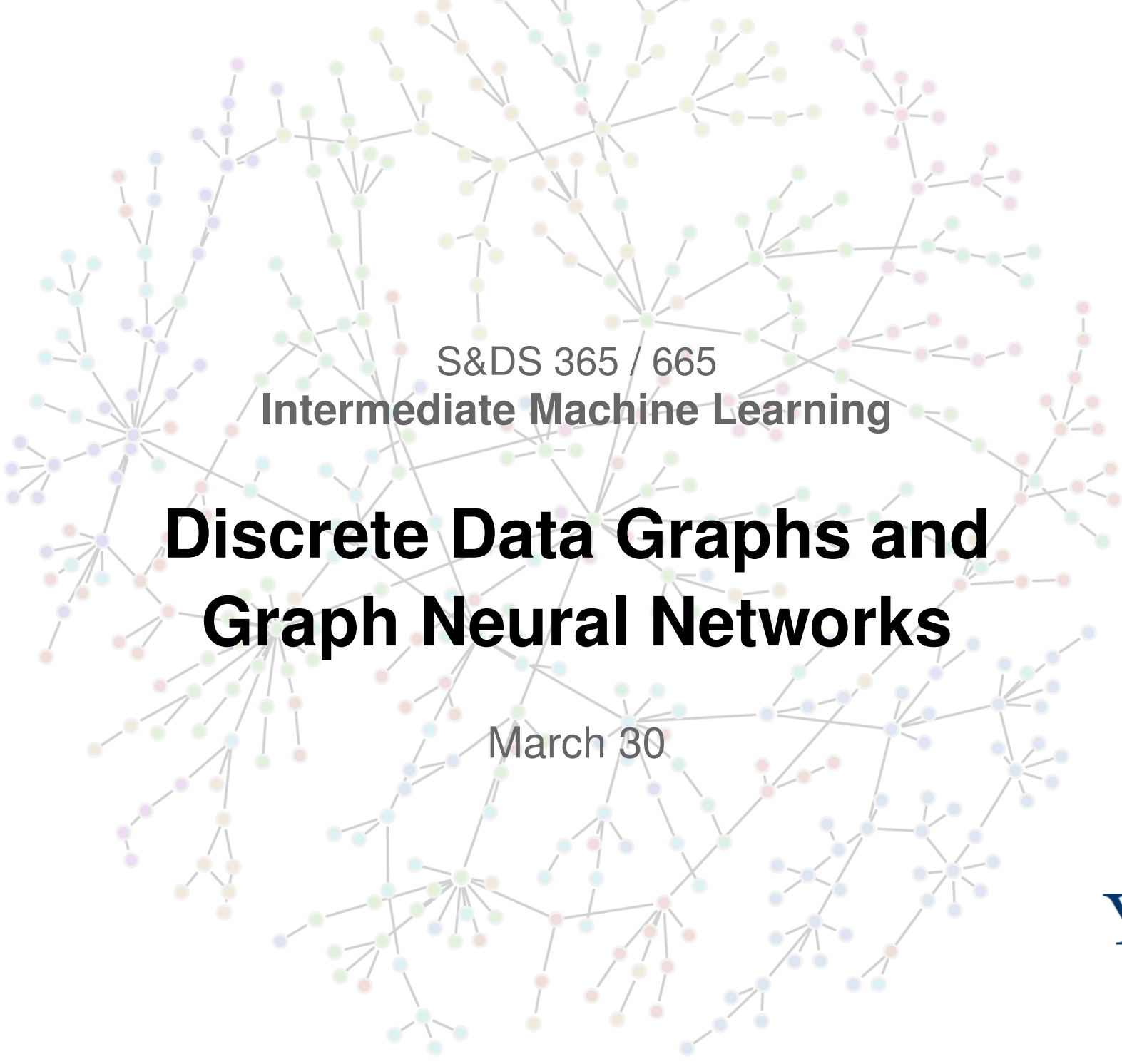


S&DS 365 / 665

Intermediate Machine Learning

Discrete Data Graphs and Graph Neural Networks

March 30

Yale

The Alignment Problem: Machine Learning and Human Values

Yale University's Wu Tsai Institute and the Schmidt Program on Artificial Intelligence, Emerging Technologies, and National Power will co-host the talk, "The Alignment Problem: Machine Learning and Human Values," by **Brian Christian**, an award-winning author and Science Communicator in Residence at the Simons Institute for the Theory of Computing at University of California – Berkeley.

Christian is recognized as a leading authority on artificial intelligence and the ethical challenges associated with emerging technologies. His latest book, "The Alignment Problem: Machine Learning and Human Values," is a blend of history and on-the-ground reporting, tracing the explosive growth of machine learning and the wide range of resulting risks, opportunities, and unintended consequences. The book is a Los Angeles Times Finalist for Best Science & Technology Book of the Year, and Microsoft CEO Satya Nadella has named it one of the five books that inspired him in 2021.

Christian is the author of the acclaimed bestsellers "The Most Human Human" and "Algorithms to Live By." His writing has appeared in The New Yorker, The Atlantic, Wired, and The Wall Street Journal, as well as peer-reviewed journals. He holds degrees in computer science, philosophy, and poetry from Brown University and the University of Washington.

The talk will be moderated by John Lafferty, John C. Malone Professor of Statistics & Data Science, and Director of the Center for Neurocomputation and Machine Intelligence at Yale.

The in-person event is open to members of the Yale campus community with Yale ID.

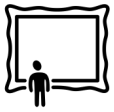
Thursday, March 31 | 4:30pm
Watson Center, Room A51
60 Sachem Street, New Haven

