Graphical Models: Learning

Pradeep Ravikumar

Co-Instructor: Ziv Bar-Joseph

Slides Courtesy: Carlos Guestrin

Machine Learning 10-701

ML MACHINE LEARNING DEPARTMENT

Carnegie Mellon.
School of Computer Science

Topics in Graphical Models

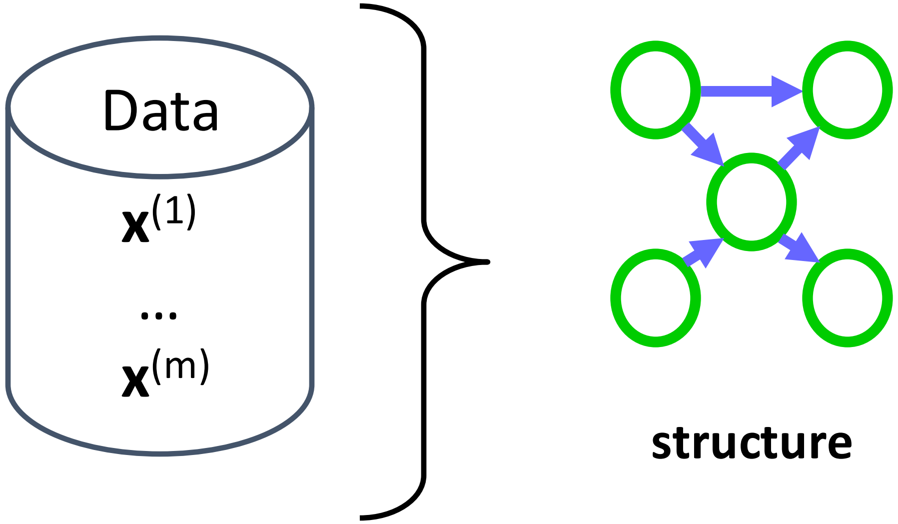
- Representation
  - Which joint probability distributions does a graphical model represent?
- Inference
  - How to answer questions about the joint probability distribution?
    - Marginal distribution of a node variable
    - Most likely assignment of node variables
- Learning
  - How to learn the parameters and structure of a graphical model?

Topics in Graphical Models

- Representation
  - Which joint probability distributions does a graphical model represent?
- Inference
  - How to answer questions about the joint probability distribution?
    - Marginal distribution of a node variable
    - Most likely assignment of node variables
- Learning
  - How to learn the parameters and structure of a graphical model?

Learning Directed Graphical Models/Bayes Nets

Learning Directed Graphical Models



+

CPTs –

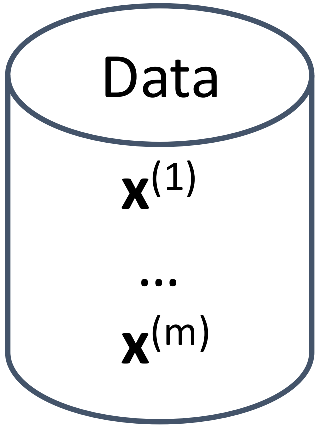
$P(X_i \mid \mathbf{Pa}_{X_i})$

parameters

Given set of m independent samples (assignments of random variables),

find the best (most likely?) Bayes Net (graph Structure + CPTs)

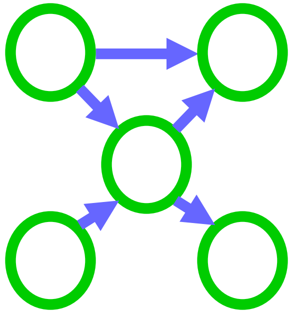
Learning the CPTs (given structure)



For each discrete variable X_k

Compute MLE or MAP estimates for

$$p(x_k | \mathrm{pa}_k)$$

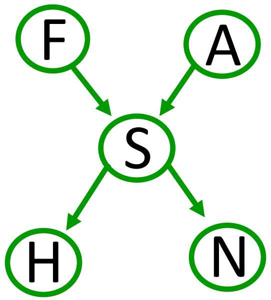


Recall

MLE: $$P(X_i = x_i \mid X_j = x_j) = \frac{\text{Count}(X_i = x_i, X_j = x_j)}{\text{Count}(X_j = x_j)}$$

MAP: Add psuedocounts

MLEs decouple for each CPT in Bayes Nets



- Given structure, log likelihood of data

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

$$= \log \prod_{j=1}^{m} P(f^{(j)}) P(a^{(j)}) P(s^{(j)} | f^{(j)}, a^{(j)}) P(h^{(j)} | s^{(j)}) P(n^{(j)} | s^{(j)})$$

$$= \sum_{j=1}^{m} [\log P(f^{(j)}) + \log P(a^{(j)}) + \log P(s^{(j)} | f^{(j)}, a^{(j)}) + \log P(h^{(j)} | s^{(j)}) + \log P(n^{(j)} | s^{(j)})]$$

$$= \underbrace{\sum_{j=1}^{m} \log P(f^{(j)})}_{\theta_F} + \underbrace{\sum_{j=1}^{m} \log P(a^{(j)})}_{\theta_A} + \underbrace{\sum_{j=1}^{m} \log P(s^{(j)} | f^{(j)}, a^{(j)})}_{\theta_{F,A}} +$$

$$\underbrace{\sum_{j=1}^{m} \log P(h^{(j)} | s^{(j)})}_{\theta_{H|S}} + \underbrace{\sum_{j=1}^{m} \log P(n^{(j)} | s^{(j)})}_{\theta_{N|S}}]$$

Depends only on

Can compute MLEs of each parameter independently!

