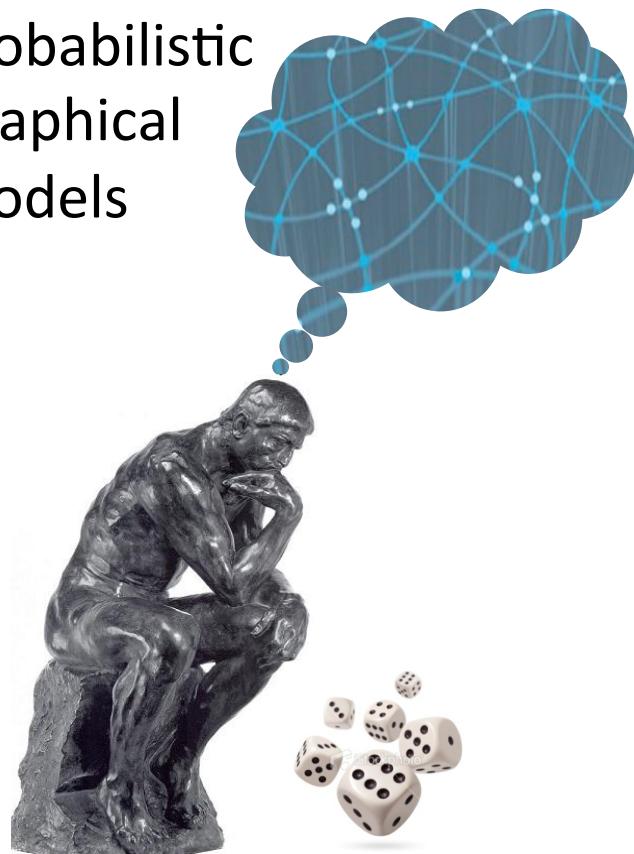


Probabilistic
Graphical
Models



Inference

MAP

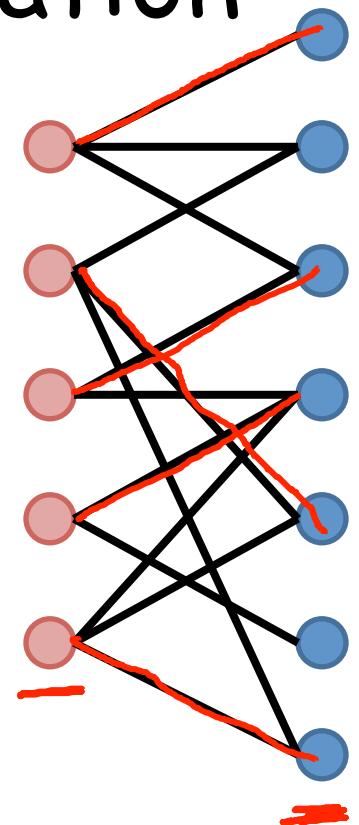
Tractable
MAP
Problems

Correspondence / data association

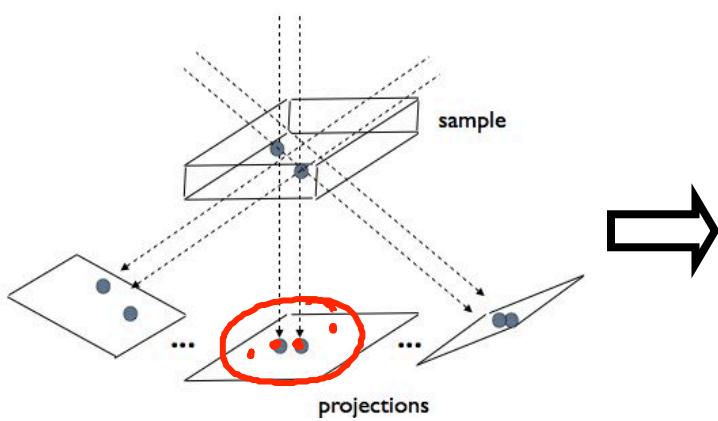
$$\underline{X_{ij}} = \begin{cases} 1 & \text{if } i \text{ matched to } j \\ 0 & \text{otherwise} \end{cases}$$