The Alignment Problem: Machine Learning and Human Values

Tomorrow, 4:30
Watson Center, Room A51
60 Sachem Street

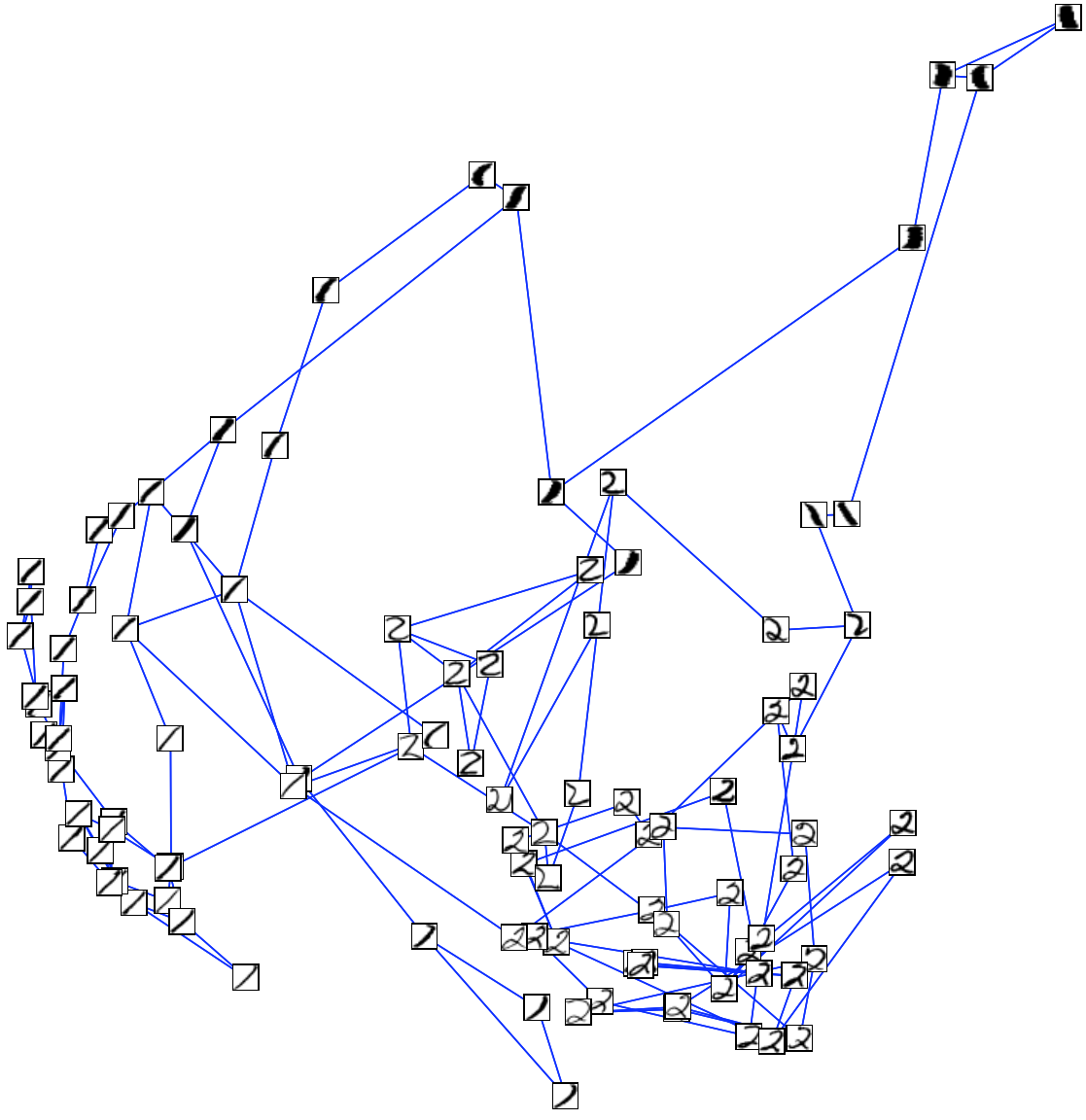
Work schedule

- Assignment 3 out today
- Quiz 3 next Wednesday
  - Variational inference and VAEs
  - Undirected graphs and glasso
  - Graph neural nets

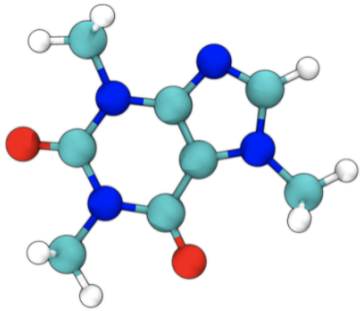


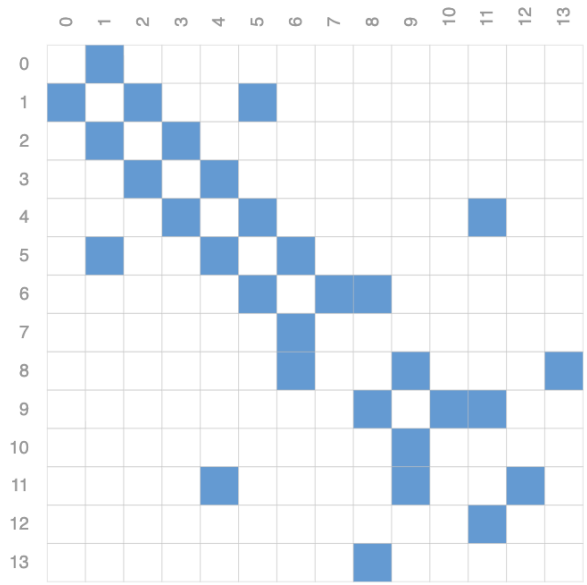
Graphs

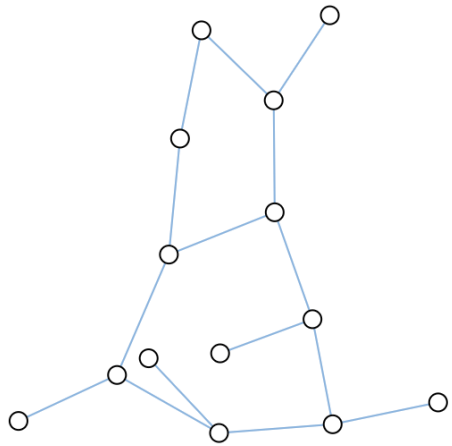
- A natural language for describing various data
- Give information about relationships between variables
- Associated with each multivariate distribution



semi-supervised learning

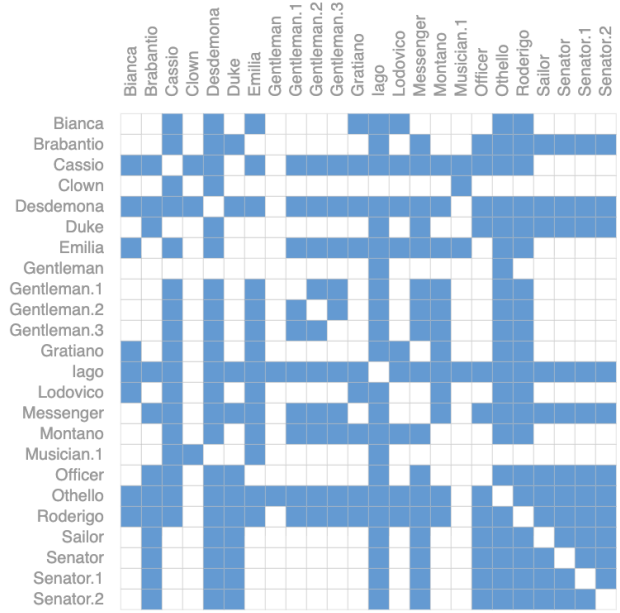


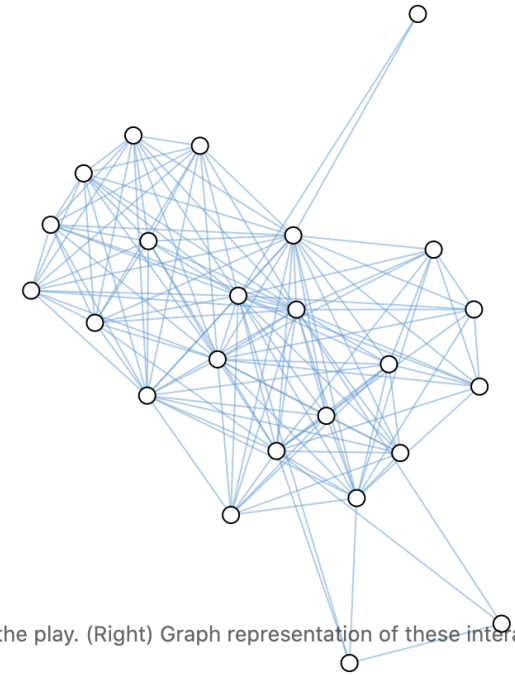




(Left) 3d representation of the Caffeine molecule (Center) Adjacency matrix of the bonds in the molecule (Right) Graph representation of the molecule.

