Machine Learning: Chenhao Tan

University of Colorado Boulder

LECTURE 22

Slides adapted from Jordan Boyd-Graber, Chris Ketelsen

Logistics

- Prelim 2
- HW5
- Project updates

Learning objectives

- Understand what unsupervised learning is for
- Learn principal component analysis

Outline

Unsupervised learning

Principal Component Analysis

Outline of CSCI 4622

We've already covered stuff in blue!

- Problem formulations: classification, regression
- Supervised techniques: decision trees, nearest neighbors, perceptron, linear models, neural networks, support vector machine, kernel methods
- Unsupervised techniques: clustering, linear dimensionality reduction
- "Meta-techniques": ensembles, expectation-maximization
- Understanding ML: limits of learning, practical issues, bias & fairness
- Recurring themes: (stochastic) gradient descent

Supervised vs. unsupervised learning

Data	Labels
X	Y

- **Supervised methods** find patterns in **fully observed** data and then try to predict something from **partially observed** data.
- For example, in sentiment analysis, after learning something from annotated reviews, we want to take new reviews and automatically identify sentiments.

Supervised vs. unsupervised learning

Data

X

Hidden Structure

Z

- **Unsupervised methods** find **hidden structure** in data, structure that we can never formally observe.

When do we need unsupervised learning?

When do we need unsupervised learning?

- Acquiring labels is expensive
- You may not even know what labels to acquire

When do we need unsupervised learning?

- Exploratory data analysis
- Learn patterns/representations that can be useful for supervised learning (representation learning)
- Generate data
- . . .

When do we need unsupervised learning?

Generative adversarial networks

https://qz.com/1090267/artificial-intelligence-can-now-show-you-how-those-pants-will-fit/

Unsupervised learning

- Dimensionality reduction
- Clustering
- Topic modeling

Outline

Unsupervised learning

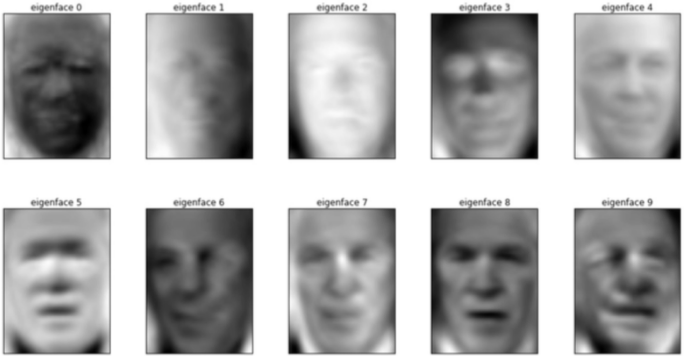
Principal Component Analysis

Example: Eigenfaces / Facial Recognition

- "Labeled Faces in the Wild" dataset
- Roughly 1300 images of 7 different people's faces in various orientation and lighting
- Images are 50x37 grayscale or 1850 features

Example: Eigenfaces / Facial Recognition

Example: Eigenfaces / Facial Recognition



Principal Component Analysis - Motivation

- We need to shift our perspective
- Change the definition of up-down-left-right
- Choose new features as linear combinations of old features
- Change of feature-basis

Principal Component Analysis - Motivation

- We need to shift our perspective
- Change the definition of up-down-left-right
- Choose new features as linear combinations of old features
- Change of feature-basis

Important: Center and normalize data before performing PCA. We will assume that this has already been done in this lecture.

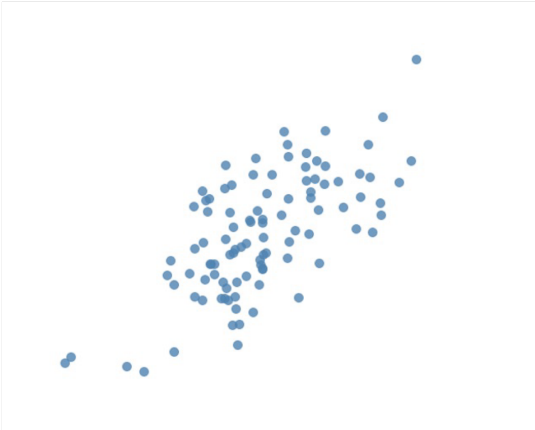
Principal Component Analysis - Motivation

Proceed incrementally:

- If we could choose one combination to describe data?
- Which combination leads to the least loss of information?
- Once we've found that one, look for another one, perpendicular to the first, the retains the next most amount of information
- Repeat until done (or good enough)

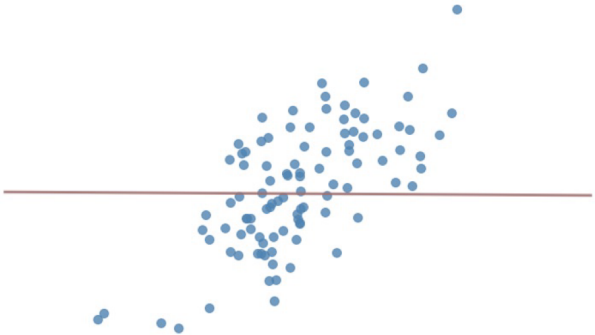
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



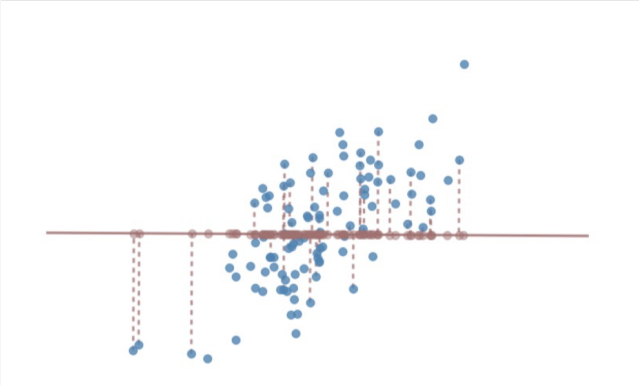
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



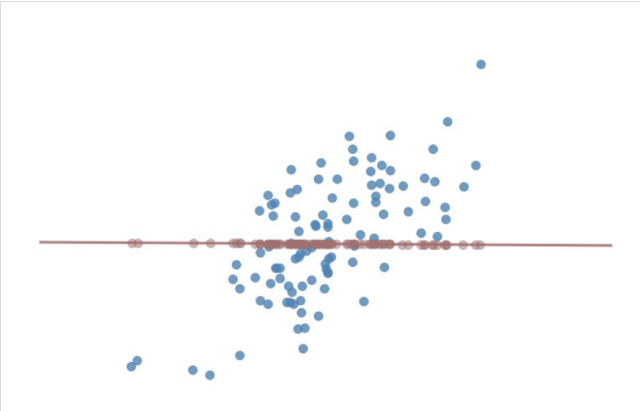
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



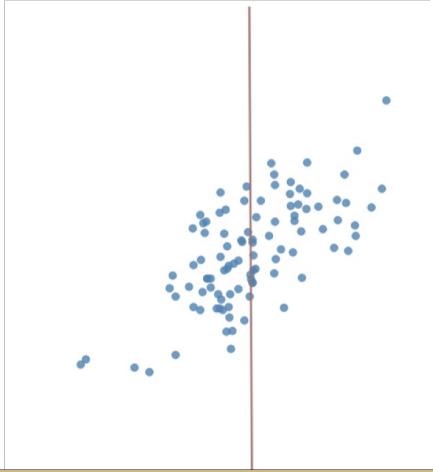
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



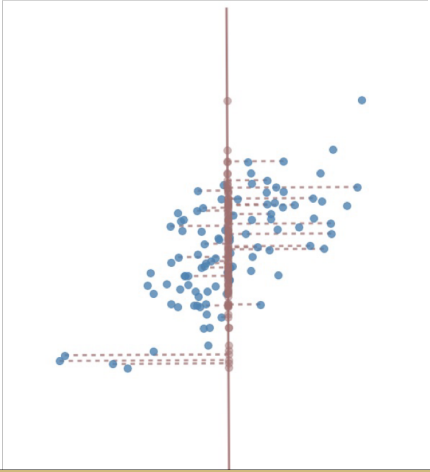
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



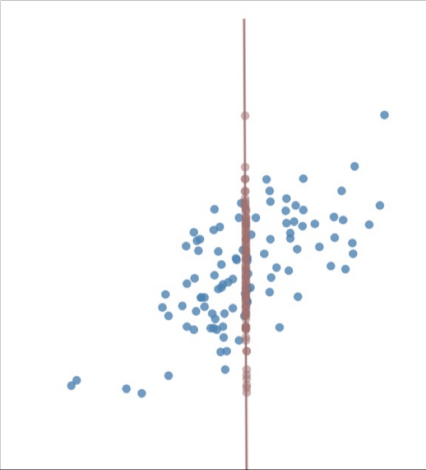
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



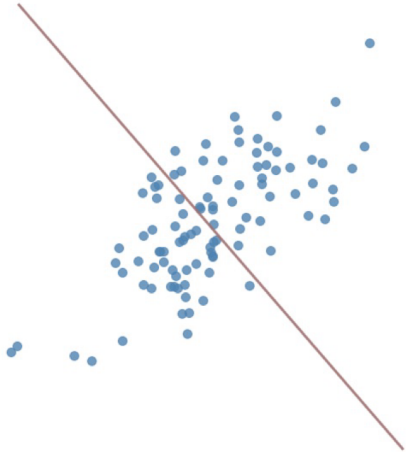
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



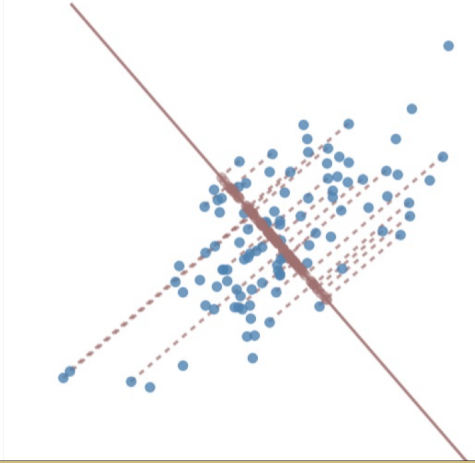
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



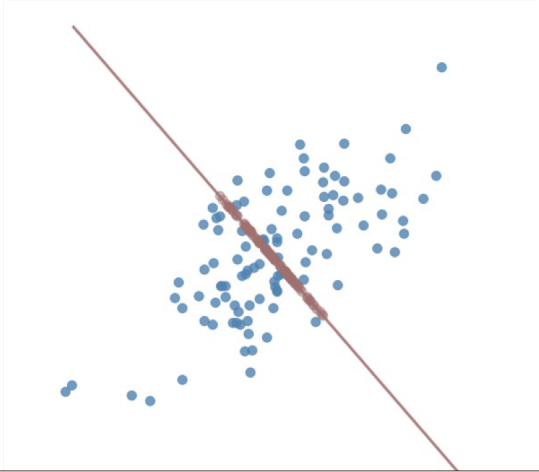
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



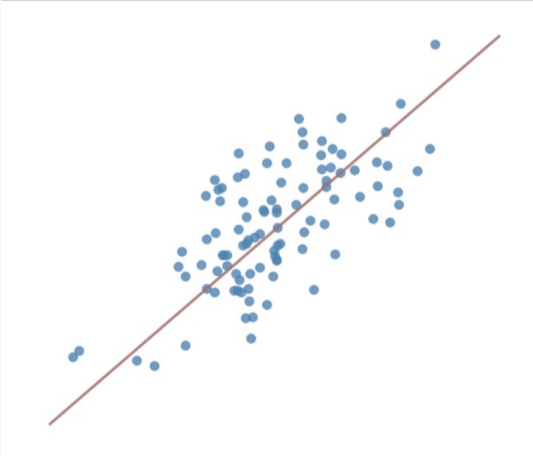
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



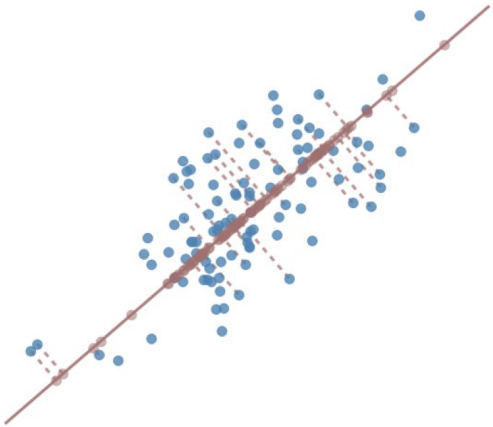
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



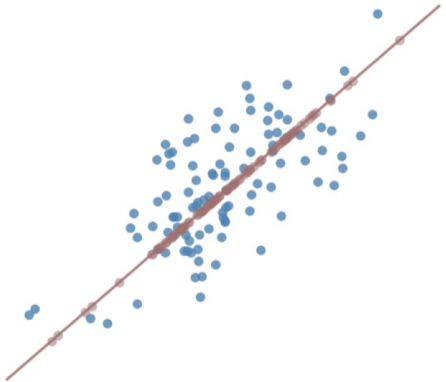
Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



Principal Component Analysis - 1D

How should we reduce this dataset to one dimension?



Principal Component Analysis

The best vector to project onto is called the **1st principal component**.

Principal Component Analysis

The best vector to project onto is called the **1st principal component**.
What properties should it have?

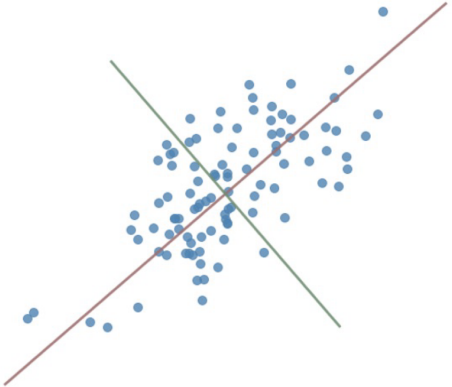
- Should capture largest variance in data
- Should probably be a unit vector

After we've found the first, look the second which:

- Captures largest amount of leftover variance
- Should probably be a unit vector
- Should be orthogonal to the one that came before it

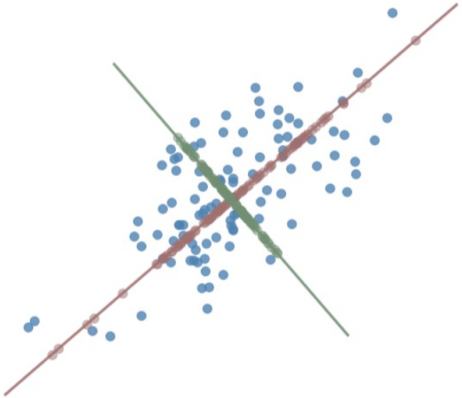
Principal Component Analysis

Principal components of the previous example



Principal Component Analysis

Principal components of the previous example



Principal Component Analysis

Main idea: The principal components give a new perpendicular coordinate system to view data where each principle component describes successively less and less information.

Principal Component Analysis

OK, so how do we find the first principle component?
Store data in an $m \times D$ matrix X (where $\mathbf{x}_i$ are rows)
Define covariance matrix $C^X = \frac{1}{m-1} X^T X$
Claim: First principle component $\mathbf{v}_1$ is the eigenvector of C^X corresponding to the largest eigenvalue
Recall: $\mathbf{v}$ is an eigenvector of A with associated eigenvalue λ if

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

Proof?

Facts about $C^X = \frac{1}{m-1} X^T X$

- Symmetric
- All eigenvalues are real (b/c symmetric)
- All eigenvalues are nonnegative (because it is positive semidefinite)
- C^X has D mutually orthogonal eigenvectors (which can be scaled to unit length)

Facts about $C^X = \frac{1}{m-1} X^T X$

- Symmetric
- All eigenvalues are real (b/c symmetric)
- All eigenvalues are nonnegative (because it is positive semidefinite)
- C^X has D mutually orthogonal eigenvectors (which can be scaled to unit length)

$$C^X = \sum_{i=1}^{D} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T,$$

where λ_i are the eigenvalues ($\lambda_1 \geq \lambda_2 \geq \ldots, \geq \lambda_D$), $\boldsymbol{v}_i$ is the eigenvector associated with λ_i.

Principal Component Analysis

Proof: Let $\mathbf{w}$ represent the first pc (which we know nothing about).
Just know that we want it to be unit length and capture the most variance in the data.
We project each training example onto the first pc via dot product.

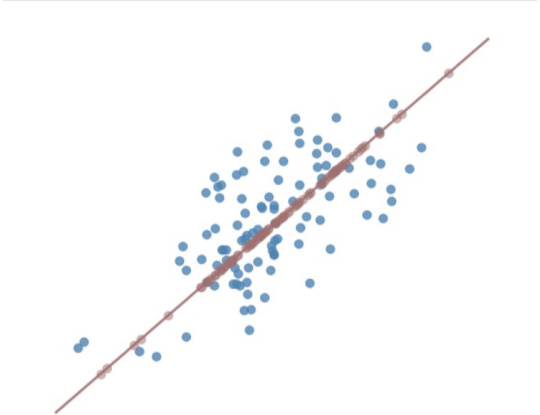
$$\mathbf{x}_i = (\mathbf{x}_i \cdot \mathbf{w})\ \mathbf{w}$$

The scalar component of the projection is 1D representation of $\mathbf{x}_i$
Get all scalar components:

$$X\mathbf{w} \qquad [X\mathbf{w}]_i = \mathbf{x}_i \cdot \mathbf{w}$$

Principal Component Analysis

An example of projection.



Principal Component Analysis

But how do we find $\mathbf{w}$?
$[X\mathbf{w}]_i$ are features in pc space. As X is already centered, so $\text{mean}(X\mathbf{w}) = 0$. Their variance is

$$\frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i \cdot \mathbf{w})^2 = \frac{1}{m-1}(X\mathbf{w})^T(X\mathbf{w}) = \frac{1}{m-1}\mathbf{w}X^TX\mathbf{w}$$

$$\frac{1}{m-1}\sum_{i=1}^{m}(\mathbf{x}_i \cdot \mathbf{w})^2 = \mathbf{w}^T C^X \mathbf{w} =: \sigma^2_{\mathbf{w}}$$

Want to choose $\mathbf{w}$ to have unit length, and make $\sigma^2_{\mathbf{w}}$ as large as possible. This is constrained optimization!

Principal Component Analysis

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathbf{w}^T C^X \mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^T \mathbf{w} = 1 \end{aligned}$$

Principal Component Analysis

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathbf{w}^T C^X \mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^T \mathbf{w} = 1 \end{aligned}$$

Constrained optimization!
Define Lagrangian

$$L(\mathbf{w}, \lambda) = \mathbf{w}^T C^X \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{w} - 1)$$

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= \mathbf{w}^T \mathbf{w} - 1 &&= 0 \\ \nabla_{\mathbf{w}} L &= 2C^X \mathbf{w} - 2\lambda \mathbf{w} &&= 0 \end{aligned}$$

Principal Component Analysis

$$\begin{aligned} \frac{\partial L}{\partial \lambda} &= \mathbf{w}^T\mathbf{w} - 1 &&= 0 \\ \nabla_{\mathbf{w}} L &= 2C^X\mathbf{w} - 2\lambda\mathbf{w} &&= 0 \end{aligned}$$

Solution is $\mathbf{w}$ and λ such that

$$C^X\mathbf{w} = \lambda\mathbf{w} \qquad \text{and} \qquad \mathbf{w}^T\mathbf{w} = 1$$

w is an eigenvector, and max variance is eigenvalue.

$$\sigma^2_{\mathbf{w}} = \mathbf{w}^T C^X \mathbf{w} = \lambda \mathbf{w}^T\mathbf{w} = \lambda$$

So the first principal component is the eigenvector associated with the largest eigenvalue.

Principal Component Analysis

Now what is the second principal component?
Claim: The second principal component is the eigenvector of C^X with second largest eigenvalue.

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathbf{w}^T C^{X'} \mathbf{w} \\ \text{s.t.} \quad & \mathbf{w}^T \mathbf{w} = 1 \end{aligned},$$

where $X' = X - X\boldsymbol{v}_1\boldsymbol{v}^T$, $\boldsymbol{v}_1$ is the eigenvector for λ_1, the largest eigenvalue.
Use the facts about C^X that we mentioned earlier. A question in HW5.

Principal Component Analysis

How much stuff should we keep?

Principal Component Analysis

How much stuff should we keep?
Eigenvalues tell you variance capture

- OK Idea. Make a plot, look for elbows
- Better idea. Decide based on **explained variance**

$$EV = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{D} \lambda_i} \quad \text{usually choose } k \text{ s.t. } EV > 99\%$$

Wrap up

PCA

- Calculate the covariance matrix X of data points.
- Calculate eigenvectors and their corresponding eigenvalues.
- Sort the eigenvectors according to their eigenvalues in decreasing order.
- Choose first k eigenvectors which satisfies the target explained variance.
- Transform the original n dimensional data points into k dimensions.

Wrap up

Dimensionality reduction can be a useful way to

- explore data
- visualize data
- represent data