_i] \ni X_j} \theta_i f_i[x'_j, u_j] \right\} \right).$$

$$\frac{\partial}{\partial \theta_i} \ln P(x_j \mid x_{-j}) = f_i[x_j, x_{-j}] - \mathbb{E}_{x'_j \sim P_{\theta}(X_j \mid x_{-j})} \left[f_i[x'_j, x_{-j}] \right].$$

Proposition 20.6.1:

$$\frac{\partial}{\partial \theta_i} \ell_{\text{pseudo}}(\theta : \mathcal{D}) = \sum_{j : X_j \in Scope[f_i]} \left(\frac{1}{M} \sum_m f_i[\xi[m]] - E_{x'_j \sim P_{\theta}(X_j \mid x_{-j}[m])} \left[f_i[x'_j, x_{-j}[m]] \right] \right). \tag{20.31}$$

Consistency of PL

- Thm 20.6.2 (Besag). If data is generated from our model with params θ*, then as M->inf, argmax PL(θ) -> θ*.
- Pf. The empirical approaches P(θ*). Hence

$$\frac{1}{M}\sum_m f_i[\xi[m]] \longrightarrow \mathbb{E}_{\xi\sim P_{\theta^*}(\mathcal{X})}[f_i[\xi]].$$

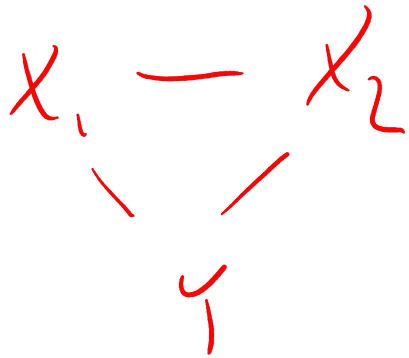
- And

$$\begin{aligned}\frac{1}{M}\sum_m E_{x'_j\sim P_{\theta^*}(X_j\mid \boldsymbol{x}_{-j}[m])}[f_i[x'_j,\boldsymbol{x}_{-j}[m]]] &= \sum_{\boldsymbol{x}_{-j}} P_{\mathcal{D}}(\boldsymbol{x}_{-j})\sum_{x'_j} P_{\theta^*}(x'_j \mid \boldsymbol{x}_{-j}) f_i[x'_j,\boldsymbol{x}_{-j}] \\ &\longrightarrow \sum_{\boldsymbol{x}_j} P_{\theta^*}(\boldsymbol{x}_{-j})\sum_{x'_j} P_{\theta^*}(x'_j \mid \boldsymbol{x}_{-j}) f_i[x'_j,\boldsymbol{x}_{-j}] \\ &= \mathbb{E}_{\xi\sim P_{\theta^*}}[f_i[\xi]].\end{aligned}$$

- Hence gradient of PL is zero at θ*.

Problem with PL

- Ex 20.6.3 (cf Hinton’s greek vase)



Assume X1, X2 are strongly correlated (eg mirror images),
And X1,Y and X2,Y are less strongly correlated.
PL will learn that X1 can be predicted from X2, and will ignore Y.
At test time, if we observe Y and want to predict X1, we are hosed.

Sample based learning

- Recall loglik is $$\frac{1}{M}\ell(\theta : \mathcal{D}) = \sum_i \theta_i E_{\mathcal{D}}[f_i[d_i]] - \ln Z(\theta),$$

$$\begin{aligned} Z(\theta) &= \sum_{\xi} \exp\left\{\sum_i \theta_i f_i[\xi]\right\} \\ &= \sum_{\xi} \frac{Q(\xi)}{Q(\xi)} \exp\left\{\sum_i \theta_i f_i[\xi]\right\} \\ &= E_Q\left[\frac{1}{Q(\mathcal{X})} \exp\left\{\sum_i \theta_i f_i[\mathcal{X}]\right\}\right]. \end{aligned}$$

Sample K x's given θ
Compute ln $Z(\theta)$
Update θ
Repeat

$$\begin{aligned} Z(\theta) &= E_{P_{\theta^0}}\left[\frac{Z(\theta^0)\exp\{\sum_i \theta_i f_i[\mathcal{X}]\}}{\exp\{\sum_i \theta_i^0 f_i[\mathcal{X}]\}}\right] \\ &= Z(\theta^0) E_{P_{\theta^0}}\left[\exp\left\{\sum_i (\theta_i - \theta_i^0) f_i(\mathcal{X})\right\}\right]. \end{aligned}$$

$$\ln Z(\theta) \approx \ln\left(\frac{1}{K}\sum_{k=1}^{K} \exp\left\{\sum_i (\theta_i - \theta_i^0) f_i(\xi^k)\right\}\right) + \ln Z(\theta^0).$$

Contrastive divergence

- Might need to sample many x's to accurately approximate Z, but this is slow
- So define a set D- of randomly perturbed neighbors of D, and use