https://distill.pub/2021/gnn-intro/

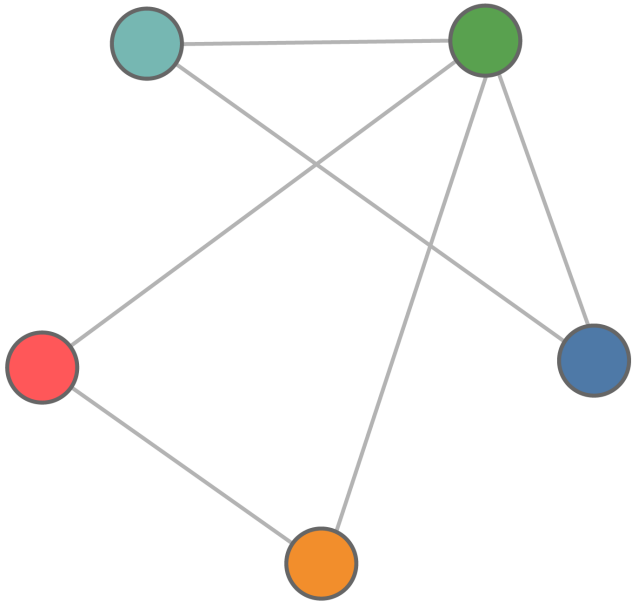




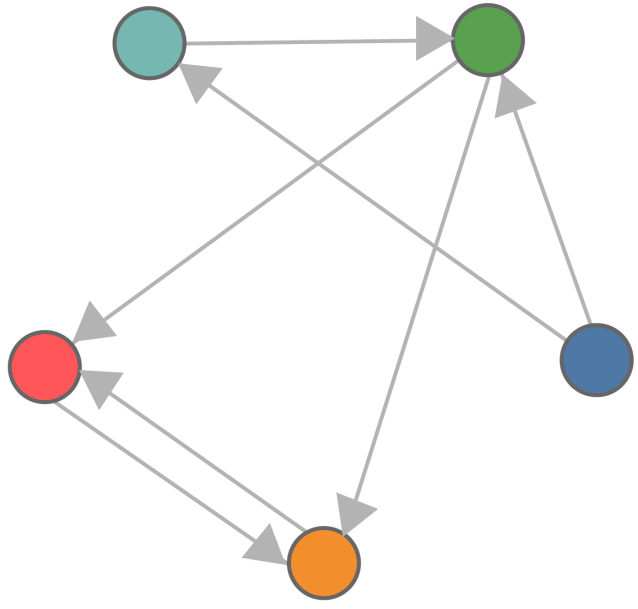
(Left) Image of a scene from the play "Othello". (Center) Adjacency matrix of the interaction between characters in the play. (Right) Graph representation of these interactions.

https://distill.pub/2021/gnn-intro/

Undirected graph



Directed graph

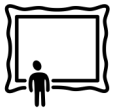


Undirected Graphs

A graph $G = (V, E)$ has vertices V, edges E.

If $X = (X_1, \ldots, X_p)$ is a random variable, we will study graphs where there are p vertices, one for each X_j.

The graph will encode conditional independence relations among the variables.



Graphs for data/distributions

- Graphs give us a new way of understanding data
- Allow us to make structural assumptions
- Central to causal inference

Undirected graphs

Simplest case:

Here $V = \{X, Y, Z\}$ and $E = \{(X, Y), (Y, Z)\}$.

This encodes the independence relation

$$X \perp\!\!\!\perp Z \mid Y$$

which means that *X and Z are independent conditioned on Y*.

Markov Property

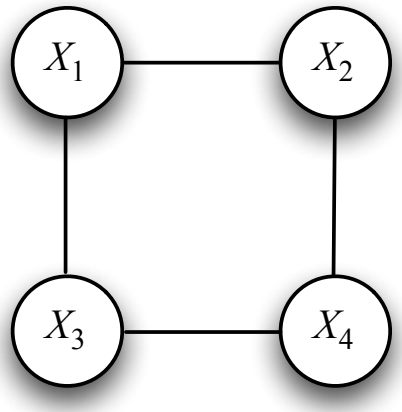
A probability distribution P satisfies the *global Markov property* with respect to a graph G if:

for any disjoint vertex subsets A, B, and C such that C separates A and B,

$$X_A \perp\!\!\!\perp X_B \mid X_C.$$

- X_A are the random variables X_j with $j \in A$.
- C separates A and B means that there is no path from A to B that does not pass through C.

Example

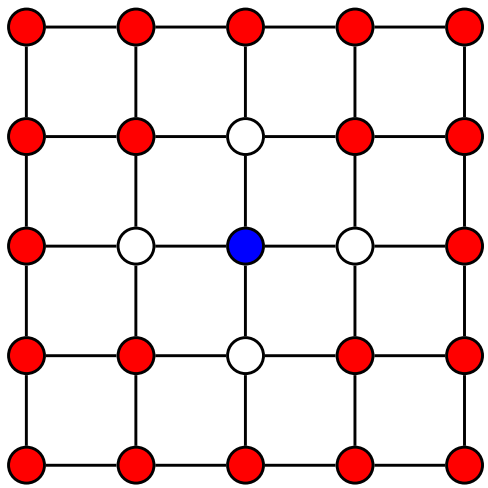


$$X_1 \perp\!\!\!\perp X_4 \mid X_2, X_3$$

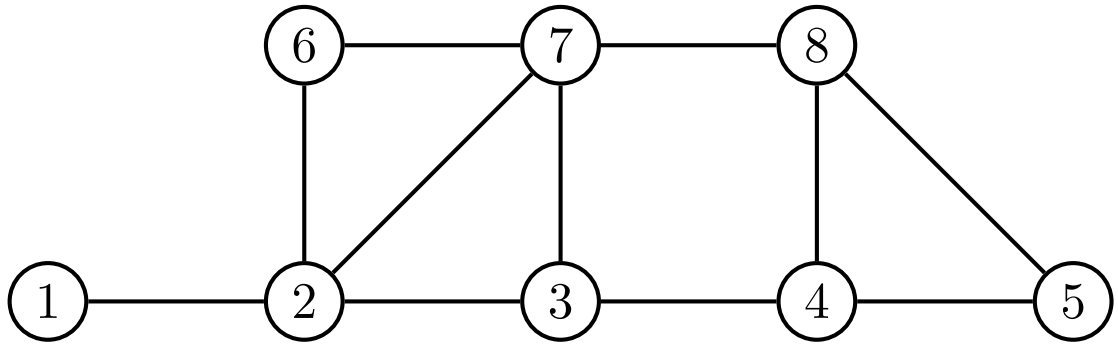
$$X_2 \perp\!\!\!\perp X_3 \mid X_1, X_4$$

Example: 2-dimensional grid

The blue node is independent of the red nodes given the white nodes.



