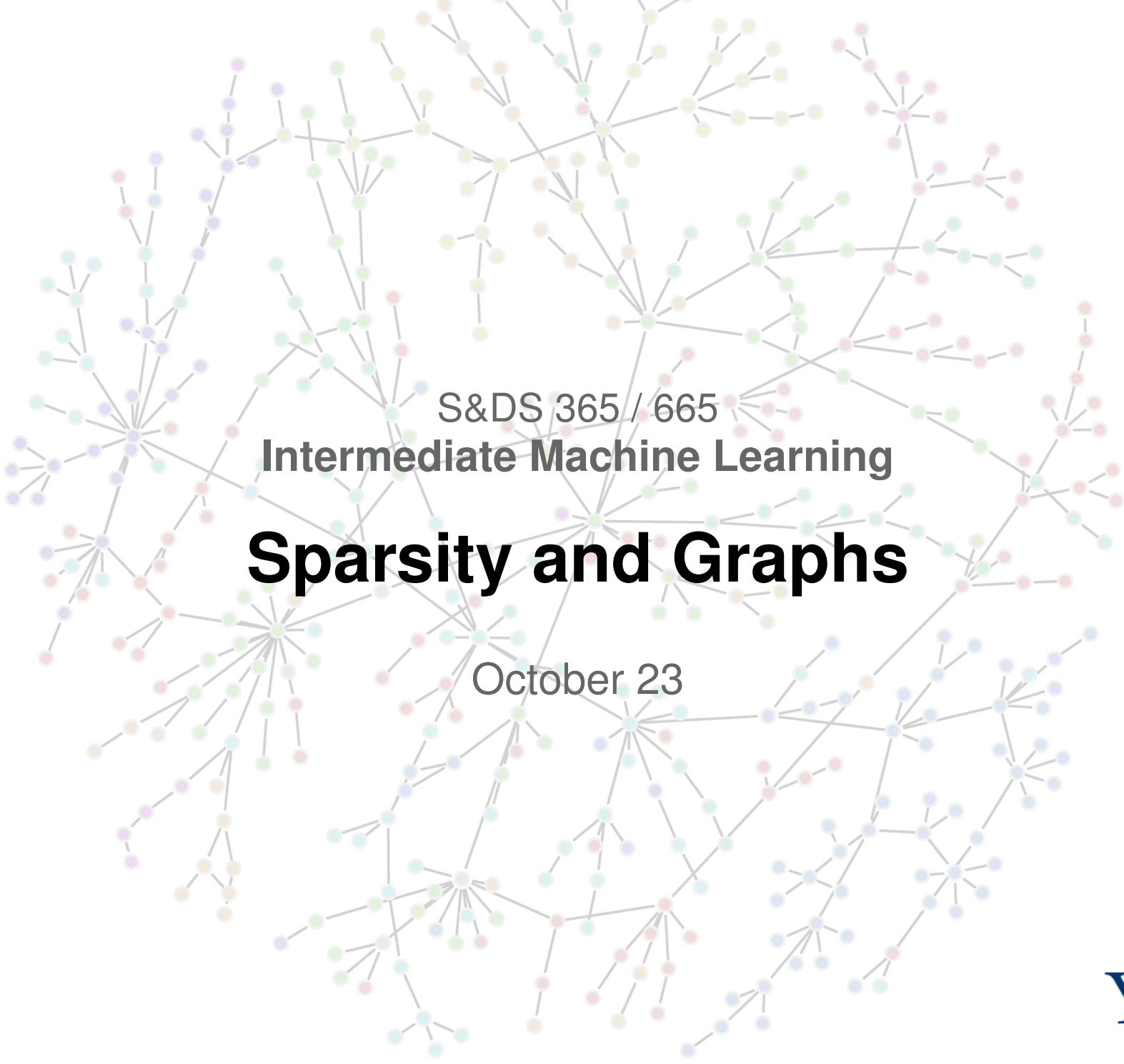


S&DS 365 / 665

**Intermediate Machine Learning**

# Sparsity and Graphs

October 23

Yale

# Welcome back!

For today:

- Where have we been? Where are we going?
- Graphs of data/distributions

# But first...a word from our sponsor

- Midterm exam grades posted before Thursday
- Mid-semester grades announced at same time
- Contact me with any concerns about grades
- Assignment 3 out; due next Wed, Nov. 1
- Assignment 4 posted at same time

# Where have we been?

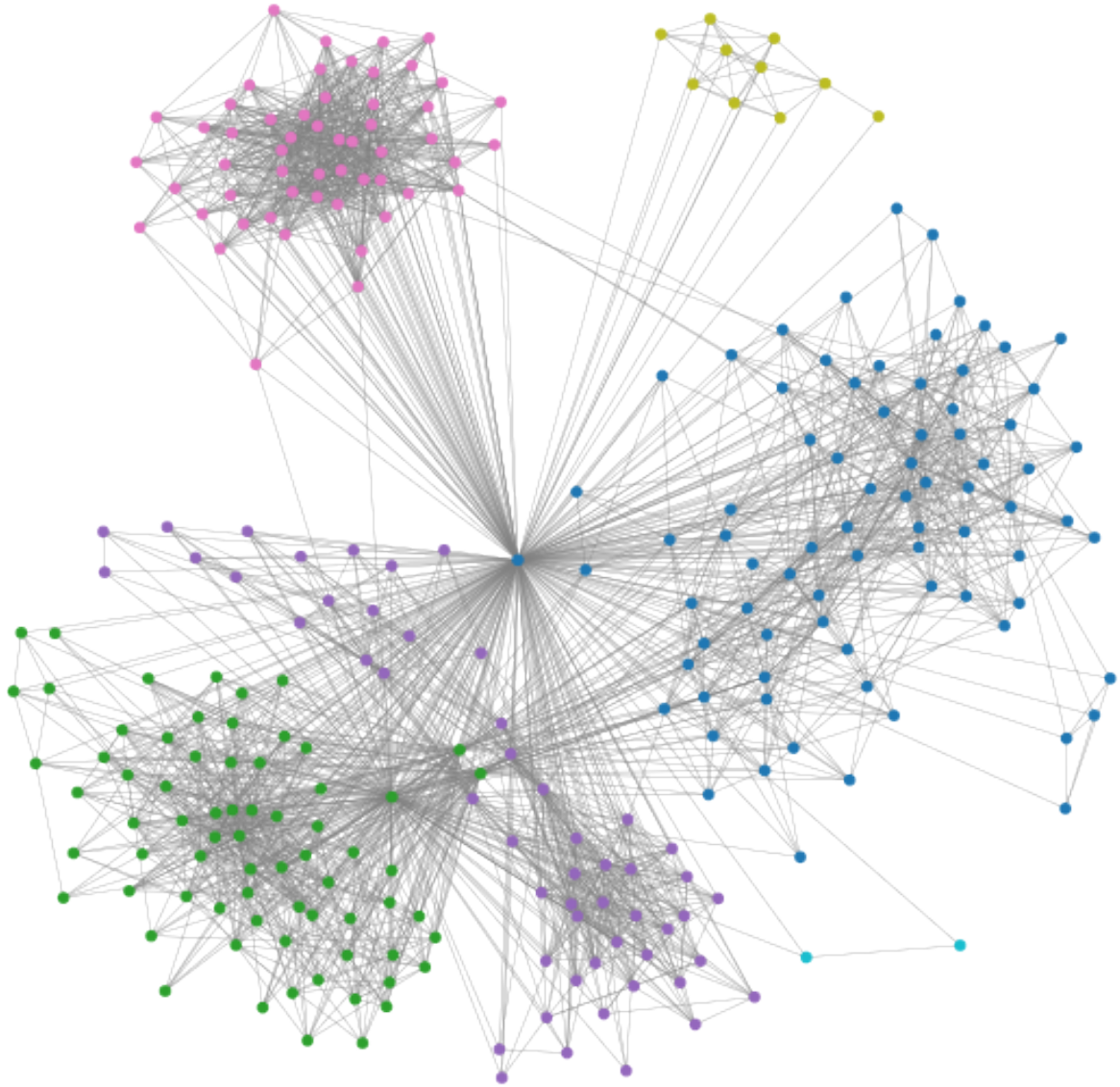
| Week | Dates | Topics | Demos & Tutorials | Lecture Slides | Readings & Notes | Assignments & Exams |
|---|---|---|---|---|---|---|
| 1 | Aug 30, Sep 1 | Sparse regression | Python elements<br>Pandas and regression<br>Lasso example | Wed: Course overview<br>Fri: Sparse regression | PML Section 11.4<br>Notes on linear regression | |
| 2 | Sep 6 | Smoothing and kernels | Smoothing example<br>Using different kernels | Wed: Smoothing | PML Sections 16.3, 17.1<br>Notes on computing the lasso | Quiz 1 |
| 3 | Sep 11, 13 | Density estimation and Mercer kernels | Density estimation demo<br>Mercer kernels (1/2)<br>Mercer kernels (2/2) | Mon: Density estimation<br>Wed: Mercer kernels | Risk bounds for local smoothing<br>Notes on Mercer kernels | Assn 1 out |
| 4 | Sep 18, 20 | Neural networks and overparameterized models | np-complete example (1/2)<br>np-complete example (2/2)<br>TensorFlow playground | Mon: Neural networks<br>Wed: Double descent | PML Sections 13.1, 13.2<br>Notes on backpropagation<br>Notes on double descent | Quiz 2 |
| 5 | Sep 25, 27 | Convolutional neural networks | Convolution demo<br>CNN demo | Mon: Double descent (continued) and Convolutional neural networks<br>Wed: CNNs (continued) | PML Section 17.2<br>Notes on Bayesian inference<br>Notes on nonparametric Bayes | Assn 1 in<br>Assn 2 out |
| 6 | Oct 2, 4 | Gaussian processes and approximate inference | Parametric Bayes<br>Gaussian processes<br>Gibbs sampling for image denoising | Mon: Gaussian processes<br>Wed: Introduction to approximate inference | Notes on simulation | Quiz 3 |
| 7 | Oct 9, 11 | Variational inference | Variational autoencoders | Mon: Variational inference<br>Wed: VAEs | PML Section 20.3<br>Notes on variational inference | Assn 2 in<br>Assn 3 out |
| 8 | Oct 16 | Midterm | | | Practice midterms | Oct 16: Midterm exam |

