CS 540 Introduction to Artificial Intelligence
Principal Component Analysis

University of Wisconsin–Madison
Fall 2025, Section 3
September 12, 2025

Announcements

- **HW 1 online:**
  - Writing assignment---nothing too stressful
  - Deadline **Friday, 9/19, 11:59PM**
- **HW 2 released Friday 9/19**
  - Probability & Statistics
- Optional review sessions with TA Guy Zamir
  - Thursdays 5:30-6:30 pm, Morgridge 3610

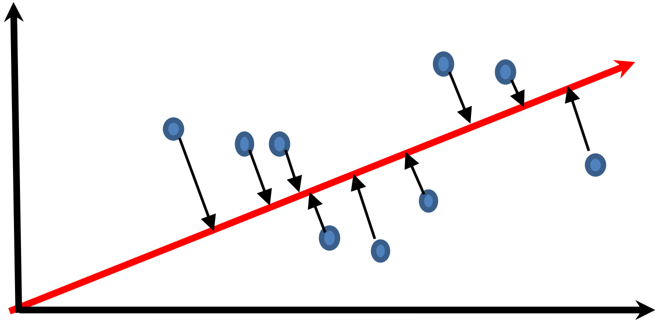
Class Roadmap

Probability & Statistics
Linear Algebra
Principal Component Analysis (PCA)
Logic
NLP
Machine Learning: Introduction

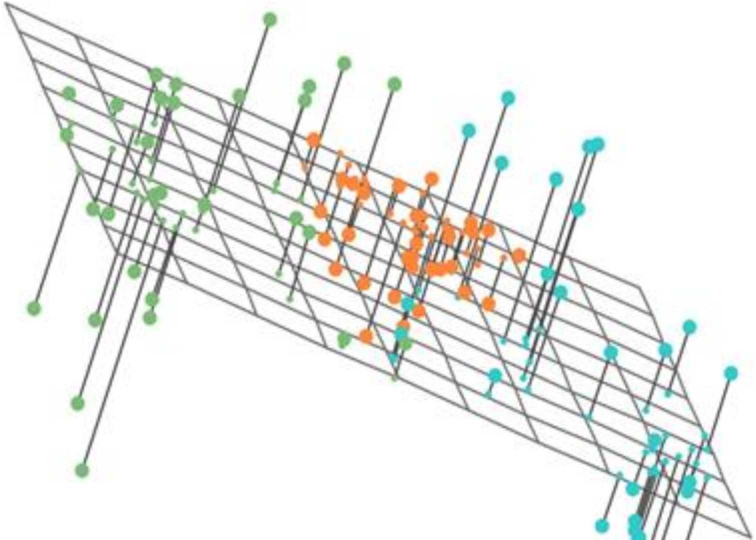
Mostly Foundations

Principal Components Analysis (**PCA**)

