ECE 6504: Advanced Topics in Machine Learning

Probabilistic Graphical Models and Large-Scale Learning

Topics
- Markov Random Fields
  - (Finish) Inference
  - (Start) Parameter Learning

Readings: KF 20.1-3, Barber 9.6

Dhruv Batra

Virginia Tech

Administrativia

- HW2
  - Solutions released

- Project Presentations
  - When: April 22, 24
  - Where: in class
  - 5 min talk
    - Main results
    - Semester completion 2 weeks out from that point so nearly finished results expected
    - Slides due: April 21 11:55pm

Recap of Last Time

MAP in Pairwise MRFs

- Integer Program

$$\max_{\boldsymbol{\mu}} \quad \boldsymbol{\theta}^T \boldsymbol{\mu}$$

$$\mu_i(s) \in \{0,1\}$$

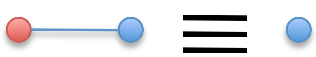
$$\mu_{ij}(s,t) \in \{0,1\}$$

Indicator Variables

$$\sum_s \mu_i(s) = 1$$

$$\sum_{s,t} \mu_{i,j}(s,t) = 1$$

Unique Label



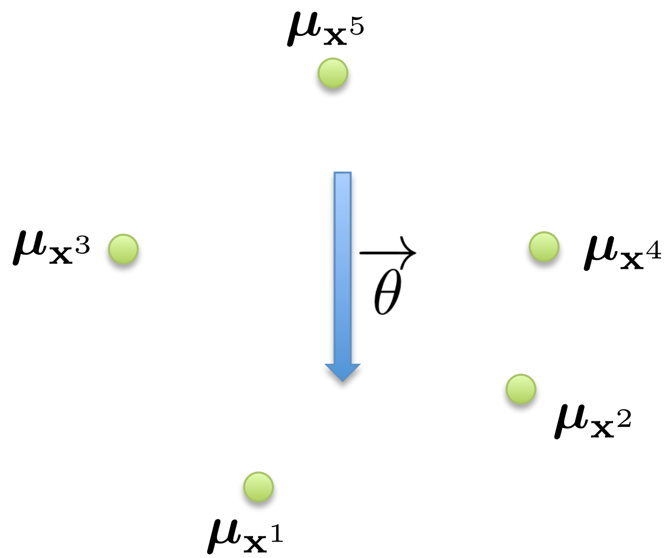
$$\sum_s \mu_{ij}(s,t) = \mu_j(t)$$

Consistent Assignments

MAP in Pairwise MRFs

- MAP Integer Program

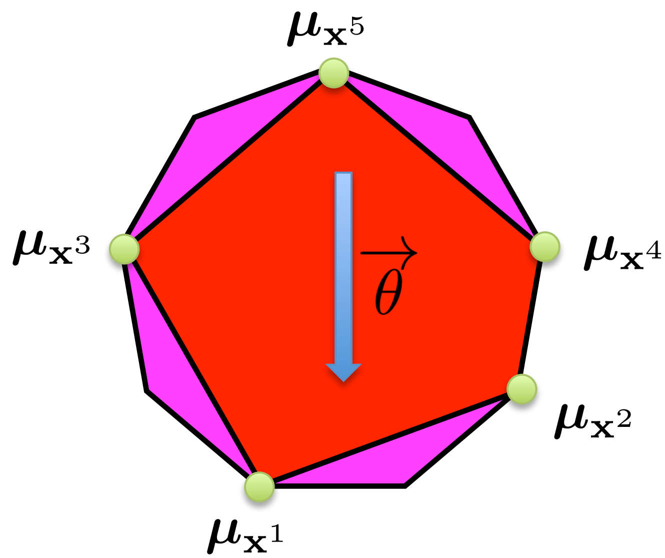
$$
\begin{aligned}
\max_{\boldsymbol{\mu}} \quad & \boldsymbol{\theta}^T \boldsymbol{\mu} \\
s.t. \quad & A\boldsymbol{\mu} = b \\
& \boldsymbol{\mu}(\cdot) \in \{0, 1\}
\end{aligned}
$$