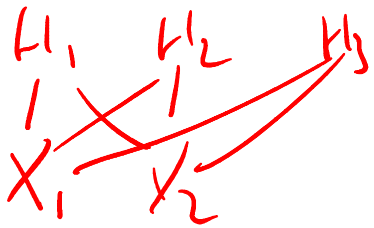
$$\ell_{\mathrm{CD}}(\boldsymbol{\theta} : \mathcal{D} \| \mathcal{D}^-) = \left[\boldsymbol{E}_{\xi \sim \hat{P}_{\mathcal{D}}}[\ln P_{\boldsymbol{\theta}}(\xi)] - \boldsymbol{E}_{\xi \sim \hat{P}_{\mathcal{D}^-}}[\ln P_{\boldsymbol{\theta}}(\xi)] \right],$$

$$\frac{\partial}{\partial \theta_i} \ell_{\mathrm{CD}}(\boldsymbol{\theta} : \mathcal{D} \| \mathcal{D}^-) = \boldsymbol{E}_{\hat{P}_{\mathcal{D}}}[f_i[\mathcal{X}]] - \boldsymbol{E}_{\hat{P}_{\mathcal{D}^-}}[f_i].$$

- Often x_i- is generated by applying 1 step of Gibbs sampling to x_i

CD for RBMs

- RBMs have 1 layer of hidden variables, so we need an additional expectation



$$
\begin{aligned}
\nabla_i &= E_{\mathbf{x}\sim D} E_{\mathbf{h}} f_i(\mathbf{x},\mathbf{h}) - E_{\mathbf{x}\sim D^-} E_{\mathbf{h}} f_i(\mathbf{x},\mathbf{h}) \\
&\approx \frac{1}{N}\sum_n E_{\mathbf{h}_n} f_i(\mathbf{x}_n,\mathbf{h}_n) - E_{\mathbf{h}_n} f_i(\mathbf{x}_n^-,\mathbf{h}_n) \\
&\approx \frac{1}{N}\sum_n f_i(\mathbf{x}_n,\mathbf{h}_n^+) - f_i(\mathbf{x}_n^-,\mathbf{h}_n^-)
\end{aligned}
$$

Stop learning when your dreams match reality

$x \to h^+$

$\mathbf{h}_n^+ \sim p(\mathbf{h}|\mathbf{x}_n,\boldsymbol{\theta})$ Interpretation of data

$x \to h^+ \to x^-$

$\mathbf{x}_n^- \sim p(\mathbf{x}|\mathbf{h}_n^+,\boldsymbol{\theta})$ Reconstruction/ Fantasy data

$x \to h^+ \to x^- \to h^-$

$\mathbf{h}_n^- \sim p(\mathbf{h}|\mathbf{x}_n^-,\boldsymbol{\theta})$ Interpretation of your fantasies

MAP approximation (perceptron training)

- Let us approximate Z (sum over all X) by the MAP estimate. Objective becomes

$$\frac{1}{M}\ell(\boldsymbol{\theta} : \mathcal{D}) - \ln P(\xi^{\mathrm{MAP}}(\boldsymbol{\theta}) \mid \boldsymbol{\theta}), \qquad \frac{1}{M}\sum_{m=1}^{M} \ln \tilde{P}(\xi[m] \mid \boldsymbol{\theta}) - \ln \tilde{P}(\xi^{\mathrm{MAP}}(\boldsymbol{\theta}) \mid \boldsymbol{\theta}).$$

- For a single data term

$$\begin{aligned}
\ln P(\xi[m] \mid \boldsymbol{\theta}) &- \ln P(\xi^{\mathrm{MAP}}(\boldsymbol{\theta}) \mid \boldsymbol{\theta}) \\
&= [\ln \tilde{P}(\xi[m] \mid \boldsymbol{\theta}) - \ln Z(\boldsymbol{\theta})] - [\ln \tilde{P}(\xi^{\mathrm{MAP}}(\boldsymbol{\theta}) \mid \boldsymbol{\theta}) - \ln Z(\boldsymbol{\theta})] \\
&= \ln \tilde{P}(\xi[m] \mid \boldsymbol{\theta}) - \ln \tilde{P}(\xi^{\mathrm{MAP}}(\boldsymbol{\theta}) \mid \boldsymbol{\theta}) \\
&= \sum_i \theta_i [f_i[\xi[m]] - f_i[\xi^{\mathrm{MAP}}(\boldsymbol{\theta})]].
\end{aligned}$$

- Hence gradient is

$$E_{\mathcal{D}}[f_i[\mathcal{X}]] - f_i[\xi^{\mathrm{MAP}}(\boldsymbol{\theta})].$$

Problem with MAP approximation

- The objective is always negative or 0 since

$$\frac{1}{M}\ell(\theta : \mathcal{D}) - \ln P(\xi^{\mathrm{MAP}}(\theta) \mid \theta),$$

$$\ln P(\xi[m] \mid \theta) \leq \ln P(\xi^{\mathrm{MAP}}(\theta) \mid \theta).$$

