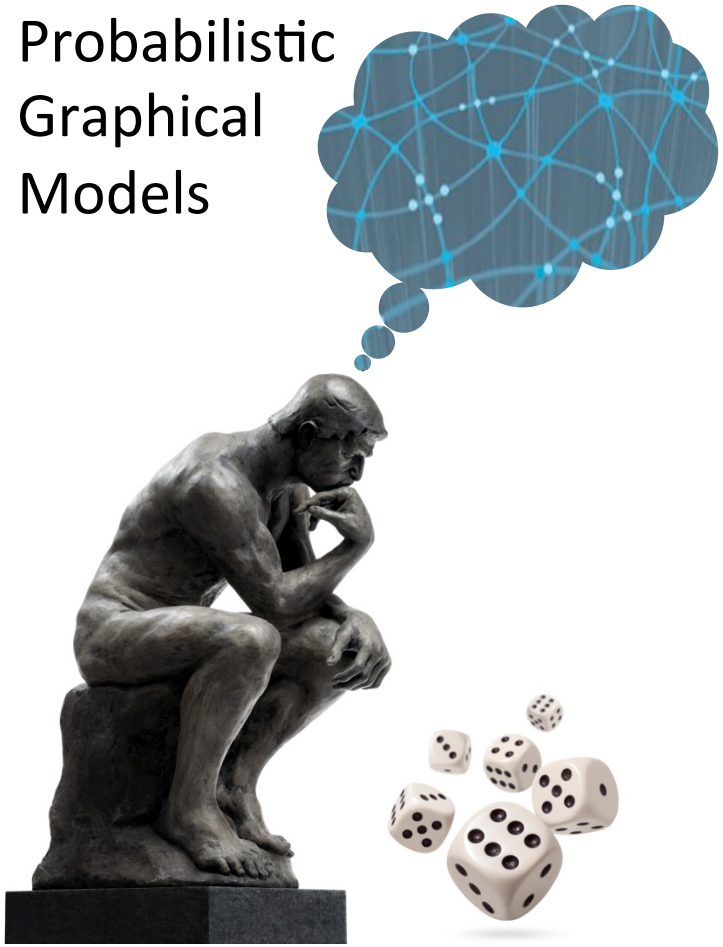
Probabilistic Graphical Models



Inference

MAP

Max-Sum Exact Inference

Product ⇨ Summation

$$P_\Phi(\boldsymbol{x}) \propto \prod_k \phi_k(\boldsymbol{D}_k)$$

$$\operatorname{argmax} \prod_k \phi_k(\boldsymbol{D}_k)$$

$\log \phi_k(D_k)$

$$\operatorname{argmax} \sum_k \underbrace{\theta_k(\boldsymbol{D}_k)}_{\theta(X_1,\ldots,X_n)}$$

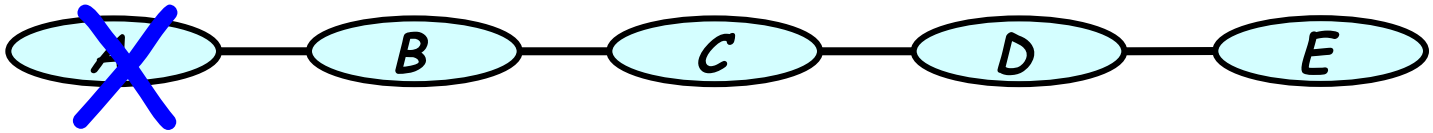
a^1	b^1	8
a^1	b^2	1
a^2	b^1	0.5
a^2	b^2	2

$\log_2$

a^1	b^1	3
a^1	b^2	0
a^2	b^1	-1
a^2	b^2	1

Daphne Koller

Max-Sum Elimination in Chains

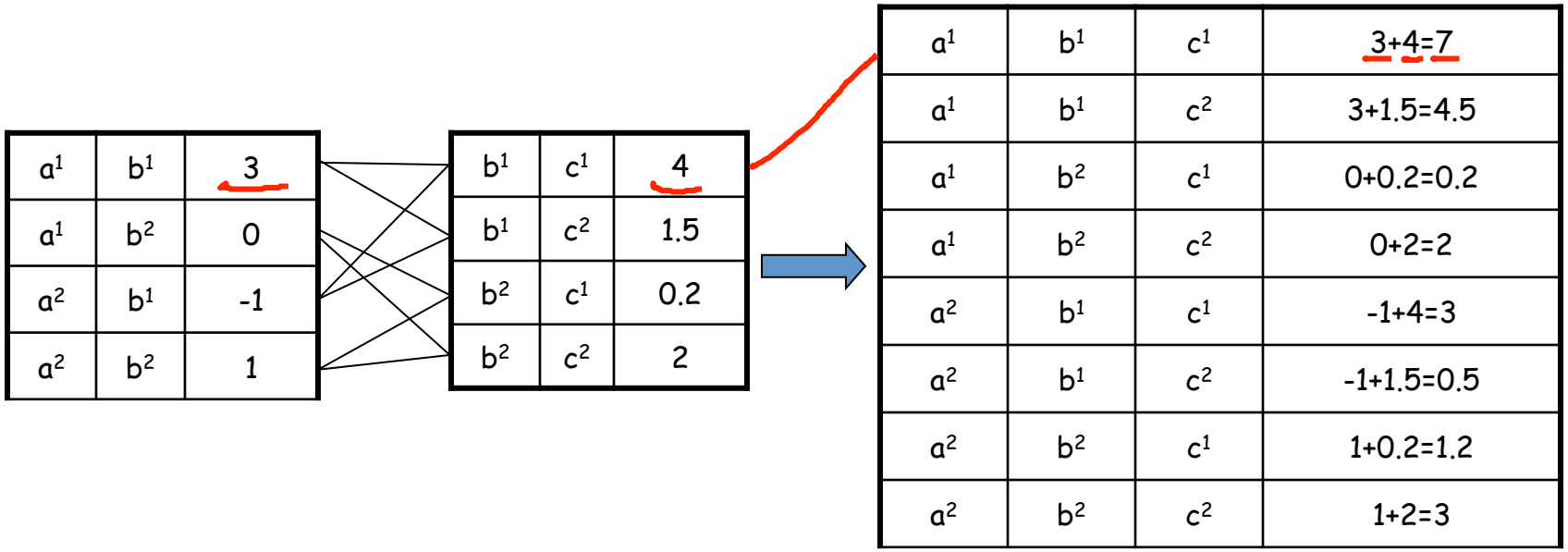


$$\max_D \max_C \max_B \max_A \left(\theta_1(A,B) + \theta_2(B,C) + \theta_3(C,D) + \theta_4(D,E)\right)$$

$$\max_D \max_C \max_B \left(\theta_2(B,C) + \theta_3(C,D) + \theta_4(D,E) + \max_A \theta_1(A,B)\right)$$

$$\max_D \max_C \max_B \left(\theta_2(B,C) + \theta_3(C,D) + \theta_4(D,E) + \lambda_1(B)\right)$$

Factor Summation



a^1	b^1	3
a^1	b^2	0
a^2	b^1	-1
a^2	b^2	1

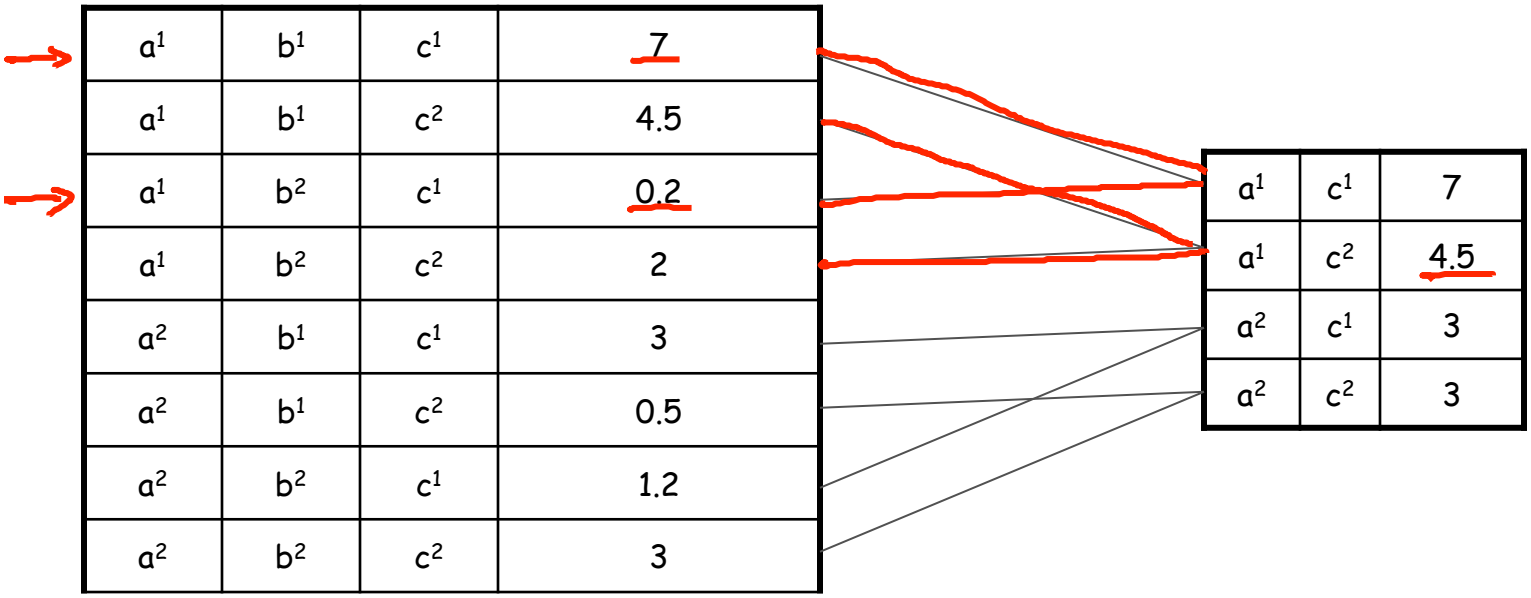
b^1	c^1	4
b^1	c^2	1.5
b^2	c^1	0.2
b^2	c^2	2

a^1	b^1	c^1	3+4=7
a^1	b^1	c^2	3+1.5=4.5
a^1	b^2	c^1	0+0.2=0.2
a^1	b^2	c^2	0+2=2
a^2	b^1	c^1	-1+4=3
a^2	b^1	c^2	-1+1.5=0.5
a^2	b^2	c^1	1+0.2=1.2
a^2	b^2	c^2	1+2=3