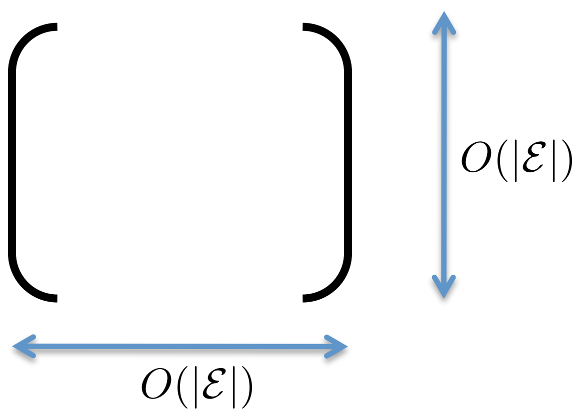
MAP in Pairwise MRFs

- MAP Linear Program

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \boldsymbol{\theta}^T \boldsymbol{\mu} \\ s.t. \quad & A\boldsymbol{\mu} = b \\ & \boldsymbol{\mu}(\cdot) \in [0, 1] \end{aligned}$$



$A =$



Off-the-shelf solvers
CPLEX
Mosek
etc

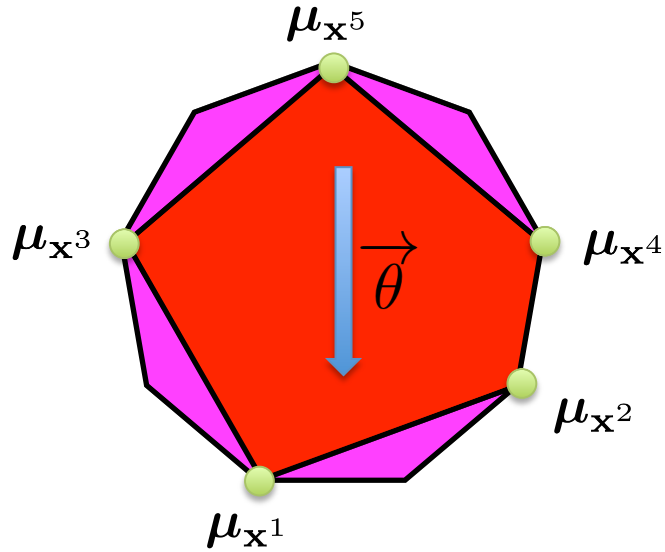
MAP in Pairwise MRFs

- MAP Linear Program

$$\max_{\boldsymbol{\mu}} \quad \boldsymbol{\theta}^T \boldsymbol{\mu}$$

$$s.t. \quad A\boldsymbol{\mu} = b$$

$$\boldsymbol{\mu}(\cdot) \in [0,1]$$



- Properties
  - If LP-opt is integral, MAP is found
  - LP always integral for trees
  - Efficient message-passing schemes for solving LP