# Where are we going?

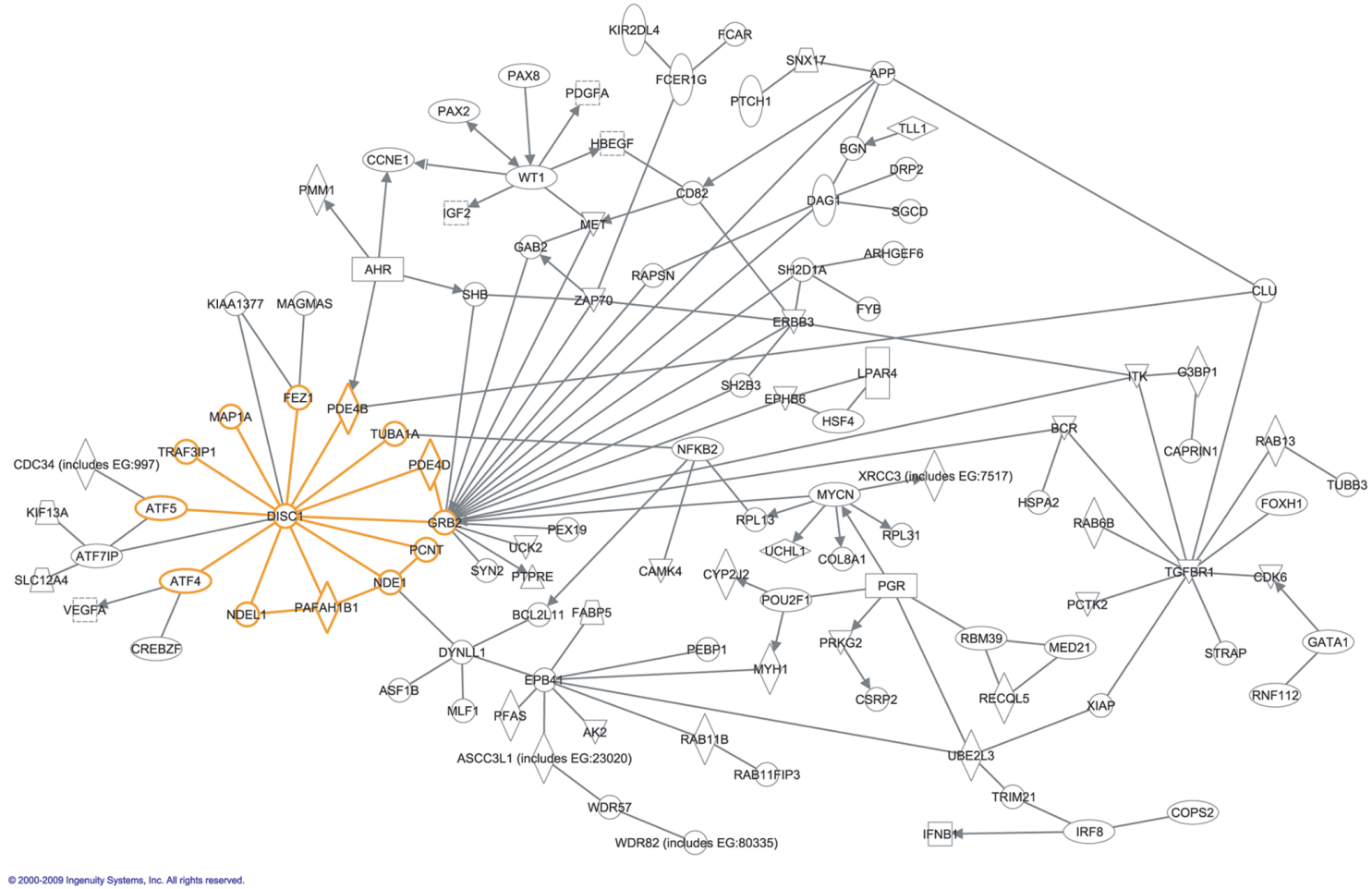
| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | Oct 16 | Midterm | | | Practice midterms | Oct 16: Midterm exam |
| 9 | Oct 23, 25 | Graphs and structure learning | Graphical lasso demo | Mon: Sparsity and graphs<br>Wed: Discrete data and graph neural nets | Notes on graphs and structure learning<br>Graph neural networks<br>PML Section 23.4 | |
| 10 | Oct 30, Nov 1 | Deep reinforcement learning | Q-learning demo<br>DQN demo | Mon: Reinforcement learning<br>Wed: Deep reinforcement learning | Sutton and Barto, Section 6.5 | Nov 1: Assn 3 in<br>Assn 4 out |
| 11 | Nov 6, 8 | Policy gradient methods | Policy gradients demo<br>Actor-critic demo | Mon: Policy gradient methods<br>Wed: Actor-critic methods | Sutton and Barto, Section 13.1-13.3, 13.5 | Quiz 4 |
| 12 | Nov 13, 15 | Sequential models | vanilla RNN<br>Fakespeare GRU | Mon: HMMs and RNNs<br>Wed: RNNs, GRUs, LSTMs, and all that | TensorFlow: Text generation<br>Notes on HMMs and Kalman filters<br>PML Chapter 15 | Assn 4 in<br>Assn 5 out |
| 13 | Nov 20, 22 | No class, Thanksgiving break | | | | |
| 14 | Nov 27, 29 | Sequence-to-sequence models and Transformers | GPT-3 demo<br>Codex demo | Mon: Sequence-to-sequence models<br>Wed: Attention and transformers | Notes on mixtures<br>PML Sections 15.4, 15.5 | Quiz 5 |
| 15 | Dec 4, 6 | Transformers<br>Societal issues | Transformer demo | Mon: Transformers and AI/ML ethics<br>Wed: Course wrap up | | Assn 5 in |
| 17 | Wed Dec 20, 9am, Room TBA | Final exam | | | Practice exams | Registrar: final exam schedule |

# Graphs

- Next two lectures touch on ML for data having graphical structure
- This is quite common

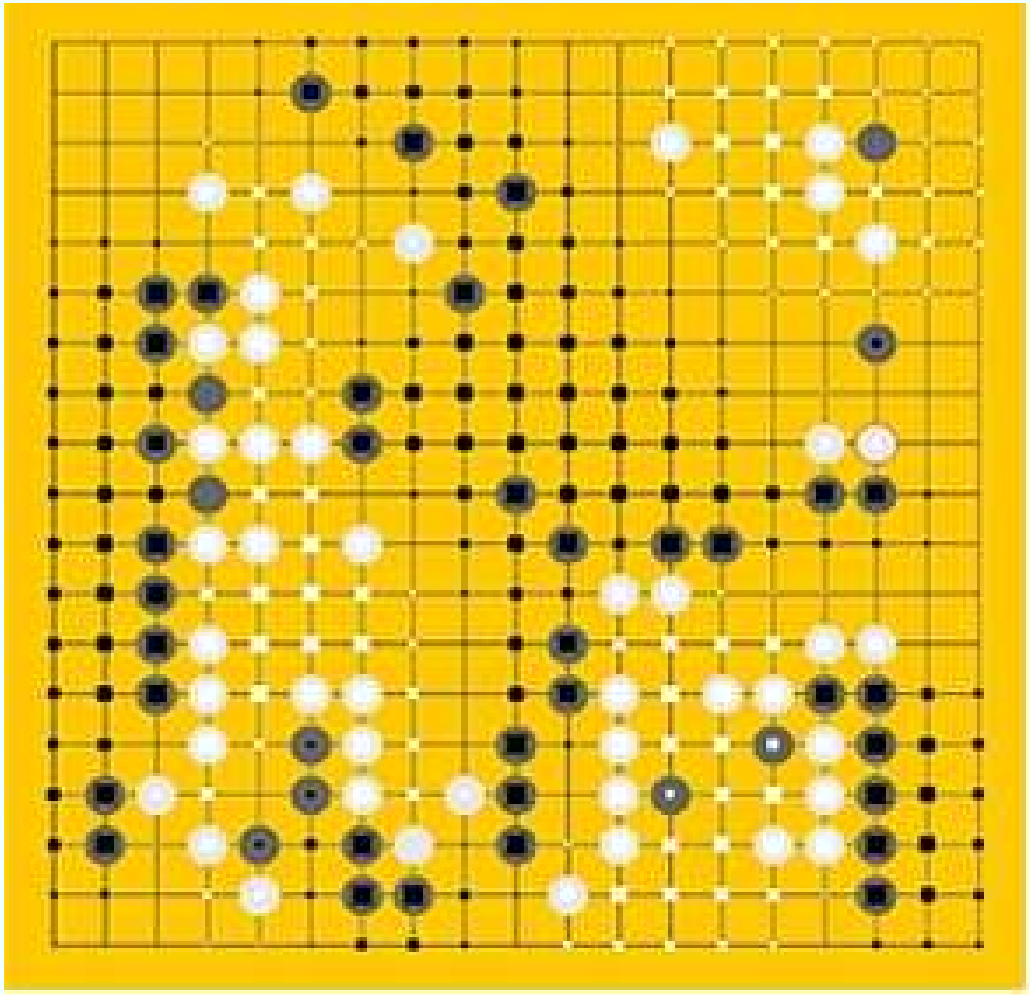


social networks (Facebook friends graph)

