Dimensionality Reduction

Machine Learning Decal

Hosted by Machine Learning at Berkeley

ML@B

Agenda

Why reduce dimensions of data?

PCA - Principle Component Analysis

LDA - Linear Discriminant Analysis

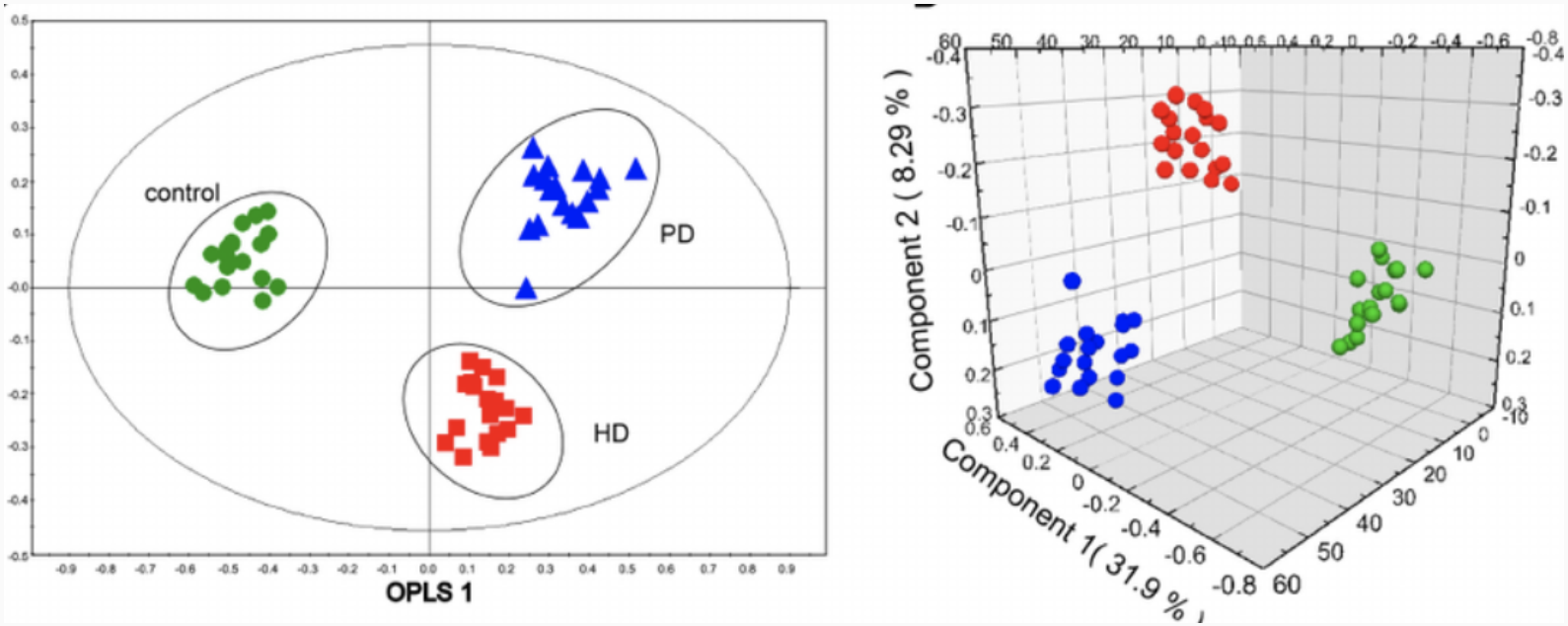
Why reduce dimensions of data?

Visualizing Higher Dimensional Spaces

ML@B

- We can only easily visualize up to 3 dimensions
- Given a dataset with more dimensions, it is hard to understand how the points of different classes are separated

Visualizing Higher Dimensional Spaces



For this dataset, we don't need all 3 dimensions to visualize the clusters

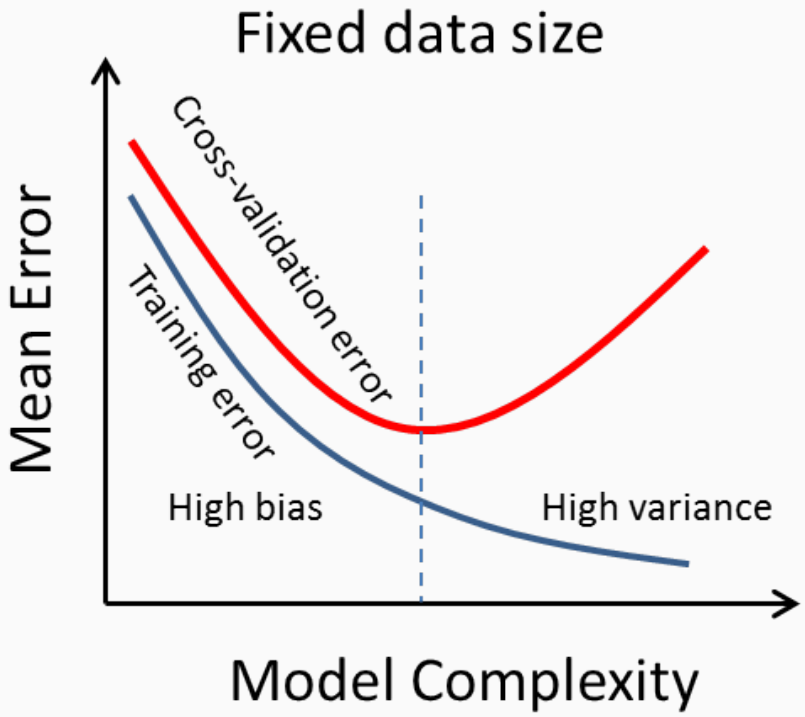
Give Classifiers Less Features

ML@B

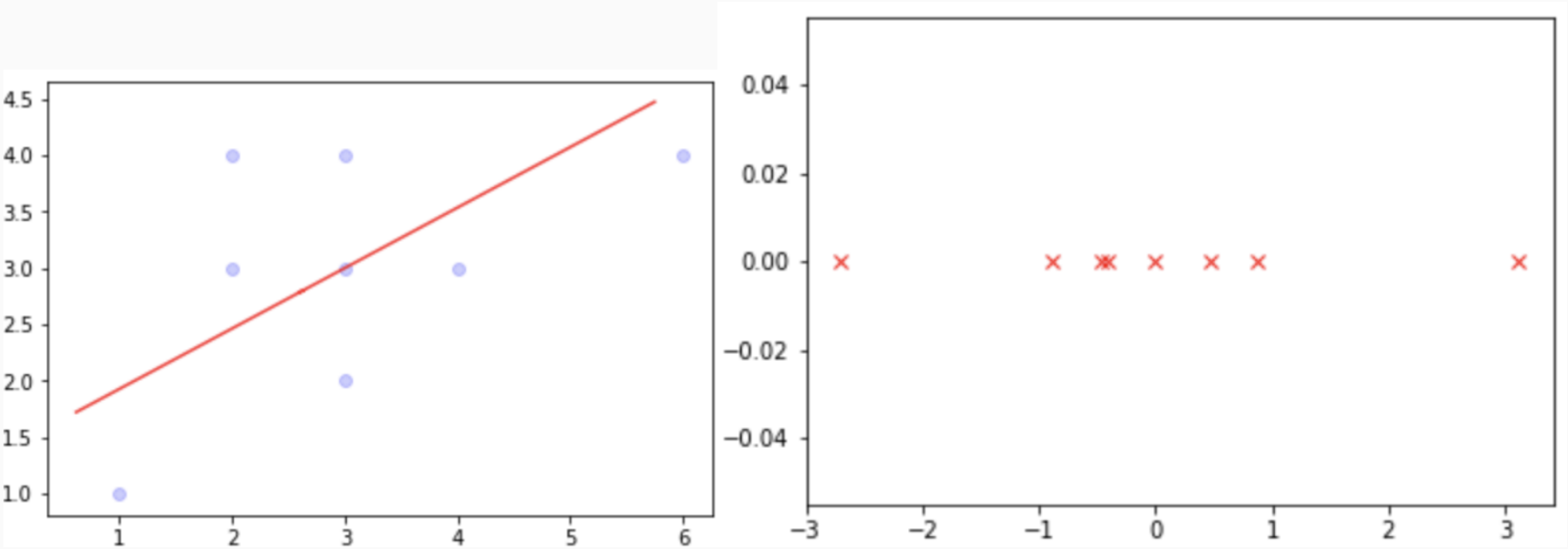
Why do we want a classifier with less features?