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



2D → 1D

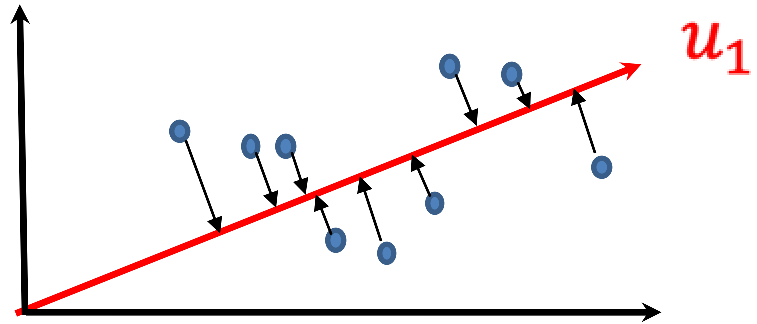


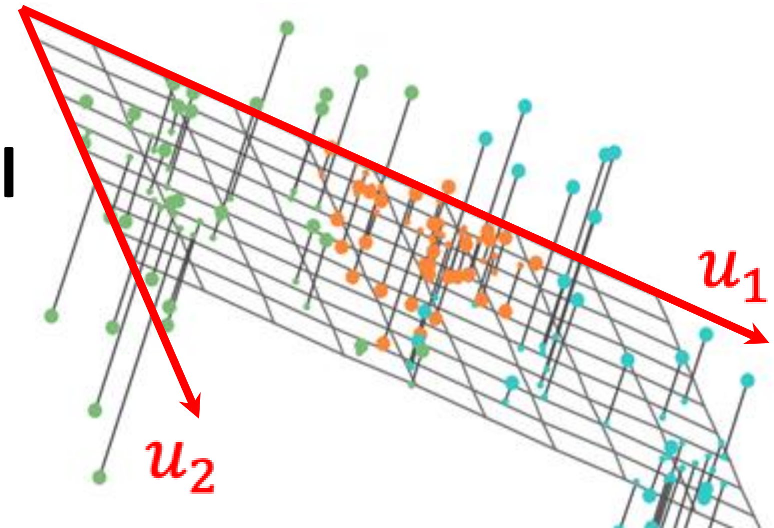
3D → 2D

Principal Components Analysis (**PCA**)

- Find axes $u_1, \ldots, u_k \in \mathbb{R}^d$ of subspace
- Project to this subspace

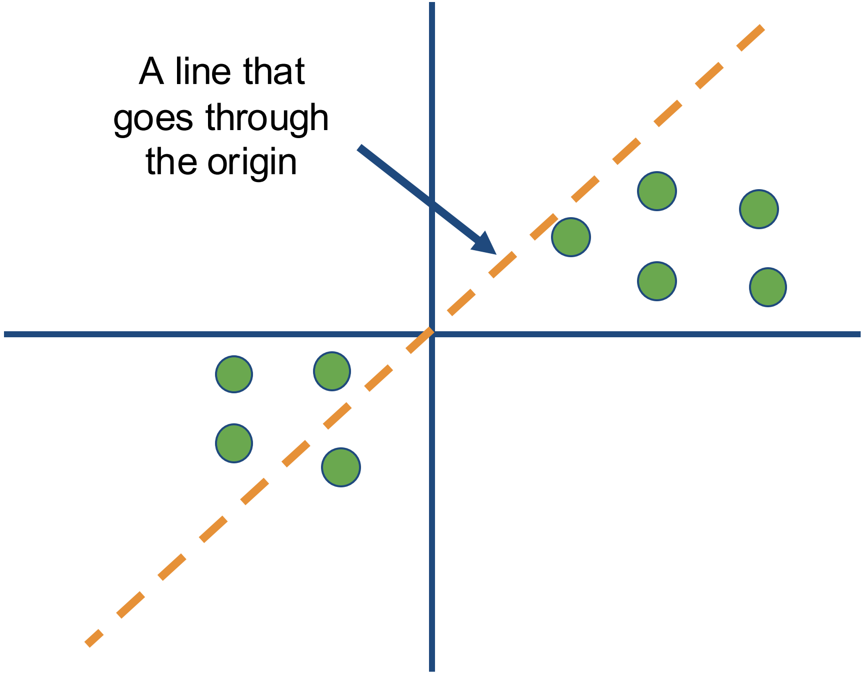
- These vectors are **principal components**





