University of Colorado **Boulder**

Department of Computer Science

CSCI 5622: Machine Learning

Chenhao Tan

Lecture 17: Unsupervised Learning I (Dimensionality Reduction)

Slides adapted from Jordan Boyd-Graber, Chris Ketelsen

Administrative

- Guest lecture on Wednesday Sam Carlson
- Office hour moved to 3-4 on Friday
- Project midterm expectation
- Feedback

Learning objectives

- Understand what unsupervised learning is for
- Learn principal component analysis
- Learn singular value decomposition

Supervised learning

Data: X

Labels: Y

Unsupervised learning

Data: X

Supervised learning

Data: *X*

Labels: *Y*

Unsupervised learning

Data: *X*

Latent structure: *Z*

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?

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

Unsupervised learning

- Dimensionality reduction
- Clustering
- Topic modeling

Unsupervised learning

- Dimensionality reduction
- Clustering
- Topic modeling

Principal Component Analysis - Motivation

Given some data $\{\mathbf{x}_i\}_{i=1}^m$ where $\mathbf{x}_i \in \mathbb{R}^n$

Principal Component Analysis - Motivation

Data’s features almost certainly correlated

Principal Component Analysis - Motivation

Makes it hard to see hidden structure

Principal Component Analysis - Motivation

To make this easier, let try to reduce this to 1-dimension

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

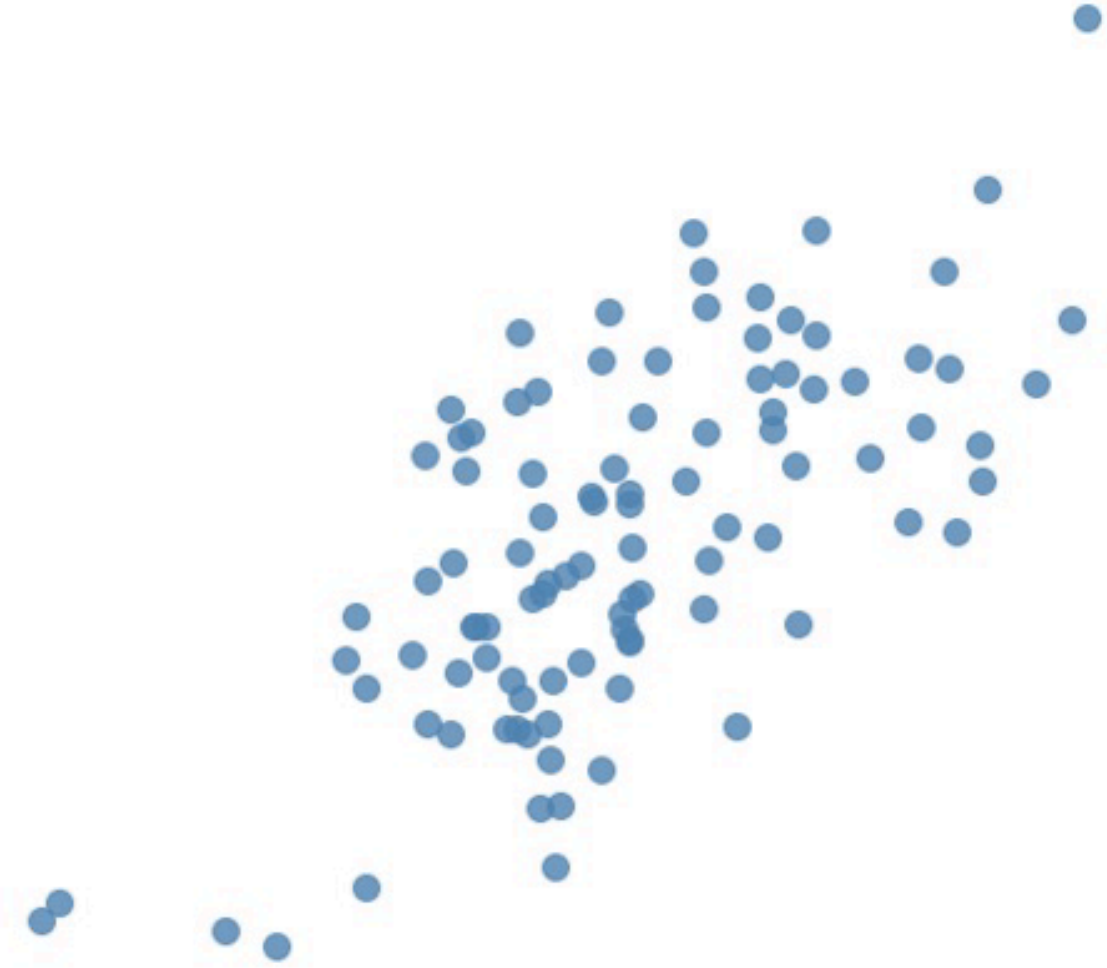
Important: Center and normalize data before performing PCA
We will assume that this has already been done in this lecture.

Principal Component Analysis - Motivation

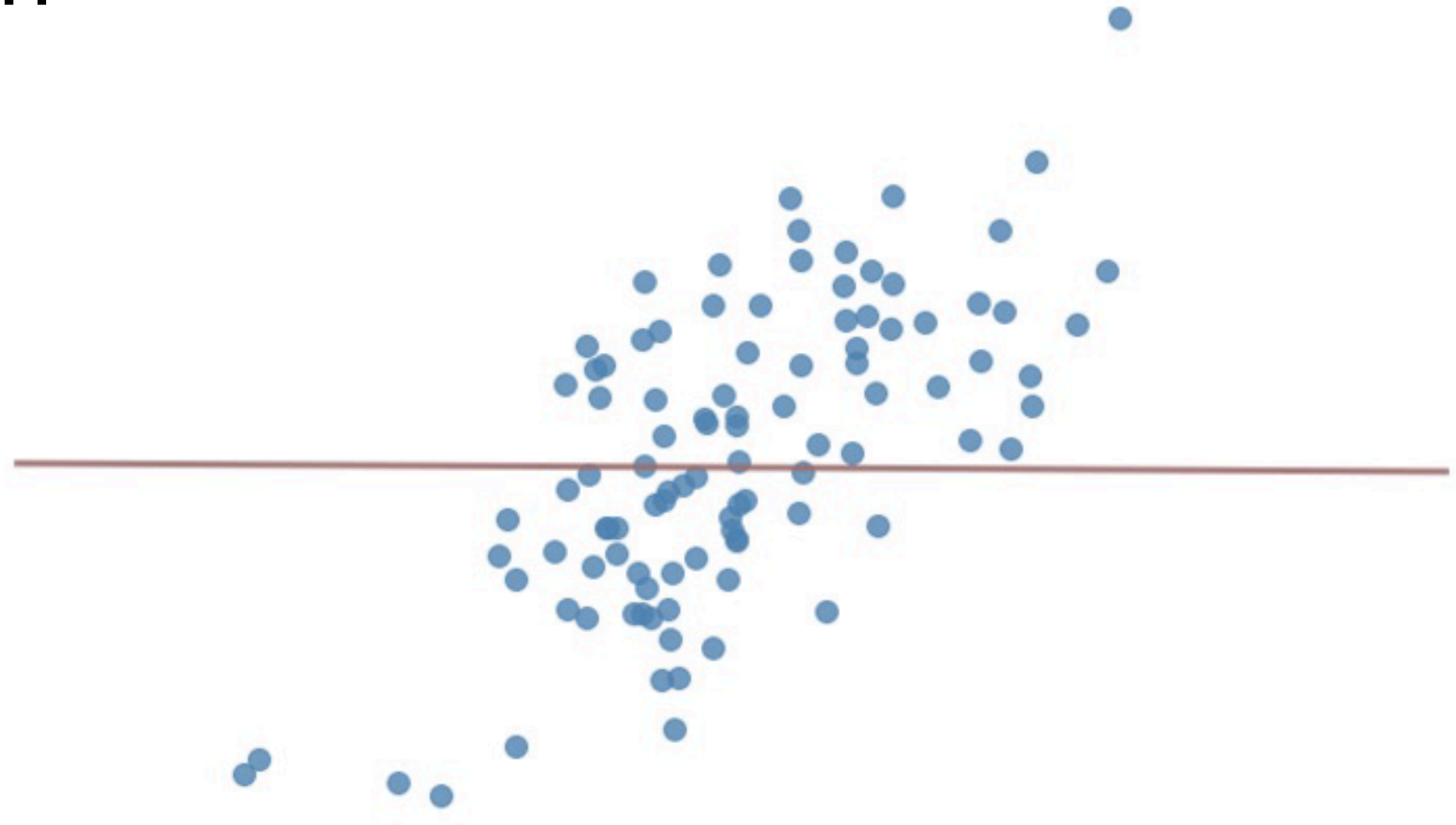
Proceed incrementally:

- 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)

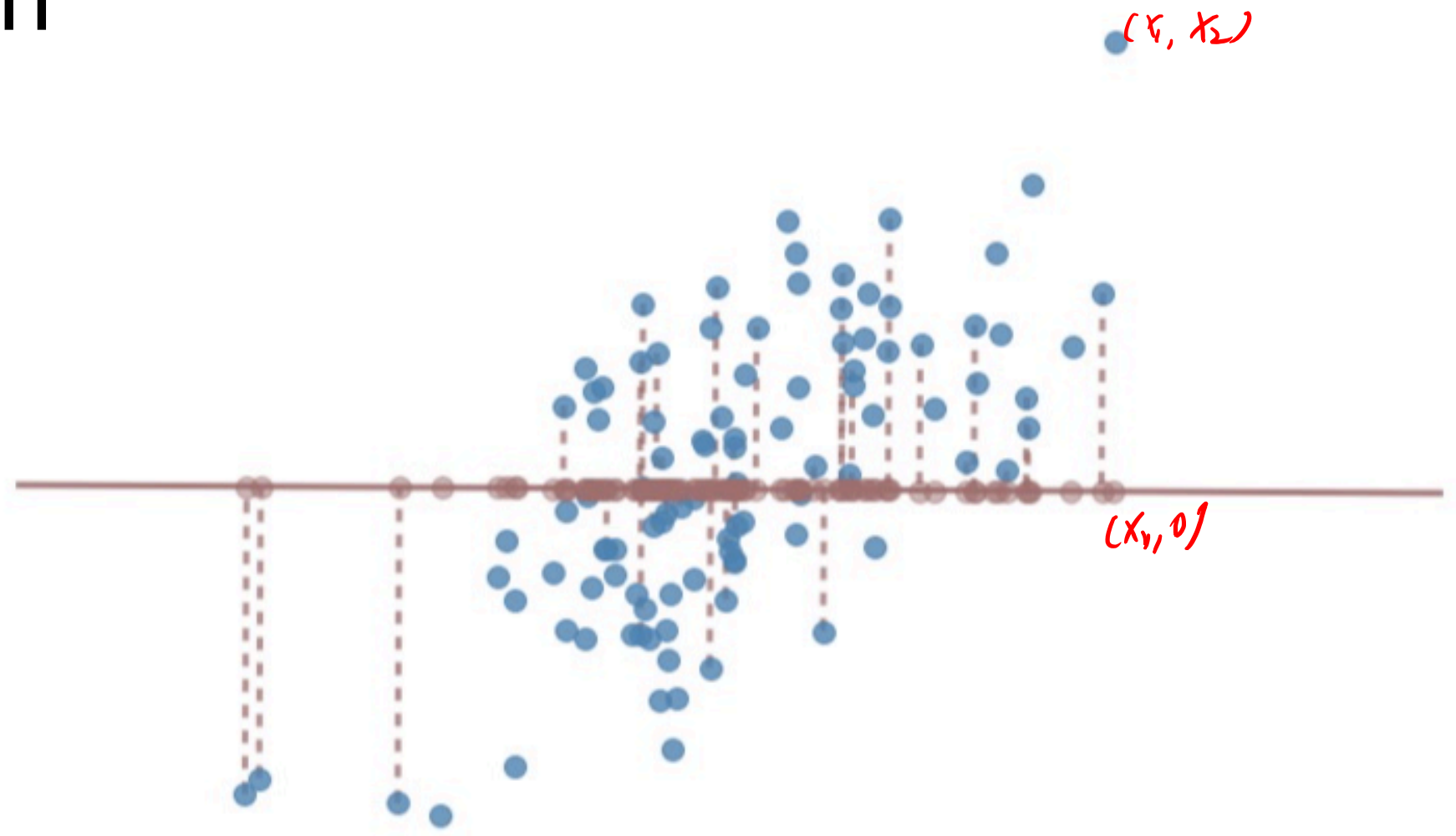
Principal Component Analysis - Motivation



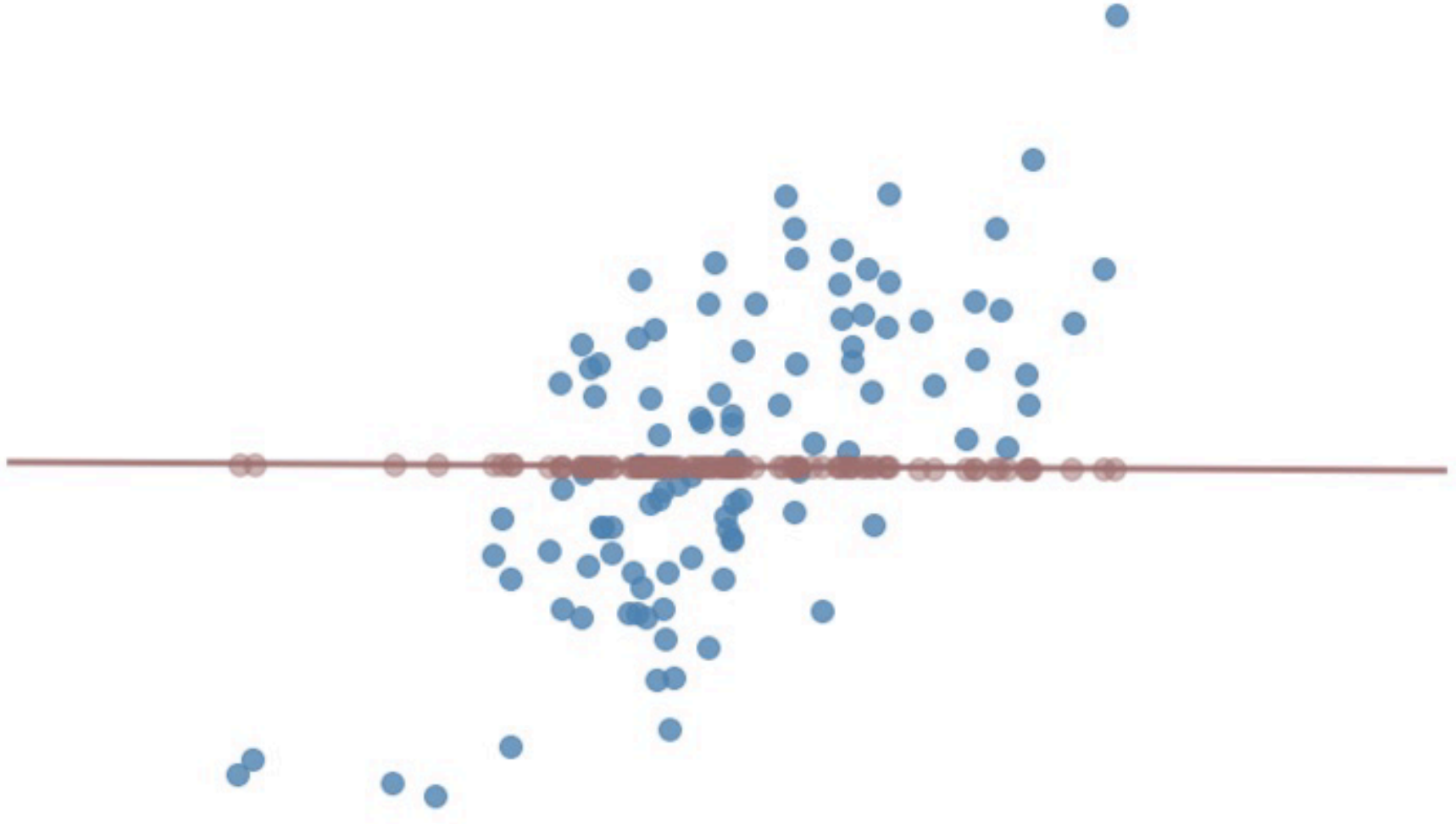
Principal Component Analysis - Motivation



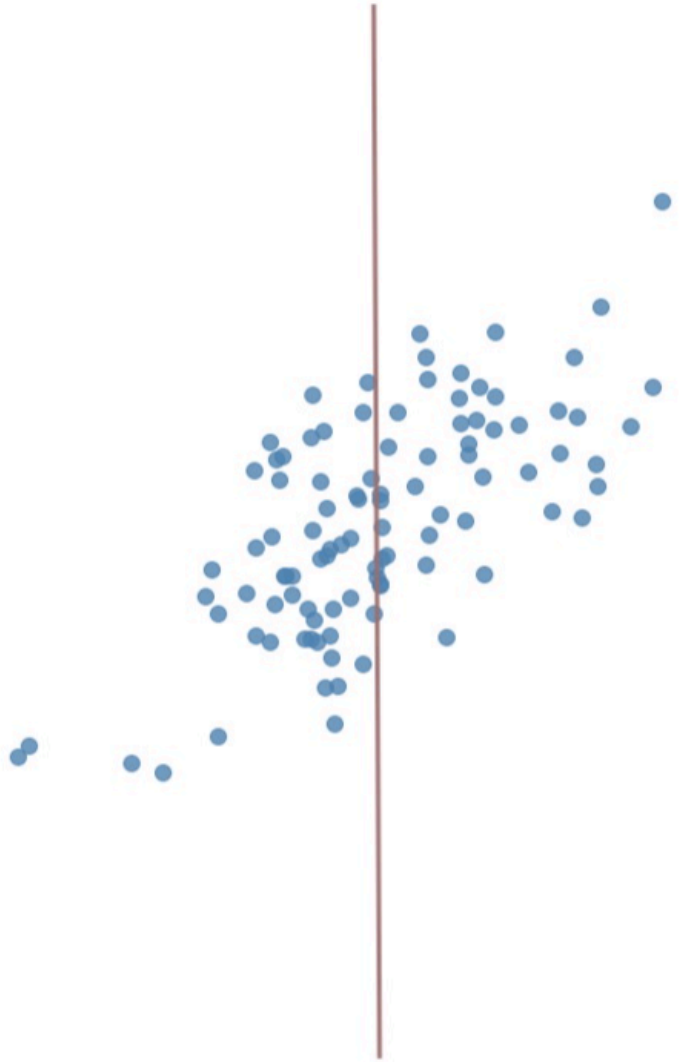
Principal Component Analysis - Motivation



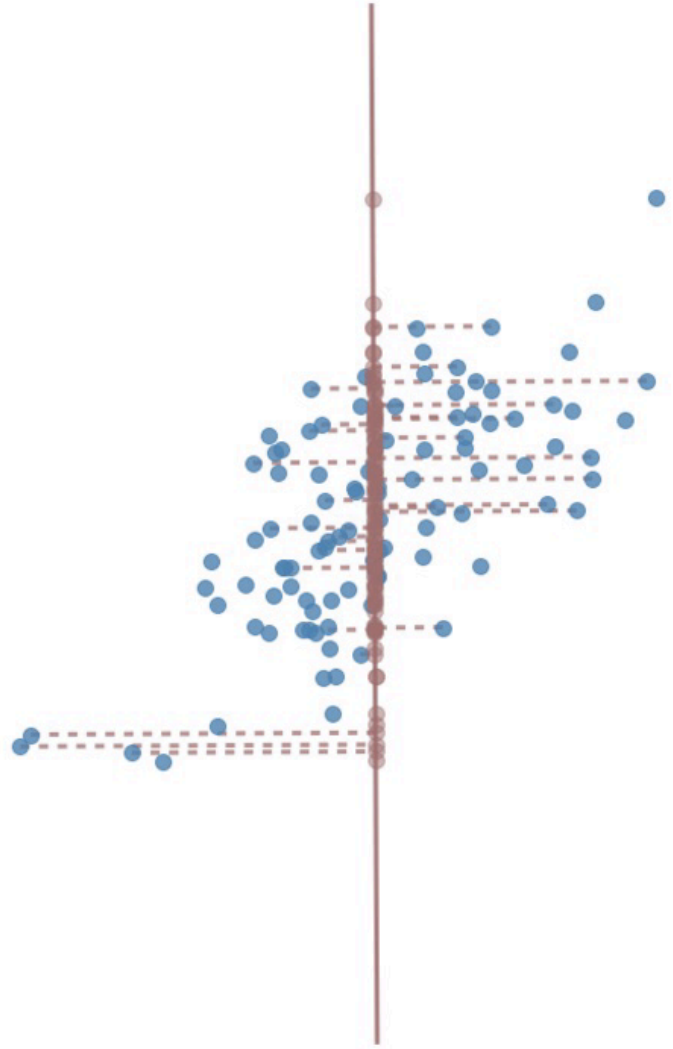
Principal Component Analysis - Motivation



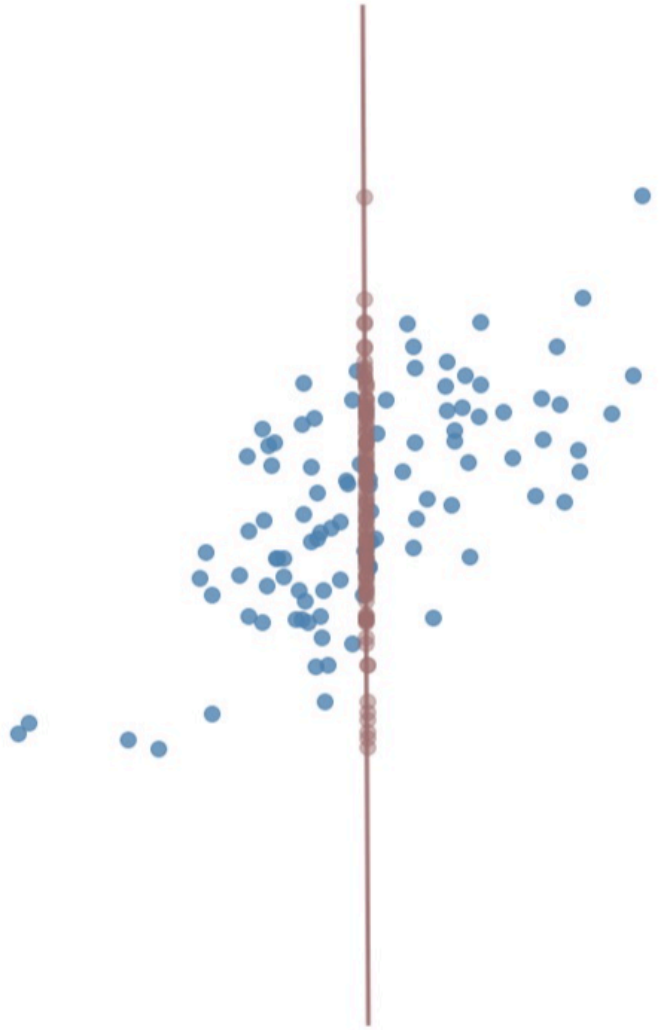
Principal Component Analysis - Motivation



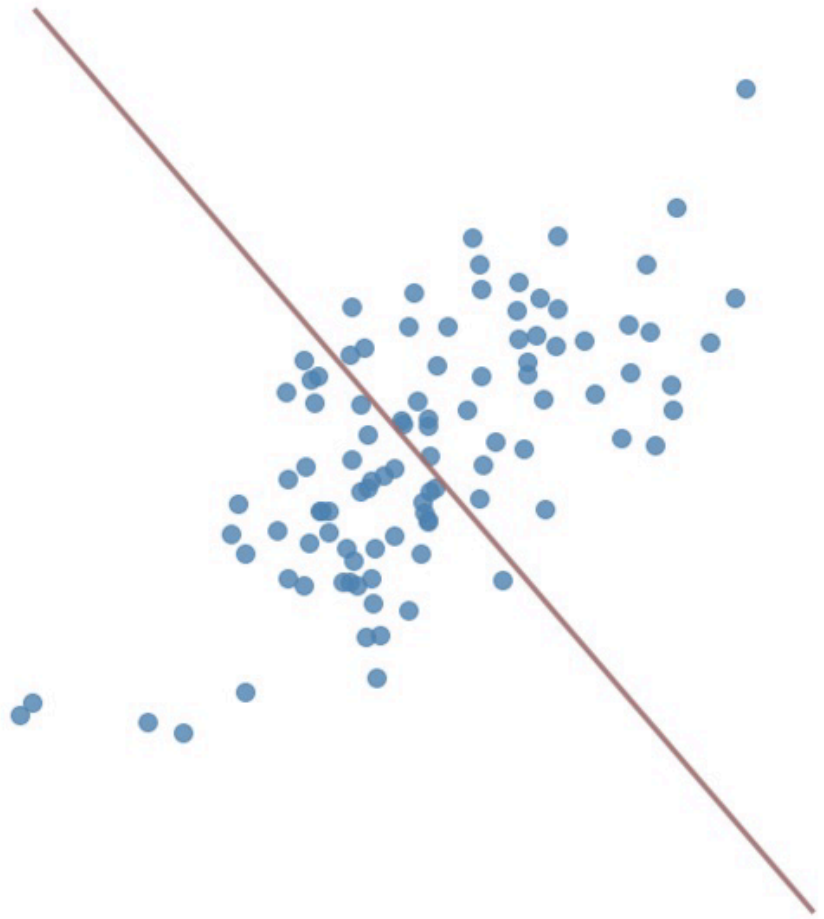
Principal Component Analysis - Motivation



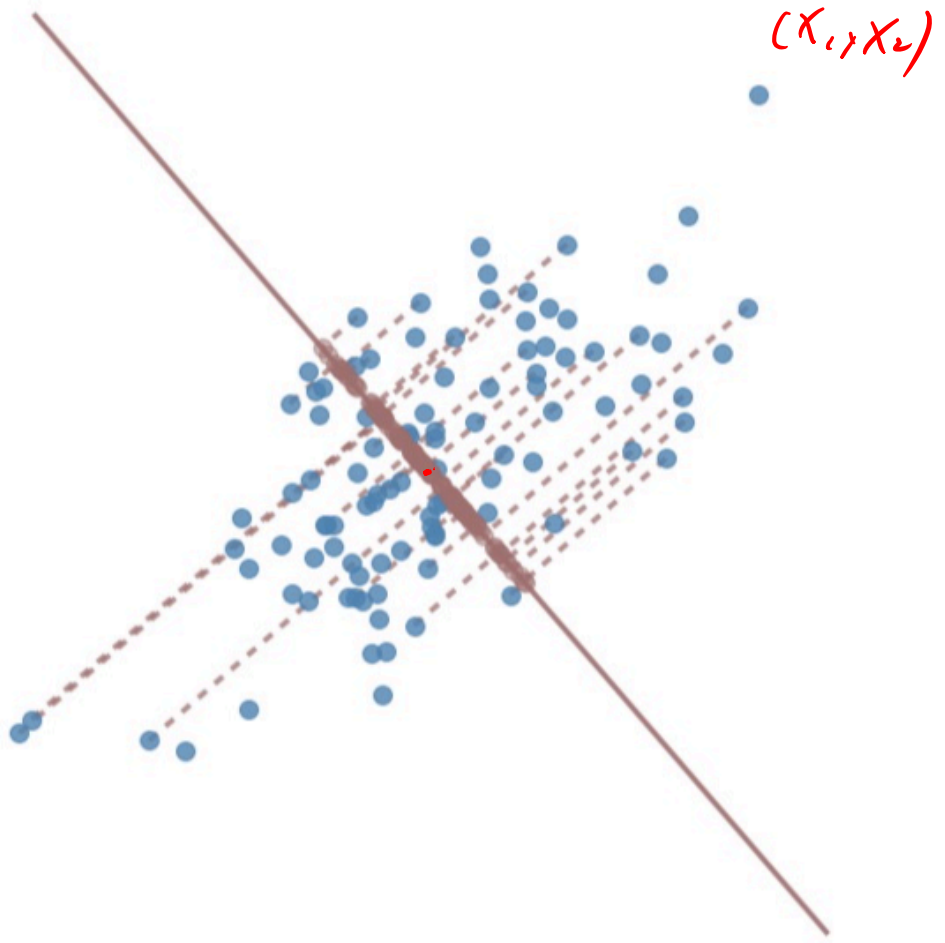
Principal Component Analysis - Motivation



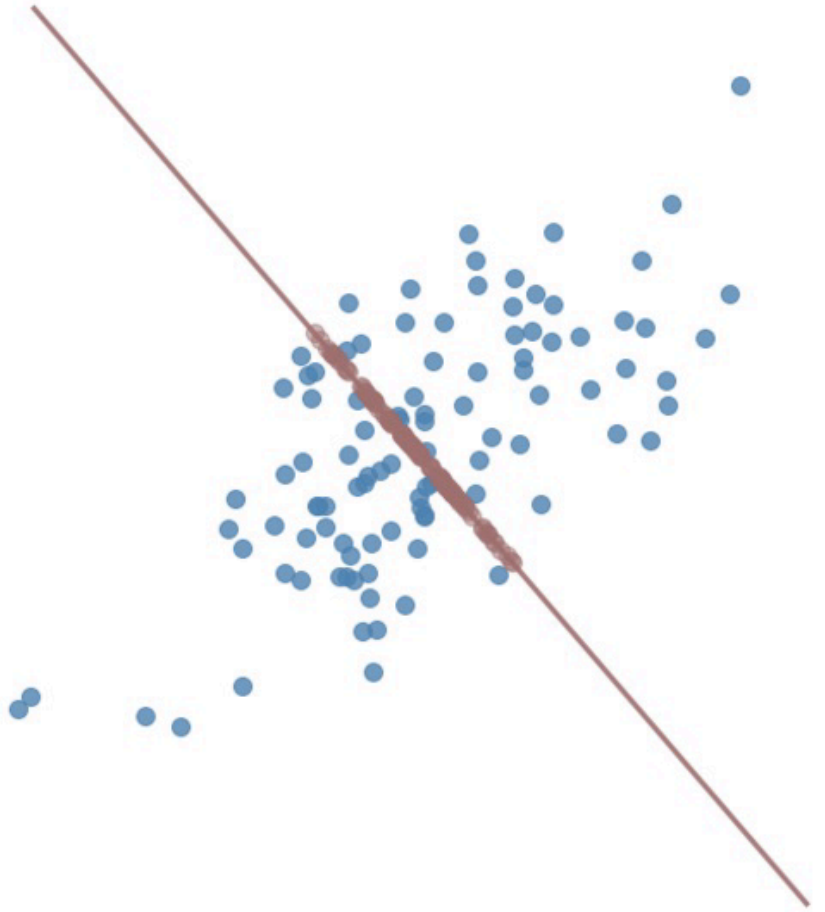
Principal Component Analysis - Motivation



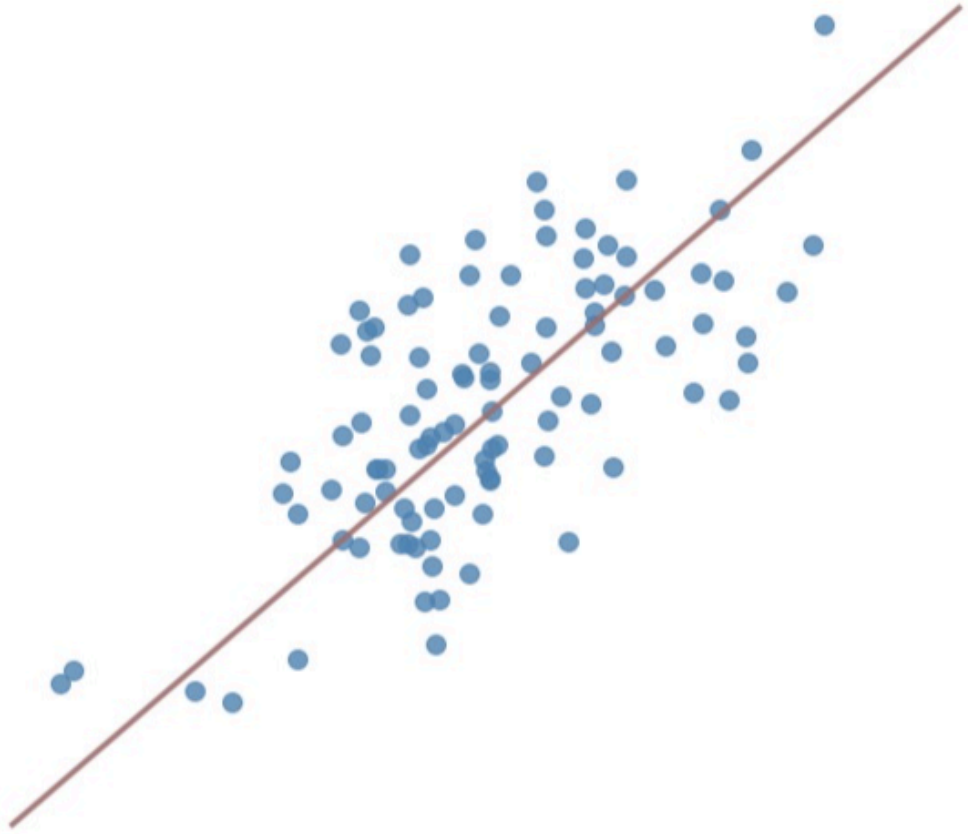
Principal Component Analysis - Motivation



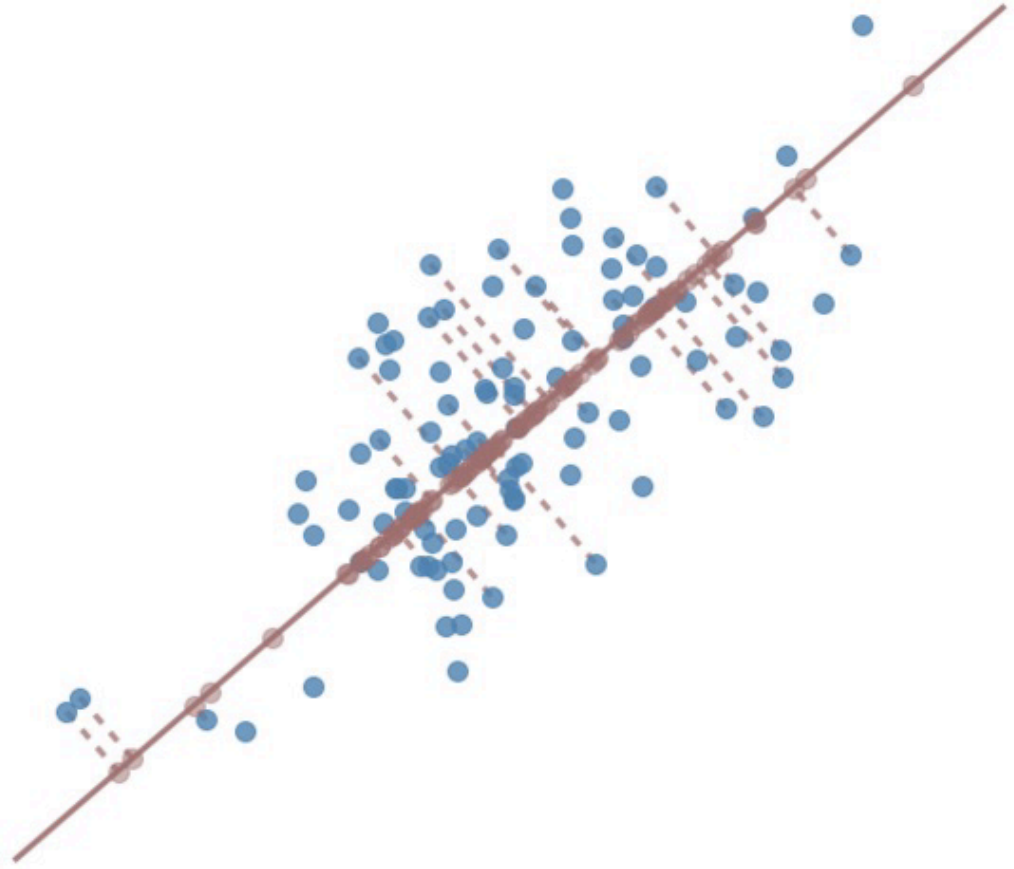
Principal Component Analysis - Motivation



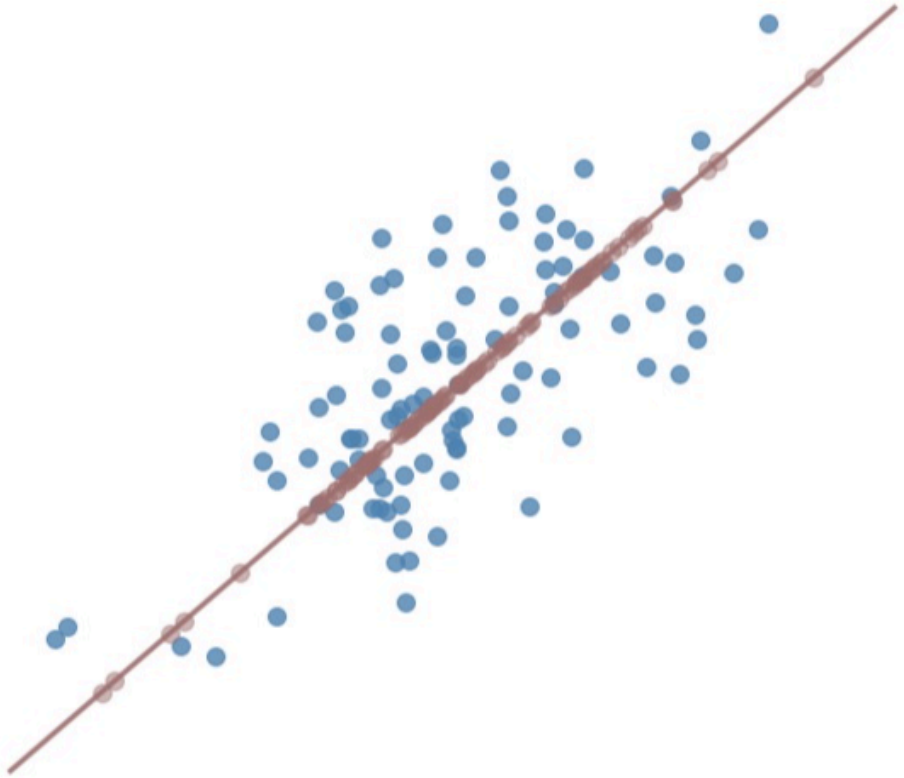
Principal Component Analysis - Motivation



Principal Component Analysis - Motivation



Principal Component Analysis - Motivation



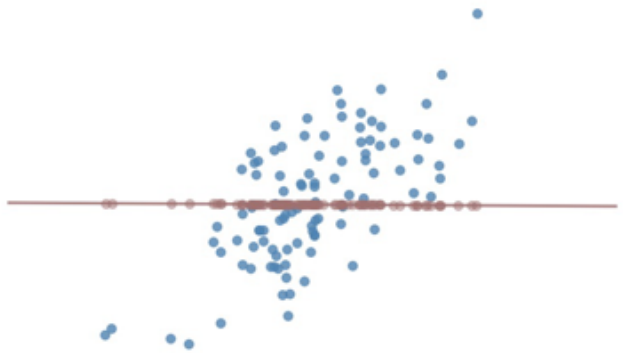
Principal Component Analysis - Motivation

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

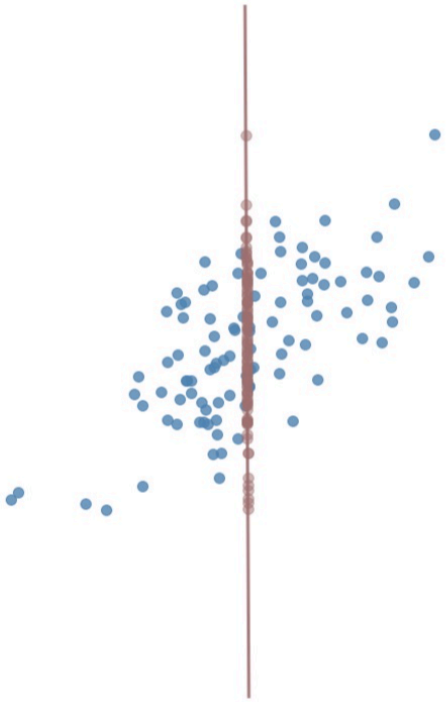
What properties should it have?

Principal Component Analysis - Motivation

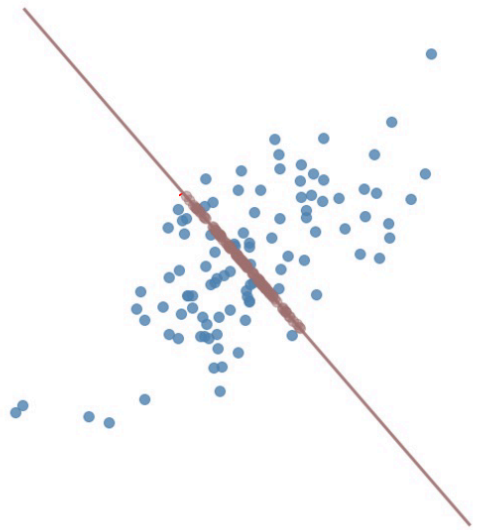
A



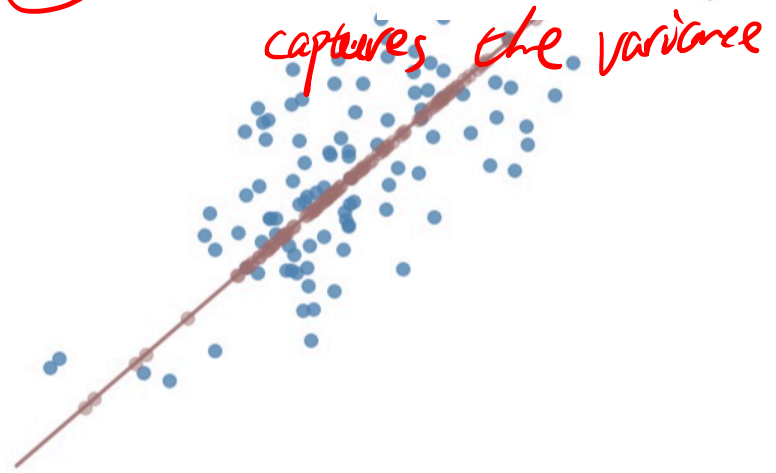
B



C



D



Principal Component Analysis - Motivation

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

Principal Component Analysis - Motivation

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

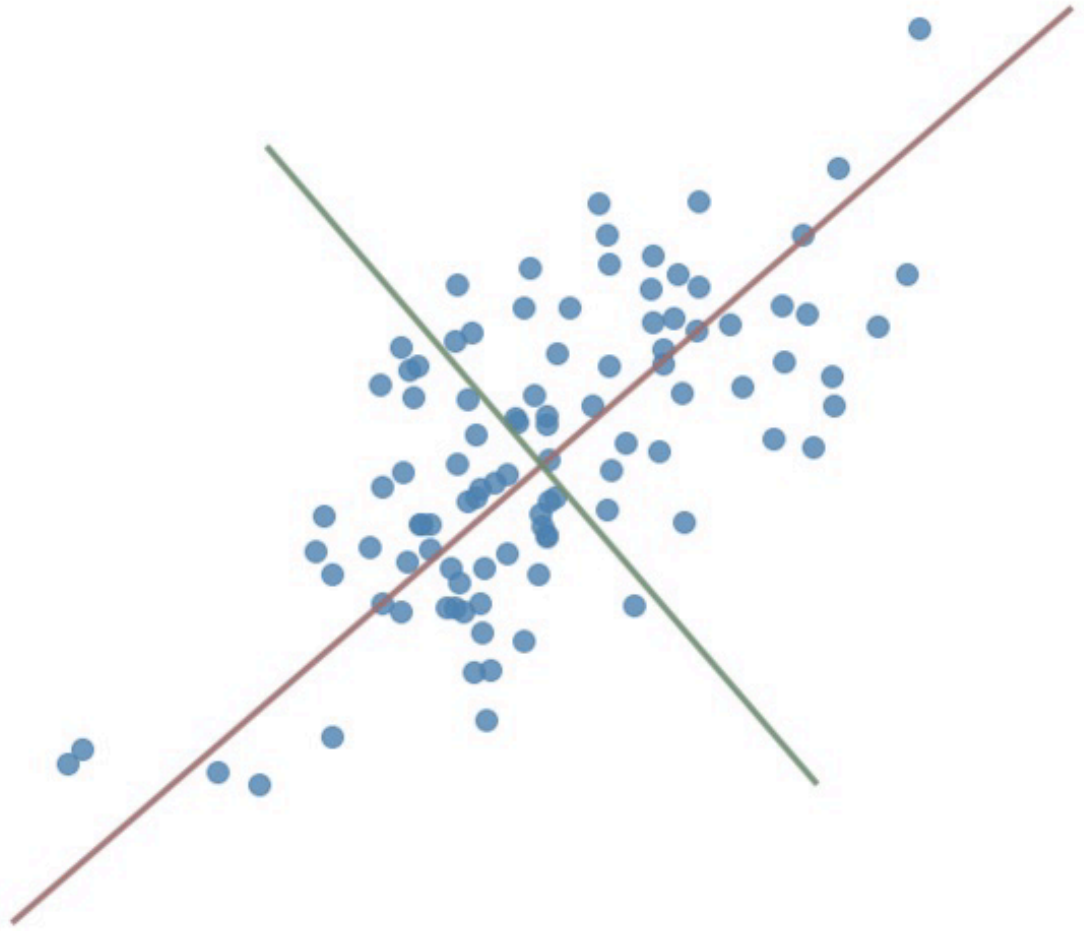
What properties should it have?

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

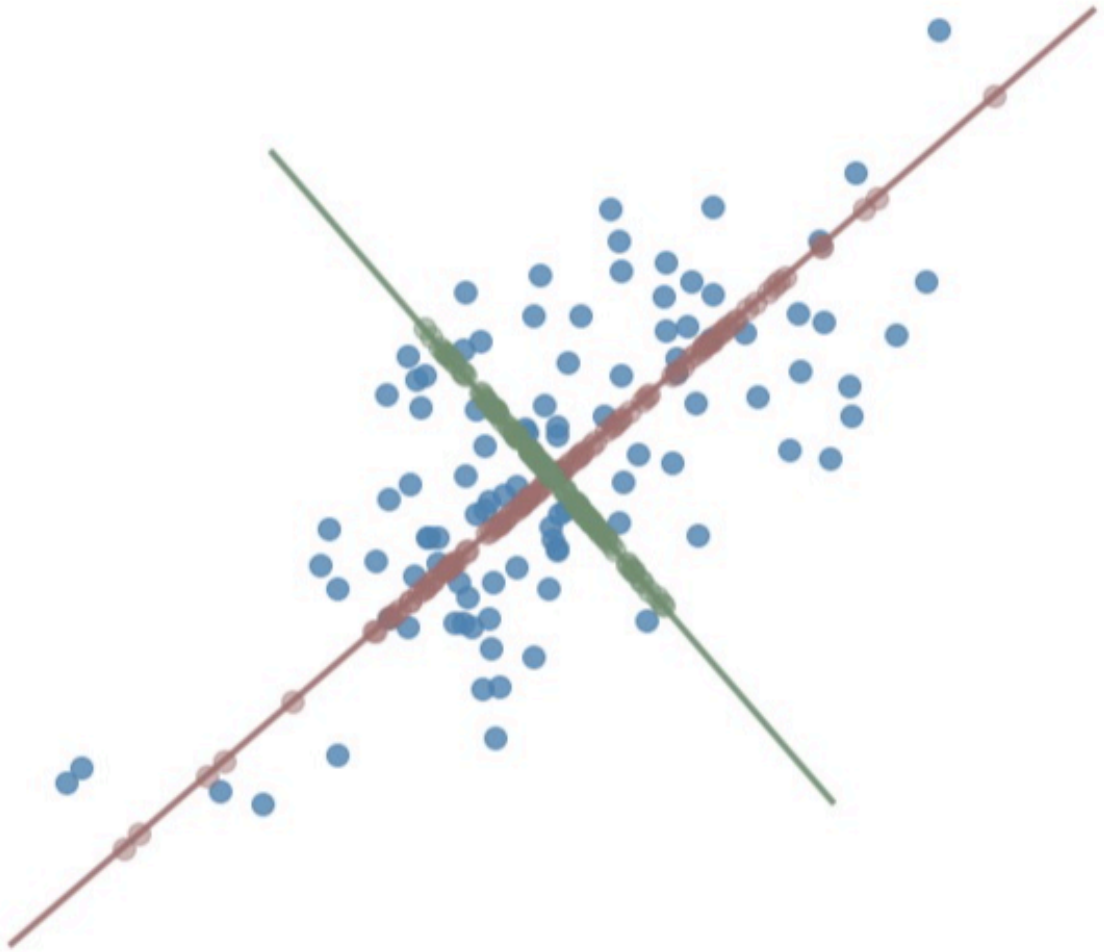
After we've found the first, look the second which:

- 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 - Motivation



Principal Component Analysis - Motivation



Principal Component Analysis - Motivation

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 - Motivation

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

So far: All we've done is a change of basis on the feature space.

But when do we reduce the dimension?

Principal Component Analysis - Motivation

But when do we reduce the dimension?

Picture data points in a 3D feature space

What if the points lied mostly along a single vector?

Principal Component Analysis - Motivation

The other two principal components are still there

But they do not carry much information

Principal Component Analysis - Motivation

The other two principal components are still there

But they do not carry much information

Throw them away and work with low dimensional representation!

Reduce 3D data to 1D

Principal Component Analysis – The How

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?

Xv
s.t. $v^T v = 1$

Principal Component Analysis – The How

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

- Symmetric $(X^T X)^T = X^T X$
- All eigenvalues are real (b/c symmetric)
- All eigenvalues are nonnegative (because Gram-Matrix)
- C^X has D mutually orthogonal eigenvectors (which can be scaled to unit length)

Principal Component Analysis – The How

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.

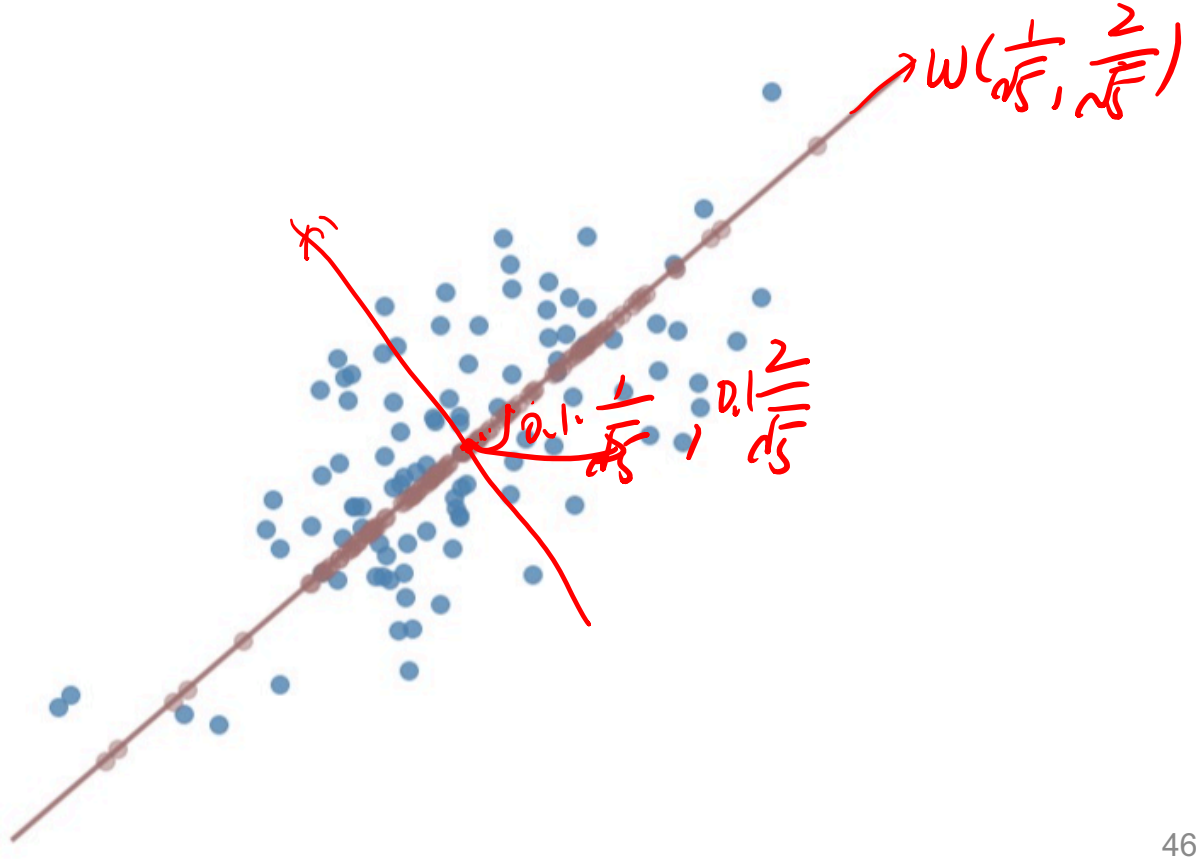
$$\mathbf{x}_i = (\mathbf{x}_i \cdot \mathbf{w})\, \mathbf{w}$$

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

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

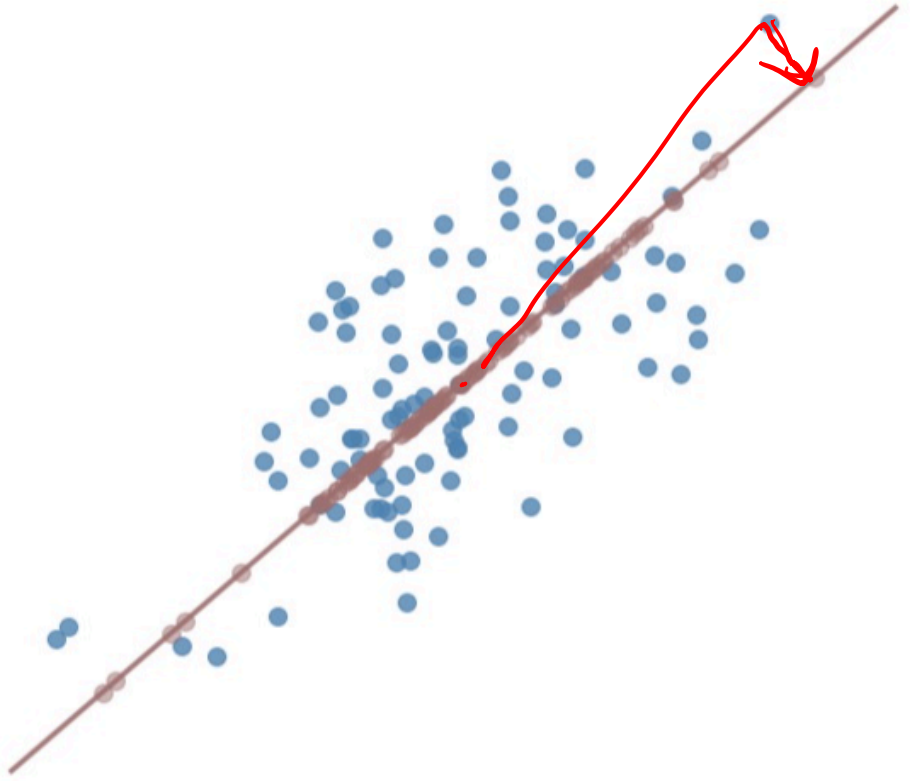
Principal Component Analysis – The How

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



Principal Component Analysis – The How

But how do we find ***w***?



Principal Component Analysis – The How

But how do we find $\boldsymbol{w}$?

Principal Component Analysis – The How

But how do we find $\mathbf{w}$?

$[X\mathbf{w}]_i$ are features in pc space. 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 – The How

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

Define Lagrangian

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

Principal Component Analysis – The How

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

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 – The How

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

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

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

Solution is eigenvector, and max variance is eigenvalue

Principal Component Analysis – The How

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

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

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

Solution is eigenvector, and max variance is eigenvalue

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

Principal Component Analysis – The How

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

Proof?

Orthogonal from the first one

unit vector

X

$X u_1$ ($D \times 1$)

$X' = (X - X u_1 u_1^T)$

$m \times D \quad m \times 1 \quad 1 \times D$

$C^{X'} = \frac{1}{m-1}(X^T - u_1 u_1^T X^T)(X - X u_1 u_1^T)$

Principal Component Analysis – The How

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

Proof?

To find all principal components, find all eigenpairs of C^X, and sort by decreasing size of eigenvalue

Notice:

$$\begin{aligned} C^X[\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_D] &= [C^X\mathbf{v}_1\ C^X\mathbf{v}_2\ \cdots\ C^X\mathbf{v}_D] \\ &= [\lambda_1\mathbf{v}_1\ \ \lambda_2\mathbf{v}_2\ \ \cdots\ \lambda_D\mathbf{v}_D] \end{aligned}$$

Principal Component Analysis – The How

Notice:

$$\begin{aligned} C^X[\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_D] &= [C^X\mathbf{v}_1\ C^X\mathbf{v}_2\ \cdots\ C^X\mathbf{v}_D] \\ &= [\lambda_1\mathbf{v}_1\ \ \lambda_2\mathbf{v}_2\ \ \cdots\ \lambda_D\mathbf{v}_D] \\ &= [\mathbf{v}_1\ \ \mathbf{v}_2\ \ \cdots\ \mathbf{v}_D]\ \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_D\} \end{aligned}$$

$$\text{Define}: V = [\mathbf{v}_1 \ldots \mathbf{v}_D], \quad \Lambda = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_D\}$$

Relationship becomes

$$C^X V = V\Lambda$$

$C^X = V\Lambda V^T$

This one relationship tells us literally everything

Principal Component Analysis – The How

Notice:

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

Principal Component Analysis – The How

Notice:

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

$$\frac{1}{m-1}(X - X\boldsymbol{v}_1\boldsymbol{v}_1^T)^T(X - X\boldsymbol{v}_1\boldsymbol{v}_1^T) = \frac{1}{m-1}(X^TX - (m-1)\lambda_1\boldsymbol{v}_1\boldsymbol{v}_1^T)$$
$$= \sum_{i=2}^{D} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T$$

The second principal component is thus $\boldsymbol{v}_2$

Principal Component Analysis – The How

Notice: Matrix V is orthogonal, so $V^T = V^{-1}$

$$C^X V = V\Lambda \quad \Leftrightarrow \quad V^T C^X V = \Lambda$$

How do we find the representation of X in the pc space?
Before we had $X\mathbf{w}$, now we have $Y = XV$. Plug in

$$\Lambda = \frac{1}{m-1} Y^T Y := C^Y = \frac{1}{m-1} V^T X^T X V$$
$$= V^T V \Lambda V^T V$$
$$= \Lambda$$

What do you notice?

Principal Component Analysis – The How

Notice: C^Y is the covariance matrix of the data in pc-space

- The variance of the principal component features are exactly the λ's (by construction)
- The principle component features are **uncorrelated**!
- The variance in the diagonal Λ tell you how important each feature is

AMAZING!

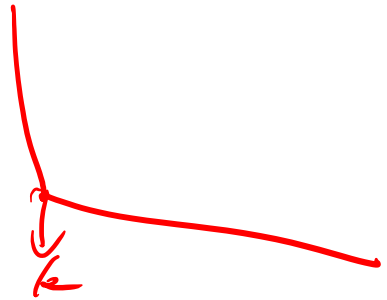
PCA – Dimensionality reduction

Questions:

- How do we reduce dimensionality?
- How much stuff should we keep?

PCA – Dimensionality reduction

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**

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

PCA – Dimensionality reduction

How do we reduce dimensionality?
Training example $\mathbf{x}_i$ turns into

$$\mathbf{y}_i = \mathbf{x}_i V = [\mathbf{x}_i \cdot \mathbf{v}_1 \quad \cdots \quad \mathbf{x}_i \cdot \mathbf{v}_k \quad \cdots \quad \mathbf{x} \cdot \mathbf{v}_D]$$

Principle vectors $\mathbf{v}_j$ are ordered by importance. Throw out the unimportant ones!
Define $V_k = V[:, :k]$, then $Y_k = XV_k$
Has dimension $m \times k$ (instead of $m \times D$)

Quiz

Which of the following statements are true?

A. Feature scaling is not useful for PCA, since the eigenvector calculation takes care of this automatically.

B. Given an input $x \in \mathbb{R}^n$, PCA compresses it to a lower-dimensional vector $z \in \mathbb{R}^k$.

C. PCA can be used only to reduce the dimensionality of data by 1 (such as 3D to 2D, or 2D to 1D).

D. If the input features are on very different scales, it is a good idea to perform feature scaling before applying PCA.

PCA - applications

Example 1: PC Regression

- Compute pc's, form $Y = XV$.
- Pass Y into learning algorithm instead of X

PCA - applications

Example 1: PC Regression

- Compute pc's, form $Y = XV$.
- Pass Y into learning algorithm instead of X

Pros:

- Y is lower dimensional - better VC bounds
- Potentially better because covariates are uncorrelated

PCA - applications

Example 1: PC Regression

- Compute pc's, form $Y = XV$.
- Pass Y into learning algorithm instead of X

$y w + b = 0$

Pros:

- Y is lower dimensional
- Potentially better because covariates are uncorrelated

Cons:

- Hard to interpret ?
- Possibly bad if chose k too small

PCA - applications

Example 2: Eigenfaces / Facial Recognition

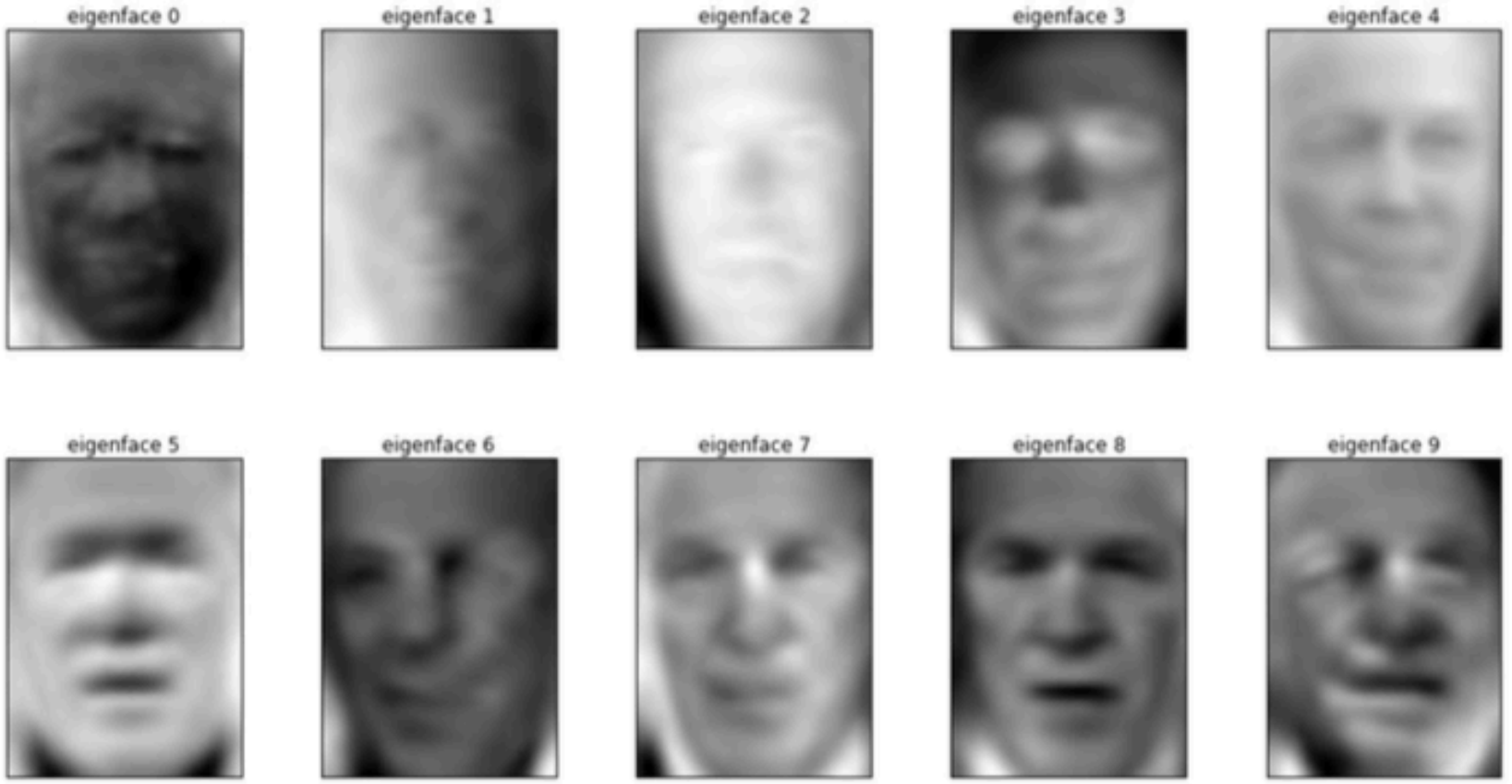
PCA - applications

Example 2: Eigenfaces / Facial Recognition

- "Labeled Faces in the Wild" dataset
- Roughly 1300 images of 7 difference people's faces in various orientation and lighting
- Images are 50x37 grayscale or 1850 features
- Perform PCA on mean-centered images
- Keep roughly 150 principal components
- Features reduced 1850 -> 150
- Use PCA coefficients in SVM to classify

PCA - applications

Example 2: Eigenfaces / Facial Recognition



Connecting PCA and SVD

- PCA:

$$\frac{1}{m-1}X^T X = V\Lambda V^T$$

- SVD:

$$X = U\Lambda V^T$$

SVD Applications

- SVD for word embeddings (Latent Semantic Analysis):

$$X = U\Lambda V^T$$

  - $X : D \times |V|$, document by vocabulary
  - $U : D \times k$, documents embeddings
  - $V : |V| \times k$, word embeddings
  - XV gives a continuous representation of a document

Wrap up

Dimensionality reduction can be a useful way to

- explore data
- visualize data
- represent data