- Recall from the bias-variance tradeoff that any feature added will increase the variance of a model
  - Models with more features are more expressive
- Ideally we want to only include features that give enough information to greatly decrease the bias to justify the increase in variance
  - Goal is to get rid of features with redundant information
- Model with fewer features can run faster

Less Complex Classifiers have Lower Variance



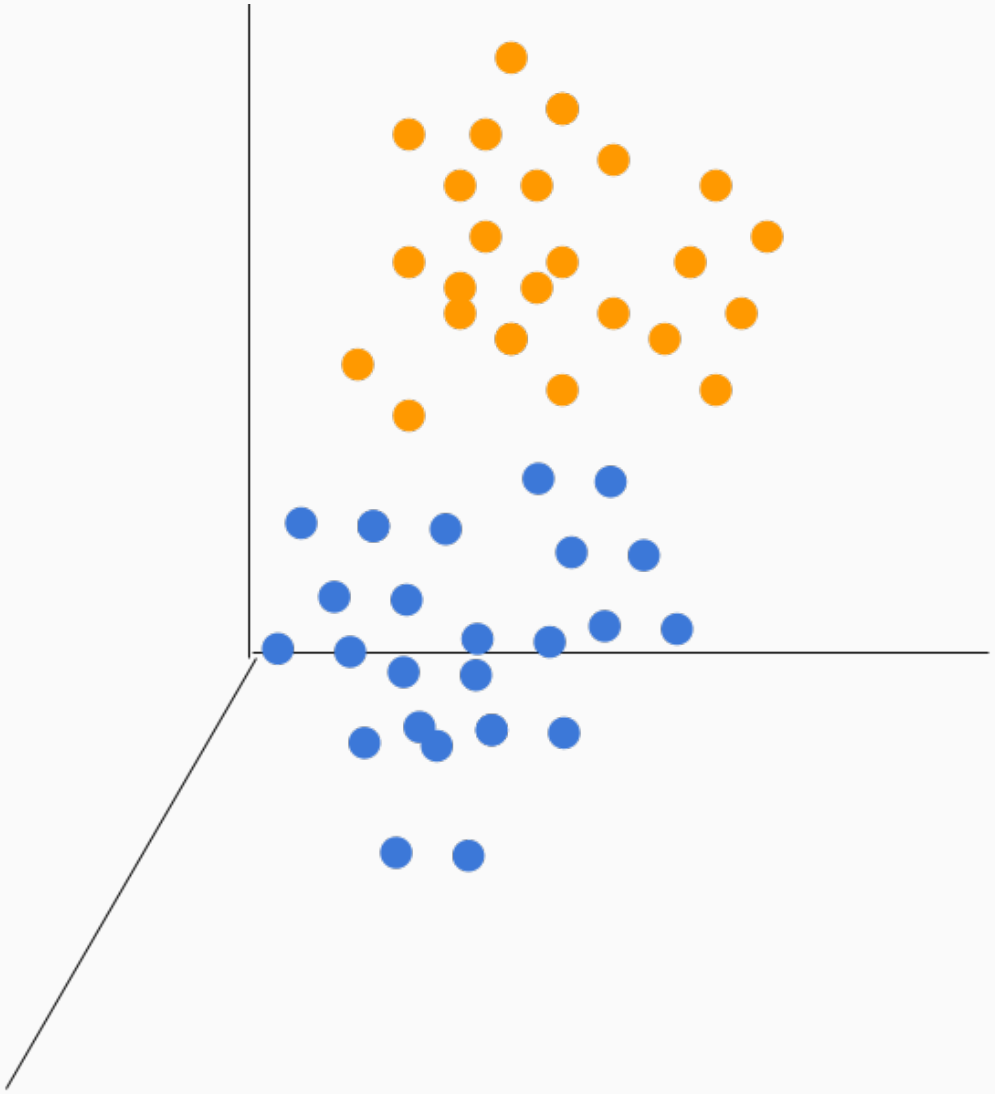
Data Compression



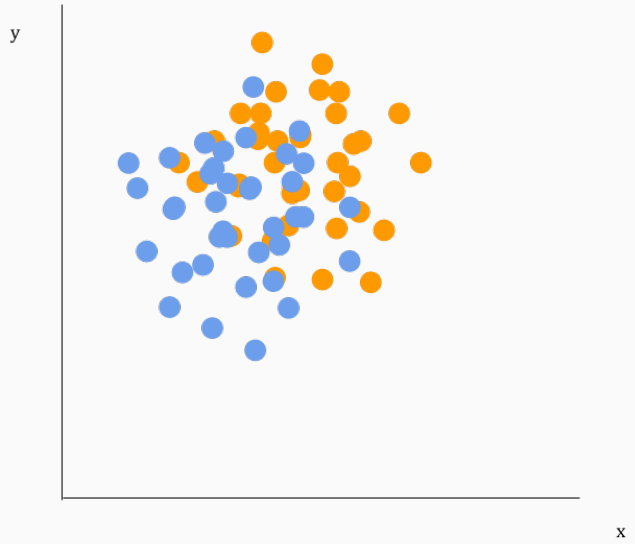
Transform $x \in R^2$ to $z \in R$

This is a form of lossy compression, but may capture enough information about data for given task

What is a Good Way to Reduce this Dataset to 2D?