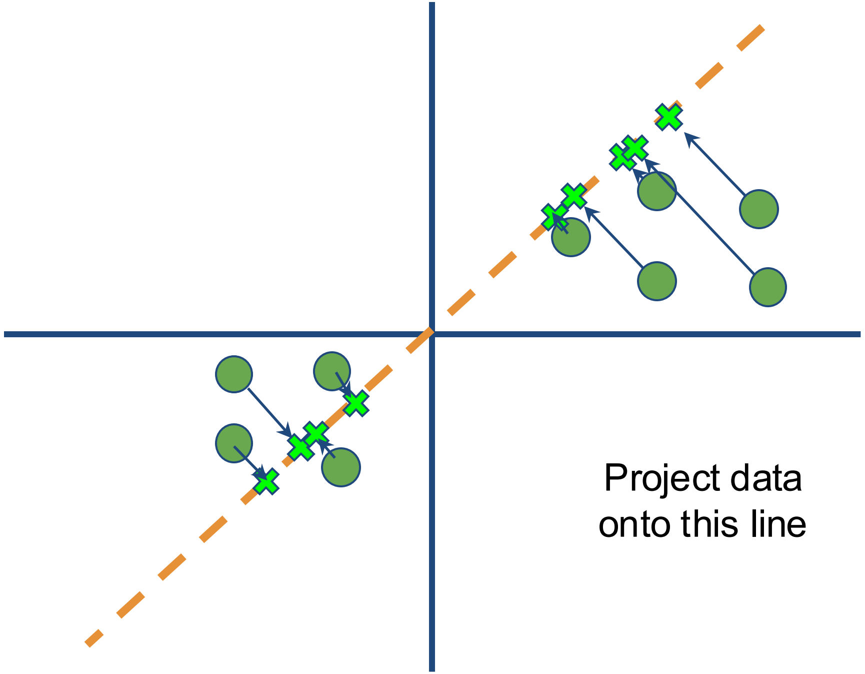
Projection: An Example

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



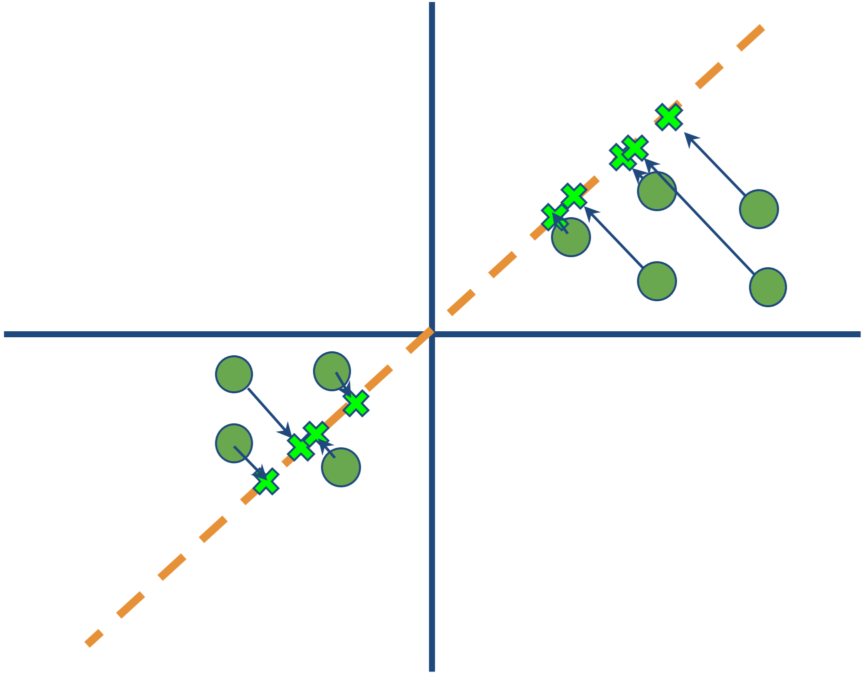
Projection: An Example

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



Projection: An Example

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



After projection, we see a one-dimensional dataset

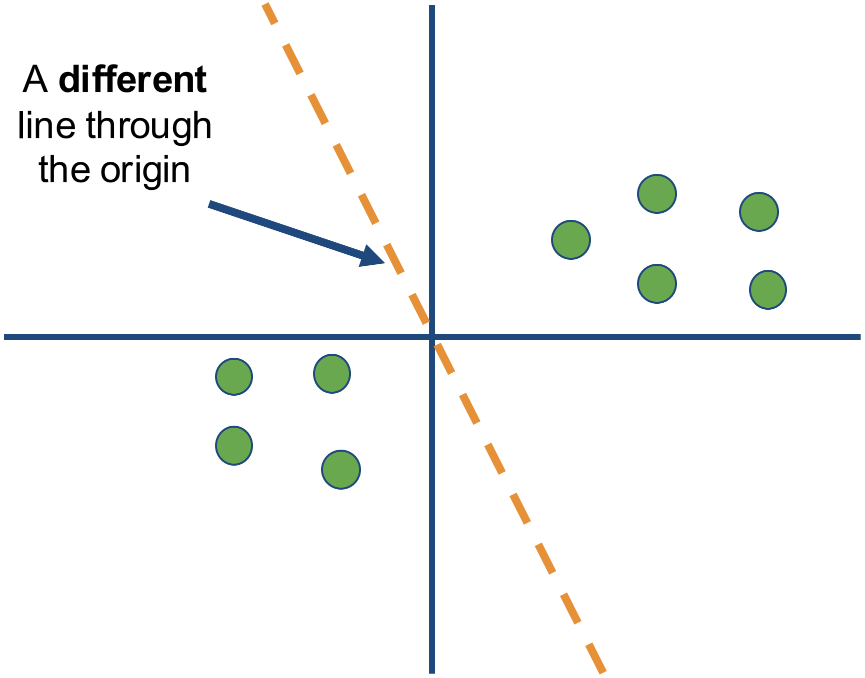
This projection **preserves** information about the data.