Information theoretic interpretation of MLE

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log P\left(X_i = x_i^{(j)} \mid \mathbf{Pa}_{X_i} = \mathbf{x}_{\mathbf{Pa}_{X_i}}^{(j)}\right)$$

$$= \sum_{i=1}^{n} \sum_{x_i} \sum_{\mathbf{x}_{\mathbf{Pa}_{X_i}}} \mathrm{count}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{x}_{\mathbf{Pa}_{X_i}}) \log P\left(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{x}_{\mathbf{Pa}_{X_i}}\right)$$

Plugging in MLE estimates: ML score

$$\log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = \sum_{j=1}^{m} \sum_{i=1}^{n} \log \widehat{P}\left(x_i^{(j)} \mid \mathbf{x}_{\mathbf{Pa}_{X_i}}^{(j)}\right)$$

$$= m \sum_{i=1}^{n} \sum_{x_i} \sum_{\mathbf{x}_{\mathbf{Pa}_{X_i}}} \widehat{P}(x_i, \mathbf{x}_{\mathbf{Pa}_{X_i}}) \log \widehat{P}\left(x_i \mid \mathbf{x}_{\mathbf{Pa}_{X_i}}\right)$$

Reminds of entropy

Information theoretic interpretation of MLE

$$\log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = m \sum_{i=1}^{n} \sum_{x_i} \sum_{\mathbf{x}_{\mathbf{Pa}_{X_i}}} \widehat{P}(x_i, \mathbf{x}_{\mathbf{Pa}_{X_i}}) \log \widehat{P}\left(x_i \mid \mathbf{x}_{\mathbf{Pa}_{X_i}}\right)$$

$$= -m \sum_{i=1}^{n} \widehat{H}(X_i \mid \mathbf{Pa}_{X_i})$$

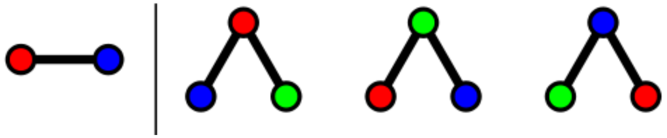
$$= m \sum_{i=1}^{n} [\widehat{I}(X_i, \mathbf{Pa}_{X_i}) - \underbrace{\widehat{H}(X_i)}_{\text{Doesn't depend on graph structure } \mathcal{G}}]$$

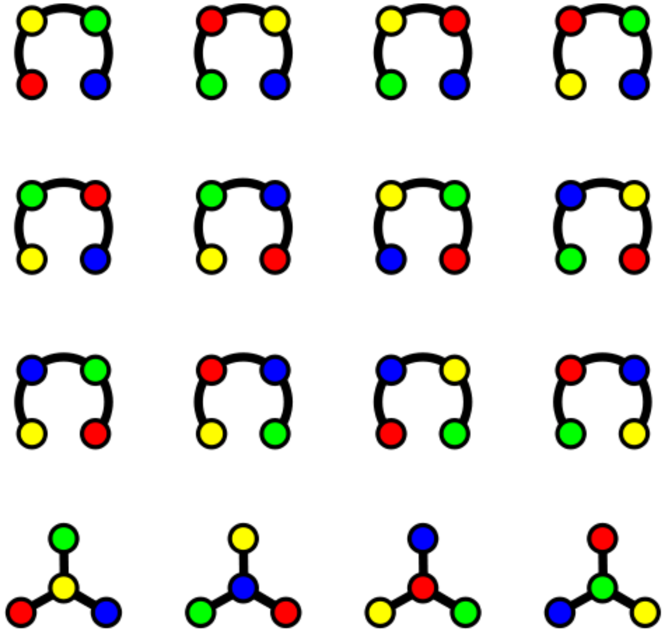
ML score for graph structure $\mathcal{G}$

$$\arg\max_{\mathcal{G}} \log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = \arg\max_{\mathcal{G}} \sum_{i=1}^{n} \widehat{I}(X_i, \mathbf{Pa}_{X_i})$$

How many trees are there?

- Trees – every node has at most one parent
- n^{n-2} possible trees (Cayley's Theorem)



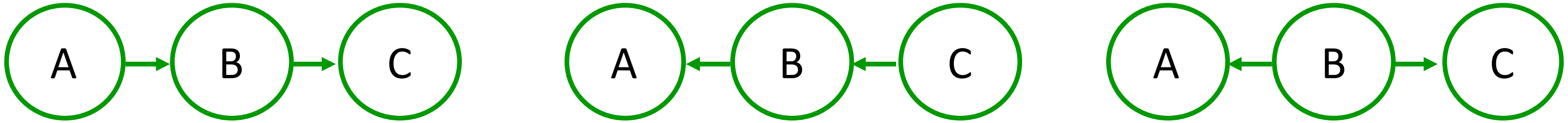


Nonetheless – Efficient optimal algorithm finds best tree!