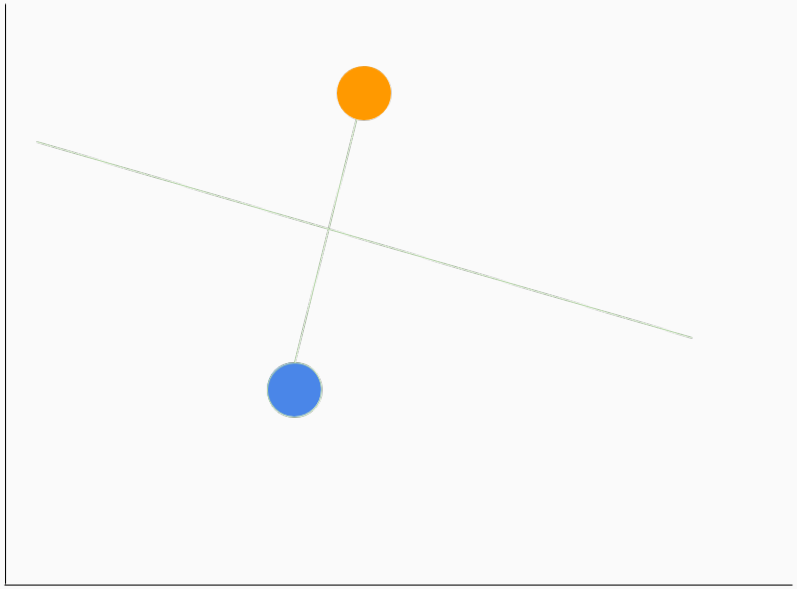
Idea 1: Remove One Axis



Drawbacks:

- This method got rid of an important feature of the data
- Possible that a dataset has important information along all axes so removing any feature removes important information

Idea 2: Cluster the Points



Drawbacks:

- In higher dimensions, points are more spread out, so distance between two points looks similar $\Rightarrow$ standard clustering doesn't work well in high dimensions

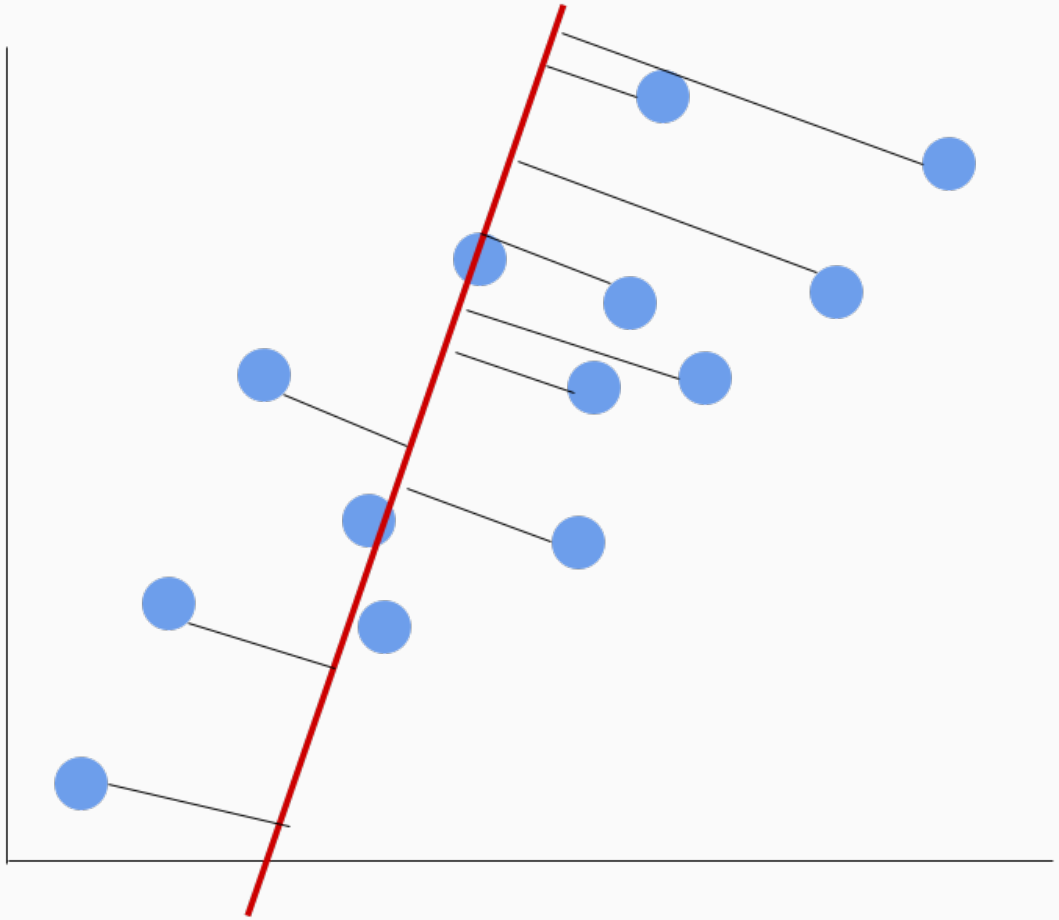
Properties of Good Dimensionality Reduction

ML@B

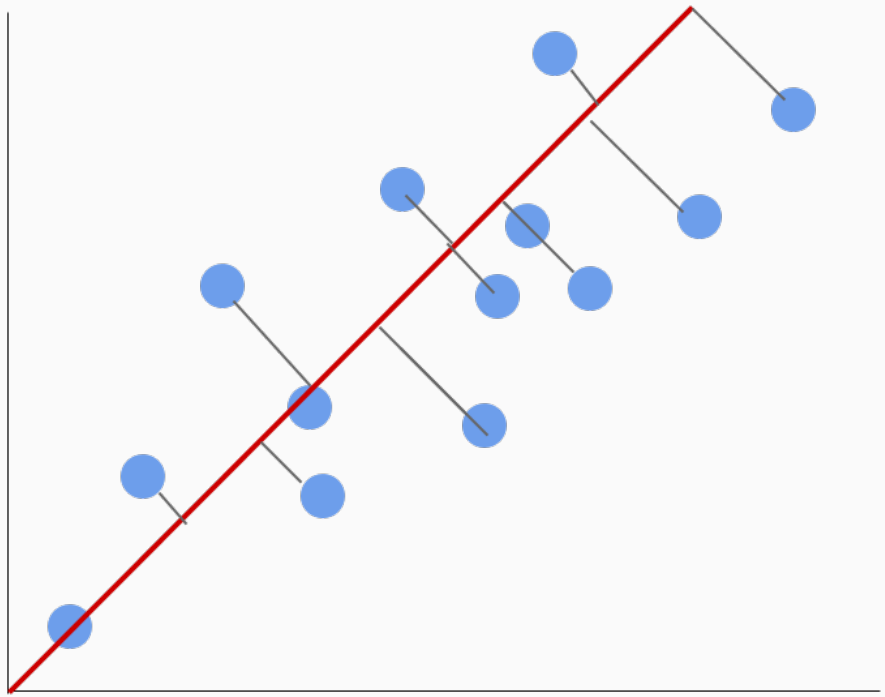
- Combination of many features rather than just eliminating a few
- Should be usable in higher dimensions

PCA - Principle Component Analysis

Which is Best Line to Project Data Onto?

