CPSC 540: Machine Learning

Undirected Graphical Models

Mark Schmidt

University of British Columbia

Winter 2018

Last Time: Learning and Inference in DAGs

- We discussed learning in DAG models,

$$\log p(X \mid W) = \sum_{i=1}^{n} \sum_{j=1}^{d} \log p(x_j^i \mid x_{\text{pa}(j)}^i, w^j),$$

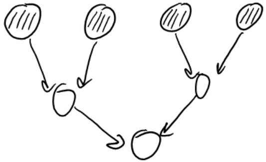
which becomes a supervised learning problem for each feature j.

  - "Tabular" parameterization is common but requires small number of parents.
  - Gaussian belief networks use least squares (defines a multivariate Gaussian).
  - Sigmoid belief networks use logistic regression.

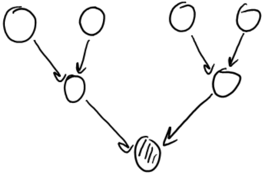
- For inference in DAGs (decoding, computing marginals, computing conditionals):
  - We can use ancestral sampling to compute Monte Carlo approximations.
  - We can apply message passing, but messages may be huge.
    - Only guarantee $O(dk^2)$ cost if each node has at most one parent ("tree" or "forest").

Conditional Sampling in DAGs

- What about conditional sampling in DAGs?
  - Could be easy or hard depending on what we condition on.
- For example, easy if we condition on the first variables in the order:
  - Just fix these and run ancestral sampling.



- Hard to condition on the last variables in the order:
  - Conditioning on descendent makes ancestors dependent.

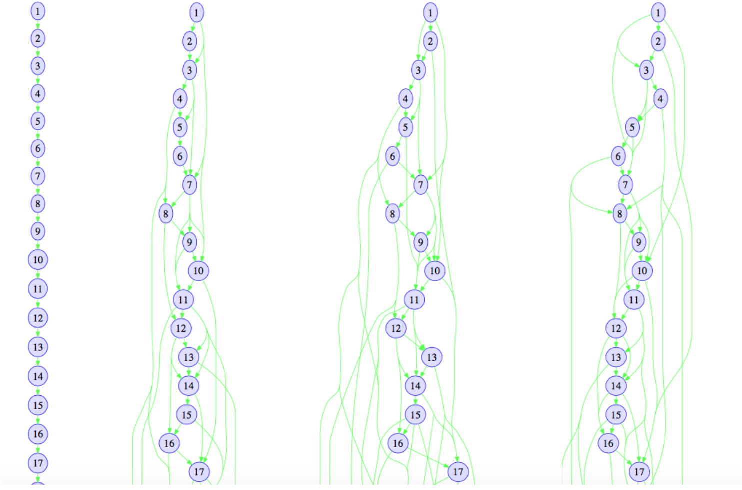


DAG Structure Learning

- Structure learning is the problem of choosing the graph.
  - Input is data X.
  - Output is a graph G.
- The "easy" case is when we're given the ordering of the variables.
  - So the parents of j must be chosen from $\{1, 2, \ldots, j-1\}$.
- Given the ordering, structure learning reduces to feature selection:
  - Select features $\{x_1, x_2, \ldots, x_{j-1}\}$ that best predict "label" x_j.
  - We can use any feature selection method to solve these d problems.

Example: Structure Learning in Rain Data Given Ordering

- Structure learning in rain data using L1-regularized logistic regression.
  - For different λ values, assuming chronological ordering.