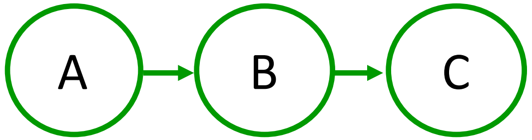
Scoring a tree

$$\arg\max_{\mathcal{G}} \log \widehat{P}(\mathcal{D} \mid \widehat{\theta}_{\mathcal{G}}, \mathcal{G}) = \arg\max_{\mathcal{G}} \sum_{i=1}^{n} \widehat{I}(X_i, \mathbf{Pa}_{X_i})$$

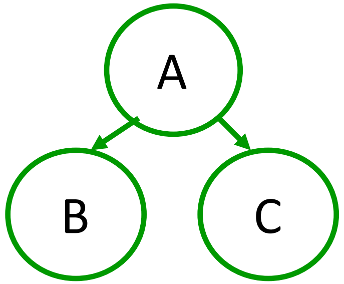
Equivalent Trees (same score): I(A,B) + I(B,C)



Score provides indication of structure:



I(A,B) + I(B,C)



I(A,B) + I(A,C)

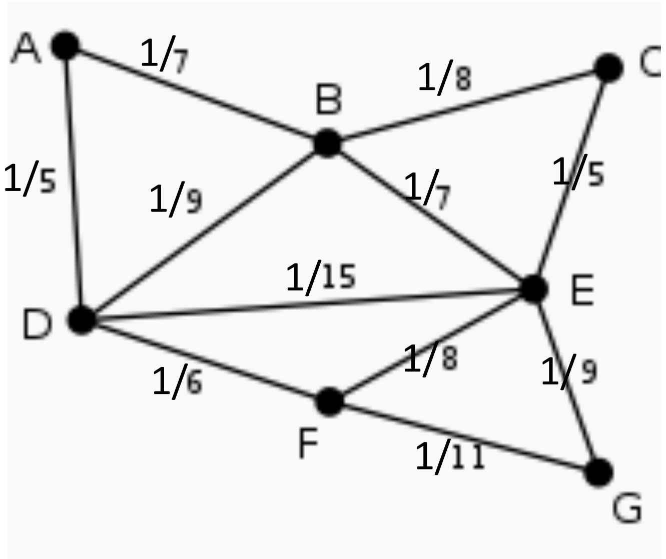
Chow-Liu algorithm

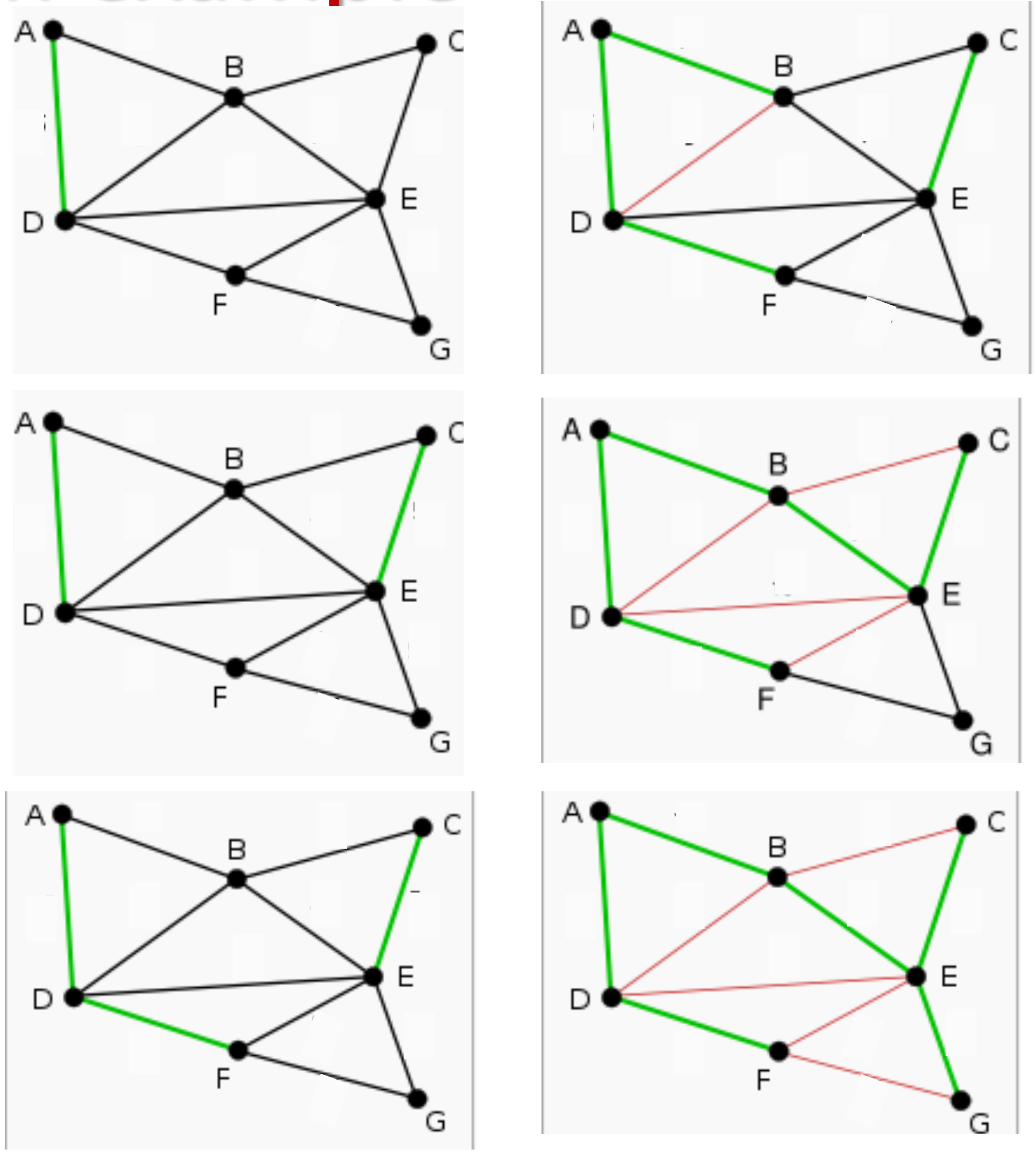
- For each pair of variables X_i, X_j
  - Compute empirical distribution: $\widehat{P}(x_i, x_j) = \frac{\text{Count}(x_i, x_j)}{m}$
  - Compute mutual information:

$$\widehat{I}(X_i, X_j) = \sum_{x_i, x_j} \widehat{P}(x_i, x_j) \log \frac{\widehat{P}(x_i, x_j)}{\widehat{P}(x_i)\widehat{P}(x_j)}$$

- Define a graph
  - Nodes $X_1, \ldots, X_n$
  - Edge (i,j) gets weight $\widehat{I}(X_i, X_j)$
- Optimal tree BN
  - Compute maximum weight spanning tree (e.g. Prim's, Kruskal's algorithm O(nlog n))
  - Directions in BN: pick any node as root, breadth-first-search defines directions

Chow-Liu algorithm example





Scoring general graphical models

- Graph that maximizes ML score -> complete graph!
-

  Adding a parent always increases ML score

  $I(A,B,C) \geq I(A,B)$

- The more edges, the fewer independence assumptions, the higher the likelihood of the data, but will overfit…

- Why does ML for trees work?

  Restricted model space – tree graph

Learning BNs for general graphs

Theorem: The problem of learning a BN structure with at most *d* parents is NP-hard for any (fixed) *d>1* (Note: tree d=1)

- Mostly heuristic (exploit score decomposition)
- Chow-Liu: provides best tree approximation to any distribution.
- Start with Chow-Liu tree. Add, delete, invert edges. Evaluate BIC score

Learning Undirected Graphical Models

Graphical models as exponential families

>Graphical Model: $p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \Psi_c(x_c)$

>As an exponential family:

$$p(x;\theta) = \exp\left\{ \sum_{c \in \mathcal{C}} \theta_c \, \phi_c(x_c) - A(\theta) \right\}$$

:: product as exponential of sum

>Ingredients:

$$\phi(x) = \{\phi_c(x_c)\}_{c \in \mathcal{C}}$$ Sufficient statistics

$$\theta = \{\theta_c\}_{c \in \mathcal{C}}$$ Parameters

$$A(\theta) = \log\left\{ \sum_x \exp\langle \theta, \phi(x) \rangle \right\}$$ Log-partition function

We will focus on pairwise graphical models

$$p(X;\theta,G) = \frac{1}{Z(\theta)} \exp\Big(\sum_{(s,t)\in E(G)} \theta_{st}\, \phi_{st}(X_s, X_t)\Big)$$

$\phi_{st}(x_s, x_t)$: arbitrary potential functions

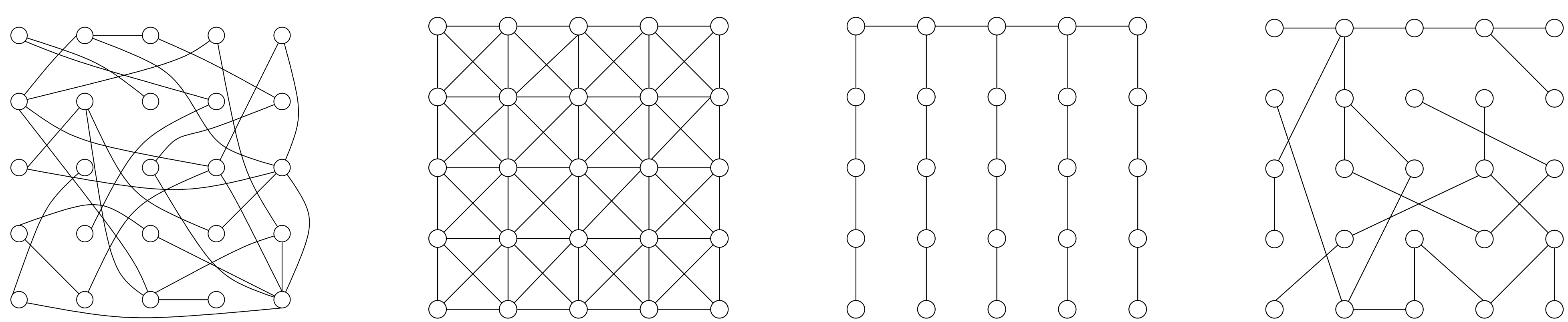
Ising	$x_s\, x_t$
Potts	$I(x_s = x_t)$
Indicator	$I(x_s, x_t = j, k)$