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

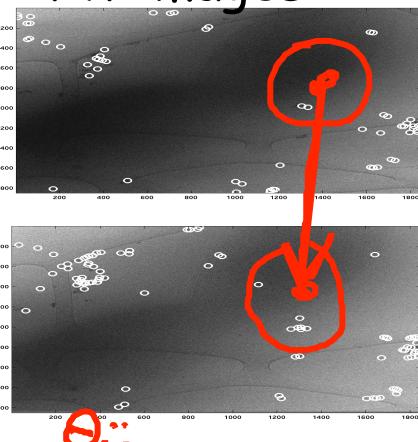
- Find highest scoring matching
 - maximize $\sum_{ij} \theta_{ij} X_{ij}$
 - subject to mutual exclusion constraint
- Easily solved using matching algorithms
- Many applications
 - matching sensor readings to objects
 - matching features in two related images ←
 - matching mentions in text to entities



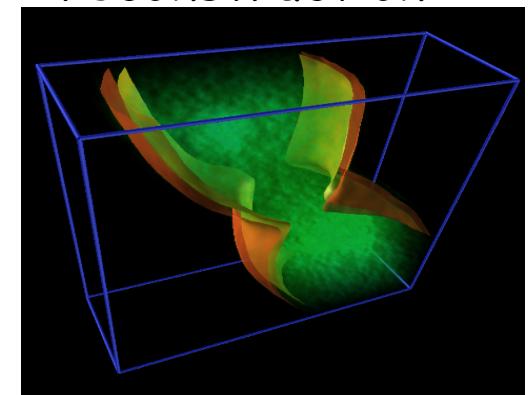
3D Cell Reconstruction



correspond
tilt images



compute 3D
reconstruction

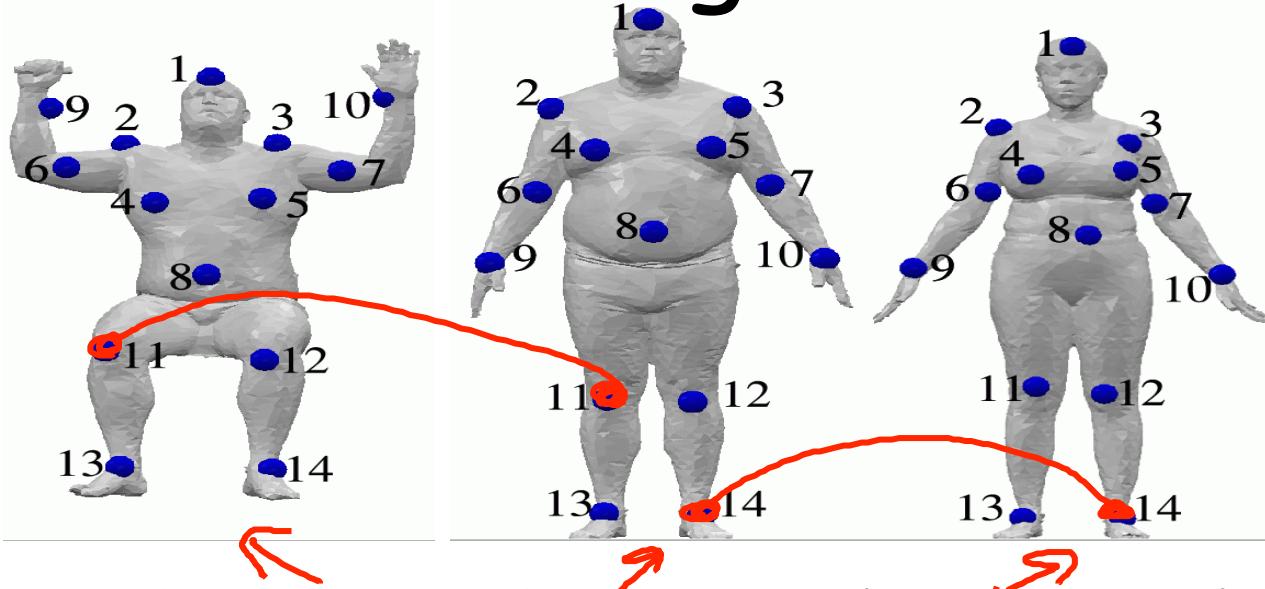


- Matching weights: similarity of location and local neighborhood appearance

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

Daphne Koller

Mesh Registration



- Matching ~~pose~~ weights; similarity ~~of gap~~ of location and local neighborhood appearance

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

Daphne Koller

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 ^{vision}

	0	1
0	a	b
1	c	d

$$\underline{a+d \geq b+c}$$

Example: Depth Reconstruction



view 1

view 2

depth
reconstruction

denoising , infilling , FG/BG segmentation

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

Daphne Koller

Cardinality Factors

- A factor over arbitrarily many binary variables X_1, \dots, X_k
- Score(X_1, \dots, X_k) = $f(\sum_i X_i)$
- 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	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	1
0	1	0	1	2
0	1	1	0	3
0	1	1	1	4
1	0	0	0	0
1	0	0	1	1
1	0	1	0	2
1	0	1	1	3
1	1	0	0	1
1	1	0	1	2
1	1	1	0	3
1	1	1	1	4

Daphne Koller

Sparse Pattern Factors

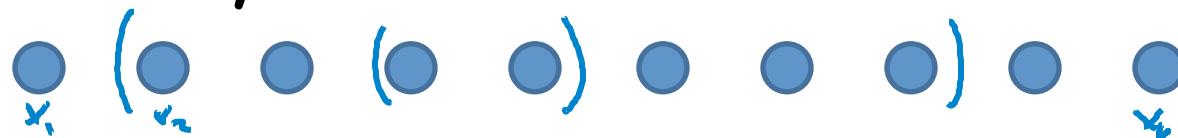
- A factor over variables X_1, \dots, X_k
 - Score(X_1, \dots, X_k) specified for some small # of assignments x_1, \dots, x_k
 - Constant for all other assignments
- Examples: give higher score to combinations that occur in real data
 - In spelling, letter combinations that occur in dictionary
 - 5x5 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, \dots, X_k
- Convexity constraints

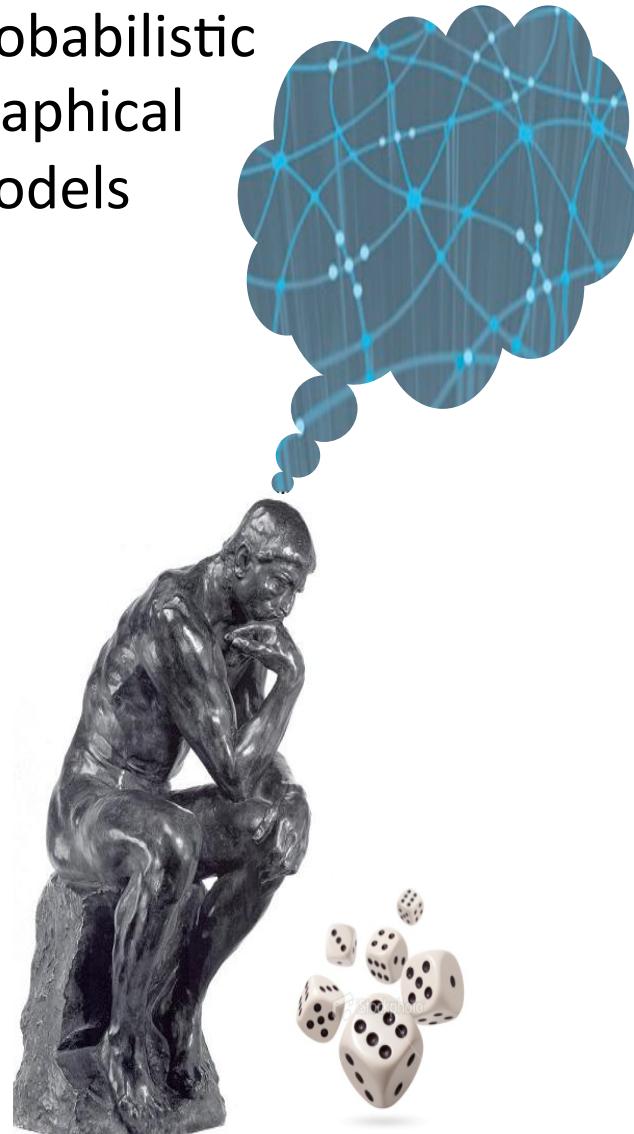


- Examples:
 - Convexity of “parts” in image segmentation
 - Contiguity of word labeling in text
 - Temporal contiguity of subactivities

