Machine Learning for Data Science (CS4786)
Lecture 27

Last Lecture

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2016fa/

Announcements

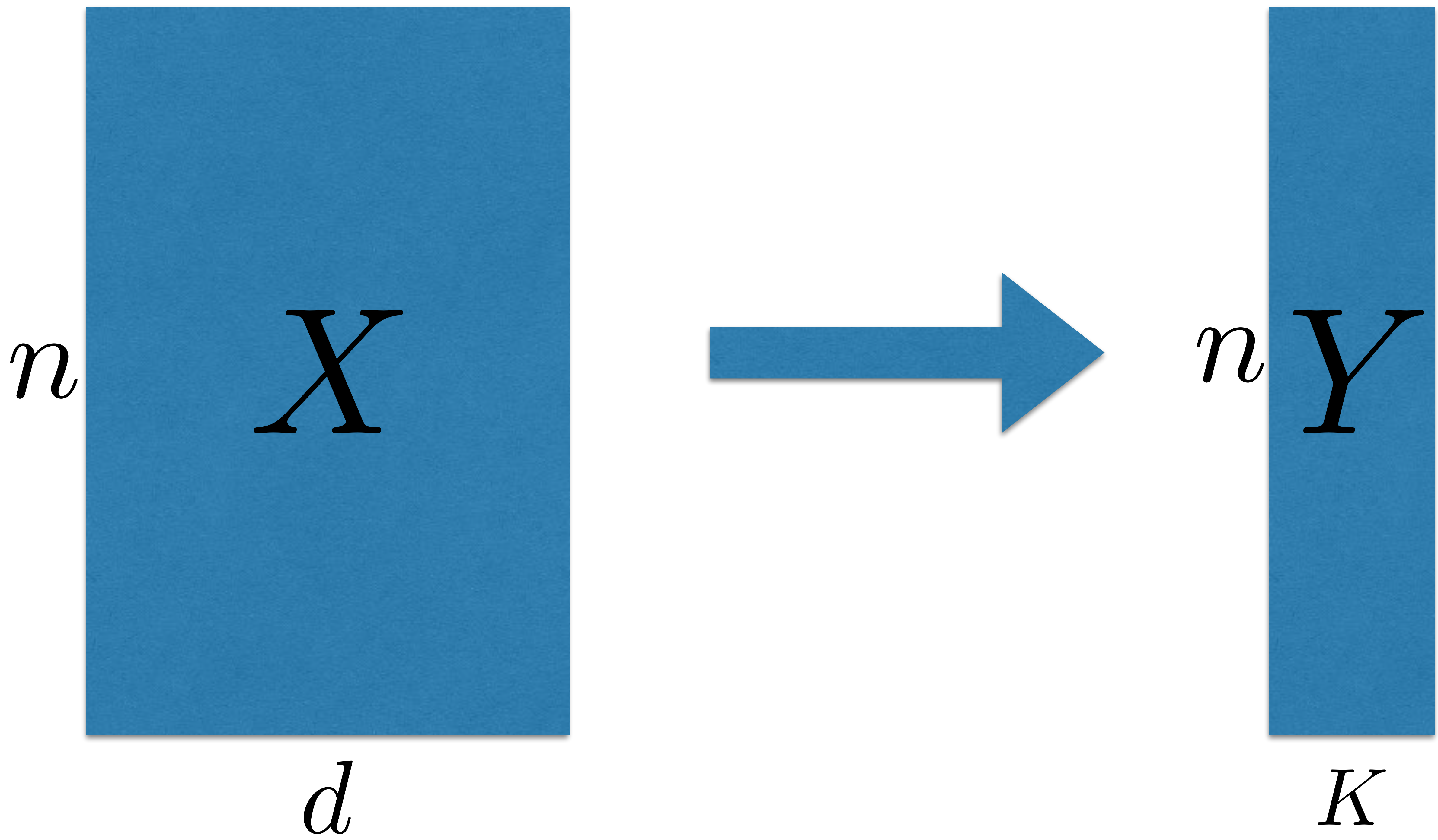
- Competition II deadline is a hard one.
- Assignment 5 and 6 we should finish grading tonight. Hope to upload by tomorrow.
- Make sure all previous assignments are graded, if not make sure you email me.
- Make sure you fill out the course eval forms.

Competition II: some advice

- Report is extremely important
- Make sure you hit all the points in the grading rubric in your reports
- Explain how you set up the model to use the data, your exact model and rationale in a clear fashion
- Don't have a laundry list of methods with corresponding figures, explain!
- We want to know your thought process through the report.

