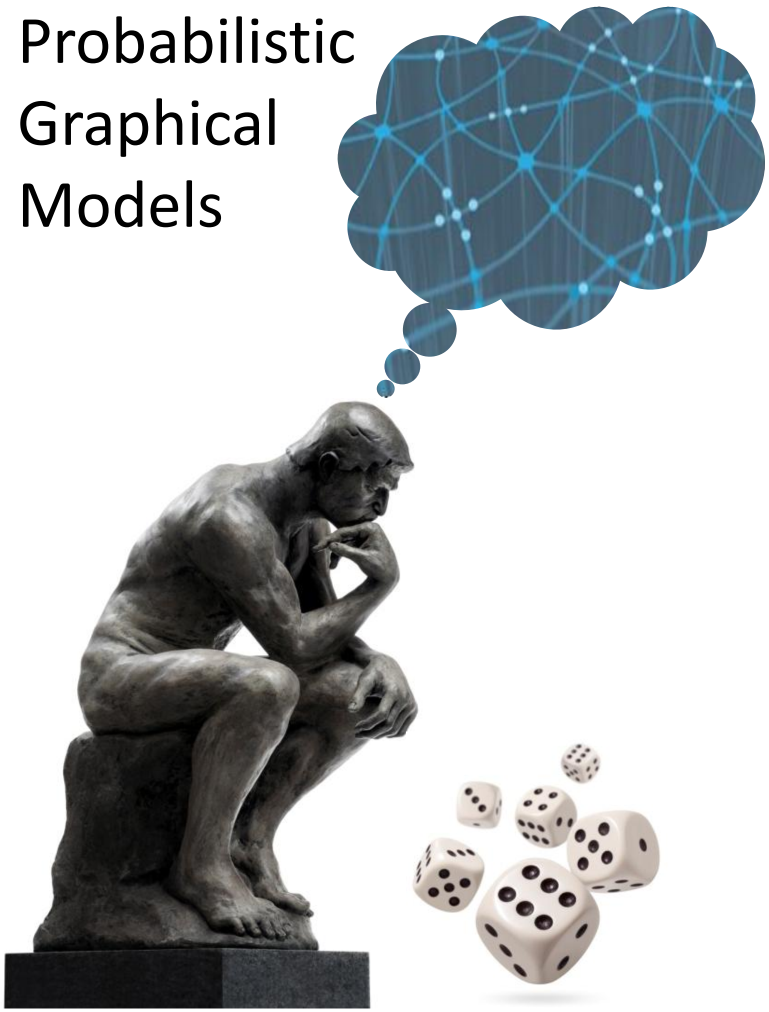
Probabilistic
Graphical
Models



Learning

Parameter Estimation

Max Likelihood for Log-Linear Models

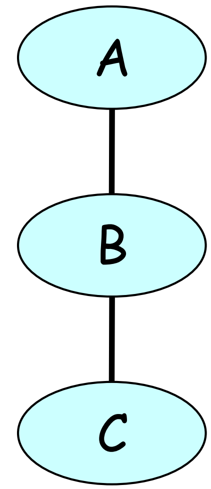
Log-Likelihood for Markov Nets

$P_\Phi(a, b, c) = \frac{1}{Z} \phi_1(a,b) \cdot \phi_2(b,c)$

$$\ell(\boldsymbol{\theta} : \mathcal{D}) = \sum_m \left(\ln \phi_1(a[m], b[m]) + \ln \phi_2(b[m], c[m]) - \ln Z(\boldsymbol{\theta}) \right)$$

$$= \sum_{a,b} \boxed{M[a,b]} \ln \phi_1(a,b) + \sum_{b,c} \boxed{M[b,c]} \ln \phi_2(b,c) - M \ln Z(\boldsymbol{\theta})$$