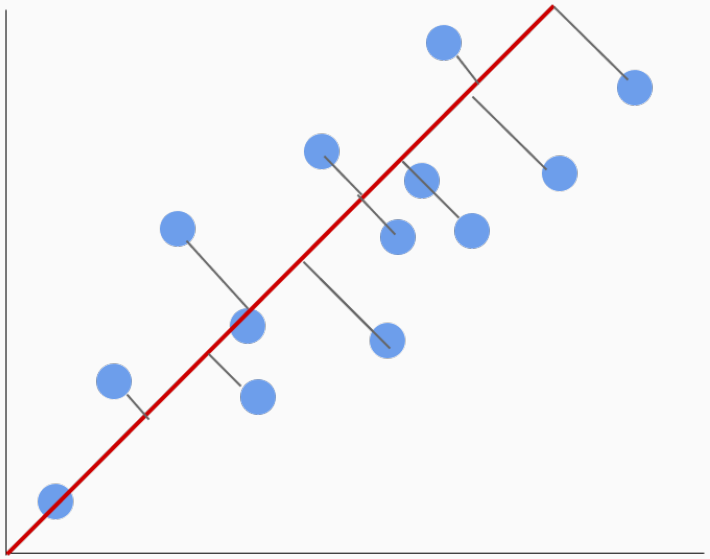


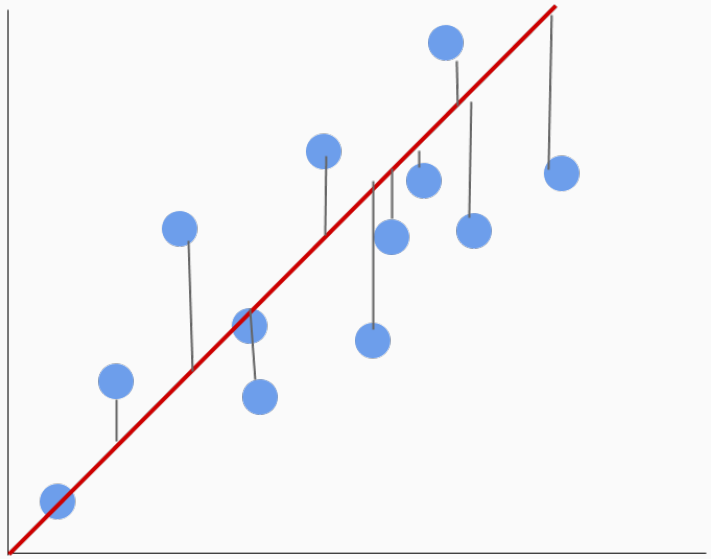
Which is Best Line to Project Data Onto?



- Goal of PCA is to find k dimensions that minimize projection error of a n dimensional dataset

Projection Error vs Linear Regression Residuals



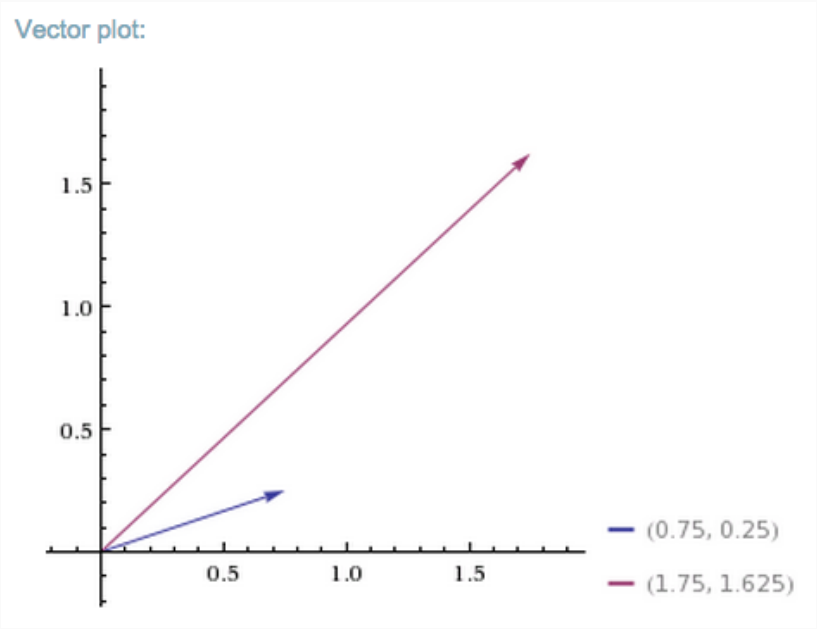


Math Review - Eigenvectors and Eigenvalues

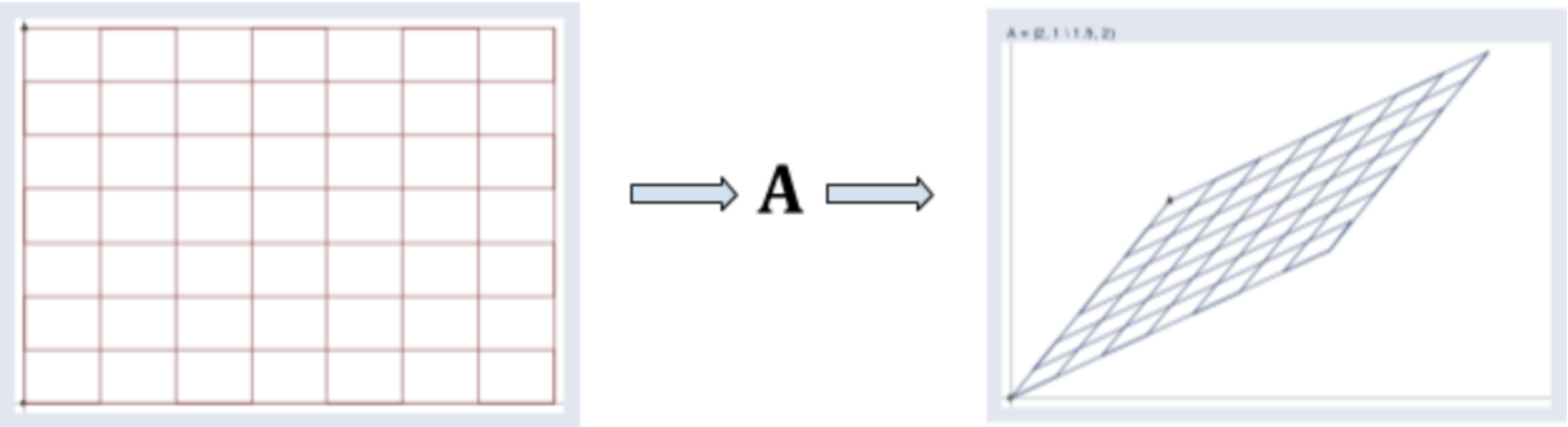
What does a matrix do to a vector?

$$A = \begin{bmatrix} 2 & 1 \\ 1.5 & 2 \end{bmatrix}, v = \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix}$$

$$Av = \begin{bmatrix} 2 & 1 \\ 1.5 & 2 \end{bmatrix} \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix} = \begin{bmatrix} 1.75 \\ 1.625 \end{bmatrix}$$

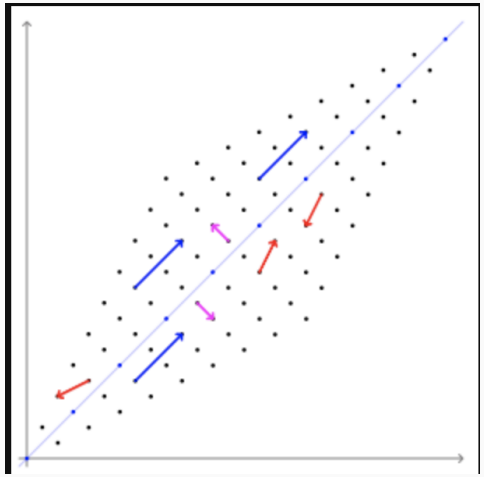


Math Review - Eigenvectors and Eigenvalues



- Matrix can be thought of as an operator that linearly transforms a vector

Math Review - Eigenvectors and Eigenvalues



- All vectors are affected after matrix is applied, but direction of eigenvectors is preserved and only length is changed
- Eigenvectors define axis of linear transformation
    - They define directions of force matrix puts on input

Math Review - Eigenvectors and Eigenvalues

ML@B

"Many mathematical objects can be understood better by breaking them into constituent parts, or finding some properties of them that are universal, not caused by the way we choose to represent them.