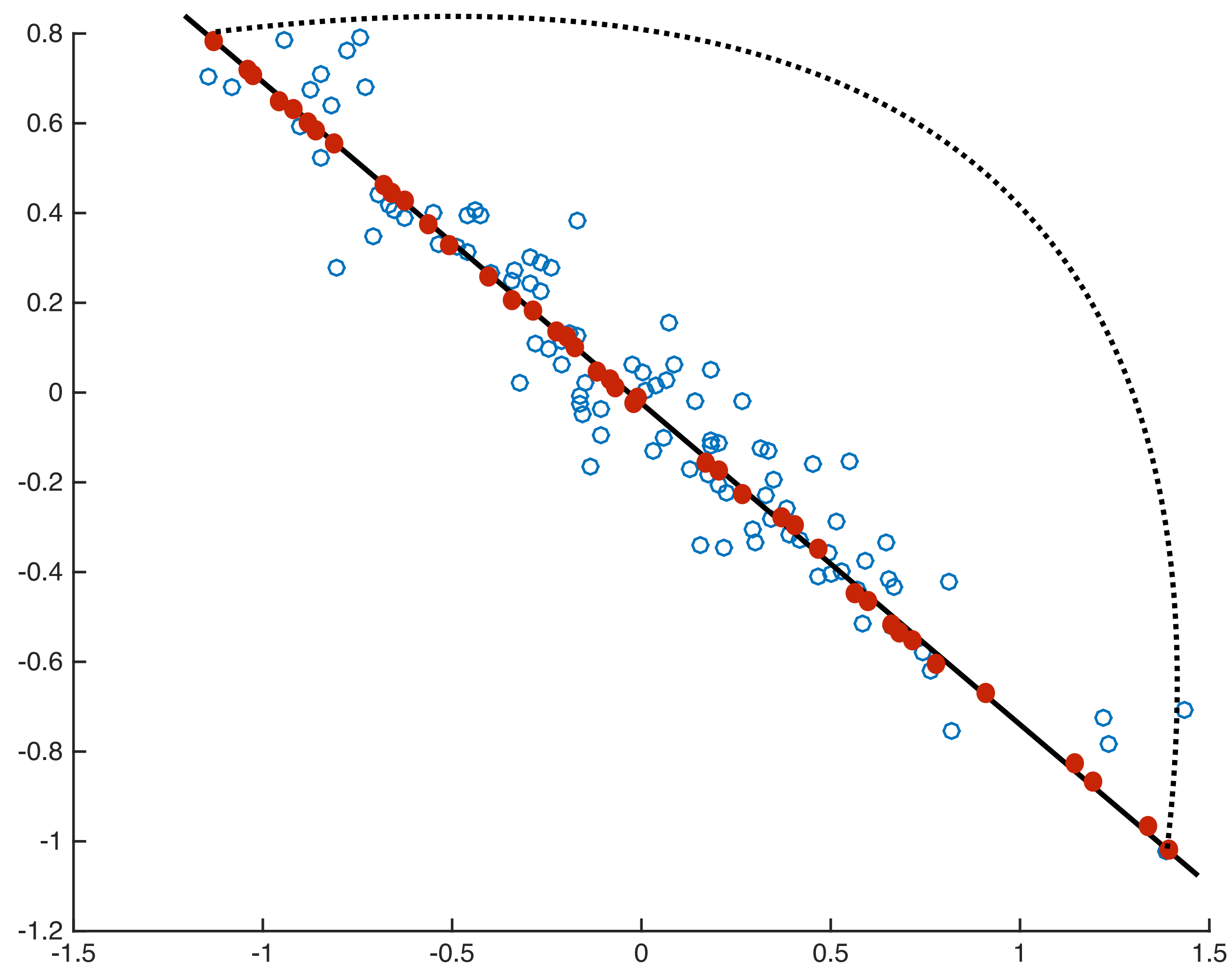
What have we covered so far?

Dimensionality Reduction



PCA: Variance Maximization

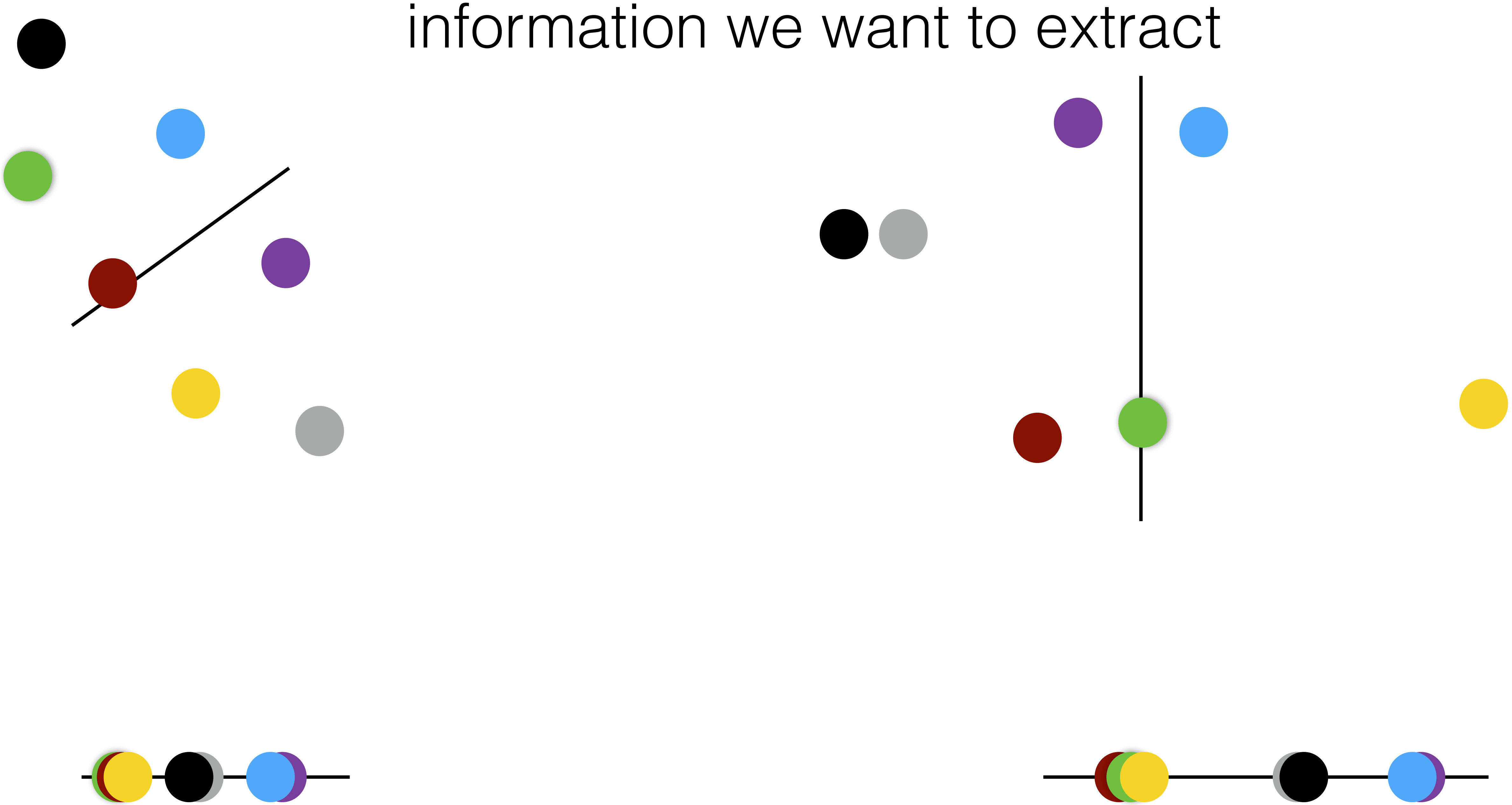
Keep only directions with maximal information (spread)



$$\text{First principal direction} = \text{ Top eigen vector}$$

Which Direction to Pick?