Linear Programming Duality

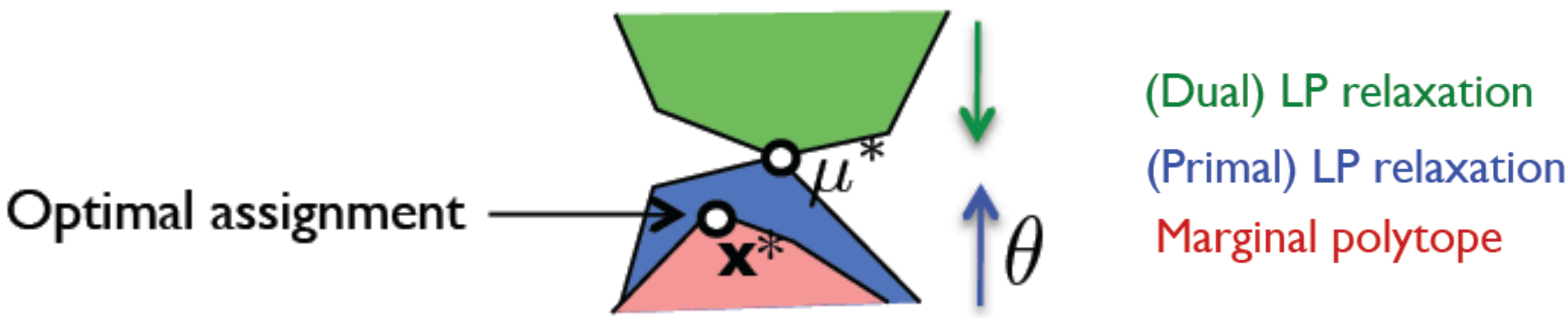
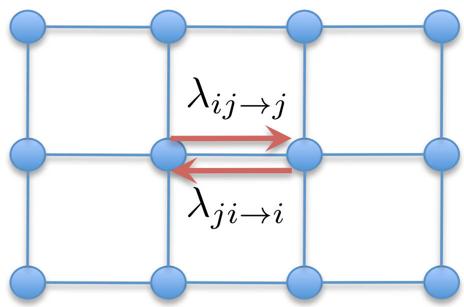
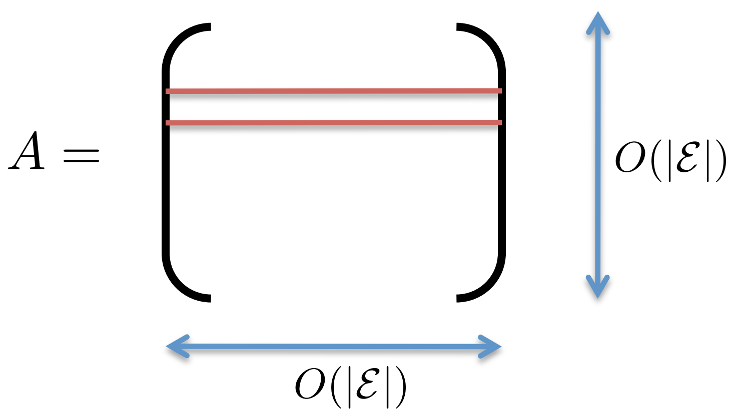