Graphical Model Selection

GIVEN: n samples of $X = (X_1, \ldots, X_p)$ with distribution $p(X; \theta^*; G)$, where

$$p(X; \theta^*) = \exp\left\{ \sum_{(s,t) \in E(G)} \theta_{st} \phi_{st}(x_s, x_t) - A(\theta^*) \right\}$$

PROBLEM: Estimate graph G given just the n samples.



?

Learning Graphical Models

Learning Graphical Models

- Two Step Procedures:

Learning Graphical Models

- Two Step Procedures:
  - 1. **Model Selection**; estimate graph structure

Learning Graphical Models

- Two Step Procedures:
  - 1. **Model Selection**; estimate graph structure
  - 2. **Parameter Inference** given graph structure

Learning Graphical Models

- Two Step Procedures:
  - 1. **Model Selection**; estimate graph structure
  - 2. **Parameter Inference** given graph structure
- **Score Based Approaches**: **search** over space of graphs, with a score for graph based on parameter inference

Learning Graphical Models

- Two Step Procedures:
  - 1. **Model Selection**; estimate graph structure
  - 2. **Parameter Inference** given graph structure
- **Score Based Approaches**: **search** over space of graphs, with a score for graph based on parameter inference
- **Constraint-based Approaches**: estimate individual edges by **hypothesis tests** for conditional independences

Learning Graphical Models

- Two Step Procedures:
  - 1. **Model Selection**; estimate graph structure
  - 2. **Parameter Inference** given graph structure
- **Score Based Approaches**: **search** over space of graphs, with a score for graph based on parameter inference
- **Constraint-based Approaches**: estimate individual edges by **hypothesis tests** for conditional independences
- Caveats: (a) difficult to provide guarantees for estimators; (b) estimators are NP-Hard

Sparse Graphical Model Inference

$$p(X;\theta,G) = \frac{1}{Z(\theta)} \exp\Big(\sum_{(s,t)\in E(G)} \theta_{st}\,\phi_{st}(X_s, X_t)\Big)$$

- Consider the zero-padded parameter vector $\theta \in \mathbb{R}^{\binom{p}{2}}$ (with a parameter for each node-pair)
- Graph being sparse **equiv. to** parameter vector \theta being sparse
- Can be expressed as the constraint that $\|\theta\|_0 \leq k$
- **One step inference**: Parameter Inference subject to sparsity constraint (in contrast to model selection first, with parameter inference in an inner loop)

Sparsity Constrained MLE

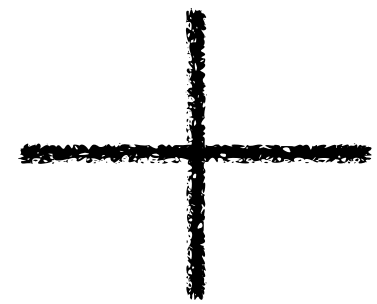
$$\widehat{\theta} \in \arg \min_{\theta : \|\theta\|_0 \leq k} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log p(x^{(i)}; \theta) \right\}$$

sparsity constraint **neg. log-likelihood**

- Optimization problem **intractable** because of
  - Sparsity Constraint :: Non-convex
  - Log-partition function $A(\theta)$ :: NP-Hard to **compute**

Intractable Components

- Sparsity Constraint is non-convex



- Log-partition function requires exponential time to compute

Unnormalized Probability: $$p(x;\theta) \propto \exp(\theta^T \phi(x))$$

Log-normalization Const: $$A(\theta) = \log\left\{\sum_{\mathbf{x}} \exp(\theta^T \phi(x))\right\}$$

Exponentially many vectors

Pairwise Binary Graphical Models

Pairwise: $\mathbb{P}_\theta(X) = \exp\left\{ \sum_{(s,t)\in E} \theta_{st} X_s X_t - A(\theta) \right\}$

