10-301/601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

PCA + K-Means

Matt Gormley & Henry Chai

Lecture 24

Nov. 17, 2021

Q&A

Q: I've had such a great experience with this class, especially with your excellent TAs: how can I be more like them and contribute to future iterations of this class?

A: You can apply to be TA for this course next semester (S22)!

https://www.ml.cmu.edu/academics/ta.html

Q&A

Q: I've had such a great experience with this class, especially with the ML content: what courses should I take next given my specific interests in ML?

A: Well instead of asking us old fogeys (who have never taken any of these courses), why not see what some fellow students have to say on the subject, courtesy of your aforementioned excellent TAs:

https://piazza.com/class/ksg77m9s2cx3d6?cid=1780

(Latest pinned post on Piazza)

Q&A

Q: I've seen the term Markov boundary used before: is that related to a Markov blanket?

A: In a BayesNet, the Markov blanket for X is *any* set S s.t. X is conditionally independent of all other variables when conditioned on S.

The Markov boundary for X is the smallest possible Markov blanket, which happens to be the children, parents and co-parents of X (note this is the definition of a Markov blanket we presented)

Every Markov boundary is a Markov blanket but not vice versa.

Deep Q-learning

- Algorithm 4: Online learning of Q^* (parametric form)
  - Inputs: discount factor γ,

    an initial state s_0,

    learning rate α
  - Initialize parameters $\Theta^{(0)}$
  - For $t = 0, 1, 2,$ ...
    - Gather training sample (s_t, a_t, r_t, s_{t+1})
    - Update $\Theta^{(t)}$ by taking a step opposite the gradient

      $$\Theta^{(t+1)} \leftarrow \Theta^{(t)} - \underline{\alpha \nabla_{\Theta^{(t+1)}} \ell\left(\Theta^{(t)}, \Theta^{(t+1)}\right)}$$

      where

      $$\nabla_{\Theta^{(t+1)}} \ell\left(\Theta^{(t)}, \Theta^{(t+1)}\right) = 2\left(y - Q\left(s, a; \Theta^{(t+1)}\right)\right) \underbrace{\nabla_{\Theta^{(t+1)}} Q\left(s, a; \Theta^{(t+1)}\right)}_{}$$

Deep Q-learning: Experience Replay

- SGD assumes i.i.d. training samples but in RL, samples are *highly* correlated
- Idea: keep a “replay memory” $\mathcal{D} = \{e_1, e_2, \dots, e_N\}$ of N most recent experiences $\underline{e_t = (s_t, a_t, r_t, s_{t+1})}$ (Lin, 1992)
  - Also keeps the agent from “forgetting” about recent experiences
- Alternate between:
  1. Sampling some e_i uniformly at random from $\mathcal{D}$ and applying a Q-learning update (repeat T times)
  2. Adding a new experience to $\mathcal{D}$
- Can also sample experiences from $\mathcal{D}$ according to some distribution that prioritizes experiences with high error (Schaul et al., 2016)

RL Learning Goals: Q-Leaning and Deep RL

a. Apply Q-Learning to a real-world environment
b. Implement Q-learning
c. Identify the conditions under which the Q-learning algorithm will converge to the true value function
d. Adapt Q-learning to Deep Q-learning by employing a neural network approximation to the Q function
e. Describe the connection between Deep Q-Learning and regression

BIG PICTURE

ML Big Picture

Learning Paradigms:

What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Problem Formulation:

What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Application Areas

Key challenges?

NLP, Speech, Computer Vision, Robotics, Medicine, Search

Theoretical Foundations:

What principles guide learning?

- ❑ probabilistic
- ❑ information theoretic
- ❑ evolutionary search
- ❑ ML as optimization

Facets of Building ML Systems:

How to build systems that are robust, efficient, adaptive, effective?

1. *Data prep*
2. *Model selection*
3. *Training (optimization / search)*
4. *Hyperparameter tuning on validation data*
5. *(Blind) Assessment on test data*

Big Ideas in ML:

Which are the ideas driving development of the field?

- *inductive bias*
- *generalization / overfitting*
- *bias-variance decomposition*
- *generative vs. discriminative*
- *deep nets, graphical models*
- *PAC learning*
- *distant rewards*

Learning Paradigms

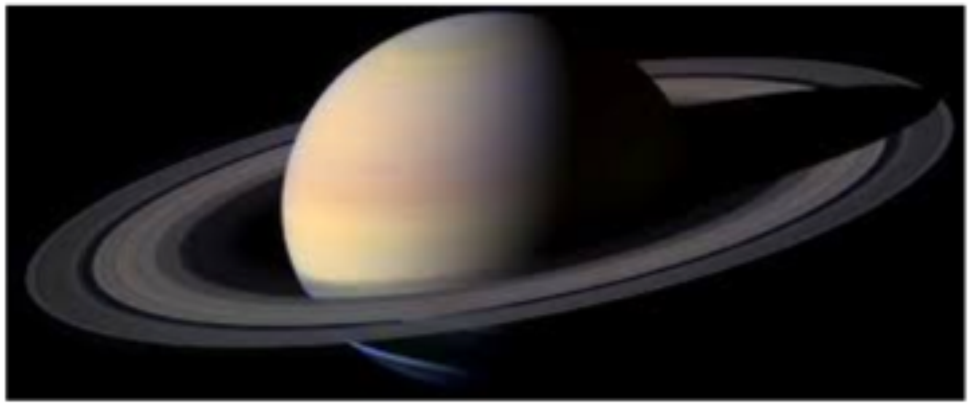
Paradigm	Data
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$ $\quad$ $\mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\boxed{\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}}$ $\quad$ $\mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \boxed{\{\mathbf{x}^{(i)}, y^{(i)}\}}_{i=1}^{N_1} \cup \boxed{\{\mathbf{x}^{(j)}\}}_{j=1}^{N_2}$ $\leftarrow$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\boxed{\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}}$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots\}$
Reinforcement Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots\}$

DIMENSIONALITY REDUCTION

High Dimension Data

Examples of high dimensional data:

- High resolution images (millions of pixels)



High Dimension Data

Examples of high dimensional data:

– Multilingual News Stories (vocabulary of hundreds of thousands of words)

Hilo Tribune.

KILAUEA CRATER CONTINUES IN ERUPTION

FOUR DAYS OF ACTIVITY

ANOTHER OUTBREAK OCCURS

HONOLULU GETS APPROPRIATIONS FOR HARBOR AND LIGHTHOUSE

River and Harbor Bill Passes.

Postmaster Pratt Confirmed.

Latest Sugar Quotations.

Fire in New Orleans.

叻報

Македония

PRAVA BRAĆA

Истински братя

DEKLARACIJA

Декларация

குடி அரசு

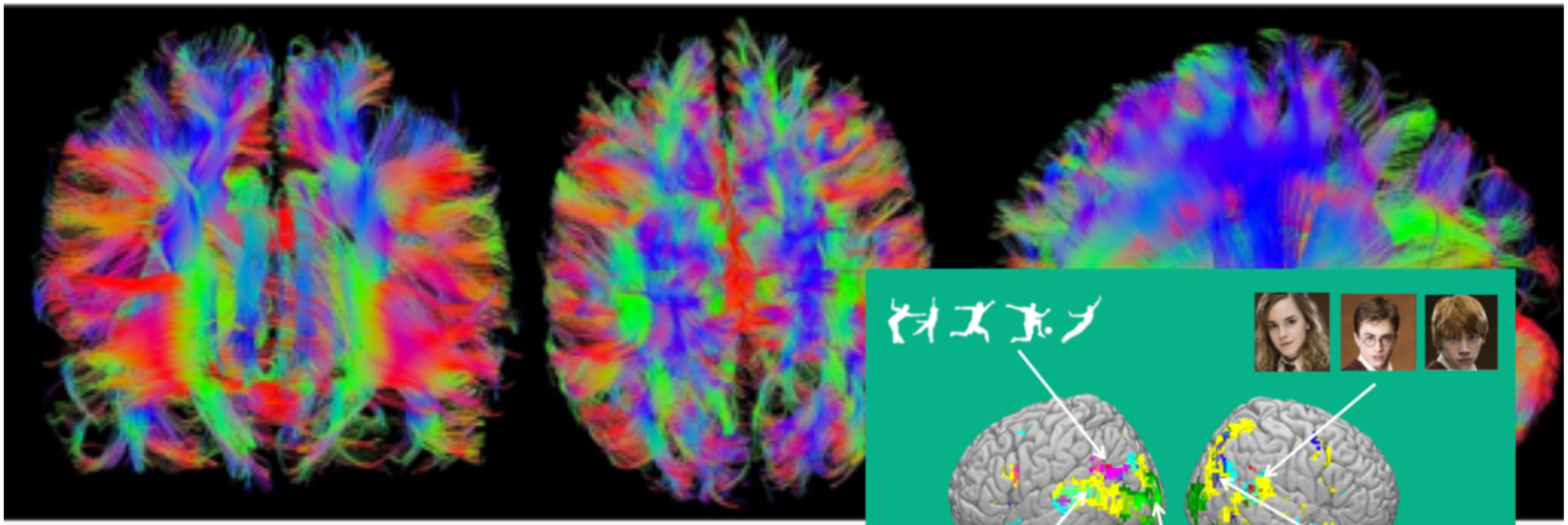
"KUDI ARASU"

வீழ்க இந்தி!

High Dimension Data

Examples of high dimensional data:

– Brain Imaging Data (100s of MBs per scan)



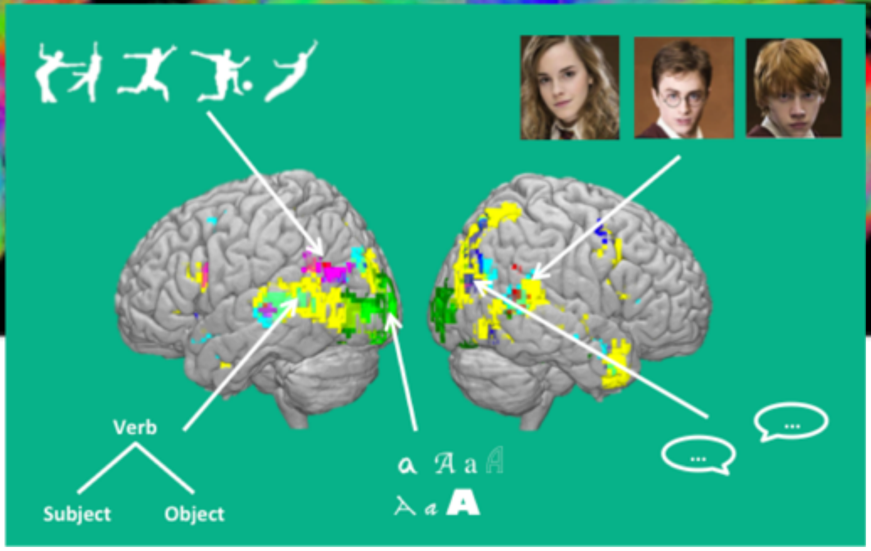


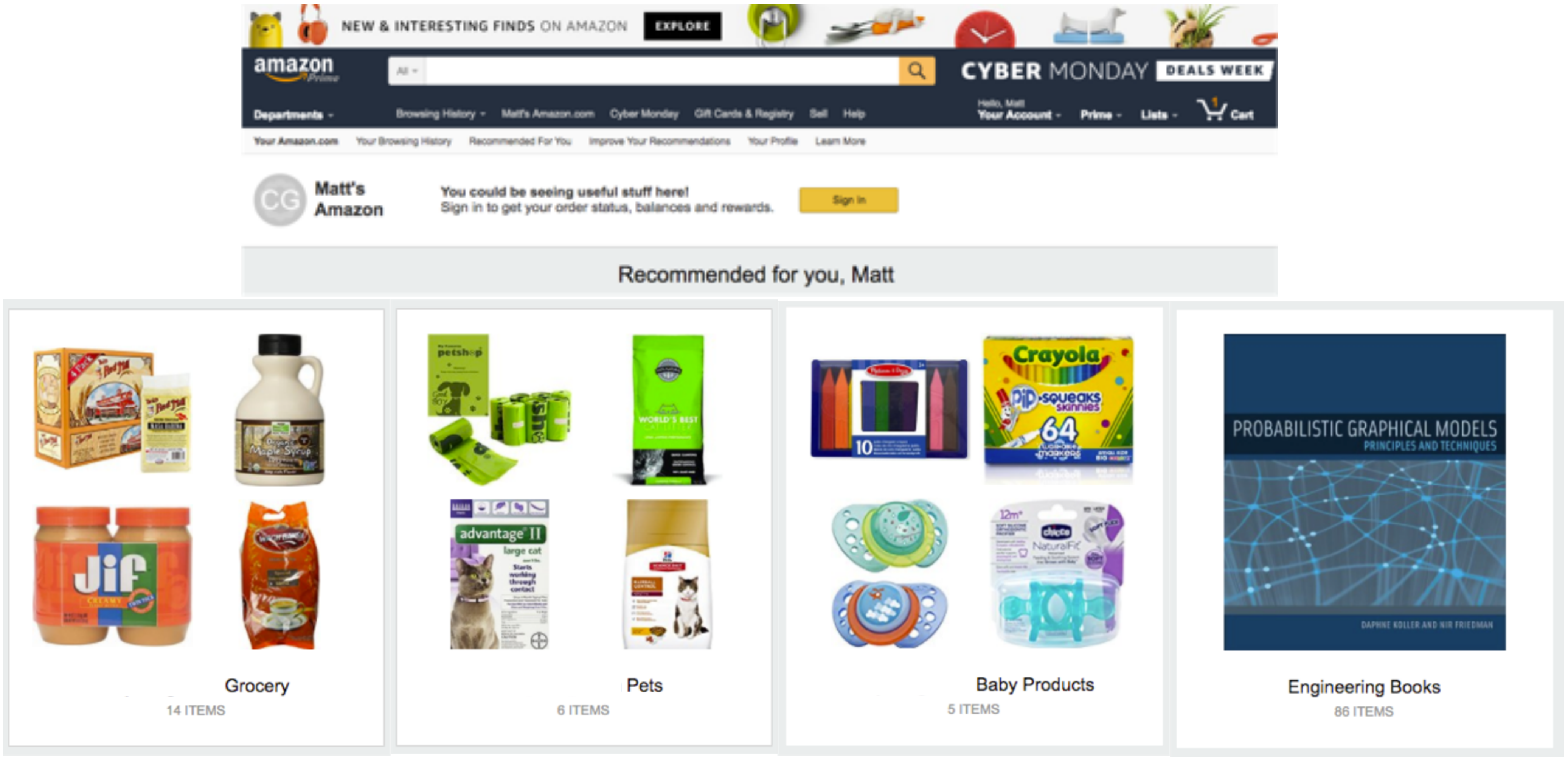
Image from (Wehbe et al., 2014)

Image from https://pixabay.com/en/brain-mrt-magnetic-resonance-imaging-1728449/

High Dimension Data

Examples of high dimensional data:

– Customer Purchase Data



Learning Representations

Dimensionality Reduction Algorithms:
Powerful (often unsupervised) learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.

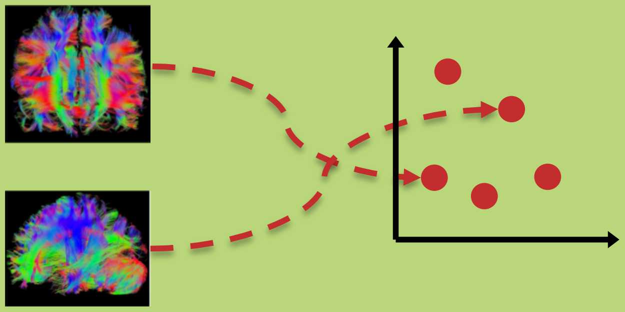
Examples:
PCA, Kernel PCA, ICA, CCA, Autoencoders, Matrix Factorization

Useful for:
- Visualization
- More efficient use of resources (e.g., time, memory, communication)
- Statistical: fewer dimensions → better generalization
- Noise removal (improving data quality)

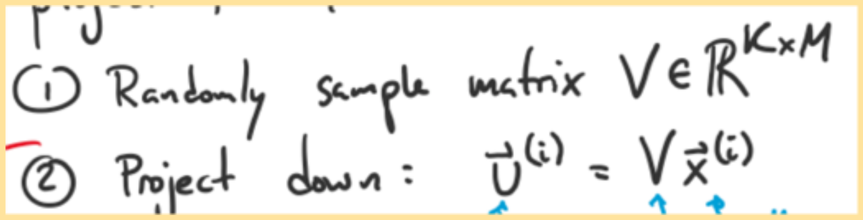
Slide adapted from Nina Balcan

This section in one slide…

1. Dimensionality reduction:



2. Random Projection:



① Randomly sample matrix $V \in \mathbb{R}^{K \times M}$

② Project down: $\vec{u}^{(i)} = V\vec{x}^{(i)}$

3. Definition of PCA:

Choose the matrix V that either…

1. minimizes reconstruction error
2. consists of the K eigenvectors with largest eigenvalue

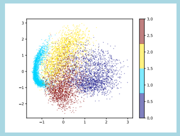
The above are equivalent definitions.

4. Algorithm for PCA:

The option we'll focus on:

Run Singular Value Decomposition (SVD) to obtain all the eigenvectors. Keep just the top-K to form V. Play some tricks to keep things efficient.

5. An Example



DIMENSIONALITY REDUCTION BY RANDOM PROJECTION

Random Projection

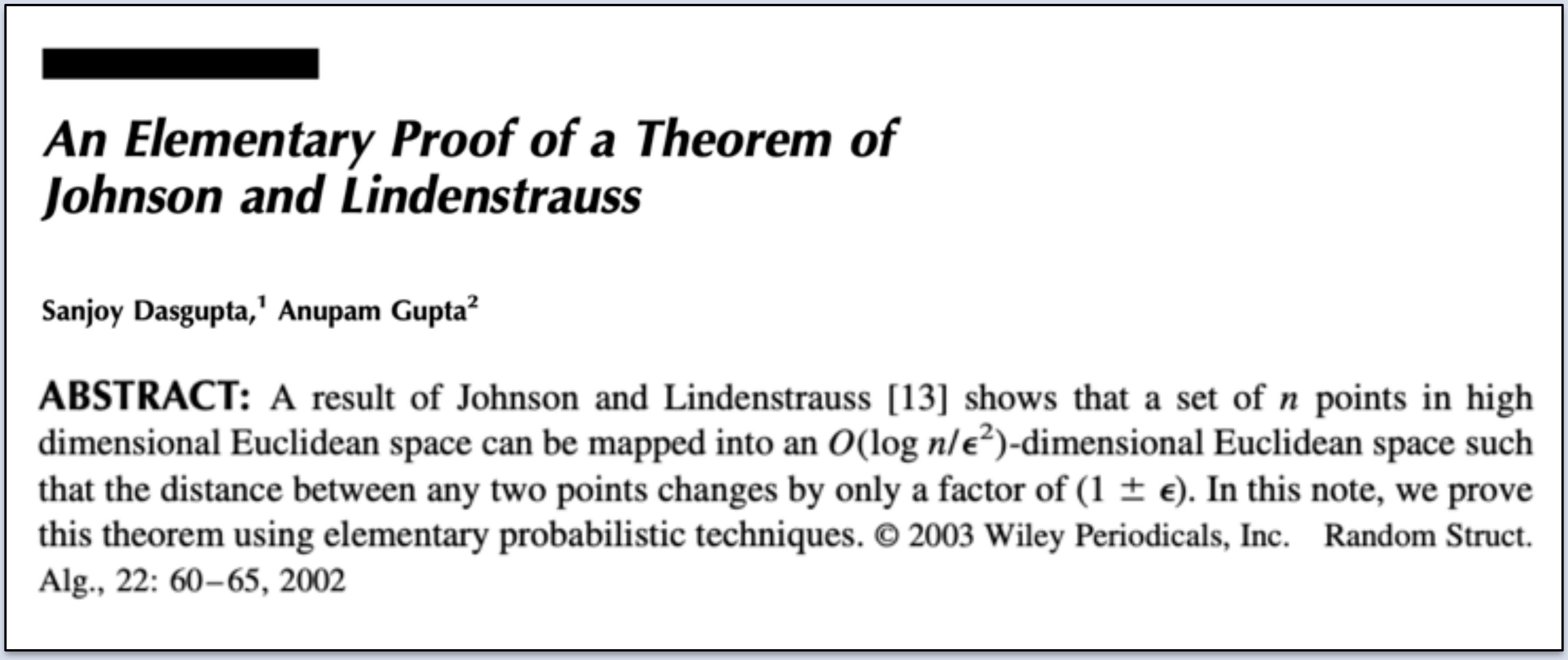
Whiteboard

- Random linear projection

Johnson-Lindenstrauss Lemma

Q: But how could we ever hope to preserve any useful information by randomly projecting into a low-dimensional space?

A: Even random projection enjoys some surprisingly impressive properties. In fact, a standard of the J-L lemma starts by assuming we have a random linear projection obtained by sampling each matrix entry from a Gaussian(0,1).



An Elementary Proof of a Theorem of Johnson and Lindenstrauss

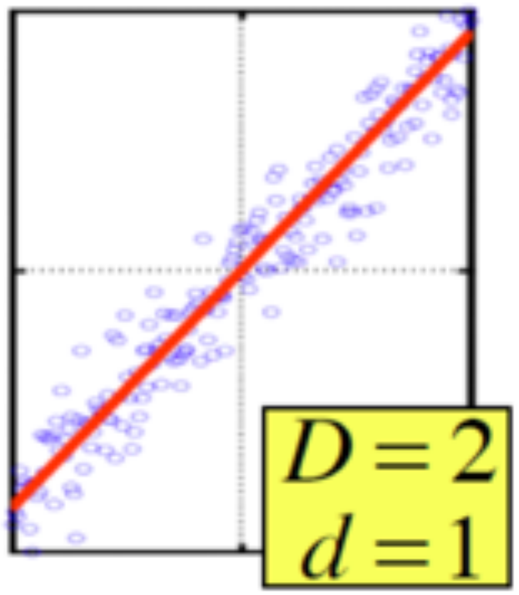
Sanjoy Dasgupta,[1] Anupam Gupta[2]

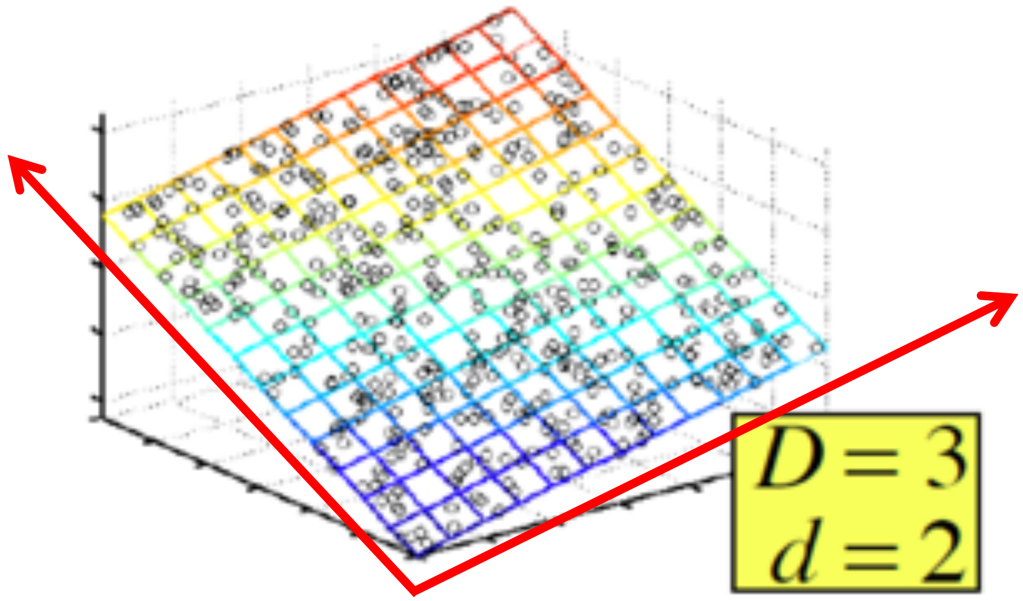
ABSTRACT: A result of Johnson and Lindenstrauss [13] shows that a set of n points in high dimensional Euclidean space can be mapped into an $O(\log n/\epsilon^2)$-dimensional Euclidean space such that the distance between any two points changes by only a factor of $(1 \pm \epsilon)$. In this note, we prove this theorem using elementary probabilistic techniques. © 2003 Wiley Periodicals, Inc. Random Struct. Alg., 22: 60–65, 2002

http://www.cs.cmu.edu/~anupamg/papers/jl.pdf

DEFINITION OF PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal Component Analysis (PCA)



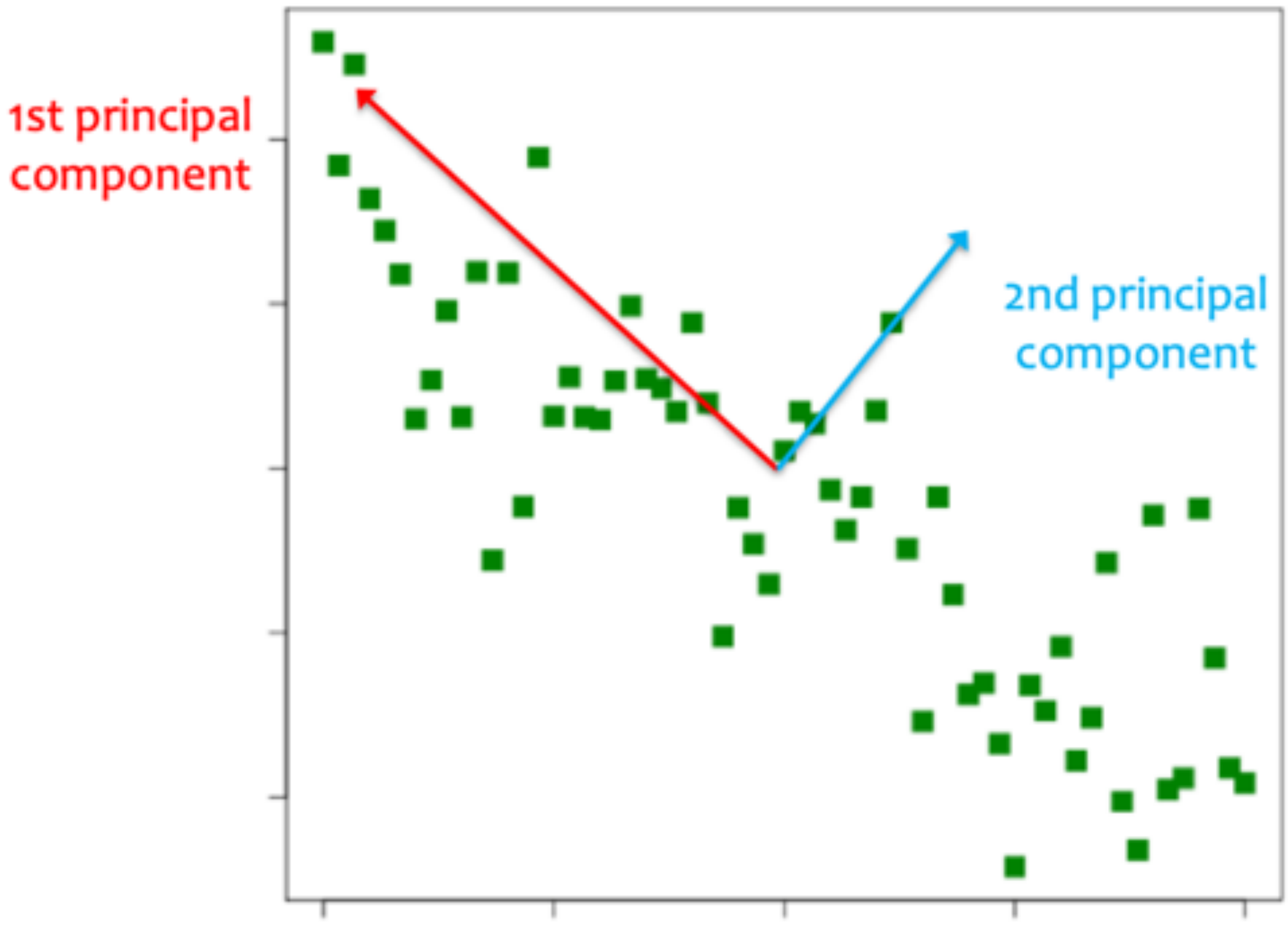


In case where data lies on or near a low d-dimensional linear subspace, axes of this subspace are an effective representation of the data.

Identifying the axes is known as Principal Components Analysis, and can be obtained by using classic matrix computation tools (Eigen or Singular Value Decomposition).

Slide from Nina Balcan

PCA Example: 2D Gaussian Data



https://commons.wikimedia.org/wiki/File:Scatter_diagram_for_quality_characteristic_XXX.svg

Data for PCA

$$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$$

$$\mathbf{x}^{(i)} \in \mathbb{R}^M$$

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ (\mathbf{x}^{(2)})^T \\ \vdots \\ (\mathbf{x}^{(N)})^T \end{bmatrix}$$

We assume the data is **centered**

$$\mu = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)} = \mathbf{0}$$

Q: What if your data is **not** centered?

A: Subtract off the sample mean

$$\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} - \mu, \;\; \forall i$$

Sample Covariance Matrix

The sample covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{\underline{M \times M}}$ is given by:

$$\Sigma_{jk} = \frac{1}{N} \sum_{i=1}^{N} (\underline{x_j^{(i)}} - \underline{\mu_j})(\underline{x_k^{(i)}} - \underline{\mu_k})$$

Since the data matrix is centered, we rewrite as:

$$\boxed{\boldsymbol{\Sigma} = \frac{1}{N} \mathbf{X}^T \mathbf{X}}$$

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ (\mathbf{x}^{(2)})^T \\ \vdots \\ (\mathbf{x}^{(N)})^T \end{bmatrix}$$

Principal Component Analysis (PCA)

Linear Projection:
Given KxM matrix **V**, and Mx1 vector **x**$^{(i)}$ we obtain the Kx1 projection **u**$^{(i)}$ by:

$$\mathbf{u}^{(i)} = \mathbf{V}^{\cancel{T}}\mathbf{x}^{(i)}$$

$$\underset{K\times M}{V} = \begin{bmatrix} — & \vec{v}_1^T & — \\ — & \vec{v}_2^T & — \\ & \vdots & \\ — & \vec{v}_k^T & — \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} = \begin{bmatrix} v_1^T x \\ v_2^T x \\ \vdots \\ v_k^T x \end{bmatrix} \begin{matrix} \in \mathbb{R} \\ \in \mathbb{R} \\ \\ \in \mathbb{R} \end{matrix}$$

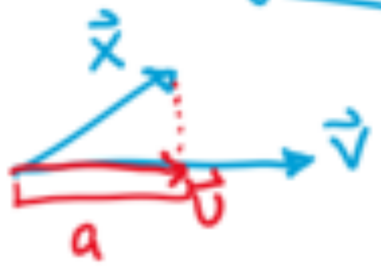
Definition of PCA:
PCA repeatedly chooses a next vector **v**$_j$ that minimizes the reconstruction error s.t. **v**$_j$ is orthogonal to **v**$_1$, **v**$_2$,…, **v**$_{j-1}$.

Vector **v**$_j$ is called the **jth principal component.**

Notice: Two vectors a and b are **orthogonal** if aTb = 0.
➔the K-dimensions in PCA are uncorrelated

Vector Projection

Recall: Projection



length of projection of $\vec{x}$ onto $\vec{v}$

$a = \vec{v}^T\vec{x}$ if $\|v\|_2 = 1$

$a = \frac{\vec{v}^T\vec{x}}{\|v\|_2}$ otherwise

projection of $\vec{x}$ onto $\vec{v}$

$\vec{u} = a\vec{v} = (\vec{v}^T\vec{x})\vec{v}$ if $\|v\|_2 = 1$

$\vec{u} = a\vec{v} = \frac{(\vec{v}^T\vec{x})\vec{v}}{\|\vec{v}\|_2^2}$ otherwise

Principal Component Analysis (PCA)

Whiteboard

- Objective functions for PCA

Maximizing the Variance

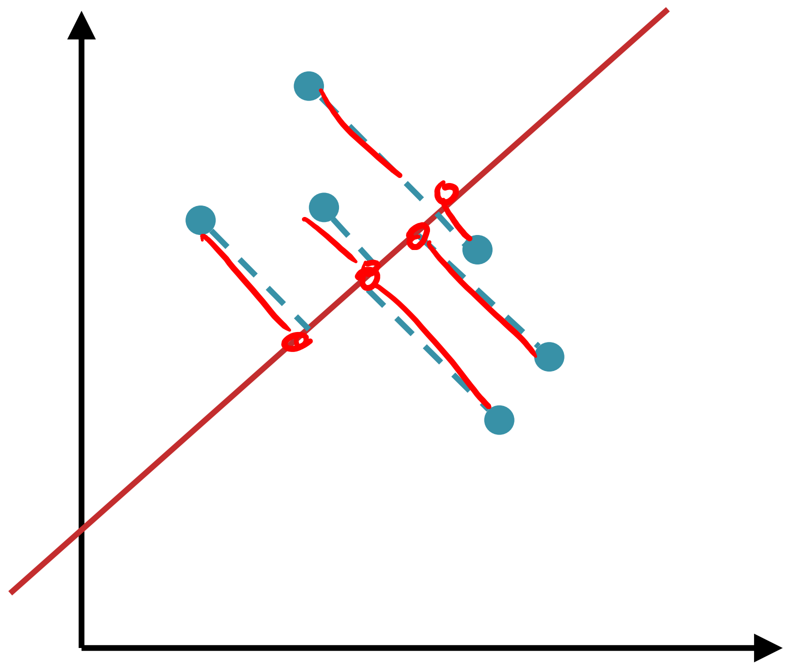
Quiz: Consider the two projections below

Q1 1. Which maximizes the variance? A : 35% B : 65%

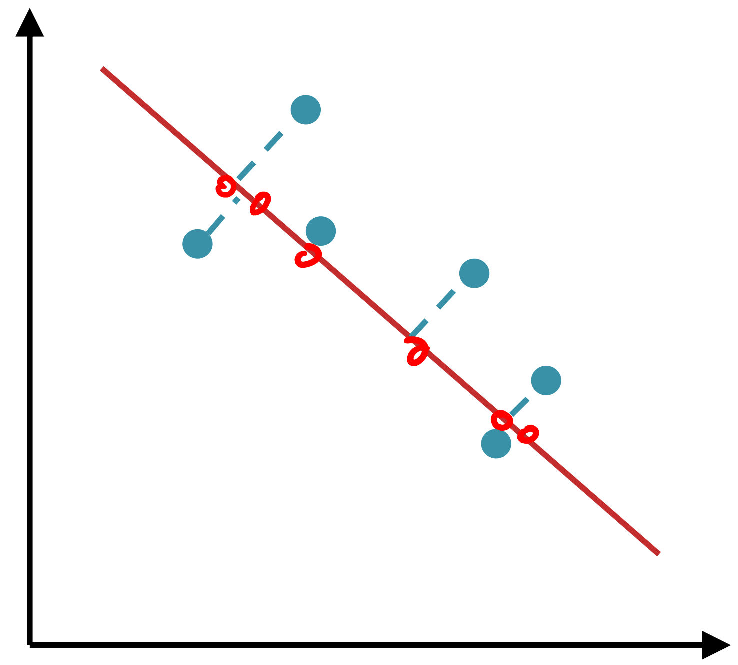
Q2 2. Which minimizes the reconstruction error? A : 10 B : 90%

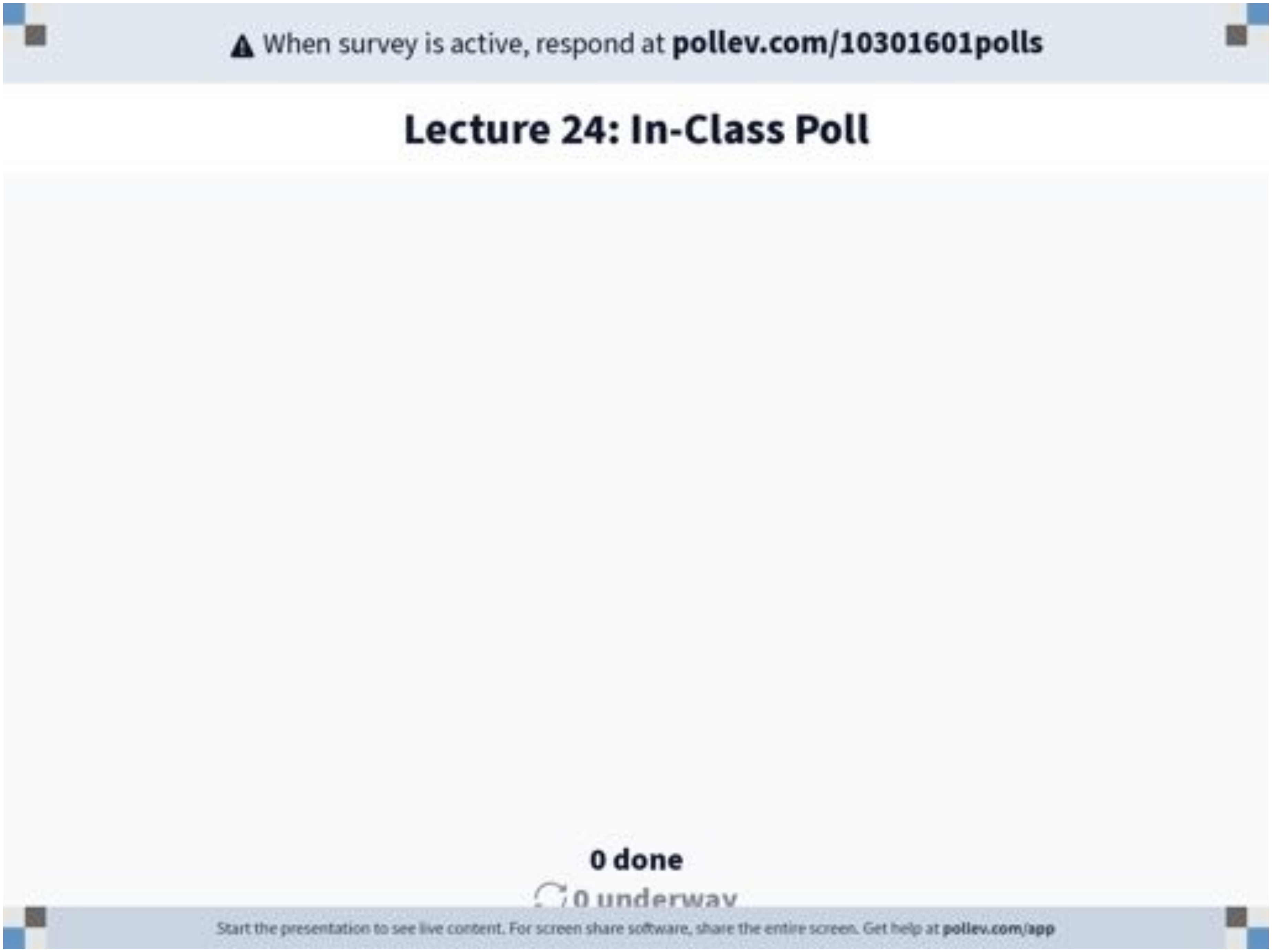
C = toxic

Option A



Option B





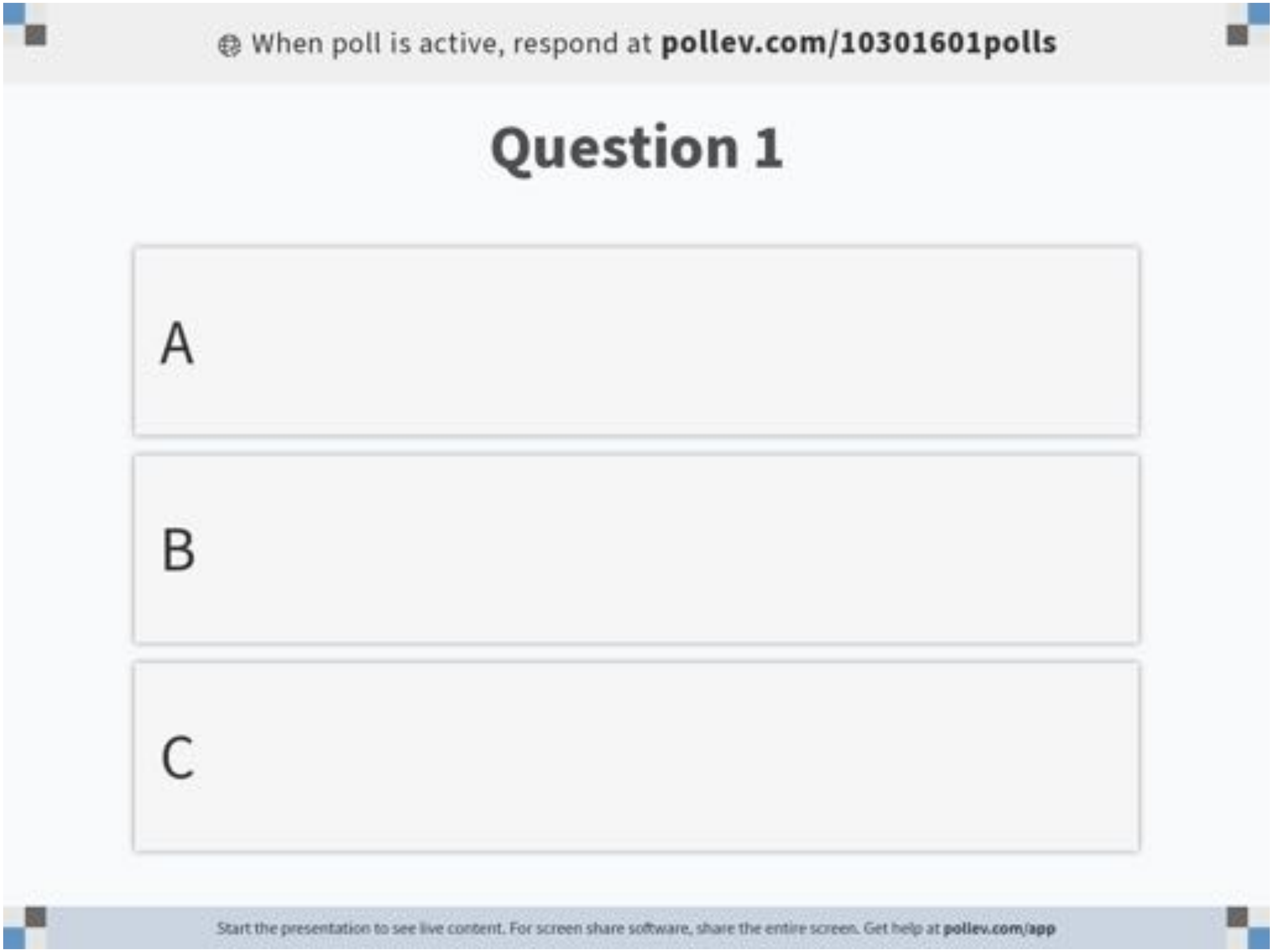
⚠ When survey is active, respond at **pollev.com/10301601polls**

Lecture 24: In-Class Poll

0 done

0 underway

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**



When poll is active, respond at **pollev.com/10301601polls**

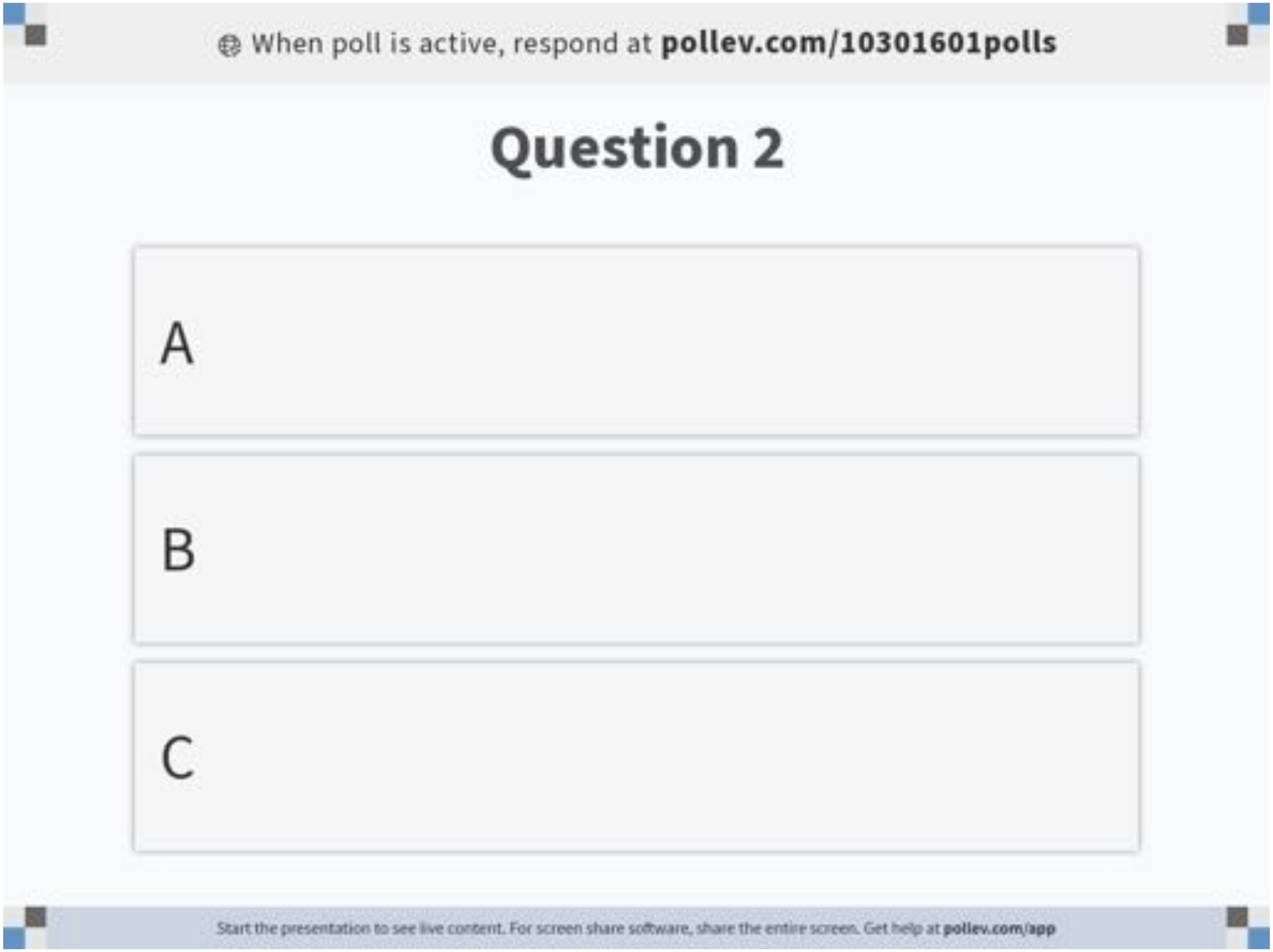
Question 1

A

B

C

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**



When poll is active, respond at **pollev.com/10301601polls**

Question 2

A

B

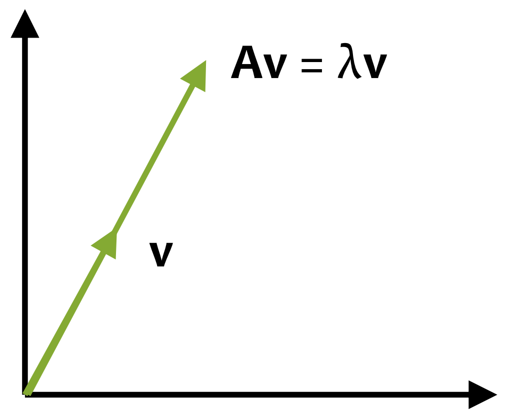
C

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Background: Eigenvectors & Eigenvalues

For a square matrix **A** (n x n matrix), the vector **v** (n x 1 matrix) is an **eigenvector** iff there exists **eigenvalue** λ (scalar) such that:

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$



The linear transformation **A** is only stretching vector **v**.

That is, λ**v** is a *scalar multiple* of **v.**

Principal Component Analysis (PCA)

Whiteboard

– PCA, Eigenvectors, and Eigenvalues

Equivalence of Maximizing Variance and Minimizing Reconstruction Error

PCA

Claim: Minimizing the reconstruction error is equivalent to maximizing the variance.

Proof: First, note that:

$$\|\mathbf{x}^{(i)} - (\mathbf{v}^T\mathbf{x}^{(i)})\mathbf{v}\|^2 = \underline{\|\mathbf{x}^{(i)}\|^2} - \underline{(\mathbf{v}^T\mathbf{x}^{(i)})^2} \qquad (1)$$

since $\boxed{\mathbf{v}^T\mathbf{v} = \|\mathbf{v}\|^2 = 1.}$

Substituting into the minimization problem, and removing the extraneous terms, we obtain the maximization problem.

$$\mathbf{v}^* = \operatorname*{argmin}_{\mathbf{v}:\|\mathbf{v}\|^2=1} \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}^{(i)} - (\mathbf{v}^T\mathbf{x}^{(i)})\mathbf{v}\|^2 \qquad (2)$$

$$= \operatorname*{argmin}_{\mathbf{v}:\|\mathbf{v}\|^2=1} \frac{1}{N}\sum_{i=1}^{N} \boxed{\|\mathbf{x}^{(i)}\|^2} - (\mathbf{v}^T\mathbf{x}^{(i)})^2 \qquad (3)$$

$$= \operatorname*{argmax}_{\mathbf{v}:\|\mathbf{v}\|^2=1} \frac{1}{N}\sum_{i=1}^{N} (\mathbf{v}^T\mathbf{x}^{(i)})^2 \qquad (4)$$

The First Principal Component

PCA

Claim: The vector that maximizes the variances is the eigenvector of $\mathbf{\Sigma}$ with largest eigenvalue.

Proof Sketch: To find the first principal component, we wish to solve the following constrained optimization problem (variance minimization).

$$\mathbf{v}_1 = \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\operatorname{argmax}} \underline{\mathbf{v}^T\mathbf{\Sigma}\mathbf{v}} \tag{1}$$

So we turn to the method of Lagrange multipliers. The Lagrangian is:

$$\mathcal{L}(\mathbf{v}, \lambda) = \mathbf{v}^T\mathbf{\Sigma}\mathbf{v} - \lambda(\mathbf{v}^T\mathbf{v} - 1) \tag{2}$$

Taking the derivative of the Lagrangian and setting to zero gives:

$$\frac{d}{d\mathbf{v}}\left(\mathbf{v}^T\mathbf{\Sigma}\mathbf{v} - \lambda(\mathbf{v}^T\mathbf{v} - 1)\right) = 0 \tag{3}$$

$$\mathbf{\Sigma}\mathbf{v} - \lambda\mathbf{v} = 0 \tag{4}$$

$$\underline{\mathbf{\Sigma}\mathbf{v} = \lambda\mathbf{v}} \tag{5}$$

Recall: For a square matrix $\mathbf{A}$, the vector $\mathbf{v}$ is an **eigenvector** iff there exists **eigenvalue** λ such that:

$$\boxed{\mathbf{A}\mathbf{v} = \lambda\mathbf{v}} \tag{6}$$

Rewriting the objective of the maximization shows that not only will the optimal vector $\mathbf{v}_1$ be an eigenvector, it will be one with maximal eigenvalue.

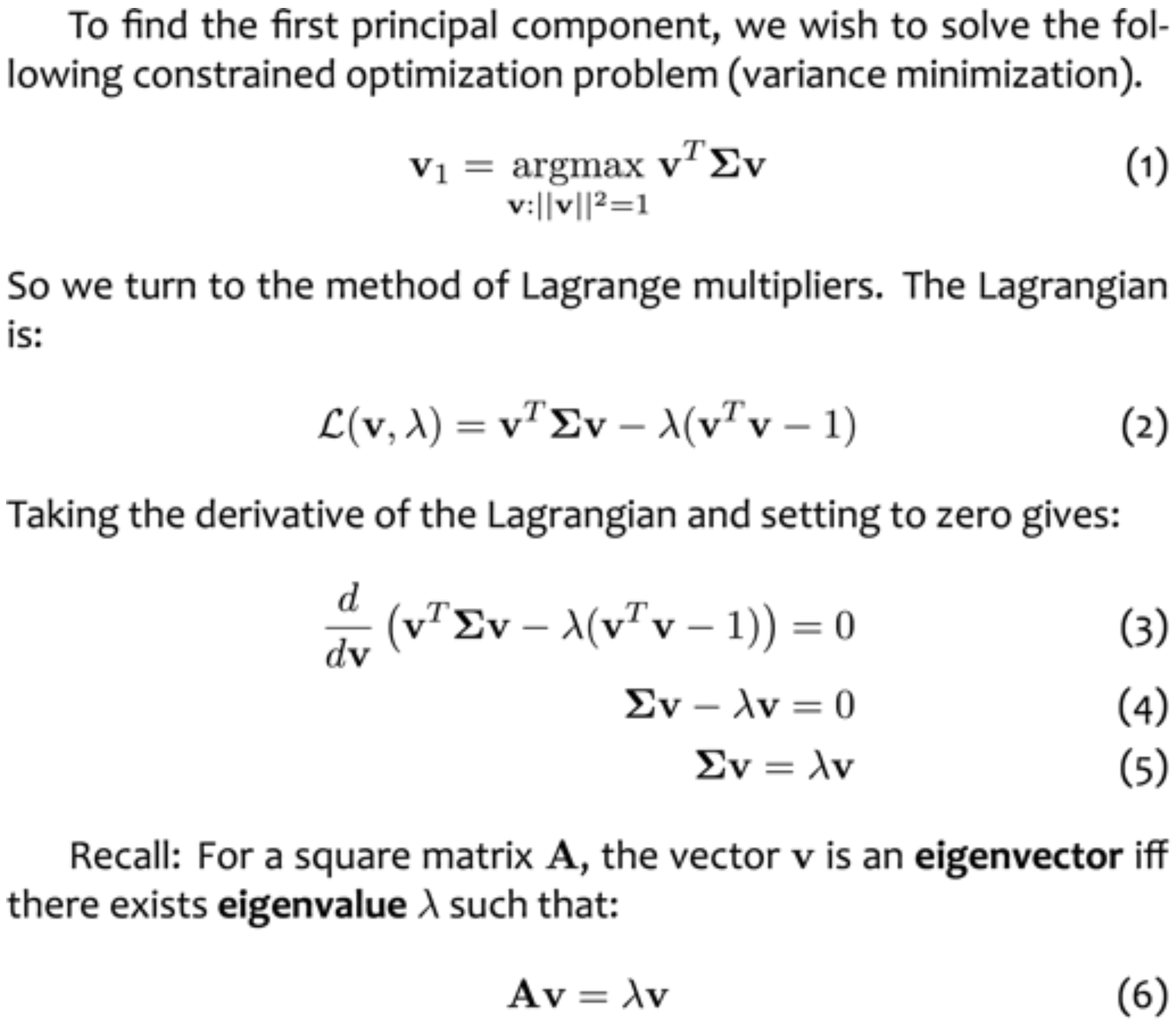
$$\underline{\mathbf{v}^T\mathbf{\Sigma}\mathbf{v}} = \underline{\mathbf{v}^T\lambda\mathbf{v}} \tag{7}$$

$$= \underline{\lambda\mathbf{v}^T\mathbf{v}} \tag{8}$$

$$= \underline{\lambda||\mathbf{v}||^2} \tag{9}$$

$$= \underline{\lambda} \tag{10}$$

PCA: the First Principal Component



To find the first principal component, we wish to solve the following constrained optimization problem (variance minimization).

$$\mathbf{v}_1 = \underset{\mathbf{v}:||\mathbf{v}||^2=1}{\operatorname{argmax}} \ \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v} \tag{1}$$

So we turn to the method of Lagrange multipliers. The Lagrangian is:

$$\mathcal{L}(\mathbf{v}, \lambda) = \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v} - \lambda(\mathbf{v}^T \mathbf{v} - 1) \tag{2}$$

Taking the derivative of the Lagrangian and setting to zero gives:

$$\frac{d}{d\mathbf{v}} \left(\mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v} - \lambda(\mathbf{v}^T \mathbf{v} - 1) \right) = 0 \tag{3}$$

$$\boldsymbol{\Sigma}\mathbf{v} - \lambda \mathbf{v} = 0 \tag{4}$$

$$\boldsymbol{\Sigma}\mathbf{v} = \lambda \mathbf{v} \tag{5}$$

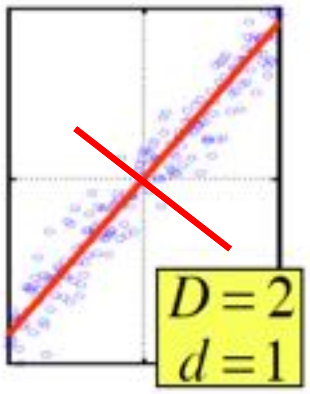
Recall: For a square matrix $\mathbf{A}$, the vector $\mathbf{v}$ is an **eigenvector** iff there exists **eigenvalue** λ such that:

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v} \tag{6}$$

Principal Component Analysis (PCA)

$(X X^T)v = \lambda v$, so v (the first PC) is the eigenvector of sample correlation/covariance matrix $X X^T$

Sample variance of projection $v^T X X^T v = \lambda v^T v = \lambda$



Thus, the eigenvalue λ denotes the amount of variability captured along that dimension (aka amount of energy along that dimension).

Eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$

Σ :

- The 1st PC v_1 is the the eigenvector of the sample covariance matrix $X X^T$ associated with the largest eigenvalue
- The 2nd PC v_2 is the the eigenvector of the sample covariance matrix $X X^T$ associated with the second largest eigenvalue
- And so on …

Slide from Nina Balcan

ALGORITHMS FOR PCA

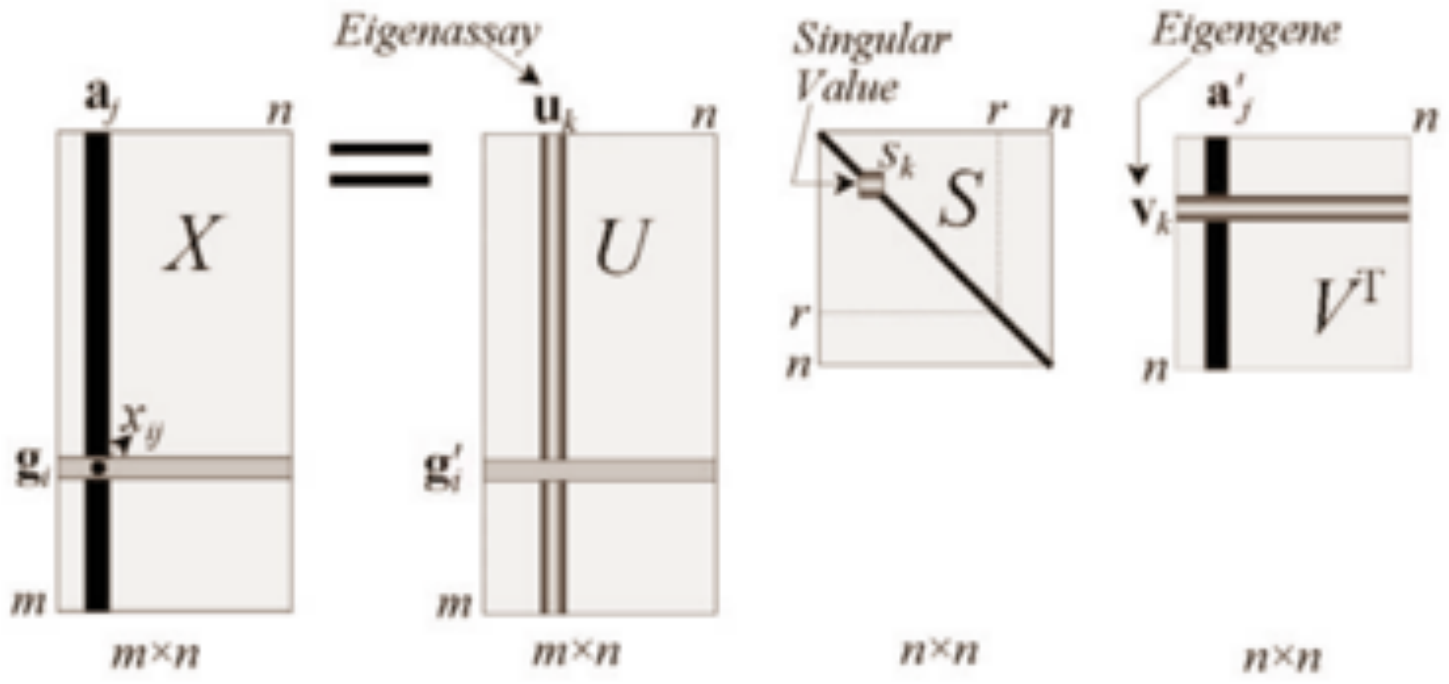
Algorithms for PCA

How do we find principal components (i.e. eigenvectors)?

- Power iteration (aka. Von Mises iteration)
  - finds **each** principal component **one at a time** in order
- Singular Value Decomposition (SVD)
  - finds **all** the principal components **at once**
  - two options:
    - Option A: run SVD on $X^T X$
    - Option B: run SVD on X
      (not obvious why Option B should work…)
- Stochastic Methods (approximate)
  - **very efficient** for high dimensional datasets with lots of points

SVD

$$X = USV^{T}$$



Data X, one row per data point

US gives coordinates of rows of X in the space of principle components

S is diagonal, $S_k > S_{k+1}$, S_k^2 is kth largest eigenvalue

Rows of V^T are unit length eigenvectors of X^TX $= \Sigma$

If cols of X have zero mean, then $X^TX = c\ \Sigma$ and eigenvects are the Principle Components

[from Wall et al., 2003]

Slide from Tom Mitchell

Singular Value Decomposition

To generate principle components:

- Subtract mean $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x^n$ from each data point, to create zero-centered data
- Create matrix X with one row vector per (zero centered) data point
- Solve SVD: $X = USV^T$
- Output Principle components: columns of V (= rows of V^T)
  - Eigenvectors in V are sorted from largest to smallest eigenvalues
  - S is diagonal, with s_k^2 giving eigenvalue for kth eigenvector

Slide from Tom Mitchell

Singular Value Decomposition

To project a point (column vector x) into PC coordinates:

$V^T x$

If x_i is ith row of data matrix X, then

- (ith row of US) = $V^T x_i^T$
- $(US)^T = V^T X^T$

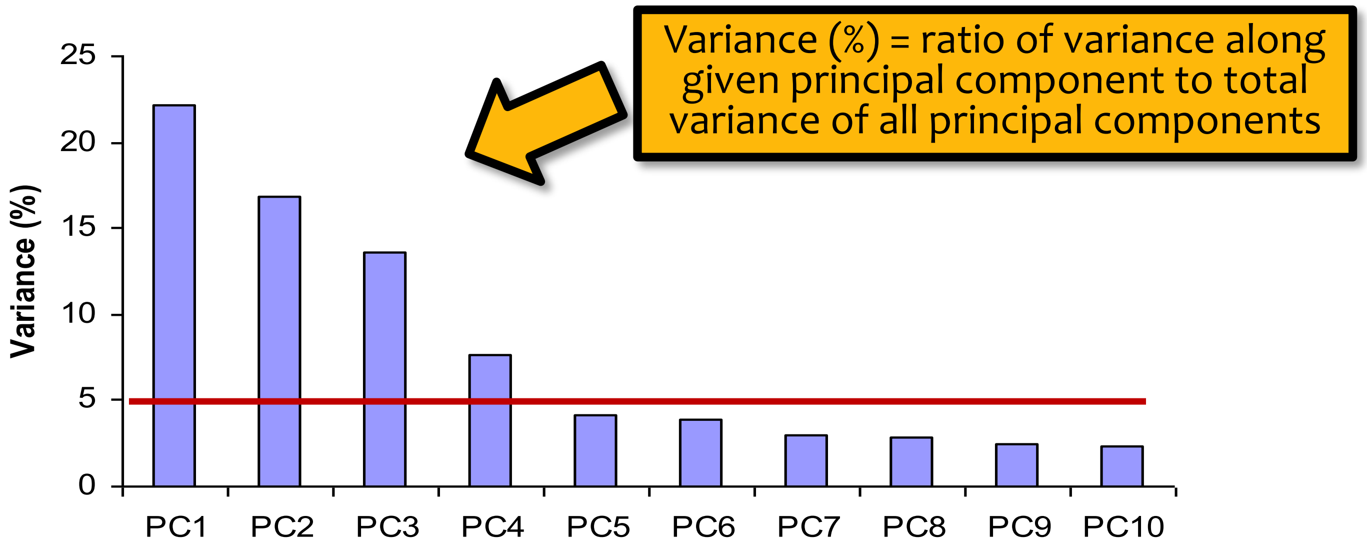
To project a column vector x to ~~M~~ K dim Principle Components subspace, take just the first ~~M coordinates~~ K row of $V^T x$

Slide from Tom Mitchell

How Many PCs?

- For M original dimensions, sample covariance matrix is MxM, and has up to M eigenvectors. So M PCs.
- Where does dimensionality reduction come from?

  Can *ignore* the components of lesser significance.



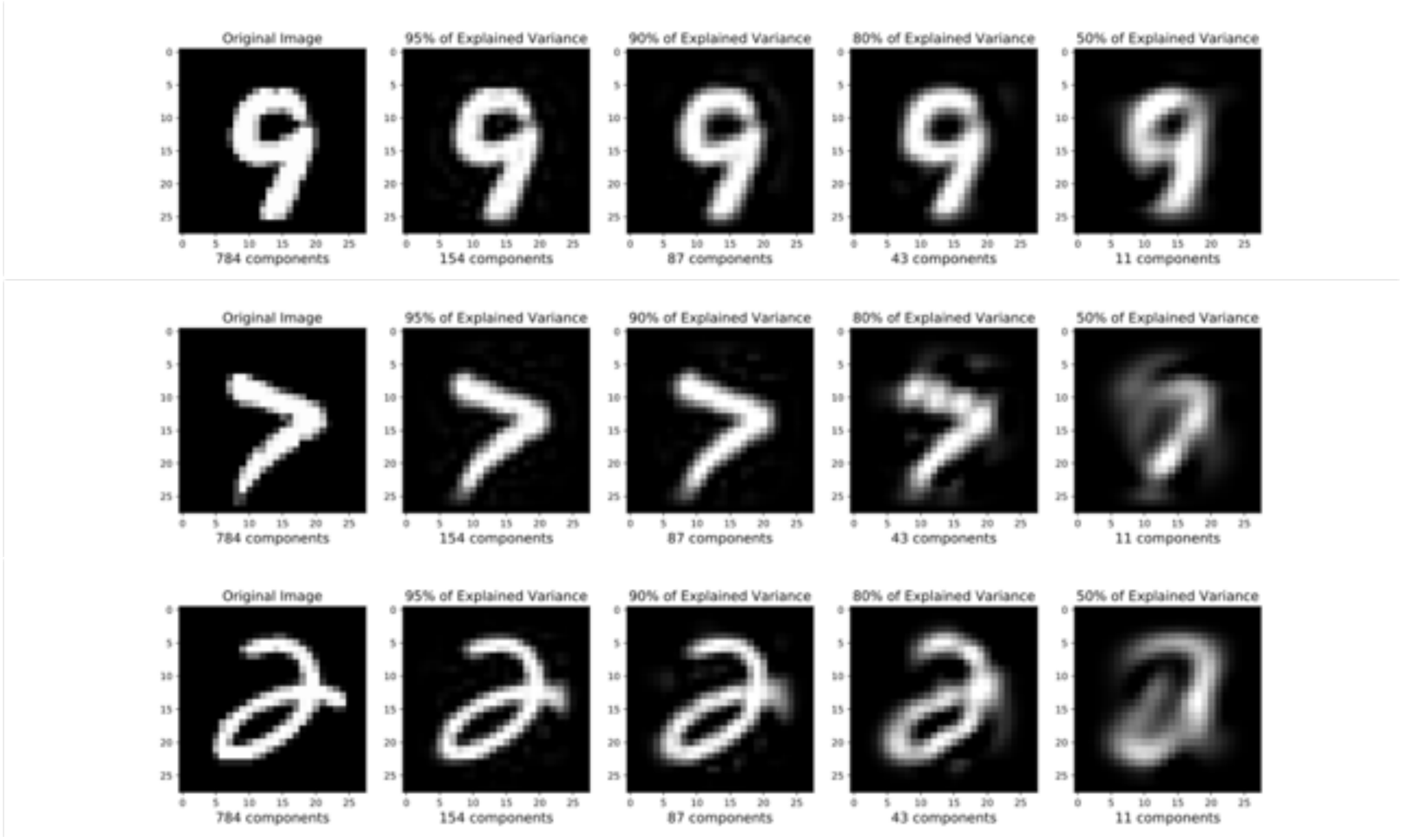
- You do lose some information, but if the eigenvalues are small, you don't lose much
  - M dimensions in original data
  - calculate M eigenvectors and eigenvalues
  - choose only the first D eigenvectors, based on their eigenvalues
  - final data set has only D dimensions

PCA EXAMPLES

Projecting MNIST digits

Task Setting:

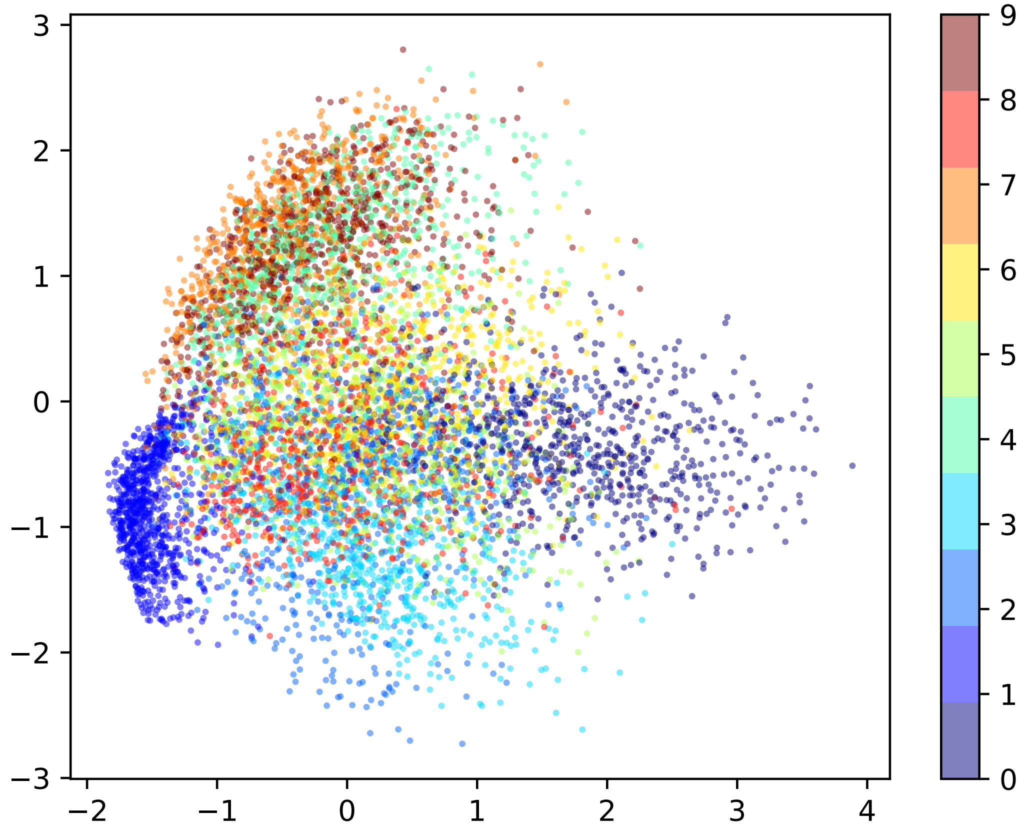
1. Take 25x25 images of digits and project them down to K components
2. Report percent of variance explained for K components
3. Then project back up to 25x25 image to visualize how much information was preserved



Projecting MNIST digits

Task Setting:

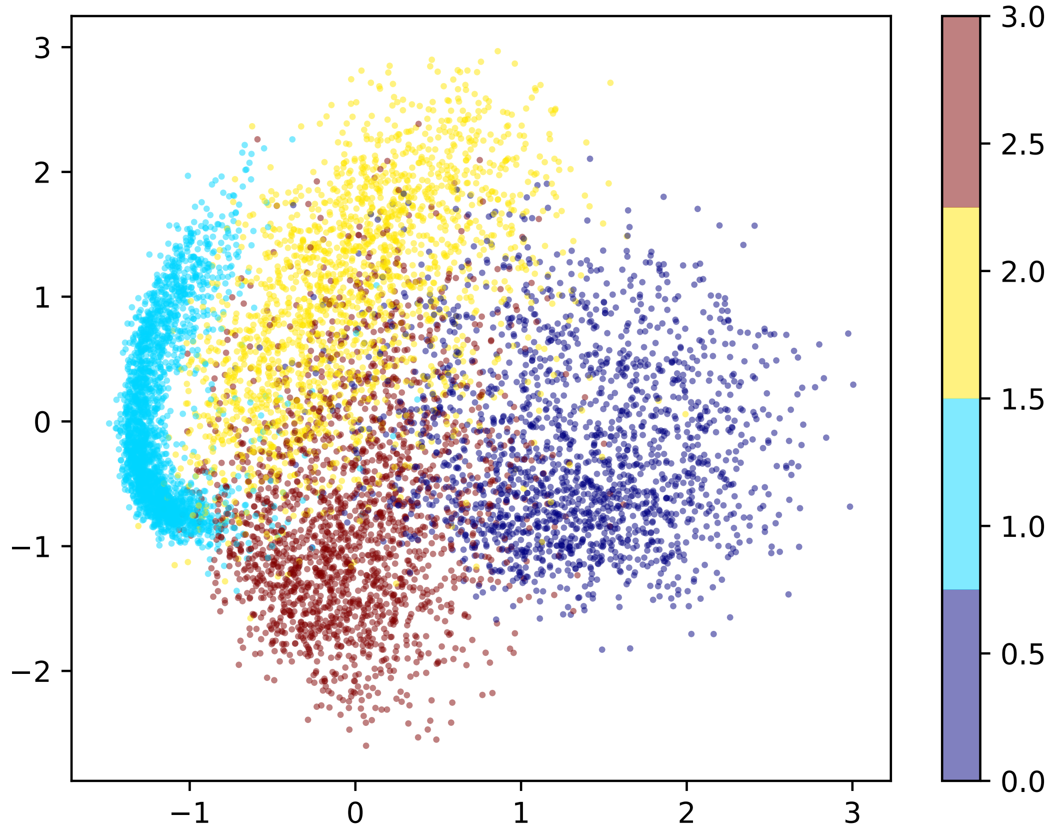
1. Take 25x25 images of digits and project them down to 2 components
2. Plot the 2 dimensional points
3. Here we look at all ten digits 0 - 9



Projecting MNIST digits

Task Setting:

1. Take 25x25 images of digits and project them down to 2 components
2. Plot the 2 dimensional points
3. Here we look at just four digits 0, 1, 2, 3



Learning Objectives

Dimensionality Reduction / PCA

You should be able to…

1. Define the sample mean, sample variance, and sample covariance of a vector-valued dataset
2. Identify examples of high dimensional data and common use cases for dimensionality reduction
3. Draw the principal components of a given toy dataset
4. Establish the equivalence of minimization of reconstruction error with maximization of variance
5. Given a set of principal components, project from high to low dimensional space and do the reverse to produce a reconstruction
6. Explain the connection between PCA, eigenvectors, eigenvalues, and covariance matrix
7. Use common methods in linear algebra to obtain the principal components

CLUSTERING

Clustering, Informal Goals

Goal: Automatically partition unlabeled data into groups of similar data points.

Question: When and why would we want to do this?

Useful for:

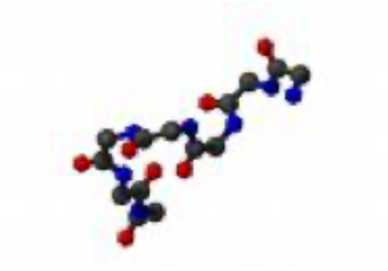
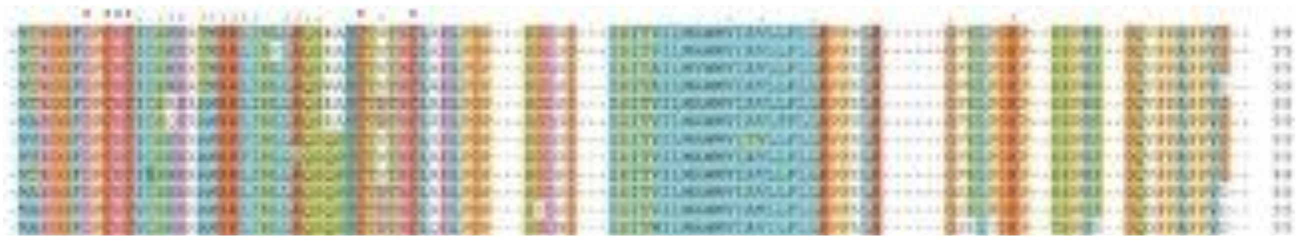
- Automatically organizing data.
- Understanding hidden structure in data.
- Preprocessing for further analysis.
  - Representing high-dimensional data in a low-dimensional space (e.g., for visualization purposes).

Slide courtesy of Nina Balcan

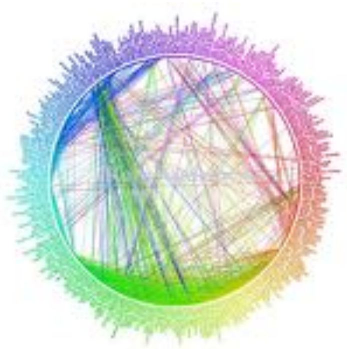
Applications (Clustering comes up everywhere…)

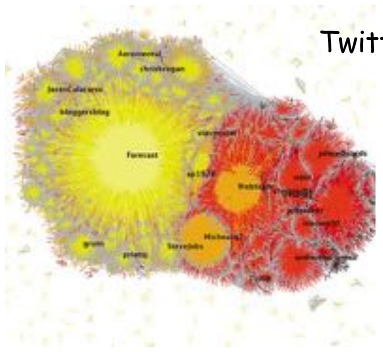
- Cluster news articles or web pages or search results by topic.

- Cluster protein sequences by function or genes according to expression profile.

- Cluster users of social networks by interest (community detection).


Facebook network


Twitter Network

Slide courtesy of Nina Balcan

Applications (Clustering comes up everywhere…)

- Cluster customers according to purchase history.

- Cluster galaxies or nearby stars (e.g. Sloan Digital Sky Survey)

- And many many more applications….

Slide courtesy of Nina Balcan

Clustering

Question: Which of these partitions is "better"?