=

Good projection!

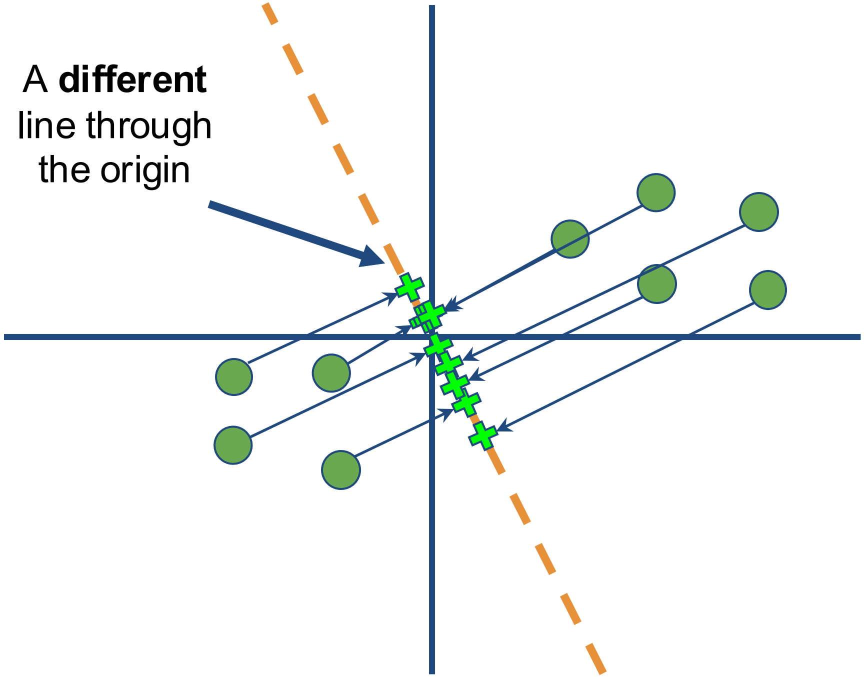
Projection: Another Example

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



Projection: Another Example

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



After projection, we see a one-dimensional dataset

This projection **loses** information about the data.

=

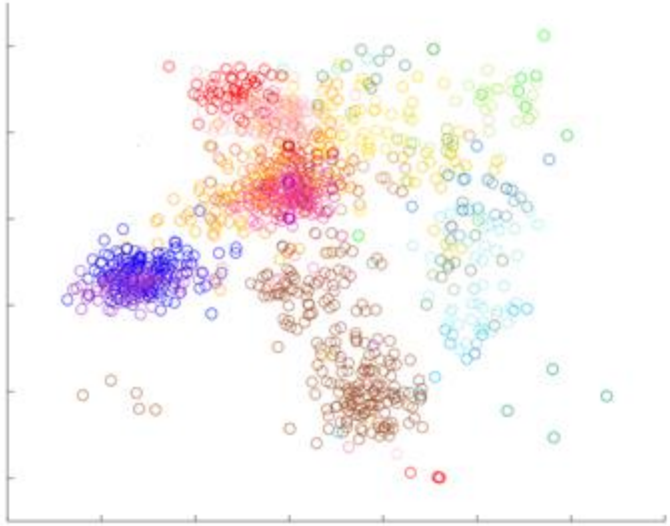
Bad projection!

Plan for this lecture

- Applications of PCA
- Formalizing: what makes a good projection?
- Computing PCA
  - How do we find good projections?
  - Connections to eigenvectors