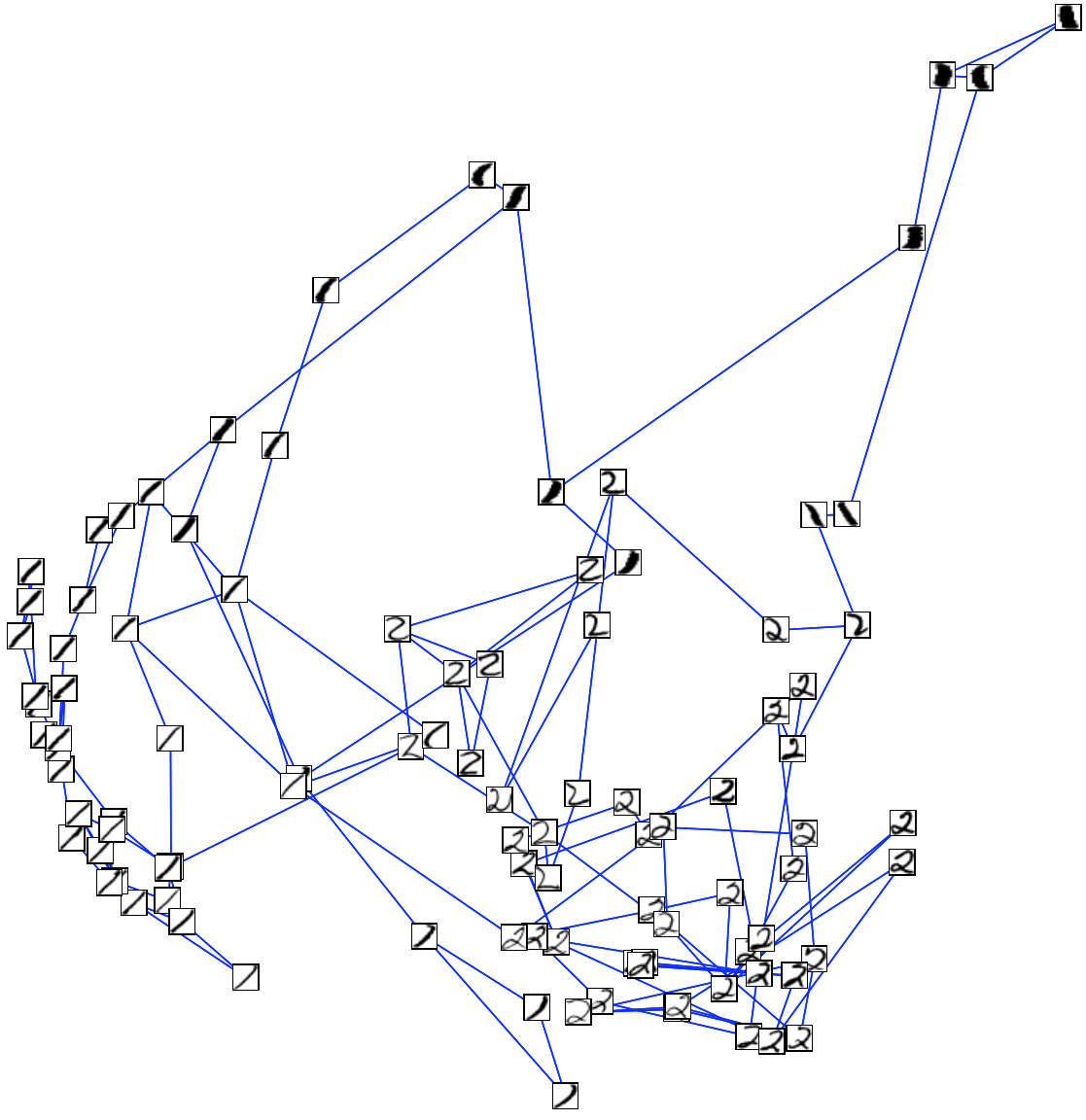




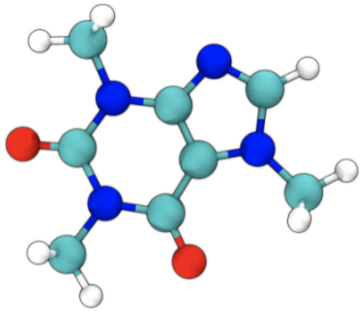

gene interaction networks — https://en.wikipedia.org/wiki/Interactome

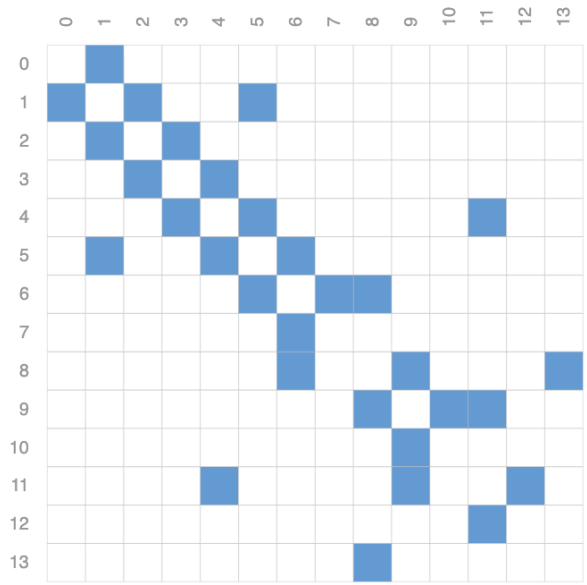


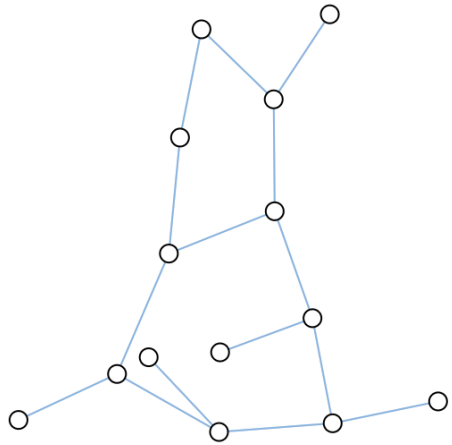
games



semi-supervised learning

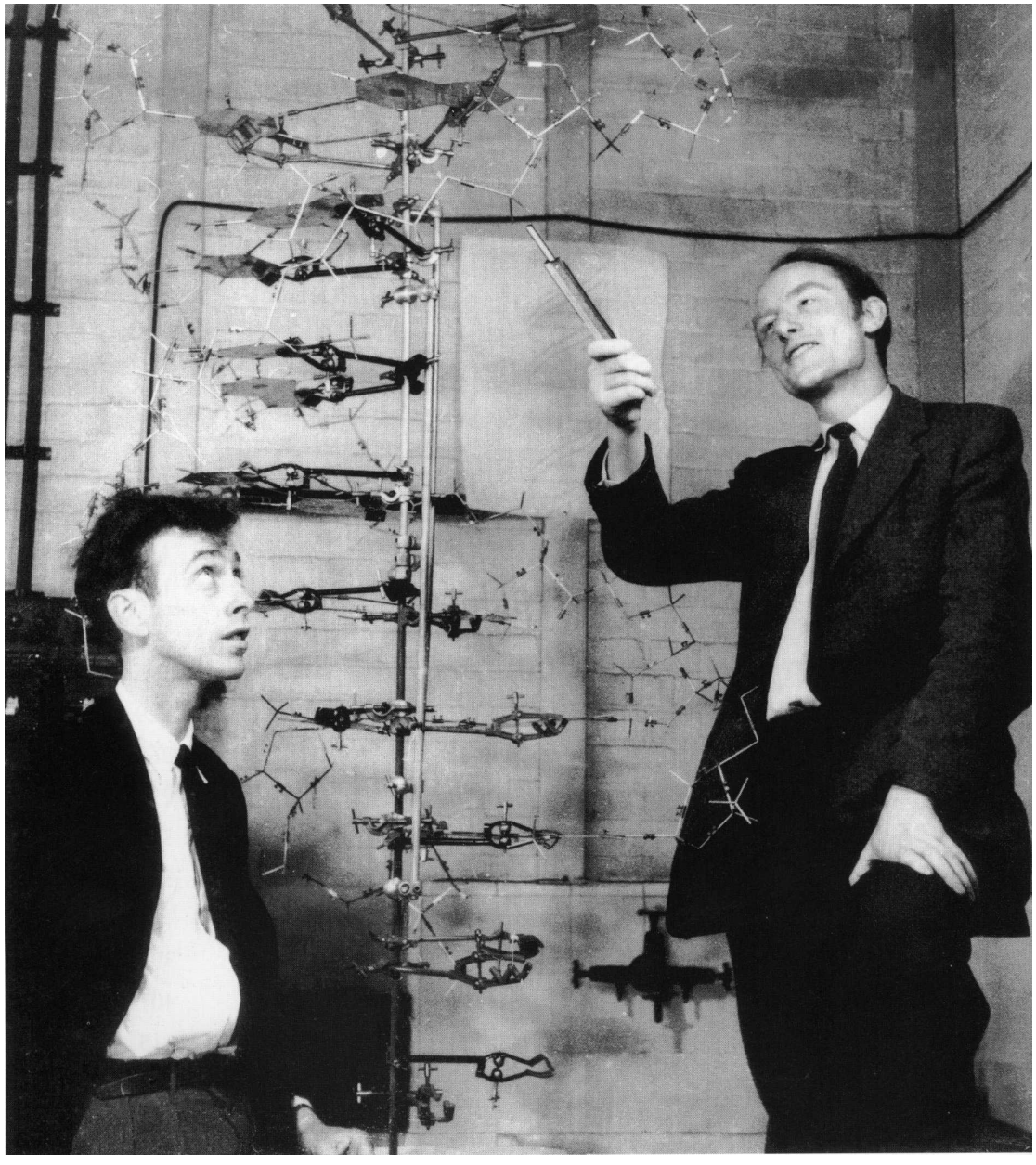


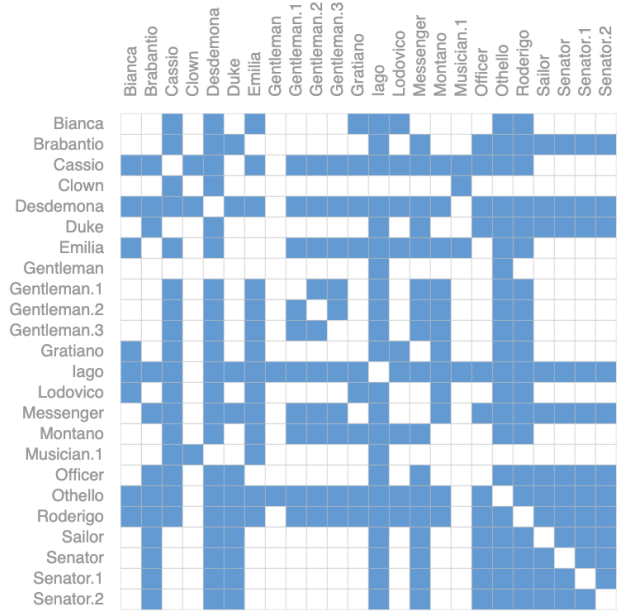


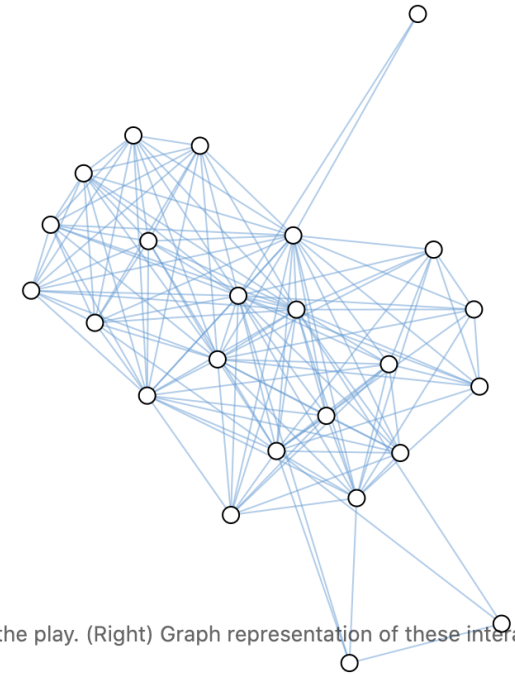



(Left) 3d representation of the Caffeine molecule (Center) Adjacency matrix of the bonds in the molecule (Right) Graph representation of the molecule.

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



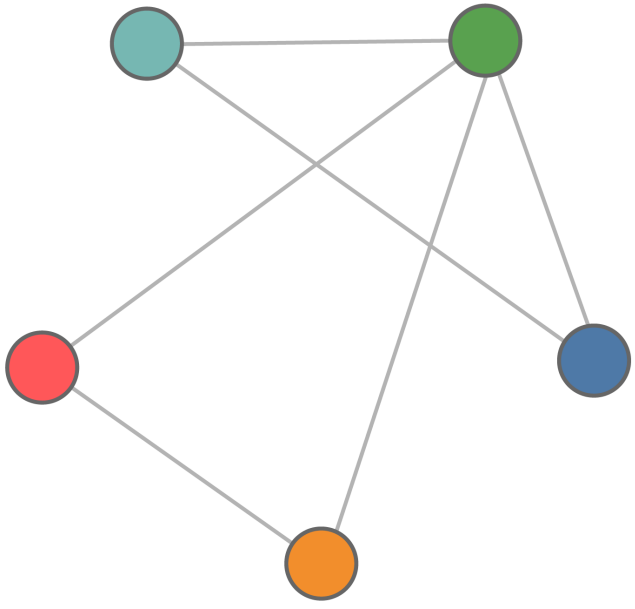


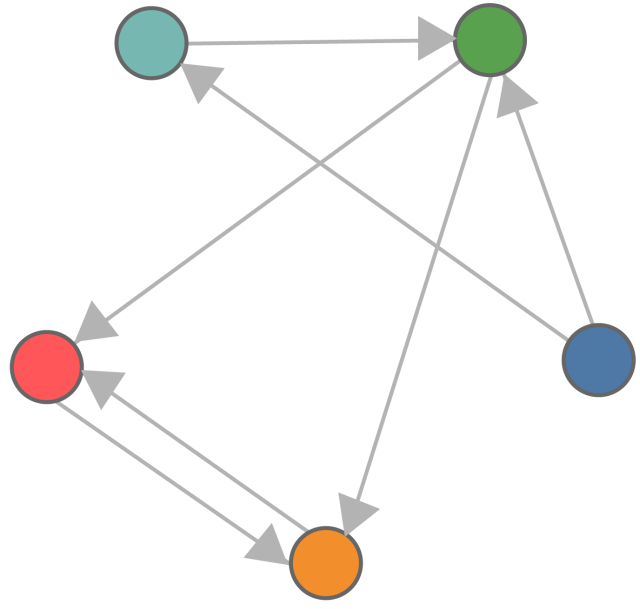
(Left) Image of a scene from the play "Othello". (Center) Adjacency matrix of the interaction between characters in the play. (Right) Graph representation of these interactions.

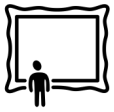
https://distill.pub/2021/gnn-intro/

Undirected graph



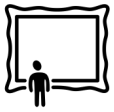
Directed graph





# Graphs

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



# 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

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.