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 $\theta_F(x_F)$

$$\text{MAP}(\boldsymbol{\theta}) = \max_{\boldsymbol{x}} \left(\sum_{i=1}^n \theta_i(x_i) + \sum_F \theta_F(x_F) \right)$$

Divide and Conquer

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



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

local decision making

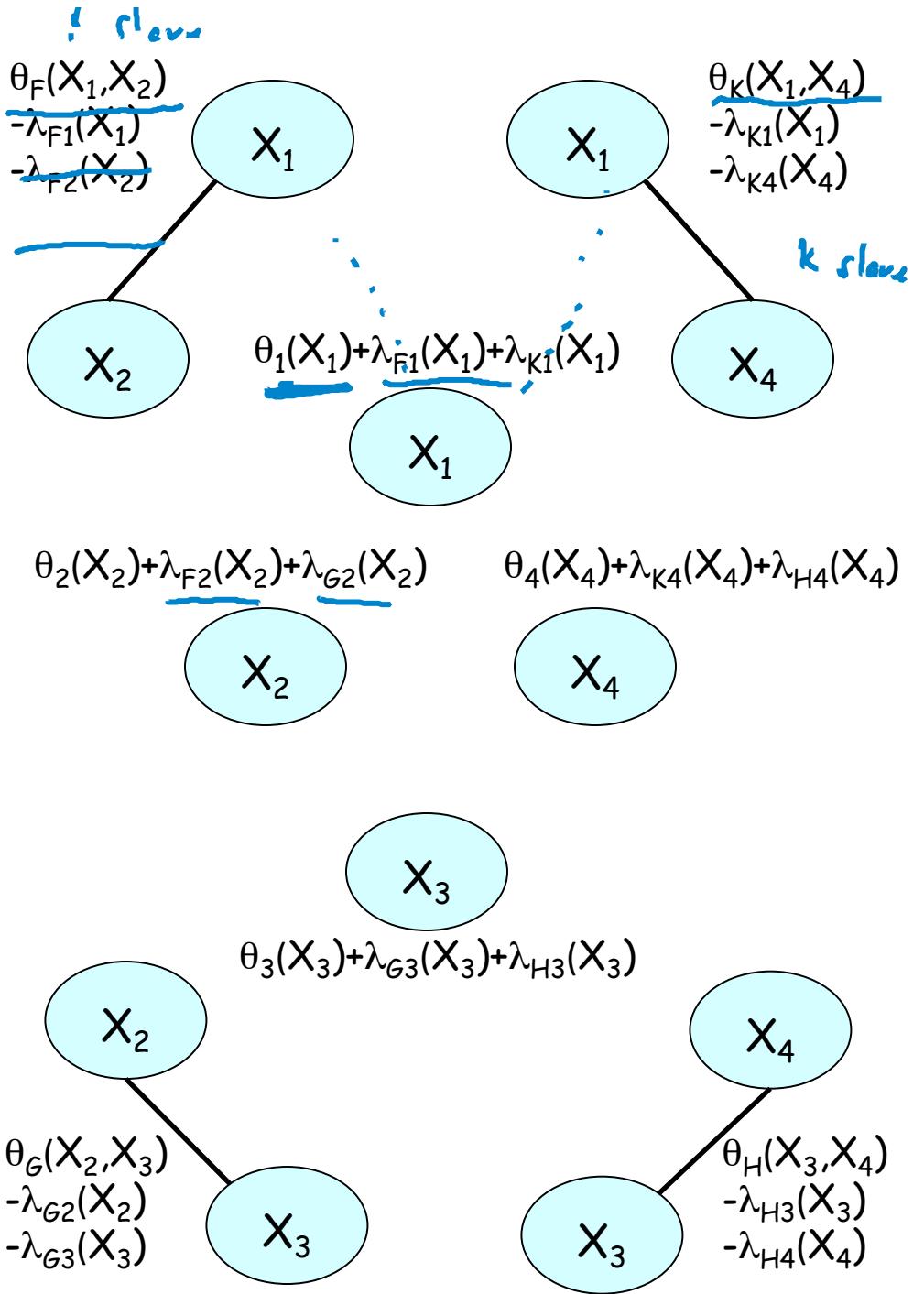
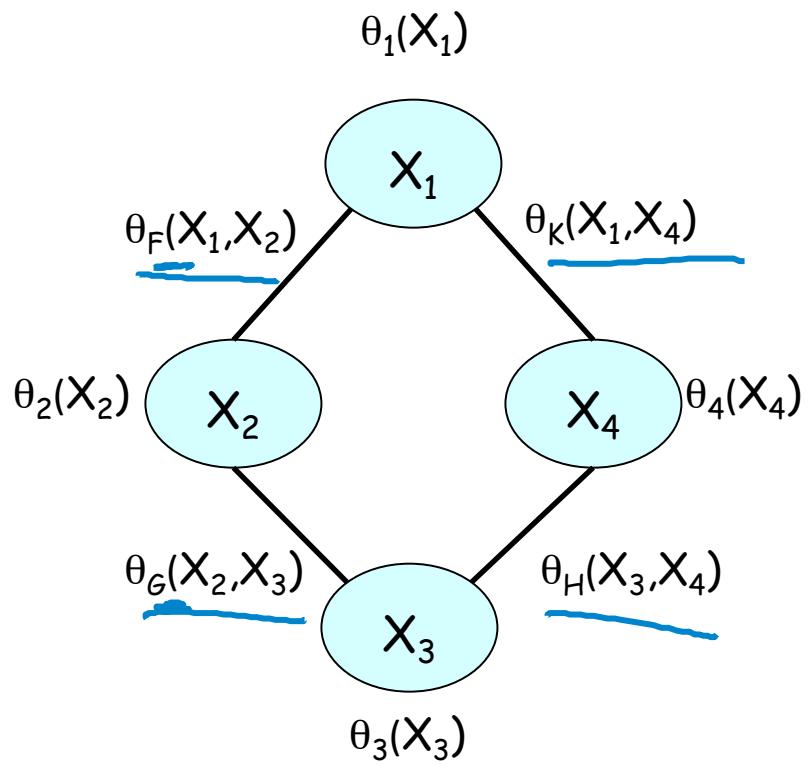
Divide and Conquer