Daphne Koller

Factor Maximization

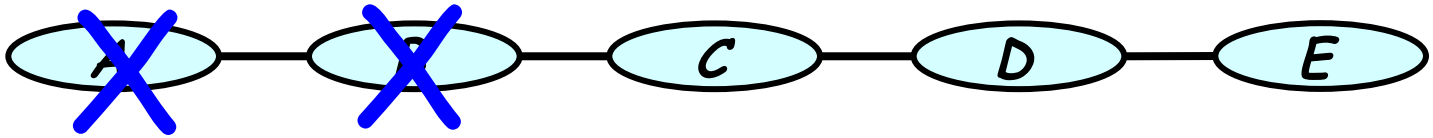
max-marginalization



a^1	b^1	c^1	7
a^1	b^1	c^2	4.5
a^1	b^2	c^1	0.2
a^1	b^2	c^2	2
a^2	b^1	c^1	3
a^2	b^1	c^2	0.5
a^2	b^2	c^1	1.2
a^2	b^2	c^2	3

a^1	c^1	7
a^1	c^2	4.5
a^2	c^1	3
a^2	c^2	3

Max-Sum Elimination in Chains

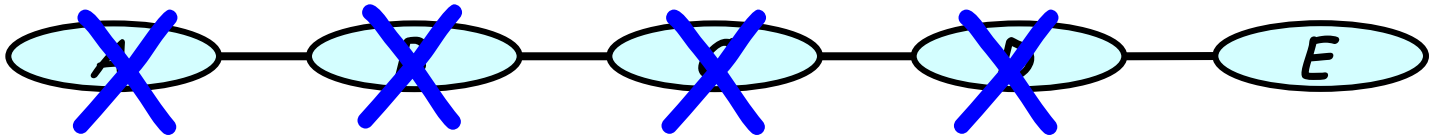


$$\max_D \max_C \max_B \left(\theta_2(B,C) + \theta_3(C,D) + \theta_4(D,E) + \lambda_1(B)\right)$$

$$\max_D \max_C \left(\theta_3(C,D) + \theta_4(D,E) + \max_B \left(\theta_2(B,C) + \lambda_1(B)\right)\right)$$

$$\max_D \max_C \left(\theta_3(C,D) + \theta_4(D,E) + \lambda_2(C)\right)$$

Max-Sum Elimination in Chains



$$\max_D \max_C \left(\theta_3(C,D) + \theta_4(D,E) + \lambda_2(C)\right)$$

$$\max_D \left(\theta_4(D,E) + \lambda_3(D)\right)$$

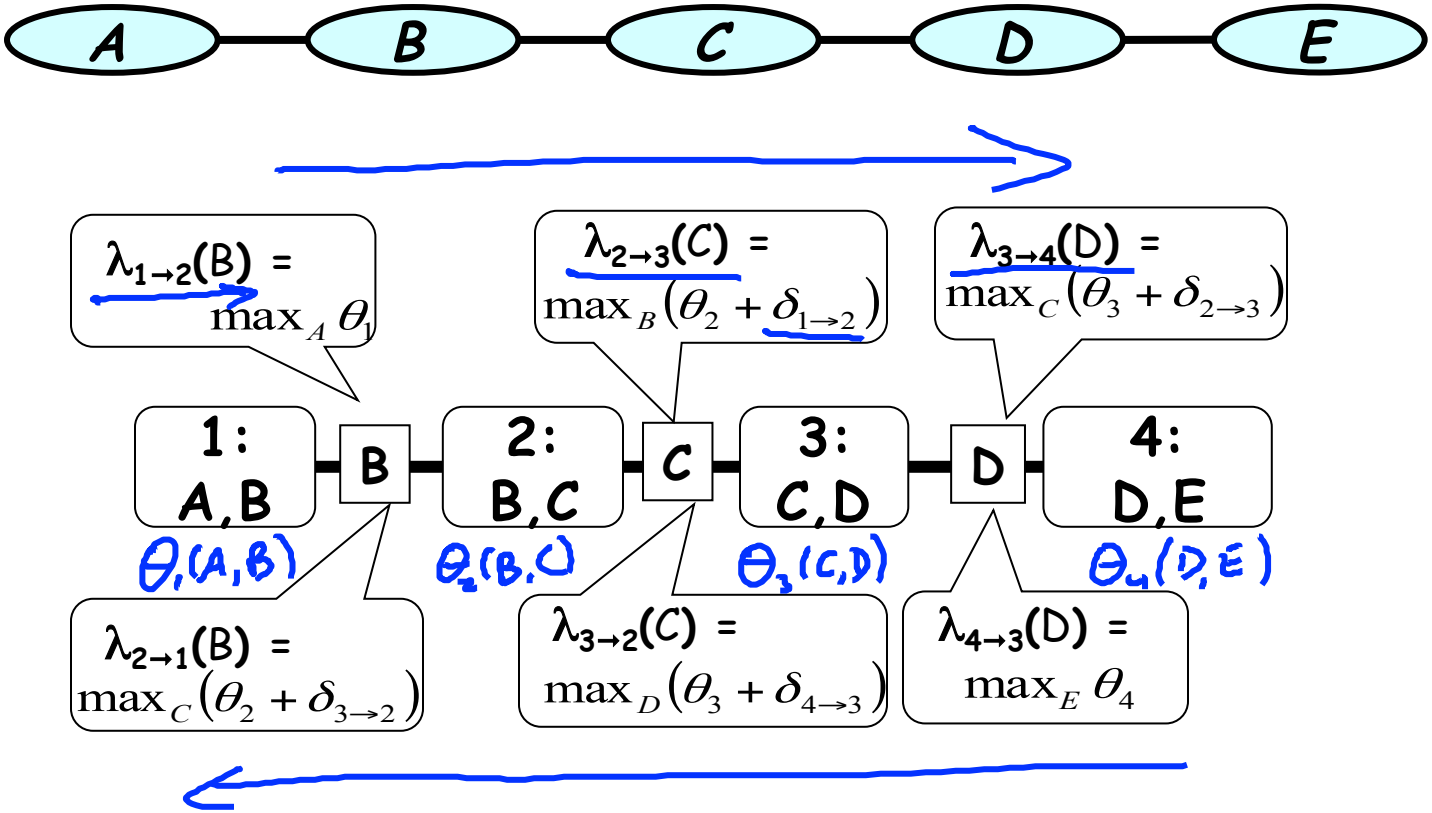
$\lambda_4(E)$

$\lambda_4(e) = \max_{a,b,c,d} \theta(a,b,c,d,e)$

max-marginal

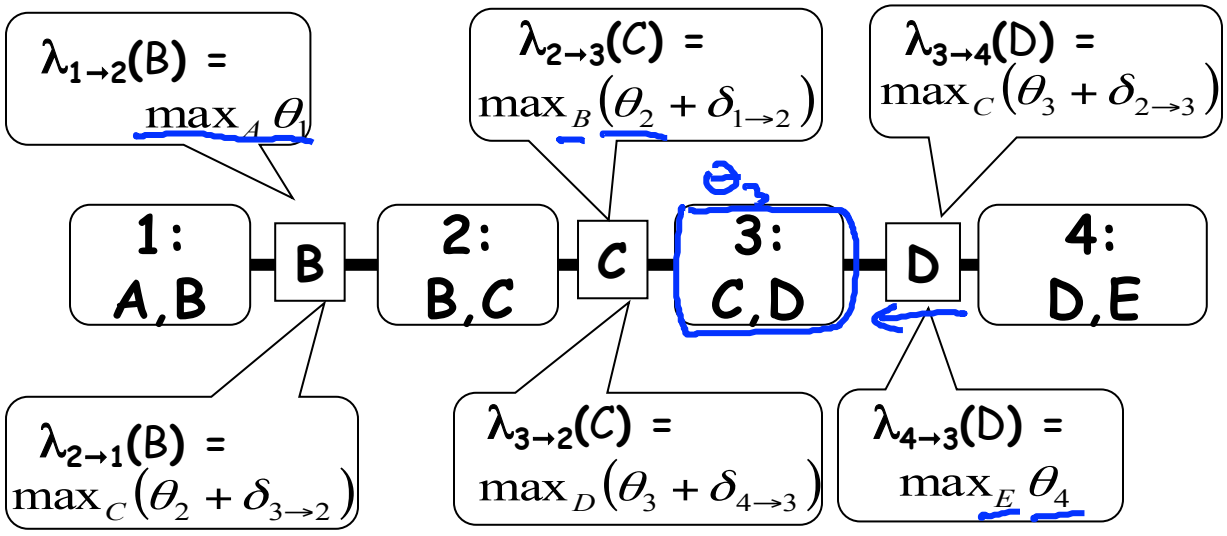
best value that I can get if we mandate $E = e$