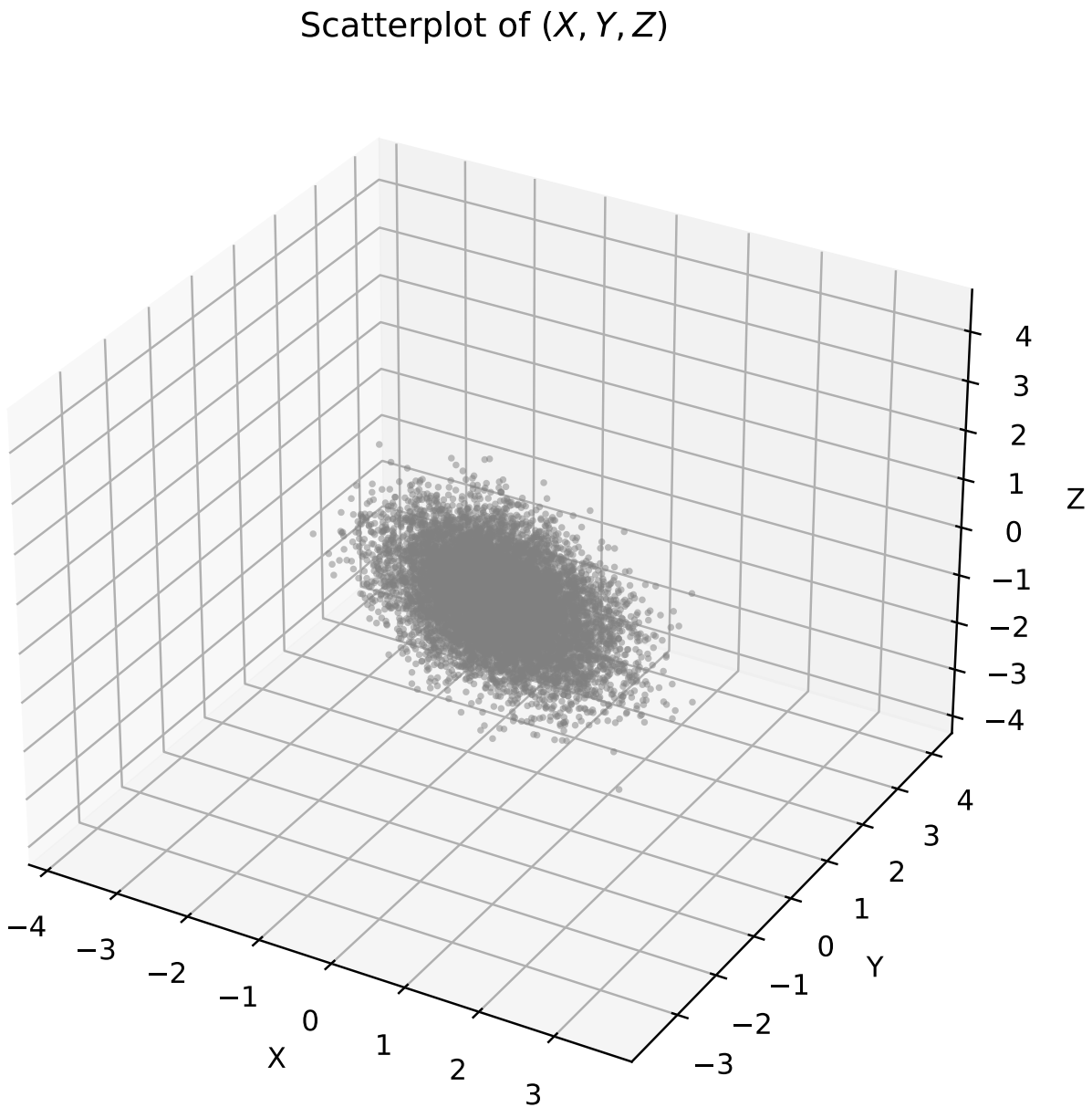
# Example: Gaussian data

We have a three-dimensional Gaussian $(X, Y, Z)$ with covariance

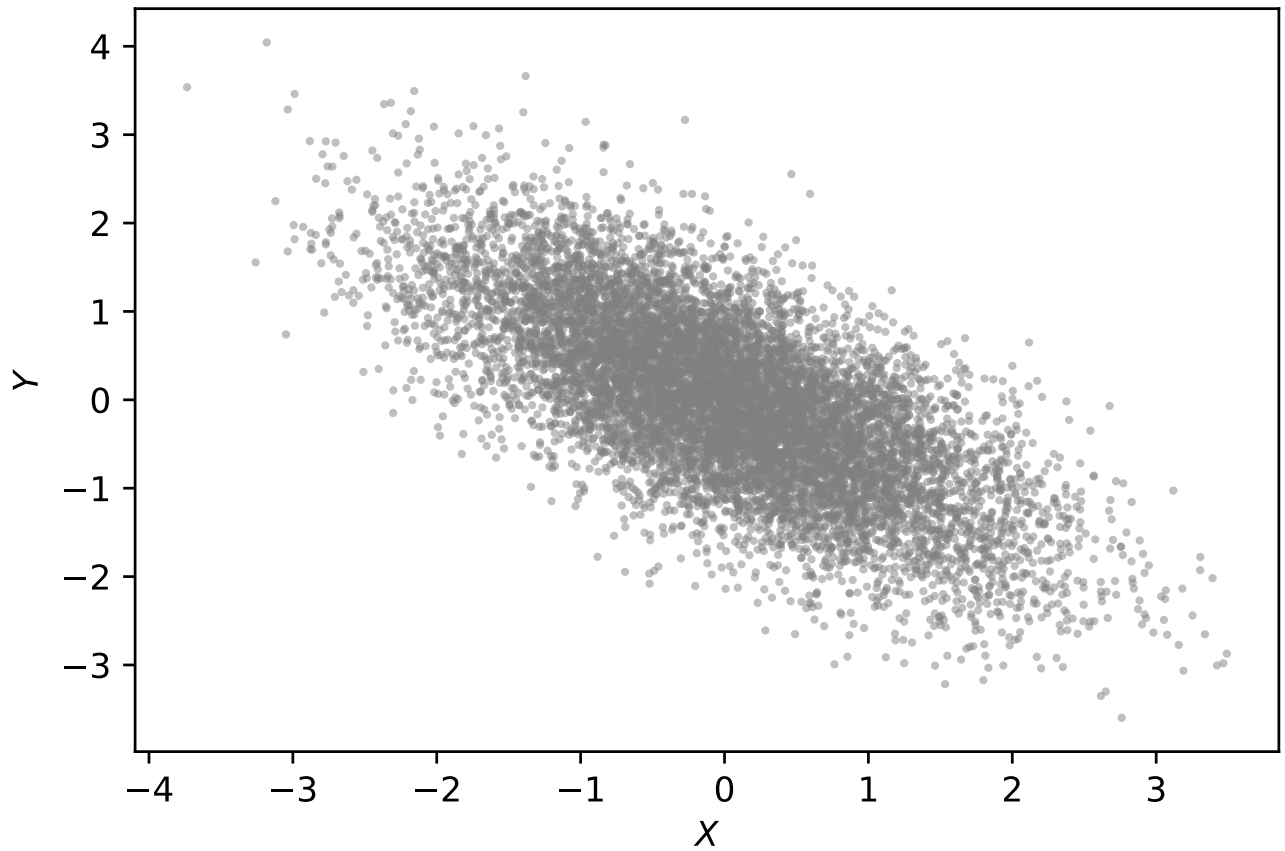
$$\Sigma = \begin{pmatrix} 3.1 & -2.4 & 2.1 \\ -2.4 & 2.6 & -2.4 \\ 2.1 & -2.4 & 3.1 \end{pmatrix}$$