Example



$C = \{3, 7\}$ separates $A = \{1, 2\}$ and $B = \{4, 8\}$. Hence,

$$\{X_1, X_2\} \perp\!\!\!\perp \{X_4, X_8\} \mid \{X_3, X_7\}$$

Special case

If $(i, j) \notin E$ then

$$X_i \perp\!\!\!\perp X_j \mid \{X_k : k \neq i, j\}$$

Special case

If $(i, j) \notin E$ then

$$X_i \perp\!\!\!\perp X_j \mid \{X_k : k \neq i, j\}$$

Lack of an edge from i to j implies that X_i and X_j are independent given all of the other random variables.

Graph estimation

- A graph G represents the class of distributions, $\mathcal{P}(G)$, the distributions that are Markov with respect to G
- Graph estimation: Given n samples $X_1, \ldots, X_n \sim P$, estimate the graph G.

Gaussian case

Let $\Omega = \Sigma^{-1}$ be the precision matrix.

A zero in Ω indicates a *lack of the corresponding edge* in the graph

So, the adjacency matrix of the graph is

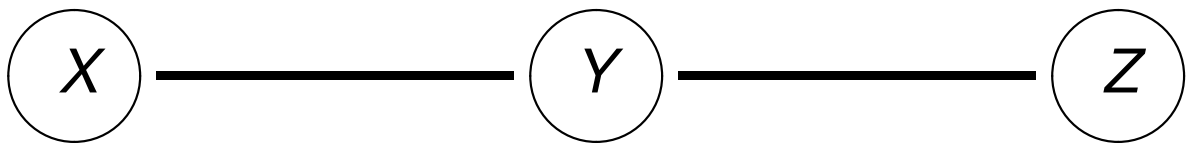
$$A = \left(\mathbb{1}(\Omega_{ij} \neq 0)\right)$$

That is,

$$A_{ij} = \begin{cases} 1 & \text{if } |\Omega_{ij}| > 0 \\ 0 & \text{otherwise} \end{cases}$$

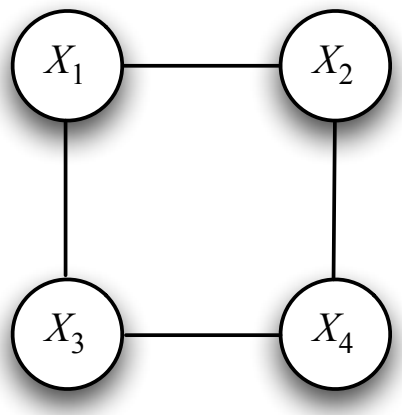
Gaussian case

$$\Omega \equiv \Sigma^{-1} = \begin{pmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{pmatrix}$$



Gaussian case

$$\Omega \equiv \Sigma^{-1} = \begin{pmatrix} * & * & * & 0 \\ * & * & 0 & * \\ * & 0 & * & * \\ 0 & * & * & * \end{pmatrix}$$



$$X_1 \perp\!\!\!\perp X_4 \mid X_2, X_3$$

Gaussian case: Algorithms

Two approaches:

- parallel lasso
- graphical lasso

Parallel Lasso:

1. For each $j = 1, \ldots, p$ (in parallel): Regress X_j on all other variables using the lasso.
2. Put an edge between X_i and X_j if each appears in the regression of the other.

Graphical Lasso (glasso)

- Assume a multivariate Gaussian model
- Subtract out the sample mean
- Minimize the negative log-likelihood of the data, subject to a constraint on the sum of the absolute values of the inverse covariance

Graphical Lasso (glasso)

The glasso optimizes the parameters of $\Omega = \Sigma^{-1}$ by minimizing:

$$\text{trace}(\Omega S_n) - \log|\Omega| + \lambda \sum_{j \neq k} |\Omega_{jk}|$$

where $|\Omega|$ is the determinant and S_n is the sample covariance

$$S_n = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T$$

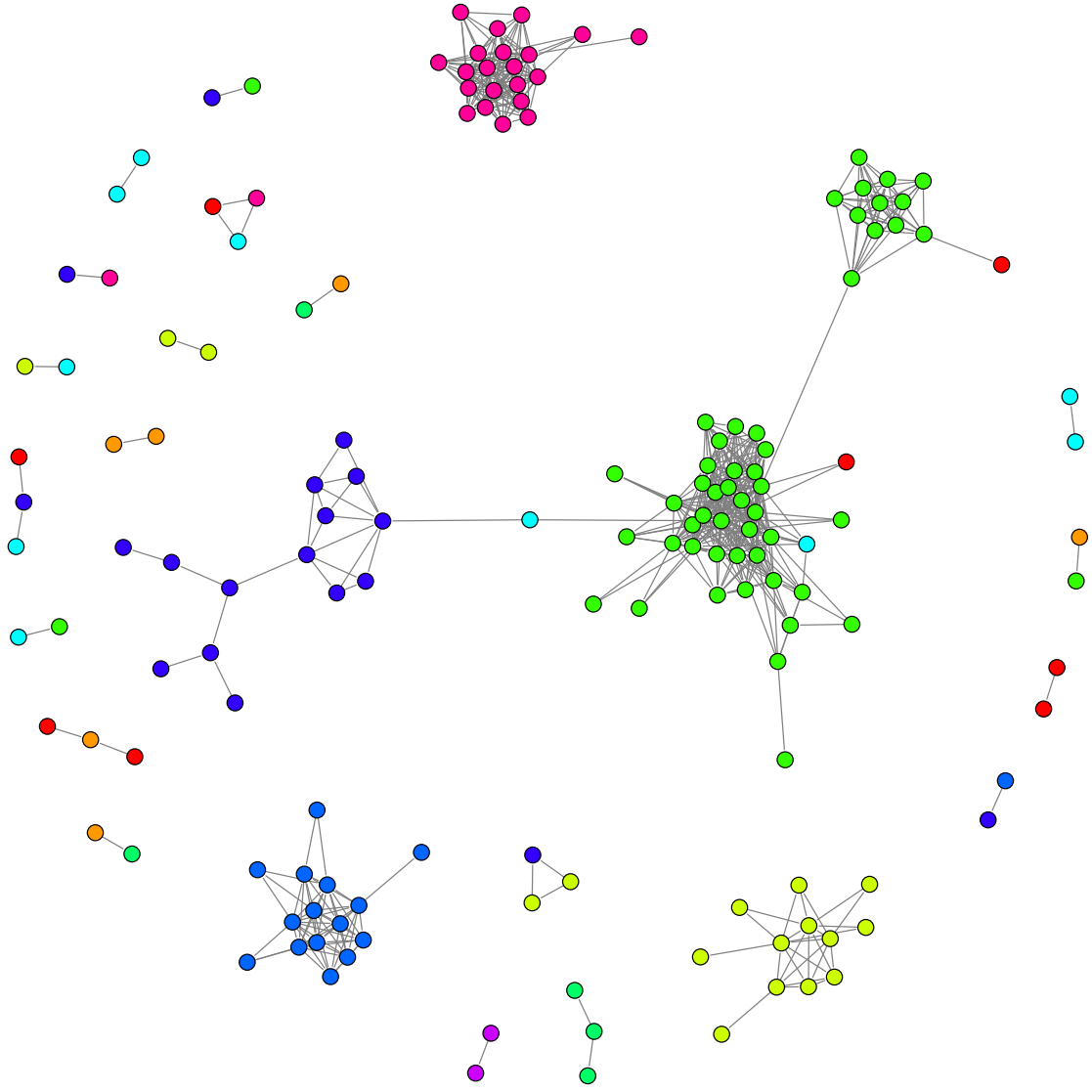
There is a blockwise gradient descent algorithm to minimize this, using iterative lassos