Figure Credit: David Sontag

LP Relaxation

- Block Co-ordinate / Sub-gradient Descent on Dual





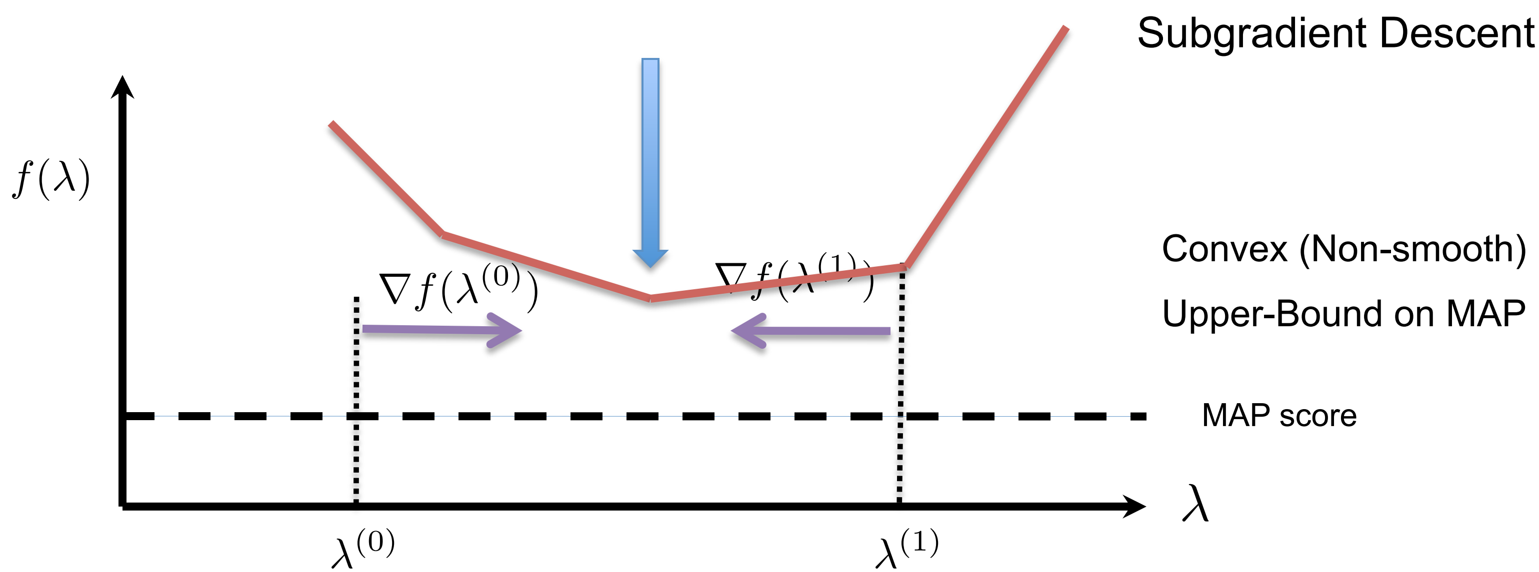
MAP LP

- Lagrangian Relaxation

$$f(\lambda) = \max_{\boldsymbol{\mu} \in \mathcal{C}} \sum_i \boldsymbol{\theta}_i \cdot \boldsymbol{\mu}_i + \sum_{(i,j)} \boldsymbol{\theta}_{ij} \cdot \boldsymbol{\mu}_{ij} \quad -\lambda \cdot (A\boldsymbol{\mu} - b)$$

$$s.t. \quad \boldsymbol{\mu}_i(\cdot), \boldsymbol{\mu}_{ij}(\cdot) \in \{0, 1\}$$

Dual $\min_{\lambda \geq 0} \quad f(\lambda)$



Plan for today

- MRF Inference
  - (Specialized) MAP Inference
    - Dual Decomposition
      - As a general algorithm
- MRF Parameter Learning
  - MLE

Dual Decomposition

- Primal problem

$$\max_{\mathbf{x}} \quad \sum_i f_i(\mathbf{x})$$
$$s.t. \quad \mathbf{x} \in \mathcal{D}$$

- Re-formulate

$$\max_{\mathbf{x},\mathbf{x}^i} \quad \sum_i f_i(\mathbf{x}^i)$$
$$s.t. \quad \mathbf{x}^i \in \mathcal{D}, \mathbf{x}^i = \mathbf{x}$$

- Langragian Relaxation

$$L\big(\{\mathbf{x}^i, \mathbf{x}\}, \{\boldsymbol{\lambda}^i\}\big) = \sum_i f_i(\mathbf{x}^i) - \sum_i \boldsymbol{\lambda}^i \cdot (\mathbf{x} - \mathbf{x}^i)$$

$$g(\{\boldsymbol{\lambda}^i\}) = \max_{\mathbf{x}^i \in \mathcal{D}, \mathbf{x}} \quad \sum_i f_i(\mathbf{x}^i) - \sum_i \boldsymbol{\lambda}^i \cdot (\mathbf{x} - \mathbf{x}^i)$$

Komodakis ICCV '07; Bertsekas '99

Dual Decomposition

- Dual (Master) Problem

$$\min_{\{\boldsymbol{\lambda}^i\} \in \Lambda} \quad \sum_i g_i(\boldsymbol{\lambda}^i)$$