Data naturally split into two parts: both carry common information we want to extract



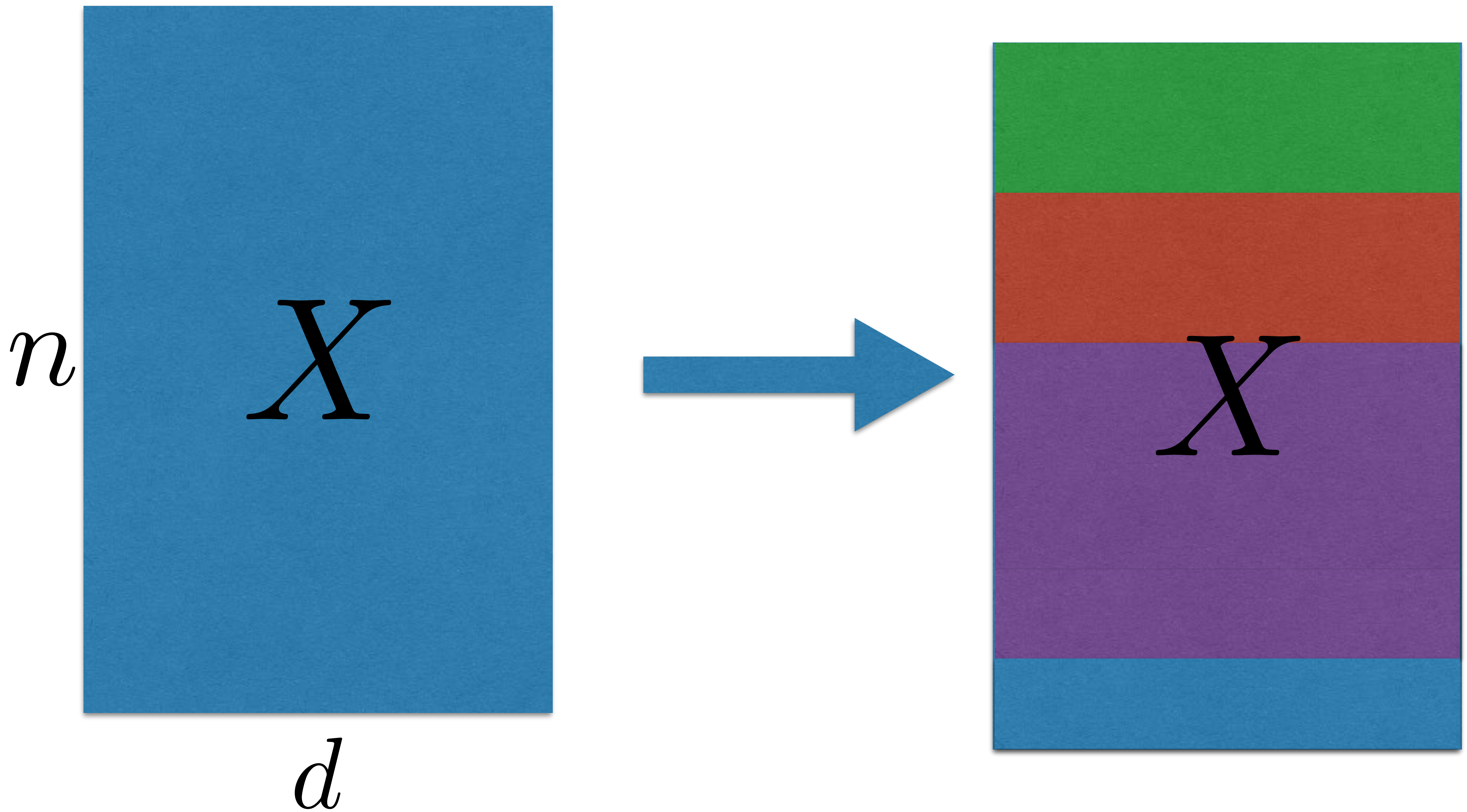
Direction has large correlation

Pick a Random W

Handle lots of very large dimensional data
Preserve interpoint distances

$$Y = X \times \left. \begin{array}{c} \begin{bmatrix} +1 & \dots & -1 \\ -1 & \dots & +1 \\ +1 & \dots & -1 \\ & \vdots & \\ +1 & \dots & -1 \end{bmatrix} d \\ K \end{array} \right/ \sqrt{K}$$

CLUSTERING



K-MEANS CLUSTERING

Look for nice round clusters

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^1$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \|\mathbf{x}_t - \hat{\mathbf{r}}_j^m\|$$

  2. For each $j \in [K]$, set new representative as

$$\hat{\mathbf{r}}_j^{m+1} = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t$$