Graphs on the S&P 500

- Data from Yahoo! Finance (`finance.yahoo.com`).
- Daily closing prices for 452 stocks in the S&P 500 between 2003 and 2008 (before onset of the "financial crisis").
- Log returns $X_{tj} = \log\left(S_{t,j}/S_{t-1,j}\right)$.
- Outliers capped at $\pm 6\sigma$.
- In following graphs, each node is a stock, and color indicates an industry sector

```
Consumer Discretionary          Consumer Staples
Energy                          Financials
Health Care                     Industrials
Information Technology          Materials
Telecommunications Services     Utilities
```

S&P 500: Graphical Lasso

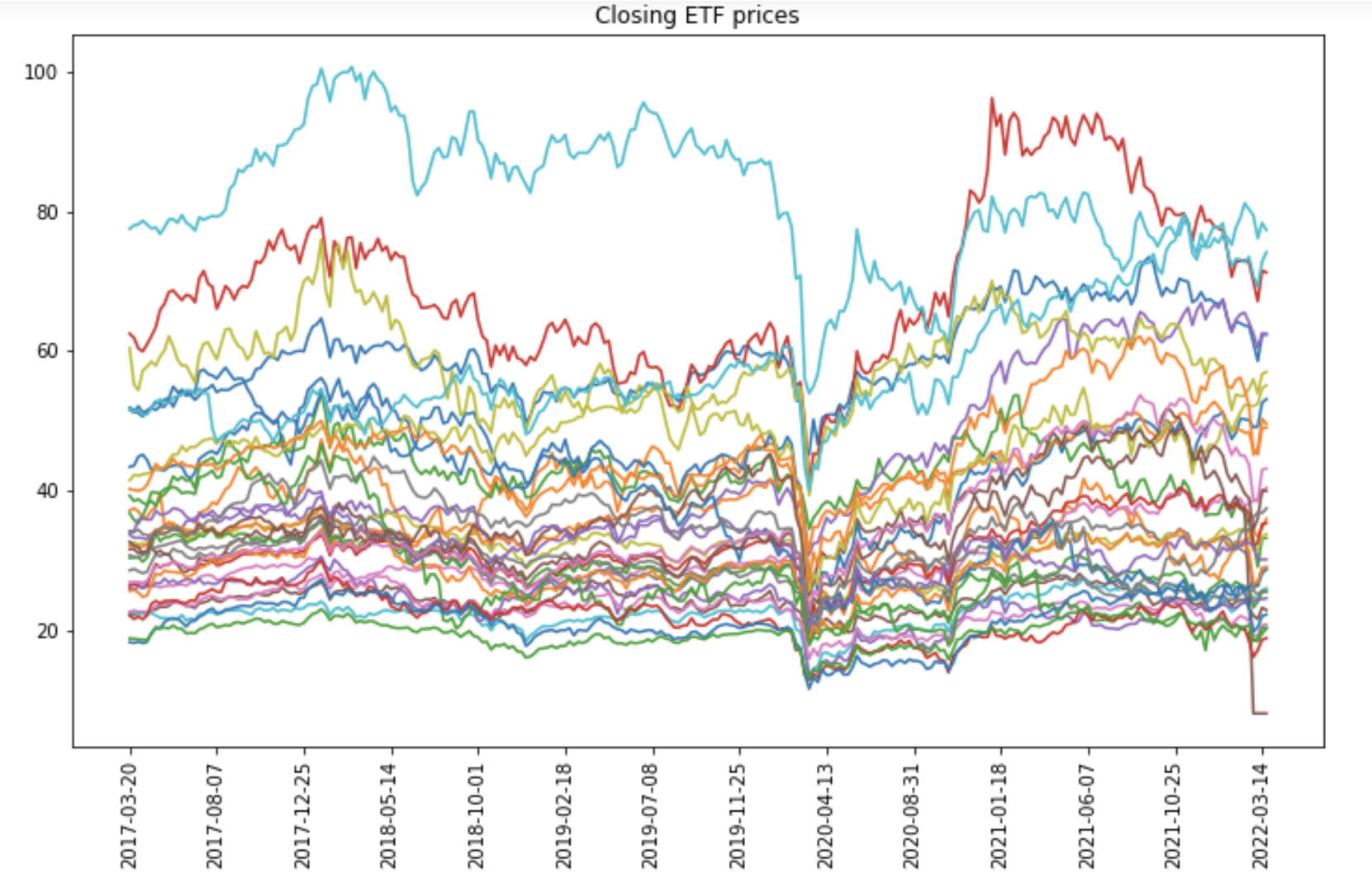


Example Neighborhood

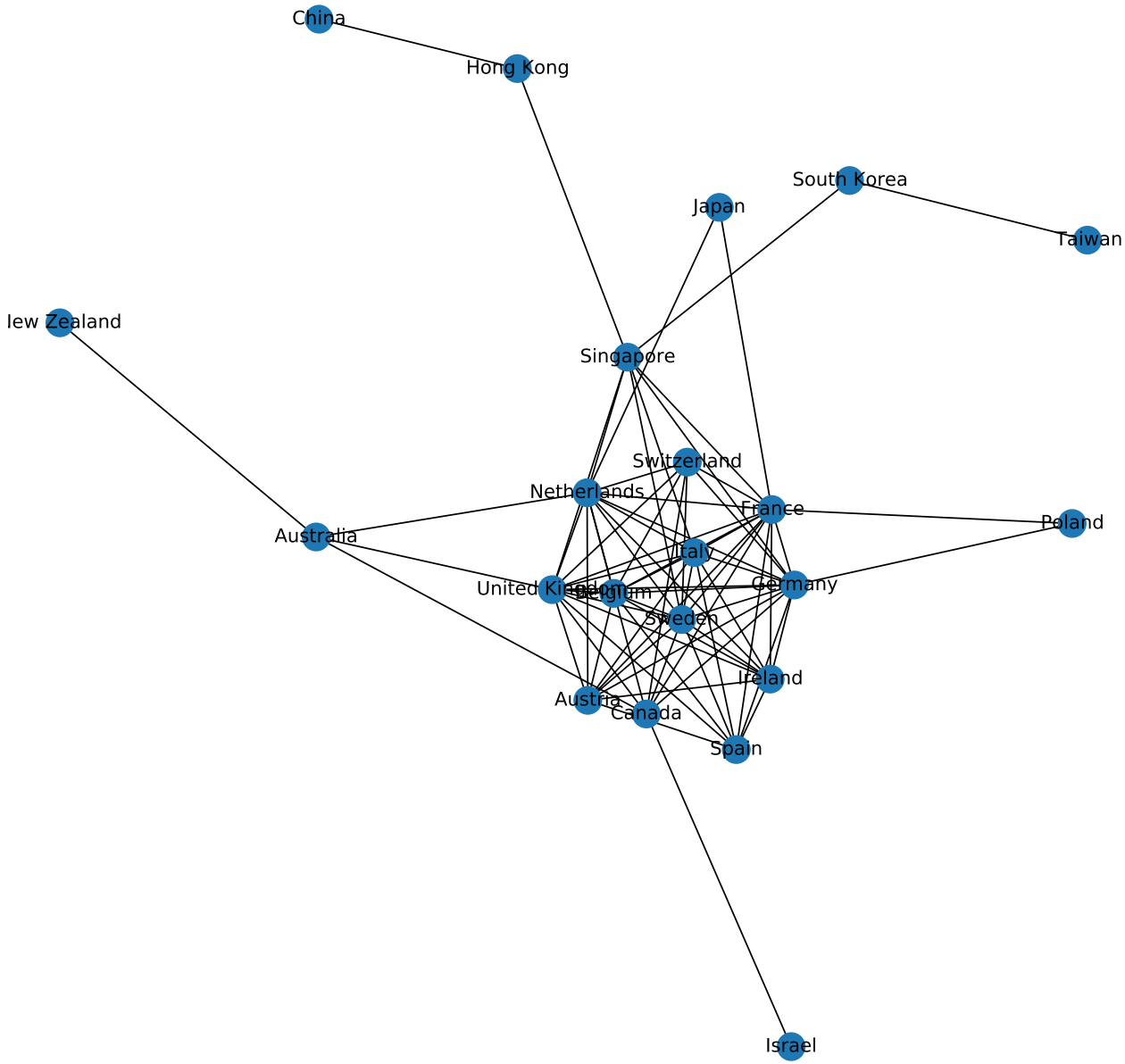
Target Corp. (Consumer Discretionary):

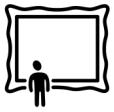
- Big Lots, Inc. (Consumer Discretionary)
- Costco Co. (Consumer Staples)
- Family Dollar Stores (Consumer Discretionary)
- Kohl's Corp. (Consumer Discretionary)
- Lowe's Cos. (Consumer Discretionary)
- Macy's Inc. (Consumer Discretionary)
- Wal-Mart Stores (Consumer Staples)

Demo: glasso on ETF data



Demo: glasso on ETF data





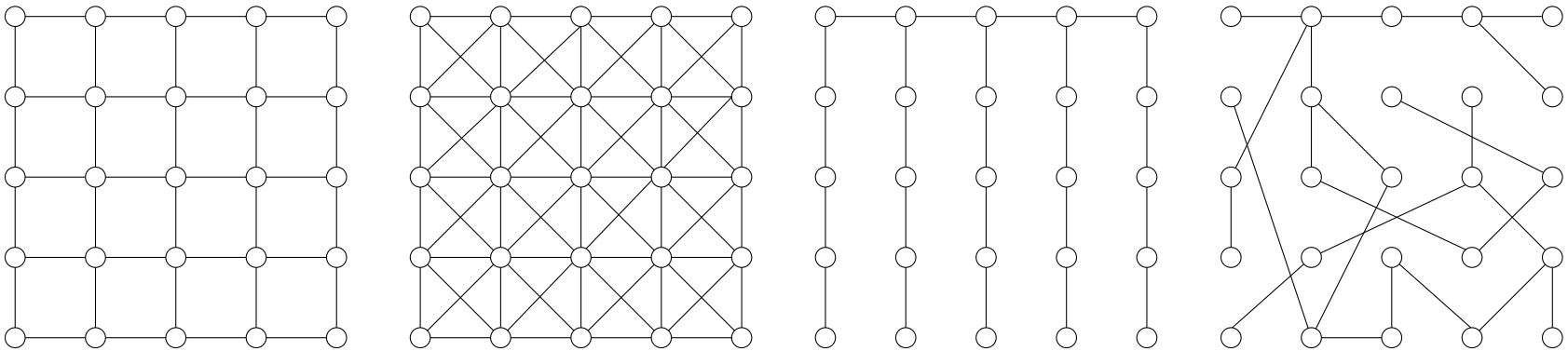
Discrete Graphical Models

Challenges of handling discrete data:

- Models don’t have closed from
- Recall: Need to use Gibbs sampling, variational inference
- No analogue of the graphical lasso

Discrete Graphical Models

Let $G = (V, E)$ be an undirected graph on $m = |V|$ vertices