- Dual (Slave) Problems

$$g_i(\boldsymbol{\lambda}^i) = \max_{\mathbf{x}^i} \; f_i(\mathbf{x}^i) + \boldsymbol{\lambda}^i \cdot \mathbf{x}^i$$
$$s.t. \;\; \mathbf{x}^i \in \mathcal{D}$$

- Solve master with (projected) Subgradient Descent

$$\boldsymbol{\lambda}^i \longleftarrow \left[\boldsymbol{\lambda}^i - \alpha_t \nabla g^i(\boldsymbol{\lambda}^i) \right]_{\Lambda}$$

Dual Decomposition

- Projected Subgradient Ascent

$$\boldsymbol{\lambda}^i \longleftarrow \left[\boldsymbol{\lambda}^i - \alpha_t \nabla g^i(\boldsymbol{\lambda}^i)\right]_\Lambda$$

- What is subgradient?

$$\begin{aligned} \nabla g_i(\boldsymbol{\lambda}^i) = \underset{\mathbf{x}^i}{\operatorname{argmax}} \quad & f_i(\mathbf{x}^i) + \boldsymbol{\lambda}^i \cdot \mathbf{x}^i \\ s.t. \quad & \mathbf{x}^i \in \mathcal{D} \end{aligned}$$

Main Issues in PGMs

- Representation
  - How do we store $P(X_1, X_2, \ldots, X_n)$
  - What does my model mean/imply/assume? (Semantics)

- Inference
  - How do I answer questions/queries with my model? such as
  - Marginal Estimation: $P(X_5 \mid X_1, X_4)$
  - Most Probable Explanation: argmax $P(X_1, X_2, \ldots, X_n)$

- Learning
  - How do we learn parameters and structure of $P(X_1, X_2, \ldots, X_n)$ from data?
  - What model is the right for my data?

Recall -- Learning Bayes Nets

True Distribution P*
(Maybe corresponds to a BN G*
maybe not)

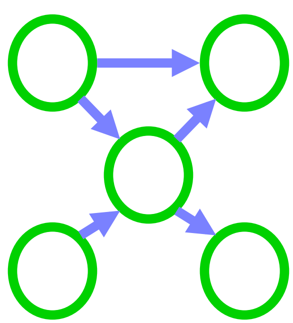
Domain Experts

Data

$\mathbf{x}^{(1)}$

…

$\mathbf{x}^{(m)}$



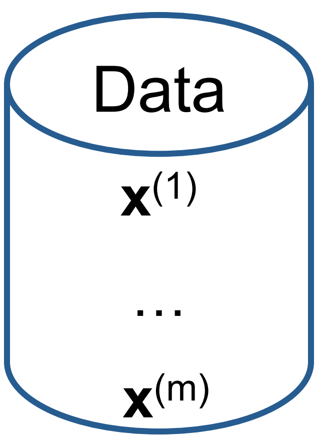
+

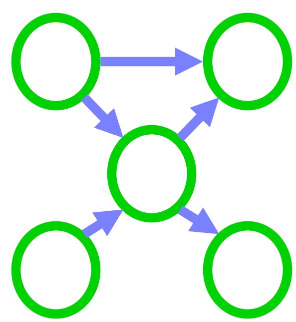
CPTs –

$P(X_i | \mathbf{Pa}_{X_i})$

Learning Bayes Nets

	Known structure	Unknown structure
Fully observable data	Very easy	Hard
Missing data	Somewhat easy (EM)	Very very hard