Binary: $X_s \in \{-1, +1\}; s \in V$

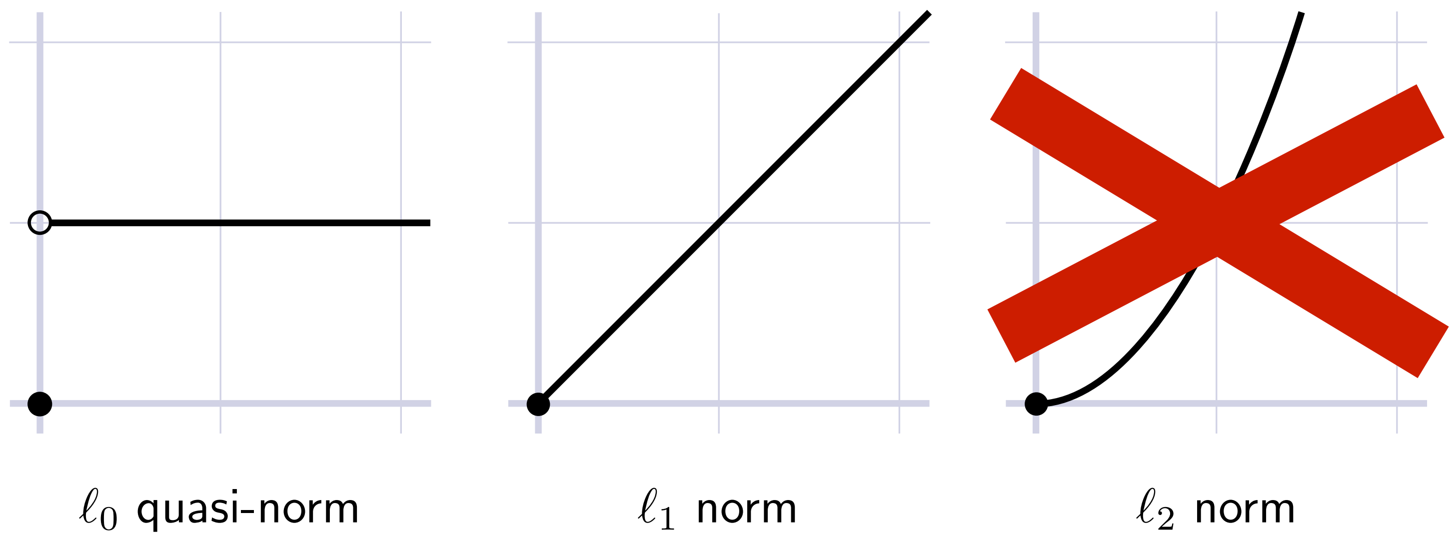
Tractable Estimator:

> Sparsity: **ell_1**

> Likelihood: **pseudolikelihood**

R., Wainwright, Lafferty 06,08

Sparsity



[From **Tropp, J.** 2004]

Sparsity: ell_0(params) is small
Convex relaxation: ell_1(params) is small

$$||\theta||_1 = \sum_{j=1}^{p} |\theta_j|$$

<u>Some past work:</u> Tibshirani, 1996; Chen et al., 1998; Donoho/Xuo, 2001; Tropp, 2004; Fuchs, 2004; Meinshausen/Buhlmann, 2005; Candes/Tao, 2005; Donoho, 2005; Haupt & Nowak, 2006; Zhao/Yu, 2006; Wainwright, 2006; Zou, 2006; Koltchinskii, 2007; Meinshausen/Yu, 2007; Tsybakov et al., 2008

Pseudo-likelihood

$$\mathbb{P}_\theta^{\mathrm{PL}}(X) = \prod_{i=1}^{p} \mathbb{P}_\theta(X_i | X_{V \setminus i})$$

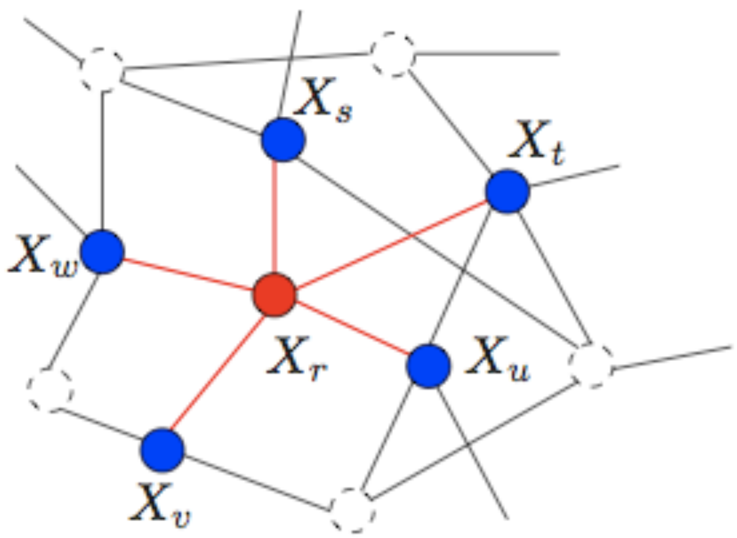
> **Approximate likelihood** via product of node-conditional distributions

> Sparsity constrained pseudolik. MLE equivalent to **neighborhood estimation*   :**

. Estimate neighborhood of each node; via sparsity constrained node conditional MLE

. Combine neighborhoods to form graph estimate

Neighborhood Estimation in Ising Models



For Ising models, node conditional dist. is logistic:

$$p(X_r | X_{V \setminus r}; \theta, G) = \frac{\exp(\sum_{t \in N(r)} 2\,\theta_{rt} X_r X_t)}{\exp(\sum_{t \in N(r)} 2\,\theta_{rt} X_r X_t) + 1}$$

- Sparsity pattern of conditional distribution parameters: neighborhood structure in original graph.

- Estimate sparsity constrained node conditional distribution (ell_1 regularized logistic regression)

Graph selection via neighborhood regression

Observation: Recovering graph G equivalent to recovering neighborhood set $N(s)$ for all $s \in V$.