Max-Sum in Clique Trees

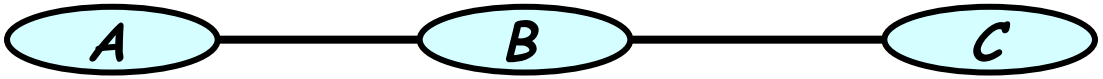


Convergence of Message Passing

- Once C_i receives a final message from all neighbors except C_j, then $\lambda_{i \to j}$ is also final (will never change)
- Messages from leaves are immediately final



Simple Example



$\theta_1(A,B)$

a^1	b^1	3
a^1	b^2	0
a^2	b^1	-1
a^2	b^2	1

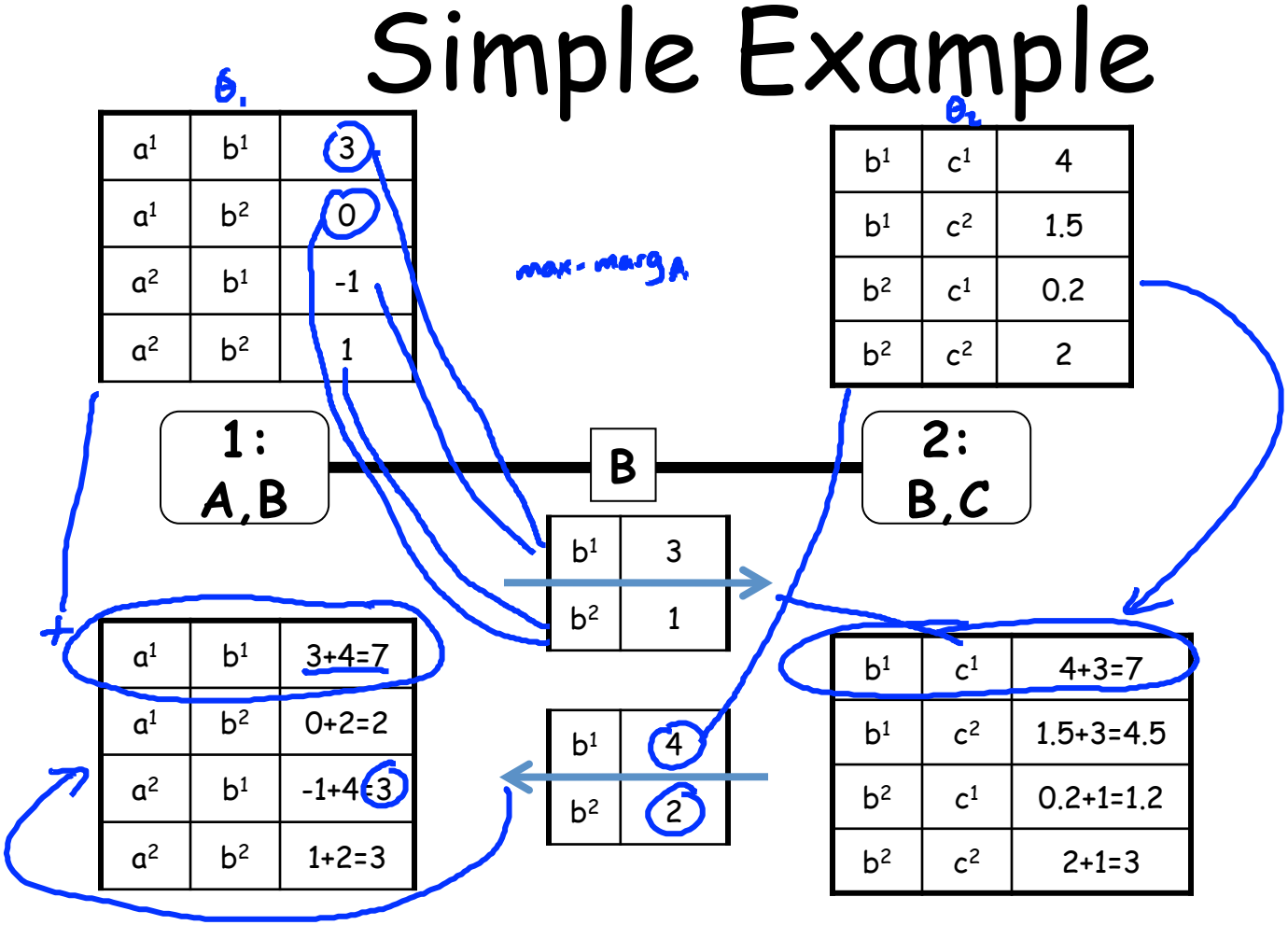
$\theta_2(B,C)$

b^1	c^1	4
b^1	c^2	1.5
b^2	c^1	0.2
b^2	c^2	2

$\theta = \theta_1 + \theta_2$

a^1	b^1	c^1	3+4=7
a^1	b^1	c^2	3+1.5=4.5
a^1	b^2	c^1	0+0.2=0.2
a^1	b^2	c^2	0+2=2
a^2	b^1	c^1	-1+4=3
a^2	b^1	c^2	-1+1.5=0.5
a^2	b^2	c^1	1+0.2=1.2
a^2	b^2	c^2	1+2=3

Simple Example



Max-Sum BP at Convergence

- Beliefs at each clique are max-marginals

$$\beta_i(C_i) = \theta_i(C_i) + \sum_k \lambda_{k \to i}$$

incoming msgs

$$\beta_i(\boldsymbol{C}_i) = \max_{\boldsymbol{W}_i} \theta(\boldsymbol{C}_i, \boldsymbol{W}_i)$$

$$\boldsymbol{W}_i = \{X_1, \ldots, X_n\} - \boldsymbol{C}_i$$

- Calibration: cliques agree on shared variables

$$\max_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i(\boldsymbol{C}_i) = \max_{\boldsymbol{C}_j - \boldsymbol{S}_{i,j}} \beta_j(\boldsymbol{C}_j)$$

$\beta_1 = \theta_1 + \lambda_{2 \to 1}$

a^1	b^1	3+4=7
a^1	b^2	0+2=2
a^2	b^1	-1+4=3
a^2	b^2	1+2=3

b^1 7

b^2 3

b_1 7

b_2 3

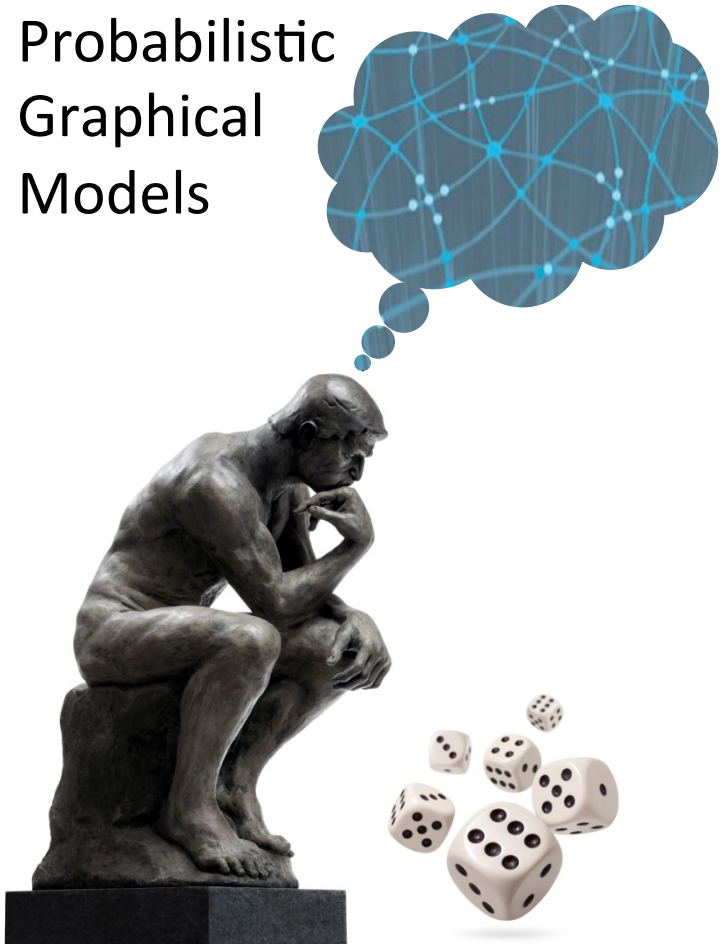
b^1	c^1	4+3=7
b^1	c^2	1.5+3=4.5
b^2	c^1	0.2+1=1.2
b^2	c^2	2+1=3

$\beta_2 = \theta_2 + \lambda_{1 \to 2}$

Summary

- The same clique tree algorithm used for sum-product can be used for max-sum
- As in sum-product, convergence is achieved after a single up-down pass
- Result is a max-marginal at each clique $\mathbf{C}$:
  - For each assignment $\mathbf{c}$ to $\mathbf{C}$, what is the score of the best completion to $\mathbf{c}$

Probabilistic Graphical Models



Inference

MAP

Finding a MAP Assignment

Decoding a MAP Assignment

- Easy if MAP assignment is unique
  - Single maximizing assignment at each clique
  - Whose value is the θ value of the MAP assignment
  - Due to calibration, choices at all cliques must agree

a^1	b^1	c^1	7
a^1	b^1	c^2	4.5
a^1	b^2	c^1	0.2
a^1	b^2	c^2	2
a^2	b^1	c^1	3
a^2	b^1	c^2	0.5
a^2	b^2	c^1	1.2
a^2	b^2	c^2	3

a^1	b^1	3+4=7
a^1	b^2	0+2=2
a^2	b^1	-1+4=3
a^2	b^2	1+2=3

b^1	c^1	4+3=7
b^1	c^2	1.5+3=4.5
b^2	c^1	0.2+1=1.2
b^2	c^2	2+1=3

Decoding a MAP assignment

- If MAP assignment is not unique, we may have multiple choices at some cliques
- Arbitrary tie-breaking may not produce a MAP assignment