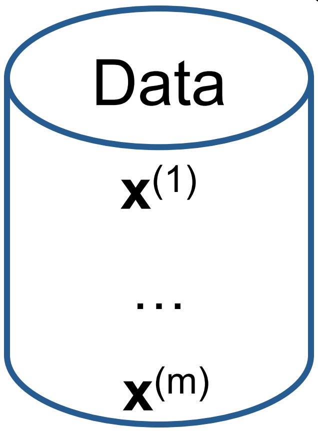
structure

+

CPTs – $P(X_i \mid \mathbf{Pa}_{X_i})$

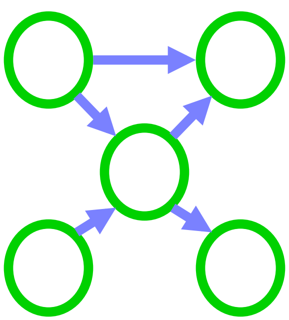
parameters

Learning the CPTs



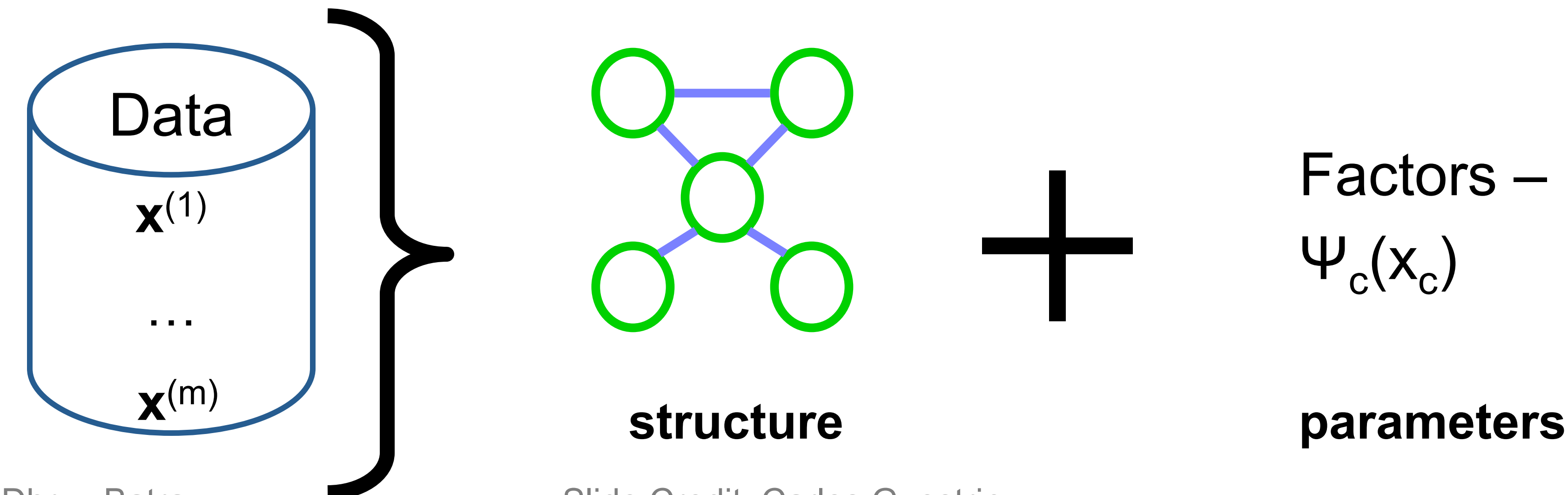
For each discrete variable X_i

$$\hat{P}_{MLE}(X_i = a \mid \mathrm{Pa}_{X_i} = b) = \frac{\mathrm{Count}(X_i = a, \mathrm{Pa}_{X_i} = b)}{\mathrm{Count}(\mathrm{Pa}_{X_i} = b)}$$



Learning Markov Nets

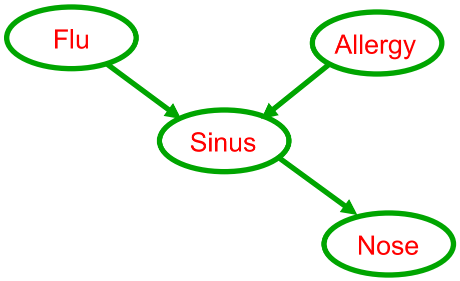
	Known structure	Unknown structure
Fully observable data	NP-Hard (but doable)	Harder
Missing data	Harder (EM)	Don't try this at home