- (Hammersley, Clifford) A positive distribution over random variables $Z_1, \ldots, Z_p$ that satisfies the Markov properties of graph G can be represented as

$$p(Z) \propto \prod_{c \in \mathcal{C}} \psi_c(Z_c)$$

where $\mathcal{C}$ is the set of cliques in the graph G.

Discrete Graphical Models

- Positive distributions can be represented by an exponential family,

$$p(Z;\beta) \propto \exp\left(\sum_{c\in\mathcal{C}} \beta_c \phi_c(Z_c)\right)$$

- Special case: Ising Model (binary Gaussian)

$$p(Z;\beta) \propto \exp\left(\sum_{i\in V} \beta_i Z_i + \sum_{(i,j)\in E} \beta_{ij} Z_i Z_j\right).$$

  Here, the set of cliques $\mathcal{C} = \{V \cup E\}$, and the potential functions are $\{Z_i,\ i \in V\} \cup \{Z_i Z_j,\ (i,j) \in E\}$.

Discrete Gaussian?

Note that we can write a multivariate Gaussian as follows:

$$p(z) \propto exp\left(\sum_i \beta_i z_i + \sum_{i,j} \beta_{ij} z_i z_j\right)$$

Can you see what β_i and β_{ij} are?

From edges to cliques

Take $\beta_i \equiv 0$ for simplicity

If we have a triangle (i, j, k) in the graph then the potential function corresponds to

$$\psi_{(ijk)}(Z_i, Z_j, Z_k) = e^{\beta_{ij} Z_i Z_j} \cdot e^{\beta_{jk} Z_j Z_k} \cdot e^{\beta_{ik} Z_i Z_k}$$

Recall from a few weeks ago

We have a graph with edges E and vertices V. Each node i has a random variable Z_i that can be "up" ($Z_i = 1$) or "down" ($Z_i = 0$)

$$\mathbb{P}_\beta(z_1, \ldots, z_n) \propto \exp\left(\sum_{s \in V} \beta_s z_s + \sum_{(s,t) \in E} \beta_{st} z_s \, z_t\right)$$

This is called an "Ising model" and is central to statistical physics.