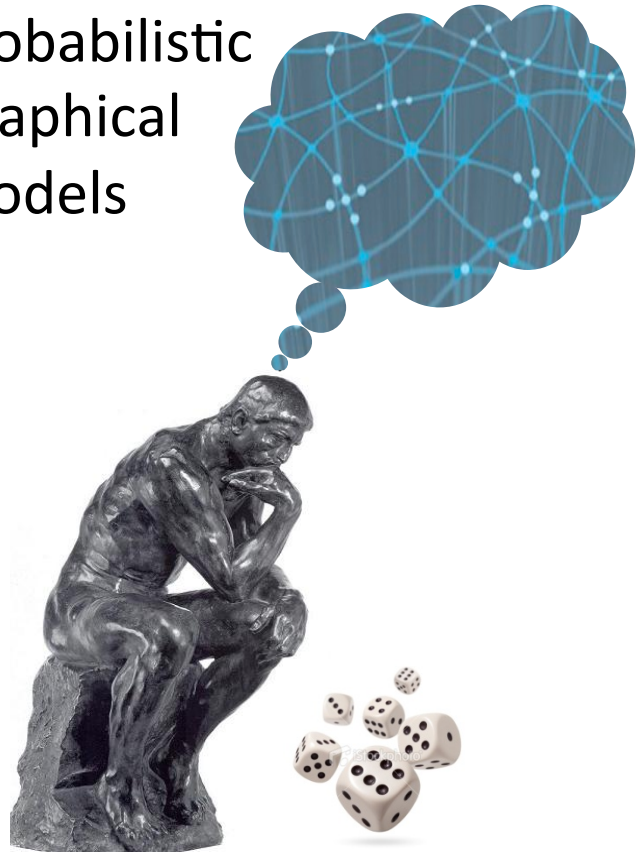
a^1	b^1	2
a^1	b^2	1
a^2	b^1	1
a^2	b^2	2

b^1	c^1	2
b^1	c^2	1
b^2	c^1	1
b^2	c^2	2

Decoding a MAP assignment

- If MAP assignment is not unique, we may have multiple choices at some cliques
- Arbitrary tie-breaking may not produce a MAP assignment
- Two options:
  - Slightly perturb parameters to make MAP unique
  - Use traceback procedure that incrementally builds a MAP assignment, one variable at a time

Probabilistic Graphical Models



Inference

MAP

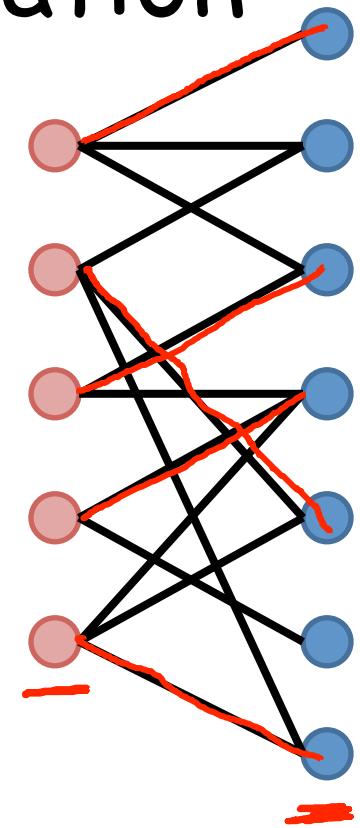
Tractable MAP Problems

Correspondence /data association

X_{ij} = { 1 if i matched to j ; 0 otherwise }

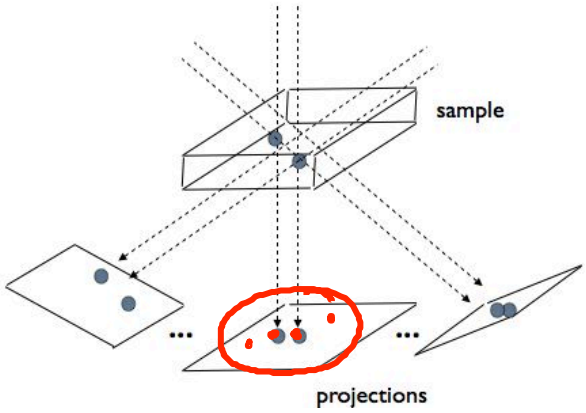
θ_{ij} = quality of "match" between i and j

- Find highest scoring matching
  - maximize $\Sigma_{ij}\ \theta_{ij}\ X_{ij}$
  - subject to mutual exclusion constraint
- Easily solved using matching algorithms
- Many applications
  - matching sensor readings to objects
  - matching features in two related images
  - matching mentions in text to entities

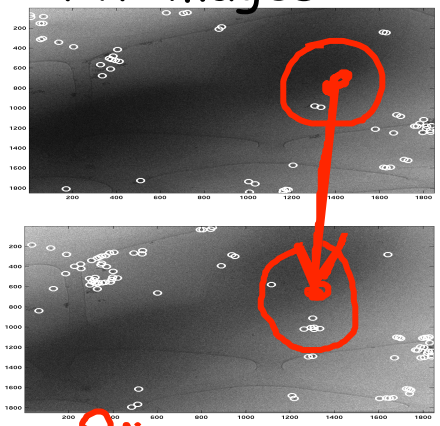


Daphne Koller

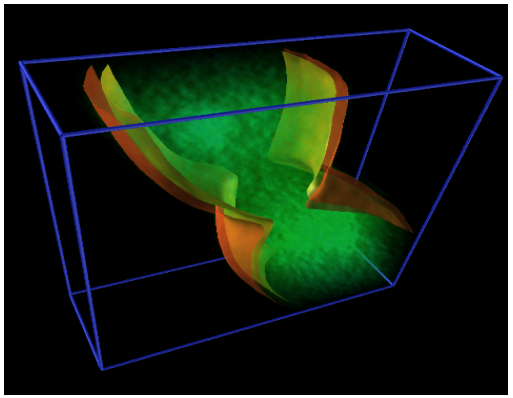
3D Cell Reconstruction



correspond tilt images



compute 3D reconstruction

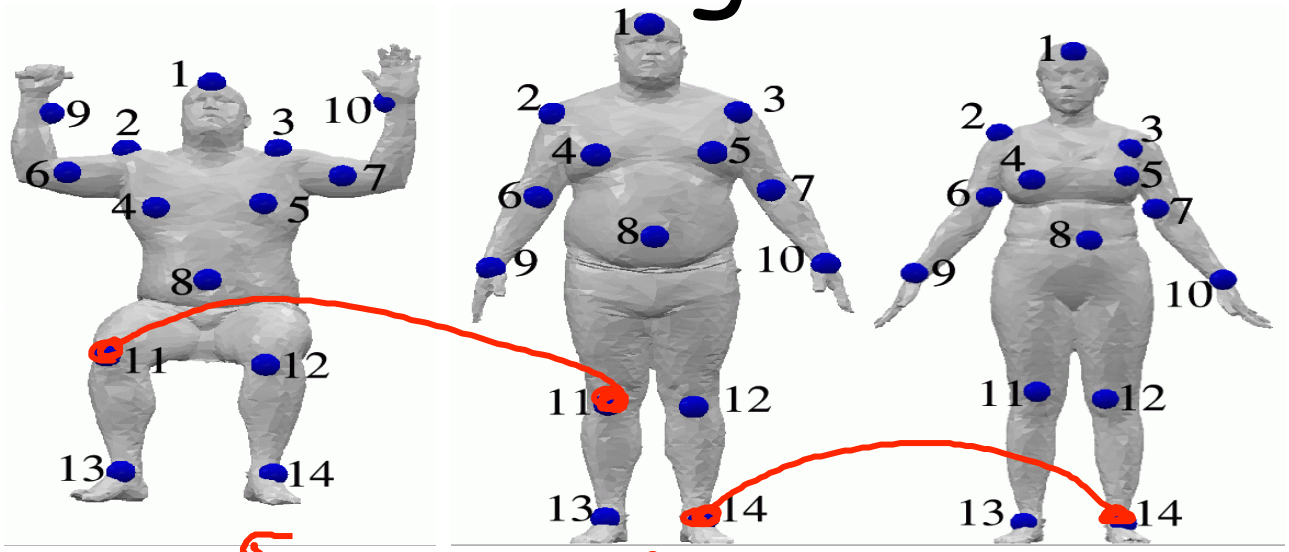


- Matching weights: similarity of location and local neighborhood appearance

Duchi, Tarlow, Elidan, and Koller, NIPS 2006. Amat, Moussavi, Comolli, Elidan, Downing, Horowitz, Journal of Strurctural Biology, 2006.