$$\boxed{Z(\boldsymbol{\theta}) = \sum_{a,b,c} \phi_1(a,b)\phi_2(b,c)}$$

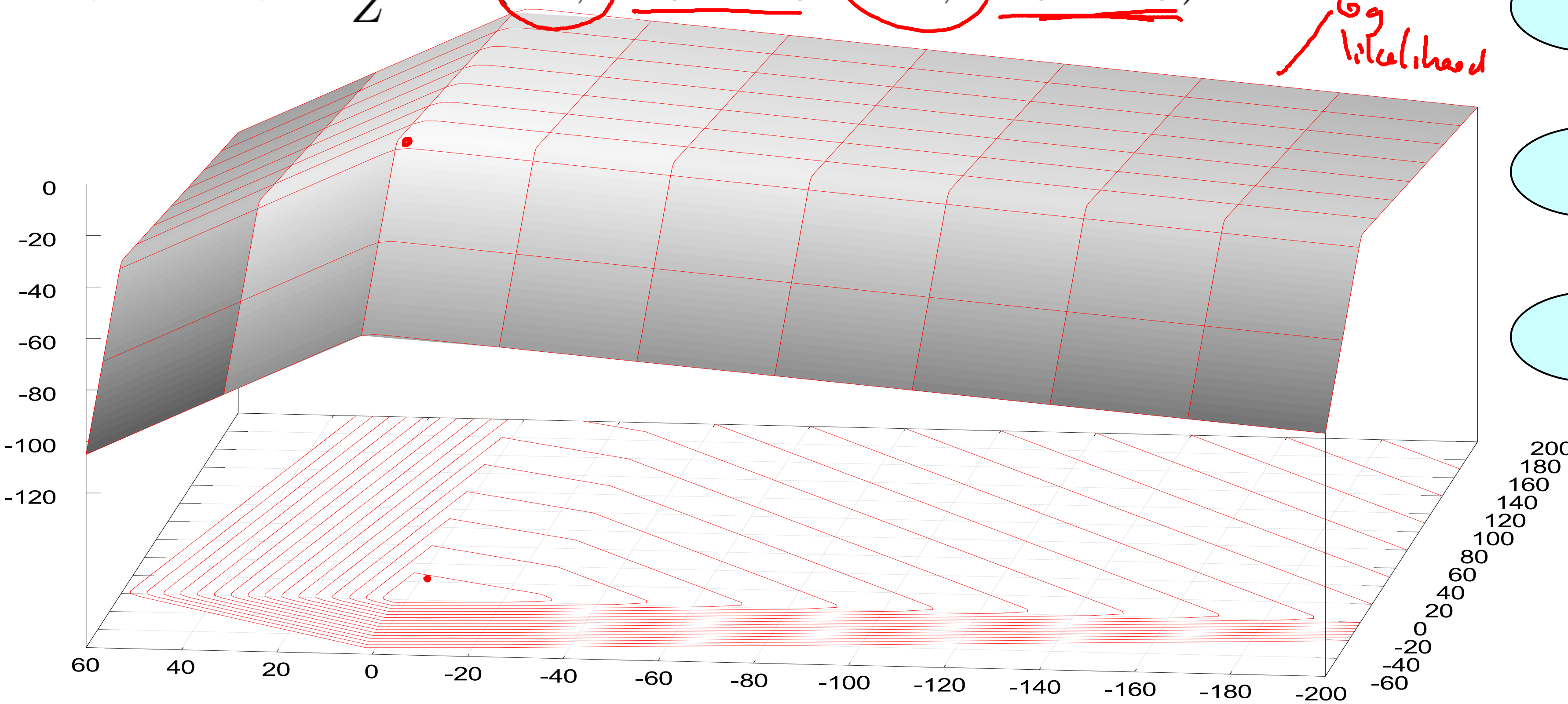


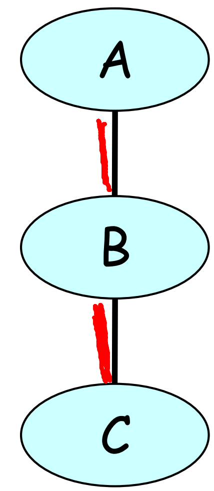
- Partition function couples the parameters
  - No decomposition of likelihood
  - No closed form solution

Example: Log-Likelihood Function

$$P_\Phi(A,B,C) = \frac{1}{Z}\exp\left(\theta_{a^1,b^1}\,\mathbb{1}\{a^1,b^1\} + \theta_{b^0,c^1}\,\mathbb{1}\{b^0,c^1\}\right)$$

log likelihood





Daphne Koller

Log-Likelihood for Log-Linear Model

parameters

features

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

partition function

$$\ell(\boldsymbol{\theta} : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(\boldsymbol{x}[m]) \right) - M \ln Z(\boldsymbol{\theta})$$

feature f_i applied to the m'th instance

$$\ln Z(\boldsymbol{\theta}) = \ln \sum_{\boldsymbol{x}} \exp\left\{ \sum_i \theta_i f_i(\boldsymbol{x}) \right\}$$

log-sum-exp

exponentially large space

Daphne Koller

The Log-Partition Function

Theorem:

vector of derivatives

$$\frac{\partial}{\partial \theta_i} \ln Z(\boldsymbol{\theta}) = \boldsymbol{E}_{\boldsymbol{\theta}}[f_i]$$

expectation of f_i relative P_θ $\sum_x P_\theta(x) f_i(x)$

matrix (Hessian)

$$\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln Z(\boldsymbol{\theta}) = \boldsymbol{C}ov_{\boldsymbol{\theta}}[f_i; f_j]$$

$$\frac{\partial}{\partial \theta_i} \theta_j f_j = \begin{cases} 0 & i \neq j \\ f_i & i = j \end{cases}$$

Proof:

$$\frac{\partial}{\partial \theta_i} \ln Z(\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \sum_{\boldsymbol{x}} \frac{\partial}{\partial \theta_i} \exp \left\{ \sum_j \theta_j f_j(\boldsymbol{x}) \right\}$$

$$= \frac{1}{Z(\boldsymbol{\theta})} \sum_{\boldsymbol{x}} f_i(\boldsymbol{x}) \exp \left\{ \sum_j \theta_j f_j(\boldsymbol{x}) \right\}$$

$$= \sum_{\boldsymbol{x}} \boxed{\frac{1}{Z(\boldsymbol{\theta})} \exp \left\{ \sum_j \theta_j f_j(\boldsymbol{x}) \right\}} f_i(\boldsymbol{x}) = \sum_{\boldsymbol{x}} P_{\boldsymbol{\theta}}(\boldsymbol{x}) f_i(\boldsymbol{x})$$