Recall from a few weeks ago

We have a graph with edges E and vertices V. Each node i has a random variable Z_i that can be "up" ($Z_i = 1$) or "down" ($Z_i = 0$)

$$\mathbb{P}_\beta(z_1, \ldots, z_n) \propto \exp\left(\sum_{s \in V} \beta_s z_s + \sum_{(s,t) \in E} \beta_{st} z_s z_t\right)$$

E are the set of edges, V are the vertices. Imagine the Z_i are votes of politicians, and the edges encode the social network of party affiliations

Stochastic approximation

Gibbs sampler

Iterate until converged:

1. Choose vertex $s \in V$ at random
2. Sample z_s holding others fixed

$$\theta_s = \text{sigmoid}\left(\beta_s + \sum_{t \in N(s)} \beta_{st} z_t\right)$$

$$Z_s \,|\, \theta_s \sim \text{Bernoulli}(\theta_s)$$

$\text{sigmoid}(u) = 1/(1 + e^{-u})$

Deterministic approximation

Mean field variational algorithm

Iterate until converged:

1. Choose vertex $s \in V$ at random
2. Update mean μ_s holding others fixed

$$\mu_s = \text{sigmoid}\left(\beta_s + \sum_{t \in N(s)} \beta_{st}\, \mu_t\right)$$

---

sigmoid$(u) = 1/(1 + e^{-u})$

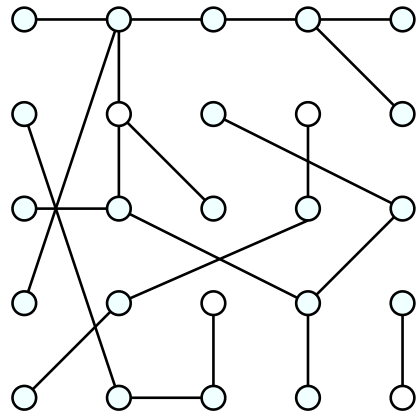
Determimnistic vs. stochastic approximation

- The z_s variables are random
- The μ_s variables are deterministic
- The Gibbs sampler convergence is in distribution
- The mean field convergence is numerical
- The Gibbs sampler approximates the full distribution
- The mean field algorithm approximates the mean of each node

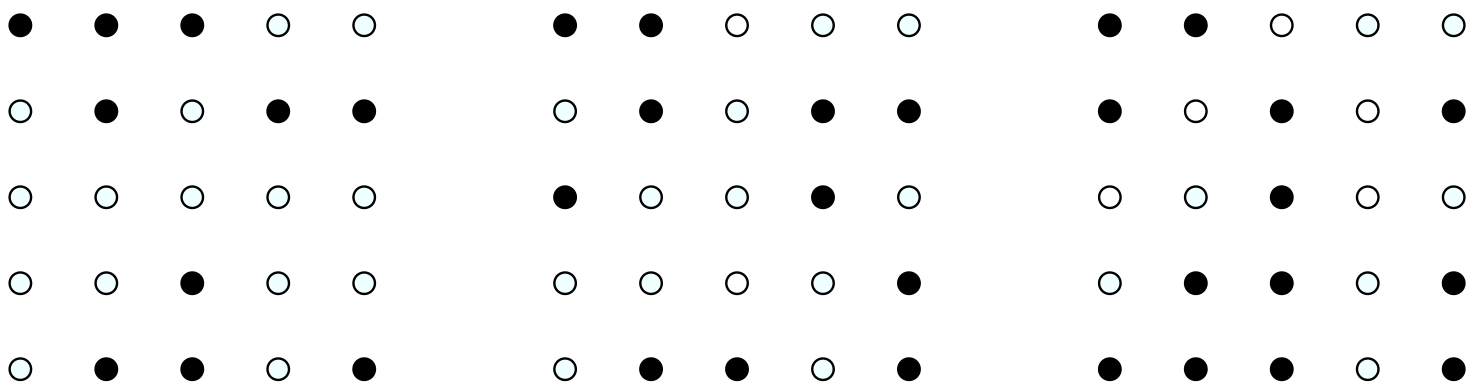
Think of how to interpret this with Z_s the vote of politician s

Graph Estimation

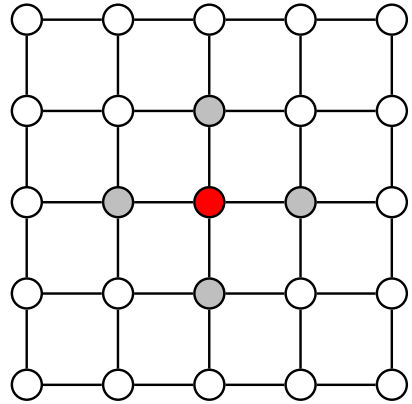
- Given n i.i.d. samples from an Ising distribution, $\{Z_i, i = 1, \ldots, n\}$, (each is a p-vector of $\{0, 1\}$ values) identify underlying graph



- Multiple examples are observed:



Local Distributions



- Consider Ising model $p_\beta(Z) \propto \exp\left(\sum_{i \in V} \beta_i Z_i + \sum_{(i,j) \in E} \beta_{ij} Z_i Z_j\right)$.
- Conditioned on $(z_2, \ldots, z_p)$, variable $Z_1 \in \{0, 1\}$ has probability mass function given by a logistic function,

$$\mathbb{P}(Z_1 = 1 \mid z_2, \ldots, z_p) \quad = \quad \text{sigmoid}\left(\beta_1 + \sum_{j \in \mathcal{N}(1)} \beta_{1j} z_j\right)$$