Daphne Koller

Mesh Registration



- Matching weights: similarity of location and local neighborhood appearance

[Anguelov, Koller, Srinivasan, Thrun, Pang, Davis, NIPS 2004]

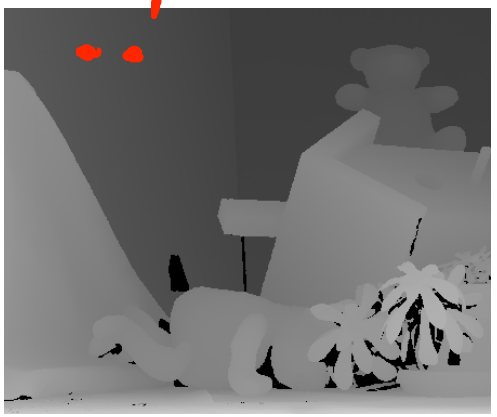
Associative potentials

- Arbitrary network over binary variables using only singleton θ_i and supermodular pairwise potentials θ_{ij}
  - Exact solution using algorithms for finding minimum cuts in graphs
- Many related variants admit efficient exact or approximate solutions
  - Metric MRFs

	0	1
0	a	b
1	c	d

$a + d \geq b + c$

Example: Depth Reconstruction



view 1

view 2

depth reconstruction

denoising, inpainting, FG/BG segmentation

Scharstein & Szeliski, "High-accuracy stereo depth maps using structured light" Proc. IEEE CVPR 2003

Cardinality Factors

- A factor over arbitrarily many binary variables $X_1, \ldots, X_k$
- $\text{Score}(X_1, \ldots, X_k) = f(\Sigma_i X_i)$
- Example applications:
  - soft parity constraints
  - prior on # pixels in a given category
  - prior on # of instances assigned to a given cluster

A	B	C	D	score
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Daphne Koller

Sparse Pattern Factors

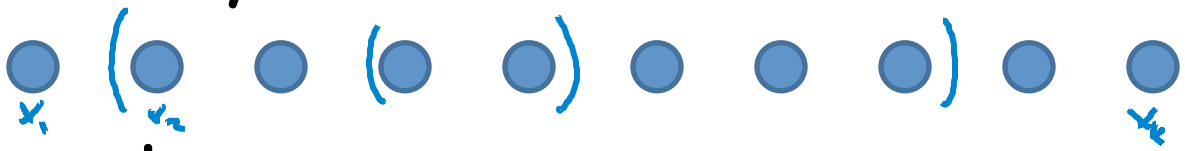
- A factor over variables $X_1,\ldots,X_k$
  - Score($X_1, \ldots,X_k$) specified for some small # of assignments $x_1,\ldots,x_k$
  - Constant for all other assignments
- Examples: give higher score to combinations that occur in real data
  - In spelling, letter combinations that occur in dictionary
  - 5×5 image patches that appear in natural images

A	B	C	D	score
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Daphne Koller

Convexity Factors

- Ordered binary variables $X_1,\ldots,X_k$
- Convexity constraints

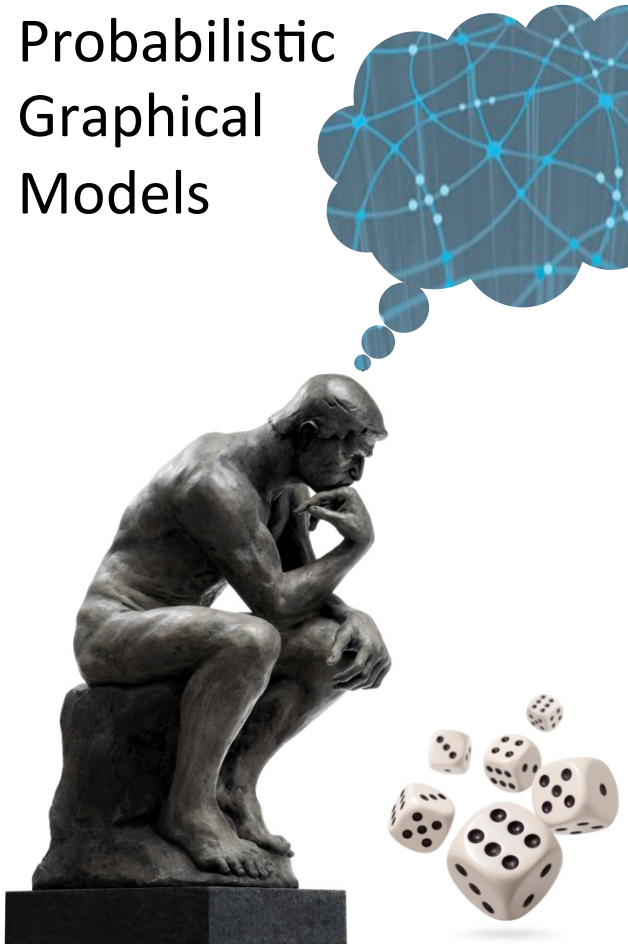


- Examples:
  - Convexity of "parts" in image segmentation
  - Contiguity of word labeling in text
  - Temporal contiguity of subactivities

Summary

- Many specialized models admit tractable MAP solution
  - Many do not have tractable algorithms for computing marginals
- These specialized models are useful
  - On their own
  - As a component in a larger model with other types of factors

Probabilistic Graphical Models



Inference

MAP

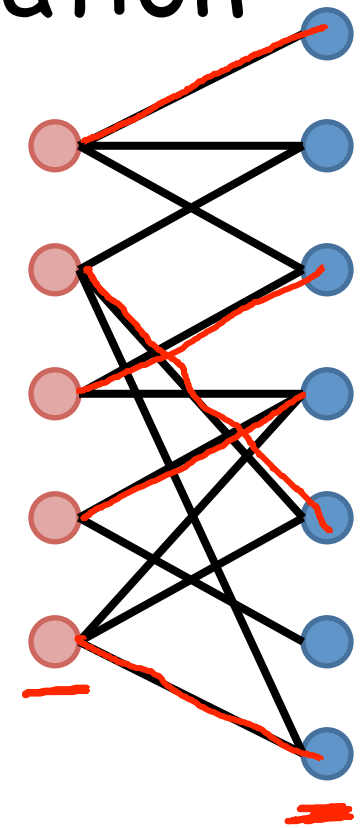
Tractable MAP Problems

Correspondence /data association

X_{ij} = { 1 if i matched to j ; 0 otherwise }

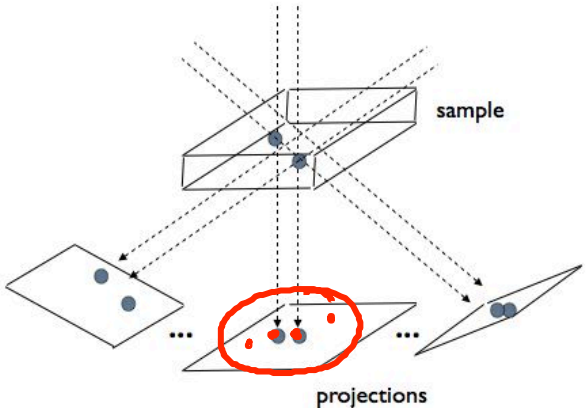
θ_{ij} = quality of "match" between i and j

- Find highest scoring matching
  - maximize $\Sigma_{ij}\ \theta_{ij}\ X_{ij}$
  - subject to mutual exclusion constraint
- Easily solved using matching algorithms
- Many applications
  - matching sensor readings to objects
  - matching features in two related images
  - matching mentions in text to entities

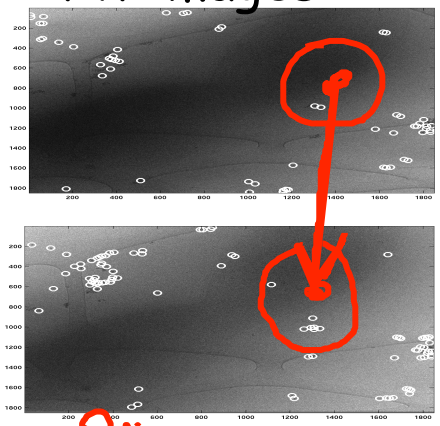


Daphne Koller

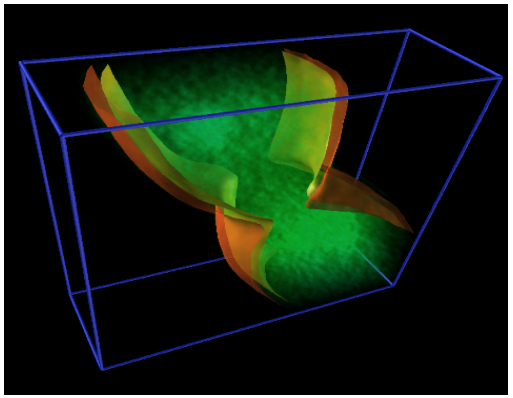
3D Cell Reconstruction



correspond tilt images



compute 3D reconstruction

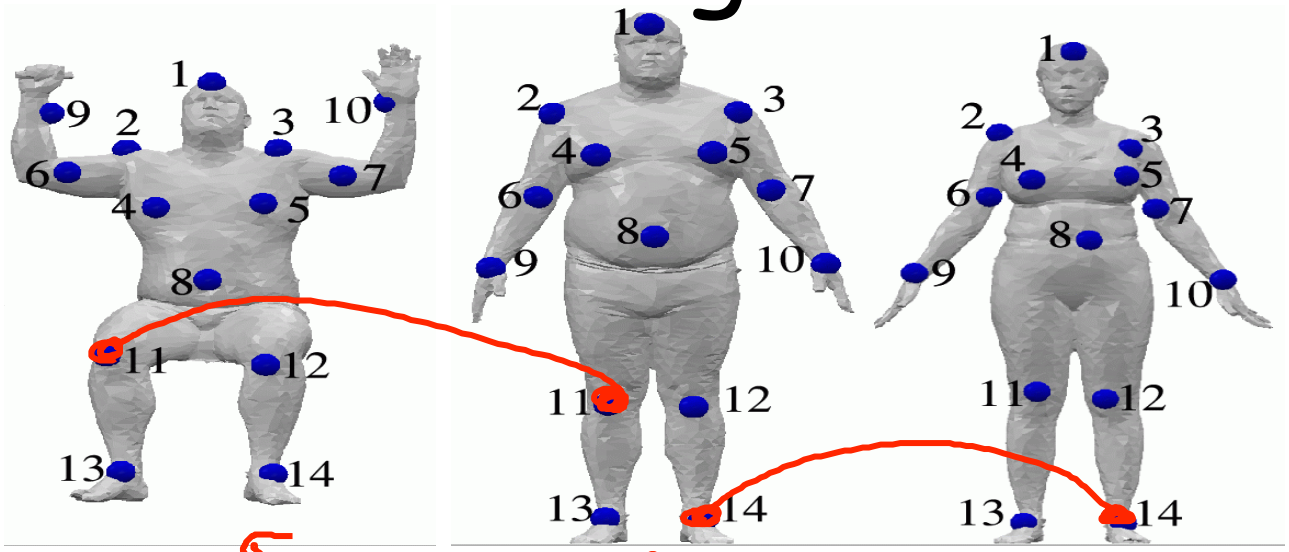


- Matching weights: similarity of location and local neighborhood appearance

Duchi, Tarlow, Elidan, and Koller, NIPS 2006. Amat, Moussavi, Comolli, Elidan, Downing, Horowitz, Journal of Strurctural Biology, 2006.

