Dimensionality Reduction

Dr. Nimisha Roy
Georgia Tech

These slides are based on slides from Mahdi Roozbahani and Rod Borela

Outline

- Overview
- Principle Component Analysis: Main Idea
- The PCA Algorithm
- PCA and SVD
- Summary

Motivating Example: Data Visualization

53 blood and urine samples (features) from 65 people

- Matrix format (65x53)

	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

Instances

Features

Difficult to see the correlations of different features

Motivating Example: Data Visualization

Is there a representation better than the coordinate axes?

Is it really necessary to show all the 53 dimensions?

- … what if there are strong correlations between the features?

How could we find
the *smallest* subspace of the 53-D space that
keeps the *most information* about the original data?

A Solution: **Dimension Reduction**

Another Example: Dimension Reduction for Text

Preprocessing for textual data is dimensionality reduction

1 Document → 1 datapoint

Goal

Spam

Not Spam

Vocabulary

freq

word$_1$

word$_2$

word$_{3\text{million}}$

all unique words

$X = \begin{bmatrix} \text{freq}_1 & \cdots & \cdots & \text{sender length} \end{bmatrix}_{n \times d}$

Remove stop words – a, an, the,

Lemmatization –

Bag-of-Words Representations

document 1

Machine learning concerns the construction and study of systems that can learn from data.

learn

document 2

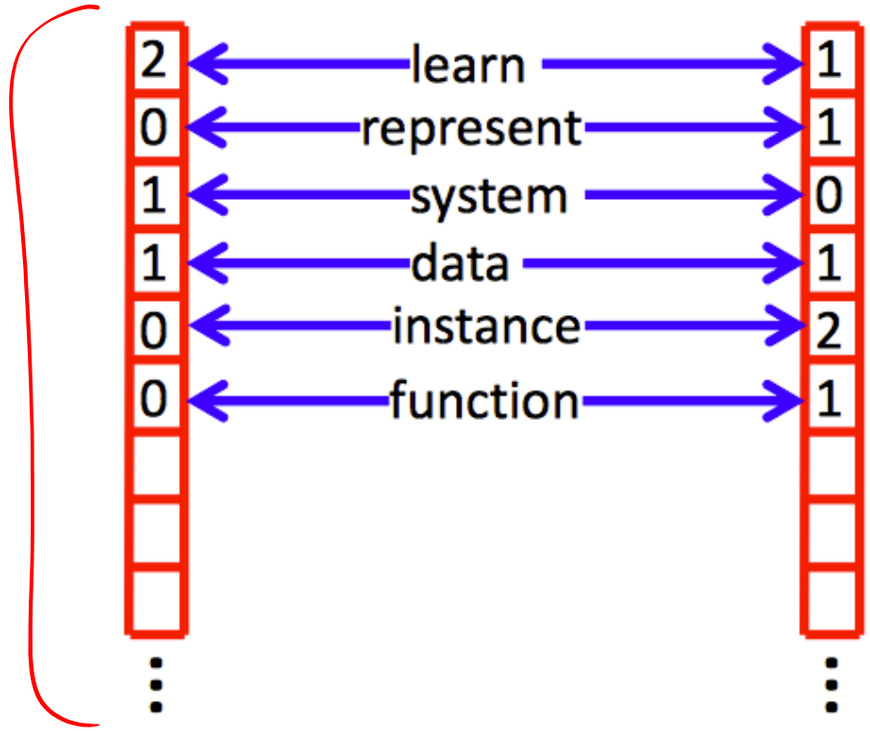
Representation of data instances and functions evaluated on these instances are part of all machine learning systems

Each document is an Instance

...

Each word is a feature

Bag of Words



Vector in R^d

Term-Document Data Matrix – Bag-of-words

database

	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

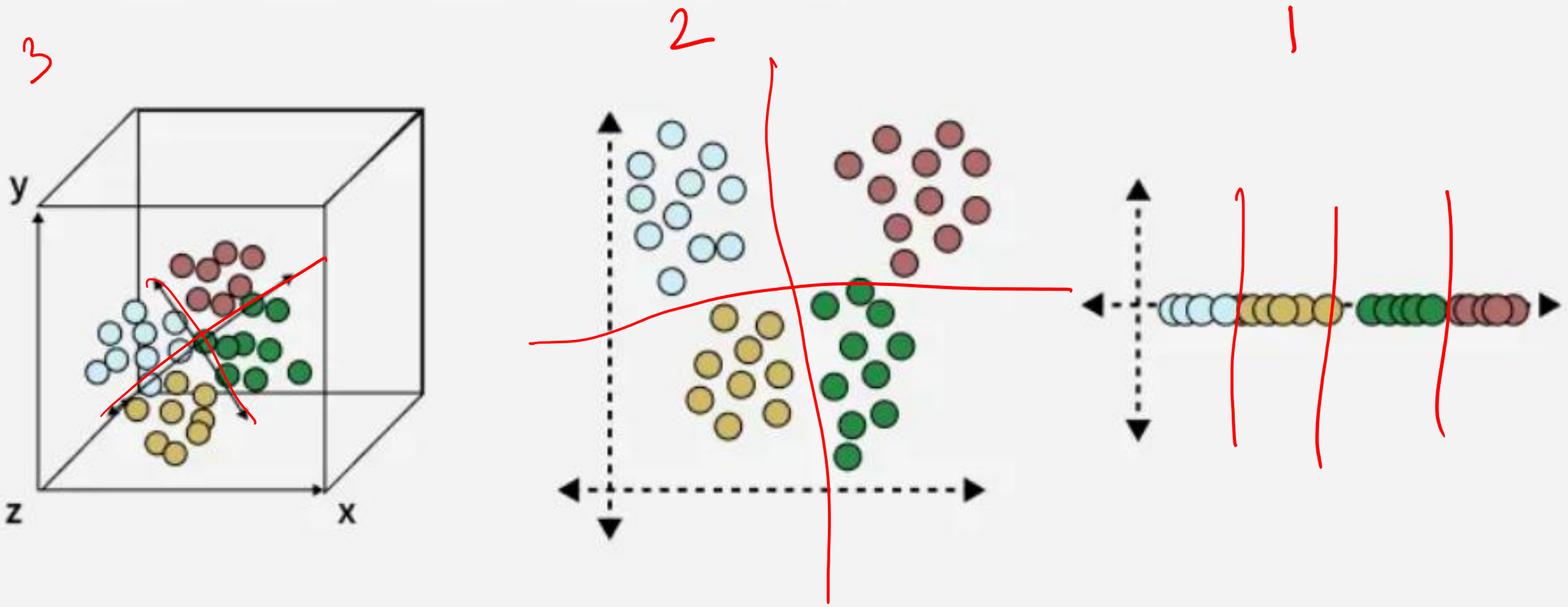
ML core algo

▪▪▪ Many more features

Solution:
Dimension Reduction

What is Dimensionality Reduction

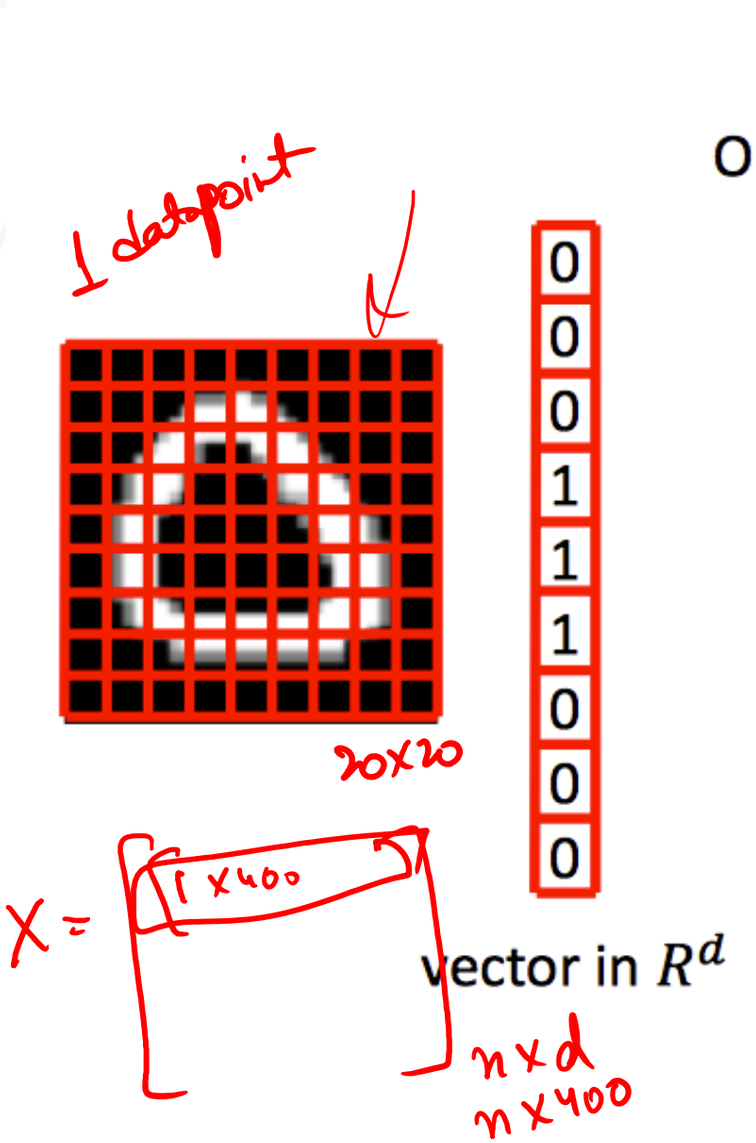
Dimensionality Reduction is the process of reducing the number of input variables (features) in a dataset while preserving as much important information as possible.



From geeksforgeeks

What is Dimension Reduction?

- The process of reducing the number of random variables under consideration
  - One can combine, transform or select variables
  - One can use linear or nonlinear operations



Applications of Dimension Reduction

- The dimension-reduced data can be used for
  - Visualizing, exploring and understanding the data
  - Aggregating weak signals in the data
  - Cleaning the data
  - Speeding up subsequent learning task
  - Building simpler model later

- Key questions of a dimensionality reduction algorithm
  - What is the criterion for carrying out the reduction process?
  - What are the algorithm steps?

Outline

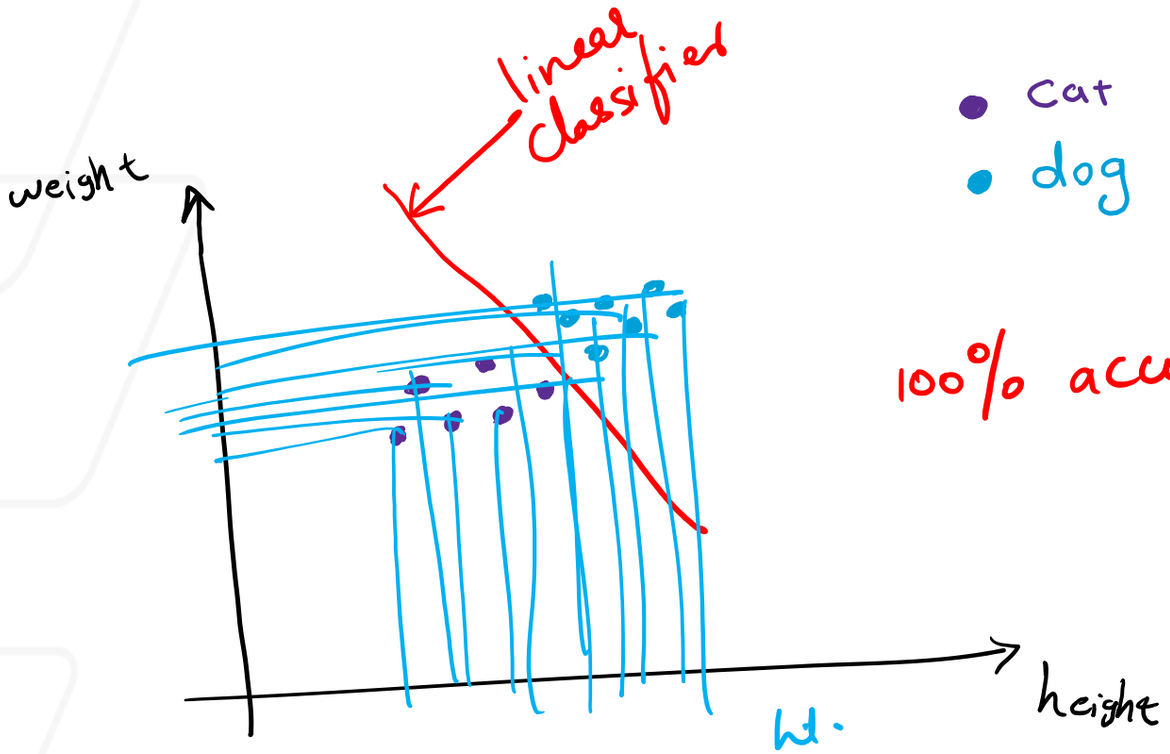
- Overview
- Principle Component Analysis: Main Idea
- The PCA Algorithm
- PCA and SVD
- Summary

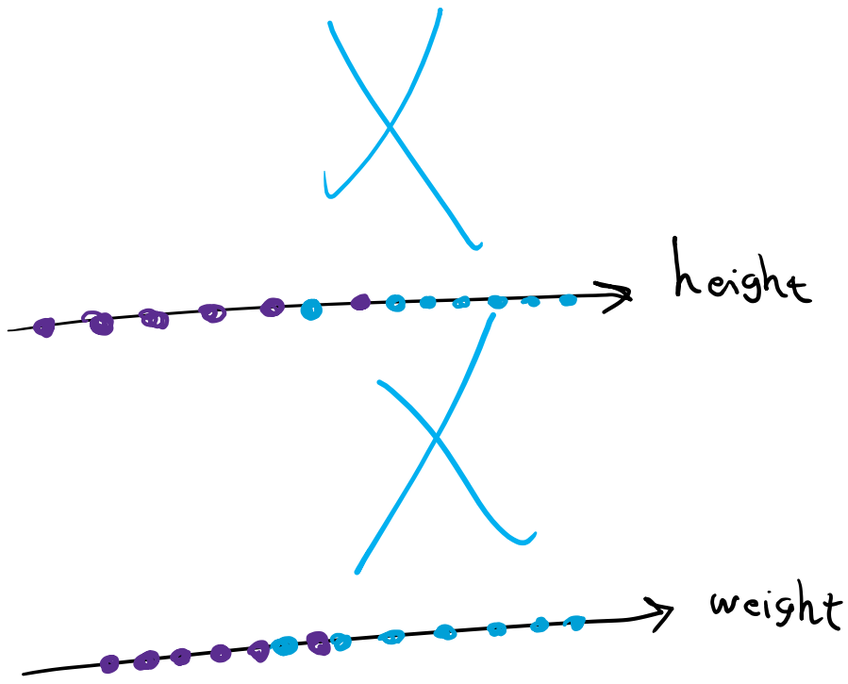
PCA: Dimension Reduction by Capturing Variation

- There are many criteria (geometric based, information theory based, etc.)

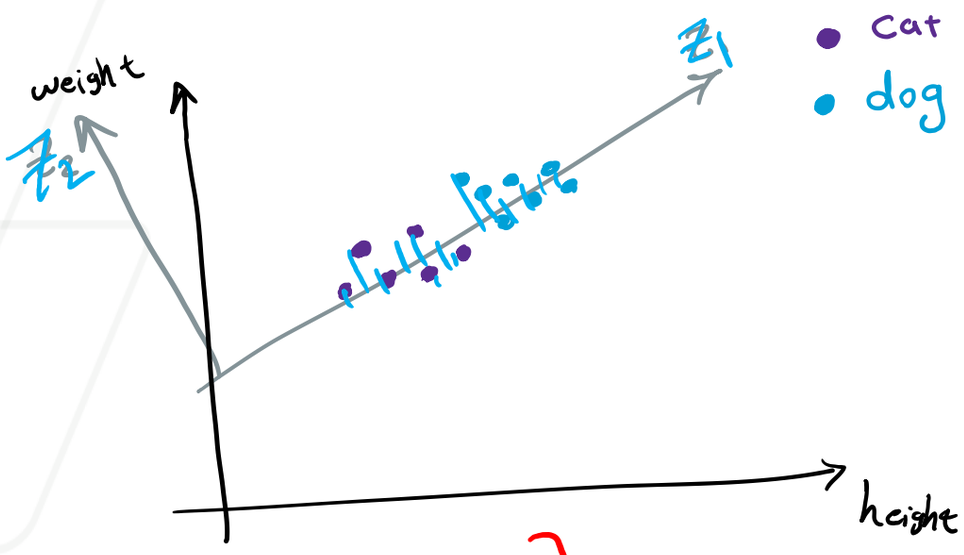
- One criterion: want to capture variation in data
  - variations are "signals" or information in the data
  - need to normalize each variables first

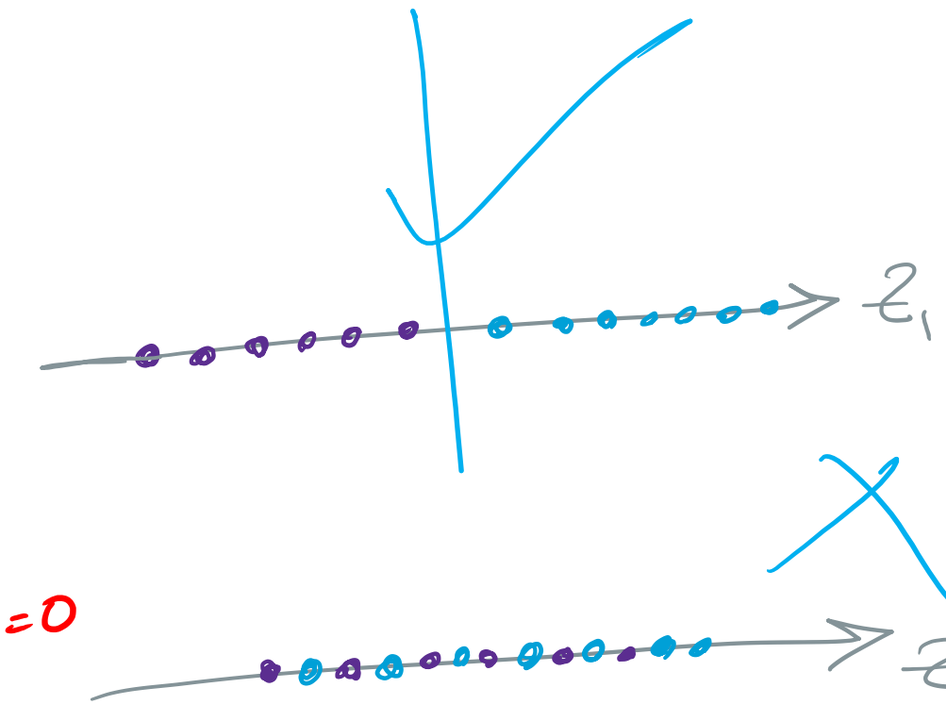
- In the process, also discover variables or dimensions highly correlated
  - represent highly related phenomena
  - combine them to form a stronger signal
  - lead to simpler presentation





Dropping / Truncating dimensions is not good





Normalization $\rightarrow [0,1]$

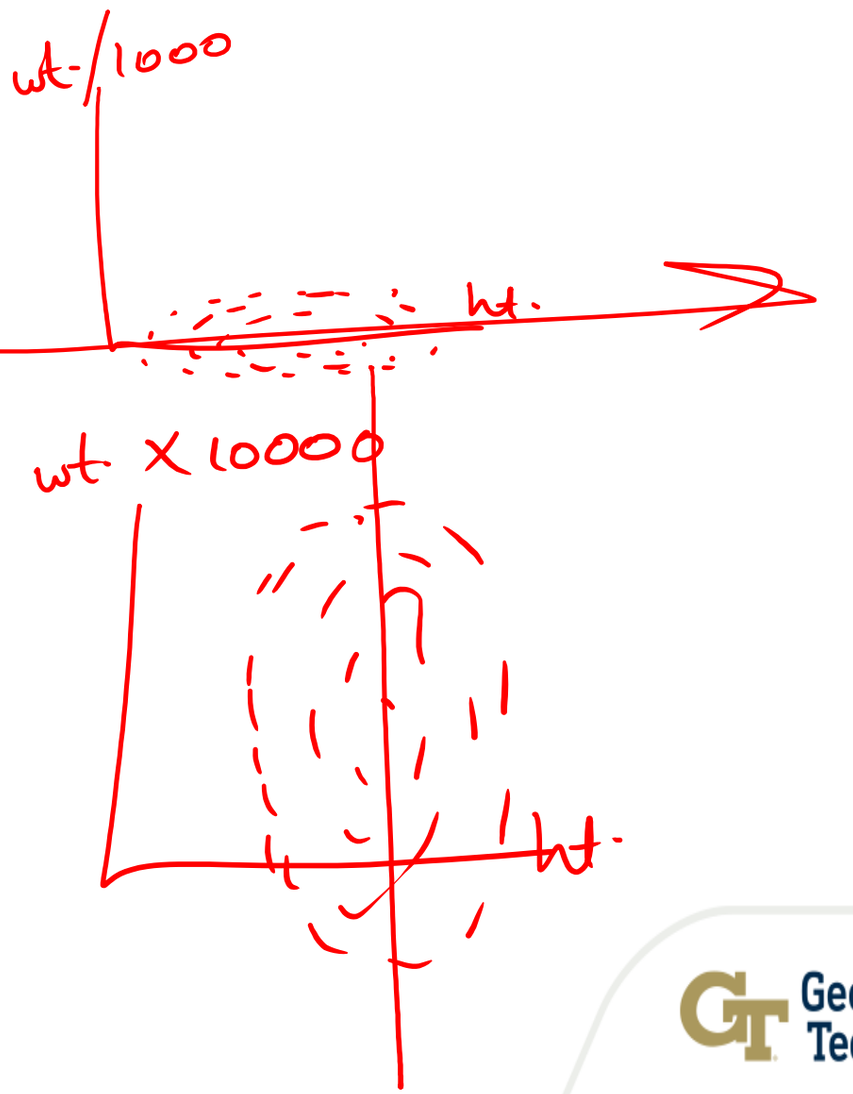
$$\frac{x - x_{min}}{x_{max} - x_{min}}$$

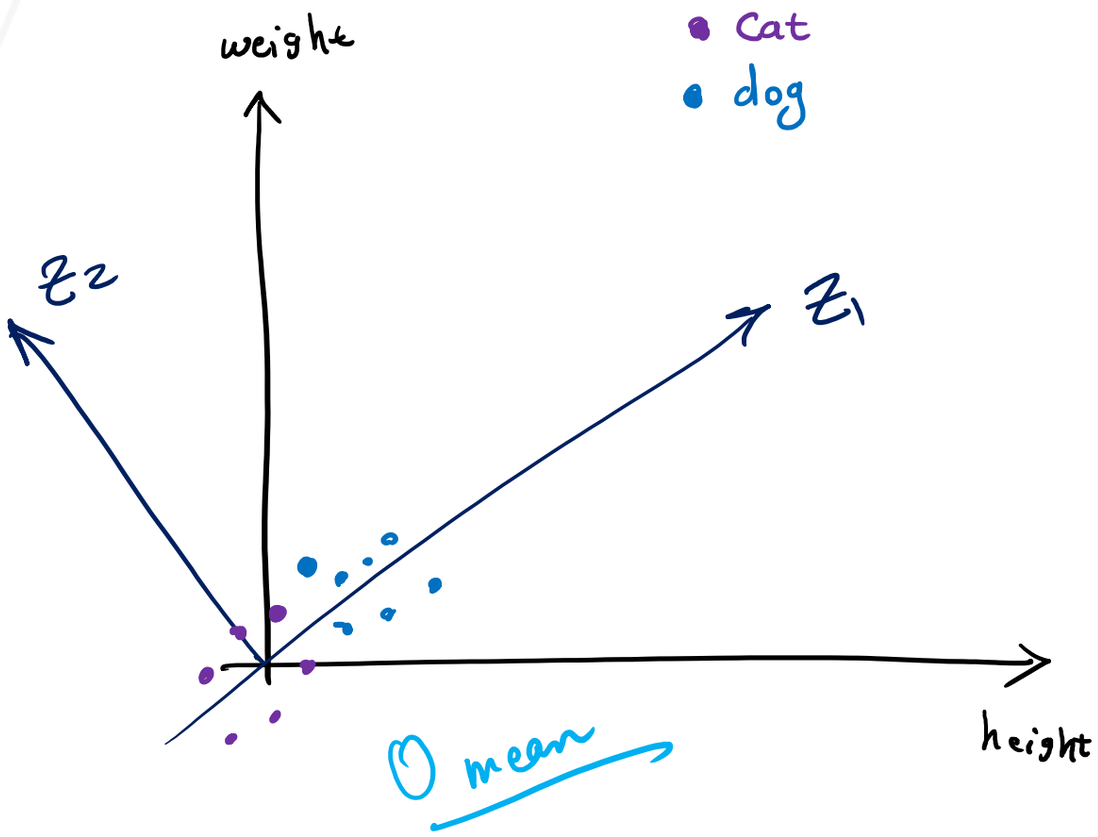
Standardization

$$\frac{X - \mu}{\sigma}$$

$\rightarrow$ mean = 0

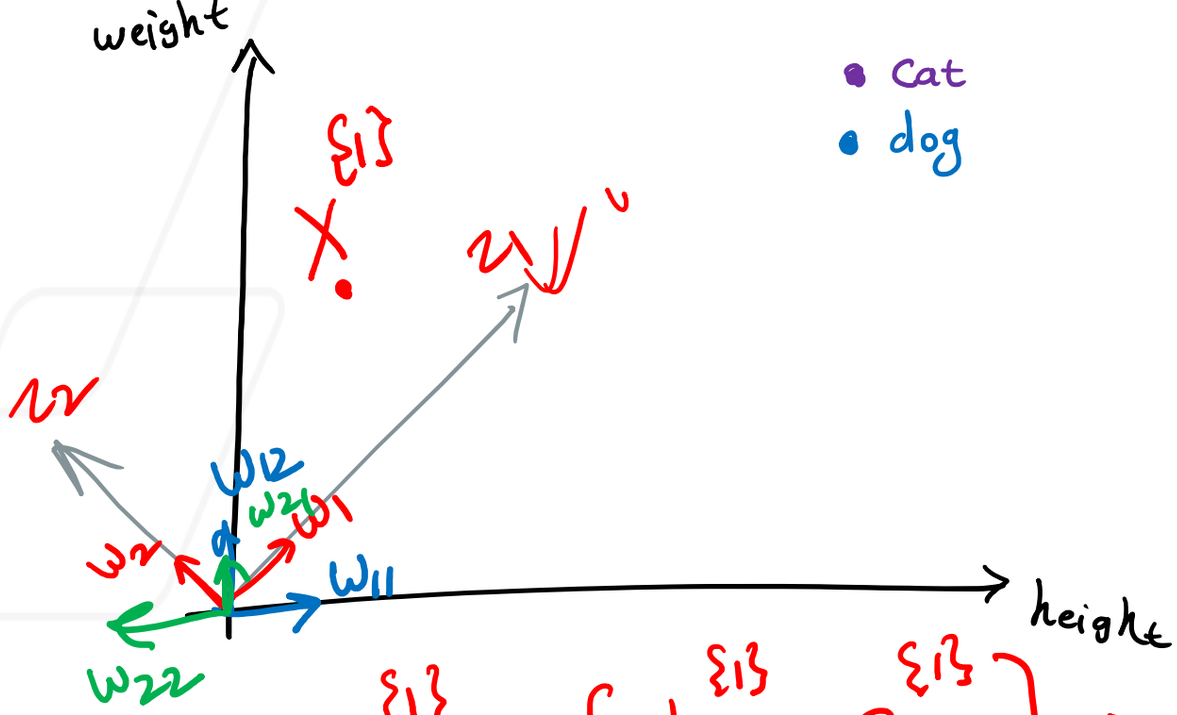
$\rightarrow$ Std. = 1





Georgia Tech

Projection



$$x \cdot \omega = x \cdot \frac{z}{\|z\|_2}$$

$$X^{\{i\}} = \begin{bmatrix} h^{\{i\}} & e^{\{i\}} \end{bmatrix} \longrightarrow \text{move to } z \text{ space}$$

unit vectors in z space is $\begin{bmatrix} \omega_1 & \omega_2 \end{bmatrix}$

in x space $= \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix}$

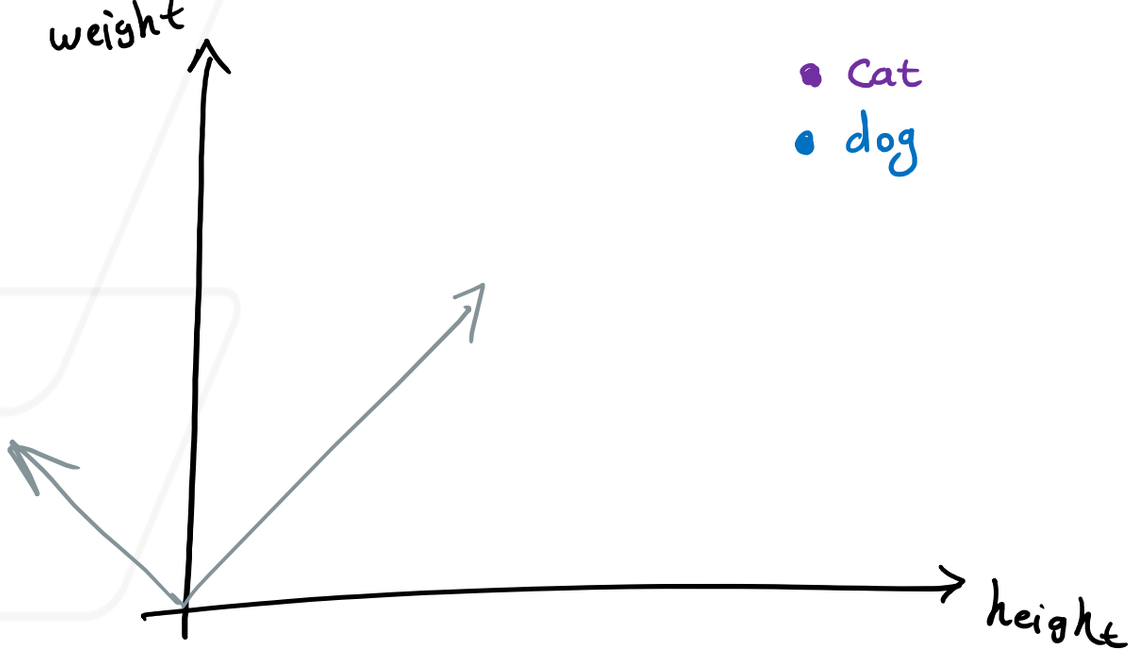
$$Z_1^{\{i\}} = X^{\{i\}} \cdot \omega_1 = \begin{bmatrix} h^{\{i\}} \\ e^{\{i\}} \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{12} \end{bmatrix} = h^{\{i\}} \omega_{11} + e^{\{i\}} \omega_{12}$$

$$Z_2^{\{i\}} = X^{\{i\}} \cdot \omega_2 = \begin{bmatrix} h^{\{i\}} \\ e^{\{i\}} \end{bmatrix} \begin{bmatrix} \omega_{21} & \omega_{22} \end{bmatrix} = h^{\{i\}} \omega_{21} + e^{\{i\}} \omega_{22}$$

$$\begin{bmatrix} h^{\{i\}} & e^{\{i\}} \end{bmatrix} \xrightarrow[\text{to } z \text{ space}]{\text{transformed}} \begin{bmatrix} h^{\{i\}} \omega_{11} + e^{\{i\}} \omega_{12} & h^{\{i\}} \omega_{21} + e^{\{i\}} \omega_{22} \end{bmatrix}$$

Georgia Tech

Standardization



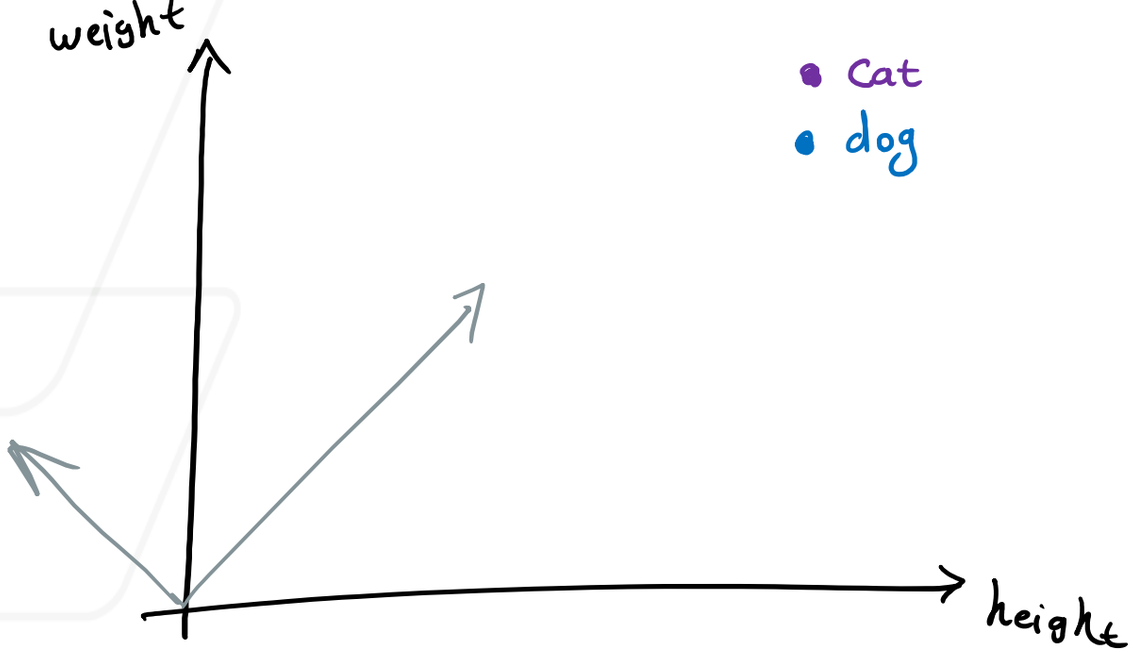
Standardize $\star$

$$X \longrightarrow \overline{X} = \frac{X - \mu}{\sigma}$$

$$\overline{X} \cdot w \longrightarrow$$

Truncate features

Objective Function



$$Var(x) = \frac{1}{N} \sum_i (x_i - \mu)^2$$

Maximize variance in the z space

$$\max. \quad Var(\omega) = \frac{1}{N} \sum_i (x_i \cdot \omega - \mu \cdot \omega)^2$$

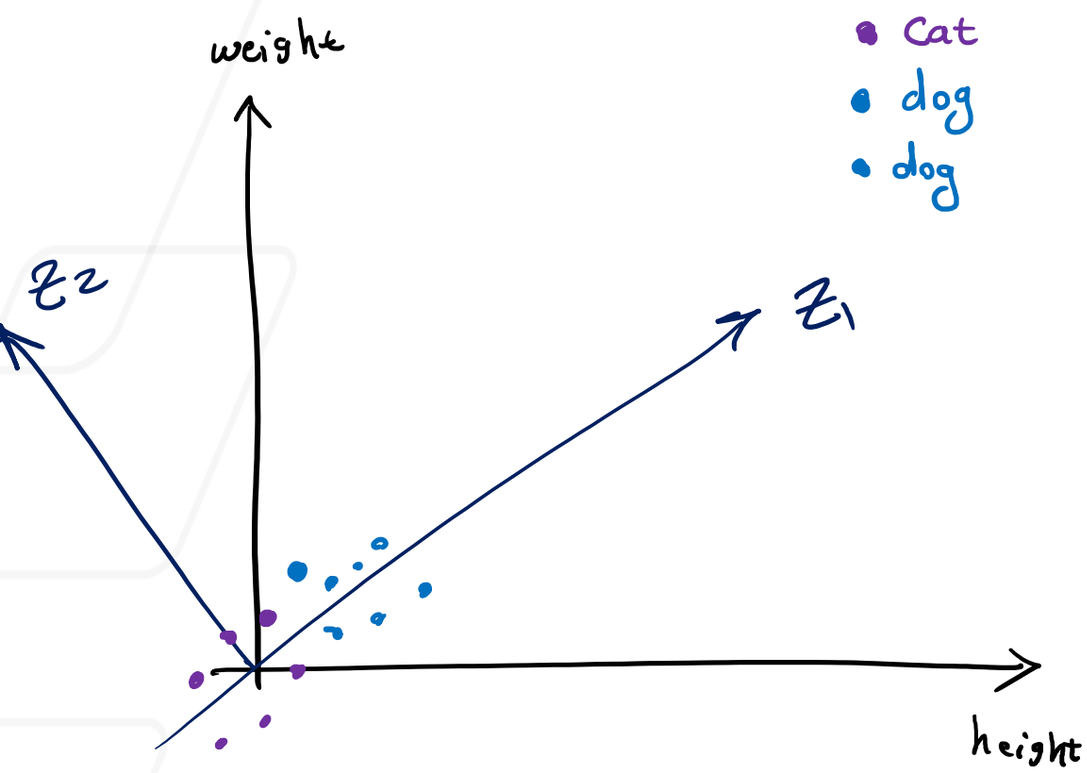
$$s.t. \quad \|\omega\|_2 = 1$$

Objective Function

$$L(\omega) = f(\omega) - \lambda g(\omega)$$

$\frac{\partial L}{\partial \omega} = 0 \longrightarrow$ find that ω

Georgia Tech



Were ht. and wt. features correlated?

Yes

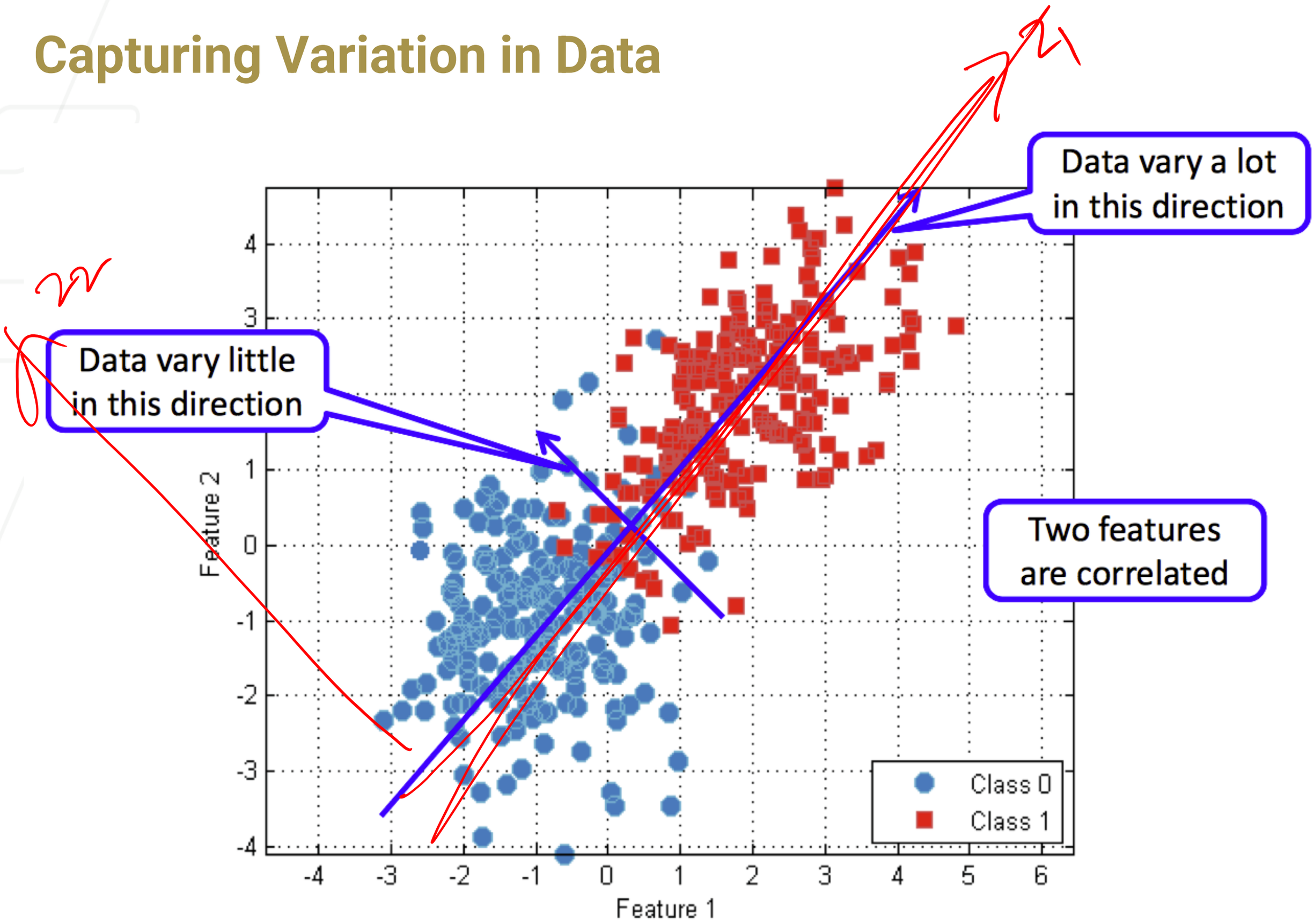
So what does PCA do?

- **Combines correlated features** into a single axis that captures their shared information.
- The new direction z is aggregation of existing features. Aggregation of weak signals.

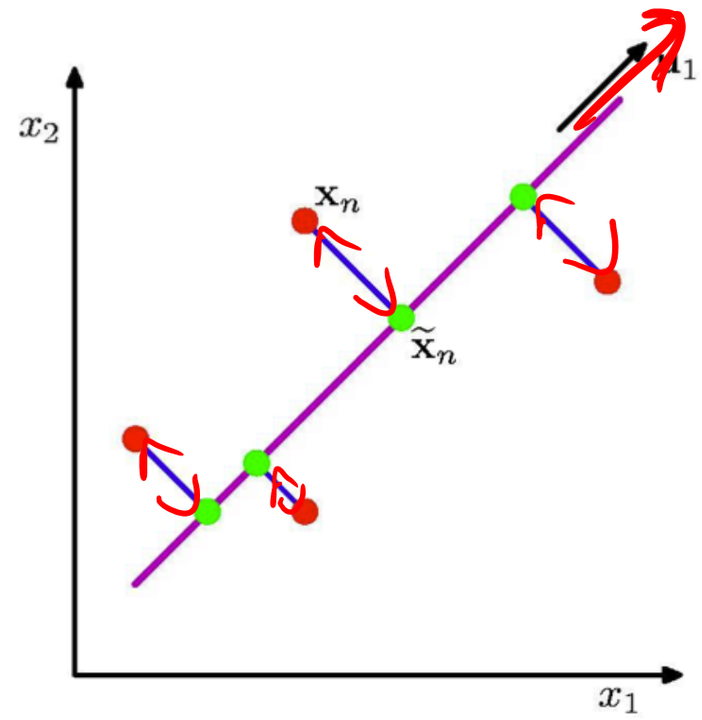
- **Reduces dimensionality** by keeping only the most informative directions — leading to simpler data representation while preserving most of the signal.

Georgia Tech

Capturing Variation in Data



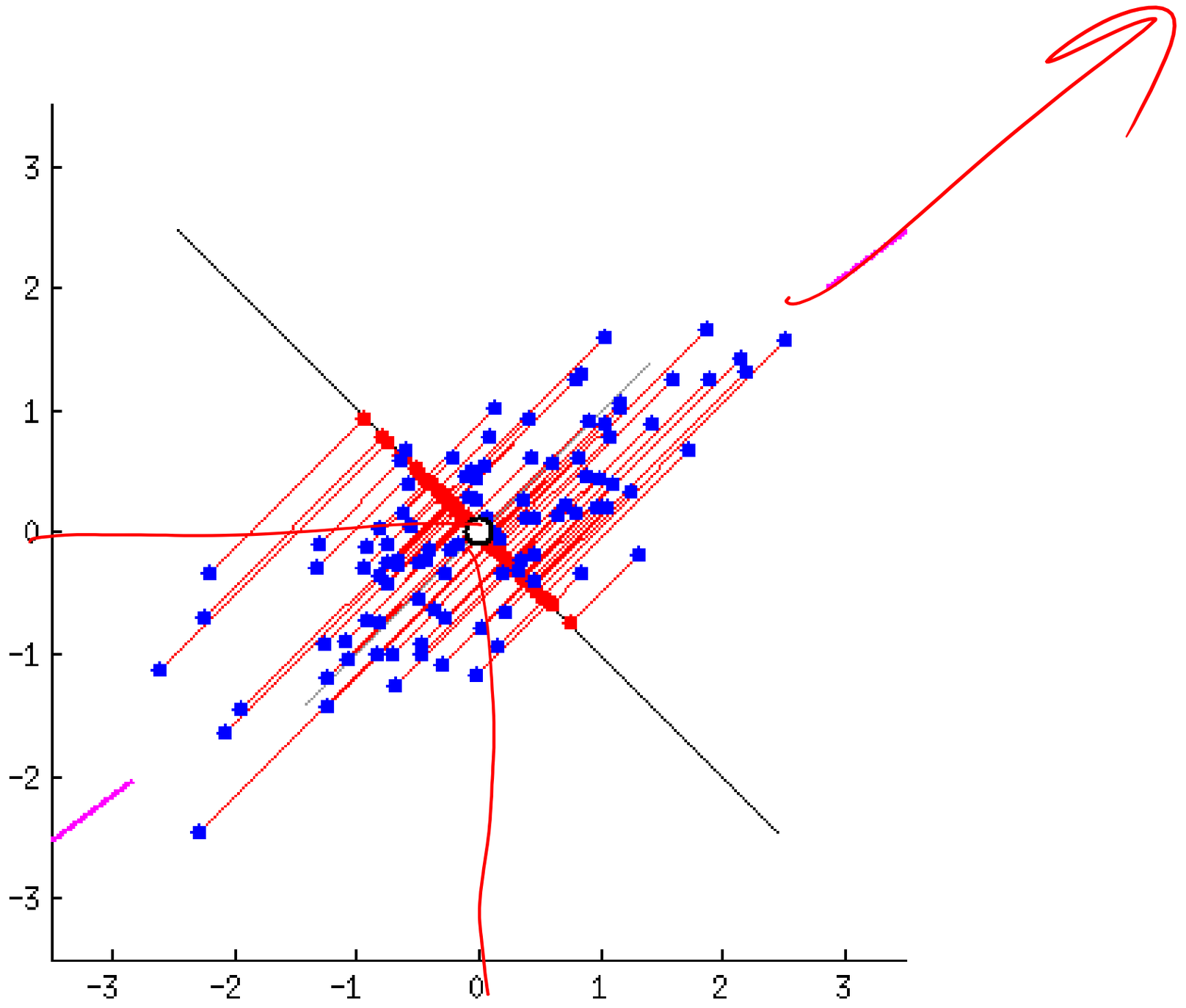
Two Equivalent Perspectives of PCA



PCA:

Orthogonal projection of the data onto a lower-dimension linear space that...

- maximizes variance of projected data (purple line)
- minimizes mean squared distance between
  - data point and
  - projections (sum of blue lines)


3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

Outline

- Overview
- Principle Component Analysis: Main Idea
- The PCA Algorithm
- PCA and SVD
- Summary

What is variance equation?

$$Var(x) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Formulating the Problem

- Given n data points, $\{x_1, x_2, \ldots, x_n\} \in R^d$ with their mean $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
- Find a direction $w \in R^d$ where

$$\|w\| = \sqrt{\sum_{j \in d} \omega_j^2} = 1$$

We constrain the norm of w to be equal to one to avoid having very large variance in each new dimension.

- Given n data points, $\{x_1, x_2, \ldots, x_n\} \in R^d$ with their mean μ

$$\|w\| = \sqrt{\sum_{j \in d} \omega_j^2} = 1 \qquad \mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

- Such that the variance (or variation) of the data along direction w is maximized

$$\max_{||w||=1} \frac{1}{n} \sum_{i=1}^{n} (x_i w - \mu w)^2$$

variance in new feature space

An Optimization Problem

- Manipulate the objective with linear algebra

$$\frac{1}{n}\sum_{i=1}^{n}(x_i w - \mu w)^2 \quad = \frac{1}{n}\sum_{i=1}^{n}((x_i - \mu)w)^2 \; =$$

$$= \frac{1}{n}\sum_{i=1}^{n}\underbrace{\big((x_i - \mu)}_{A}\underbrace{w\big)}_{B}^{T}((x_i - \mu)w) = \frac{1}{n}\sum_{i=1}^{n} w^T (x_i - \mu)^T (x_i - \mu)w$$

$$(AB)^T = B^T A^T$$

$$w^T\Big(\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^T (x_i - \mu)\Big)w \quad = w^T C w$$

Covariance matrix

Let's optimize it

$f(w) = w^T C w$

$g(w) = \|w\|_2 - 1 = w^T w - 1$

$L(w, \lambda) = w^T C w - \lambda (w^T w - 1)$

$\frac{\partial L}{\partial w} = 0 \Rightarrow 2Cw - 2\lambda w = 0$

$\Rightarrow Cw = \lambda w$

C: square, symmetrical

w: eigenvector of C

λ: eigenvalues of C

X: Data

v: eigenvector

λ: eigenvalues

$Xv = \lambda v$

$X \rightarrow$ find w or a different basis which captures max. variation in the data $\rightarrow$ find $C \rightarrow$ find eigenvectors of C

w is

Equivalence to The Eigenvalue Problem

Objective function: $\max_{||w||=1} w^T C w$

- Form lagrangian function of the optimization problem

$$L(w, \lambda) = w^T C w + \lambda(1 - w^T w)$$

If w is a maximum of the original optimization problem, then there exist a λ, where (w, λ) is a stationary point of $\mathrm{L}(\mathrm{w}, \lambda)$

Therefore:

$$\frac{\partial \mathrm{L}(\mathrm{w}, \lambda)}{\partial w} = 0 = 2Cw - 2\lambda w \Rightarrow \qquad Cw = \lambda w$$

Eigen-Value Problem

- Eigen-value problem

d: dimension

  - Given a symmetric matrix $C \in R^{d \times d}$

C is also a positive semidefinite matrix

  - Find a vector $w \in R^d$ and $\|w\| = 1$
  - Such that

$$Cw = \lambda w$$

- There will be multiple solution of $w_1, w_2, \ldots, w_d$ for its corresponding $\lambda_1, \lambda_2, \ldots, \lambda_d$

  - They are ortho-normal: $w_i^T w_i = 1$ $\quad$ $w_i^T w_j = 0$

Principal Direction of the Data

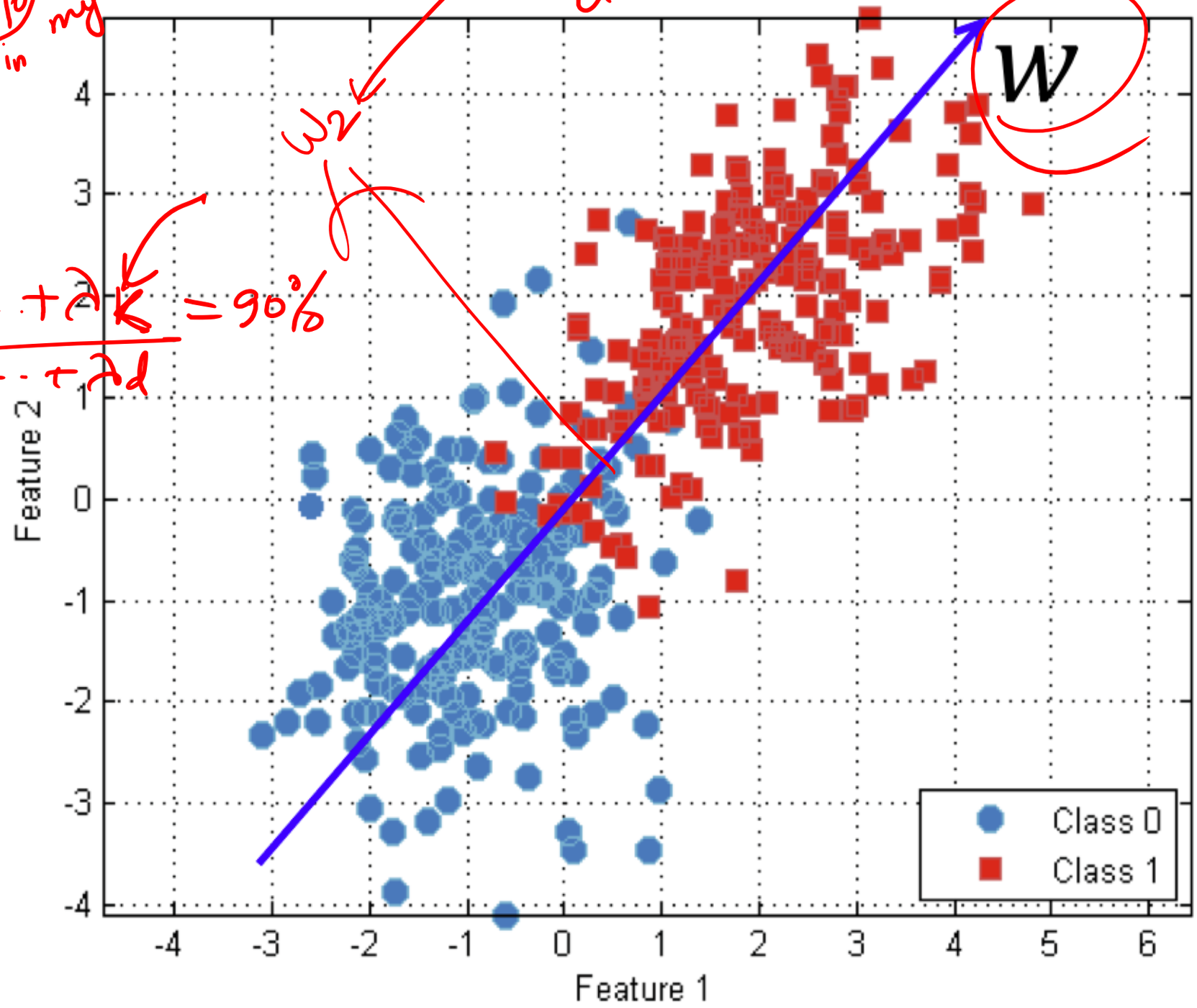
hyperparameter

I want to capture 90% variation in my data

principal direction for the rest of the subspace

principal direction

$$\frac{\lambda_1+\lambda_2+\cdots+\lambda_k}{\lambda_1+\lambda_2+\cdots+\lambda_d} = 90\%$$



Variance in the Principal Direction

- Principal direction w satisfies

$$Cw = \lambda w = w\lambda$$

eq from solving optimization

- Variance in principal direction is

$$w^T Cw$$

$$= w^T w\lambda$$

$$= \lambda$$

eigen-value

Variance in z direction

$\lambda_1 + \lambda_2 + \cdots + \lambda_d$ all variation in data

$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_K}{\lambda_1 + \lambda_2 + \cdots + \lambda_d} = 90\%$

Multiple Principal Directions

- Directions $w_1, w_2, \ldots$ which has
  - the largest variances
  - but are orthogonal to each other

- Take the eigenvectors $w_1, w_2, \ldots$ of C corresponding to
  - the largest eigenvalue λ_1,
  - the second largest eigenvalue λ_2
  - …

Relations Between Principal Components

PC direction → eigenvector direction of C

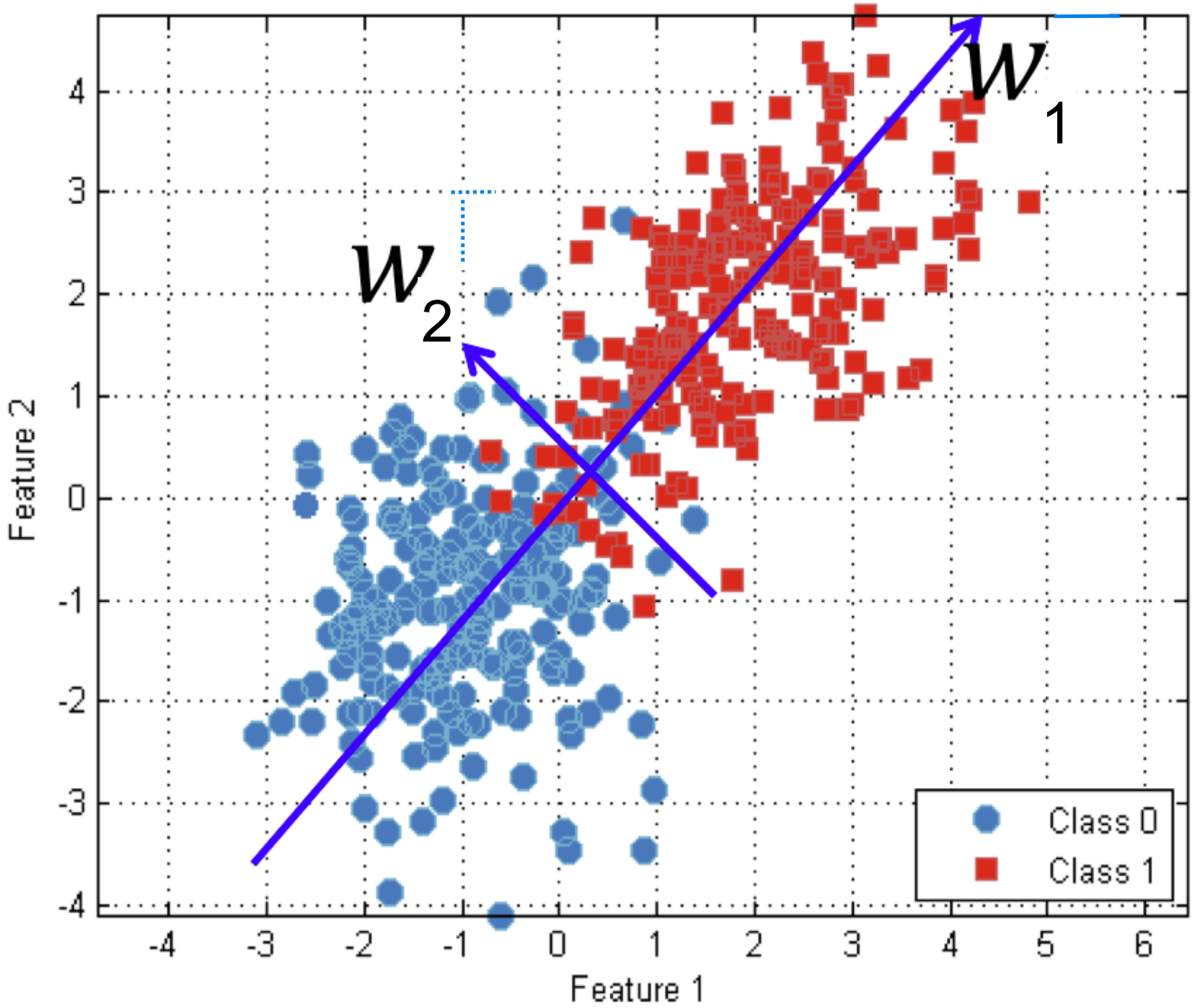
PC score → variation value

Principal component #1: points in the direction of the **largest variance**.

Each subsequent principal component

- is **orthogonal** to the previous ones, and
- points in the directions of the **largest variance of the residual subspace**

How to decide how many dimensions we want to reduce



The PCA Algorithm

- Given n data points, $\{x_1, x_2, \ldots, x_n\} \in R^d$ with mean

- Step 1: Estimate the mean and covariance matrix from data

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad and \quad C = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^T (x_i - \mu)$$

Principal directions

- Step 2: Take the eigenvectors $w_1, w_2, \ldots$ of C corresponding to the largest eigenvalue λ_1, the second largest eigenvalue λ_2 …

eigenvector of largest eigenvalue

- Step 3: Compute reduced representation

$$z_i = \left(\frac{(x_i - \mu_1)}{\sigma_1} w_1 \quad \frac{(x_i - \mu_2)}{\sigma_2} w_2 \quad \ldots \right)$$

$$z \Rightarrow \mathrm{n} \times k$$

$$k \ll d$$

Standardization

Outline

- Overview
- Principle Component Analysis: Main Idea
- The PCA Algorithm
- PCA and SVD
- Summary

Very computationally expensive

$$C = \frac{X^T X}{n}$$

eigen decomposition
→ square
→ orthonormal → Symmetric

X can be any shape.

(np.svd(X)) → numerically stable
→ Pretty fast

Singular Value Decomposition

$\bar{X}_{n\times d}$

n: instances
d: dimensions
X is a centered matrix

$\bar{X} = U\Sigma V^T$

$U_{n\times n} \rightarrow unitary\ matrix \rightarrow U \times U^T = I$

$\Sigma_{n\times d} \rightarrow diagonal\ matrix$

$V_{d\times d} \rightarrow unitary\ matrix \rightarrow V \times V^T = I$

Principle direction

$$X = \begin{bmatrix} u_{1\times1} & \cdots & \cdots & \cdots & u_{1\times n} \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ u_{1\times1} & \cdots & \cdots & \cdots & u_{n\times n} \end{bmatrix} \times \begin{bmatrix} \sum_{1\times1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sum_{d\times d} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} v_{1\times1} & \cdots & \cdots & \cdots & v_{1\times d} \\ \vdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ v_{d\times1} & \cdots & \cdots & \cdots & v_{d\times d} \end{bmatrix}$$

U $n \times m$ Σ $n \times d$ $d < n$ V^T $d \times d$

Matrix compression: K dimensions out of d

According to PCA ➔ $Cw = \lambda w = w\lambda$

Centering X

$$\text{Covariance } C_{d\times d} = \frac{1}{n}\sum_{i=1}^{n}\left(x^i - \mu\right)^T\left(x^i - \mu\right) = \frac{\bar{X}^T\bar{X}}{n}$$

$$\bar{X} = U\Sigma V^T$$

$$C = \frac{\bar{X}^T\bar{X}}{n}$$

$$C = \frac{V\Sigma^T U^T U\Sigma V^T}{n} = \frac{V\Sigma^2 V^T}{n}$$

$$C = \frac{V\Sigma^2 V^T}{n} = V\frac{\Sigma^2}{n}V^T$$

$CV = V\frac{\Sigma^2}{n}$ — ω — eigenvalues

$$CV = V\frac{\Sigma^2}{n}V^TV = V\frac{\Sigma^2}{n}$$

According to Eigen-decomposition definition ➔ $CV = V\Lambda$

V is the eigen vectors of covariance (Principal directions)

$\lambda_i = \frac{\sigma_i^2}{n}$ ➔ The eigenvalues of covariance matrix

Let's project the data (X) on principal directions:

$$\bar{X}V = U\Sigma V^TV = U\Sigma$$

$\bar{X}\boldsymbol{V}$ is linear combination of the original data (x-space) features

Projection of one instance (x) on the first principal direction using k dimensions

$$\mathrm{p}_1 = [u_{1\times1}\Sigma_{1\times1}\,, u_{1\times2}\Sigma_{2\times2}\,, \dots\,, u_{1\times k}\Sigma_{k\times k}]$$

$$\mathrm{p}_2 = [u_{2\times1}\Sigma_{1\times1}\,,\ u_{2\times2}\Sigma_{2\times2}\,, \dots\,, u_{2\times k}\Sigma_{k\times k}]$$

$U \Rightarrow n \times k$

$\Sigma \Rightarrow k \times k$

Upper left corner

Georgia Tech

Eigen values $\lambda = \frac{\Sigma\hat{}2}{n}$

Eigenvectors (principal directions) V

$$\bar{X} = U\Sigma V^T$$

$n = 50$

$k = 10$

$U = {}_{50\times 50}$

$\Sigma_{50\times d} = \Sigma_{10\times d}$

$U[:10, :10]$

10×10

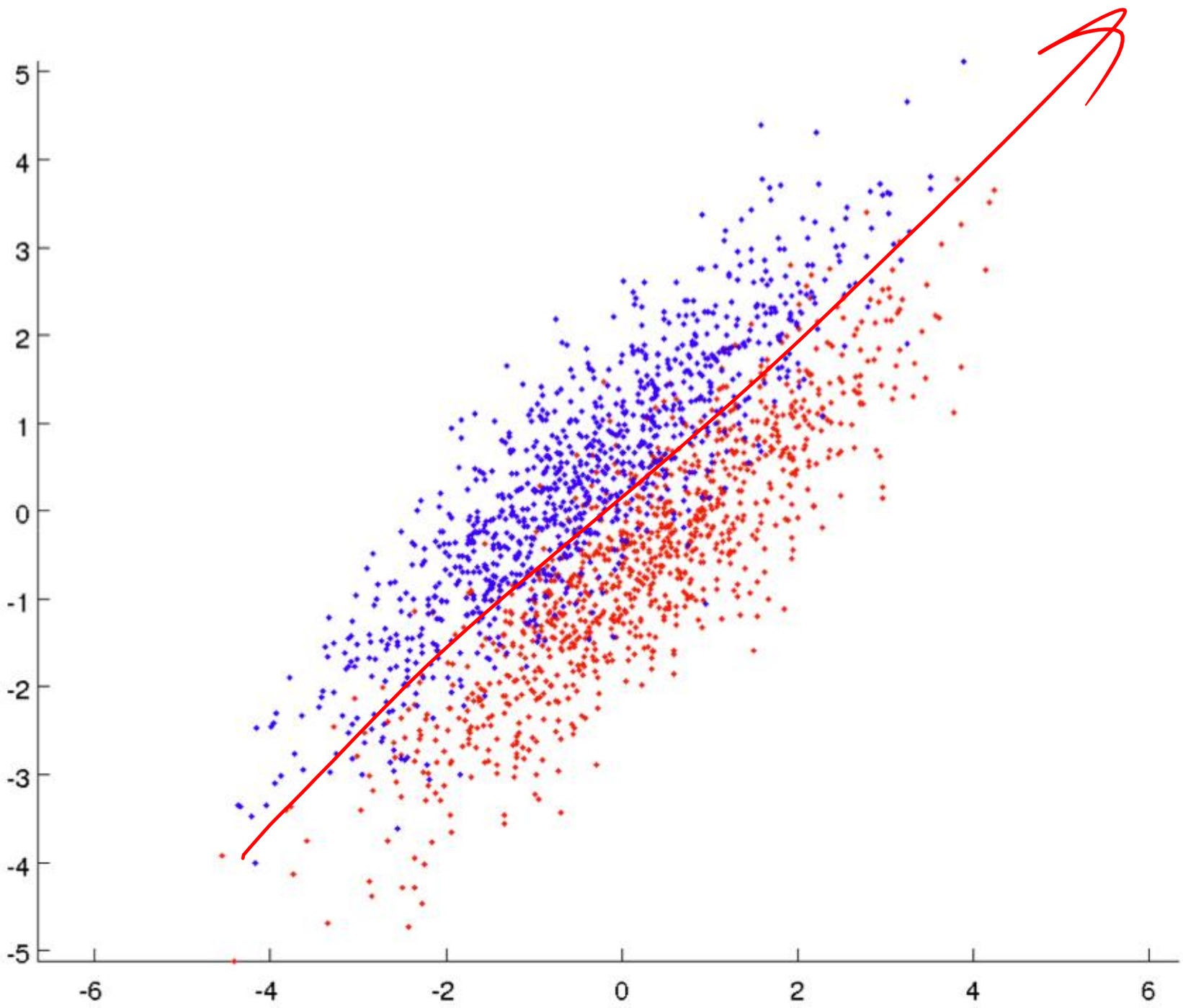
Principal components (Scores) or projections on principal directions

In fact, using the SVD to perform PCA makes much better sense numerically than forming the covariance matrix to begin with, since the formation of $X^T X$ can cause loss of precision.

Georgia Tech

Maybe

Are Principal Components Good for Classification?



Why PCA potentially works in classification?

The dimension with the largest variance corresponds to the dimension with the largest entropy and thus encodes the most information (Information Theory). The smallest eigenvectors will often simply represent noise components.

If a dimension has very little variance, the values are almost the same for all samples, so that dimension does not help distinguish between classes.

Georgia Tech

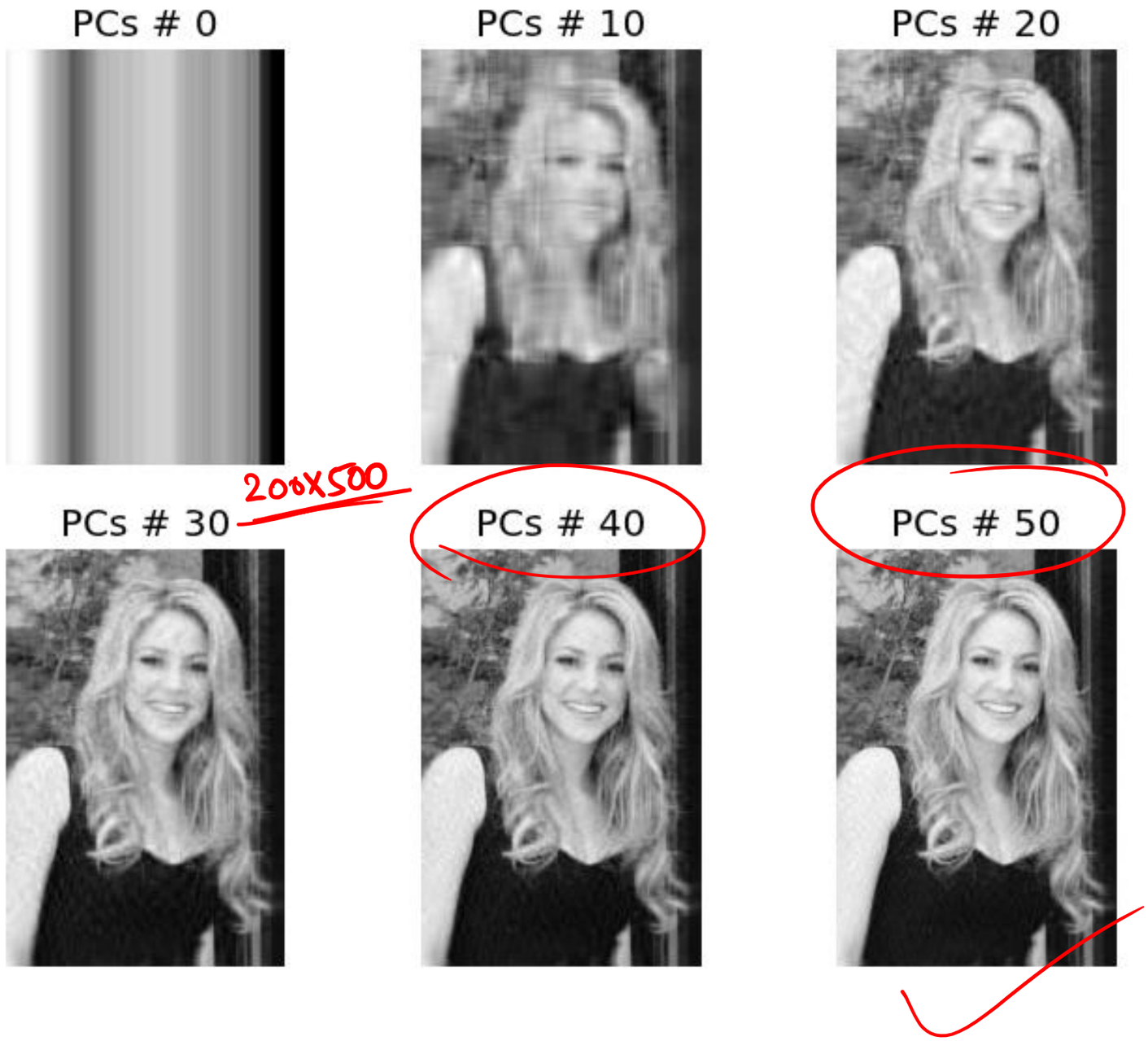
Outline

- Overview
- Principle Component Analysis: Main Idea
- The PCA Algorithm
- PCA and SVD
- Summary

Summary

- PCA
  - Finds orthonormal basis for data
  - Sorts dimensions in order of “importance”
  - Discard low significance dimensions
- Uses
  - Get concise low-dimensional representations
  - Remove noise
- Not magic
  - Doesn’t know class labels
  - Can only capture linear variations

Image compression using PCA



When does PCA work and for what?

Use Case	Why PCA Works	When to Use It
1. Aggregating correlated dimensions (feature reduction)	merges correlated features into a smaller number of *uncorrelated* components that explain most of the data's variance.	When features are linearly related (e.g., many sensors measuring similar signals).
2. Data visualization (2D or 3D)	finds the axes of greatest variance, which are usually the most informative for viewing high-dimensional data.	When you want to plot or explore high-dimensional data (e.g., projecting 100D embeddings to 3D).
3. Noise reduction / data compression	keeps only the top (k) components with the largest eigenvalues, effectively filtering out low-variance (noise) directions.	When you want to compress data while preserving the most structure — e.g., image compression, signal denoising.
4. Preprocessing before classification or clustering	PCA can remove redundancy and noise, making downstream algorithms (like k-NN, SVM, or k-means) faster and less sensitive to irrelevant features.	When input features are highly correlated or too many for your classifier to handle efficiently.

Georgia Tech

When PCA struggles or fails?

Issue	Why it matters
Nonlinear relationships	PCA only captures *linear* patterns. If your data lies on a nonlinear manifold (like a spiral or circle), PCA will fail to uncover it. (Techniques like *kernel PCA* or *t-SNE* work better.)
Outliers	PCA is *variance-based*, so outliers can distort the direction of principal components.
Different scales	If you don't normalize your features, those with larger scales dominate the components.
Interpretability	Principal components are linear combinations of features — they may not have a clear real-world meaning.
When all features are uncorrelated	If features are already independent, PCA won't reduce dimensionality — it can't compress noise.

Georgia Tech

Quick Knowledge Check

- In PCA, which mathematical property defines the new axes (principal components)? A. Eigenvectors of the data. B. Eigenvectors of the covariance matrix. C. Feature correlation coefficients
- Why is it necessary to normalize features before applying PCA? A. To make features linearly independent. B. To ensure features with larger scales do not dominate the variance. C. To remove correlation between variables
- What does a large eigenvalue represent in PCA? A. A noisy or redundant direction. B. A direction with low variance. C. A direction capturing significant variation (signal)
- Why is SVD preferred over directly computing eigenvectors of $X^T X$? A. SVD can be used only for square matrices. B. $X^T X$ can lose precision and is computationally expensive. C. SVD requires manual normalization
- Which of the following is **not** a common application of PCA? A. Image compression. B. Noise reduction. C. Feature extraction for visualization. D. Handling categorical variables directly
- When does PCA tend to fail? A. When features are correlated. B. When data has nonlinear relationships. C. When dimensionality is high
- What do the smallest eigenvalues in PCA typically represent?. A. The most informative directions. B. Noise or unimportant variation in data. C. The directions of maximum entropy

Georgia Tech