So, all pairs are correlated

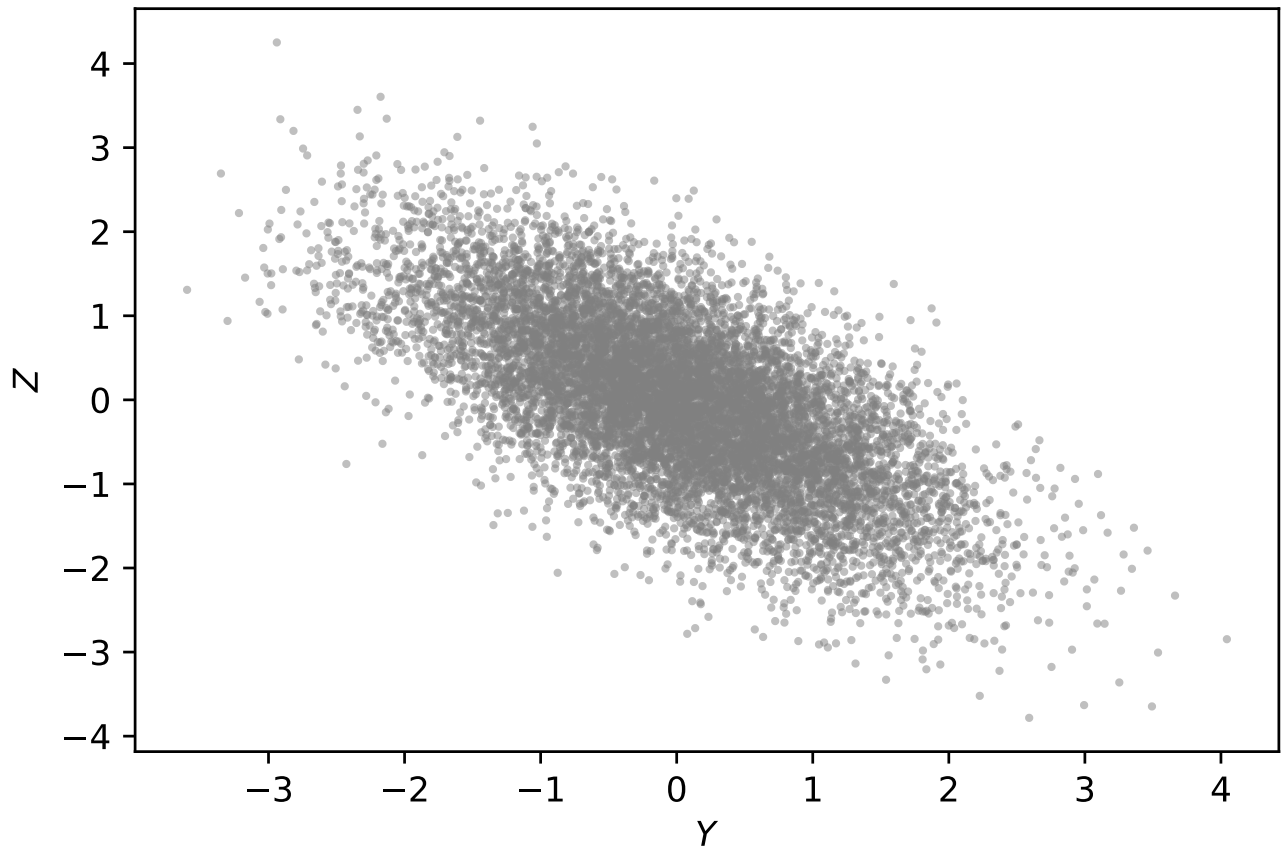
# Example: Gaussian data



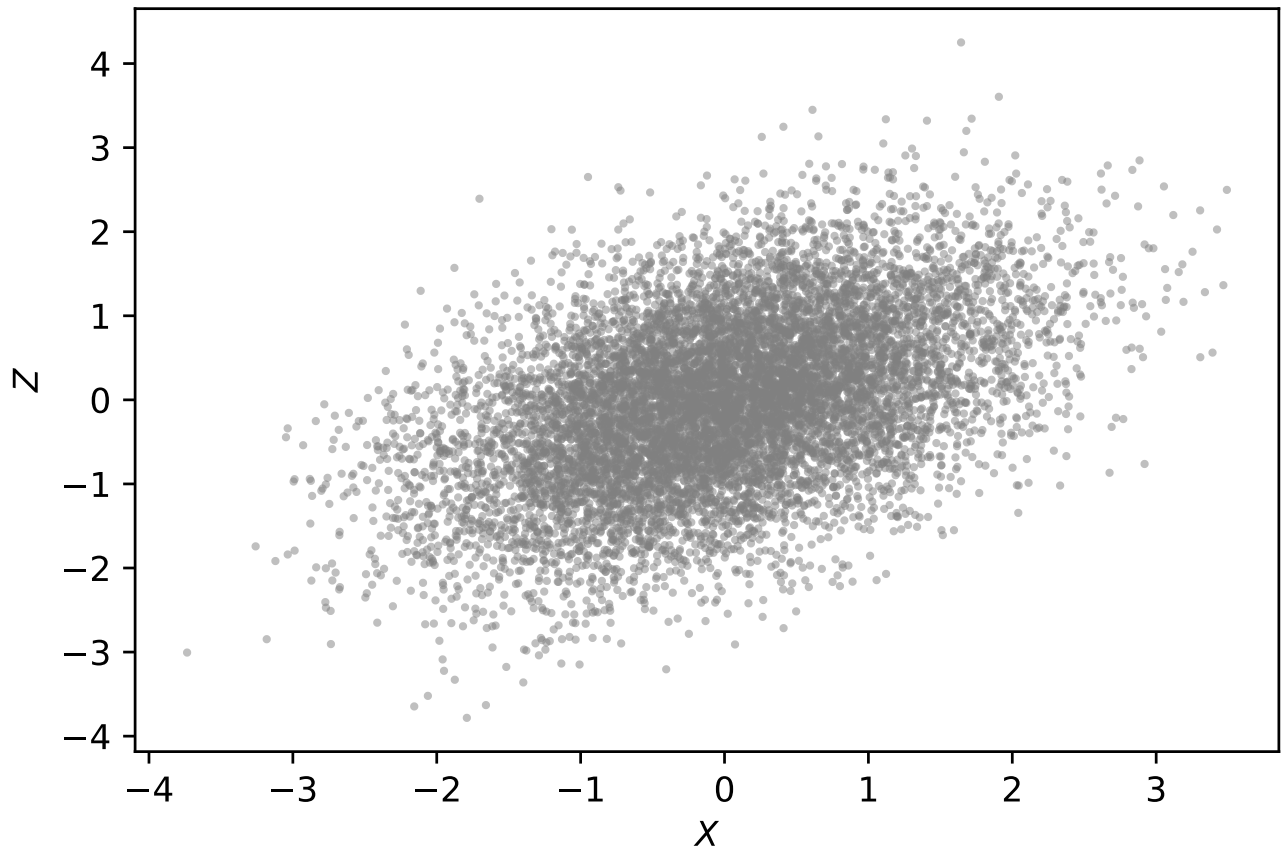
# Example: Gaussian data



# Example: Gaussian data

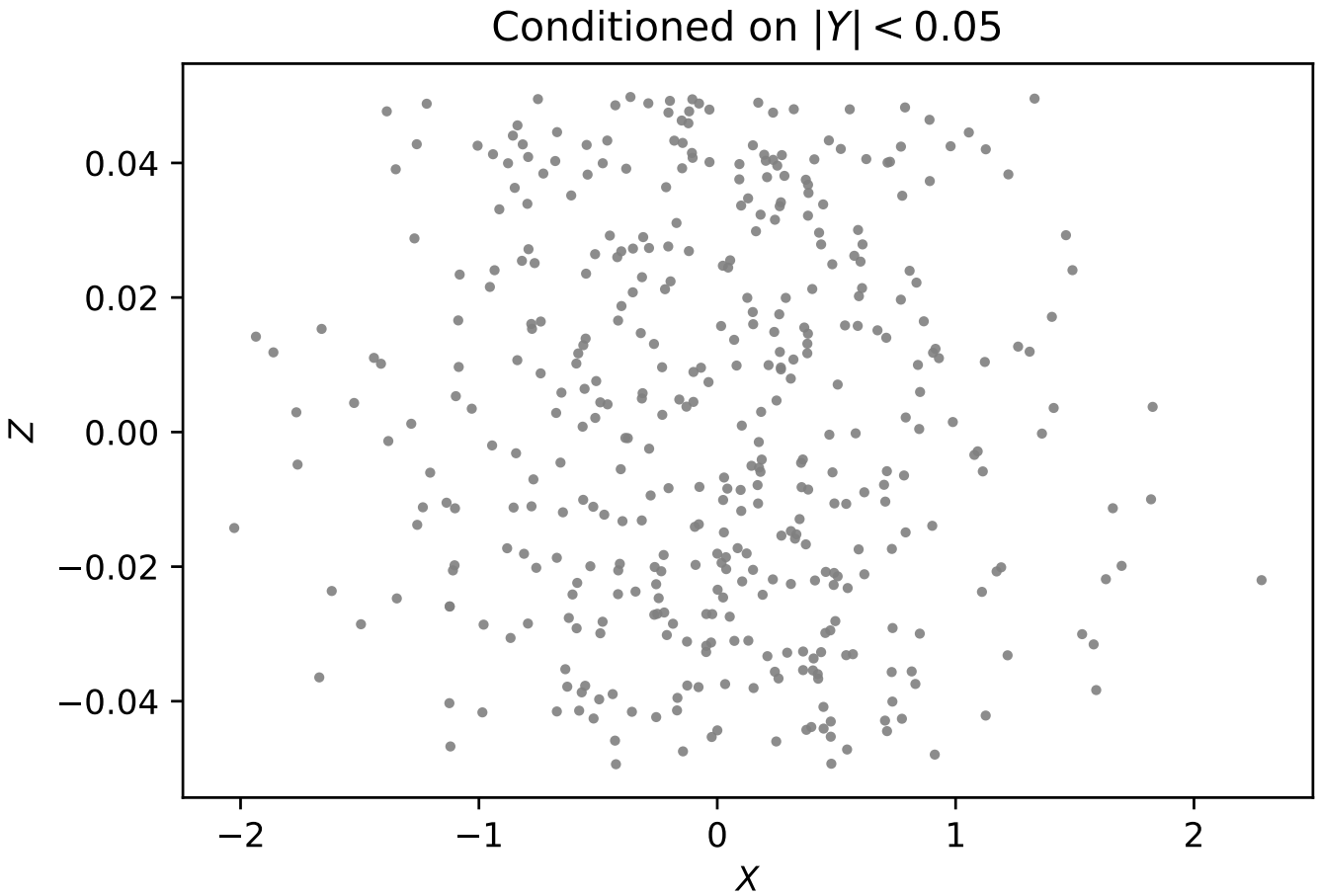


# Example: Gaussian data



# Example: Gaussian data

But when we condition on $Y \approx 0$:



# Gaussian example

This is revealed in the “precision matrix”

$$\Omega \equiv \Sigma^{-1} = \begin{pmatrix} 1 & \frac{9}{10} & 0 \\ \frac{9}{10} & 2 & \frac{9}{10} \\ 0 & \frac{9}{10} & 1 \end{pmatrix}$$

The zeros indicate conditional independence assumptions

# 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

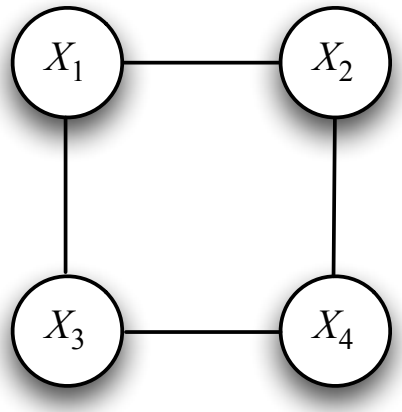
A probability distribution $P$ satisfies the *global Markov property* with respect to a graph $G$ if:

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

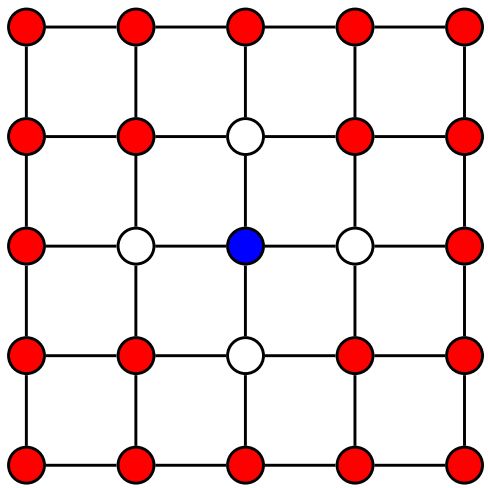



$$X_1 \perp\!\!\!\perp X_4 \mid X_2, X_3$$

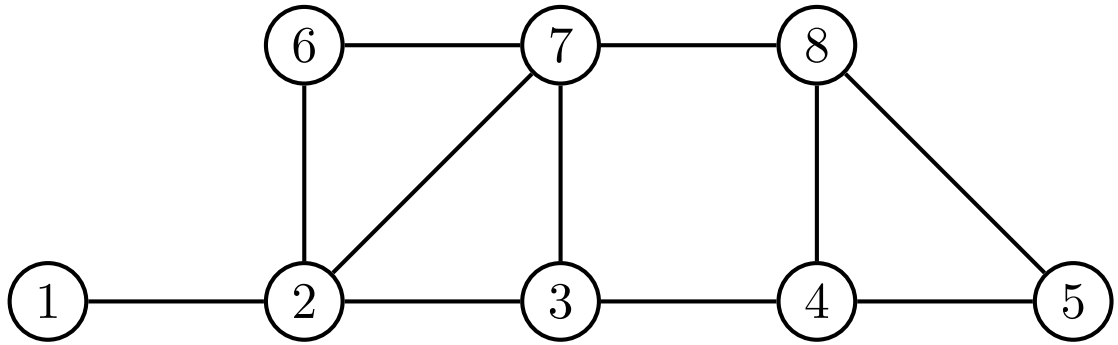
$$X_2 \perp\!\!\!\perp X_3 \mid X_1, X_4$$

# 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,

$$\{X_1, X_2\} \perp\!\!\!\perp \{X_4, X_8\} \quad \Big| \quad \{X_3, X_7\}$$

# Special case

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