For example, integers can be decomposed into prime factors. The way we represent the number 12 will change depending on whether we write it in base ten or in binary, but it will always be true that $12 = 2 \times 2 \times 3$."

- Yoshua Bengio

Math Review - Eigenvectors and Eigenvalues

ML@B

"Much as we can discover something about the true nature of an integer by decomposing it into prime factors, we can also decompose matrices in ways that show us information about their functional properties that is not obvious from the representation of the matrix as an array of elements.

One of the most widely used kinds of matrix decomposition is called eigen-decomposition, in which we decompose a matrix into a set of eigenvectors and eigenvalues."

- Yoshua Bengio

Math Review - Eigenvectors and Eigenvalues

$$Av = \lambda v$$

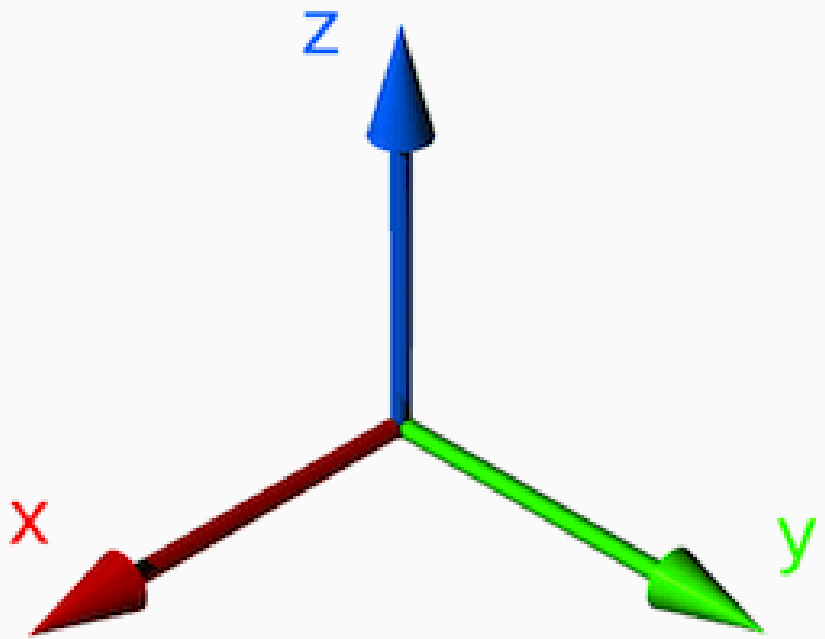
where A is a matrix, v is a eigenvector of A, λ is eigenvalue of A

- Notice that applying A to v only scales the length of v (doesn't change the direction)
- $A_{n \times n}$ can have up to n unique eigenvectors and eigenvalues

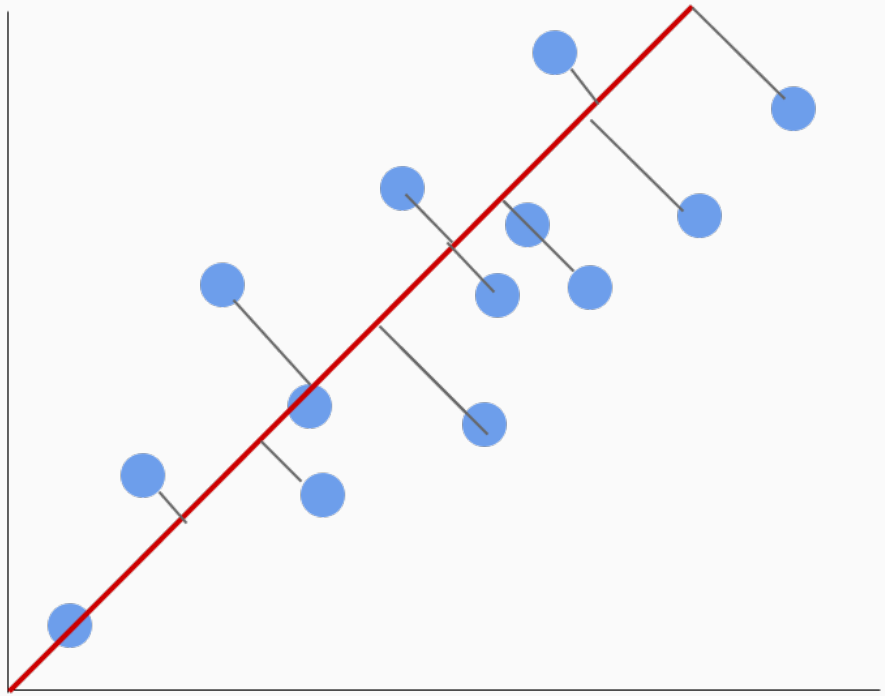
Math Review - Eigenvectors and Eigenvalues

If A is symmetric ($A = A^T$) then eigenvectors are orthogonal ($v_i^T v_j = 0$)

- Orthogonal vectors define different dimension of space

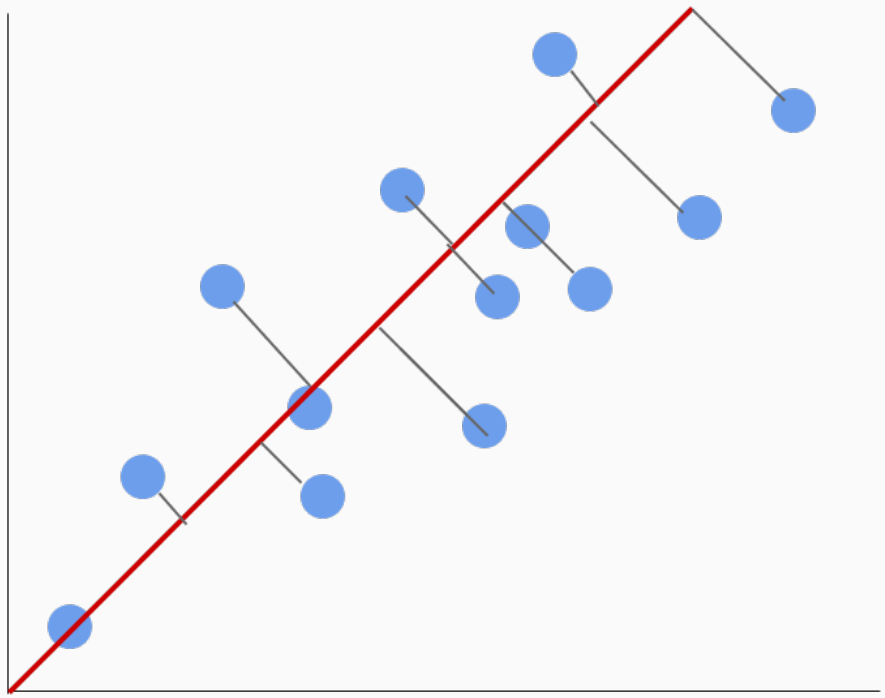


PCA Formulation



- Goal of PCA is to find k dimensions that minimize projection error of a n dimensional dataset

PCA Formulation



- Minimizing projection error is equivalent to maximizing variance of points after projection (Proof)