Daphne Koller

Mesh Registration



- Matching weights: similarity of location and local neighborhood appearance

[Anguelov, Koller, Srinivasan, Thrun, Pang, Davis, NIPS 2004]

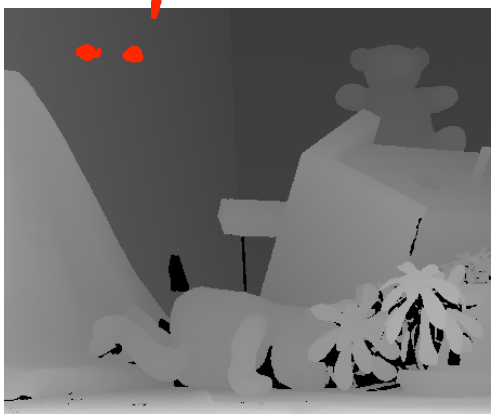
Associative potentials

- Arbitrary network over binary variables using only singleton θ_i and supermodular pairwise potentials θ_{ij}
  - Exact solution using algorithms for finding minimum cuts in graphs
- Many related variants admit efficient exact or approximate solutions
  - Metric MRFs

	0	1
0	a	b
1	c	d

$a + d \geq b + c$

Example: Depth Reconstruction



view 1

view 2

depth reconstruction

denoising, inpainting, FG/BG segmentation

Scharstein & Szeliski, "High-accuracy stereo depth maps using structured light" Proc. IEEE CVPR 2003

Cardinality Factors

- A factor over arbitrarily many binary variables $X_1, \ldots, X_k$
- $\text{Score}(X_1, \ldots, X_k) = f(\Sigma_i X_i)$
- Example applications:
  - soft parity constraints
  - prior on # pixels in a given category
  - prior on # of instances assigned to a given cluster

A	B	C	D	score
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Daphne Koller

Sparse Pattern Factors

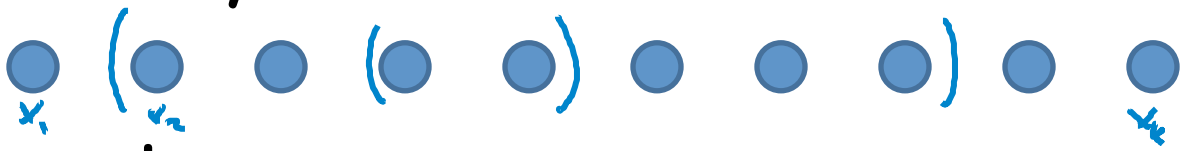
- A factor over variables $X_1,\ldots,X_k$
  - Score($X_1, \ldots,X_k$) specified for some small # of assignments $x_1,\ldots,x_k$
  - Constant for all other assignments
- Examples: give higher score to combinations that occur in real data
  - In spelling, letter combinations that occur in dictionary
  - 5×5 image patches that appear in natural images

A	B	C	D	score
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Daphne Koller

Convexity Factors

- Ordered binary variables $X_1,\ldots,X_k$
- Convexity constraints

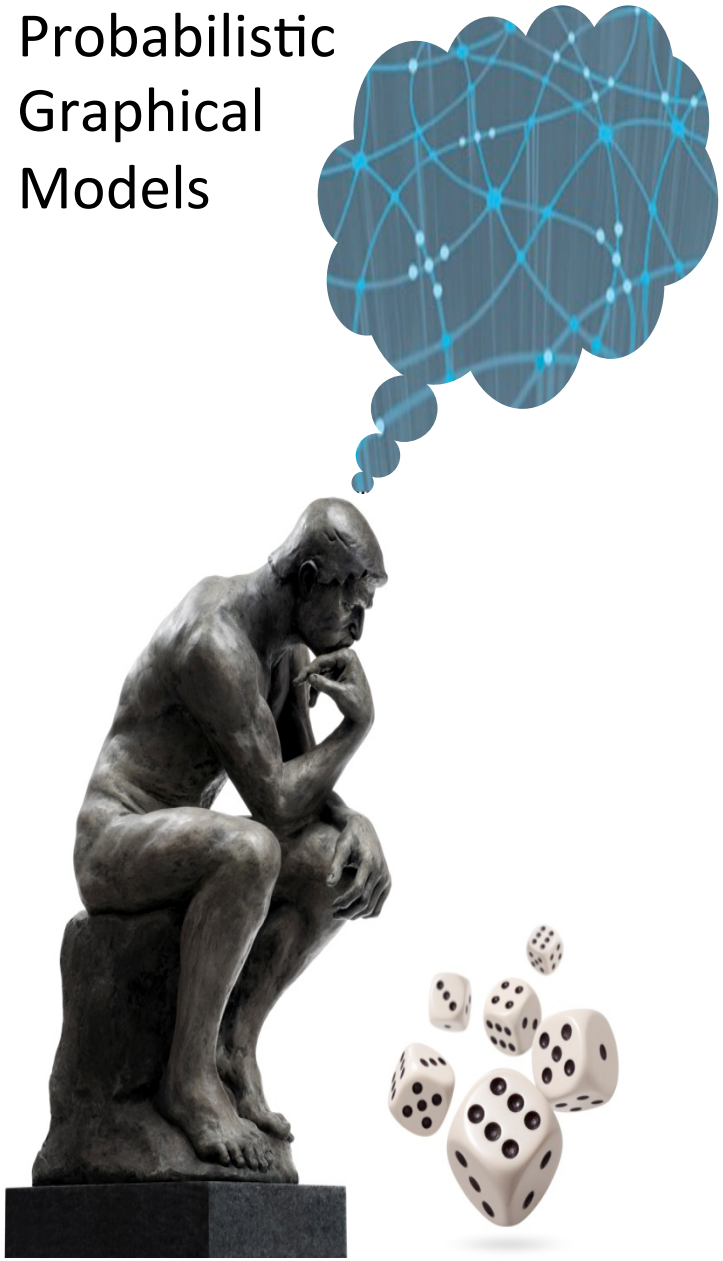


- Examples:
  - Convexity of "parts" in image segmentation
  - Contiguity of word labeling in text
  - Temporal contiguity of subactivities

Summary

- Many specialized models admit tractable MAP solution
  - Many do not have tractable algorithms for computing marginals
- These specialized models are useful
  - On their own
  - As a component in a larger model with other types of factors

Probabilistic Graphical Models



Inference

MAP

Dual Decomposition

Problem Formulation

- Singleton factors $\theta_i(x_i)$
- Large factors $\theta_F(\boldsymbol{x}_F)$

$$\mathrm{MAP}(\boldsymbol{\theta}) = \max_{\boldsymbol{x}} \left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{F} \theta_F(\boldsymbol{x}_F) \right)$$

Divide and Conquer

$$\mathrm{MAP}(\boldsymbol{\theta}) = \max_{\boldsymbol{x}} \left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{F} \theta_F(\boldsymbol{x}_F) \right)$$

$$\sum_{i=1}^{n} \max_{x_i} \theta_i(x_i) + \sum_{F} \max_{\boldsymbol{x}_F} \theta_F(\boldsymbol{x}_F)$$

local decision making

Divide and Conquer

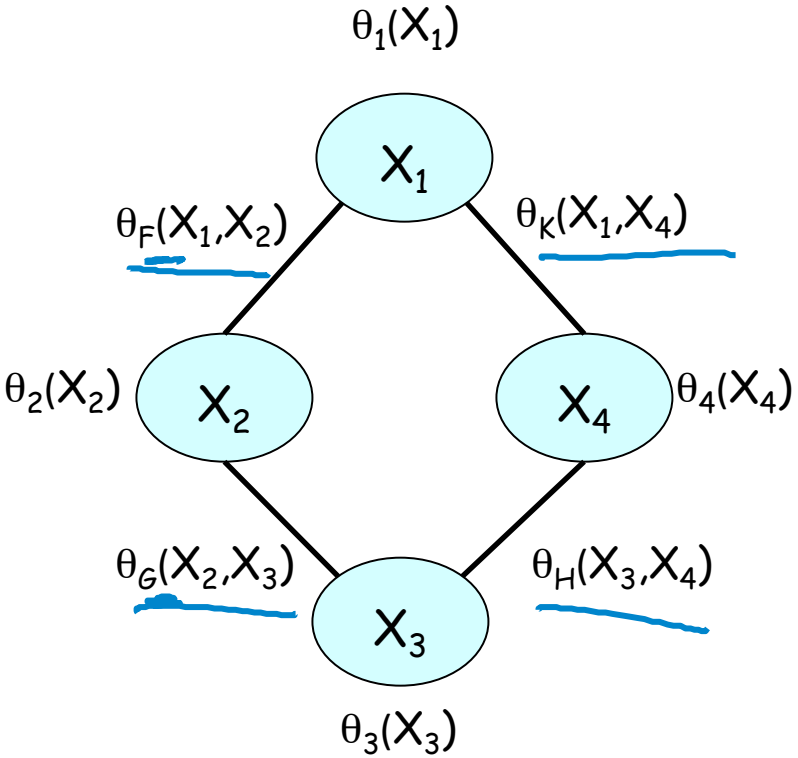
$$\mathrm{MAP}(\boldsymbol{\theta}) = \max_{\boldsymbol{x}} \left(\sum_{i=1}^{n} \theta_i(x_i) + \sum_{F} \theta_F(\boldsymbol{x}_F) \right)$$

