Structure Learning in Undirected Graphical Models

Mark Schmidt

INRIA - SIERRA team
Laboratoire d'Informatique de l'Ecole Normale Suprieure

January 20, 2011

Outline

Motivation for Graphical Model Structure Learning

car	drive	files	hockey	mac	league	pc	win
0	0	1	0	1	0	1	0
0	0	0	1	0	1	0	1
1	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0
0	0	1	0	0	0	1	1

- What words are related?
- Is a post with (car,drive,hockey,pc,win) spam?
- What is $p(\text{car}|\text{drive})$? What about $p(\text{car}|\text{drive,files})$?
- Can we 'fill in' some variables given the others?
- Can we generate more items that look like this?

Motivation for Graphical Model Structure Learning

car	drive	files	hockey	mac	league	pc	win
0	0	1	0	1	0	1	0
0	0	0	1	0	1	0	1
1	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0
0	0	1	0	0	0	1	1

- What words are related?
- Is a post with (car,drive,hockey,pc,win) spam?
- What is $p(\text{car}|\text{drive})$? What about $p(\text{car}|\text{drive,files})$?
- Can we 'fill in' some variables given the others?
- Can we generate more items that look like this?

Motivation for Graphical Model Structure Learning

car	drive	files	hockey	mac	league	pc	win
0	0	1	0	1	0	1	0
0	0	0	1	0	1	0	1
1	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0
0	0	1	0	0	0	1	1

- What words are related?
- Is a post with (car,drive,hockey,pc,win) spam?
- What is $p(\text{car}|\text{drive})$? What about $p(\text{car}|\text{drive,files})$?
- Can we 'fill in' some variables given the others?
- Can we generate more items that look like this?

Motivation for Graphical Model Structure Learning

car	drive	files	hockey	mac	league	pc	win
0	0	1	0	1	0	1	0
0	0	0	1	0	1	0	1
1	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0
0	0	1	0	0	0	1	1

- What words are related?
- Is a post with (car,drive,hockey,pc,win) spam?
- What is $p(\text{car}|\text{drive})$? What about $p(\text{car}|\text{drive,files})$?
- Can we 'fill in' some variables given the others?
- Can we generate more items that look like this?

Motivation for Graphical Model Structure Learning

car	drive	files	hockey	mac	league	pc	win
0	0	1	0	1	0	1	0
0	0	0	1	0	1	0	1
1	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0
0	0	1	0	0	0	1	1

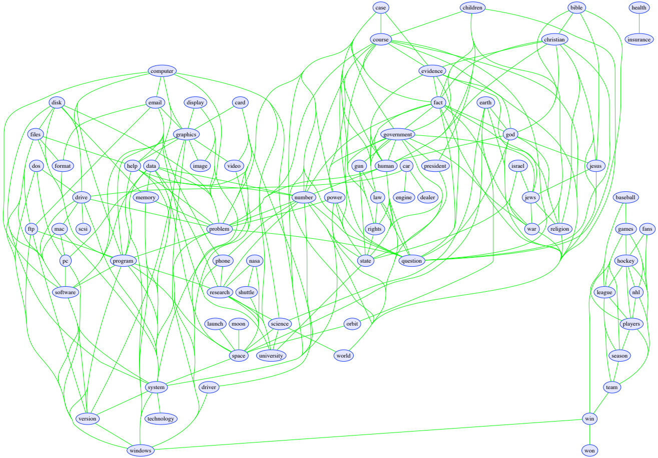
- What words are related?
- Is a post with (car,drive,hockey,pc,win) spam?
- What is $p(\text{car}|\text{drive})$? What about $p(\text{car}|\text{drive,files})$?
- Can we 'fill in' some variables given the others?
- Can we generate more items that look like this?

Motivation for Graphical Model Structure Learning

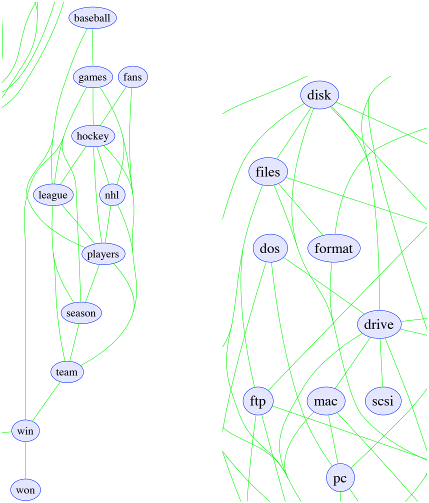
car	drive	files	hockey	mac	league	pc	win
0	0	1	0	1	0	1	0
0	0	0	1	0	1	0	1
1	1	0	0	0	0	0	0
0	1	1	0	1	0	0	0
0	0	1	0	0	0	1	1

- What words are related?
- Is a post with (car,drive,hockey,pc,win) spam?
- What is $p(\text{car}|\text{drive})$? What about $p(\text{car}|\text{drive,files})$?
- Can we 'fill in' some variables given the others?
- Can we generate more items that look like this?

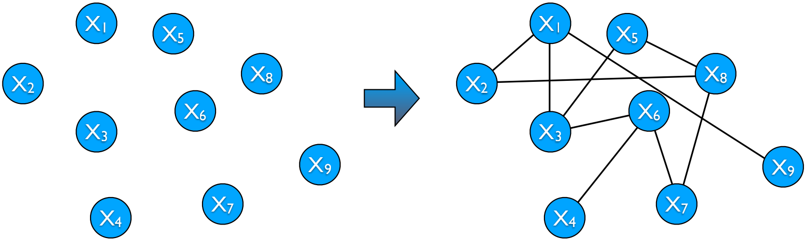
Example of Learned Graph Structure



Example of Learned Graph Structure

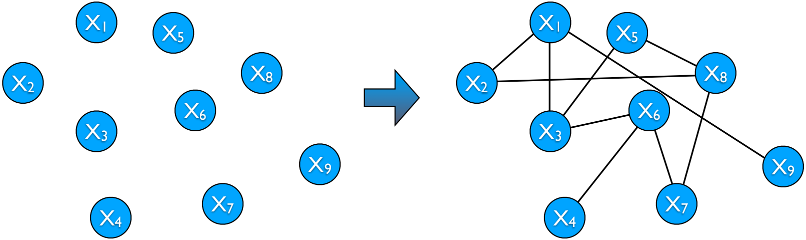


Estimation in Graphical Models with Unknown Structure



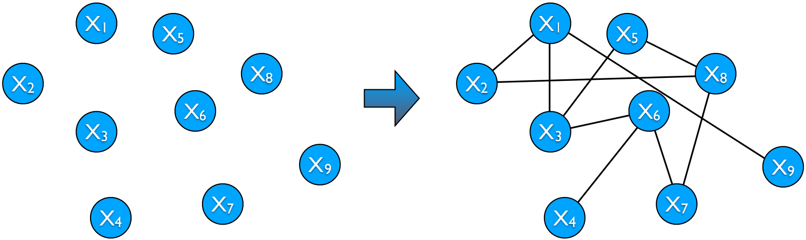
- Undirected graphical models are used to efficiently represent probability distributions in various applications.
- Often the graph structure is known (or assumed).
- We consider parameter estimation with an unknown structure.

Estimation in Graphical Models with Unknown Structure



- Undirected graphical models are used to efficiently represent probability distributions in various applications.
- Often the graph structure is known (or assumed).
- We consider parameter estimation with an unknown structure.

Estimation in Graphical Models with Unknown Structure



- Undirected graphical models are used to efficiently represent probability distributions in various applications.
- Often the graph structure is known (or assumed).
- We consider parameter estimation with an unknown structure.

Motivations for doing Structure Learning

- One approach to this task is to simply fit a dense model.
- Alternately, we can search for a sparse set of edges.
- Reasons why we might prefer the sparse approach:
  - Statistical efficiency
  - Computational efficiency
  - Structural discovery
- There are two classical methods for estimating sparse models:
  - Constraint-based approaches
  - Search and score approaches

Motivations for doing Structure Learning

- One approach to this task is to simply fit a dense model.
- Alternately, we can search for a sparse set of edges.
- Reasons why we might prefer the sparse approach:
  - Statistical efficiency
  - Computational efficiency
  - Structural discovery
- There are two classical methods for estimating sparse models:
  - Constraint-based approaches
  - Search and score approaches

Motivations for doing Structure Learning

- One approach to this task is to simply fit a dense model.
- Alternately, we can search for a sparse set of edges.
- Reasons why we might prefer the sparse approach:
  - Statistical efficiency
  - Computational efficiency
  - Structural discovery
- There are two classical methods for estimating sparse models:
  - Constraint-based approaches
  - Search and score approaches

Motivations for doing Structure Learning

- One approach to this task is to simply fit a dense model.
- Alternately, we can search for a sparse set of edges.
- Reasons why we might prefer the sparse approach:
  - Statistical efficiency
  - Computational efficiency
  - Structural discovery
- There are two classical methods for estimating sparse models:
  - Constraint-based approaches
  - Search and score approaches

Constraint-based Methods 1: Marginal Independence

- Perform a series of (in)dependence tests to discover the edges.
- One approach is using a pairwise (in)dependence statistic to:
  - Select the 'top-k' neighbors.
  - Select those above a threshold.
- Assesses marginal instead of conditional dependence:
  - 'true' neighbors may not have highest marginal dependence.
  - all variables may be marginally dependent in sparse graphs.

Constraint-based Methods 1: Marginal Independence

- Perform a series of (in)dependence tests to discover the edges.
- One approach is using a pairwise (in)dependence statistic to:
  - Select the 'top-k' neighbors.
  - Select those above a threshold.
- Assesses marginal instead of conditional dependence:
  - 'true' neighbors may not have highest marginal dependence.
  - all variables may be marginally dependent in sparse graphs.

Constraint-based Methods 1: Marginal Independence

- Perform a series of (in)dependence tests to discover the edges.
- One approach is using a pairwise (in)dependence statistic to:
  - Select the 'top-k' neighbors.
  - Select those above a threshold.
- Assesses marginal instead of conditional dependence:
  - 'true' neighbors may not have highest marginal dependence.
  - all variables may be marginally dependent in sparse graphs.

Constraint-based Methods 2: Conditional Independence

- More advanced methods use conditional independence tests. [Verman & Pearl, 1990, Spirtes and Glymour, 1991]
- In some cases, these methods recover the true structure.
- However, there are several practical drawbacks:
  - Number and size of possible conditioning sets is exponential.
  - Multiple testing gives low statistical power.
  - Potential for propagation of errors.
  - Tests don't assess ability of structure to model the data.
- Modern methods alleviate these, but aren't the focus of talk.

Constraint-based Methods 2: Conditional Independence

- More advanced methods use conditional independence tests. [Verman & Pearl, 1990, Spirtes and Glymour, 1991]
- In some cases, these methods recover the true structure.
- However, there are several practical drawbacks:
  - Number and size of possible conditioning sets is exponential.
  - Multiple testing gives low statistical power.
  - Potential for propagation of errors.
  - Tests don't assess ability of structure to model the data.
- Modern methods alleviate these, but aren't the focus of talk.

Constraint-based Methods 2: Conditional Independence

- More advanced methods use conditional independence tests. [Verman & Pearl, 1990, Spirtes and Glymour, 1991]
- In some cases, these methods recover the true structure.
- However, there are several practical drawbacks:
  - Number and size of possible conditioning sets is exponential.
  - Multiple testing gives low statistical power.
  - Potential for propagation of errors.
  - Tests don't assess ability of structure to model the data.
- Modern methods alleviate these, but aren't the focus of talk.

Constraint-based Methods 2: Conditional Independence

- More advanced methods use conditional independence tests. [Verman & Pearl, 1990, Spirtes and Glymour, 1991]
- In some cases, these methods recover the true structure.
- However, there are several practical drawbacks:
  - Number and size of possible conditioning sets is exponential.
  - Multiple testing gives low statistical power.
  - Potential for propagation of errors.
  - Tests don't assess ability of structure to model the data.
- Modern methods alleviate these, but aren't the focus of talk.

Search and Score 1: Greedy Forward/Backward

- Classical search and score methods:
  - Start with the empty structure
  - Add the edge that improves the likelihood the most.
  - Test for sufficient improvement in the likelihood.
  - Stop when the test fails.

  [Dempster, 1972, Goodman, 1971]

  (you can also start with the full structure and work backwards)
- Very expensive in high dimensions:
  - Fits $\mathcal{O}(p^2)$ models at each of $\mathcal{O}(p^2)$ steps.
  - In Gaussian graphical models, fitting model require $\mathcal{O}(p^3)$.

Search and Score 1: Greedy Forward/Backward

- Classical search and score methods:
  - Start with the empty structure
  - Add the edge that improves the likelihood the most.
  - Test for sufficient improvement in the likelihood.
  - Stop when the test fails.

  [Dempster, 1972, Goodman, 1971]

  (you can also start with the full structure and work backwards)
- Very expensive in high dimensions:
  - Fits $\mathcal{O}(p^2)$ models at each of $\mathcal{O}(p^2)$ steps.
  - In Gaussian graphical models, fitting model require $\mathcal{O}(p^3)$.

Search and Score 1: Greedy Forward/Backward

- Classical search and score methods:
  - Start with the empty structure
  - Add the edge that improves the likelihood the most.
  - Test for sufficient improvement in the likelihood.
  - Stop when the test fails.

  [Dempster, 1972, Goodman, 1971]

  (you can also start with the full structure and work backwards)
- Very expensive in high dimensions:
  - Fits $\mathcal{O}(p^2)$ models at each of $\mathcal{O}(p^2)$ steps.
  - In Gaussian graphical models, fitting model require $\mathcal{O}(p^3)$.

Search and Score 2: Restricted Model Classes

- Modern search and score methods:
  - Define a score on structure and parameters.
  - Use combinatorial-search techniques to optimize the score.
  - Consider a restricted class of models (chordal, low treewidth).
  - Use heuristics to approximately evaluate $\mathcal{O}(p^2)$ candidates.
- But these methods still have drawbacks:
  - The search space is enormous, $2^{p(p-1)/2}$ possible models.
  - Each step may still be very expensive, still need to re-fit.
  - Restricted classes may be inefficient or ineffective for modelling some distributions.

Search and Score 2: Restricted Model Classes

- Modern search and score methods:
  - Define a score on structure and parameters.
  - Use combinatorial-search techniques to optimize the score.
  - Consider a restricted class of models (chordal, low treewidth).
  - Use heuristics to approximately evaluate $\mathcal{O}(p^2)$ candidates.
- But these methods still have drawbacks:
  - The search space is enormous, $2^{p(p-1)/2}$ possible models.
  - Each step may still be very expensive, still need to re-fit.
  - Restricted classes may be inefficient or ineffective for modelling some distributions.

Motivation for NOT doing Structure Learning

- Recall the reasons we wanted to do structure learning:
  - Statistical efficiency
  - Computational efficiency
  - Structural discovery
- But, even greedy search methods are extremely expensive.
- A high-dimensional alternative is fit single dense model but:
  - use regularization to improve statistical efficiency
  - use approximations to improve computational efficiency
  - interpret our parameter estimates for structural discovery.

Motivation for NOT doing Structure Learning

- Recall the reasons we wanted to do structure learning:
  - Statistical efficiency
  - Computational efficiency
  - Structural discovery
- But, even greedy search methods are extremely expensive.
- A high-dimensional alternative is fit single dense model but:
  - use regularization to improve statistical efficiency
  - use approximations to improve computational efficiency
  - interpret our parameter estimates for structural discovery.

Motivation for NOT doing Structure Learning

- Recall the reasons we wanted to do structure learning:
  - Statistical efficiency
  - Computational efficiency
  - Structural discovery
- But, even greedy search methods are extremely expensive.
- A high-dimensional alternative is fit single dense model but:
  - use regularization to improve statistical efficiency
  - use approximations to improve computational efficiency
  - interpret our parameter estimates for structural discovery.

Graphical Model Structure Learning with ℓ_1-Regularization

- We focus on an intermediate between fitting a dense and sparse model:
  - Fit a single dense model (possibly with approximations).
  - Use ℓ_1-regularization to encourage parameter sparsity.
- We parameterize the model so that parameter sparsity is equivalent to graph sparsity.
- Estimates a sparse model by fitting a single dense model.

Summary of Contributions

- There has been growing interest in this approach:
  - Gives regularized estimate (like ℓ_2-regularization).
  - Gives sparse estimate (like search methods).
  - Formulated as a convex optimization.
- But previous work usually makes two unrealistic assumptions:
  - Parameters and edges have a one-to-one correspondence.
  - The model only includes pairwise dependencies.
- This talk outlines methods that remove these assumptions.

Summary of Contributions

- There has been growing interest in this approach:
  - Gives regularized estimate (like ℓ_2-regularization).
  - Gives sparse estimate (like search methods).
  - Formulated as a convex optimization.
- But previous work usually makes two unrealistic assumptions:
  - Parameters and edges have a one-to-one correspondence.
  - The model only includes pairwise dependencies.
- This talk outlines methods that remove these assumptions.

Summary of Contributions

- There has been growing interest in this approach:
  - Gives regularized estimate (like ℓ_2-regularization).
  - Gives sparse estimate (like search methods).
  - Formulated as a convex optimization.
- But previous work usually makes two unrealistic assumptions:
  - Parameters and edges have a one-to-one correspondence.
  - The model only includes pairwise dependencies.
- This talk outlines methods that remove these assumptions.

Outline

Pairwise Undirected Graphical Models (UGMs)

- Pairwise UGMs represent multivariate distributions as a normalized product of non-negative potential functions:

$$p(x_1, x_2, \ldots, x_p) = \frac{1}{Z} \prod_{i=1}^{p} \phi_i(x_i) \prod_{(i,j) \in E} \phi_{ij}(x_i, x_j)$$

- Z is the constant that makes the distribution integrate to one.
- Models the pairwise statistics of all pairs of variables in E.

Continuous Structure Learning in UGMs

- Pairwise UGMs represent multivariate distributions as a normalized product of non-negative potentials functions:

$$p(x_1, x_2, \ldots, x_p) = \frac{1}{Z} \prod_{i=1}^{p} \phi_i(x_i) \prod_{(i,j) \in E} \phi_{ij}(x_i, x_j)$$

- Structure learning is the task of choosing the edge set E.
- Removing the edge is the same as setting $\phi_{ij}(x_i, x_j) = 1, \forall_{ij}$.
- We parameterize so that zero parameters make $\phi_{ij}(x_i, x_j) = 1$.
- This lets us perform structure learning with ℓ_1-regularization.

Continuous Structure Learning in UGMs

- Pairwise UGMs represent multivariate distributions as a normalized product of non-negative potentials functions:

$$p(x_1, x_2, \ldots, x_p) = \frac{1}{Z} \prod_{i=1}^{p} \phi_i(x_i) \prod_{(i,j) \in E} \phi_{ij}(x_i, x_j)$$

- Structure learning is the task of choosing the edge set E.
- Removing the edge is the same as setting $\phi_{ij}(x_i, x_j) = 1, \forall_{ij}$.
- We parameterize so that zero parameters make $\phi_{ij}(x_i, x_j) = 1$.
- This lets us perform structure learning with ℓ_1-regularization.

Continuous Structure Learning in UGMs

- Pairwise UGMs represent multivariate distributions as a normalized product of non-negative potentials functions:

$$p(x_1, x_2, \dots, x_p) = \frac{1}{Z} \prod_{i=1}^{p} \phi_i(x_i) \prod_{(i,j) \in E} \phi_{ij}(x_i, x_j)$$

- Structure learning is the task of choosing the edge set E.
- Removing the edge is the same as setting $\phi_{ij}(x_i, x_j) = 1, \forall_{ij}$.
- We parameterize so that zero parameters make $\phi_{ij}(x_i, x_j) = 1$.
- This lets us perform structure learning with ℓ_1-regularization.

Optimization with ℓ_1-Regularization

- Various fields are now interested in ℓ_1-regularization:

$$\min_{\mathbf{w}} f(\mathbf{w}) + \sum_{i=1}^{p} \lambda_i |w_i|$$

- There are efficient algorithms for solving this type of problem.
- Under suitable assumptions, yields a sparse solution:
  - Many coefficients w_i are exactly zero.

Optimization with ℓ_1-Regularization

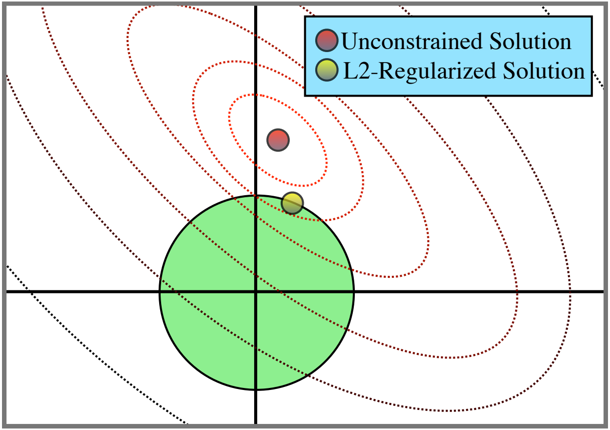
- Various fields are now interested in ℓ_1-regularization:

$$\min_{\mathbf{w}} f(\mathbf{w}) + \sum_{i=1}^{p} \lambda_i |w_i|$$

- There are efficient algorithms for solving this type of problem.
- Under suitable assumptions, yields a sparse solution:
  - Many coefficients w_i are exactly zero.

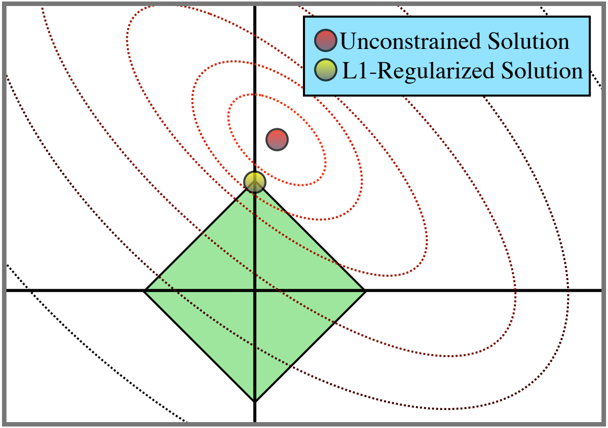
ℓ_2-Regularization vs. ℓ_1-Regularization

ℓ_2-regularization is equivalent to optimization over an ℓ_2-norm ball:



ℓ_2-Regularization vs. ℓ_1-Regularization

ℓ_1-regularization is equivalent to optimization over an ℓ_1-norm ball:



Continuous Variables: Gaussian Graphical Models (GGMs)

- Structure learning with ℓ_1-regularization was first explored for Gaussian graphical models (GGMs).
- GGMs model a multivariate distribution over continuous variables as a multivariate Gaussian distribution:

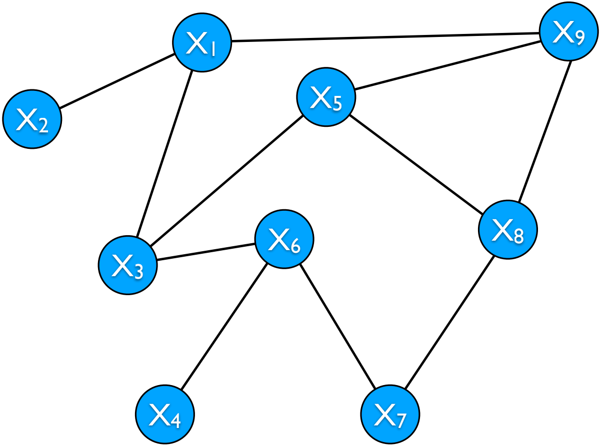
$$p(x_1, x_2, \dots, x_p) = \frac{1}{Z}\exp(-\frac{1}{2}(\mathbf{x}-\mathbf{b})^T W(\mathbf{x}-\mathbf{b}))$$

- The normalizing constant Z is

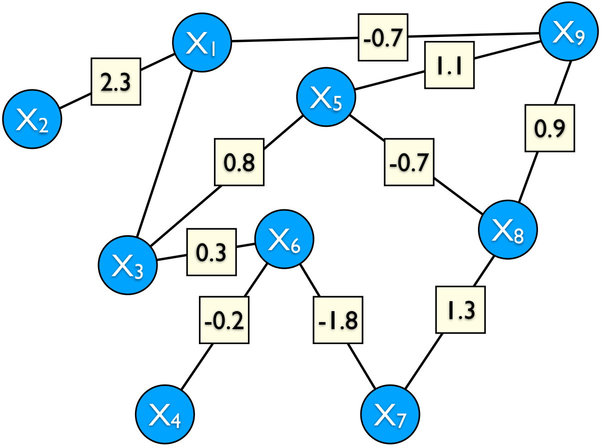
$$Z = (2\pi)^{p/2}|W|^{-1/2}$$

- Edges correspond to non-zero elements of the precision W.

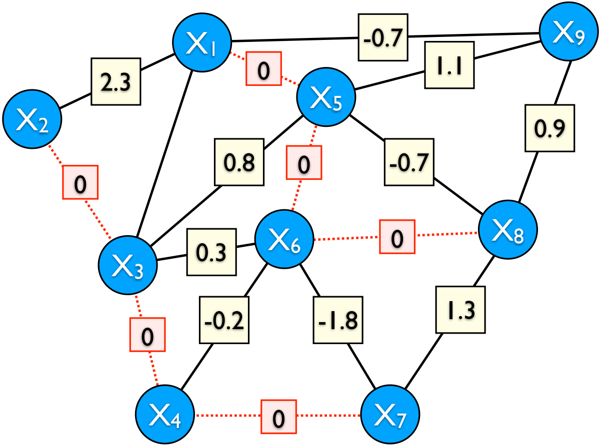
Continuous Variables: Gaussian Graphical Models (GGMs)



Continuous Variables: Gaussian Graphical Models (GGMs)



Continuous Variables: Gaussian Graphical Models (GGMs)



Continuous Variables: Gaussian Graphical Models (GGMs)

- GGM structure learning with ℓ_1-regularization of the precision:

$$\min_{W \succ \mathbf{0}, \mathbf{b}} - \sum_{m=1}^{n} \log p(\mathbf{x}^m | W, \mathbf{b}) + \sum_{i=1}^{p} \sum_{j=1}^{p} \lambda_{ij} |W_{ij}|$$

- First explored in [Dahl et al., 2005, Banerjee et al., 2006, Meinshausen & Buhlmann, 2006, Yuan and Lin, 2007].
- Sometimes called the graphical LASSO.
- Convex optimization is easily solved with 1000s of variables.