Application: Exploratory Data Analysis

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



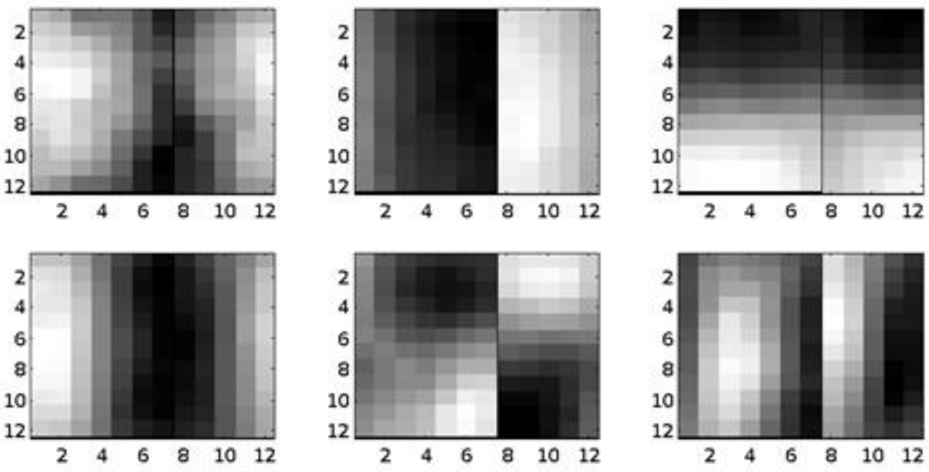
“Genes Mirror Geography in Europe”

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

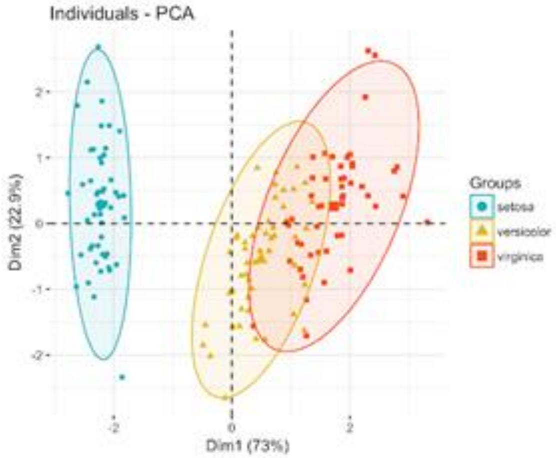
- Project to 6D

Compressed

Original

Many Variations

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



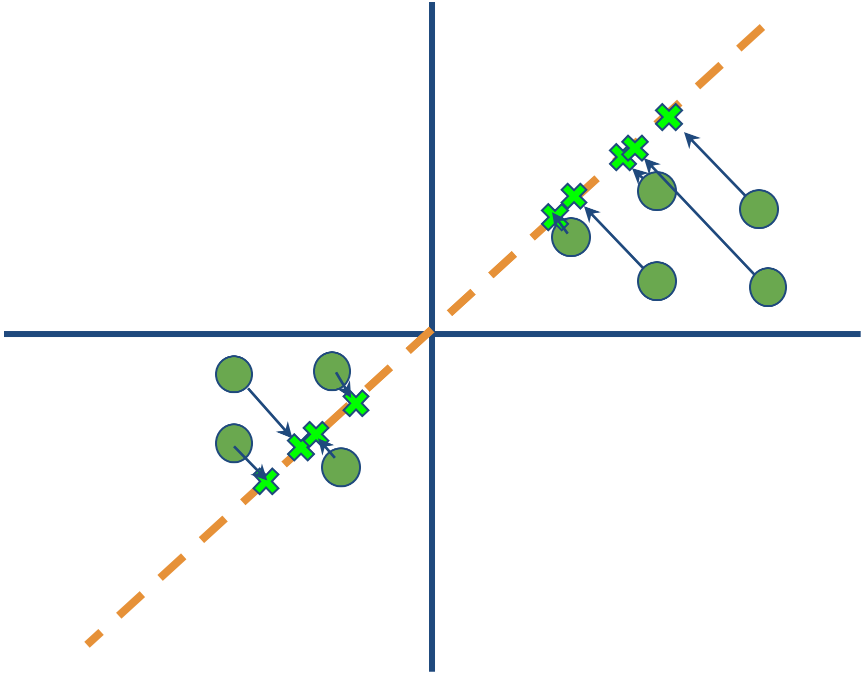
STHDA

Plan for this lecture

- Applications of PCA
- Formalizing: what makes a good projection?
- Computing PCA
  - How do we find good projections?
  - Connections to eigenvectors

What makes a good projection?