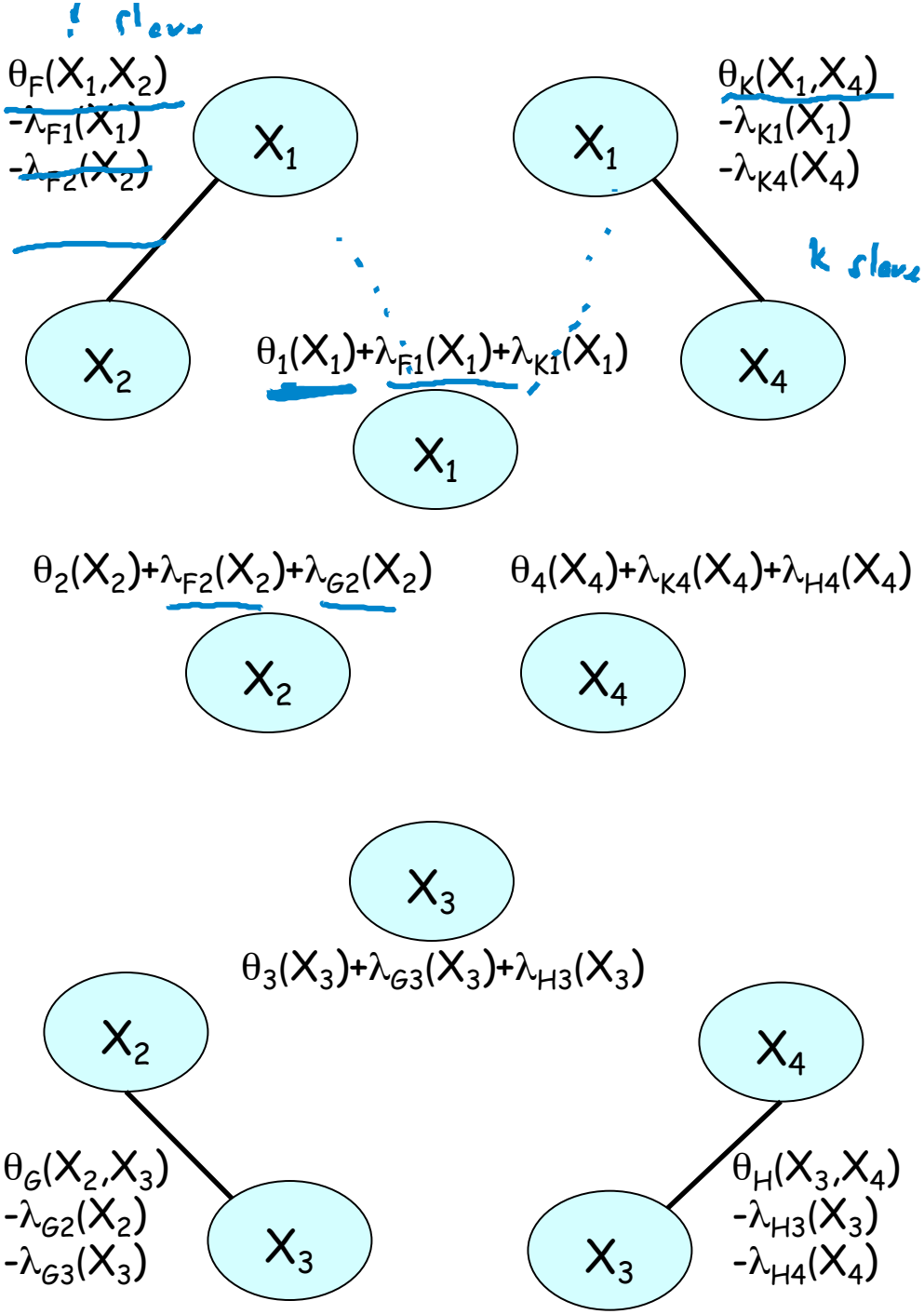
$$= \max_{\boldsymbol{x}} \left(\sum_{i=1}^{n} (\theta_i(x_i) + \sum_{F: i \in F} \lambda_{Fi}(x_i)) + \sum_{F} (\theta_F(\boldsymbol{x}_F) - \sum_{i \in F} \lambda_{Fi}(x_i)) \right)$$

$$L(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \max_{x_i} \left(\theta_i(x_i) + \sum_{F: i \in F} \lambda_{Fi}(x_i) \right) + \sum_{F} \max_{\boldsymbol{x}_F} \left(\theta_F(\boldsymbol{x}_F) - \sum_{i \in F} \lambda_{Fi}(x_i) \right)$$

$\theta_i^{\boldsymbol{\lambda}}$ $\theta_F^{\boldsymbol{\lambda}}$

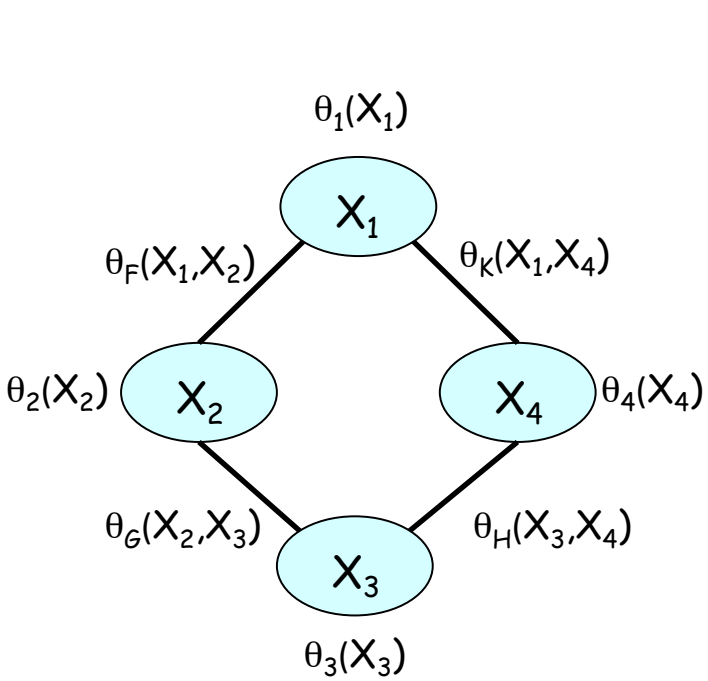
L(λ) is upper bound on MAP(θ) for any setting of λ's

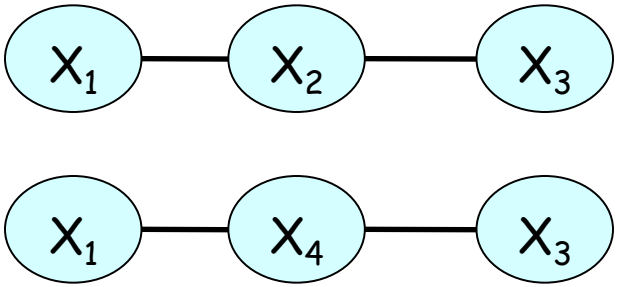

$\theta_1(X_1)$
X_1
$\theta_F(X_1,X_2)$
$\theta_K(X_1,X_4)$
$\theta_2(X_2)$
X_2
X_4
$\theta_4(X_4)$
$\theta_G(X_2,X_3)$
$\theta_H(X_3,X_4)$
X_3
$\theta_3(X_3)$


$\theta_F(X_1,X_2)$
$-\lambda_{F1}(X_1)$
$-\lambda_{F2}(X_2)$
X_1
X_2
$\theta_K(X_1,X_4)$
$-\lambda_{K1}(X_1)$
$-\lambda_{K4}(X_4)$
X_1
X_4
$\theta_1(X_1)+\lambda_{F1}(X_1)+\lambda_{K1}(X_1)$
X_1
$\theta_2(X_2)+\lambda_{F2}(X_2)+\lambda_{G2}(X_2)$
X_2
$\theta_4(X_4)+\lambda_{K4}(X_4)+\lambda_{H4}(X_4)$
X_4
X_3
$\theta_3(X_3)+\lambda_{G3}(X_3)+\lambda_{H3}(X_3)$
X_2
$\theta_G(X_2,X_3)$
$-\lambda_{G2}(X_2)$
$-\lambda_{G3}(X_3)$
X_3
X_4
$\theta_H(X_3,X_4)$
$-\lambda_{H3}(X_3)$
$-\lambda_{H4}(X_4)$
X_3

Divide and Conquer

- Slaves don't have to be factors in original model
  - Subsets of factors that admit tractable solution to local maximization task

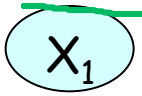




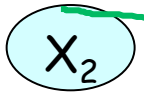
$\theta_F(X_1,X_2)+\theta_G(X_2,X_3)$
$-\lambda_{FG1}(X_1)-\lambda_{FG2}(X_2)-\lambda_{FG3}(X_3)$