Learning Parameters of a BN

- Log likelihood decomposes:

$$\log P(\mathcal{D} \mid \theta) = m \sum_i \sum_{x_i, \mathbf{Pa}_{x_i}} \widehat{P}(x_i, \mathbf{Pa}_{x_i}) \log P(x_i \mid \mathbf{Pa}_{x_i})$$



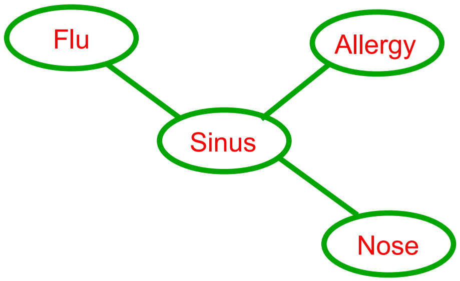
- Learn each CPT independently

$$\widehat{P}(\mathbf{u}) = \frac{\text{Count}(\mathbf{U} = \mathbf{u})}{m}$$

- Use counts

Log Likelihood for MN

- Log likelihood decomposes:



$$\log P(\mathcal{D} \mid \theta, \mathcal{G}) = m \sum_i \sum_{\mathbf{c}_i} \widehat{P}(\mathbf{c}_i) \log \psi_i(\mathbf{c}_i) - m \log Z$$

- Doesn't decompose!
  - logZ couples all parameters together

Log-linear Markov network (most common representation)

- **Feature (or Sufficient Statistic)** is some function ϕ [**D**] for some subset of variables **D**
  - e.g., indicator function

- **Log-linear model** over a Markov network *H*:
  - a set of features $\phi_1[\mathbf{D}_1],\ldots, \phi_k[\mathbf{D}_k]$
    - each $\mathbf{D}_i$ is a subset of a clique in *H*
    - two ϕ's can be over the same variables
  - a set of weights $w_1,\ldots,w_k$
    - usually learned from data
  - $$P(X_1,\ldots,X_n) = \frac{1}{Z}\exp\left[\sum_{i=1}^{k} w_i\phi_i\left(\mathbf{D}_i\right)\right]$$

Learning params for log linear models – Gradient Ascent

$$P(X_1, \ldots, X_n) = \frac{1}{Z} \exp \left[\sum_{i=1}^{k} w_i \phi_i \left(\mathbf{D}_i \right) \right]$$

- Log-likelihood of data:

$$\log P(\mathcal{D} \mid \mathbf{w}, \mathcal{G}) = \sum_{j=1}^{m} \log \frac{1}{Z} \exp \left[\sum_{i=1}^{k} w_i \phi_i (\mathbf{d}_i^{(j)}) \right]$$

- Compute derivative & optimize
  - usually with gradient ascent or L-BFGS

$$\frac{\partial \ell(\mathcal{D} : \mathbf{w})}{\partial w_i} = m \sum_{\mathbf{d}_i} \hat{P}(\mathbf{d}_i) \phi_i(\mathbf{d}_i) - m \frac{\partial \log Z}{\partial w_i}$$

Learning log-linear models with gradient ascent

- Gradient:

$$\frac{\partial \ell(\mathcal{D}:\mathbf{w})}{\partial w_i} = m\sum_{\mathbf{d}_i}\hat{P}(\mathbf{d}_i)\phi_i(\mathbf{d}_i) - m\sum_{\mathbf{d}_i}P(\mathbf{d}_i \mid \mathbf{w})\phi_i(\mathbf{d}_i)$$

- Requires one inference computation per

- Theorem: **w** is maximum likelihood solution iff
  –

- Usually, must regularize
  – E.g., L_2 regularization on parameters

What you need to know about learning MN parameters

- BN parameter learning easy
- MN parameter learning doesn't decompose!

- Learning requires inference!

- Objective Concave
- Apply gradient ascent iterations to obtain optimal parameters