$$\begin{aligned}
 \text{MAP}(\theta) &= \max_{\underline{x}} \left(\sum_{i=1}^n \theta_i(x_i) + \sum_F \theta_F(x_F) \right) \\
 &= \max_{\underline{x}} \left(\sum_{i=1}^n (\underbrace{\theta_i(x_i)}_{i: \text{ slave}} + \underbrace{\lambda_{F_i}(x_i)}_{F: i \in F}) + \sum_F \left(\theta_F(x_F) - \sum_{i \in F} \lambda_{F_i}(x_i) \right) \right) \\
 L(\lambda) &= \sum_{i=1}^n \max_{x_i} \left(\theta_i(x_i) + \sum_{F: i \in F} \lambda_{F_i}(x_i) \right) + \sum_F \max_{x_F} \left(\theta_F(x_F) - \sum_{i \in F} \lambda_{F_i}(x_i) \right)
 \end{aligned}$$

i: slave i ∈ F f slave
 messages between f and i
 agree with i slaves

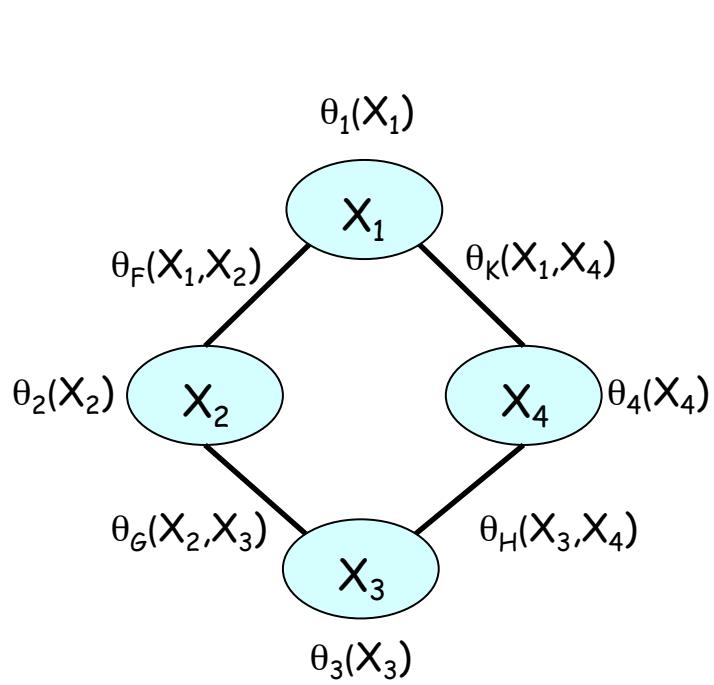
$\bar{\theta}_i^\lambda$ $\bar{\theta}_F^\lambda$

$L(\lambda)$ is upper bound on $\text{MAP}(\theta)$ for any setting of λ 's

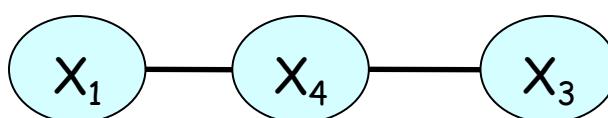


Divide and Conquer

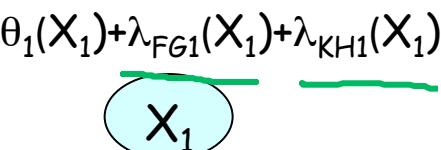
- Slaves don't have to be factors in original model
 - Subsets of factors that admit tractable solution to local maximization task



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



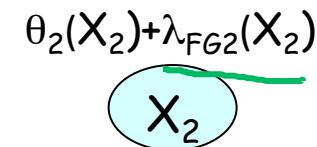
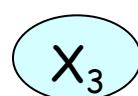
$$\begin{aligned} & \theta_K(X_1, X_4) + \theta_H(X_3, X_4) \\ & - \lambda_{KH1}(X_1) - \lambda_{KH3}(X_3) - \lambda_{KH4}(X_4) \end{aligned}$$