Daphne Koller

The Log-Partition Function

Theorem:

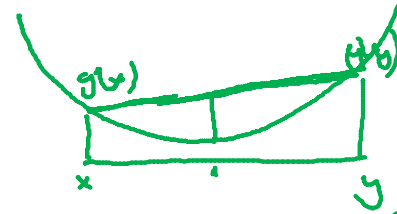
$$\frac{\partial}{\partial \theta_i} \ln Z(\boldsymbol{\theta}) = \boldsymbol{E}_{\boldsymbol{\theta}}[f_i]$$

Hessian

$$\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln Z(\boldsymbol{\theta}) = \boldsymbol{C}ov_{\boldsymbol{\theta}}[f_i; f_j]$$

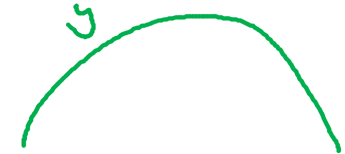
Convex

$g(\alpha x + (1-\alpha) y) \leq \alpha g(x) + (1-\alpha) g(y)$



$$\ell(\boldsymbol{\theta} : \mathcal{D}) = \underbrace{\sum_i \theta_i \left(\sum_m f_i(\boldsymbol{x}[m]) \right)}_{\text{linear in } \theta} - \underline{M \ln Z(\boldsymbol{\theta})}$$

concave



$\ell(\theta : D)$ concave function

- Log likelihood function
  - No local optima
  - Easy to optimize

Maximum Likelihood Estimation

$$\frac{1}{M}\ell(\boldsymbol{\theta}:\mathcal{D}) = \sum_i \theta_i \Bigg(\underbrace{\frac{1}{M}\sum_m f_i(\boldsymbol{x}[m])}_{\text{empirical expectation of } f_i \text{ in } D}\Bigg) - \ln Z(\boldsymbol{\theta})$$

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

expectation of f_i in P_θ

Theorem: $\hat{\boldsymbol{\theta}}$ is the MLE if and only if

$\forall i$

$$\boldsymbol{E}_{\mathcal{D}}[f_i(\boldsymbol{X})] = \boldsymbol{E}_{\hat{\boldsymbol{\theta}}}[f_i]$$

expectation in D = expectation relative to θ

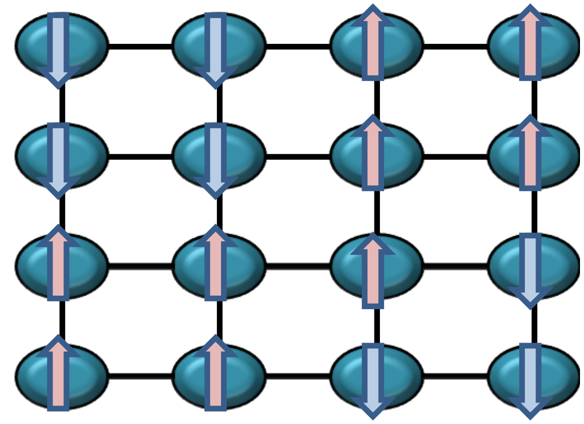
Daphne Koller

Computation: Gradient Ascent

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

- Use gradient ascent:
  - typically L-BFGS - a quasi-Newton method
- For gradient, need expected feature counts:
  - in data
  - relative to current model
- Requires inference at each gradient step

Example: Ising Model



$$E(x_1, \ldots, x_n) = -\sum_{i<j} w_{i,j} \boxed{x_i x_j} - \sum_i u_i x_i$$

$$x_i \in \{-1, +1\}$$

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

$$\frac{\partial}{\partial u_i} = \frac{1}{M} \sum_m x_i[m] - (\underbrace{P_{\boldsymbol{\theta}}(X_i = 1)}_{+1} - \underbrace{P_{\boldsymbol{\theta}}(X_i = -1)}_{-1}))$$

$$\frac{\partial}{\partial w_{ij}} = \frac{1}{M} \sum_m x_i[m] x_j[m] - \begin{pmatrix} P_{\boldsymbol{\theta}}(X_i = 1, X_j = 1) + P_{\boldsymbol{\theta}}(X_i = -1, X_j = -1) \\ -P_{\boldsymbol{\theta}}(X_i = 1, X_j = -1) - P_{\boldsymbol{\theta}}(X_i = -1, X_j = -1) \end{pmatrix}$$

Daphne Koller

Summary

- Partition function couples parameters in likelihood
- No closed form solution, but convex optimization
  – Solved using gradient ascent (usually L-BFGS) global opt.
- Gradient computation requires inference at each gradient step to compute expected feature counts
- Features are always within clusters in cluster-graph or clique tree due to family preservation
  – One calibration suffices for all feature expectations