PCA Formulation

Let there be n samples of $x_i \in \mathbb{R}^d$ so $X \in \mathbb{R}^{n \times d}$

Goal: Find the first k principle components so the variance of X projected on these components is maximized

- Must first center X around the origin so variance is not skewed
    - Variance is defined relative to the mean
- Let $\bar{X} = X - \mu$

PCA Formulation

Goal: Find the best k principle components so the variance of X projected on these components is maximized

Because covariance is given by

$$\Sigma = \frac{1}{n}\Sigma_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T = \frac{1}{n}(X - \mu)^T(X - \mu)$$

let the sample covariance after $\bar{X}$ is projected onto v be

$$\Sigma = \frac{1}{n}v^T\bar{X}^T\bar{X}v$$

For now, we will constrain v to have unit norm so $v^Tv = 1$

PCA Formulation

Goal: Find the first principle component so the variance of X projected on these components is maximized

$$\max_{v : v^T v = 1} \frac{1}{n} v^T \bar{X}^T \bar{X} v$$

$$L(v) = \frac{1}{n} v^T \bar{X}^T \bar{X} v - \lambda(v^T v - 1)$$

$$\nabla L(v) = 2\bar{X}^T \bar{X} v - 2\lambda v = 0$$

$$\bar{X}^T \bar{X} v = \lambda v$$

v is an eigenvector of the covariance matrix

PCA Formulation

Goal: Find the first principle component so the variance of X projected on these components is maximized

$$\max_{v:v^T v=1} \frac{1}{n} v^T \bar{X}^T \bar{X} v$$

$$\bar{X}^T \bar{X} v = \lambda v$$

Substituting this back into the original objective $\frac{1}{n} v^T \bar{X}^T \bar{X} v$ gives $\frac{1}{n} v^T \lambda v = \frac{1}{n} \lambda v^T v$

$$\max_{v:v^T v=1} \frac{1}{n} v^T \bar{X}^T \bar{X} v = \max_{v:v^T v=1} \frac{1}{n} \lambda v^T v = \max \frac{\lambda}{n}$$

v is the eigenvector corresponding to the largest eigenvalue of the covariance matrix

PCA Formulation

Given n samples $x_i \in \mathbb{R}^d$

1. Compute the covariance matrix of the centered data

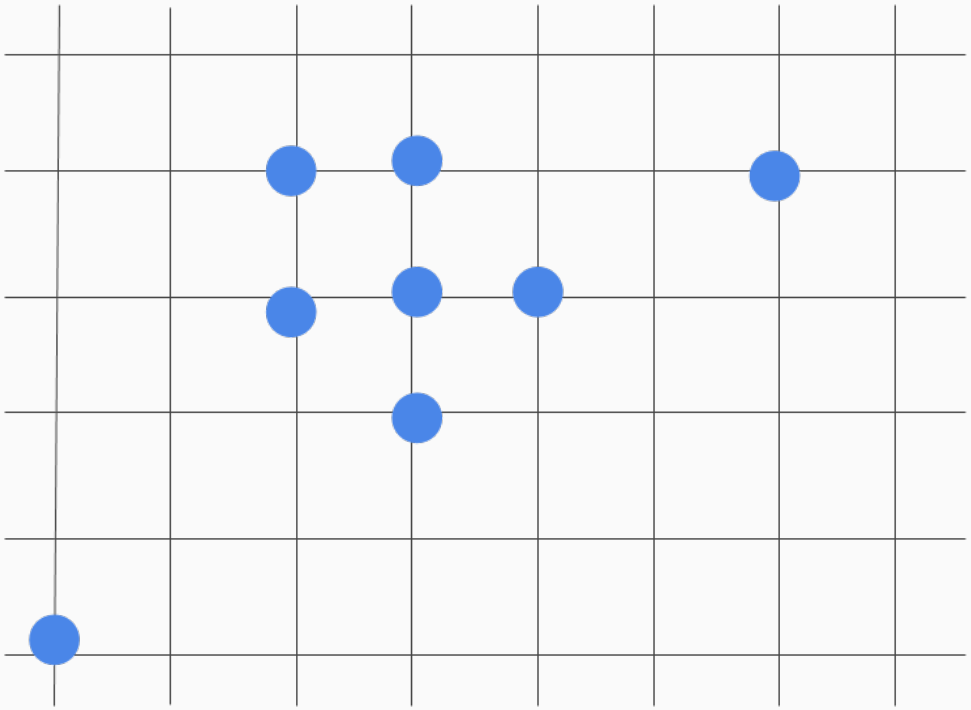
$$\Sigma = \frac{1}{n}\Sigma_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$$

2. Find the eigenvectors v_i and eigenvalues λ_i of the covariance matrix Σ
3. Choose the k largest eigenvalues. Their corresponding eigenvectors will form the projection matrix $P_{d \times k}$
4. Project each datapoint using P to get $z_i \in \mathbb{R}^k$

PCA Example

Let X contain the samples $x_i \in \mathbb{R}^2$ in it's rows

$$X = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 3 & 4 & 6 \\ 1 & 3 & 4 & 2 & 3 & 4 & 3 & 4 \end{bmatrix}^T$$



PCA Example

Let X contain the samples $x_i \in \mathbb{R}^2$ in it's rows

$$X = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 3 & 4 & 6 \\ 1 & 3 & 4 & 2 & 3 & 4 & 3 & 4 \end{bmatrix}^T$$

$$\mu = \frac{1}{8} \begin{bmatrix} 1+2+2+3+3+3+4+6 \\ 1+3+4+2+3+4+3+4 \end{bmatrix}^T = \begin{bmatrix} 3 & 3 \end{bmatrix}$$

$$X - \mu = \begin{bmatrix} -2 & -1 & -1 & 0 & 0 & 0 & 1 & 3 \\ -2 & 0 & 1 & -1 & 0 & 1 & 0 & 1 \end{bmatrix}^T$$

PCA Example

$$\Sigma = \frac{1}{m}\Sigma_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T$$

$$X - \mu = \begin{bmatrix} -2 & -1 & -1 & 0 & 0 & 0 & 1 & 3 \\ -2 & 0 & 1 & -1 & 0 & 1 & 0 & 1 \end{bmatrix}^T$$

$$\Sigma = \frac{1}{8}\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix} + \cdots + \begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0.75 \\ 0.75 & 1 \end{bmatrix}$$

PCA Example

$$\Sigma = \begin{bmatrix} 2 & 0.75 \\ 0.75 & 1 \end{bmatrix}$$

Σ has eigenvectors $v_1 = \begin{bmatrix} 0.8816746 \\ 0.47185793 \end{bmatrix}$ and $v_2 = \begin{bmatrix} -0.47185793 \\ 0.8816746 \end{bmatrix}$

Eigenvalues are $\lambda_1 = 2.4013$, $\lambda_2 = 0.5986$

PCA Example

Suppose we are trying to reduce this dataset to 1-D:

- Because the largest eigenvalue is λ_1, the projection matrix $P = v_1 = \begin{bmatrix} 0.8816746 \\ 0.47185793 \end{bmatrix}$