DAG Structure Learning without an Ordering

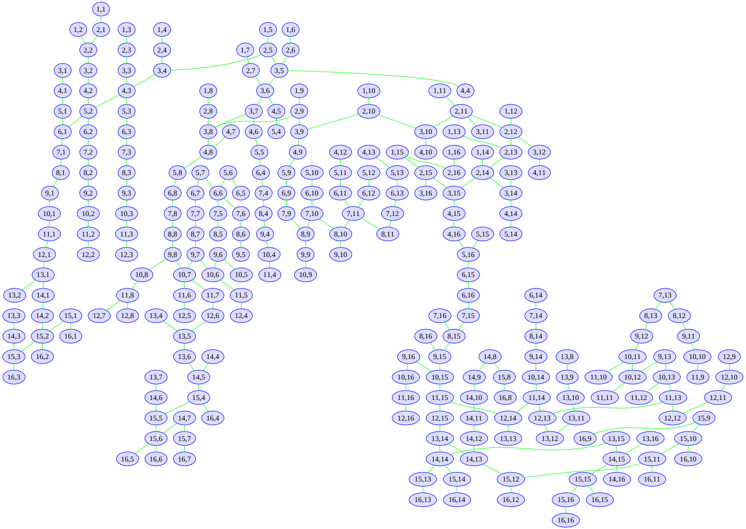
- Without an ordering, a common approach is "search and score"
  - Define a score for a particular graph structure (like BIC).
  - Search through the space of possible DAGs (greedily add/remove/reverse edges).
- Another common approach is "constraint-based" methods:
  - Based on performing a sequence of conditional independence tests.
  - Prune edge between x_i and x_j if you find variables S making them independent,

$$x_i \perp x_j \mid x_S.$$

  - Assumes "faithfulness" (all independences are reflected in graph).
    - Otherwise it's weird (a duplicated feature would be disconnected from everything.)
- Structure learning is NP-hard in general, but finding the optimal tree is poly-time:
  - For symmetric scores, can be done by minimum spanning tree.
  - For asymetric scores, can be by minimum spanning arborescence.

Structure Learning on USPS Digits

Optimal tree on USPS digits.



20 Newsgroups Data

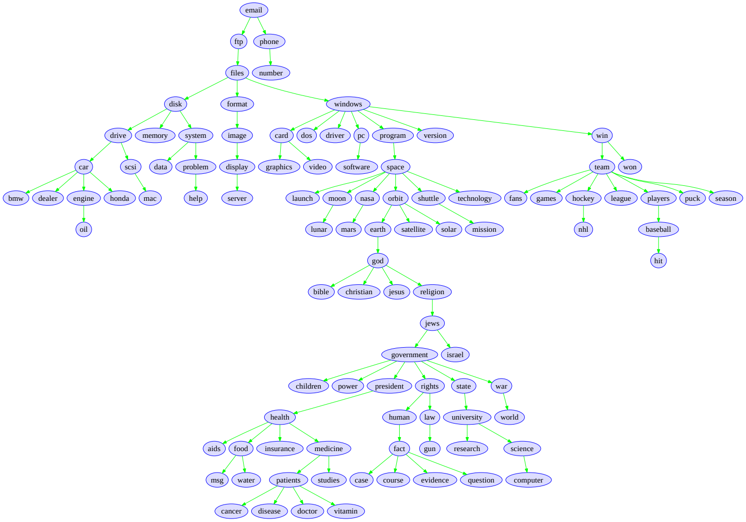
- Data containing presence of 100 words from newsgroups posts:

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

- Structure learning should give relationship between words.

Structure Learning on News Words

Optimal tree on newsgroups data:



Outline

1 Structure Learning

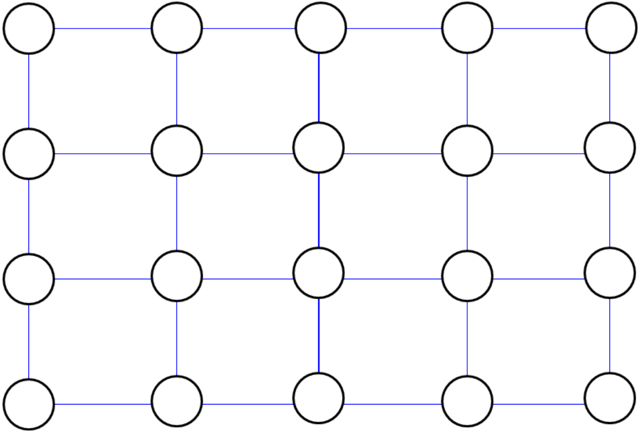
2 Undirected Graphical Models

Directed vs. Undirected Models

- In some applications we have a natural ordering of the x_j.
  - In the "rain" data, the past affects the future.
- In some applications we don't have a natural order.
  - E.g., pixels in an image.
- In these settings we often use undirected graphical models (UGMs).
  - Also known as Markov random fields (MRFs) and originally from statistical physics.