$$X_i \perp\!\!\!\perp X_j \quad \Big| \quad \{X_k : k \neq i, j\}$$

# 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 $G$.

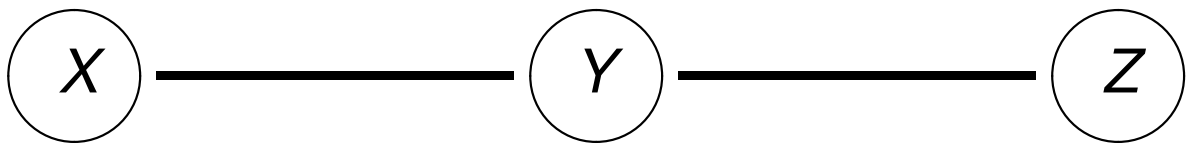
# Gaussian case

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

A zero in $\Omega$ indicates a *lack of the corresponding edge* in the graph

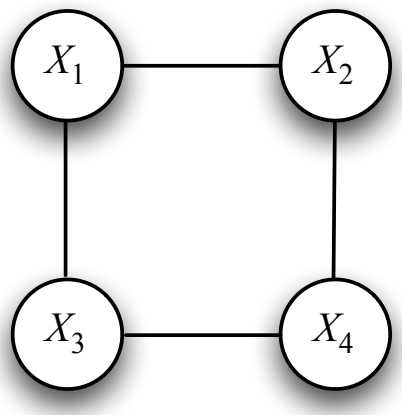
# Gaussian case

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



# Gaussian case

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




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

# The machine learning problem

*How do we estimate the graph from a sample of data?*

# 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

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

# Derivation: Where does this come from?

Assume mean is zero. Then the probability density at a data point $x$ is

$$\begin{aligned} p(x) &= \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right) \\ &= \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\left(-\frac{1}{2} x^T \Omega x\right) \\ &= \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\left(-\frac{1}{2} \text{trace}(\Omega x x^T)\right) \end{aligned}$$