Method: Given n i.i.d. samples $\{X^{(1)}, \ldots, X^{(n)}\}$, perform logistic regression of each node X_s on $X_{\setminus s} := \{X_s,\ t \neq s\}$ to estimate neighborhood structure $\widehat{N}(s)$.

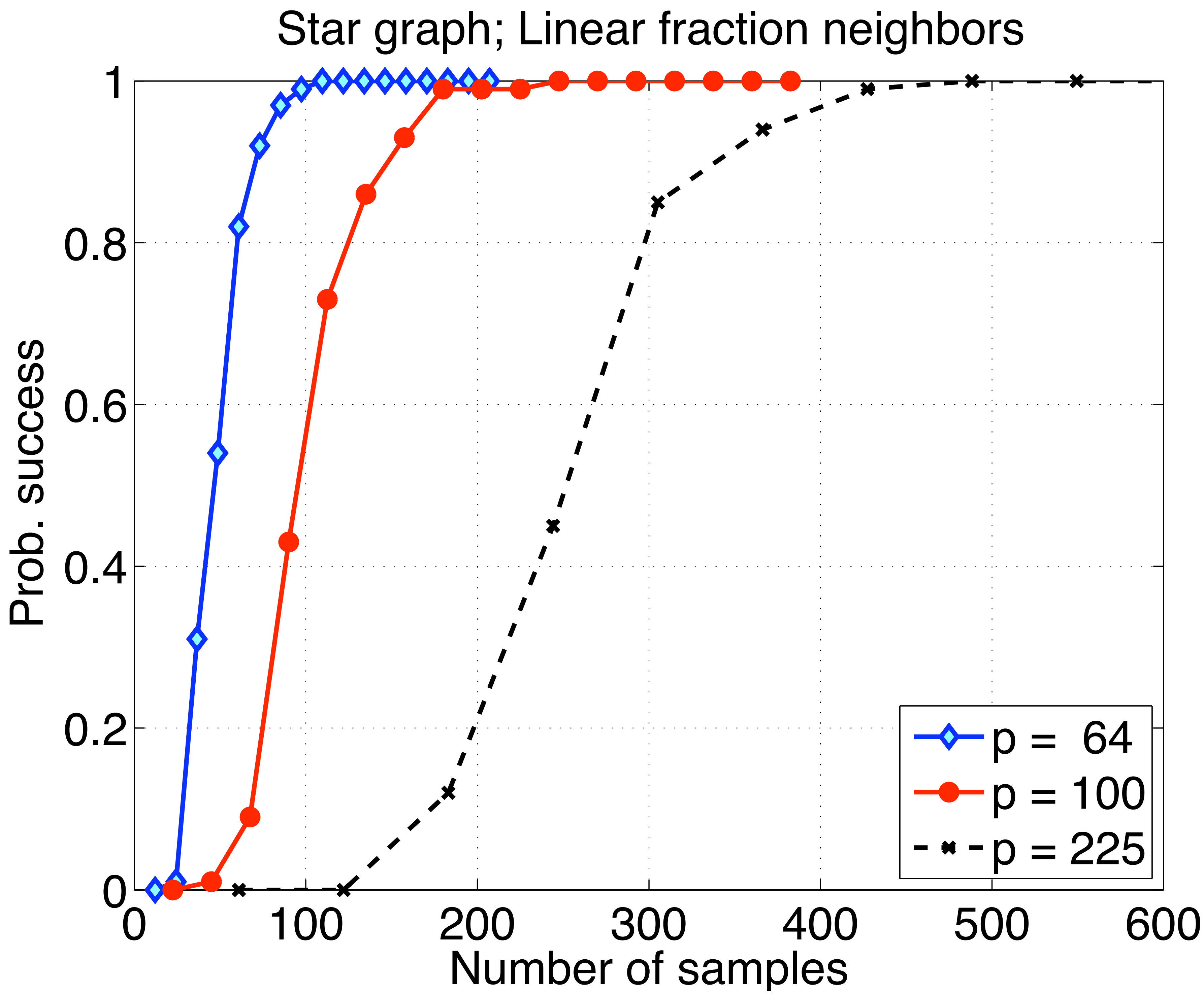
1. For each node $s \in V$, perform ℓ_1 regularized logistic regression of X_s on the remaining variables $X_{\setminus s}$:

$$\widehat{\theta}[s] \quad := \quad \arg \min_{\theta \in \mathbb{R}^{p-1}} \Big\{ \quad \frac{1}{n} \sum_{i=1}^{n} \underbrace{f(\theta; X_{\setminus s}^{(i)})}_{\text{logistic likelihood}} \quad + \quad \rho_n \underbrace{\|\theta\|_1}_{\text{regularization}} \Big\}$$