  3. $m \leftarrow m + 1$

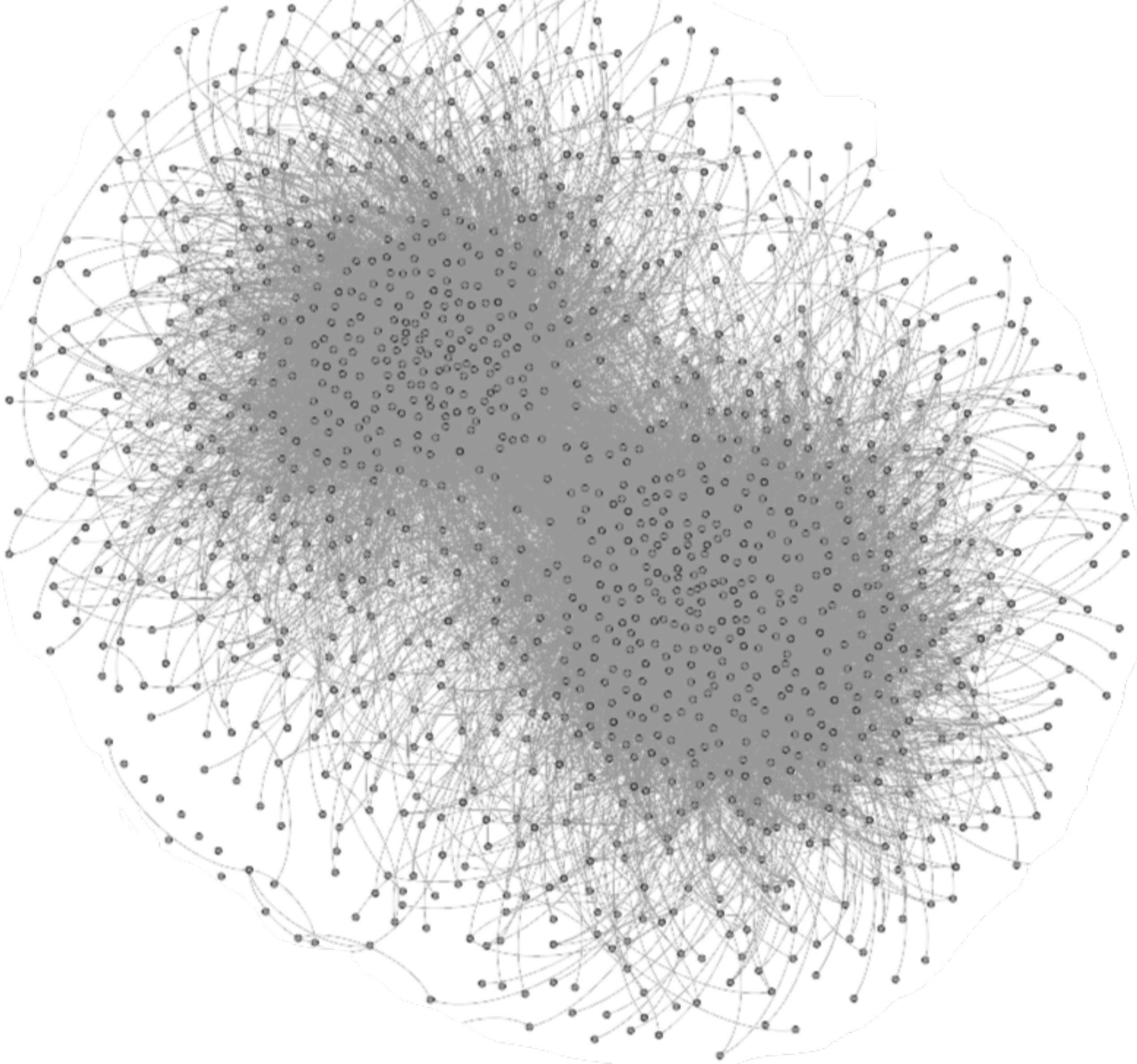
Single Link Clustering

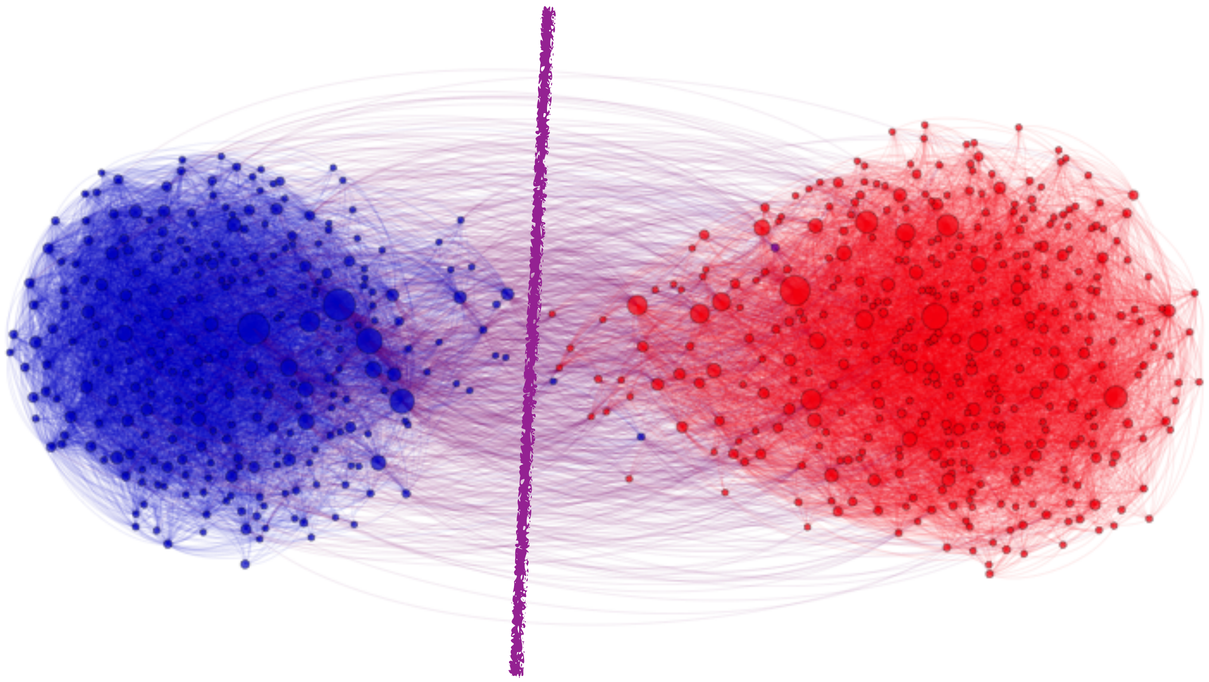
Look for tightly connected clusters

- Initialize n clusters with each point $\mathbf{x}_t$ to its own cluster
- Until there are only K clusters, do
  1. Find closest two clusters and merge them into one cluster
  2. Update between cluster distances (called proximity matrix)

Tell me who your friends are . . .