- We can always achieve the maximum of 0 by setting \theta=0

$$\begin{aligned}
\ln P(\xi[m] \mid \theta) &- \ln P(\xi^{\mathrm{MAP}}(\theta) \mid \theta) \\
&= [\ln \tilde{P}(\xi[m] \mid \theta) - \ln Z(\theta)] - [\ln \tilde{P}(\xi^{\mathrm{MAP}}(\theta) \mid \theta) - \ln Z(\theta)] \\
&= \ln \tilde{P}(\xi[m] \mid \theta) - \ln \tilde{P}(\xi^{\mathrm{MAP}}(\theta) \mid \theta) \\
&= \sum_i \theta_i [f_i[\xi[m]] - f_i[\xi^{\mathrm{MAP}}(\theta)]].
\end{aligned}$$

- “collapsing” problem

Max-margin training

- For conditional density models, we can change the objective to the following, which prevents collapsing

$$\ln P_{\boldsymbol{\theta}}(\boldsymbol{y}[m] \mid \boldsymbol{x}[m]) - \left[\max_{\boldsymbol{y} \neq \boldsymbol{y}[m]} \ln P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}[m]) \right].$$

Find $\gamma, \boldsymbol{\theta}$
that maximize γ
subject to

$$\ln P_{\boldsymbol{\theta}}(\boldsymbol{y}[m] \mid \boldsymbol{x}[m]) - \ln P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}[m]) \geq \gamma \qquad \text{for all } m,\ \boldsymbol{y} \neq \boldsymbol{y}[m] \quad (20.36)$$

$$\boldsymbol{\theta}^T \left(\boldsymbol{f}(\boldsymbol{y}[m], \boldsymbol{x}[m]) - \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x}[m]) \right) \geq \gamma.$$

To prevent margin blowing up we bound \theta

Simple-Max-Margin

Find $\boldsymbol{\theta}$
that minimize $\|\boldsymbol{\theta}\|_2^2$
subject to

$$\boldsymbol{\theta}^T \left(\boldsymbol{f}(\boldsymbol{y}[m], \boldsymbol{x}[m]) - \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{x}[m]) \right) \geq 1 \qquad \text{for all } m,\ \boldsymbol{y} \neq \boldsymbol{y}[m]$$

QP: quad obj+linear constraints

Slack variables

- We want to minimize ||w||^2 st

$$\forall_i \forall_{Y_i' \neq Y_i} \log p(Y_i|w, X_i) - \log p(Y_i'|w, X_i) \geq 1.$$

- But we may not be able to achieve this gap, so we introduce slack variables (results in a Hidden Markov Support Vector Machine)

$$\min_{w,\xi} \sum_i \xi_i + \lambda ||w||_2^2,$$

$$s.t.\ \ \forall_i \forall_{Y_i' \neq Y_i} \log p(Y_i|w, X_i) - \log p(Y_i'|w, X_i) \geq 1 - \xi_i, \ \ \forall_i\ \xi_i \geq 0$$

Thanks to Mark Schmidt

Margin rescaling

- Intuitively if Yi' is similar to Yi, we don't mind if their probabilities are similar, but if they are very different, we want the gap to grow
- This gives max-margin markov network (M3N) aka structural SVM

$$\min_{w,\xi} \sum_i \xi_i + \lambda ||w||_2^2,$$

$$. \quad \forall_i \forall_{Y_i' \neq Y_i} \log p(Y_i|w, X_i) - \log p(Y_i'|w, X_i) \geq \Delta(Y_i, Y_i') - \xi_i, \quad \forall_i \xi_i \geq 0,$$

Thanks to Mark Schmidt

Unconstrained form

- We can eliminate the slack vars to get

$$\min_w \sum_i \max_{Y_i' \neq Y_i} \left(\Delta(Y_i, Y_i') - \log p(Y_i|w, X_i) + \log p(Y_i'|w, X_i)\right)^+ + \lambda ||w||_2^2,$$

- Requires 2nd best decoding. But since $\Delta(Y_i,Y_i)=0$ we can write

$$\min_w \sum_i \max_{Y_i'} (\Delta(Y_i, Y_i') + \log p(Y_i'|w, X_i)) - \log p(Y_i|w, X_i) + \lambda ||w||_2^2,$$

- This can use generic MAP decoders that just change the local evidence potentials on Y'.
- For associative markov nets, globally optimal.

Thanks to Mark Schmidt

Cutting plane optimization

- Many possible optimization methods
- Simple approach for QP is cutting planes:
- Maximize quad objective with empty set of constraints – this is an upper bound.
- Add a violated constraint (*)
- Repeat until no violations.
- Thm: only need to add a poly num constraints.
- To find if constraints are violated: define

$$y^{map} = \arg \max_{y \neq y[m]} \tilde{P}(y, x[m]).$$

- If P(y[m],x[m]) < p(ymap,x[m]) +1, add this violation. Else all constraints for m'th case are ok

$$\tilde{P}(y[m], x[m]) > \tilde{P}(y^{map}, x[m]) + 1 \geq \tilde{P}(y, x[m]) + 1,$$