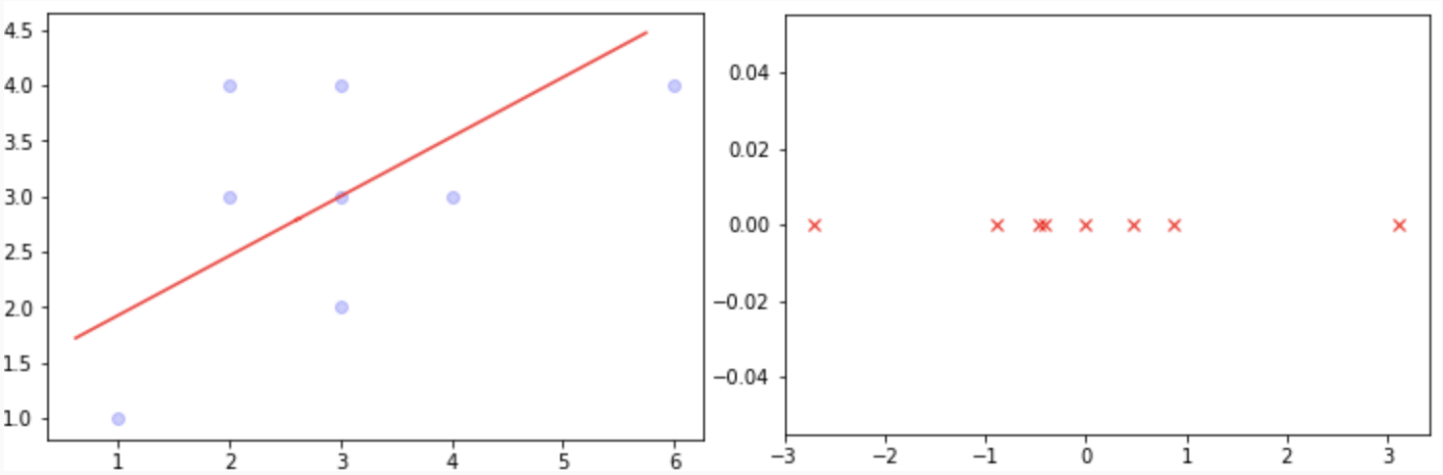
Transform $X = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 3 & 4 & 6 \\ 1 & 3 & 4 & 2 & 3 & 4 & 3 & 4 \end{bmatrix}^T$ using P

$$(X-\mu)P+\mu = \begin{bmatrix} -2.70 & -0.88 & -0.40 & -0.47 & 0.0 & 0.47 & 0.88 & 3.11 \end{bmatrix}$$

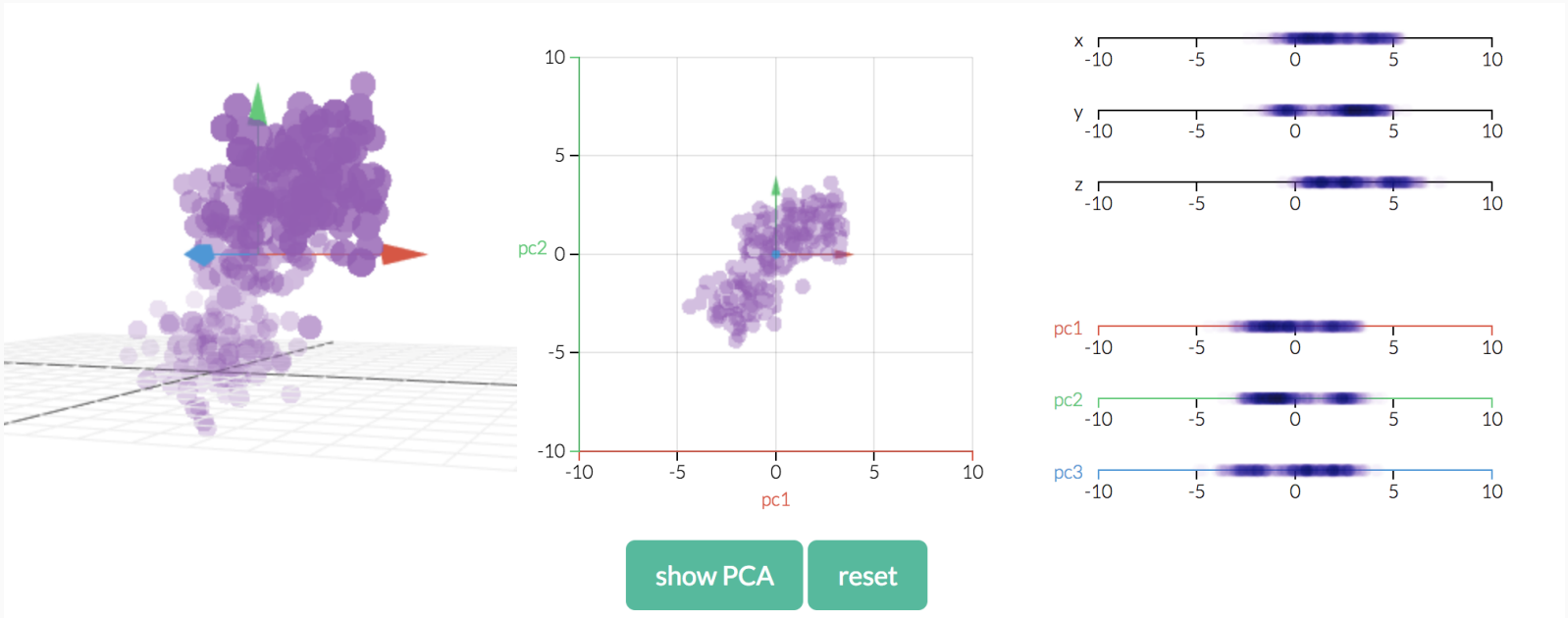
PCA Example

After the transformation, data in the 2-D space can be represented in the 1D space