$\theta_K(X_1,X_4)+\theta_H(X_3,X_4)$
$-\lambda_{KH1}(X_1)-\lambda_{KH3}(X_3)-\lambda_{KH4}(X_4)$

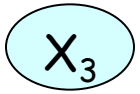
$\theta_1(X_1)+\lambda_{FG1}(X_1)+\lambda_{KH1}(X_1)$



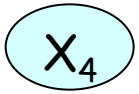
$\theta_2(X_2)+\lambda_{FG2}(X_2)$



$\theta_3(X_3)+\lambda_{FG3}(X_3)+\lambda_{KH3}(X_3)$



$\theta_4(X_4)+\lambda_{KH4}(X_4)$



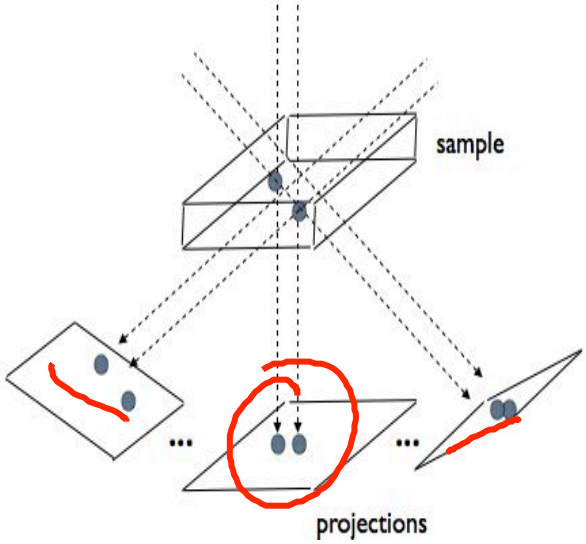
Divide and Conquer

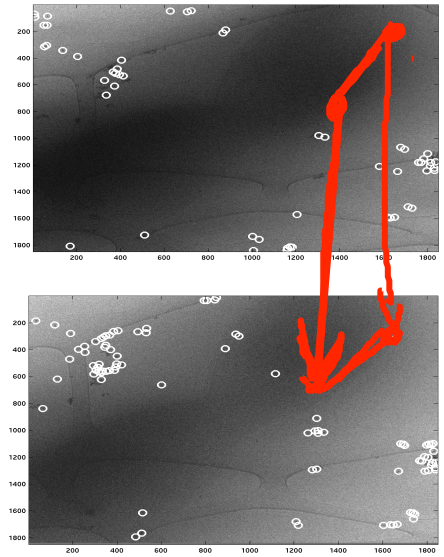
- In pairwise networks, often divide factors into set of disjoint trees
  - Each edge factor assigned to exactly one tree
- Other tractable classes of factor sets
  - Matchings
  - Associative models
  - …

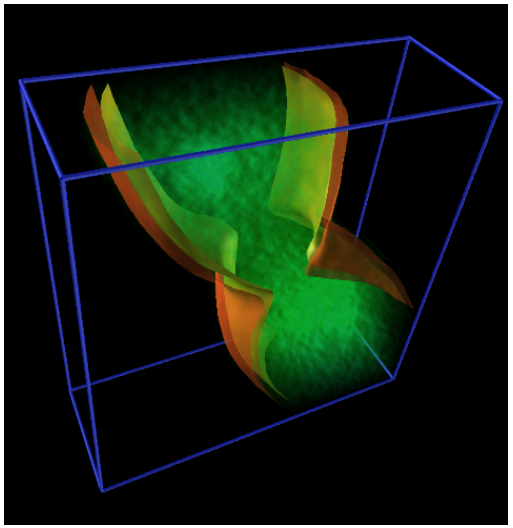
Example: 3D Cell Reconstruction

correspond tilt images

compute 3D reconstruction



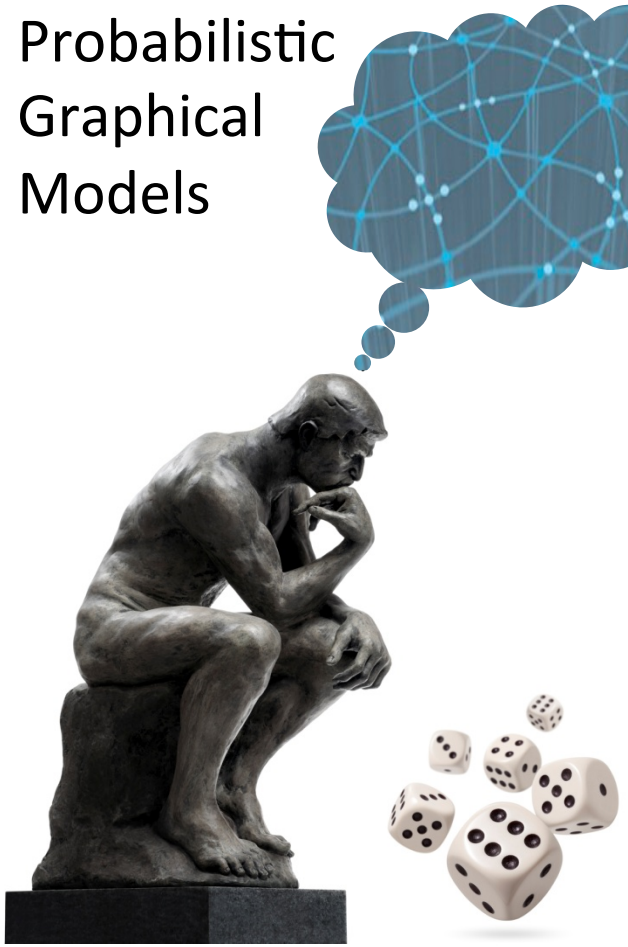




- Matching weights: similarity of location and local neighborhood appearance
- Pairwise potentials: approximate preservation of relative marker positions across images

Duchi, Tarlow, Elidan, and Koller, NIPS 2006. Amat, Moussavi, Comolli, Elidan, Downing, Horowitz, Journal of Strurctural Biology, 2006.

Probabilistic Graphical Models



Inference

MAP

Dual Decomposition Algorithm

Dual Decomposition Algorithm

$$\bar{\theta}_i^{\boldsymbol{\lambda}} = \theta_i(x_i) + \sum_{F:i \in F} \lambda_{Fi}(x_i) \qquad \bar{\theta}_F^{\boldsymbol{\lambda}} = \theta_F(\boldsymbol{x}_F) - \sum_{i \in F} \lambda_{Fi}(x_i)$$

- Initialize all λ's to be 0
- Repeat for t=1,2,…
  - Locally optimize all slaves:

$$\boldsymbol{x}_F^* = \text{argmax}_{\boldsymbol{x}_F} \bar{\theta}_F^{\boldsymbol{\lambda}}(\boldsymbol{x}_F)$$

$$x_i^* = \text{argmax}_{x_i} \bar{\theta}_i^{\boldsymbol{\lambda}}(x_i)$$

  - For all F and i∈F
    - If $x_{Fi}^* \neq x_i^*$ then

$$\lambda_{Fi}(x_i^*) := \lambda_{Fi}(x_i^*) - \alpha_t$$

$$\lambda_{Fi}(x_{Fi}^*) := \lambda_{Fi}(x_{Fi}^*) + \alpha_t$$

Dual Decomposition Convergence

- Under weak conditions on α_t, the λ's are guaranteed to converge
  - $\Sigma_t \alpha_t = \infty$
  - $\Sigma_t \alpha_t^2 < \infty$

- Convergence is to a unique global optimum, regardless of initialization

At Convergence

- Each slave has a locally optimal solution over its own variables (in its scope)
- Solutions may not agree on shared variables
- If all slaves agree, the shared solution is a guaranteed MAP assignment
- Otherwise, we need to solve the decoding problem to construct a joint assignment

Options for Decoding x*

- Several heuristics
  - If we use decomposition into spanning trees, can take MAP solution of any tree
  - Have each slave vote on X_i's in its scope & for each X_i pick value with most votes
  - Weighted average of sequence of messages sent regarding each X_i
- Score θ is easy to evaluate for any x
- **Best to generate many candidates and pick the one with highest score**

Upper Bound

- $L(\lambda)$ is upper bound on $MAP(\theta)$

candidate MAP ?

$$score(\mathbf{x}) \leq MAP(\theta) \leq L(\lambda)$$

$\theta(\mathbf{x})$

$$MAP(\theta) - score(\mathbf{x}) \leq L(\lambda) - score(\mathbf{x})$$

small enough

Important Design Choices

- Division of problem into slaves
  - Larger slaves (with more factors) improve convergence and often quality of answers
- Selecting locally optimal solutions for slaves
  - Try to move toward faster agreement
- Adjusting the step size α_t
- Methods to construct candidate solutions

Summary: Algorithm

- Dual decomposition is a general-purpose algorithm for MAP inference
  - Divides model into tractable components
  - Solves each one locally
  - Passes "messages" to induce them to agree
- Any tractable MAP subclass can be used in this setting as a slave

Summary: Theory

- Formally: a subgradient optimization algorithm on dual problem to MAP
- Provides important guarantees
  - Upper bound on distance to MAP
  - Conditions that guarantee exact MAP solution
- Even some analysis for which decomposition into slaves is better

Summary: Practice

- Pros:
  - Very general purpose
  - Best theoretical guarantees
  - Can use very fast, specialized MAP subroutines for solving large model components
- Cons:
  - Not the fastest algorithm
  - Lots of tunable parameters / design choices