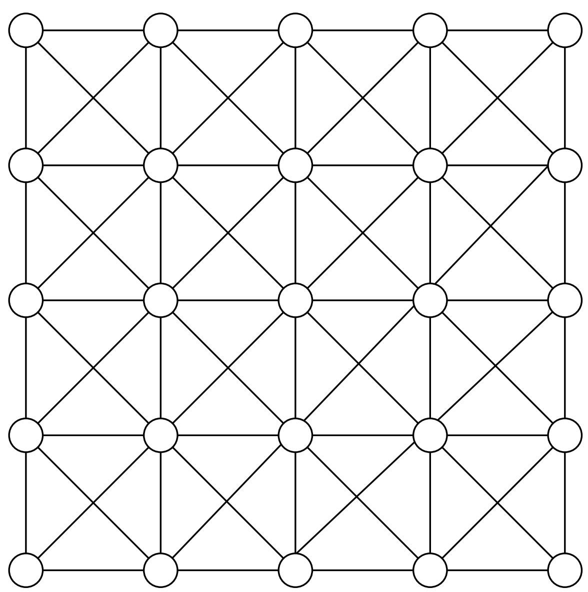
2. Estimate the local neighborhood $\widehat{N}(s)$ as the support (non-negative entries) of the regression vector $\widehat{\theta}[s]$.

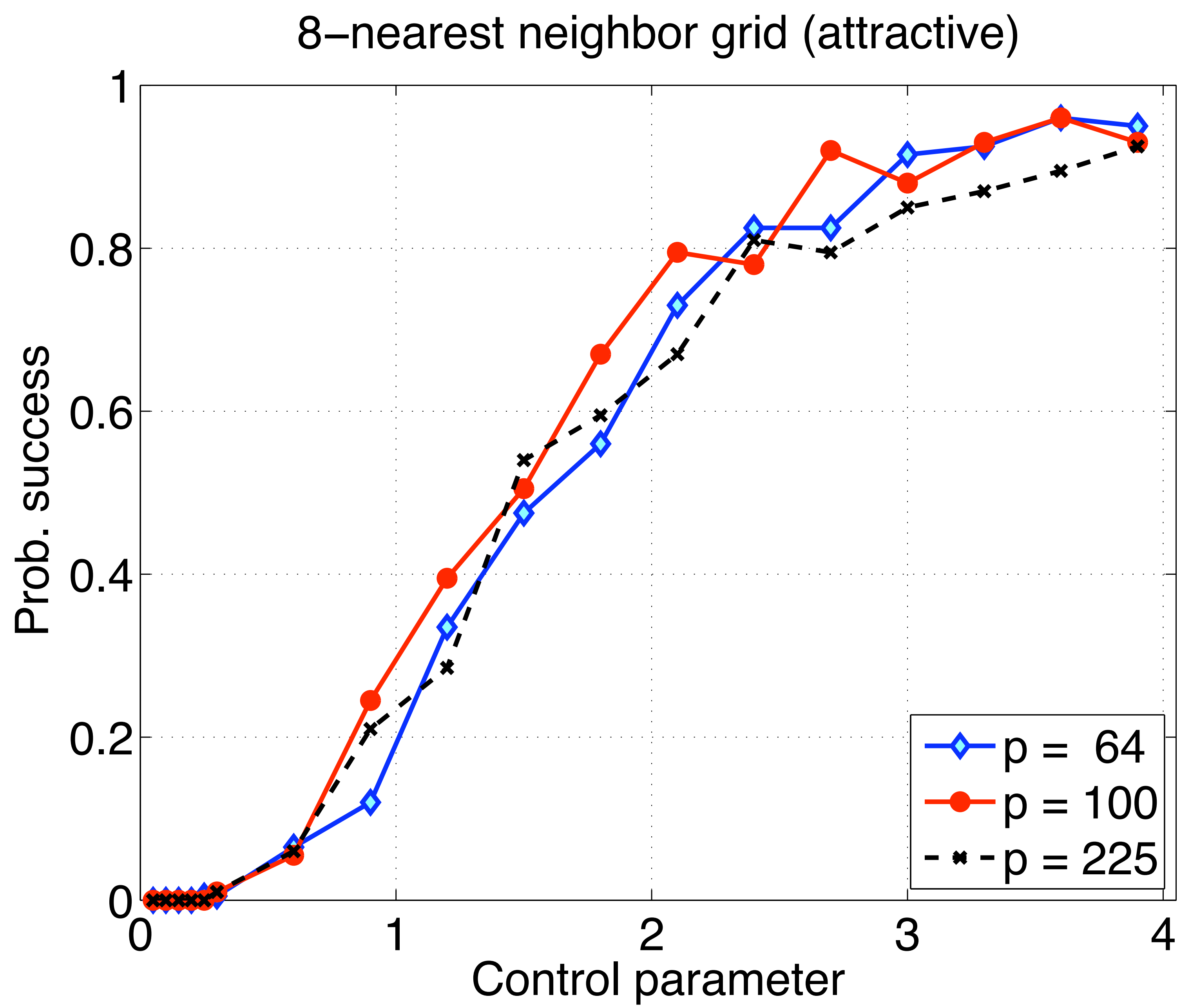
3. Combine the neighborhood estimates in a consistent manner (AND, or OR rule).

Empirical behavior: Unrescaled plots



Results for 8-grid graphs





Prob. of success $\mathbb{P}[\widehat{G} = G]$ versus rescaled sample size $\theta_{LR}(n, p, d^3) = \frac{n}{d^3 \log p}$