Directed vs. Undirected Models

- Undirected graphical models are based on undirected graphs:



- They are a classic way to model dependencies in images:
  - Can capture dependencies between neighbours without imposing an ordering.

Ising Models from Statistical Physics

- The Ising model for binary x_i is defined by

$$p(x_1, x_2, \dots, x_d) \propto \exp\left(\sum_{i=1}^{d} x_i w_i + \sum_{(i,j)\in E} x_i x_j w_{ij}\right),$$

  where E is the set of edges in an undirected graph.
  - Called a log-linear model, because $\log p(x)$ is linear plus a constant.

- Consider using $x_i \in \{-1, 1\}$:
  - If $w_i > 0$ it encourages $x_i = 1$.
  - If $w_{ij} > 0$ it encourages neighbours i and j to have the same value.
    - E.g., neighbouring pixels in the image receive the same label ("attractive" model)

- We're modeling dependencies, but haven't assumed an "ordering".

Undirected Graphical Models

- Pairwise undirected graphical models (UGMs) assume $p(x)$ has the form

$$p(x) \propto \left(\prod_{j=1}^{d} \phi_j(x_j)\right)\left(\prod_{(i,j)\in E} \phi_{ij}(x_i, x_j)\right).$$

- The ϕ_j and ϕ_{ij} functions are called potential functions:
  - They can be any non-negative function.
  - Ordering doesn't matter: more natural for things like pixels of an image.

- Ising model is a special case where

$$\phi_i(x_i) = \exp(x_i w_i), \quad \phi_{ij}(x_i, x_j) = \exp(x_i x_j w_{ij}).$$

- Bonus slides generalize Ising to non-binary case.

Label Propagation as a UGM

- Consider modeling the probability of a vector of labels $\bar{y} \in R^t$ using

$$p(\bar{y}^1, \bar{y}^2, \dots, \bar{y}^t) \propto \exp\left(-\sum_{i=1}^{n}\sum_{j=1}^{t} w_{ij}(y^i - \bar{y}^i)^2 - \frac{1}{2}\sum_{i=1}^{t}\sum_{j=1}^{t} \bar{w}_{ij}(\bar{y}^i - \bar{y}^j)^2\right).$$

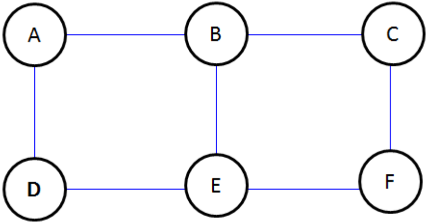
- Decoding in this model is equivalent to the label propagation problem.

- This is a pairwise UGM:

$$\phi_j(\bar{y}^j) = \exp\left(-\sum_{i=1}^{n} w_{ij}(y^i - \bar{y}^j)^2\right), \quad \phi_{ij}(\bar{y}^i, \bar{y}^j) = \exp\left(-\frac{1}{2}\bar{w}_{ij}(\bar{y}^i - \bar{y}^j)^2\right).$$

Conditional Independence in Undirected Graphical Models

- It's easy to check conditional independence in UGMs:
  - $A \perp B \mid C$ if C blocks all paths from any A to any B.
- Example:



  - $A \not\perp C$.
  - $A \not\perp C \mid B$.
  - $A \perp C \mid B, E$.
  - $A, B \not\perp F \mid C$
  - $A, B \perp F \mid C, E$.

Multivariate Gaussian and Pairwise Graphical Models

- Independence in multivariate Gaussian:
  - In Gaussians, marginal independence is determined by covariance:

$$x_i \perp x_j \Leftrightarrow \Sigma_{ij} = 0.$$

  - But how can we determine conditional independence?