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



After projection, we see a one-dimensional dataset

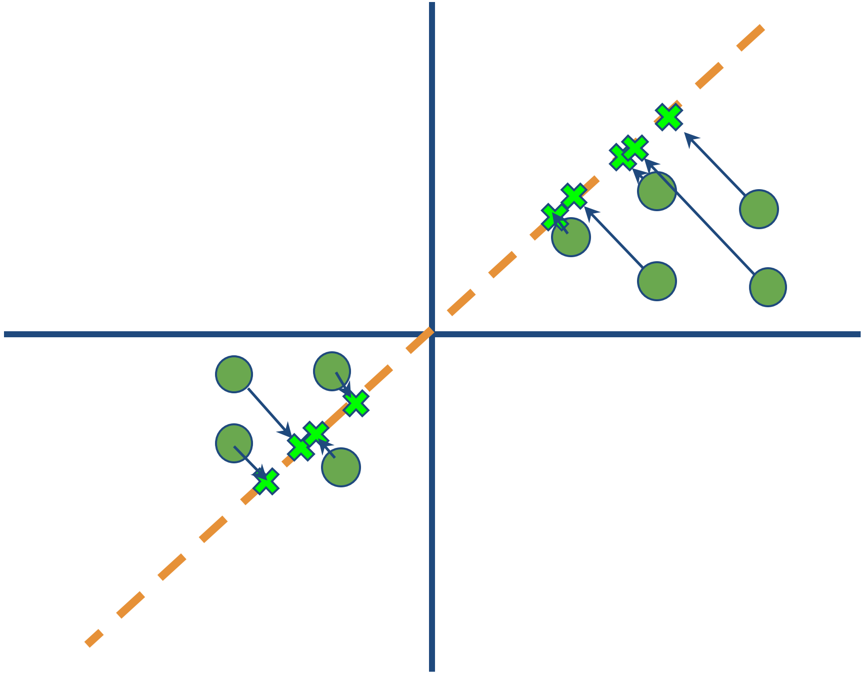
This projection **preserves** information about the data.

=

Good projection!

What makes a good projection?

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



PCA says: a good projection makes the data "spread out"

Direction to maximize variance

- Direction: vector v with length $\|v\|_2 = 1$

$$\underset{v\,:\,\|v\|_2 = 1}{\operatorname{argmax}} \sum_{i=1}^{n} \langle x_i, v \rangle^2$$

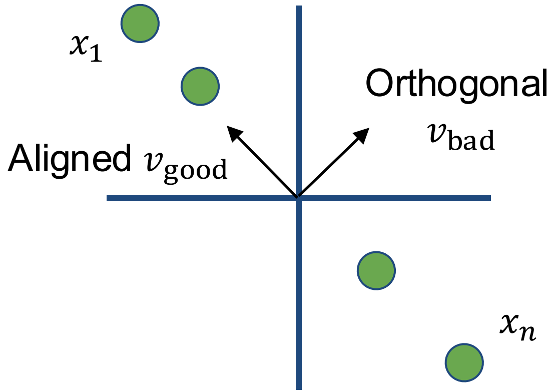
- Inner product $\Leftrightarrow$ projection

How can we solve this?
Turn to linear algebra!

A simple example in detail

- n data points: $x_1, x_2, \dots, x_n \in \mathbb{R}^2$
- All in a single line
- Exists vector $v_{\text{good}} \in \mathbb{R}^2$ such that:

  for all $i, x_i = c_i v_{\text{good}}$
- Project onto v_{good}:



$$\sum_{i=1}^{n} \langle x_i, v_{\text{good}} \rangle^2 = \sum_{i=1}^{n} \langle c_i v_{\text{good}}, v_{\text{good}} \rangle^2$$

$$= \sum_{i=1}^{n} c_i^2 \| v_{\text{good}} \|_2^2$$

$$= \sum_{i=1}^{n} c_i^2$$

This is the biggest the sum can be for any vector

- Project onto orthogonal v_{bad}:

$$\sum_{i=1}^{n} \langle x_i, v_{\text{bad}} \rangle^2 = \sum_{i=1}^{n} 0$$

Orthogonal means $\langle x_i, v_{\text{bad}} \rangle = 0$

Plan for this lecture

- Applications of PCA
- Formalizing: what makes a good projection?
- Computing PCA
  - How do we find good projections?
  - Connections to eigenvectors