Therefore, using $\log |A| = -\log |A^{-1}|$, up to an additive constant,

$$-\log p(x) = \frac{1}{2} \log |\Sigma| + \frac{1}{2} \text{trace}(\Omega x x^T) = -\frac{1}{2} \log |\Omega| + \frac{1}{2} \text{trace}(\Omega x x^T)$$

# Derivation: Where does this come from?

Summing over all the data we have

$$-\sum_{i=1}^{n} \log p(x_i) = \frac{1}{2}\sum_{i=1}^{n} \text{trace}(\Omega x_i x_i^T) - \frac{n}{2}\log|\Omega|$$
$$= \frac{n}{2}\text{trace}(\Omega S_n) - \frac{n}{2}\log|\Omega|$$

Rescaling by $2/n$ and adding the $\ell_1$ penalty, we get the objective function

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

*This is a convex function of $\Omega$*

# Graphical Lasso (glasso)

There is a simple blockwise gradient descent algorithm for minimizing this function. It is similar to the algorithm for the lasso that we studied.

Python packages: `sklearn.covariance.GraphicalLasso` and `sklearn.covariance.GraphicalLassoCV`

---

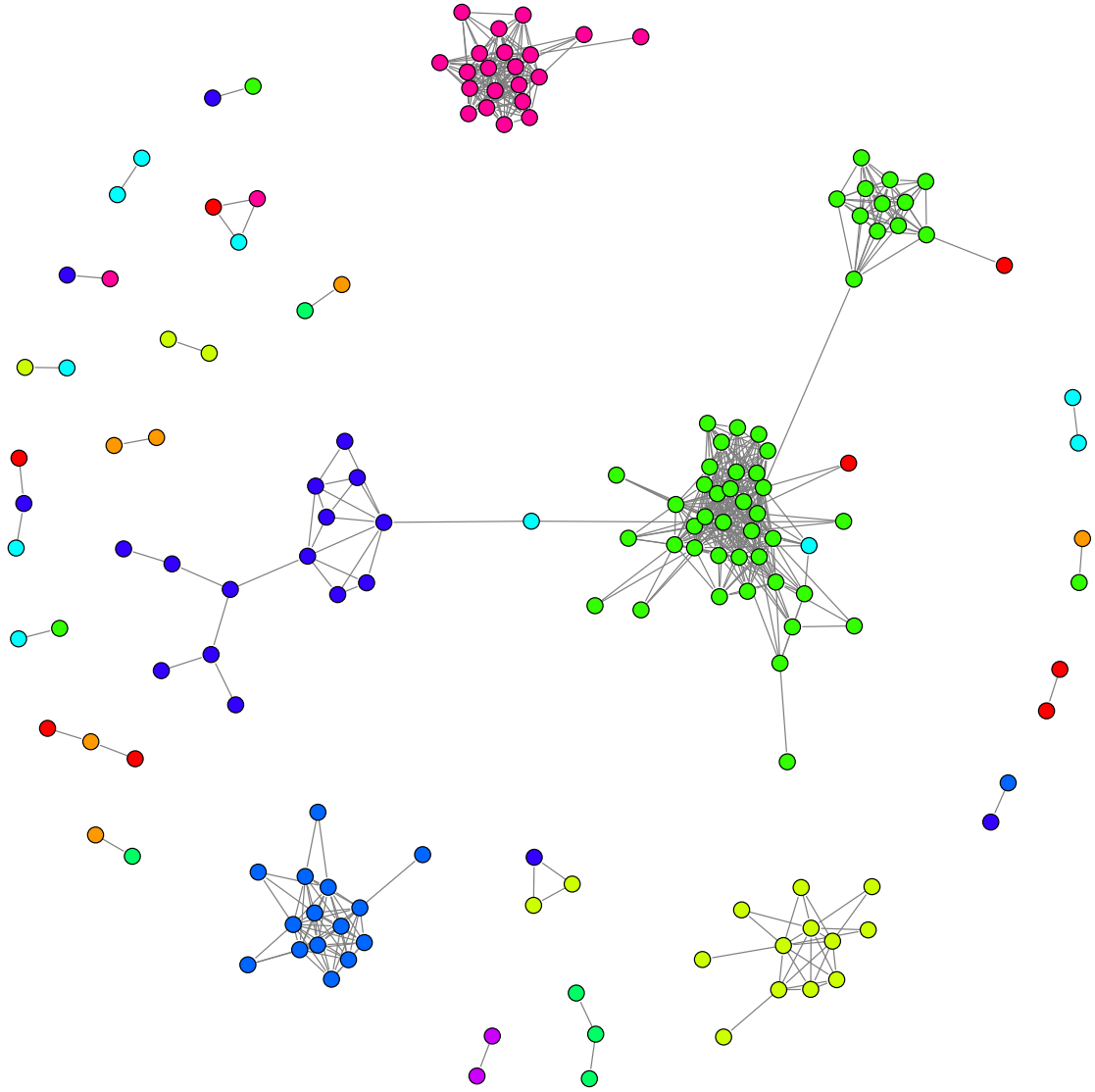
For a derivation of this algorithm, see “Notes on Graphs and Structure Learning”

# Graphs on the S&P 500

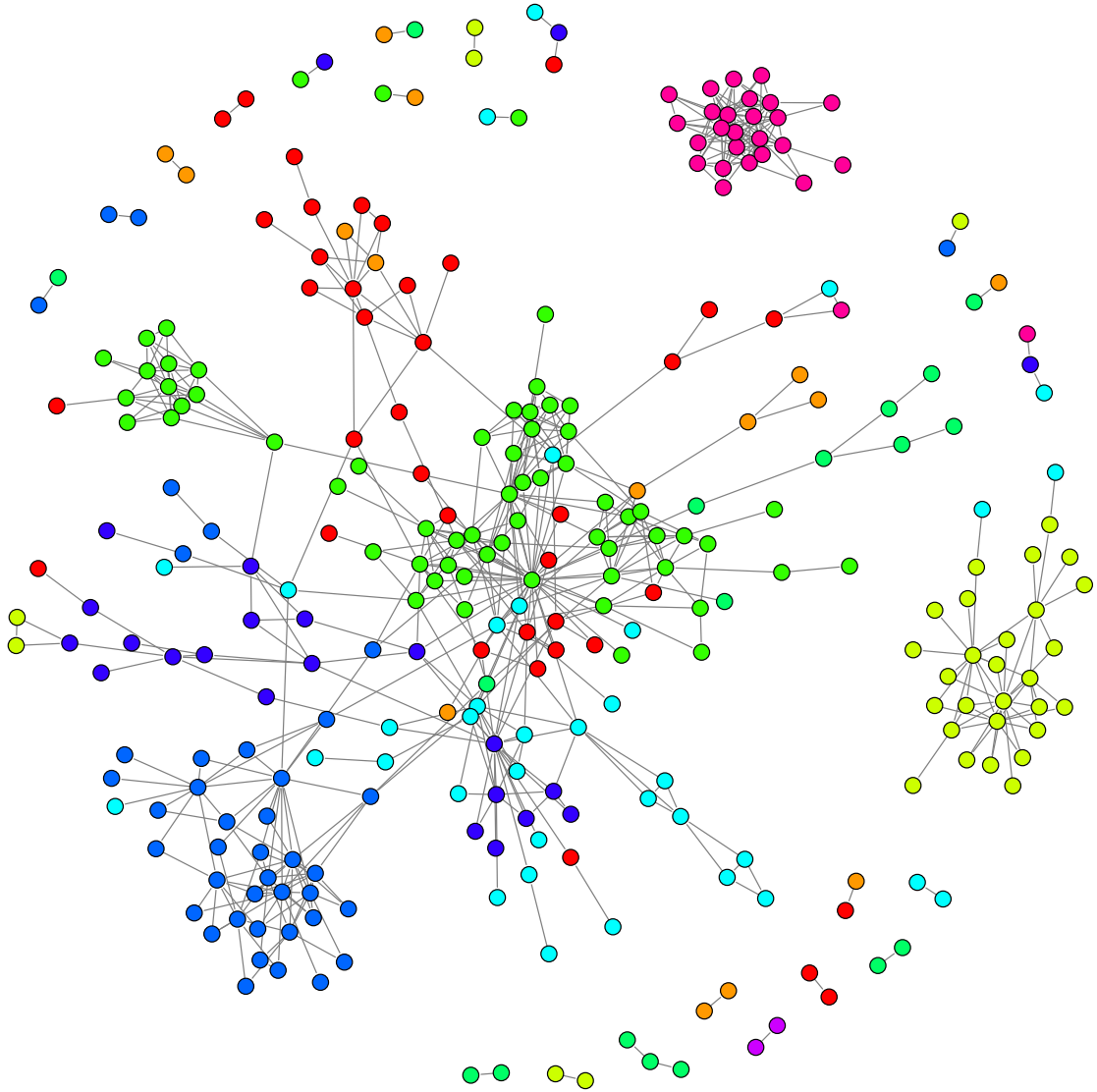
- Data from Yahoo! Finance (`finance.yahoo.com`).
- Daily closing prices for 452 stocks in the S&P 500 between 2003 and 2008 (before onset of the "financial crisis").
- Log returns $X_{tj} = \log\left(S_{t,j}/S_{t-1,j}\right)$ .
- Outliers capped at $\pm 6\sigma$.
- In following graphs, each node is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