Spectral Clustering: Look for clusters with few edges between clusters and clusters themselves are dense.





- Cluster nodes in a graph.
- Analysis of social network data.

Probabilistic Models

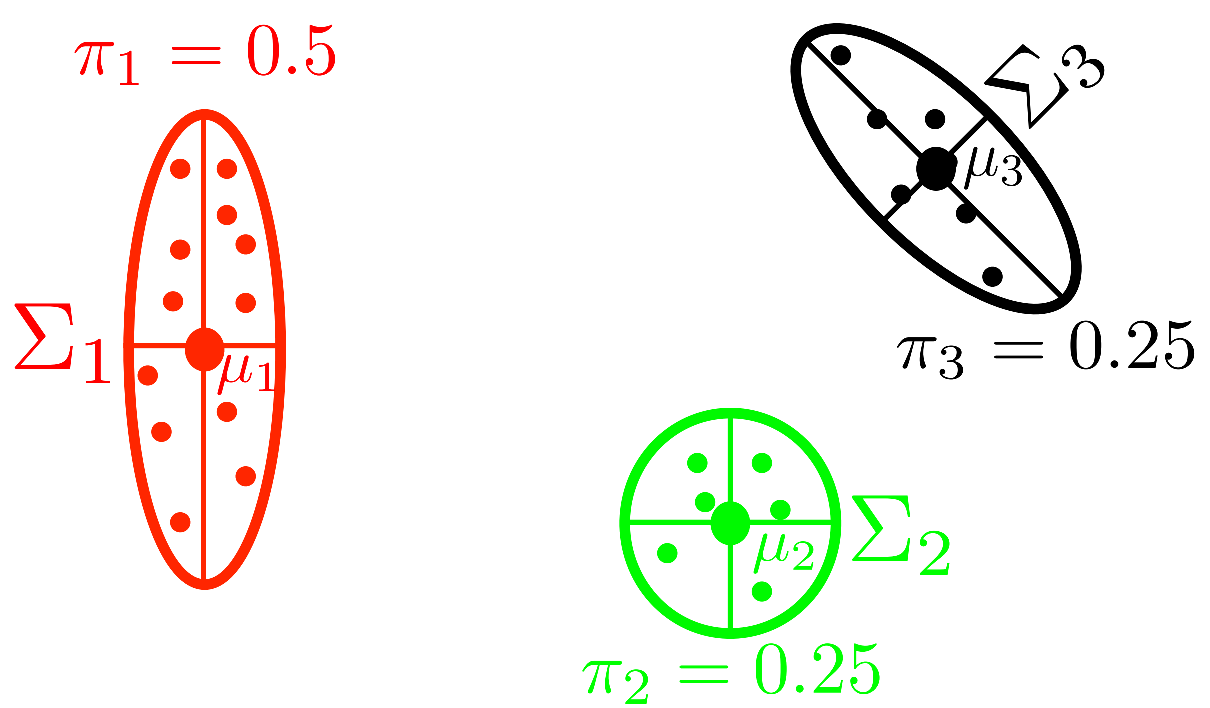
- Θ consists of set of possible parameters
- We have a distribution P_θ over the data induced by each $\theta \in \Theta$
- Data is generated by one of the $\theta \in \Theta$
- Learning: Estimate value or distribution for $\theta^* \in \Theta$ given data

Gaussian Mixture Models

Each $\theta \in \Theta$ is a model.

- Gaussian Mixture Model
  - Each θ consists of mixture distribution $\pi = (\pi_1, \ldots, \pi_K)$, means $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$ and covariance matrices $\Sigma_1, \ldots, \Sigma_K$
  - For each t, independently:

$$c_t \sim \pi, \qquad x_t \sim N(\mu_{c_t}, \Sigma_{c_t})$$



Expectation Maximization Algorithm

- For demonstration we shall consider the problem of finding MLE (MAP version is very similar)
- Initialize $\theta^{(0)}$ arbitrarily, repeat unit convergence:

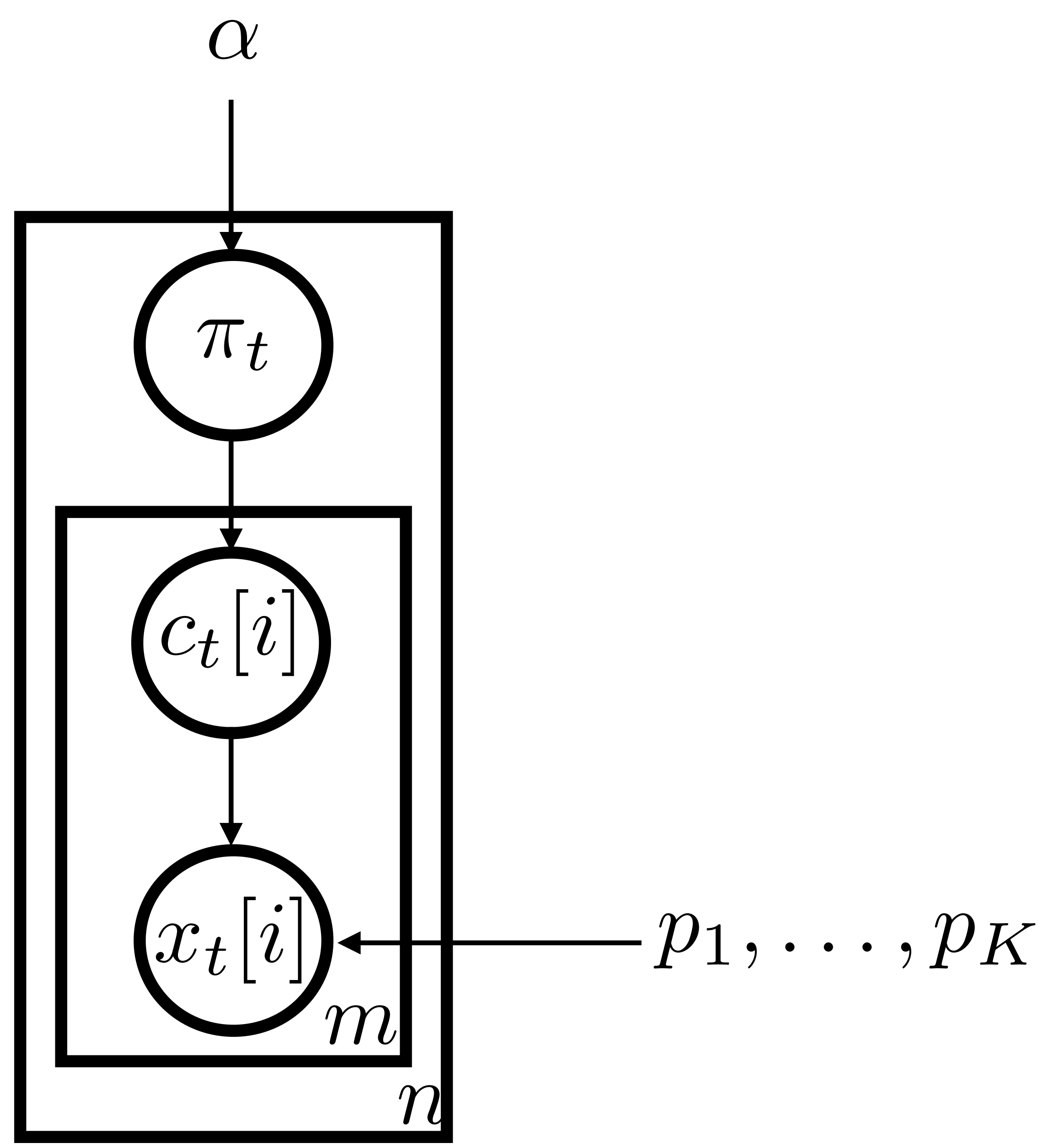
(E step) For every t, define distribution Q_t over the latent variable c_t as:

$$Q_t^{(i)}(c_t) = P(c_t | x_t, \theta^{(i-1)})$$

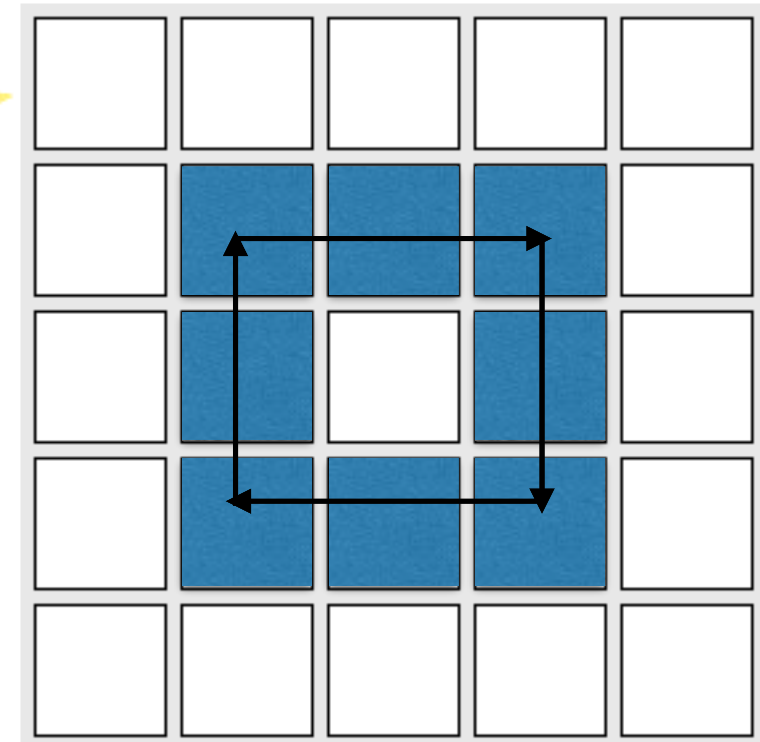
(M step)

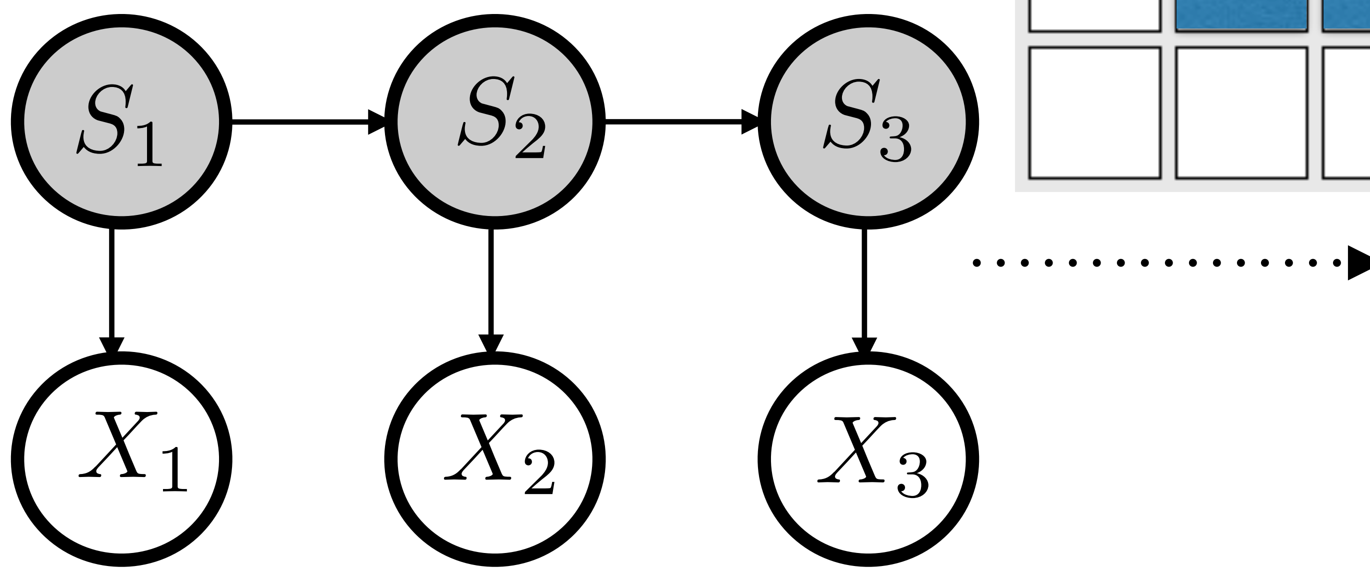
$$\theta^{(i)} = \mathrm{argmax}_{\theta \in \Theta} \sum_{t=1}^{n} \sum_{c_t} Q_t^{(i)}(c_t) \log P(x_t, c_t | \theta)$$

Latent Dirichlet Allocation

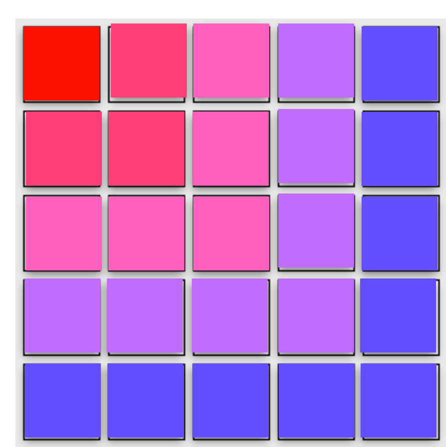


Example: Hidden Markov Model





What you hear:



Bayesian Networks

- Directed acyclic graph (DAG): $G = (V, E)$
- Joint distribution P_θ over $X_1, \ldots, X_n$ that factorizes over G:

$$P_\theta(X_1, \ldots, X_n) = \prod_{i=1}^{N} P_\theta(X_i | \text{Parent}(X_i))$$

- Hence Bayesian Networks are specified by G along with CPD's over the variables (given their parents)

Graphical Models

. Variable Elimination

. Message Passing

. Approximate Inference (via sampling)

. Parameter Estimation/learning using EM

Lessons Learnt

Relationships are key

- Between features (columns): Dimensionality reduction
- Between data points: clustering
- Between nodes in a graph: spectral clustering
- Between subsequent observations in a sequences: HMM
- Between variables in our model: Graphical models

No Free Lunch

- No model is universally good or better (remember the assignments)
- To make good models we need to make good assumptions
- Examples:
  - Probabilistic model generating the data
  - On relationship between various variables
  - Use the right latent variables to induce knowledge about the world

Bigger Picture

- Dimensionality reduction, clustering and more generally learning

There are no free lunches :(

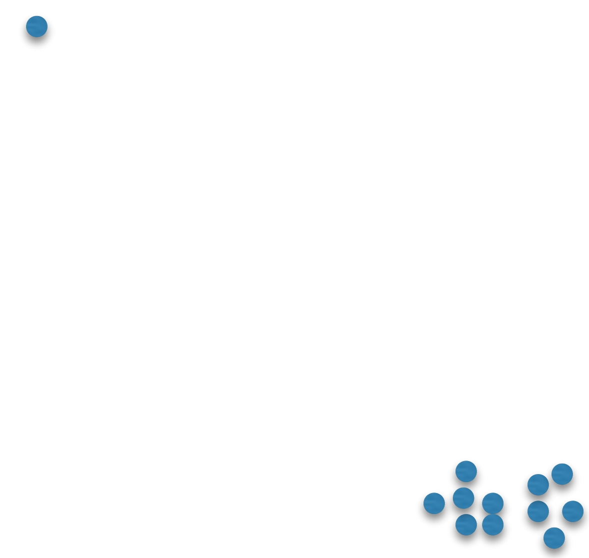
- Probabilistic modeling makes assumptions or guesses about way data is generated or how variables are related
- Caution:
  - In the real world no modeling assumption is really true ... there are good fits and bad fits
  - Choosing a model: Bias Vs Variance, Approximation error Vs estimation error, Expressiveness Vs amount of data
  - Choose the right model for the right job, there are no universally good answers
  - Feature extraction is an art (not covered in class)

Watch Out!

Good Feature Space is Key

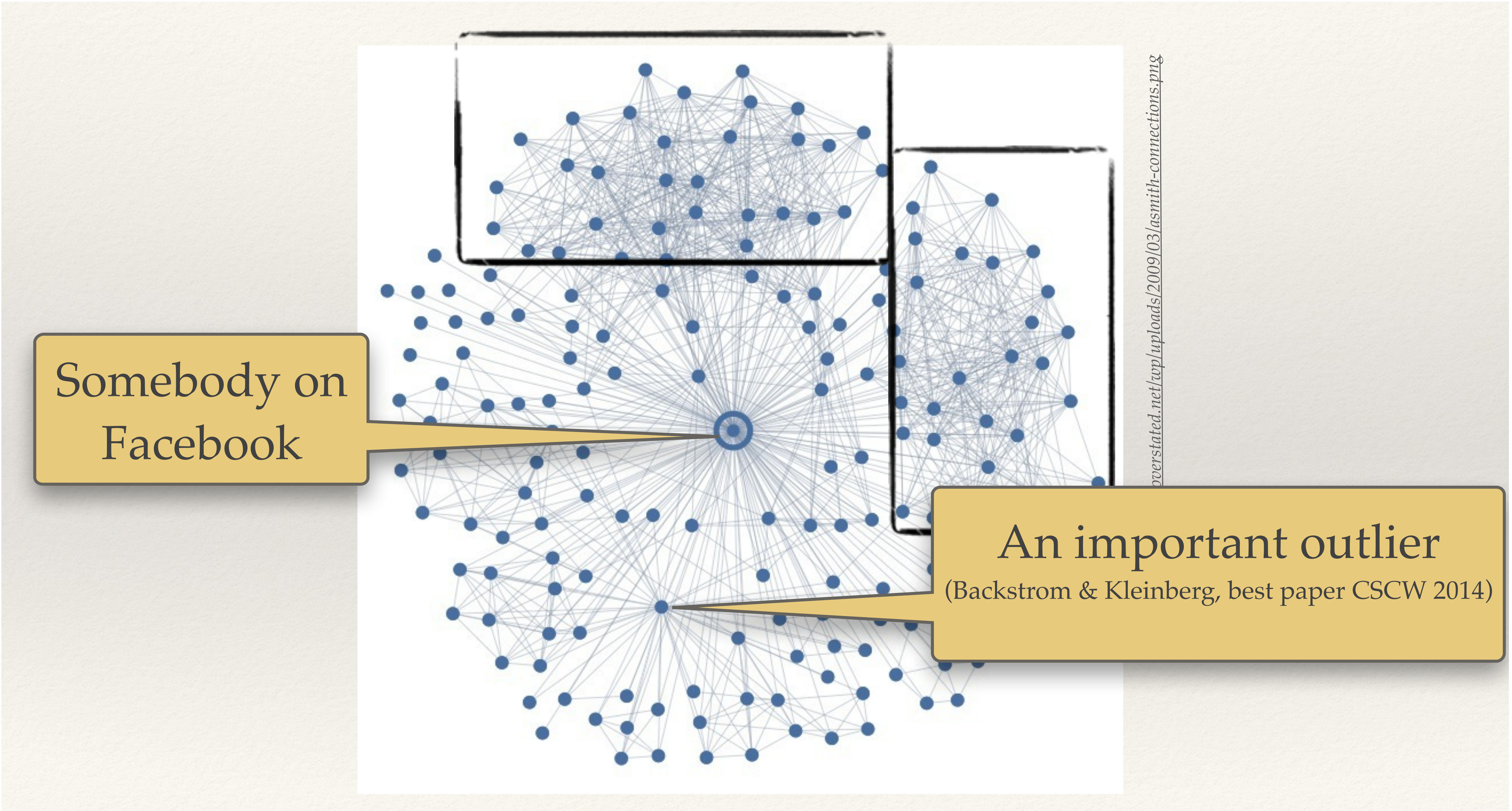
- If you don’t start with good feature space, you cant get good results
- Understand your problem, talk to practitioners and domain experts
  - Engineer features based on understanding problem Eg. Bag of words Vs N-grams
  - Engineer model based on understanding problem Eg. Convolutional networks

Watchout for Outliers



Methods can be thrown off by Outliers
Might have to remove outliers

WATCHOUT FOR OUTLIERS



Outliers can also have useful information!

Watch out for social implications . . .

When deployed in real world,

- Algorithms have to be fair, not worsen social inequities
- Be transparent and promote accountability
- Optimize not just performance but also social welfare

Its a big world out there!

Supervised Learning

- Training data: $(x_1, y_1), \ldots, (x_n, y_n)$ provided (typically assumed to be drawn from a fixed unknown distribution)
- Goal: Find a mapping $\hat{h}$ from input instances to outcome that minimizes $\mathbb{E}\left[\ell(\hat{h}(x), y)\right]$
  (ℓ is a loss function that measures error in prediction)

Generative Vs Discriminative Approaches

Generative approach:

- Input instances x_t's are generated based on/by y_t's
- We try to model $P(y, x) = P(x|y)P(y)$
- Example: Naive Bayes

Discriminative approach:

- We model $P(Y|X)$ or the boundary of classification
- Rationale: we are only concerned with predicting output y's given input x
- Example: linear regression, logistic regression

Probabilistic Story Vs Optimization Story

- Maximizing likelihood is same as minimizing negative log likelihood.
- Think of - log likelihood as loss function

$$-\log(P_\theta(Y|X)) \rightarrow \text{loss}(h_\theta(X), Y)$$

ie. θ parameterizes hypothesis for prediction or boundary

- MLE = Find hypothesis minimizing empirical loss on data
- Log Prior can be viewed as "regularization" of hypothesis

$$-\log(P(Y|X,\theta)) - \log(P(\theta)) \rightarrow \text{loss}(h_\theta(X), Y) + R(\theta)$$

- MAP = Find hypothesis minimizing empirical loss + regularization term
- Not all losses can be viewed as negative log likelihood

Semi-supervised Learning

- Can we use unlabeled examples to learn better?
- For instance, if we had a generative graphical model for the data: do example
- If we had prior information about the marginal distribution of X's and its relation to $P(Y|X)$

Active Learning

- Humans label the examples, can we get the learning algorithm in the loop?
- Learning algorithm picks the examples it wants labeled
- Eg. Margin based active learning, query points where model that fits observed data well so far disagree most

Domain Adaptation

- We learn a particular task on one corpus but want to use this learnt model to adapt with much fewer examples on another corpus
- Typical assumption: $P(Y|X)$ in both corpus remain fixed
- Marginal distributions change across the corpuses

Other Learning Frameworks

1. Transfer learning, multitask learning
2. Collaborative Filtering
3. Structured prediction
4. Online learning
5. ...

Thanks!

- For all your patience
- For all the feedback via surveys
- For helping each other on Piazza