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



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



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

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



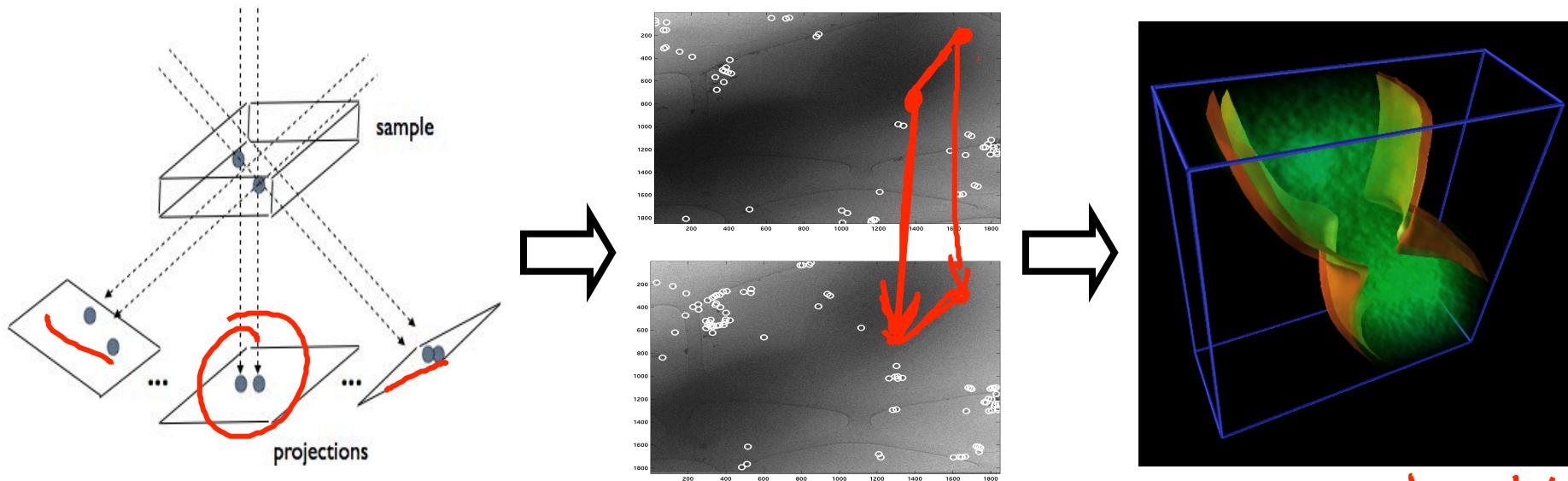
Divide and Conquer

- 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

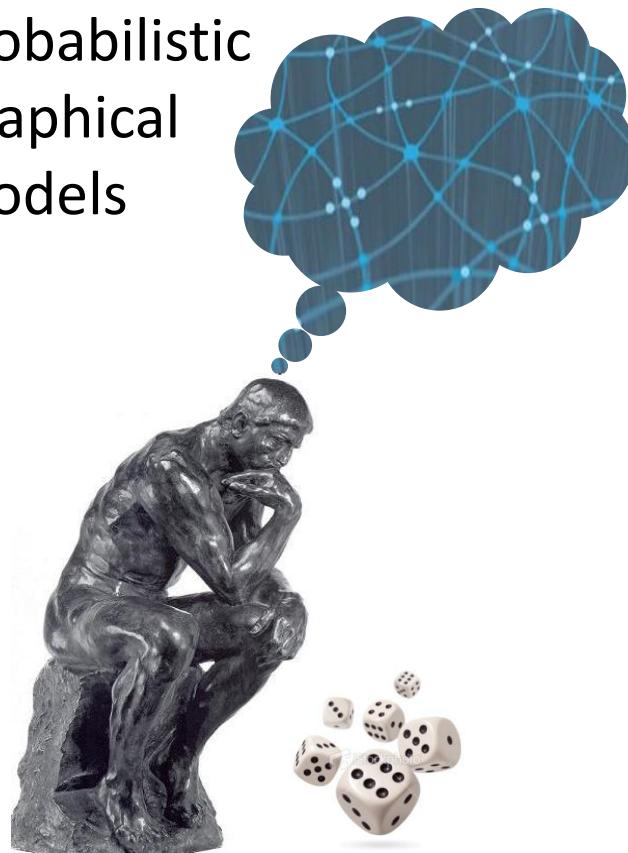
compute 3D
reconstruction



- 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 Structural Biology, 2006.

Probabilistic
Graphical
Models



Inference

MAP

Dual
Decomposition
Algorithm

Dual Decomposition Algorithm

$$\bar{\theta}_i^\lambda = \theta_i(x_i) + \sum_{F:i \in F} \lambda_{Fi}(x_i)$$

x_i $F:i \in F$

$$\bar{\theta}_F^\lambda = \theta_F(x_F) - \sum_{i \in F} \lambda_{Fi}(x_i)$$

F $i \in F$

- Initialize all λ 's to be 0

- Repeat for $t=1,2,\dots$

- Locally optimize all slaves:

- For all F and $i \in F$

- disagree* • If $x_{Fi}^* \neq x_i^*$ then

$\alpha_t > 0$

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

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

$$\underline{x}_F^* = \operatorname{argmax}_{x_F} \bar{\theta}_F^\lambda(x_F)$$

$$\underline{x}_i^* = \operatorname{argmax}_{x_i} \bar{\theta}_i^\lambda(x_i)$$

Dual Decomposition Convergence

- Under weak conditions on $\underline{\alpha_t}$, the λ 's are guaranteed to converge

$$-\sum_t \underline{\alpha_t} = \underline{\infty}$$

$$-\sum_t \underline{\alpha_t}^2 < \infty$$

- 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 $\text{MAP}(\theta)$

$$\underbrace{\text{score}(x)}_{\theta(\omega)} \leq \text{MAP}(\theta) \leq \underline{L(\lambda)}$$

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

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