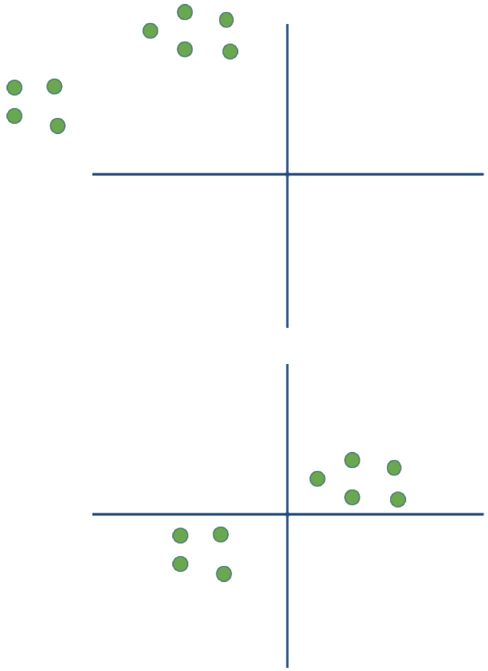
Step 0: Center the Dataset

- Compute the mean

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- Center data at zero

$$\bar{x}_1 = x_1 - \mu$$
$$\bar{x}_2 = x_2 - \mu$$
$$\vdots$$
$$\bar{x}_n = x_n - \mu$$



(simplifies remaining calculations)

Step 1: Write the Dataset as Matrix

- Write $x_1, \ldots, x_n$ as a matrix X

$$X = \begin{bmatrix} — & x_1 & — \\ — & x_2 & — \\ — & \vdots & — \\ — & x_n & — \end{bmatrix}$$

$X \in \mathbb{R}^{n \times d}$
n rows and
d columns

Step 2: Write the sum as a product

PCA objective:

$$\underset{v\,:\,\|v\|_2 = 1}{\operatorname{argmax}} \sum_{i=1}^{n} \langle x_i, v \rangle^2$$

Equivalent:

$$\underset{v\,:\,\|v\|_2 = 1}{\operatorname{argmax}} \, v^T X^T X v$$

Inner product!

$$Xv = \begin{bmatrix} — & x_1 & — \\ — & x_2 & — \\ — & \vdots & — \\ — & x_n & — \end{bmatrix} \begin{bmatrix} | \\ v \\ | \end{bmatrix} = \begin{bmatrix} \langle x_1, v \rangle \\ \langle x_2, v \rangle \\ \vdots \\ \langle x_n, v \rangle \end{bmatrix}$$

Step 3: Eigenvectors of $X^T X$

Equivalent to PCA objective:

$$\underset{v\,:\,\|v\|_2 = 1}{\operatorname{argmax}}\ v^T X^T X v$$

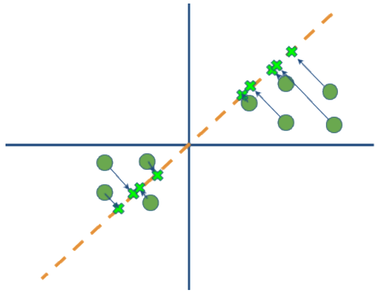
Recall: eigenvector & eigenvalue of A satisfy

$$Av = \lambda v$$

- Suppose v is an eigenvector of $X^T X$
- Then $$v^T \underline{X^T X v} = v^T \underline{(\lambda v)}$$ $$= \lambda v^T v$$ $$= \lambda \|v\|_2^2 = \lambda$$
- PCA objective $\Leftrightarrow$ top eigenvector of $X^T X$
- $\frac{1}{n} X^T X$ is **sample covariance**

Putting it all together

- Want to project down to one dimension



- Goal: maximize variance after projection

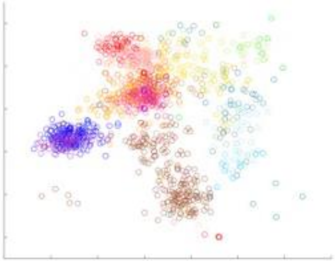
$$\underset{v\,:\,\|v\|_2 = 1}{\operatorname{argmax}} \sum_{i=1}^{n} \langle x_i, v \rangle^2$$

- Principal component = top eigenvector of sample covariance

$$(X^T X)v = \lambda v$$

Projecting to > 1 dimension

- Found single best direction: now what?



- Look at k largest eigenvalues!
- Use eigendecomposition
- Equivalent to recursion/“deflation”
  - Subtract out top eigenvector

Further Reading

- Vast literature on linear algebra.
- Local class: **Math 341**
- More on PCA (and other matrix methods in ML): **ECE/CS 532**

- Suggested reading: Lecture notes on PCA by Roughgarden and Valiant
  - https://web.stanford.edu/class/cs168/l/l7.pdf
  - https://web.stanford.edu/class/cs168/l/l8.pdf