- Multivarate Gaussian is a special case of a pairwise UGM.
  - So we can just use graph separation.

- In particular, edges of the UGM are (i, j) values where $\Theta_{i,j} \neq 0$.

- We use the term Gaussian graphical model (GGM) in this context.
  - Or Gaussian Markov random field (GMRF).

Digression: Gaussian Graphical Models

- Multivariate Gaussian can be written as

$$p(x) \propto \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \propto \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x + x^T\underbrace{\Sigma^{-1}\mu}_{v}\right),$$

and writing it in summation notation we can see that it's a pairwise UGM:

$$p(x) \propto \exp(\left(-\frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d}x_ix_j\Sigma^{-1}_{ij} + \sum_{i=1}^{d}x_iv_i\right)$$

$$= \left(\prod_{i=1}^{d}\prod_{j=1}^{d}\underbrace{\exp\left(-\frac{1}{2}x_ix_j\Sigma^{-1}_{ij}\right)}_{\phi_{ij}(x_i,x_j)}\right)\left(\prod_{i=1}^{d}\underbrace{\exp\left(x_iv_i\right)}_{\phi_i(x_i)}\right)$$

Independence in GGMs

- So Gaussians are pairwise UGMs with $\phi_{ij}(x_i, x_j) = \exp\left(-\frac{1}{2}x_i x_j \Theta_{ij}\right)$,
  - Where Θ_{ij} is element (i, j) of Σ^{-1}.
- Consider setting $\Theta_{ij} = 0$:
  - For all (x_i, x_j) we have $\phi(x_i, x_j) = 1$, which is equivalent to not having edge (i, j).
- So setting $\Theta_{ij} = 0$ is equivalent to removing $\phi_{ij}(x_i, x_j)$ from the UGM.
- Gaussian conditional independence is determined by precision matrix sparsity.
  - Diagonal Θ gives disconnected graph: all variables are independent.
  - Full Θ gives fully-connected graph: there are no independences.

Independence in GGMs

- Consider a Gaussian with the following covariance matrix:

$$\Sigma = \begin{bmatrix} 0.0494 & -0.0444 & -0.0312 & 0.0034 & -0.0010 \\ -0.0444 & 0.1083 & 0.0761 & -0.0083 & 0.0025 \\ -0.0312 & 0.0761 & 0.1872 & -0.0204 & 0.0062 \\ 0.0034 & -0.0083 & -0.0204 & 0.0528 & -0.0159 \\ -0.0010 & 0.0025 & 0.0062 & -0.0159 & 0.2636 \end{bmatrix}$$

- $\Sigma_{ij} \neq 0$ so all variables are dependent: $x_1 \not\perp x_2$, $x_1 \not\perp x_5$, and so on.
  - This would show up in graph: you would be able to reach any x_i from any x_j.
- The inverse is given by a tri-diagonal matrix:

$$\Sigma^{-1} = \begin{bmatrix} 32.0897 & 13.1740 & 0 & 0 & 0 \\ 13.1740 & 18.3444 & -5.2602 & 0 & 0 \\ 0 & -5.2602 & 7.7173 & 2.1597 & 0 \\ 0 & 0 & 2.1597 & 20.1232 & 1.1670 \\ 0 & 0 & 0 & 1.1670 & 3.8644 \end{bmatrix}$$

- So conditional independence is described by a Markov chain:

$$p(x_1 \mid x_2, x_3, x_4, x_5) = p(x_1 \mid x_2).$$

Graphical Lasso

- Conditional independence in GGMs is described by sparsity in Θ.
  - Setting a Θ_{ij} to 0 removes an edge from the graph.

- Recall fitting multivariate Gaussian with L1-regularization,

$$\operatorname*{argmin}_{\Theta \succ 0} \mathrm{Tr}(S\Theta) - \log|\Theta| + \lambda\|\Theta\|_1,$$

which is called the graphical Lasso because it encourages a sparse graph.