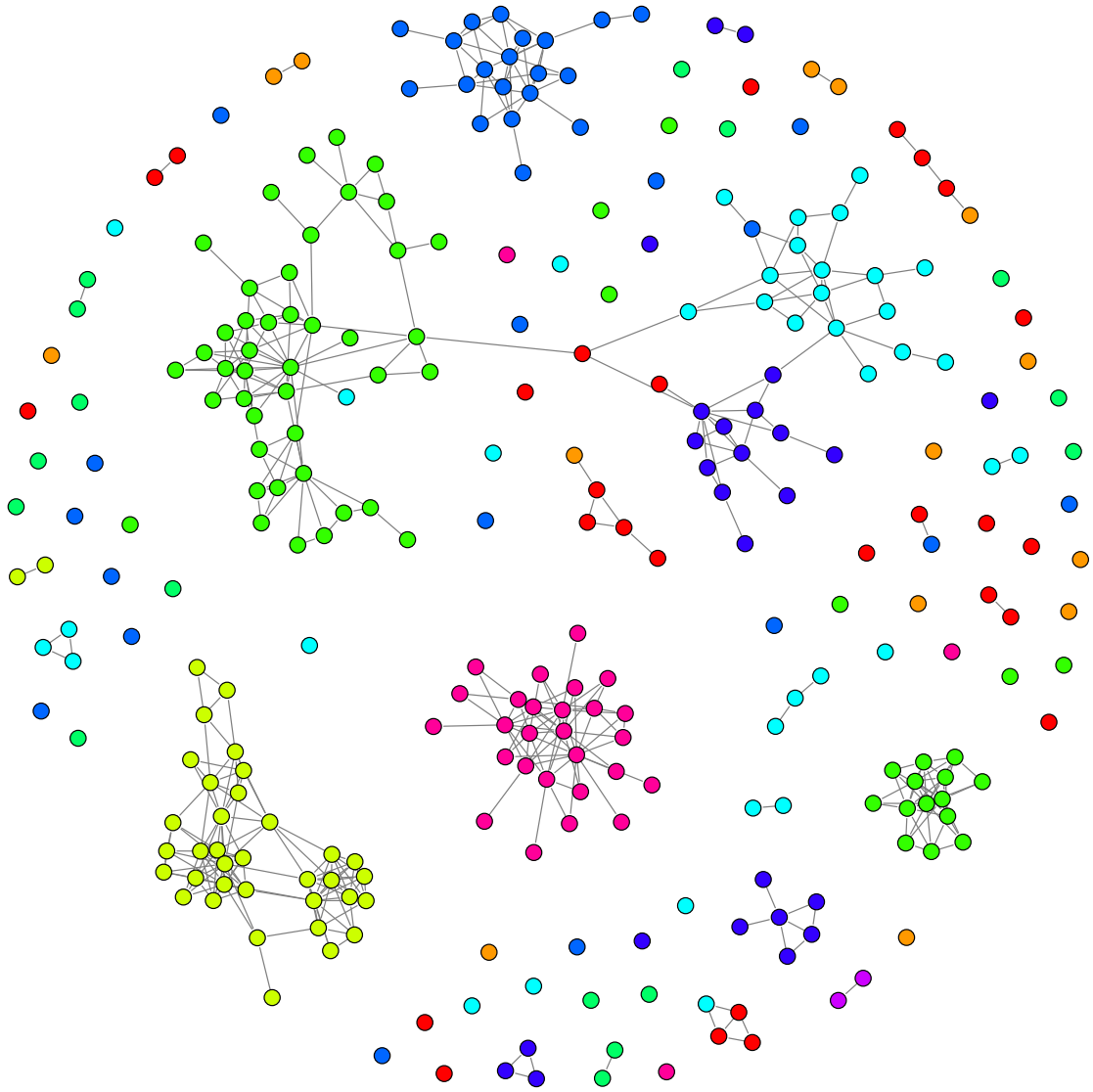
Parallel lasso (sparse logistic regressions)

Strategy

- Perform ℓ_1 regularized logistic regression of each node Z_i on $Z_{\setminus i} = \{Z_j,\ j \neq i\}$ to estimate neighors $\widehat{\mathcal{N}}(i)$
- Two versions:
  - Create an edge (i, j) if $j \in \widehat{\mathcal{N}}(i)$ *and* $i \in \widehat{\mathcal{N}}(j)$
  - Create an edge (i, j) if $j \in \widehat{\mathcal{N}}(i)$ *or* $i \in \widehat{\mathcal{N}}(j)$

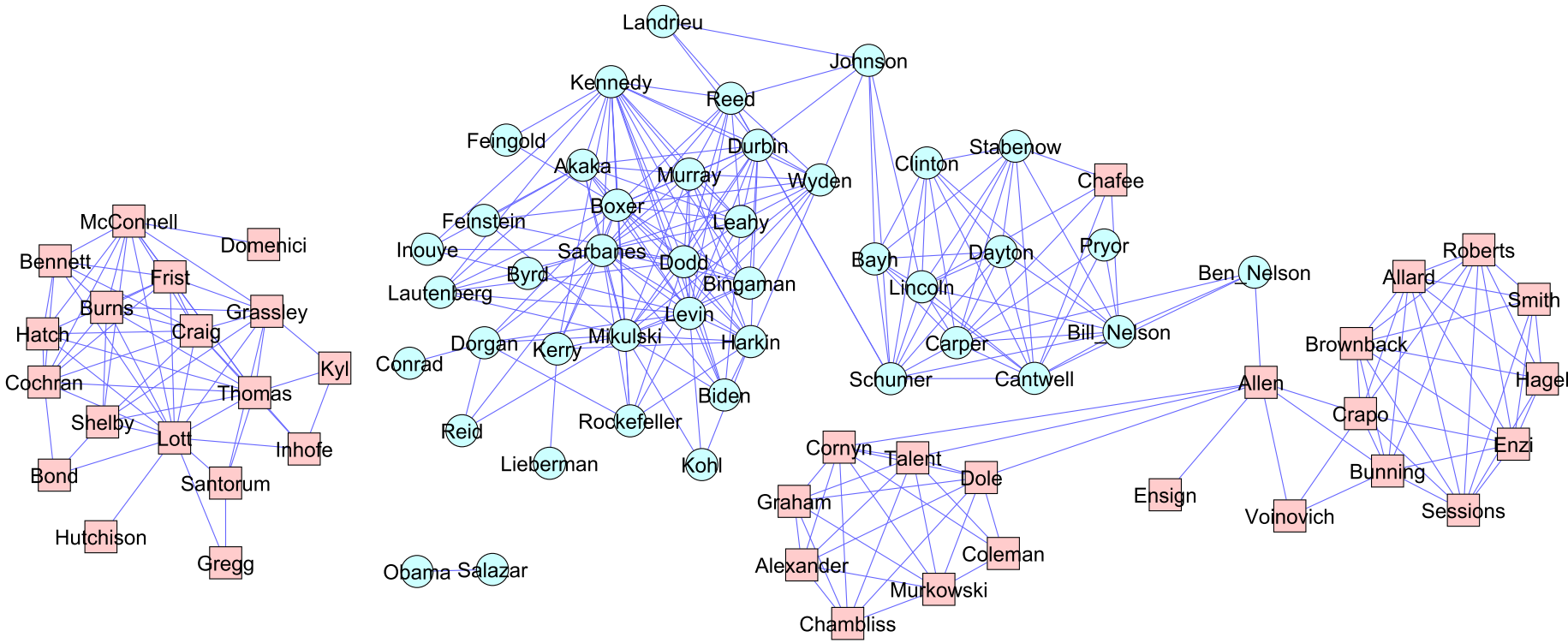
https://www.jstor.org/stable/20744454

S&P 500: Ising Model (Price up or down?)



Voting Data

Voting records of US Senate, 2006-2008



Example of Banerjee, El Gahoui, and d'Asepremont (JMLR, 2008)

Scaling behavior: Performance with data size

Maximum degree d of the p variables. Sample size n must satisfy

$$\text{Ising model:} \quad n \geq d^3 \log p$$

$$\text{Graphical lasso:} \quad n \geq d^2 \log p$$

$$\text{Parallel lasso:} \quad n \geq d \log p$$

$$\text{Lower bound:} \quad n \geq d \log p$$

- Each method makes different *incoherence assumptions*:
  - Correlations between unrelated variables not too large

Graph neural networks

Next, we'll discuss graph neural networks, following this article:

`https://distill.pub/2021/understanding-gnns/`

Summary

- A positive distribution factors into product of potential functions on the cliques of the graph
- Graphs and independence relations are same for discrete data
- Ising models are discrete Gaussians
- No version of the graphical lasso holds for discrete data; instead, we use the parallel lasso
- Graph neural networks are defined using analogues of more familiar convolution and layers