Binary Variables: Ising Graphical Models (IGMs)

- This idea was next explored for Ising graphical models:

$$p(x_1, x_2, \dots, x_p) = \frac{1}{Z}\exp(\sum_{i=1}^{p} x_i b_i + \sum_{(i,j)\in E} x_i x_j W_{ij})$$

- The normalizing constant Z is

$$Z = \sum_{\mathbf{x}'}\exp(\sum_{i=1}^{p} x_i' b_i + \sum_{(i,j)\in E} x_i' x_j' W_{ij})$$

- Setting the edge weight W_{ij} to zero removes the edge.
- IGM structure learning with ℓ_1-regularization:

$$\min_{W,\mathbf{b}} -\sum_{m=1}^{n}\log p(\mathbf{x}^m|W,\mathbf{b}) + \sum_{i=1}^{p}\sum_{j=1}^{p}\lambda_{ij}|W_{ij}|$$

Binary Variables: Ising Graphical Models (IGMs)

- This idea was next explored for Ising graphical models:

$$p(x_1, x_2, \dots, x_p) = \frac{1}{Z}\exp(\sum_{i=1}^{p} x_i b_i + \sum_{(i,j)\in E} x_i x_j W_{ij})$$

- The normalizing constant Z is

$$Z = \sum_{\mathbf{x}'} \exp(\sum_{i=1}^{p} x_i' b_i + \sum_{(i,j)\in E} x_i' x_j' W_{ij})$$

- Setting the edge weight W_{ij} to zero removes the edge.
- IGM structure learning with ℓ_1-regularization:

$$\min_{W,\mathbf{b}} -\sum_{m=1}^{n} \log p(\mathbf{x}^m | W, \mathbf{b}) + \sum_{i=1}^{p}\sum_{j=1}^{p} \lambda_{ij} |W_{ij}|$$

Approximations for IGMs

- IGM case is more difficult than GGM case because of Z:
  - Z can be computed in $\mathcal{O}(p^3)$ for GGMs
  - In general, it is #P-hard to evaluate Z in IGMs.
- Several ways to address this have been explored:
  - Asymmetric pseudo-likelihood [Wainwright et al., 2006].
  - Bethe approximation [Lee et al., 2006].
  - Symmetric pseudo-likelihood [Schmidt et al., 2008].
  - Mean-field approximation, convex Bethe approximation.
  - Logdet approximation [Banerjee et al., 2008].
  - Cutting-plane refinement [Kolar and Xing, 2008].

Outline

1. Motivation, Classical Methods
2. Gausian and Ising graphical models: ℓ_1-Regularization
3. General pairwise models: Group ℓ_1-Regularization
   - Group-Sparse Models
   - Group ℓ_1-Regularization
   - Experiments
4. High-order models: Structured Sparsity
5. Further Extensions

Structure Learning with Group ℓ_1-Regularization

- In GGMs/IGMs, there is a one-to-one correspondence between parameters and edges.
- In some case, we want sparsity in groups of parameters:
  - General log-linear models [Lee et al., 2006].
  - Blockwise-sparse models [Duchi et al., 2008].
  - Conditional random fields [Schmidt et al., 2008].
- In these cases, we can use group ℓ_1-regularization.

General Pairwise Log-Linear Models

- In log-linear models, the log-potentials are linear functions.
- IGMs are a special case with binary variables.

$$\log \phi_{ij}(x_i, x_j, w_{ij}) = x_i x_j w_{ij}$$

- But log-linear models allow non-binary discrete variables.
- Also useful for (discretized) non-Gaussian continuous data.
- The potentials for an edge between three-state variables:

$$\log \phi_{ij}(\cdot, \cdot, \mathbf{w}_{ij}) = \begin{bmatrix} w_{ij11} & w_{ij12} & w_{ij13} \\ w_{ij21} & w_{ij22} & w_{ij23} \\ w_{ij31} & w_{ij32} & w_{ij33} \end{bmatrix}$$

- We must set all 9 elements to zero to remove the edge.

General Pairwise Log-Linear Models

- In log-linear models, the log-potentials are linear functions.
- IGMs are a special case with binary variables.

$$\log \phi_{ij}(x_i, x_j, w_{ij}) = x_i x_j w_{ij}$$

- But log-linear models allow non-binary discrete variables.
- Also useful for (discretized) non-Gaussian continuous data.
- The potentials for an edge between three-state variables:

$$\log \phi_{ij}(\cdot, \cdot, \mathbf{w}_{ij}) = \begin{bmatrix} w_{ij11} & w_{ij12} & w_{ij13} \\ w_{ij21} & w_{ij22} & w_{ij23} \\ w_{ij31} & w_{ij32} & w_{ij33} \end{bmatrix}$$

- We must set all 9 elements to zero to remove the edge.

General Pairwise Log-Linear Models

- In log-linear models, the log-potentials are linear functions.
- IGMs are a special case with binary variables.

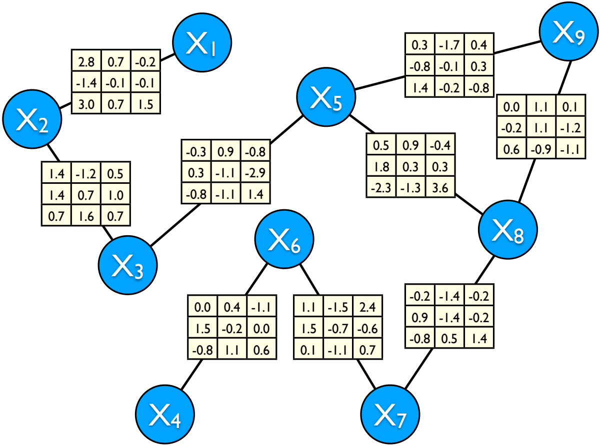
$$\log \phi_{ij}(x_i, x_j, w_{ij}) = x_i x_j w_{ij}$$

- But log-linear models allow non-binary discrete variables.
- Also useful for (discretized) non-Gaussian continuous data.
- The potentials for an edge between three-state variables:

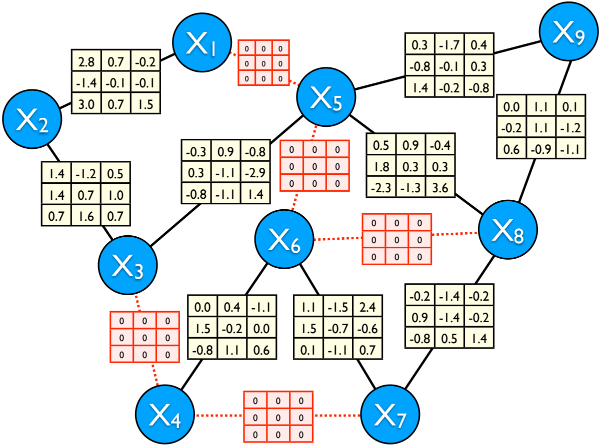
$$\log \phi_{ij}(\cdot, \cdot, \mathbf{w}_{ij}) = \begin{bmatrix} w_{ij11} & w_{ij12} & w_{ij13} \\ w_{ij21} & w_{ij22} & w_{ij23} \\ w_{ij31} & w_{ij32} & w_{ij33} \end{bmatrix}$$

- We must set all 9 elements to zero to remove the edge.

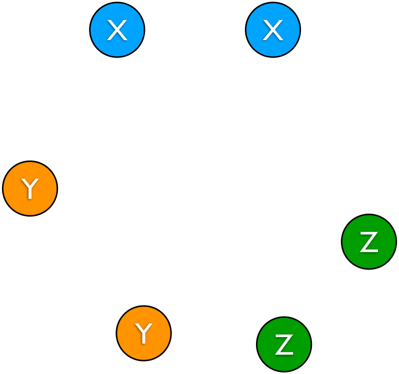
General Pairwise Log-Linear Models



General Pairwise Log-Linear Models

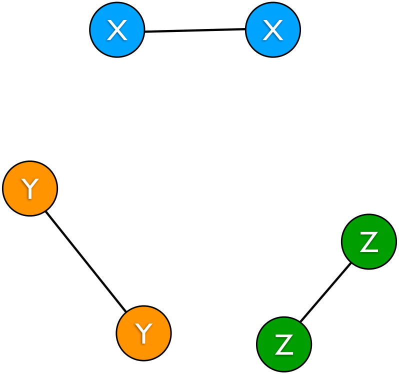


Blockwise Sparsity



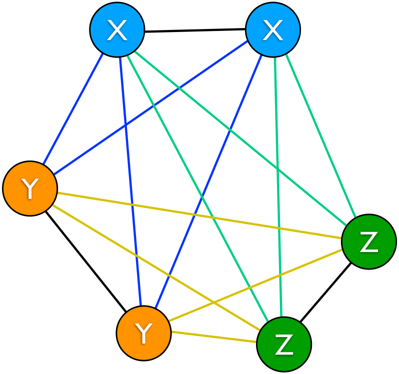
- In blockwise-sparse models, each variable has a type.
- We expect some types to be conditionally independent.

Blockwise Sparsity



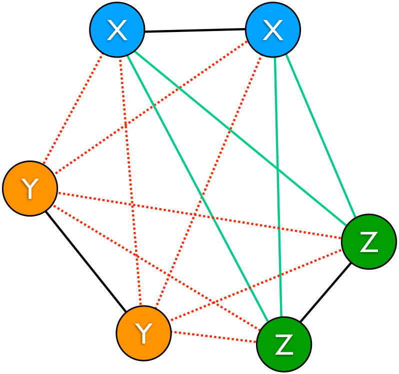
- In blockwise-sparse models, each variable has a type.
- We expect some types to be conditionally independent.

Blockwise Sparsity



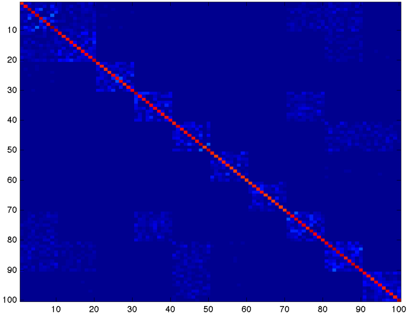
- In blockwise-sparse models, each variable has a type.
- We expect some types to be conditionally independent.

Blockwise Sparsity



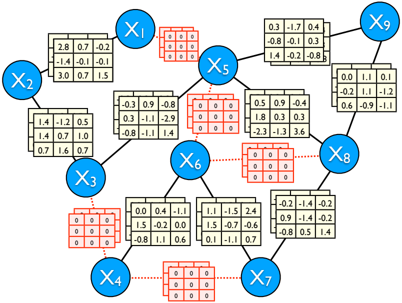
- In blockwise-sparse models, each variable has a type.
- We expect some types to be conditionally independent.

Blockwise Sparsity



- In GGMs/IGMs, corresponds to blockwise-sparsity in matrix.

Conditional Random Fields



- In some scenarios, we also have covariates.
- We can consider doing conditional structure learning.
- Here, we have a tensor of variables associated with each edge.

Group ℓ_1-Regularization

- In all these cases, we want sparsity in groups of parameters.
- This can be accomplished with group ℓ_1-regularization:

$$\min_{\mathbf{w}} f(\mathbf{w}) + \sum_{g} \lambda_g ||\mathbf{w}_g||_2$$

- Applies ℓ_1-regularization to the lengths of the groups.
- An alternative is group ℓ_1-regularization with the ℓ_∞-norm:

$$\min_{\mathbf{w}} f(\mathbf{w}) + \sum_{g} \lambda_g ||\mathbf{w}_g||_\infty$$

- Applies ℓ_1-regularization to the maximums of the groups.

Group ℓ_1-Regularization

- In all these cases, we want sparsity in groups of parameters.
- This can be accomplished with group ℓ_1-regularization:

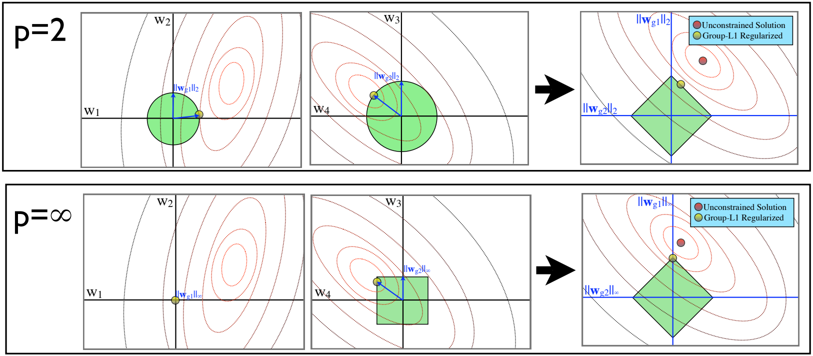
$$\min_{\mathbf{w}} f(\mathbf{w}) + \sum_g \lambda_g ||\mathbf{w}_g||_2$$

- Applies ℓ_1-regularization to the lengths of the groups.
- An alternative is group ℓ_1-regularization with the ℓ_∞-norm:

$$\min_{\mathbf{w}} f(\mathbf{w}) + \sum_g \lambda_g ||\mathbf{w}_g||_\infty$$

- Applies ℓ_1-regularization to the maximums of the groups.

Group ℓ_1-Regularization



Group ℓ_1-Regularization with Matrix Groups

- In several of the examples, the groups form matrices.
- For matrix groups, an alternative is the nuclear norm:

$$\min_{\mathbf{W}_1,\mathbf{W}_2,\dots,\mathbf{W}_G} f(\mathbf{W}_1,\mathbf{W}_2,\dots,\mathbf{W}_G) + \sum_g \lambda_g ||\mathbf{W}_g||_\sigma$$

- The nuclear norm, $||\mathbf{W}_g||_\sigma$, is the sum of singular values.
- Applies ℓ_1-regularization to the singular values of the groups.
- Encourages the matrices to be low-rank.

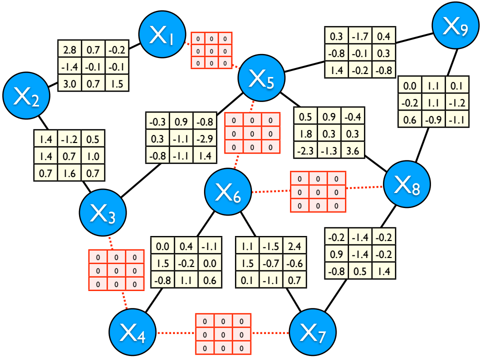
Group ℓ_1-Regularization with Matrix Groups

- In several of the examples, the groups form matrices.
- For matrix groups, an alternative is the nuclear norm:

$$\min_{\mathbf{W}_1,\mathbf{W}_2,\ldots,\mathbf{W}_G} f(\mathbf{W}_1,\mathbf{W}_2,\ldots,\mathbf{W}_G) + \sum_g \lambda_g ||\mathbf{W}_g||_\sigma$$

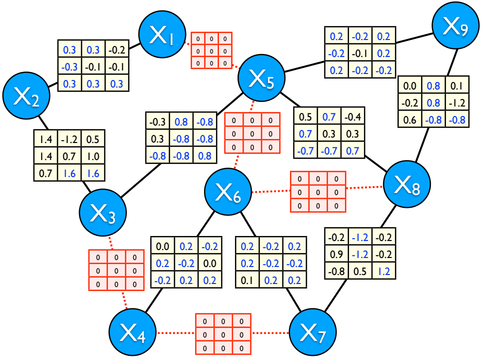
- The nuclear norm, $||\mathbf{W}_g||_\sigma$, is the sum of singular values.
- Applies ℓ_1-regularization to the singular values of the groups.
- Encourages the matrices to be low-rank.

Structure Learning with Group ℓ_1-Regularization



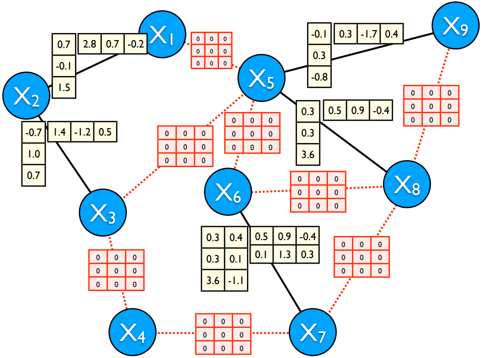
- Group ℓ_1-Regularization with the ℓ_2 group norm.
- Encourage group sparsity.

Structure Learning with Group ℓ_1-Regularization



- Group ℓ_1-Regularization with the ℓ_∞ group norm.
- Encourage group sparsity and parameter tieing.

Structure Learning with Group ℓ_1-Regularization



- Group ℓ_1-Regularization with the nuclear group norm.
- Encourage group sparsity and low-rank.

Experiments Comparing Parameterizations and Norms

- We tested three log-linear edge parameterizations:

$$\log \phi_{ij}(\cdot, \cdot, w_{ij}) = \begin{bmatrix} w_{ij} & 0 & 0 \\ 0 & w_{ij} & 0 \\ 0 & 0 & w_{ij} \end{bmatrix} \quad \text{(Ising potentials)}$$

$$\log \phi_{ij}(\cdot, \cdot, \mathbf{w}_{ij}) = \begin{bmatrix} w_{ij1} & 0 & 0 \\ 0 & w_{ij2} & 0 \\ 0 & 0 & w_{ij3} \end{bmatrix} \quad \text{(gIsing potentials)}$$

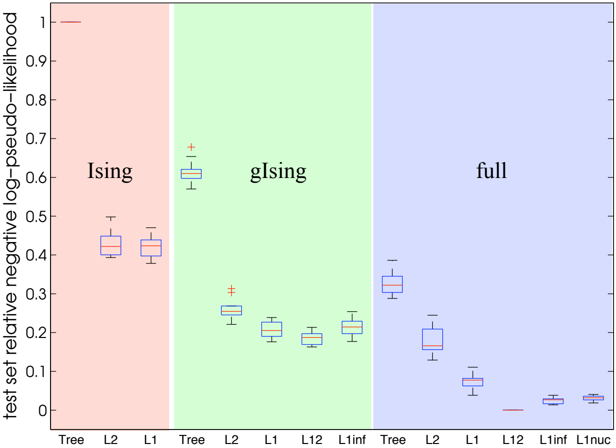
$$\log \phi_{ij}(\cdot, \cdot, \mathbf{w}_{ij}) = \begin{bmatrix} w_{ij11} & w_{ij12} & w_{ij13} \\ w_{ij21} & w_{ij22} & w_{ij23} \\ w_{ij31} & w_{ij32} & w_{ij33} \end{bmatrix} \quad \text{(full potentials)}$$

Experiments Comparing Parameterizations and Norms

- We also tested six regularization strategies:
  - **Tree**: Maximum-likelihood tree structure.
  - **L2**: ℓ_2-Regularization (squared).
  - **L1**: ℓ_1-Regularization.
  - **L12**: Group ℓ_1-Regularization (ℓ_2-norm).
  - **L1inf**: Group ℓ_1-Regularization (ℓ_∞-norm).
  - **L1nuc**: Group ℓ_1-Regularization (nuclear norm).

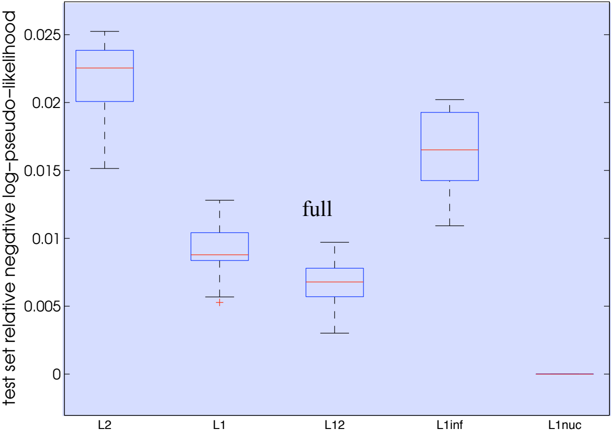
Experimental Comparison of Different Norms

Results on heart wall motion abnormality data (16 nodes, 5 states):



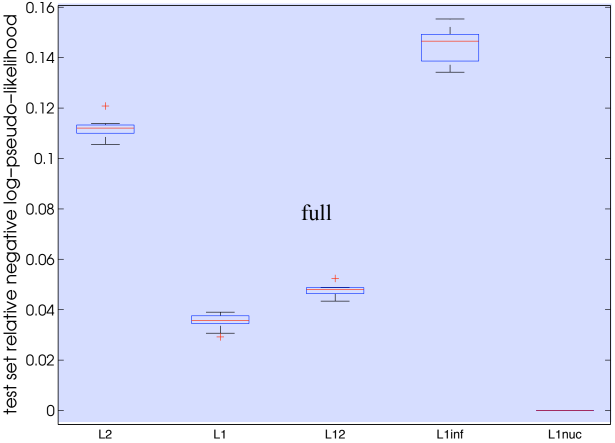
Experimental Comparison of Different Norms

Results on USPS digits data (256 nodes, 4 discretization levels):



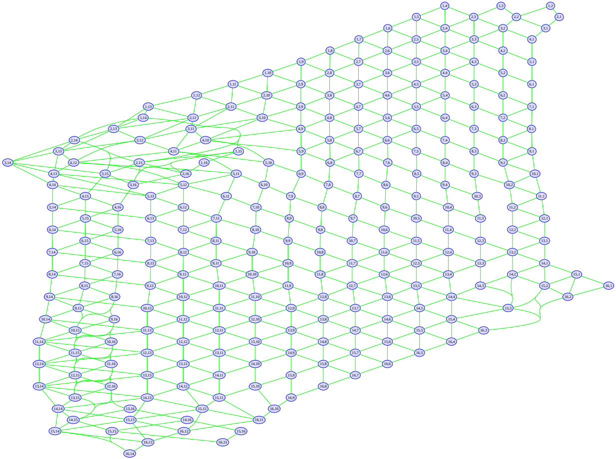
Experimental Comparison of Different Norms

Results on USPS digits data (256 nodes, 8 discretization levels):



Experimental Comparison of Different Norms

Estimated structure on USPS data:



Outline

Structure Learning with ℓ_1-Regularization

A list of papers on this topic (incomplete):

[Li & Yang, 2004], [Li & Yang, 2005], [Banerjee et al., 2006], [Huang et al., 2006], [Lee et al., 2006], [Meinshausen & Bühlmann, 2006], [Wainwright et al., 2006], [Dahinden et al., 2007], [Schmidt et al., 2007], [Shimamura et al., 2007], [Yuan & Lin, 2007], [d' Aspremont et al., 2008], [Banerjee et al., 2008], [Dahl et al., 2008], [Duchi et al., 2008], [Friedman et al., 2008], [Kolar & Xing, 2008], [Levina et al., 2008], [Schmidt et al., 2008], [Fan & Feng, 2009], [Höling & Tibshirani, 2009], [Krishnamurphy & d'Aspremont, 2009], [Lu, 2009a], [Lu, 2009b], [Marlin et al., 2009a], [Marlin et al., 2009b], [Schmidt et al., 2009], [Schmidt & Murphy, 2009], [Schnitzspan et al., 2009], [Yuan, 2009], [Vidaurre et al., 2010].

Structure Learning with ℓ_1-Regularization

Many of these papers have made the pairwise assumption:

[Li & Yang, 2004], [Li & Yang, 2005], [Banerjee et al., 2006], [Huang et al., 2006], [Lee et al., 2006], [Meinshausen & Bühlmann, 2006], [Wainwright et al., 2006], [Dahinden et al., 2007], [Schmidt et al., 2007], [Shimamura et al., 2007], [Yuan & Lin, 2007], [d' Aspremont et al., 2008], [Banerjee et al., 2008], [Dahl et al., 2008], [Duchi et al., 2008], [Friedman et al., 2008], [Kolar & Xing, 2008], [Levina et al., 2008], [Schmidt et al., 2008], [Fan & Feng, 2009], [Höling & Tibshirani, 2009], [Krishnamurphy & d'Aspremont, 2009], [Lu, 2009a], [Lu, 2009b], [Marlin et al., 2009a], [Marlin et al., 2009b], [Schmidt et al., 2009], [Schmidt & Murphy, 2009], [Schnitzspan et al., 2009], [Yuan, 2009], [Vidaurre et al., 2010].

Structure Learning with ℓ_1-Regularization

Many of these papers have made the pairwise assumption:

[Li & Yang, 2004], [Li & Yang, 2005], [Banerjee et al., 2006], [Huang et al., 2006], [Lee et al., 2006], [Meinshausen & Bühlmann, 2006], [Wainwright et al., 2006], [Dahinden et al., 2007], [Schmidt et al., 2007], [Shimamura et al., 2007], [Yuan & Lin, 2007], [d' Aspremont et al., 2008], [Banerjee et al., 2008], [Dahl et al., 2008], [Duchi et al., 2008], [Friedman et al., 2008], [Kolar & Xing, 2008], [Levina et al., 2008], [Schmidt et al., 2008], [Fan & Feng, 2009], [Höling & Tibshirani, 2009], [Krishnamurphy & d'Aspremont, 2009], [Lu, 2009a], [Lu, 2009b], [Marlin et al., 2009a], [Marlin et al., 2009b], [Schmidt et al., 2009], [Schmidt & Murphy, 2009], [Schnitzspan et al., 2009], [Yuan, 2009], [Vidaurre et al., 2010].

Structure Learning with ℓ_1-Regularization

Many of these papers have made the pairwise assumption:

[Li & Yang, 2004], [Li & Yang, 2005], [Banerjee et al., 2006], [Huang et al., 2006], [Lee et al., 2006], [Meinshausen & Bühlmann, 2006], [Wainwright et al., 2006], [Dahinden et al., 2007], [Schmidt et al., 2007], [Shimamura et al., 2007], [Yuan & Lin, 2007], [d' Aspremont et al., 2008], [Banerjee et al., 2008], [Dahl et al., 2008], [Duchi et al., 2008], [Friedman et al., 2008], [Kolar & Xing, 2008], [Levina et al., 2008], [Schmidt et al., 2008], [Fan & Feng, 2009], [Höling & Tibshirani, 2009], [Krishnamurphy & d'Aspremont, 2009], [Lu, 2009a], [Lu, 2009b], [Marlin et al., 2009a], [Marlin et al., 2009b], [Schmidt et al., 2009], [Schmidt & Murphy, 2009], [Schnitzspan et al., 2009], [Yuan, 2009], [Vidaurre et al., 2010].

Structure Learning with ℓ_1-Regularization

Many of these papers have made the pairwise assumption:

[Li & Yang, 2004], [Li & Yang, 2005], [Banerjee et al., 2006], [Huang et al., 2006], [Lee et al., 2006], [Meinshausen & Bühlmann, 2006], [Wainwright et al., 2006], [Dahinden et al., 2007], [Schmidt et al., 2007], [Shimamura et al., 2007], [Yuan & Lin, 2007], [d' Aspremont et al., 2008], [Banerjee et al., 2008], [Dahl et al., 2008], [Duchi et al., 2008], [Friedman et al., 2008], [Kolar & Xing, 2008], [Levina et al., 2008], [Schmidt et al., 2008], [Fan & Feng, 2009], [Höling & Tibshirani, 2009], [Krishnamurphy & d'Aspremont, 2009], [Lu, 2009a], [Lu, 2009b], [Marlin et al., 2009a], [Marlin et al., 2009b], [Schmidt et al., 2009], [Schmidt & Murphy, 2009], [Schnitzspan et al., 2009], [Yuan, 2009], [Vidaurre et al., 2010].

Beyond Pairwise Potentials

- The pairwise assumption is inherent to Gaussian models.
- The pairwise assumption has not traditionally been associated with log-linear models [Goodman, 1971], [Bishop et al., 1975].
- The assumption is restrictive if higher-order statistics matter.
- Eg. Mutations in both gene A and gene B lead to cancer.
- We want to go beyond pairwise potentials.

Beyond Pairwise Potentials

- The pairwise assumption is inherent to Gaussian models.
- The pairwise assumption has not traditionally been associated with log-linear models [Goodman, 1971], [Bishop et al., 1975].
- The assumption is restrictive if higher-order statistics matter.
- Eg. Mutations in both gene A and gene B lead to cancer.
- We want to go beyond pairwise potentials.

Beyond Pairwise Potentials

- The pairwise assumption is inherent to Gaussian models.
- The pairwise assumption has not traditionally been associated with log-linear models [Goodman, 1971], [Bishop et al., 1975].
- The assumption is restrictive if higher-order statistics matter.
- Eg. Mutations in both gene A and gene B lead to cancer.
- We want to go beyond pairwise potentials.

Beyond Pairwise Potentials

- The pairwise assumption is inherent to Gaussian models.
- The pairwise assumption has not traditionally been associated with log-linear models [Goodman, 1971], [Bishop et al., 1975].
- The assumption is restrictive if higher-order statistics matter.
- Eg. Mutations in both gene A and gene B lead to cancer.
- We want to go beyond pairwise potentials.

General Log-Linear Models

In log-linear models [Bishop et al., 1975] we write the probability of a vector $\mathbf{x} \in \{1, 2, \ldots, k\}^p$ as a normalized product

$$p(\mathbf{x}) \triangleq \frac{1}{Z} \prod_{A \subseteq S} \phi_A(\mathbf{x}_A),$$

over each subset A of $S \triangleq \{1, 2, \ldots, p\}$,
(except the null set)

We consider gIsing and full parameterizations of these potentials.

General Log-Linear Models

In log-linear models [Bishop et al., 1975] we write the probability of a vector $\mathbf{x} \in \{1, 2, \ldots, k\}^p$ as a normalized product

$$p(\mathbf{x}) \triangleq \frac{1}{Z} \prod_{A \subseteq S} \phi_A(\mathbf{x}_A),$$

over each subset A of $S \triangleq \{1, 2, \ldots, p\}$,
(except the null set)

We consider gIsing and full parameterizations of these potentials.

General Log-Linear Models

The full parameterization for a threeway potential on binary nodes,

$$\begin{aligned}\log \phi_{ijk}(\mathbf{x}_{ijk}) &= \mathbb{I}(x_i = 1, x_j = 1, x_k = 1) w_{ijk111} + \mathbb{I}(x_i = 1, x_j = 1, x_k = 2) w_{ijk112} \\ &+ \mathbb{I}(x_i = 1, x_j = 2, x_k = 1) w_{ijk121} + \mathbb{I}(x_i = 1, x_j = 2, x_k = 2) w_{ijk122} \\ &+ \mathbb{I}(x_i = 2, x_j = 1, x_k = 1) w_{ijk211} + \mathbb{I}(x_i = 2, x_j = 1, x_k = 2) w_{ijk212} \\ &+ \mathbb{I}(x_i = 2, x_j = 2, x_k = 1) w_{ijk221} + \mathbb{I}(x_i = 2, x_j = 2, x_k = 2) w_{ijk222}.\end{aligned}$$

$\phi_A(\mathbf{x}_A)$ has $k^{|A|}$ parameters $\mathbf{w}_A$.

Setting $\mathbf{w}_A = \mathbf{0}$ is equivalent to removing the potential.

In pairwise models we assume $\mathbf{w}_A = \mathbf{0}$ if $|A| > 2$.

General Log-Linear Models

The full parameterization for a threeway potential on binary nodes,

$$\begin{aligned}\log \phi_{ijk}(\mathbf{x}_{ijk}) = {} & \mathbb{I}(x_i = 1, x_j = 1, x_k = 1)w_{ijk111} + \mathbb{I}(x_i = 1, x_j = 1, x_k = 2)w_{ijk112} \\ & + \mathbb{I}(x_i = 1, x_j = 2, x_k = 1)w_{ijk121} + \mathbb{I}(x_i = 1, x_j = 2, x_k = 2)w_{ijk122} \\ & + \mathbb{I}(x_i = 2, x_j = 1, x_k = 1)w_{ijk211} + \mathbb{I}(x_i = 2, x_j = 1, x_k = 2)w_{ijk212} \\ & + \mathbb{I}(x_i = 2, x_j = 2, x_k = 1)w_{ijk221} + \mathbb{I}(x_i = 2, x_j = 2, x_k = 2)w_{ijk222}.\end{aligned}$$

$\phi_A(\mathbf{x}_A)$ has $k^{|A|}$ parameters $\mathbf{w}_A$.

Setting $\mathbf{w}_A = \mathbf{0}$ is equivalent to removing the potential.

In pairwise models we assume $\mathbf{w}_A = \mathbf{0}$ if $|A| > 2$.

General Log-Linear Models

The full parameterization for a threeway potential on binary nodes,

$$
\begin{aligned}
\log \phi_{ijk}(\mathbf{x}_{ijk}) = {} & \mathbb{I}(x_i = 1, x_j = 1, x_k = 1) w_{ijk111} + \mathbb{I}(x_i = 1, x_j = 1, x_k = 2) w_{ijk112} \\
& + \mathbb{I}(x_i = 1, x_j = 2, x_k = 1) w_{ijk121} + \mathbb{I}(x_i = 1, x_j = 2, x_k = 2) w_{ijk122} \\
& + \mathbb{I}(x_i = 2, x_j = 1, x_k = 1) w_{ijk211} + \mathbb{I}(x_i = 2, x_j = 1, x_k = 2) w_{ijk212} \\
& + \mathbb{I}(x_i = 2, x_j = 2, x_k = 1) w_{ijk221} + \mathbb{I}(x_i = 2, x_j = 2, x_k = 2) w_{ijk222}.
\end{aligned}
$$

$\phi_A(\mathbf{x}_A)$ has $k^{|A|}$ parameters $\mathbf{w}_A$.

Setting $\mathbf{w}_A = \mathbf{0}$ is equivalent to removing the potential.

In pairwise models we assume $\mathbf{w}_A = \mathbf{0}$ if $|A| > 2$.

Group ℓ_1-Regularization for General Log-Linear Models

We can extend the work on pairwise models to the general case by solving [Dahinden et al., 2007]:

$$\min_{\mathbf{w}} - \sum_{i=1}^{n} \log p(\mathbf{x}^i | \mathbf{w}) + \sum_{A \subseteq S} \lambda_A ||\mathbf{w}_A||_2,$$

However,

- Sparsity in the groups A does not correspond to conditional independence.
- Without a cardinality restriction, we have an exponential number of variables.

Group ℓ_1-Regularization for General Log-Linear Models

We can extend the work on pairwise models to the general case by solving [Dahinden et al., 2007]:

$$\min_{\mathbf{w}} -\sum_{i=1}^{n} \log p(\mathbf{x}^i|\mathbf{w}) + \sum_{A \subseteq S} \lambda_A ||\mathbf{w}_A||_2,$$

However,

- Sparsity in the groups A does not correspond to conditional independence.
- Without a cardinality restriction, we have an exponential number of variables.

Hierarchical Log-Linear Models

Instead of using a cardinality restriction, we use:

Hierarchical Inclusion Restriction:
If $\mathbf{w}_A = \mathbf{0}$ and $A \subset B$, then $\mathbf{w}_B = \mathbf{0}$.

We can only have $(1,2,3)$ if we also have $(1,2)$, $(1,3)$, and $(2,3)$.

Hierarchical Log-Linear Models

Instead of using a cardinality restriction, we use:

Hierarchical Inclusion Restriction:
If $\mathbf{w}_A = \mathbf{0}$ and $A \subset B$, then $\mathbf{w}_B = \mathbf{0}$.

We can only have $(1, 2, 3)$ if we also have $(1, 2)$, $(1, 3)$, and $(2, 3)$.

Hierarchical Log-Linear Models

- This is the well-known class of hierarchical log-linear models [Bishop et al., 1975].
- Much larger than the set of pairwise models.
- Can represent any positive distribution.
- Group-sparsity corresponds to conditional independence.
- But, we can't enforce the hierarchical constraint with (disjoint) group ℓ_1-regularization.

Hierarchical Log-Linear Models

- This is the well-known class of hierarchical log-linear models [Bishop et al., 1975].
- Much larger than the set of pairwise models.
- Can represent any positive distribution.
- Group-sparsity corresponds to conditional independence.
- But, we can't enforce the hierarchical constraint with (disjoint) group ℓ_1-regularization.

Hierarchical Log-Linear Models

- This is the well-known class of hierarchical log-linear models [Bishop et al., 1975].
- Much larger than the set of pairwise models.
- Can represent any positive distribution.
- Group-sparsity corresponds to conditional independence.
- But, we can't enforce the hierarchical constraint with (disjoint) group ℓ_1-regularization.

Hierarchical Log-Linear Models

- This is the well-known class of hierarchical log-linear models [Bishop et al., 1975].
- Much larger than the set of pairwise models.
- Can represent any positive distribution.
- Group-sparsity corresponds to conditional independence.
- But, we can't enforce the hierarchical constraint with (disjoint) group ℓ_1-regularization.

Structured Sparsity for Hierarchical Constraints

Bach [2008], Zhao et al. [2009] enforce hierarchical inclusion restrictions with overlapping group ℓ_1-regularization.
(also known as structured sparsity)

Example:

- We can enforce that B is zero whenever A is zero by using two groups: $\{B\}$ and $\{A, B\}$.
- The resulting regularizer is $\lambda_B||\mathbf{w}_B||_2 + \lambda_{A,B}||\mathbf{w}_{A,B}||_2$

Structured Sparsity for Hierarchical Constraints

Bach [2008], Zhao et al. [2009] enforce hierarchical inclusion restrictions with overlapping group ℓ_1-regularization.
(also known as structured sparsity)

Example:

- We can enforce that B is zero whenever A is zero by using two groups: $\{B\}$ and $\{A, B\}$.
- The resulting regularizer is $\lambda_B||\mathbf{w}_B||_2 + \lambda_{A,B}||\mathbf{w}_{A,B}||_2$

Structured Sparsity for Hierarchical Log-Linear Models

We can learn hierarchical log-linear models by solving

$$\min_{\mathbf{w}} -\sum_{i=1}^{n} \log p(\mathbf{x}^i|\mathbf{w}) + \sum_{A \subseteq S} \lambda_A \Big(\sum_{\{B|A\subseteq B\}} ||\mathbf{w}_B||_2^2 \Big)^{1/2}.$$

Under reasonable assumptions, a minimizer of this convex optimization problem will satisfy hierarchical inclusion.

Structured Sparsity for Hierarchical Log-Linear Models

We can learn hierarchical log-linear models by solving

$$\min_{\mathbf{w}} -\sum_{i=1}^{n} \log p(\mathbf{x}^i|\mathbf{w}) + \sum_{A\subseteq S} \lambda_A \Big(\sum_{\{B|A\subseteq B\}} ||\mathbf{w}_B||_2^2 \Big)^{1/2}.$$

Under reasonable assumptions, a minimizer of this convex optimization problem will satisfy hierarchical inclusion.

Active Set Method

- We want to avoid considering the exponential number of possible higher-order potentials.
- We know the solution will be hierarchical, so we propose to only consider groups that satisfy hierarchical inclusion.
- The resulting method guarantees a weak form of global optimality.

Active Set Method

- We want to avoid considering the exponential number of possible higher-order potentials.
- We know the solution will be hierarchical, so we propose to only consider groups that satisfy hierarchical inclusion.
- The resulting method guarantees a weak form of global optimality.

Active Set Method

- We want to avoid considering the exponential number of possible higher-order potentials.
- We know the solution will be hierarchical, so we propose to only consider groups that satisfy hierarchical inclusion.
- The resulting method guarantees a weak form of global optimality.

Active, Inactive, Boundary Groups

- We call A an active group if A or some superset of A is non-zero.
- If A is not active, and some subset of A is zero, we call A an inactive group.
- The remaining groups are called boundary group.
- Boundary groups can be made non-zero without violating hierarchical inclusion.

Active, Inactive, Boundary Groups

- We call A an active group if A or some superset of A is non-zero.
- If A is not active, and some subset of A is zero, we call A an inactive group.
- The remaining groups are called boundary group.
- Boundary groups can be made non-zero without violating hierarchical inclusion.

Active, Inactive, Boundary Groups

- We call A an active group if A or some superset of A is non-zero.
- If A is not active, and some subset of A is zero, we call A an inactive group.
- The remaining groups are called boundary group.
- Boundary groups can be made non-zero without violating hierarchical inclusion.

Active, Inactive, Boundary Groups

- We call A an active group if A or some superset of A is non-zero.
- If A is not active, and some subset of A is zero, we call A an inactive group.
- The remaining groups are called boundary group.
- Boundary groups can be made non-zero without violating hierarchical inclusion.

Active Set Method

Similar to Bach [2008], we use an active set method:

- Find the active groups, and sub-optimal boundary groups.
- Solve the problem with respect to these variables.

This adds groups that satisfy hierarchical inclusion, and where the model poorly estimates the higher-moment in the data.

(analogous to the greedy method of [Gevarter, 1987] for fitting maximum entropy distributions subject to marginal constraints [Cheeseman, 1983]).

Active Set Method

Similar to Bach [2008], we use an active set method:

- Find the active groups, and sub-optimal boundary groups.
- Solve the problem with respect to these variables.

This adds groups that satisfy hierarchical inclusion, and where the model poorly estimates the higher-moment in the data.

(analogous to the greedy method of [Gevarter, 1987] for fitting maximum entropy distributions subject to marginal constraints [Cheeseman, 1983]).

Active Set Method

Similar to Bach [2008], we use an active set method:

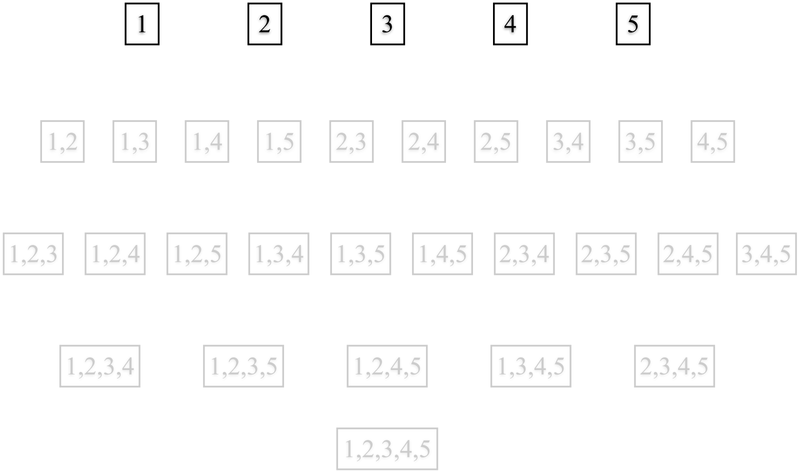
- Find the active groups, and sub-optimal boundary groups.
- Solve the problem with respect to these variables.

This adds groups that satisfy hierarchical inclusion, and where the model poorly estimates the higher-moment in the data.

(analogous to the greedy method of [Gevarter, 1987] for fitting maximum entropy distributions subject to marginal constraints [Cheeseman, 1983]).

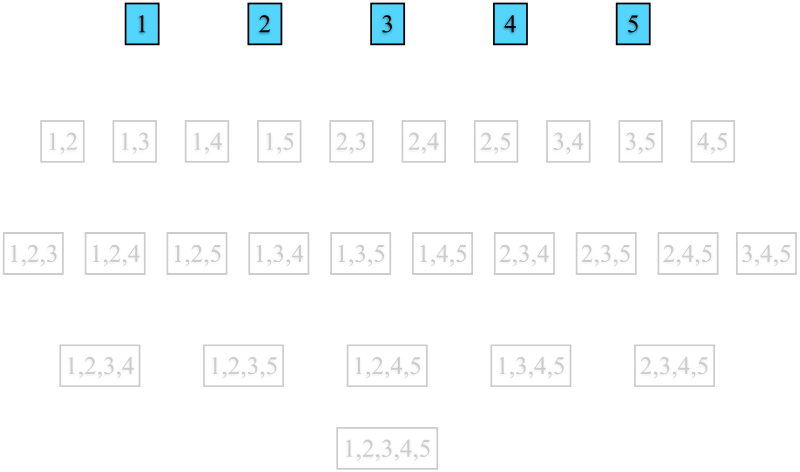
Example of Active Set Method

Initial boundary groups.



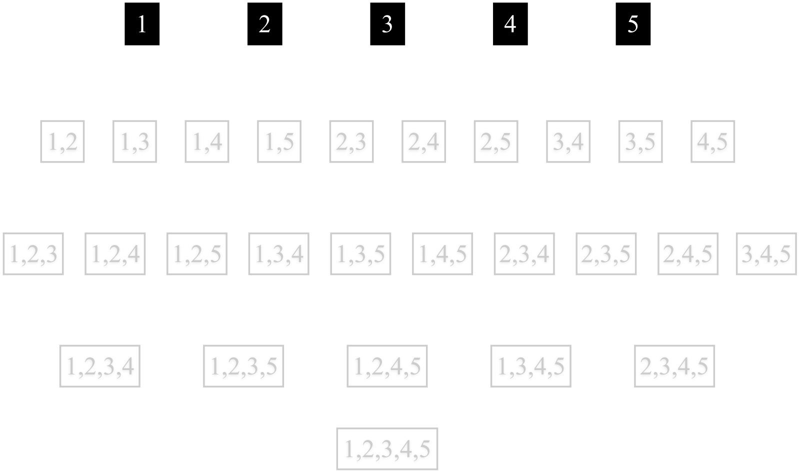
Example of Active Set Method

Optimize initial boundary groups.



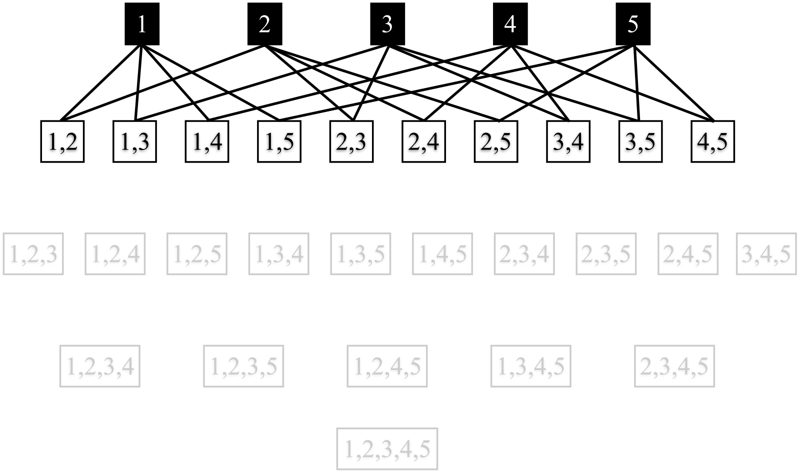
Example of Active Set Method

Find new **active groups**.



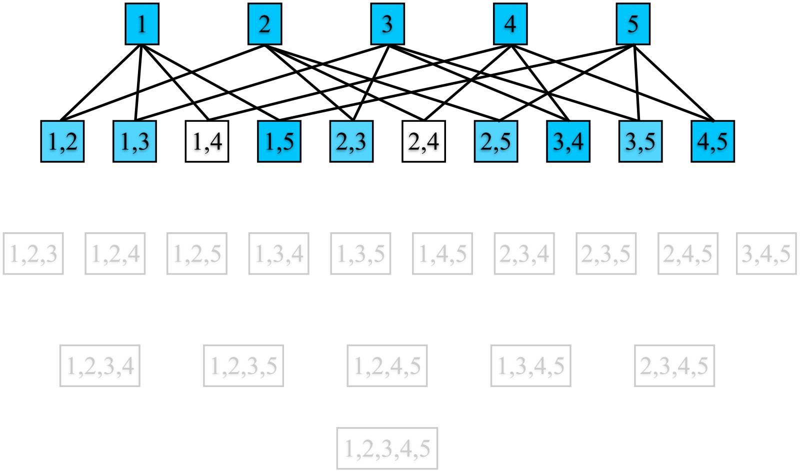
Example of Active Set Method

Find new boundary groups.



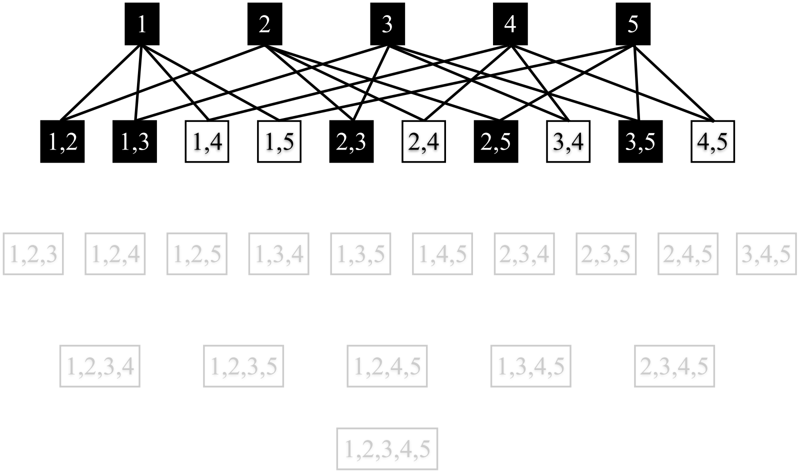
Example of Active Set Method

Optimize active groups and sub-optimal boundary groups.



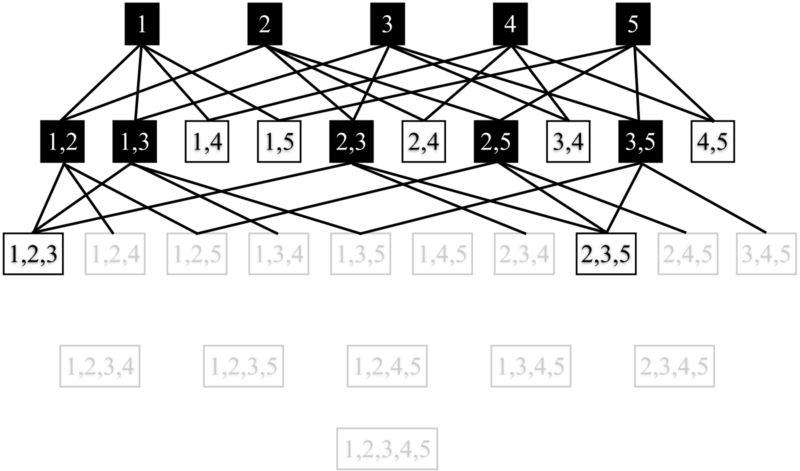
Example of Active Set Method

Find new **active groups**.



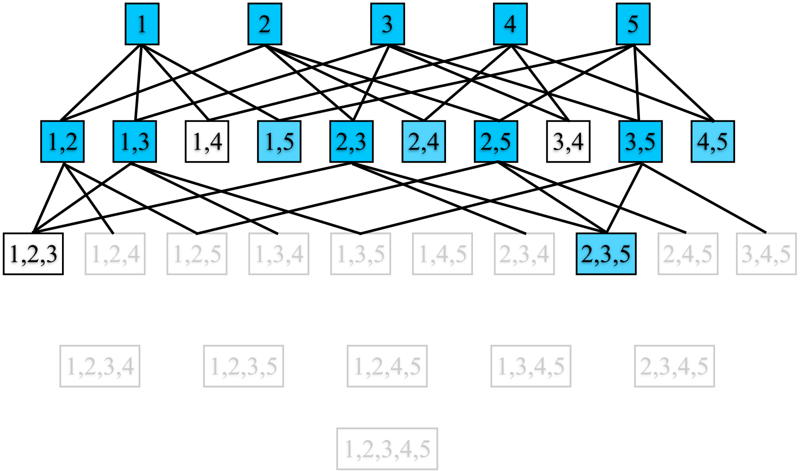
Example of Active Set Method

Find new boundary groups.



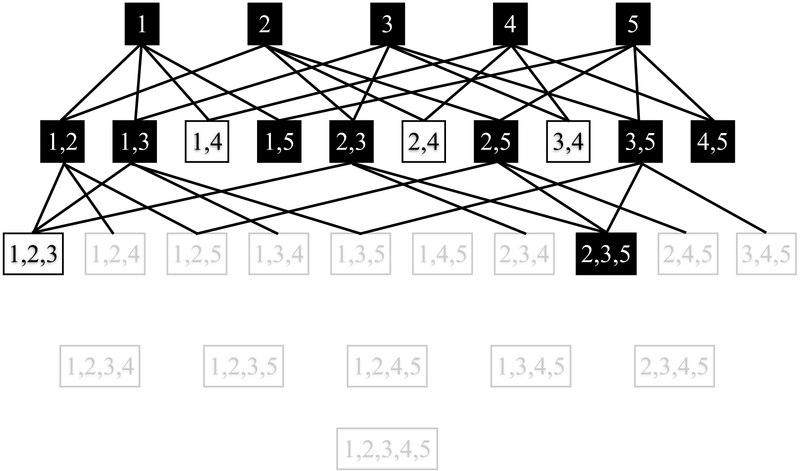
Example of Active Set Method

Optimize active groups and sub-optimal boundary groups.



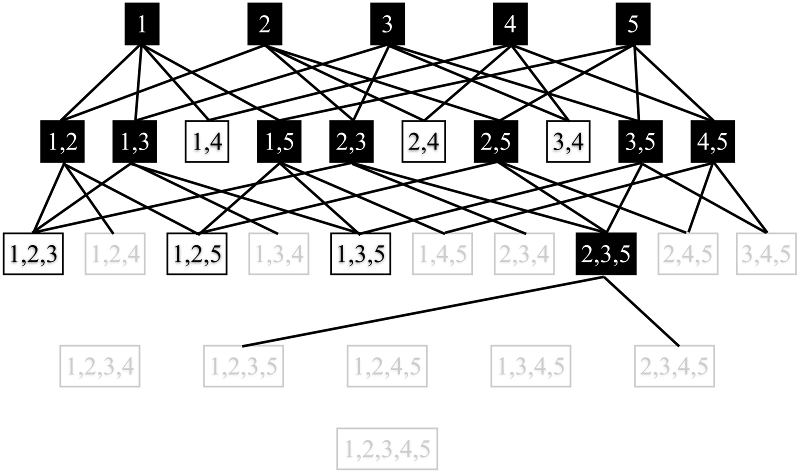
Example of Active Set Method

Find new **active groups**.



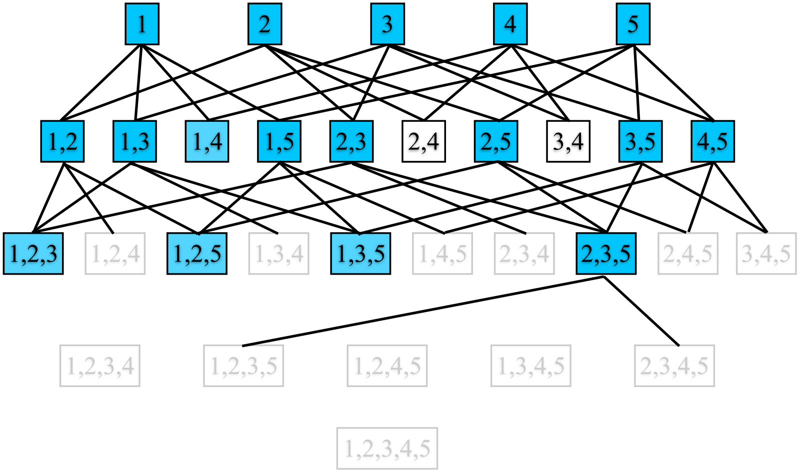
Example of Active Set Method

Find new boundary groups.



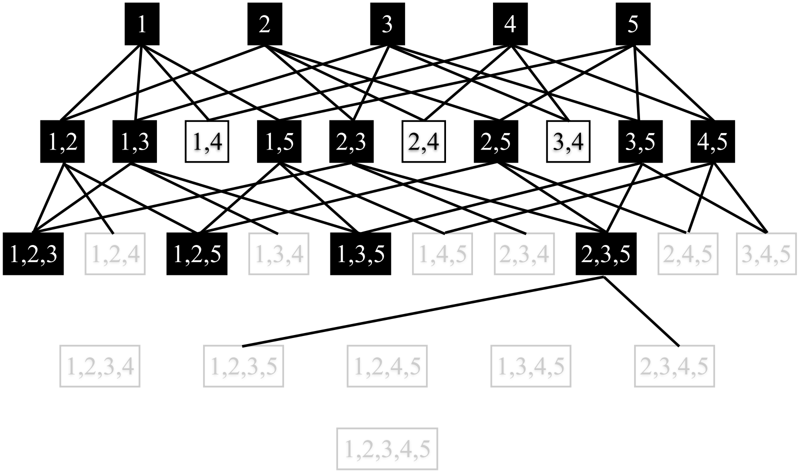
Example of Active Set Method

Optimize active groups and sub-optimal boundary groups.



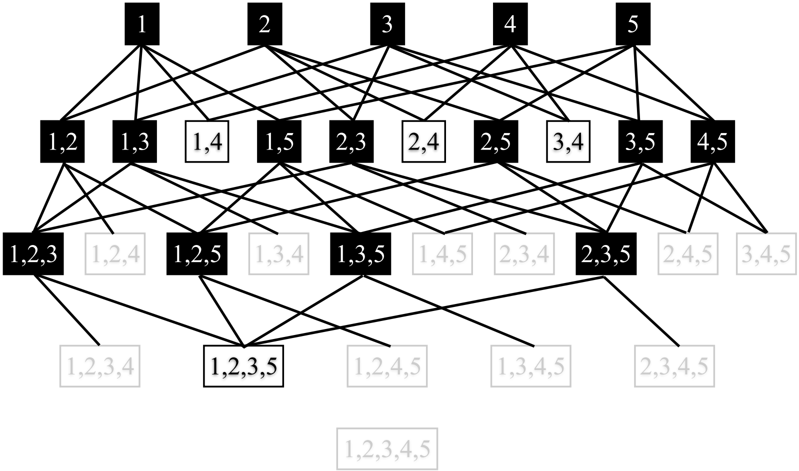
Example of Active Set Method

Find new **active groups**.



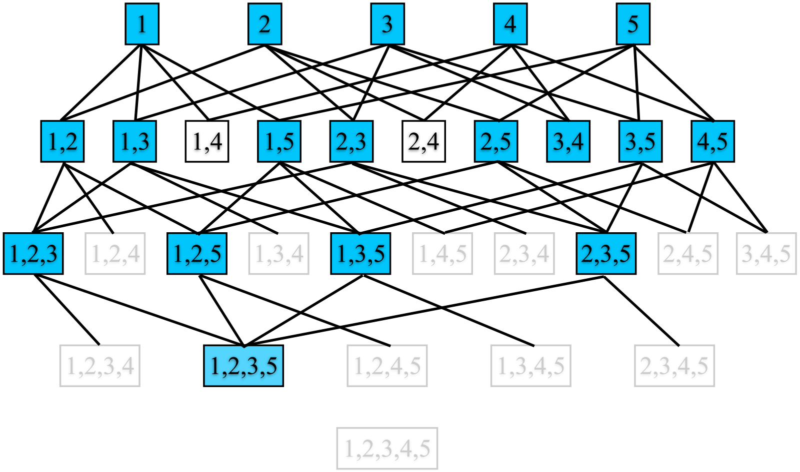
Example of Active Set Method

Find new boundary groups.



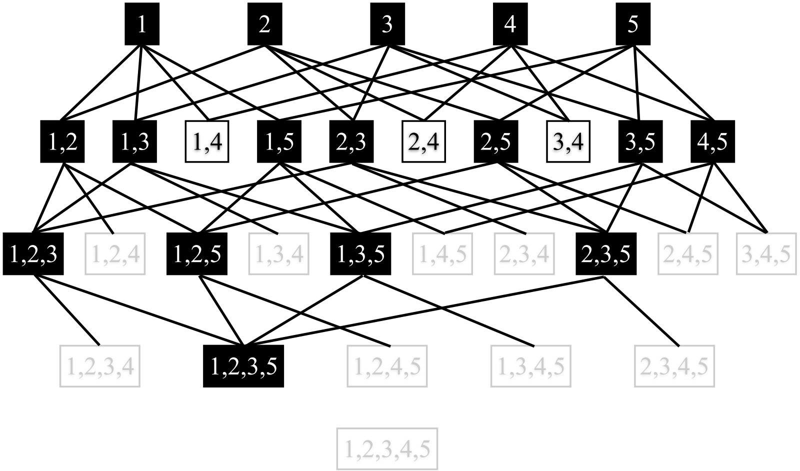
Example of Active Set Method

Optimize active groups and sub-optimal boundary groups.



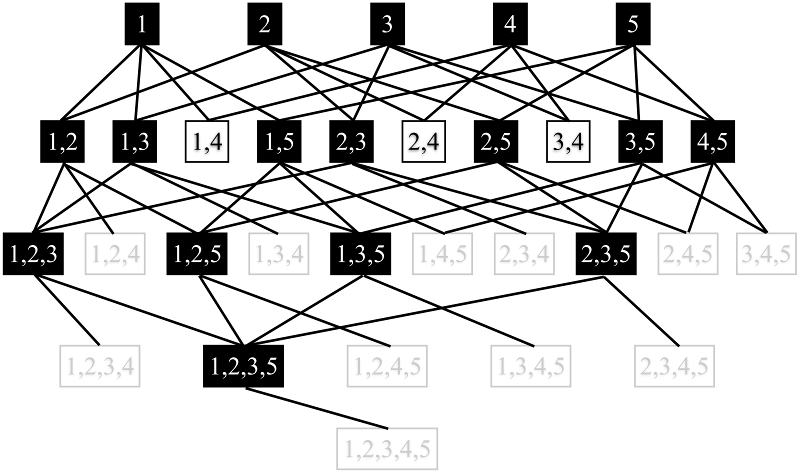
Example of Active Set Method

Find new **active groups**.



Example of Active Set Method

No new boundary groups, so we are done.



Example of Active Set Method

- We only considered 4 of 10 possible threeway interactions, 1 of 5 fourway interactions, and no fiveway interactions.
- The active set method can save us from looking at an exponential number of higher-order factors.

Example of Active Set Method

- We only considered 4 of 10 possible threeway interactions, 1 of 5 fourway interactions, and no fiveway interactions.
- The active set method can save us from looking at an exponential number of higher-order factors.

Multivariate Flow Cytometry Experiments

Does it empirically help to have higher-order potentials?

We first consider a small data set where we can tractably compute the normalizing constant:

- Multivariate flow cytometry [Sachs et al., 2005].

We compared:

- Pairwise with ℓ_2-regularization and group ℓ_1-regularization.
- Threeway with ℓ_2-regularization and group ℓ_1-regularization.
- Hierarchical with overlapping group ℓ_1-regularization.

We trained on 1/3, used 1/3 to select λ, and used 1/3 as a test set (for 10 random splits).

Multivariate Flow Cytometry Experiments

Does it empirically help to have higher-order potentials?

We first consider a small data set where we can tractably compute the normalizing constant:

- Multivariate flow cytometry [Sachs et al., 2005].

We compared:

- Pairwise with ℓ_2-regularization and group ℓ_1-regularization.
- Threeway with ℓ_2-regularization and group ℓ_1-regularization.
- Hierarchical with overlapping group ℓ_1-regularization.

We trained on 1/3, used 1/3 to select λ, and used 1/3 as a test set (for 10 random splits).

Multivariate Flow Cytometry Experiments

Does it empirically help to have higher-order potentials?

We first consider a small data set where we can tractably compute the normalizing constant:

- Multivariate flow cytometry [Sachs et al., 2005].

We compared:

- Pairwise with ℓ_2-regularization and group ℓ_1-regularization.
- Threeway with ℓ_2-regularization and group ℓ_1-regularization.
- Hierarchical with overlapping group ℓ_1-regularization.

We trained on 1/3, used 1/3 to select λ, and used 1/3 as a test set (for 10 random splits).

Multivariate Flow Cytometry Experiments

Does it empirically help to have higher-order potentials?

We first consider a small data set where we can tractably compute the normalizing constant:

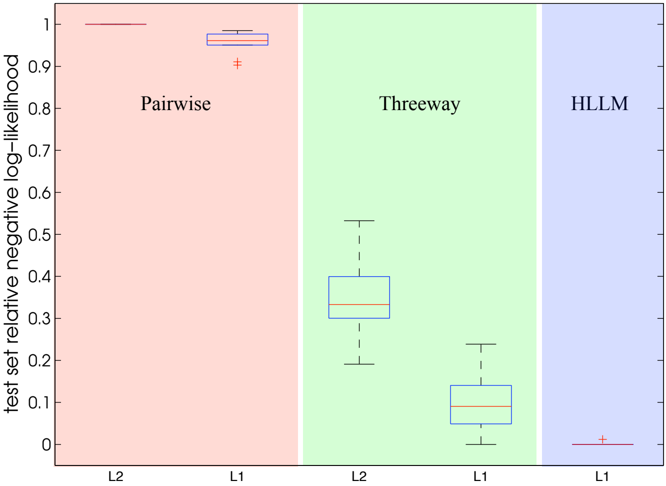
- Multivariate flow cytometry [Sachs et al., 2005].

We compared:

- Pairwise with ℓ_2-regularization and group ℓ_1-regularization.
- Threeway with ℓ_2-regularization and group ℓ_1-regularization.
- Hierarchical with overlapping group ℓ_1-regularization.

We trained on $1/3$, used $1/3$ to select λ, and used $1/3$ as a test set (for 10 random splits).

Flow Cytometry Data



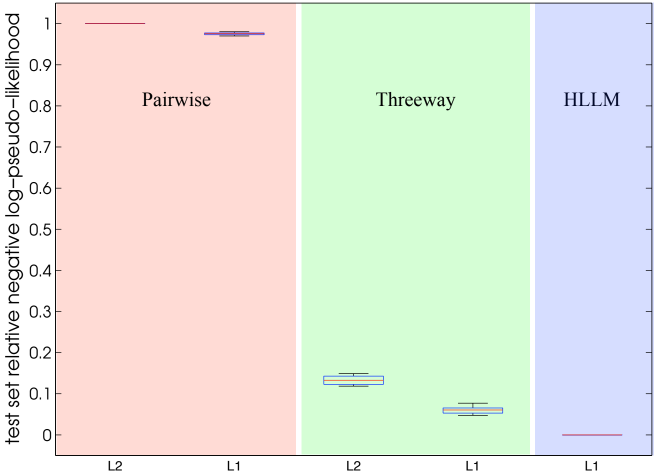
Traffic and USPS Experiments

We next consider two larger data sets:

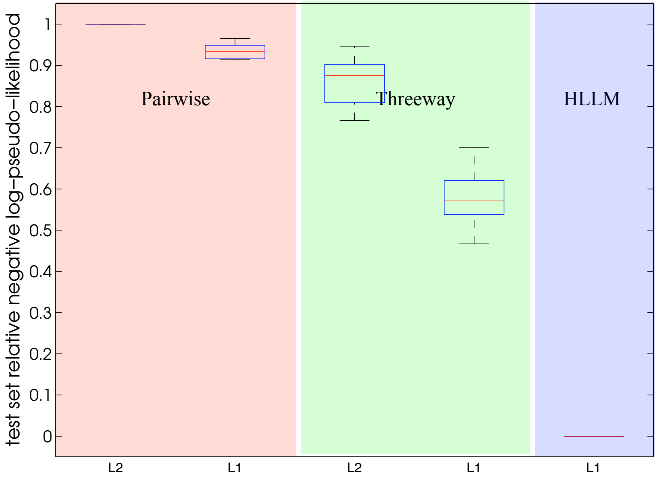
- USPS digits data discretized into four states.
- Traffic flow level [Shahaf et al., 2009].

On these experiments we used gIsing potentials, and used a pseudo-likelihood for training/test.

USPS Data


test set relative negative log−pseudo−likelihood
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Pairwise
Threeway
HLLM
L2
L1
L2
L1
L1

Traffic Flow Data



Structure Estimation

- We sought to test whether the HLLM model could recover a true structure.
- We generated samples from a 10-node data set with potentials $(2,3)(4,5,6)(7,8,9,10)$ and parameters from $\mathcal{N}(0,1)$.
- We recorded the number of false positives of different orders for the first model along the regularization path that includes the true model.
- Eg., with 20000 samples the order was (8,10)(7,9)(9,10)(7,10)(4,5)(8,9)(2,3)(4,6)(8,9,10)(7,8) (7,8,9)(7,8,10)(5,6)(1,8)(5,9)(3,8)(3,7)(4,5,6)(1,7)(7,9,10) (7,8,9,10)

Structure Estimation

- We sought to test whether the HLLM model could recover a true structure.
- We generated samples from a 10-node data set with potentials $(2,3)(4,5,6)(7,8,9,10)$ and parameters from $\mathcal{N}(0,1)$.
- We recorded the number of false positives of different orders for the first model along the regularization path that includes the true model.
- Eg., with 20000 samples the order was (8,10)(7,9)(9,10)(7,10)(4,5)(8,9)(2,3)(4,6)(8,9,10)(7,8)(7,8,9)(7,8,10)(5,6)(1,8)(5,9)(3,8)(3,7)(4,5,6)(1,7)(7,9,10)(7,8,9,10)

Structure Estimation

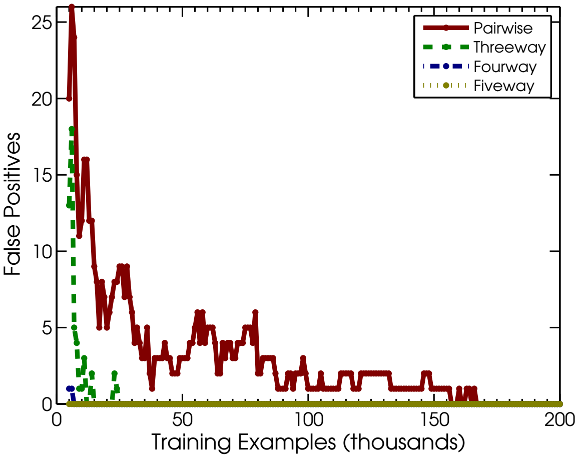
- We sought to test whether the HLLM model could recover a true structure.
- We generated samples from a 10-node data set with potentials $(2, 3)(4, 5, 6)(7, 8, 9, 10)$ and parameters from $\mathcal{N}(0, 1)$.
- We recorded the number of false positives of different orders for the first model along the regularization path that includes the true model.
- Eg., with 20000 samples the order was (8,10)(7,9)(9,10)(7,10)(4,5)(8,9)(2,3)(4,6)(8,9,10)(7,8) (7,8,9)(7,8,10)(5,6)(1,8)(5,9)(3,8)(3,7)(4,5,6)(1,7)(7,9,10) (7,8,9,10)

Structure Estimation

- We sought to test whether the HLLM model could recover a true structure.
- We generated samples from a 10-node data set with potentials $(2,3)(4,5,6)(7,8,9,10)$ and parameters from $\mathcal{N}(0,1)$.
- We recorded the number of false positives of different orders for the first model along the regularization path that includes the true model.
- Eg., with 20000 samples the order was (8,10)(7,9)(9,10)(7,10)(4,5)(8,9)(2,3)(4,6)(8,9,10)(7,8) (7,8,9)(7,8,10)(5,6)(1,8)(5,9)(3,8)(3,7)(4,5,6)(1,7)(7,9,10) (7,8,9,10)

Synthetic Data: Types of Errors

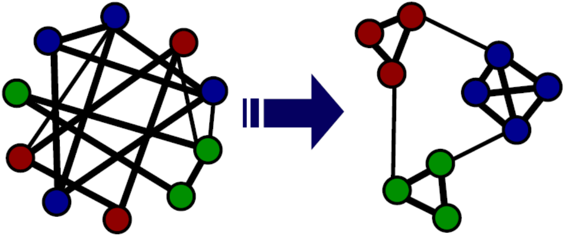
Types of errors made by HLLM:



Outline

Group Sparse Priors for Covariance Estimation

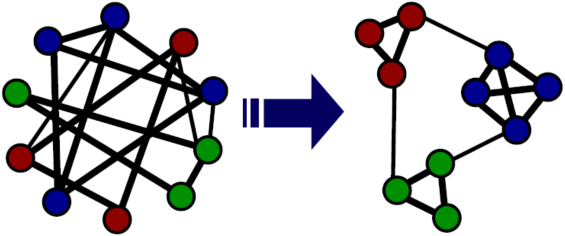
- Earlier we discussed blockwise-sparse models.



- What if the blocks aren't completely sparse?
- What if we don't know the variable types?
- We give bounds on integrals of priors over positive-definite matrices, and a variational method that learns the types. [Marlin, Schmidt, Murphy, 2009]

Group Sparse Priors for Covariance Estimation

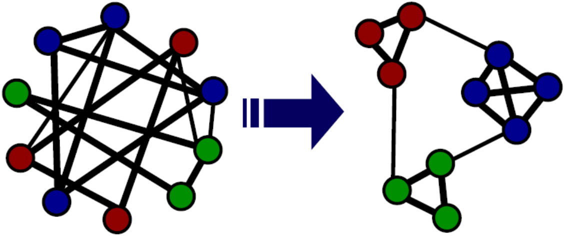
- Earlier we discussed blockwise-sparse models.



- What if the blocks aren't completely sparse?
- What if we don't know the variable types?
- We give bounds on integrals of priors over positive-definite matrices, and a variational method that learns the types. [Marlin, Schmidt, Murphy, 2009]

Group Sparse Priors for Covariance Estimation

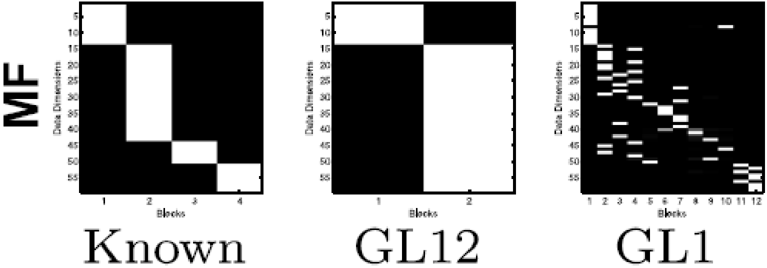
- Earlier we discussed blockwise-sparse models.



- What if the blocks aren't completely sparse?
- What if we don't know the variable types?
- We give bounds on integrals of priors over positive-definite matrices, and a variational method that learns the types. [Marlin, Schmidt, Murphy, 2009]

Group Sparse Priors for Covariance Estimation

Learned variable types on mutual fund data:
[Scott & Carvalho, 2008]



The methods discover the ‘stocks’ and ‘bonds’ groups.

Causality: Modeling Interventions

- The difference between conditioning by observation and conditioning by intervention in the 'hungry at work' problem:
  - If I see that my watch says 11:55, then it's almost lunch time
  - If I set my watch so it says 11:55, it doesn't help
- Without knowing the difference, predictions may be useless.
- Methods that model interventions are typically called causal.

Causality: Modeling Interventions

- The difference between conditioning by observation and conditioning by intervention in the 'hungry at work' problem:
  - If I see that my watch says 11:55, then it's almost lunch time
  - If I set my watch so it says 11:55, it doesn't help
- Without knowing the difference, predictions may be useless.
- Methods that model interventions are typically called causal.

Causality: Modeling Interventions

- The difference between conditioning by observation and conditioning by intervention in the ‘hungry at work’ problem:
  - If I see that my watch says 11:55, then it’s almost lunch time
  - If I set my watch so it says 11:55, it doesn’t help
- Without knowing the difference, predictions may be useless.
- Methods that model interventions are typically called causal.

Causality: Modeling Interventions

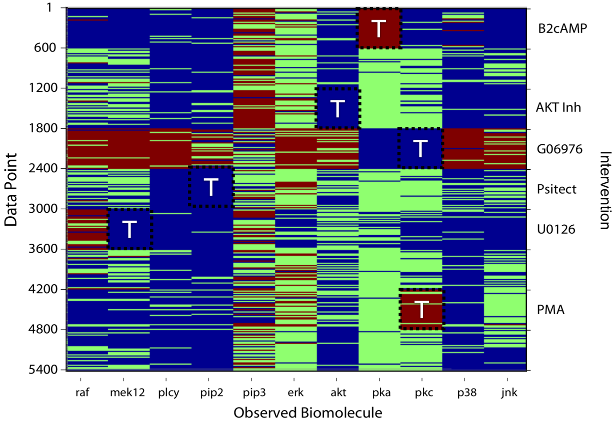
- The difference between conditioning by observation and conditioning by intervention in the 'hungry at work' problem:
  - If I see that my watch says 11:55, then it's almost lunch time
  - If I set my watch so it says 11:55, it doesn't help
- Without knowing the difference, predictions may be useless.
- Methods that model interventions are typically called causal.

Causality: Modeling Interventions

- The difference between conditioning by observation and conditioning by intervention in the 'hungry at work' problem:
  - If I see that my watch says 11:55, then it's almost lunch time
  - If I set my watch so it says 11:55, it doesn't help
- Without knowing the difference, predictions may be useless.
- Methods that model interventions are typically called causal.

Causality: Modeling Interventions

Interventional Cell Signaling Data [Sachs et al., 2005]



Causality: Modeling Interventions

- Causal learning methods are usually evaluated in terms of a 'true' underlying DAG.
- For real data, the structure may not be known, or even a DAG.
- Why not evaluate causal models in terms of modeling the effects of interventions?
- Given this task, there are a variety of approaches to causality.
  [Eaton & Murphy, 2007]
  [Schmidt & Murphy, 2009]
  [Duvenaud, Eaton, Murphy, Schmidt, 2010]

Causality: Modeling Interventions

- Causal learning methods are usually evaluated in terms of a 'true' underlying DAG.
- For real data, the structure may not be known, or even a DAG.
- Why not evaluate causal models in terms of modeling the effects of interventions?
- Given this task, there are a variety of approaches to causality.
  [Eaton & Murphy, 2007]
  [Schmidt & Murphy, 2009]
  [Duvenaud, Eaton, Murphy, Schmidt, 2010]

Causality: Modeling Interventions

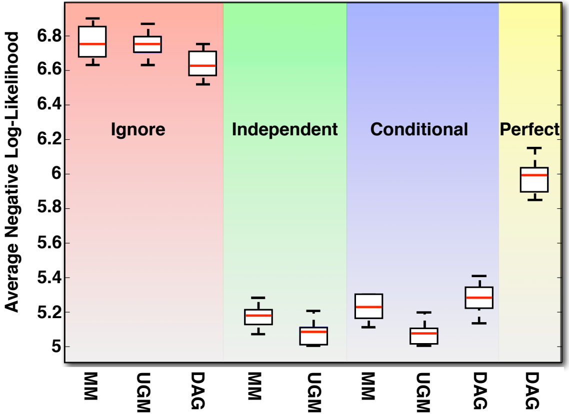
- Causal learning methods are usually evaluated in terms of a 'true' underlying DAG.
- For real data, the structure may not be known, or even a DAG.
- Why not evaluate causal models in terms of modeling the effects of interventions?
- Given this task, there are a variety of approaches to causality.
  [Eaton & Murphy, 2007]
  [Schmidt & Murphy, 2009]
  [Duvenaud, Eaton, Murphy, Schmidt, 2010]

Causality: Modeling Interventions

- Causal learning methods are usually evaluated in terms of a 'true' underlying DAG.
- For real data, the structure may not be known, or even a DAG.
- Why not evaluate causal models in terms of modeling the effects of interventions?
- Given this task, there are a variety of approaches to causality.
  [Eaton & Murphy, 2007]
  [Schmidt & Murphy, 2009]
  [Duvenaud, Eaton, Murphy, Schmidt, 2010]

Causality: Modeling Interventions

Interventional Cell Signaling Data [Sachs et al., 2005]:



Other Selected Extensions

Some topics not discussed:

- The methods can be extended to handle missing data or hidden variables.
- We can consider mixtures of sparse graphical models.
- Stochastic approximation methods allow MCMC for inference.
- Can be used as sub-routines in variational Bayes methods.
- Can be used as sub-routines in consistent estimation methods.
- Methods might be useful for other types of structure learning.
- Non-convex alternatives to ℓ_1-regularization.

Summary

- ℓ_1-Regularization is an appealing approach for graphical model structure learning.
- Prior work focuses on Gaussian and Ising graphical models.
- We considered models with group sparsity:
  - General discrete pairwise models.
  - Blockwise-sparse models.
  - Conditional models.
- We discussed methods for going beyond pairwise potentials.
- Code is on-line (or will be soon).
- Thank you for inviting me!

Summary

- ℓ_1-Regularization is an appealing approach for graphical model structure learning.
- Prior work focuses on Gaussian and Ising graphical models.
- We considered models with group sparsity:
  - General discrete pairwise models.
  - Blockwise-sparse models.
  - Conditional models.
- We discussed methods for going beyond pairwise potentials.
- Code is on-line (or will be soon).
- Thank you for inviting me!

Summary

- ℓ_1-Regularization is an appealing approach for graphical model structure learning.
- Prior work focuses on Gaussian and Ising graphical models.
- We considered models with group sparsity:
  - General discrete pairwise models.
  - Blockwise-sparse models.
  - Conditional models.
- We discussed methods for going beyond pairwise potentials.
- Code is on-line (or will be soon).
- Thank you for inviting me!

Summary

- ℓ_1-Regularization is an appealing approach for graphical model structure learning.
- Prior work focuses on Gaussian and Ising graphical models.
- We considered models with group sparsity:
  - General discrete pairwise models.
  - Blockwise-sparse models.
  - Conditional models.
- We discussed methods for going beyond pairwise potentials.
- Code is on-line (or will be soon).
- Thank you for inviting me!

Summary

- ℓ_1-Regularization is an appealing approach for graphical model structure learning.
- Prior work focuses on Gaussian and Ising graphical models.
- We considered models with group sparsity:
  - General discrete pairwise models.
  - Blockwise-sparse models.
  - Conditional models.
- We discussed methods for going beyond pairwise potentials.
- Code is on-line (or will be soon).
- Thank you for inviting me!

Summary

- ℓ_1-Regularization is an appealing approach for graphical model structure learning.
- Prior work focuses on Gaussian and Ising graphical models.
- We considered models with group sparsity:
  - General discrete pairwise models.
  - Blockwise-sparse models.
  - Conditional models.
- We discussed methods for going beyond pairwise potentials.
- Code is on-line (or will be soon).
- Thank you for inviting me!