- Graphical Lasso is a convex approach to structure learning for GGMs.
  - Examples `https://normaldeviate.wordpress.com/2012/09/17/high-dimensional-undirected-graphical-models`.

Higher-Order Undirected Graphical Models

- In UGMs, we can also define potentials on higher-order interactions.
  - A three-variable generalization of Ising potentials is:

$$\phi_{ijk}(x_i, x_j, x_k) = w_{ijk} x_i x_j x_k.$$

    - If $w_{ijk} > 0$ and $x_j \in \{0, 1\}$, encourages you to set all three to 1.
    - If $w_{ijk} > 0$ and $x_j \in \{-1, 1\}$, encourages odd number of positives.
- In the general case, a UGM just assumes $p(x)$ factorizes over subsets c,

$$p(x_1, x_2, \dots, x_d) \propto \prod_{c \in \mathcal{C}} \phi_c(x_c),$$

  from among a collection of subsets of $\mathcal{C}$.

- In this case, graph has edge (i, j) if i and j are together in at least one c.
  - Conditional independences are still given by graph separation.

Tractability of UGMs

- Without using $\propto$, we write UGM probability as

$$p(x) = \frac{1}{Z}\prod_{c \in \mathcal{C}} \phi_c(x_c),$$

where Z is the constant that makes the probabilites sum up to 1.

$$Z = \sum_{x_1}\sum_{x_2}\cdots\sum_{x_d}\prod_{c\in\mathcal{C}}\phi_c(x_c) \quad \text{or} \quad Z = \int_{x_1}\int_{x_2}\cdots\int_{x_d}\prod_{c\in\mathcal{C}}\phi_c(x_c)dx_d dx_{d-1}\dots dx_1 = 1.$$

- Whether you can compute Z depends on the choice of the ϕ_c:
  - Gaussian case: $O(d^3)$ in general, but $O(d)$ for forests (no loops).
  - Continuous non-Gaussian: usually requires numerical integration.
  - Discrete case: #P-hard in general, but $O(dk^2)$ for forests (no loops).

Summary

- Structure learning is the problem of learning the graph structure.
    - Hard in general, but easy for trees and L1-regularization gives fast heuristic.
- Undirected graphical models factorize probability into non-negative potentials.
    - Gaussians are a special case.
    - Log-linear models (like Ising) are a common choice.
    - Simple conditional independence properties.
- Next time: our first visit to the wild world of approximate inference.

General Pairwise UGM

- For general discrete x_i a generalization of Ising models is

$$p(x_1, x_2, \ldots, x_d) = \frac{1}{Z} \exp\left(\sum_{i=1}^{d} w_{i,x_i} + \sum_{(i,j) \in E} w_{i,j,x_i,x_j}\right),$$

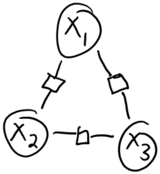
  which can represent any "positive" pairwise UGM (meaning $p(x) > 0$ for all x).

- Interpretation of weights for this UGM:
  - If $w_{i,1} > w_{i,2}$ then we prefer $x_i = 1$ to $x_i = 2$.
  - If $w_{i,j,1,1} > w_{i,j,2,2}$ then we prefer $(x_i = 1, x_j = 1)$ to $(x_i = 2, x_j = 2)$.

- As before, we can use parameter tieing:
  - We could use the same w_{i,x_i} for all positions i.
  - Ising model corresponds to a particular parameter tieing of the w_{i,j,x_i,x_j}.

Factor Graphs

- Factor graphs are a way to visualize UGMs that distinguishes different orders.
  - Use circles for variables, squares to represent dependencies.
- Factor graph if $p(x_1, x_2, x_3) \propto \phi_{12}(x_1, x_2)\phi_{13}(x_1, x_2, x_3)\phi_{23}(x_2, x_3)$:



- Factor graph if $p(x_1, x_2, x_3) \propto \phi_{123}(x_1, x_2, x_3)$: