CS 540 Introduction to Artificial Intelligence

Linear Algebra and PCA

University of Wisconsin-Madison

Fall 2025 Sections 1 & 2

Announcements

- **HW 1 will be released tomorrow**:
  - Due **Friday Sep 19 at 11:59PM**
- TA discussion – review session today at 5:30 PM in Morgridge Hall 3610
- Class roadmap:

Linear Algebra & PCA
Logic
NLP

Mostly Foundations

Regularized Estimate

- Hyperparameter $\epsilon > 0$

$$\widehat{p_i} = \frac{n_i + \epsilon}{n + k\epsilon}$$

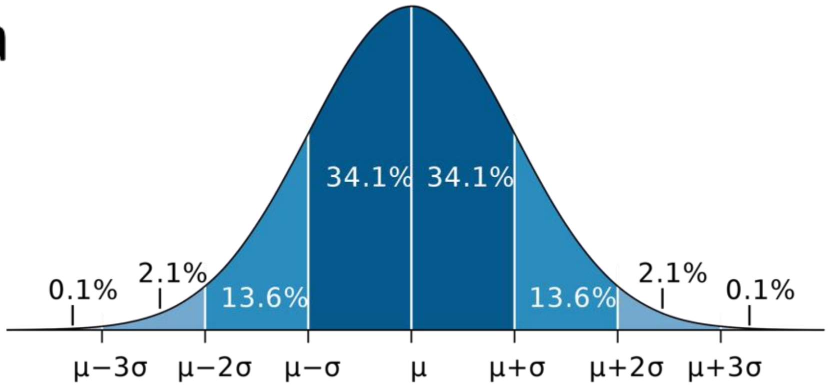
- Avoids zero when *n* is small
- Biased, but has smaller variance
- Equivalent to a specific Maximum A Posteriori (MAP) estimate, or smoothing

Estimating 1D Gaussian Parameters

- Gaussian (aka Normal) distribution $N(\mu, \sigma^2)$
  - True mean μ, true variance σ^2
- Observe *n* data points from this distribution

$$x_1, \ldots, x_n$$

- Estimate μ, σ^2 from this data



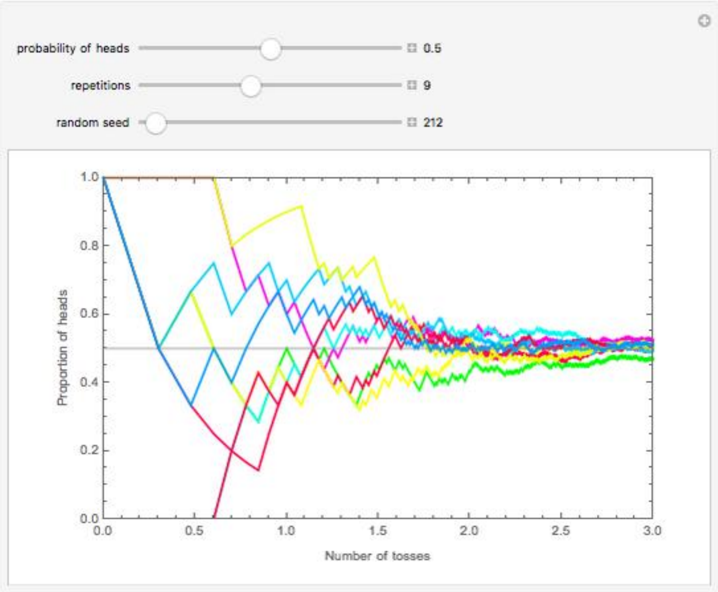
Wikipedia: Normal distribution

Estimating 1D Gaussian Parameters

- Mean estimate $\hat{\mu} = \dfrac{x_1 + \cdots + x_n}{n}$
- Variance estimates
  - Unbiased $s^2 = \dfrac{\sum_{i=1}^{n}(x_i - \hat{\mu})^2}{n-1}$
  - MLE $\hat{\sigma}^2 = \dfrac{\sum_{i=1}^{n}(x_i - \hat{\mu})^2}{n}$

Estimation Theory

- Is the sample mean a good estimate of the true mean?
  - Law of large numbers
  - Central limit theorems



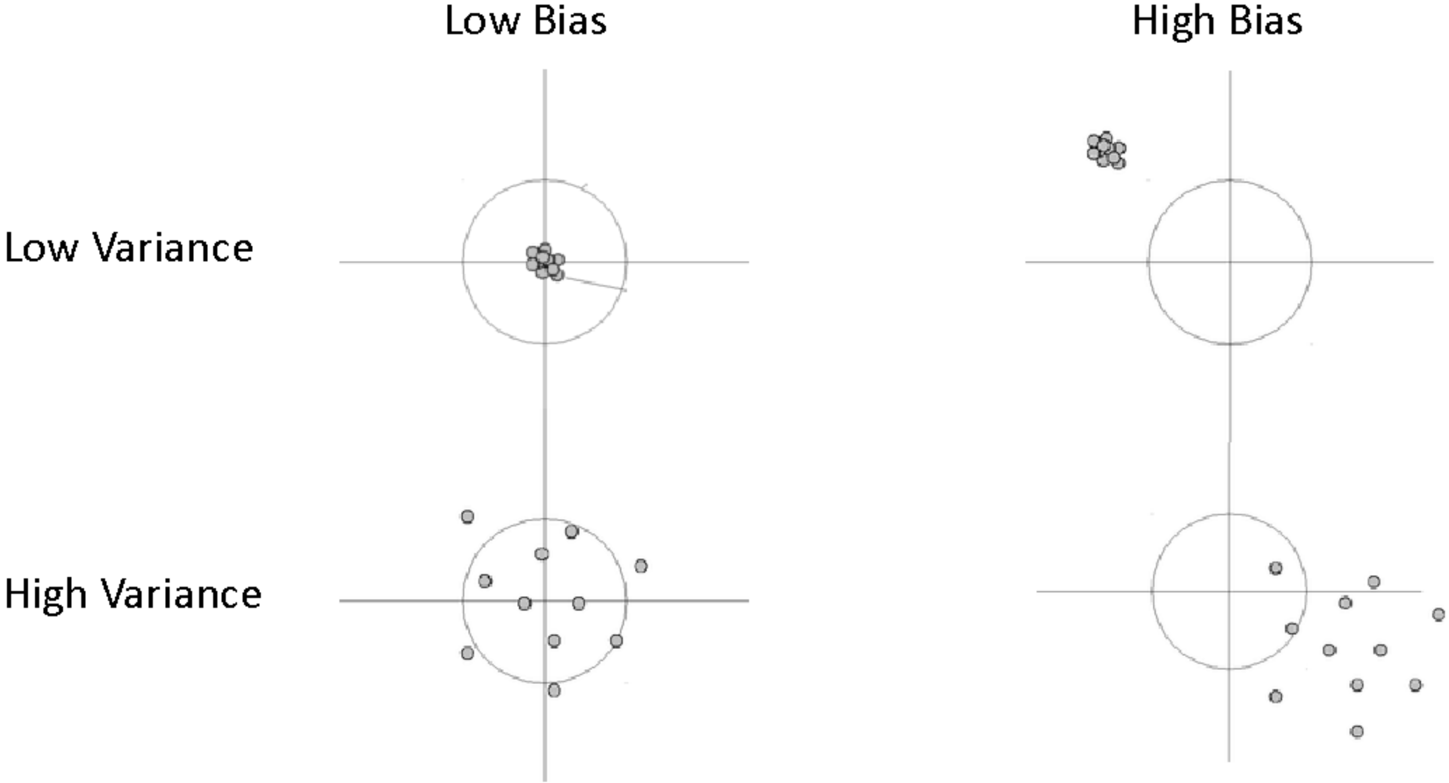
Wolfram Demo

Estimation Errors

- With finite samples, likely error in the estimate.
- Mean squared error
  - $\mathrm{MSE}[\hat{\theta}] = \mathbb{E}\,[(\hat{\theta} - \theta)^2]$
- Bias / Variance Decomposition
  - $\mathrm{MSE}[\hat{\theta}] = \mathbb{E}\left[(\hat{\theta} - E[\hat{\theta}])^2\right] + (\mathbb{E}[\hat{\theta}]\ - \theta)^2$

Variance Bias

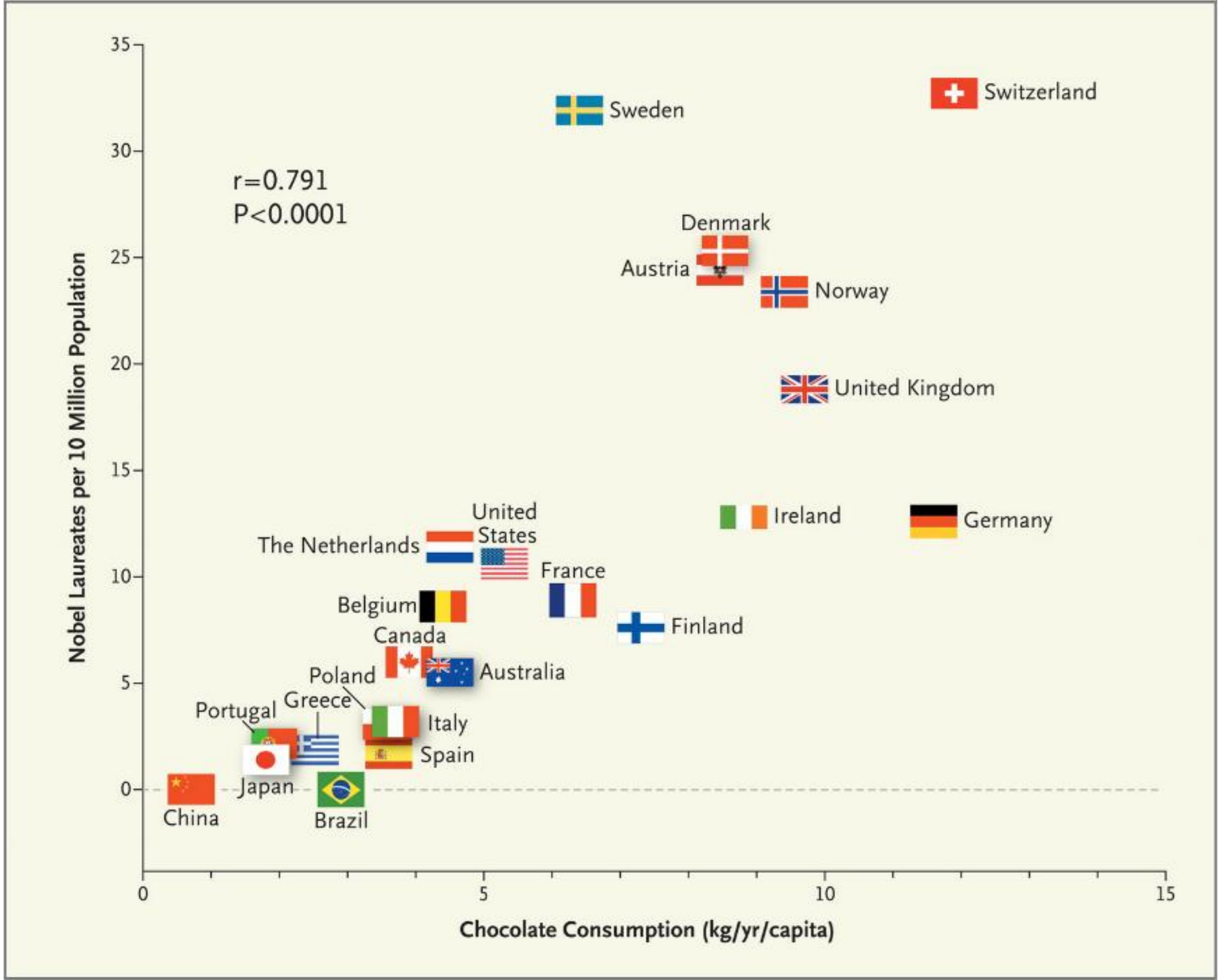
Bias / Variance



Wikipedia: Bias-variance tradeoff

Correlation vs. Causation

- Conditional probabilities only define correlation (aka association)
- $P(Y|X)$ “large” does not mean X causes Y
- Example: X=yellow finger, Y=lung cancer
- Common cause: smoking

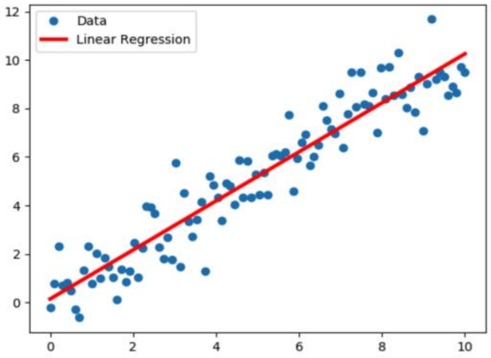


https://www.nejm.org/doi/full/10.1056/NEJMon1211064

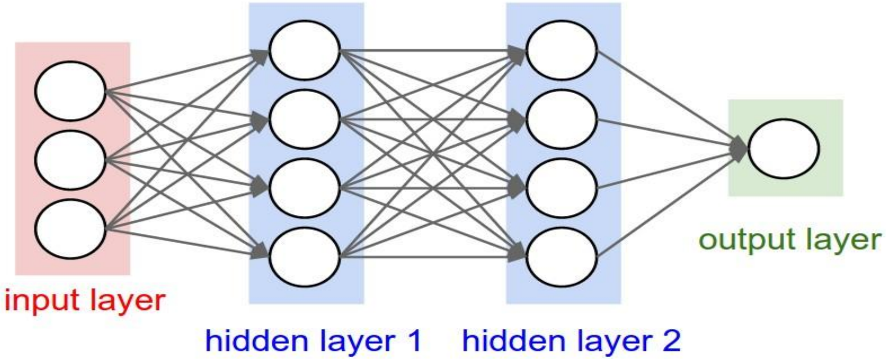
Linear Algebra

Linear Algebra: What is it good for?

- Study of Linear functions: simple, tractable
- In AI/ML: building blocks for **all models**
  - e.g., linear regression; part of neural networks



Hieu Tran

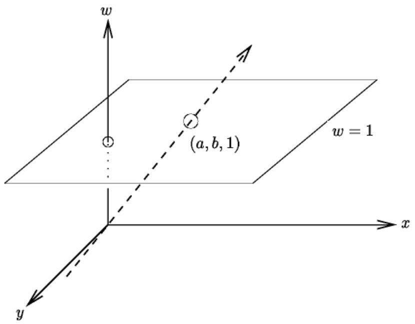


Stanford CS231n

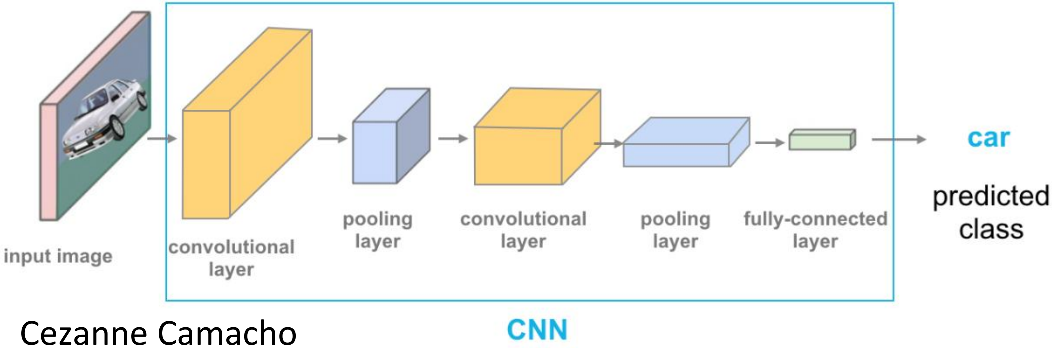
Basics: **Vectors**

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \in \mathbb{R}^5$$

- Many interpretations
  - List of values (represents information)
  - **Point in space**



- Dimension: number of values: $x \in \mathbb{R}^d$
- AI/ML: often use very high dimensions:
  - Ex: images!



Cezanne Camacho

Basics: **Matrices**

- Many interpretations
  - Table of values; list of vectors
  - Represent linear transformations
  - Apply to a vector, get another vector
- Dimensions: # rows × # columns, $A \in \mathbb{R}^{m \times n}$
  - indexing

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{33} & A_{33} \\ A_{41} & A_{43} & A_{43} \end{bmatrix}$$

Basics: **Transposition**

- Transposes: flip rows and columns
  - Vector: standard is a column. Transpose: row vector
  - Matrix: go from $m \times n$ to $n \times m$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad x^T = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}$$

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \quad A^T = \begin{bmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \\ A_{13} & A_{23} \end{bmatrix}$$

Matrix & Vector **Operations**

- **Vectors**
  - **Addition:** component-wise
    - Commutative: $x + y = y + x$
    - Associative: $(x + y) + z = x + (y + z)$

$$x + y = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ x_3 + y_3 \end{bmatrix}$$

  - **Scalar Multiplication**
    - Uniform stretch / scaling

$$cx = \begin{bmatrix} cx_1 \\ cx_2 \\ cx_3 \end{bmatrix}$$

Matrix & Vector **Operations**

- **Vector products**
  - **Inner product** (e.g., dot product)

$$< x, y >:= x^T y = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1 y_1 + x_2 y_2 + x_3 y_3$$

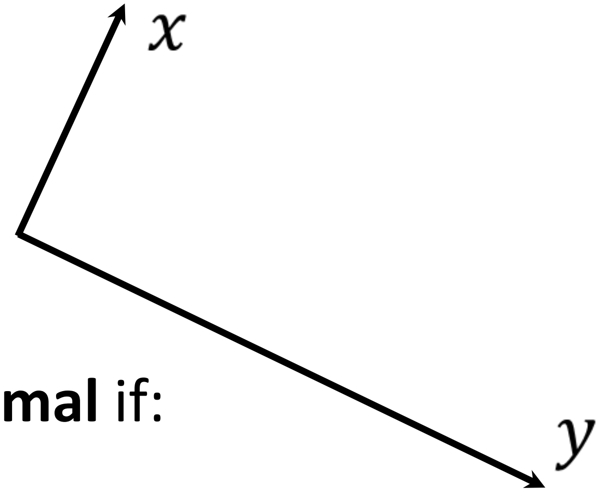
  - **Outer product**

$$xy^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 \end{bmatrix}$$

Matrix & Vector **Operations**

- x and y are **orthogonal** if $\langle x, y \rangle = 0$.
- Vector **norms**: "length"

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$



- A set of vectors $\{x_1, x_2, .. x_n\}$ is **orthonormal** if:
  – For all pairs x_i, xj we have $\langle x_i, xj \rangle = 0$
  – For all x_i, we have $\|x\|_2 = 1$

Matrix & Vector **Operations**

- **Matrices:**
  - **Addition:** Component-wise
  - Commutative, Associative

$$A + B = \begin{bmatrix} A_{11} + B_{11} & A_{12} + B_{12} \\ A_{21} + B_{21} & A_{22} + B_{22} \\ A_{31} + B_{31} & A_{32} + B_{32} \end{bmatrix}$$

  - **Scalar Multiplication**
  - “Stretching” the linear transformation

$$cA = \begin{bmatrix} cA_{11} & cA_{12} \\ cA_{21} & cA_{22} \\ cA_{31} & cA_{32} \end{bmatrix}$$

Matrix & Vector **Operations**

- **Matrix-Vector multiplication:**
  - **Linear transformation; plug in vector, get another vector**
  - Each entry in Ax is the inner product of a row of A with x

$$x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$$

$$Ax = \begin{bmatrix} \langle A_{1:}, x \rangle \\ \langle A_{2:}, x \rangle \\ \vdots \\ \langle A_{m:}, x \rangle \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n \end{bmatrix}$$

Matrix & Vector **Operations**

Ex: feedforward neural networks. Input *x*.

- Output of layer *k* is

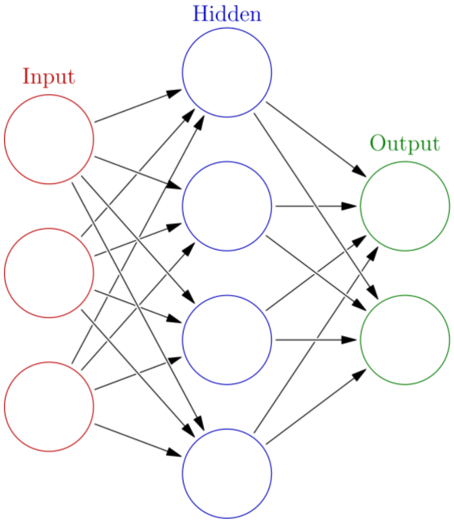
nonlinearity

$$f^{(k)}(x) = \sigma(W_k^T f^{(k-1)}(x))$$

Output of layer k: vector

Weight **matrix** for layer k:
Note: linear transformation!

Output of layer k-1: **vector**

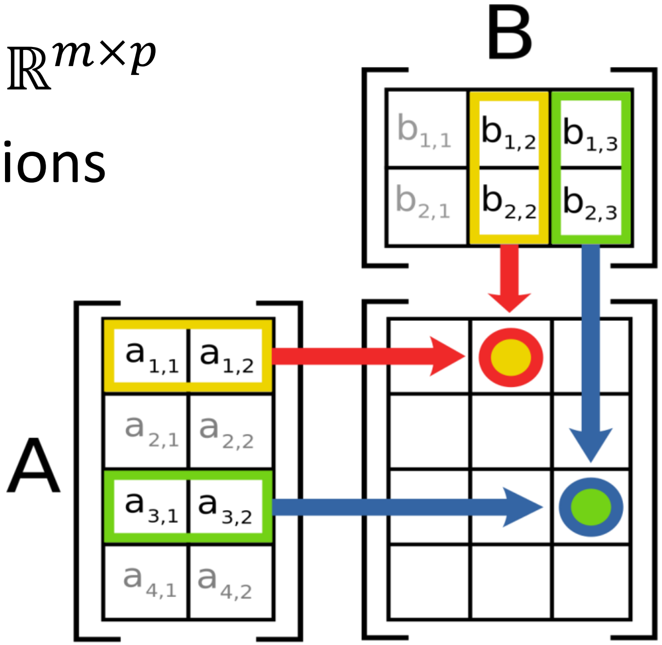


Wikipedia

Matrix & Vector **Operations**

- Matrix multiplication
  - $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times p}$, then $AB \in \mathbb{R}^{m \times p}$
  - "Composition" of linear transformations
  - Not commutative in general!

$$AB \neq BA$$



Wikipedia

Identity Matrix

- Like "1"
- Multiplying by it gets back the same matrix or vector
- Rows & columns are the "**standard basis vectors**" e_i

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

e_1 e_2 e_n

Break & Quiz

- **Q 1.1**: What is $\begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix}$?
- A. $[-1\ 1\ 1]^T$
- B. $[2\ 1\ 1]^T$
- C. $[1\ 3\ 1]^T$
- D. $[1.5\ 2\ 1]^T$

Break & Quiz

- **Q 1.1**: What is $\begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix}$?
- A. $[-1\ 1\ 1]^T$
- **B. $[2\ 1\ 1]^T$**
- C. $[1\ 3\ 1]^T$
- D. $[1.5\ 2\ 1]^T$

Break & Quiz

- **Q 1.1**: What is $\begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix}$?

- A. $[-1\ 1\ 1]^T$
- **B. $[2\ 1\ 1]^T$**
- C. $[1\ 3\ 1]^T$
- D. $[1.5\ 2\ 1]^T$

Check dimensions: answer must be 3 x 1 matrix (i.e., column vector).

$$\begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 * 1 + 1 * 2 \\ 0 * 3 + 1 * 1 \\ 0 * 1 + 1 * 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

Break & Quiz

- **Q 1.2**: Given matrices $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{d \times m}, C \in \mathbb{R}^{p \times n}$ What are the dimensions of BAC^T

- A. *n x p*
- B. *d x p*
- C. *d x n*
- D. Undefined

Break & Quiz

- **Q 1.2**: Given matrices $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{d \times m}, C \in \mathbb{R}^{p \times n}$

What are the dimensions of BAC^T

- A. *n x p*
- **B. *d x p***
- C. *d x n*
- D. Undefined

Break & Quiz

- **Q 1.2**: Given matrices $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{d \times m}, C \in \mathbb{R}^{p \times n}$

What are the dimensions of BAC^T

- A. *n x p*
- **B. *d x p***
- C. *d x n*
- D. Undefined

To rule out (D), check that for each pair of adjacent matrices XY, the # of columns of X = # of rows of Y

Then, B has d rows so solution must have d rows. C^T has p columns so solution has p columns.

Break & Quiz

- **Q 1.3**: A and B are matrices, neither of which is the identity. Is $AB = BA$?

- A. Never
- B. Always
- C. Sometimes

Break & Quiz

- **Q 1.3**: A and B are matrices, neither of which is the identity. Is $AB = BA$?

- A. Never
- B. Always
- **C. Sometimes**

Break & Quiz

- **Q 1.3**: A and B are matrices, neither of which is the identity. Is $AB = BA$?

- A. Never
- B. Always
- **C. Sometimes**

Matrix multiplication is not necessarily commutative.

Matrix Inverses

- If for *A* there is a *B* such that $AB = BA = I$
  - Then *A* is invertible/nonsingular, B is its inverse
  - Some matrices are **not** invertible!

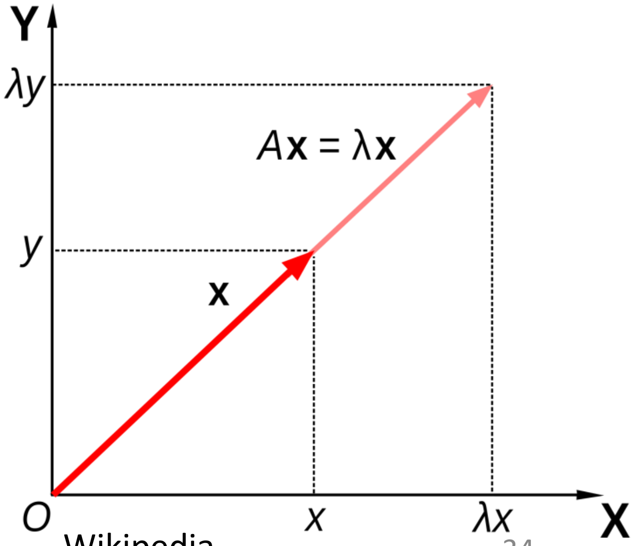
  - Usual notation: A^{-1}

$$\begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} \times \begin{bmatrix} 3 & -1 \\ -2 & 1 \end{bmatrix} = I$$

Eigenvalues & Eigenvectors

- For a square matrix *A*, solutions to $Av = \lambda v$
  - *v* (nonzero) is a vector: **eigenvector**
  - λ is a scalar: **eigenvalue**

  - Intuition: A is a linear transformation;
  - Can stretch/rotate vectors;
  - E-vectors: only stretched (by e-vals)

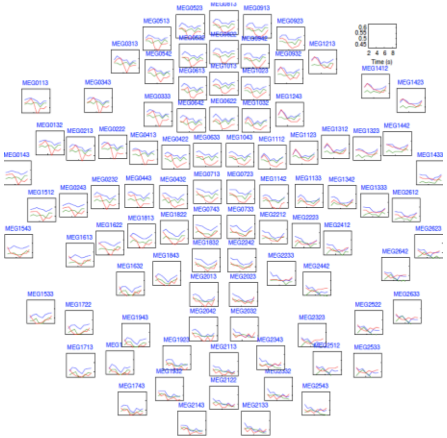


Wikipedia

Dimensionality Reduction

- Vectors store features. Lots of features!
  - Document classification: thousands of words per doc
  - Netflix surveys: 480189 users x 17770 movies
  - MEG Brain Imaging: 120 locations x 500 time points x 20 objects

	movie 1	movie 2	movie 3
Tom	5	?	?
George	?	?	3
Susan	4	3	1
Beth	4	3	?



Dimensionality Reduction

Reduce dimensions

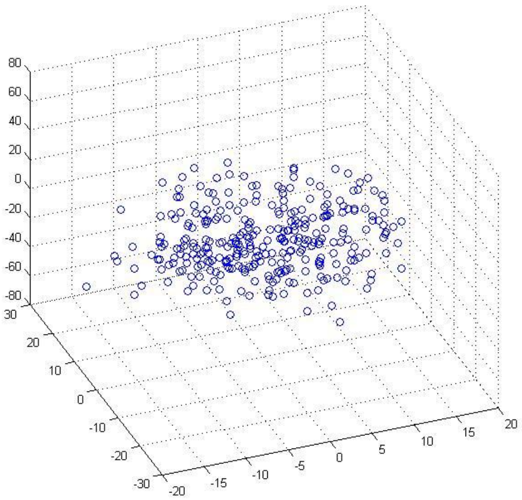
- Why?
  - Lots of features redundant
  - Storage & computation costs

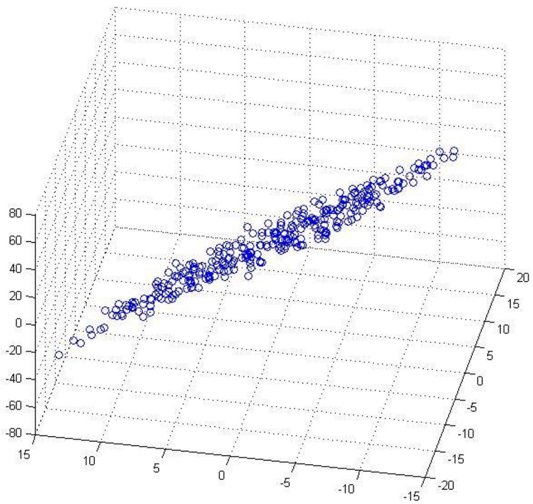
CreativeBlog

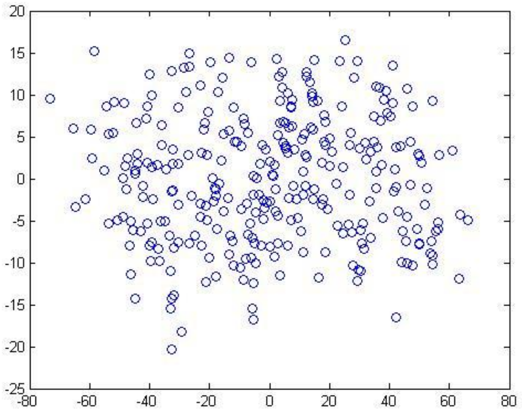
- Goal: take $x \in \mathbb{R}^d \rightarrow x \in \mathbb{R}^r$ for $r << d$
  - But minimize information loss

Dimensionality Reduction

Examples: 3D to 2D







Andrew Ng

Break & Quiz

Q 2.1: What is the inverse of $A = \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix}$

A: $A^{-1} = \begin{bmatrix} -3 & 0 \\ 0 & -2 \end{bmatrix}$

B: $A^{-1} = \begin{bmatrix} 0 & \frac{1}{3} \\ \frac{1}{2} & 0 \end{bmatrix}$

C: Undefined / *A* is not invertible

Break & Quiz

Q 2.1: What is the inverse of $A = \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix}$

A: $A^{-1} = \begin{bmatrix} -3 & 0 \\ 0 & -2 \end{bmatrix}$

B: $A^{-1} = \begin{bmatrix} 0 & \frac{1}{3} \\ \frac{1}{2} & 0 \end{bmatrix}$

C: Undefined / *A* is not invertible

$$AA^{-1} = \begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 * a + c * 2 & 0 * b + 2 * d \\ 3 * a + c * 0 & 3 * b + 0 * d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$2c = 1$$
$$3a = 0$$
$$2d = 0$$
$$3b = 1$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 1/3 \\ 1/2 & 0 \end{bmatrix}$$

Break & Quiz

Q 2.2: What are the eigenvalues of $A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

A. -1, 2, 4
B. 0.5, 0.2, 1.0
C. 0, 2, 5
D. 2, 5, 1

Break & Quiz

Q 2.2: What are the eigenvalues of $A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

A. -1, 2, 4
B. 0.5, 0.2, 1.0
C. 0, 2, 5
D. 2, 5, 1

Break & Quiz

Q 2.2: What are the eigenvalues of $A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

A. -1, 2, 4
B. 0.5, 0.2, 1.0
C. 0, 2, 5
D. 2, 5, 1

Solution #1: You may recall from a linear algebra course that the eigenvalues of a diagonal matrix (in which only diagonal entries are non-zero) are just the entries along the diagonal. Hence D is the correct answer.

Break & Quiz

Q 2.2: What are the eigenvalues of $A = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

A. -1, 2, 4
B. 0.5, 0.2, 1.0
C. 0, 2, 5
D. 2, 5, 1

Solution #2: Use the definition of eigenvectors and values: $Av = \lambda v$

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} v_1 \\ v_2 \\ v_3 \end{matrix} = \begin{bmatrix} 2v_1 + 0v_2 + 0v_3 \\ 0v_1 + 5v_2 + 0v_3 \\ 0v_1 + 0v_2 + 1v_3 \end{bmatrix} = \begin{bmatrix} 2v_1 \\ 5v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \lambda v_1 \\ \lambda v_2 \\ \lambda v_3 \end{bmatrix}$$

Since A is a 3x3 matrix, A has 3 eigenvalues and so there are 3 combinations of values for λ and v that will satisfy the above equation. The simple form of the equations suggests starting by checking each of the standard basis vectors* as v and then solving for λ. Doing so gives D as the correct answer.

*Each standard basis vector $e_i \in \mathbb{R}^n$ is the vector in which all components are zero except component i is 1.

Break & Quiz

Q 2.3: Suppose we are given a dataset with n=10000 samples with 100-dimensional binary feature vectors. Our storage device has a capacity of 50000 bits. What's the lowest compression ratio we can use?

A. 20X
B. 100X
C. 5X
D. 1X

Break & Quiz

Q 2.3: Suppose we are given a dataset with n=10000 samples with 100-dimensional binary feature vectors. Our storage device has a capacity of 50000 bits. What's the lower compression ratio we can use?

A. **20X**
B. 100X
C. 5X
D. 1X

Break & Quiz

Q 2.3: Suppose we are given a dataset with n=10000 samples with 100-dimensional binary feature vectors. Our storage device has a capacity of 50000 bits. What's the lower compression ratio we can use?

A. **20X**
B. 100X
C. 5X
D. 1X

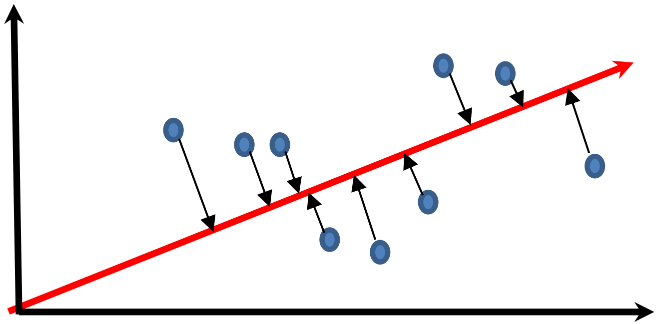
50,000 bits / 10,000 samples means compressed version must have 5 bits / sample.

Dataset has 100 bits / sample.

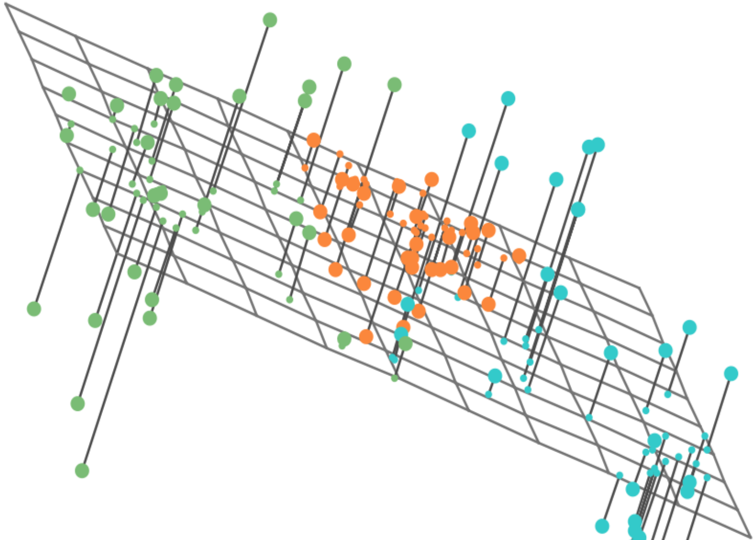
Must compress 20x smaller to fit on device.

Principal Components Analysis (**PCA**)

- A type of dimensionality reduction approach
  - For when data is **approximately lower dimensional**



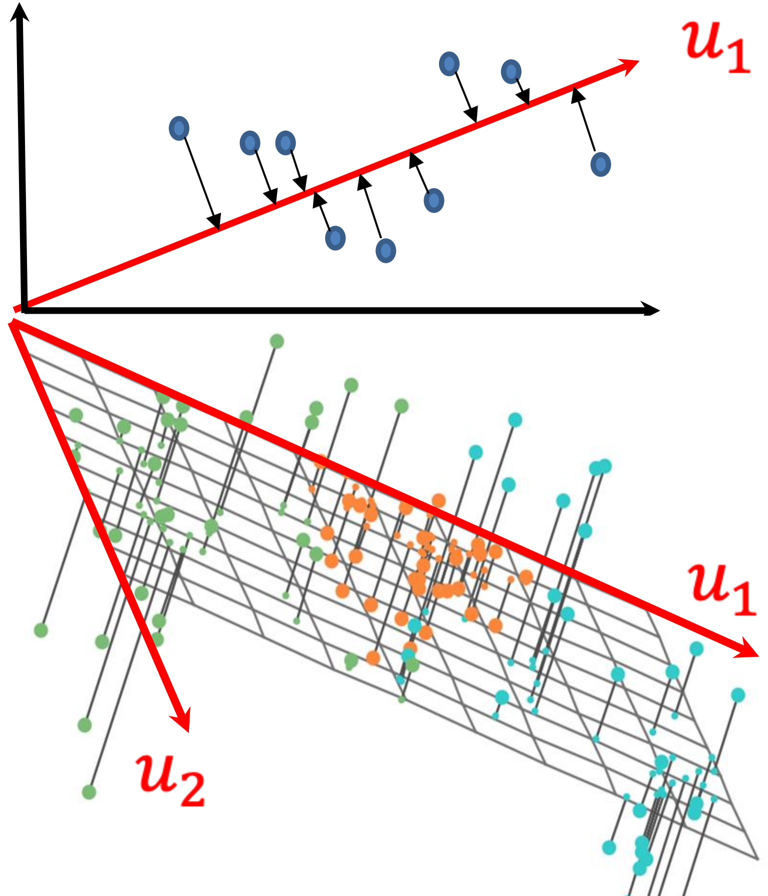
2D
↓
1D



3D
↓
2D

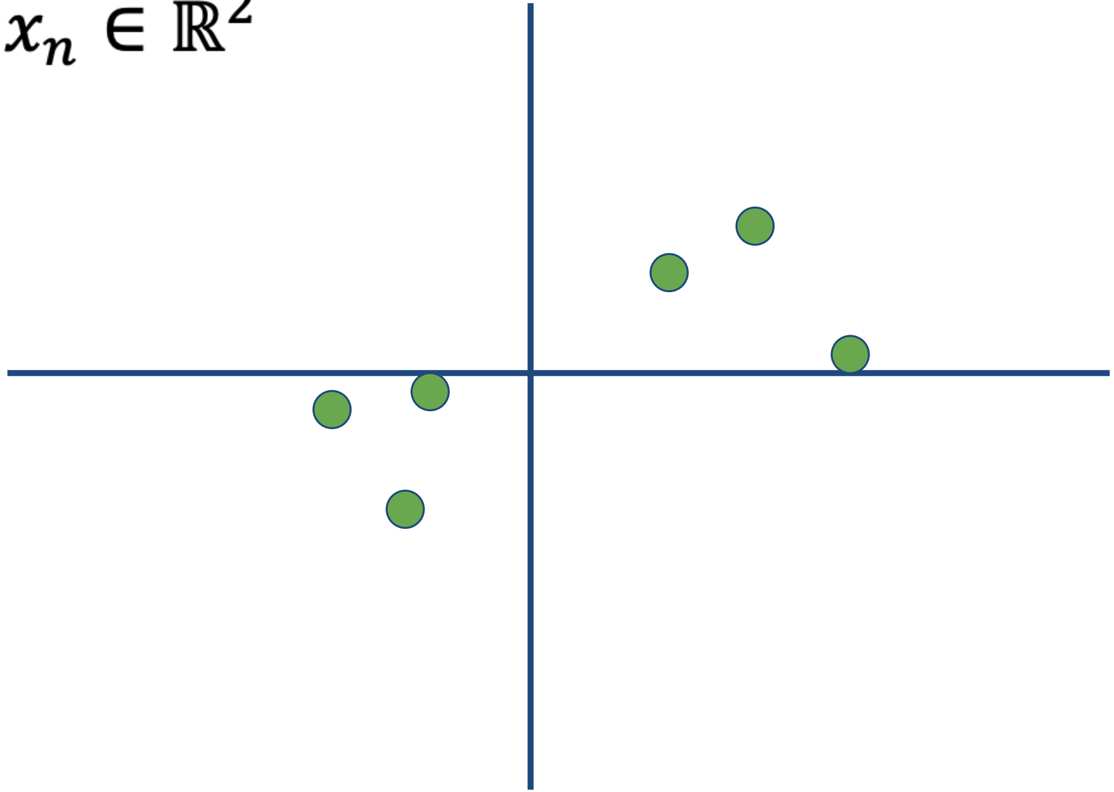
Principal Components Analysis (**PCA**)

- Find axes $u_1, u_2, \ldots, um \in \mathbb{R}^d$ of a subspace
  - Will project to this subspace
- Want to preserve data
  - Minimize projection error
- These vectors are the **principal components**



Projection: An Example

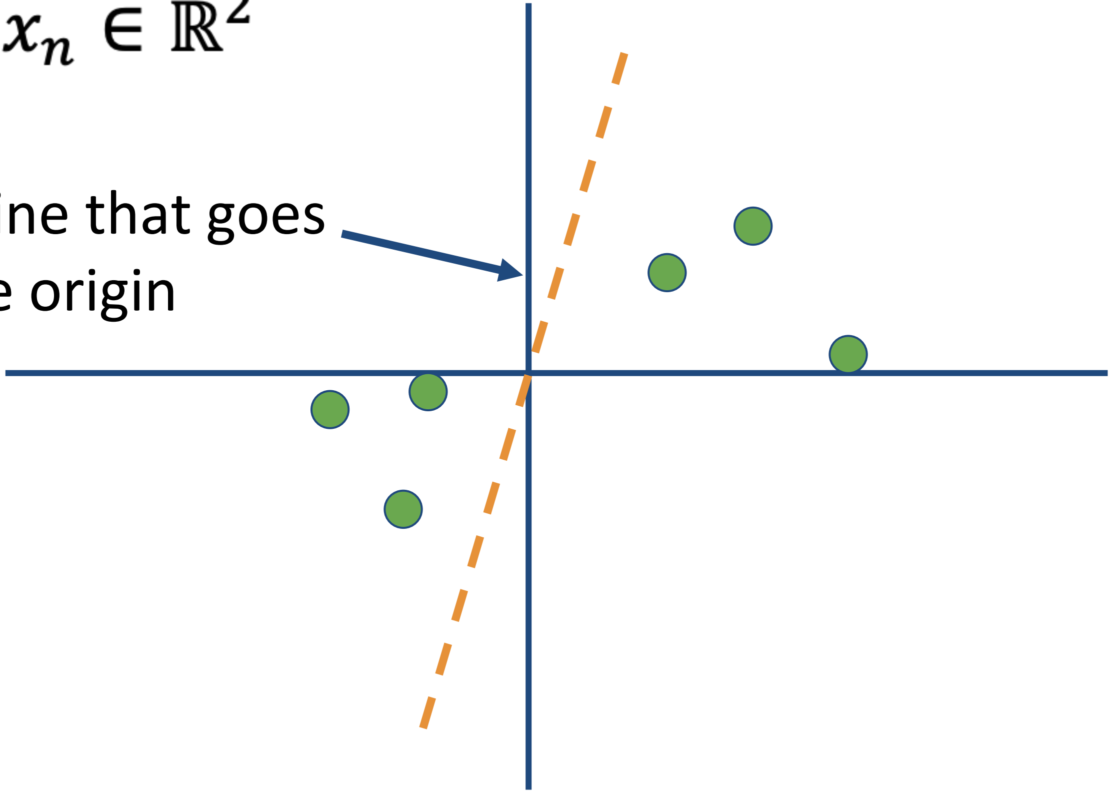
$x_1, x_2, \dots, x_n \in \mathbb{R}^2$



Projection: An Example

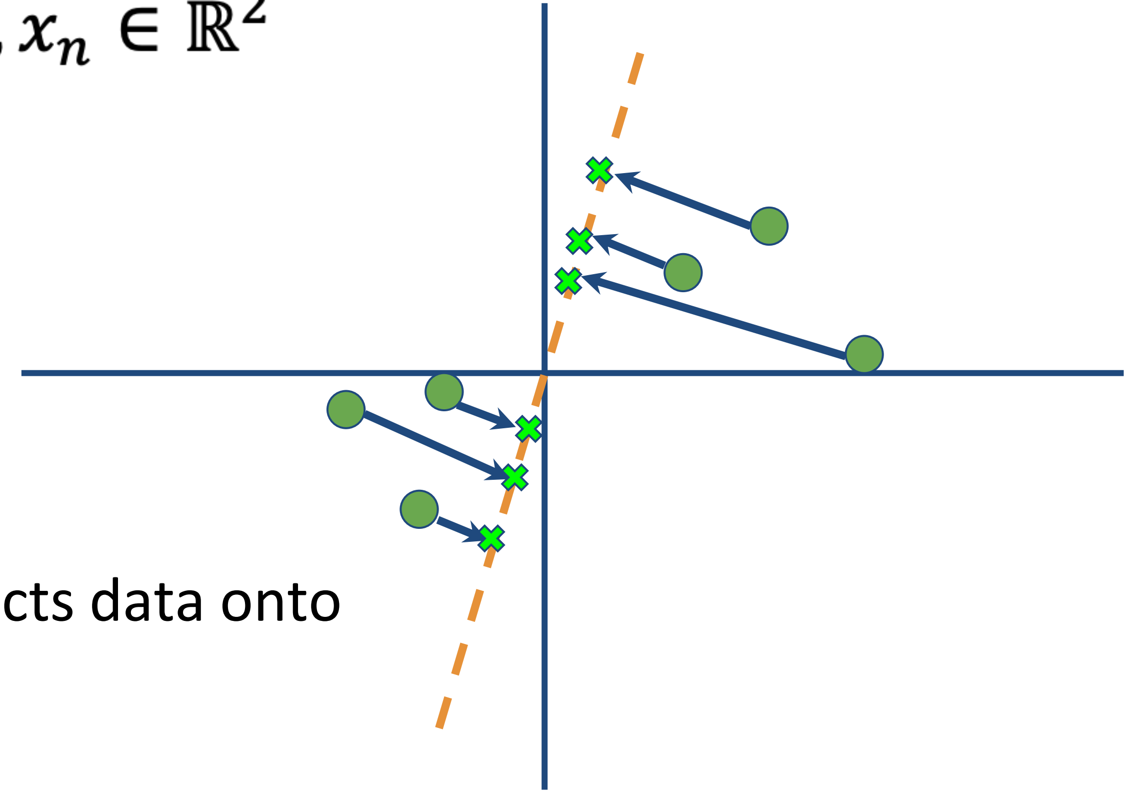
$x_1, x_2, \dots, x_n \in \mathbb{R}^2$

A random line that goes through the origin



Projection: An Example

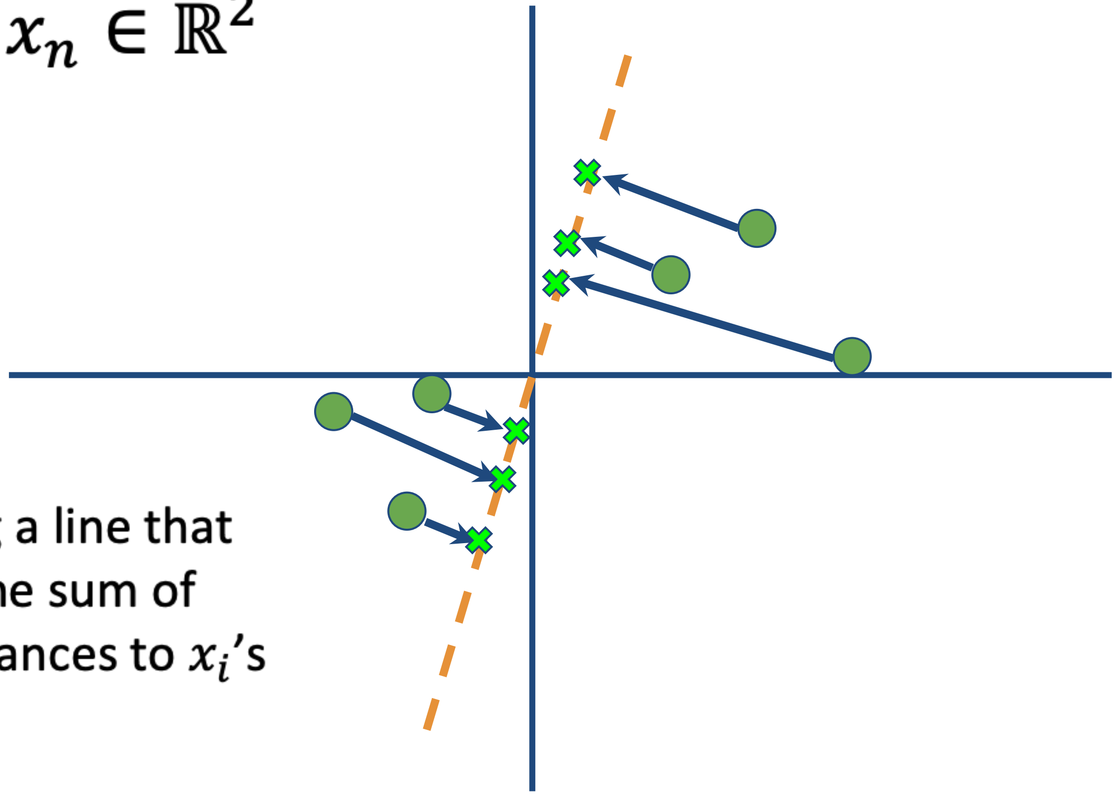
$x_1, x_2, \ldots, x_n \in \mathbb{R}^2$



PCA projects data onto this line

Projection: An Example

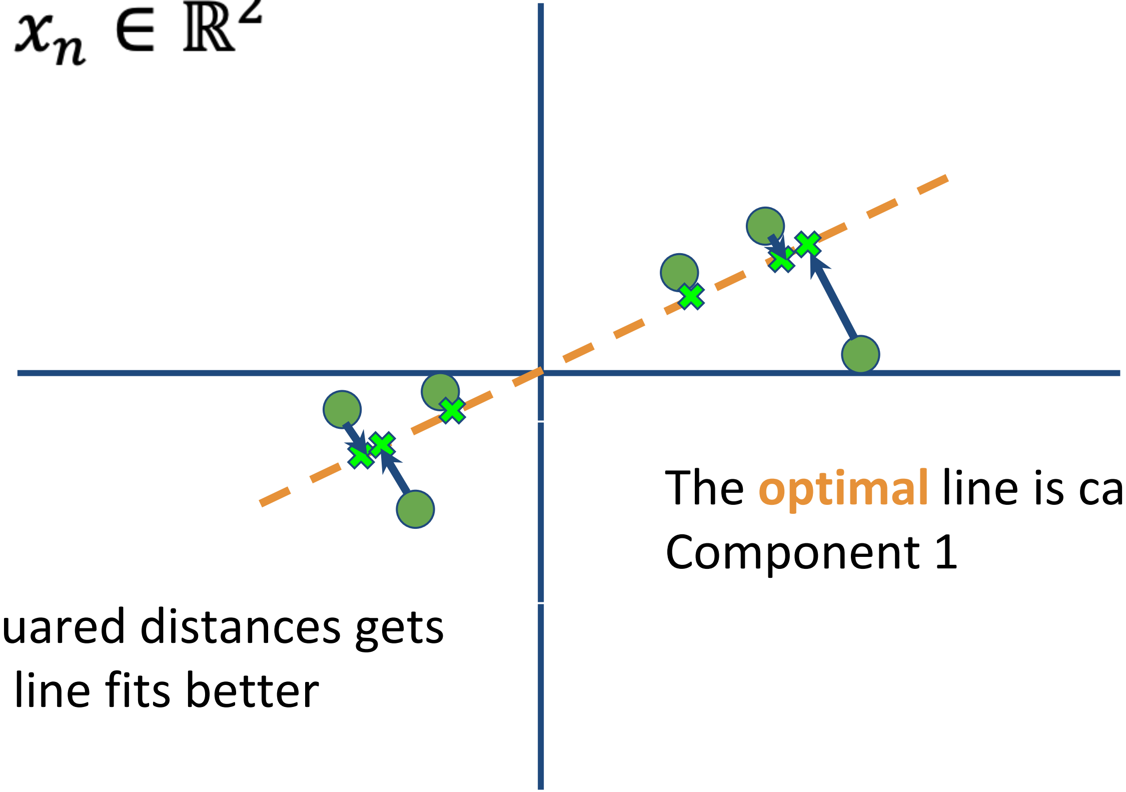
$x_1, x_2, \ldots, x_n \in \mathbb{R}^2$



Goal: finding a line that **minimizes** the sum of squared distances to x_i's

Projection: An Example

$x_1, x_2, \ldots, x_n \in \mathbb{R}^2$



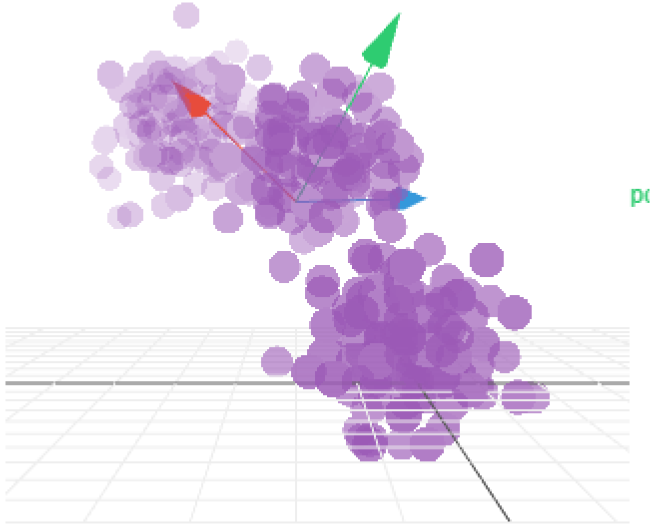
The **optimal** line is called Principal Component 1

The sum of squared distances gets smaller as the line fits better

PCA Procedure

Inputs: data $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$

- **Center data so that $\frac{1}{n}\sum_{i=1}^{n} x_i = 0$**



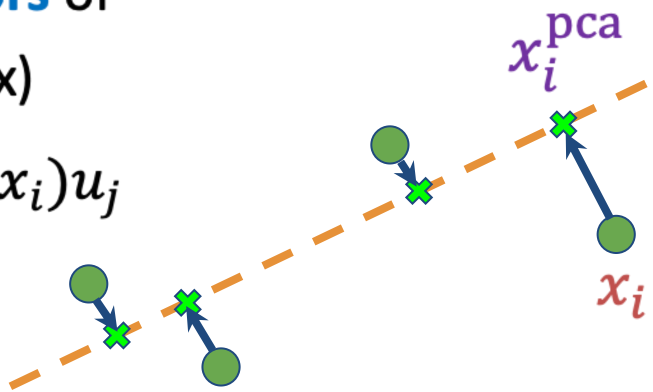
Victor Powell

PCA Procedure

Output:

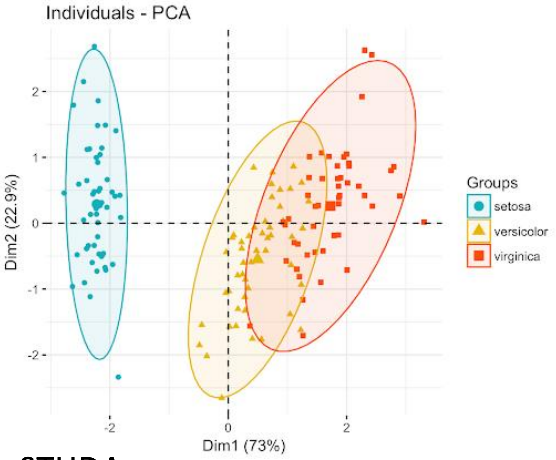
principal components $u_1, \dots, u_m \in \mathbb{R}^d$

- Can show: they are top-m **eigenvectors** of $S = \frac{1}{n-1}\sum_{i=1}^{n} x_i x_i^\top$ (covariance matrix)
- Each x_i projected to $x_i^{\text{pca}} = \sum_{j=1}^{m}(u_j^\top x_i)u_j$



Many Variations

- PCA, Kernel PCA, ICA, CCA
  - Extract structure from high dimensional dataset
- Uses:
  - **Visualization**
  - Efficiency
  - Noise removal
  - Downstream machine learning use



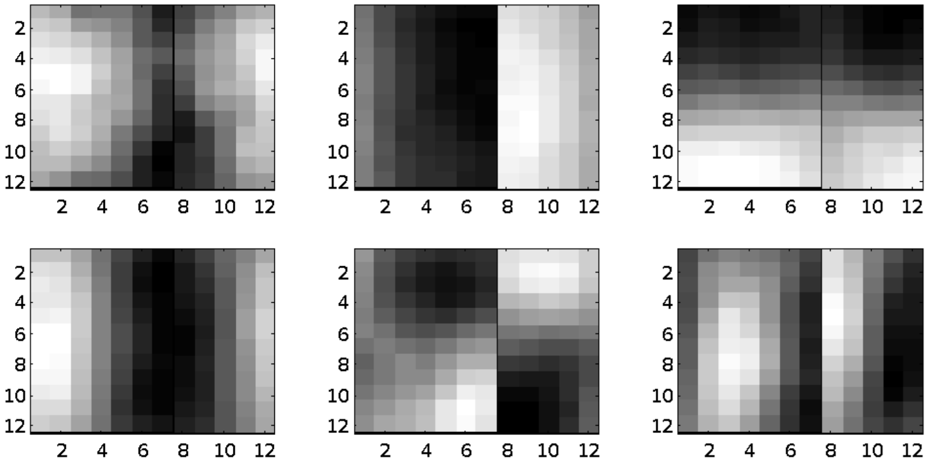
STHDA

Application: Image Compression

- Start with image; divide into 12x12 patches
  - That is, 144-D vector
  - **Original image:**

Application: Image Compression

- 6 principal components (as an image)



Application: Image Compression

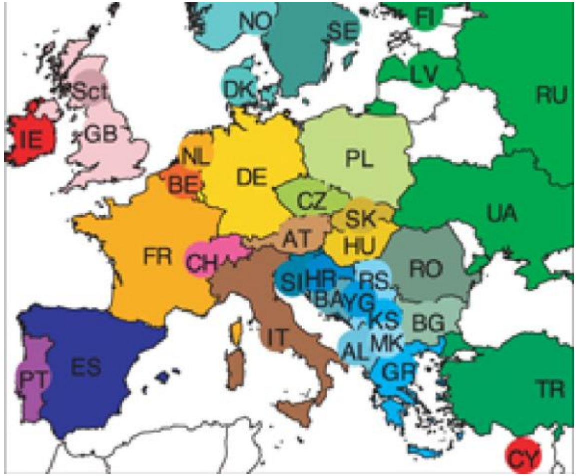
- Project to 6D

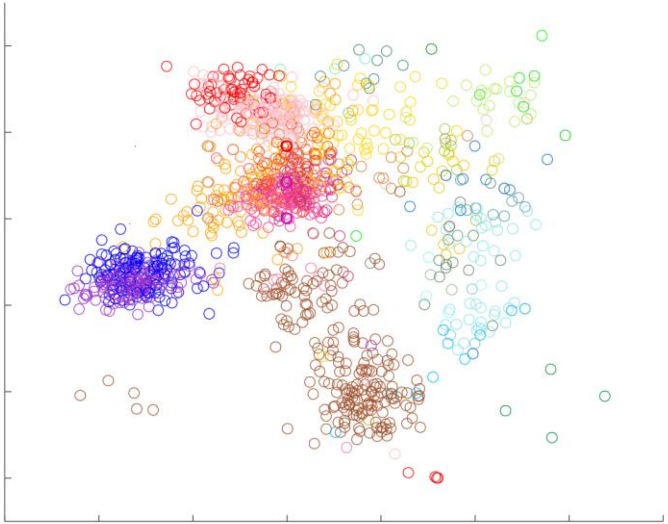
Compressed

Original

Application: Exploratory Data Analysis

- [**Novembre et al. ’08**]: Take top two singular vectors of people x SNP matrix (POPRES)





“Genes Mirror Geography in Europe”

Readings

- Local classes: Math/Stat 431
- More on PCA (and other matrix methods in ML): **CS 532**
- **Suggested reading:**
  - Probability and Statistics: The Science of Uncertainty, Michael J. Evans and Jeff S. Rosenthal http://www.utstat.toronto.edu/mikevans/jeffrosenthal/book.pdf
    (Chapters 1-3, excluding "advanced" sections)
  - Textbook: Artificial Intelligence: A Modern Approach (4th edition). Stuart Russell and Peter Norvig. Pearson, 2020. Appendix A
  - Lecture notes on PCA by Roughgarden and Valiant
    https://web.stanford.edu/class/cs168/l/l7.pdf
  - 760 notes by Zhu https://pages.cs.wisc.edu/~jerryzhu/cs760/PCA.pdf