# S&P 500: Parallel Lasso



# Example Neighborhood

Yahoo Inc. (Information Technology):

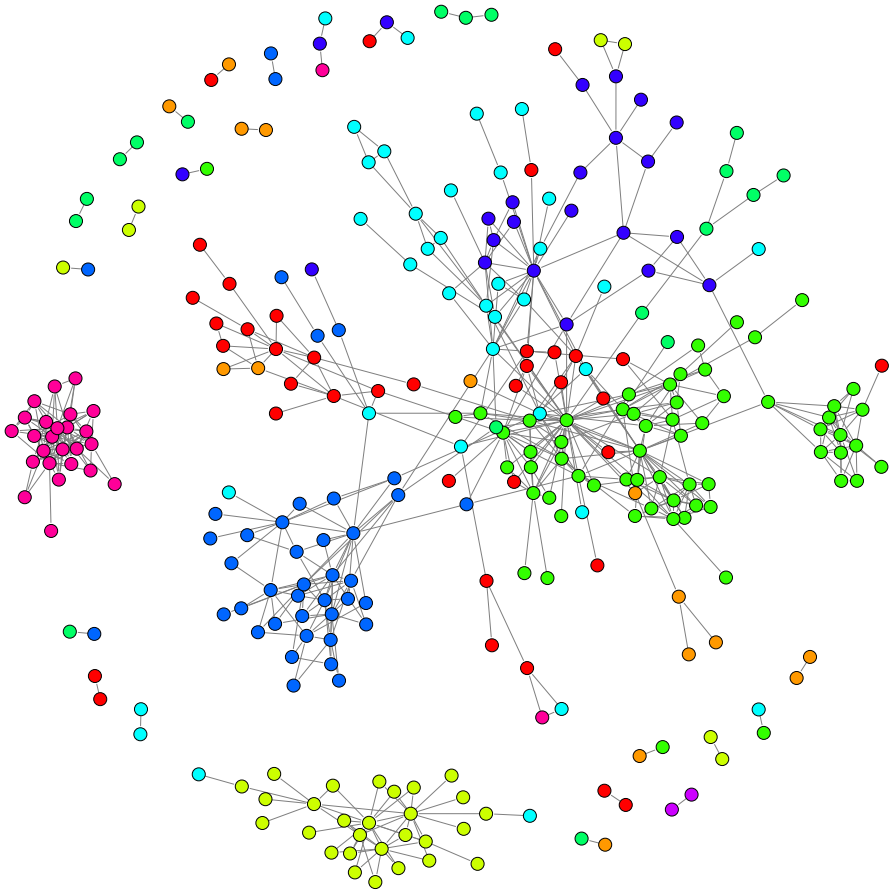
- Amazon.com Inc. (Consumer Discretionary)
- eBay Inc. (Information Technology)
- NetApp (Information Technology)

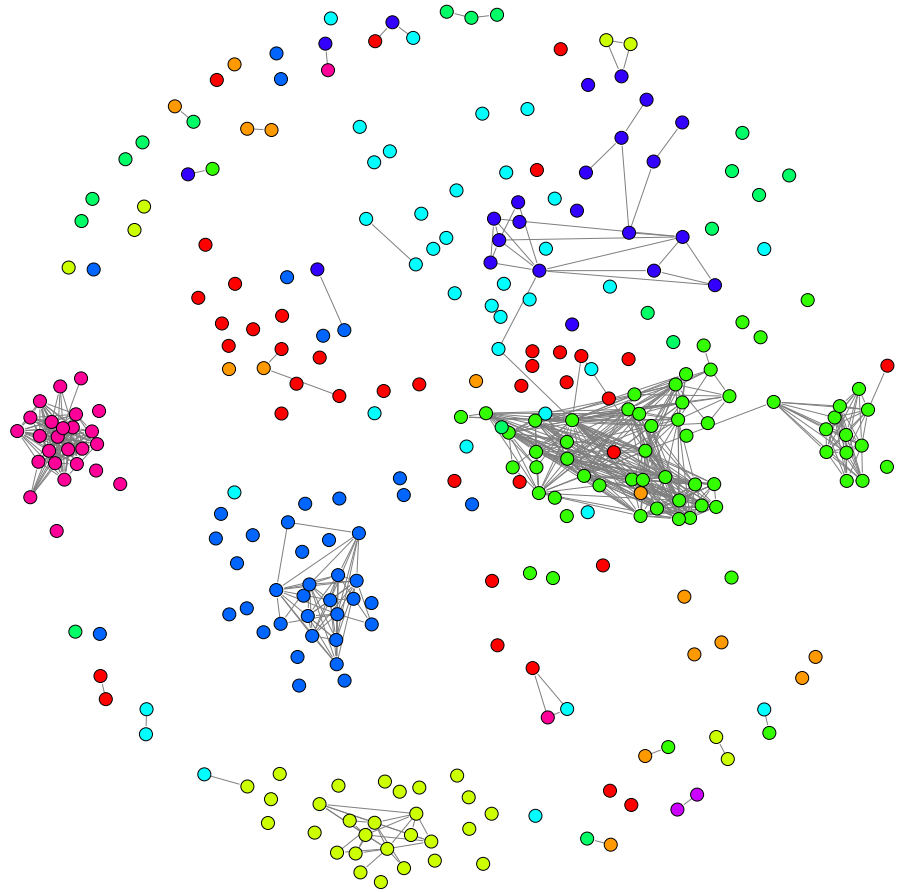
# Example Neighborhood

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)

# Parallel vs. Graphical





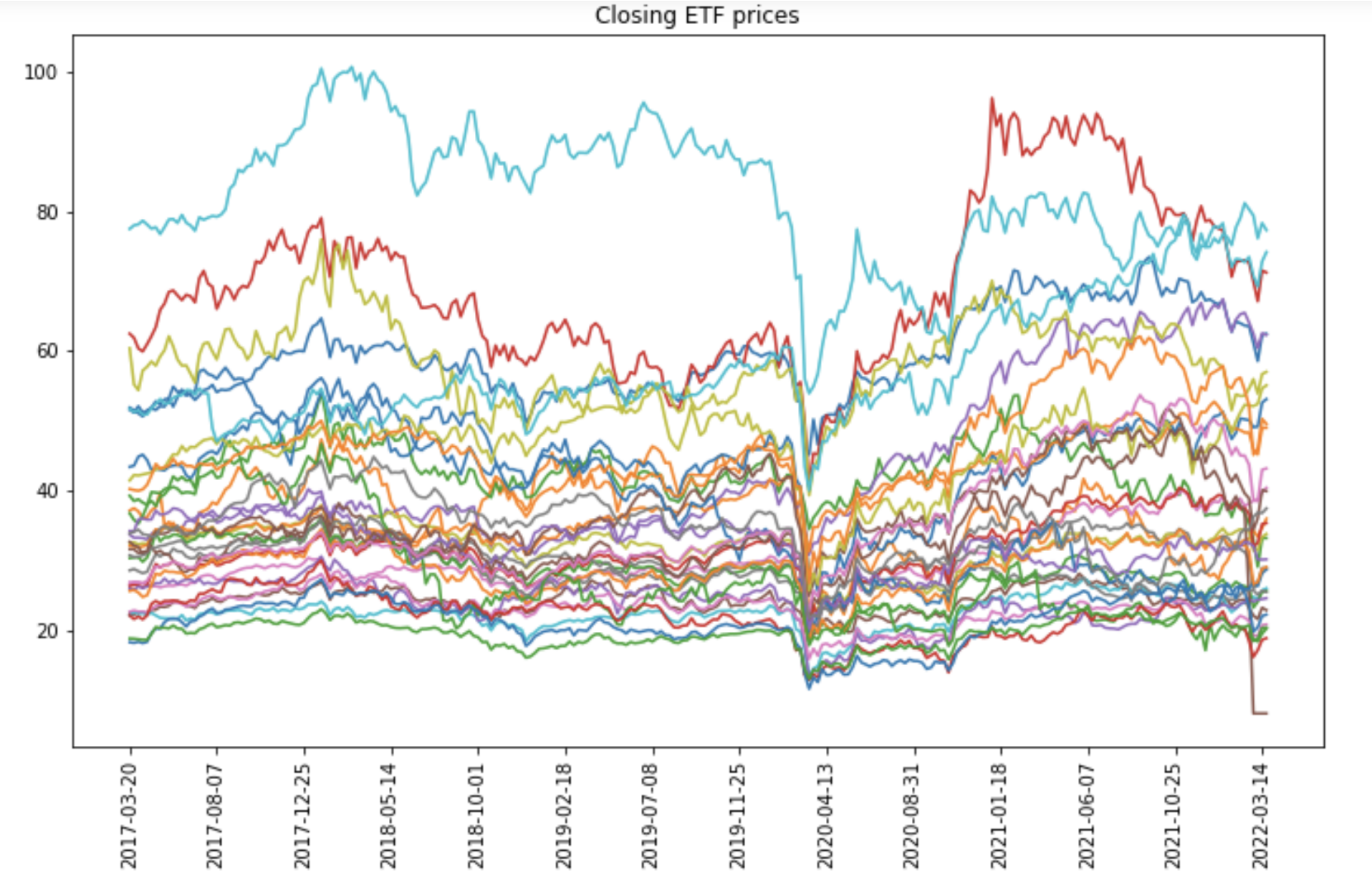
# Choosing $\lambda$

Can use:

1. Cross-validation
2. BIC = log-likelihood - $(p/2)\log n$
3. AIC = log-likelihood - $p$

where $p$ = number of parameters.

# Let’s go to the demo!



# Summary

- Graphs encode conditional independence assumptions
- Sparse graphs represent low-dimensional structure in high dimensional data
- Gaussian case: Graph read off from precision matrix
- Graphical lasso used to estimate the graph