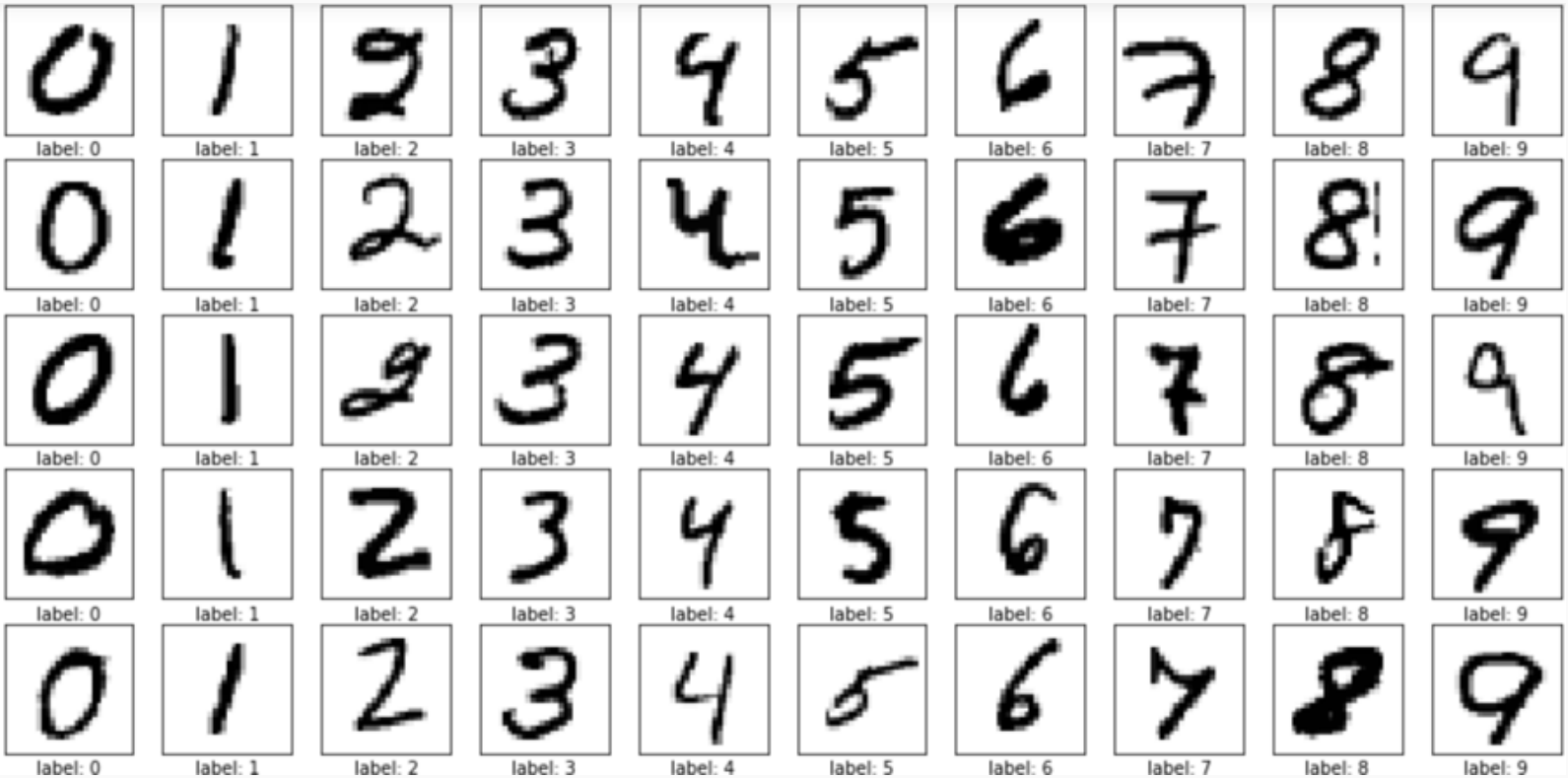
$$XP = \begin{bmatrix} -2.70 & -0.88 & -0.40 & -0.47 & 0.0 & 0.47 & 0.88 & 3.11 \end{bmatrix}^T$$



PCA Visualization



PCA Demo on MNIST



LDA - Linear Discriminant Analysis

PCA vs LDA

ML@B

PCA

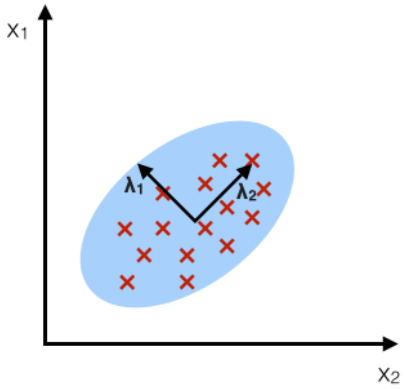
- Only uses the features of the data to find a lower dimensional subspace $\Rightarrow$ unsupervised technique
- Maximizes the variance of the data points in the projected space

LDA

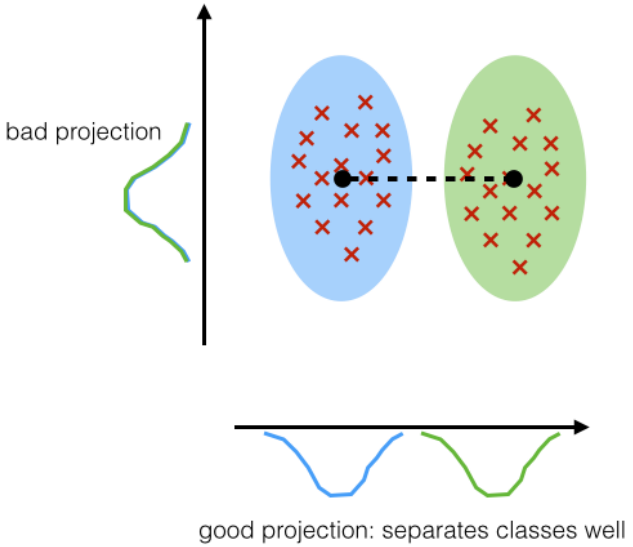
- Uses both features and data to reduce dimensions $\Rightarrow$ supervised technique
- Maximizes the separation between two classes in projected

PCA vs LDA

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



Goals of LDA

Find a lower dimensional space so when the data is projected on that space

- Distance between classes is maximized
- Distance of points within a class is minimized

LDA Formulation

Distance between classes is the distance between the **means of each class**

Suppose there are 2 classes:

$$\mu_1 = \frac{1}{N_1}\Sigma_{x \in C_1} x, \mu_2 = \frac{1}{N_2}\Sigma_{x \in C_2} x$$

After projecting onto a vector w:

$$\mu_1' = \frac{1}{N_1}\Sigma_{x \in C_1} w^T x, \mu_2' = \frac{1}{N_2}\Sigma_{w^T x \in C_2} x$$

The distance between means

$$\|\mu_1' - \mu_2'\|_2^2 = \|w^T \mu_1 - w^T \mu_2\|_2^2$$

LDA Formulation

Distance between classes is the distance between the **means of each class**

$$\|\mu_1' - \mu_2'\|_2^2 = \|w^T\mu_1 - w^T\mu_2\|_2^2$$

This can be put in matrix form as

$$(w^T\mu_1 - w^T\mu_2)^2 = w^T(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = w^T\Sigma_m w$$

where

$$\Sigma_m = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

LDA Formulation

We can check points in a class are close to each other by measuring the in-class variance

$$\sigma_i^2 = \Sigma_{x \in C_i} (w^T x - \mu_i')^2$$

Then the total in-class variance is

$$\Sigma_{i=1}^{c} \sigma_i^2$$

LDA Formulation

We can check points in a class are close to each other by measuring the in-class variance

$$\sigma_i^2 = \Sigma_{x \in C_i} (w^T x - \mu_i')^2$$

This can be put in matrix form as

$$\sigma_i^2 = \Sigma_{x \in C_i} (w^T x - w^T \mu_i)^2 = \Sigma_{x \in C_i} w^T (x - \mu_i)(x - \mu_i)^T w$$

$$= w^T (\Sigma_{x \in C_i} (x - \mu_i)(x - \mu_i)^T) w = w^T \Sigma_c w$$

where

$$\Sigma_c = \Sigma_{x \in C_i} (x - \mu_i)(x - \mu_i)^T$$

LDA Formulation

Suppose there are two classes c_1 and c_2

We can maximize the distance between classes and minimize the in-class variance with:

$$\text{argmax}_w \frac{\|\mu_1' - \mu_2\|_2^2}{\sigma_1 + \sigma_2} = \text{argmax}_w \frac{w^T \Sigma_m w}{w^T \Sigma_{c1} w + w^T \Sigma_{c2} w}$$

$$= \text{argmax}_w \frac{w^T \Sigma_m w}{w^T (\Sigma_{c1} + \Sigma_{c2}) w}$$

LDA Formulation

$$\text{argmax}_w \frac{w^T \Sigma_m w}{w^T (\Sigma_{c1} + \Sigma_{c2}) w} = \text{argmax}_w \frac{w^T \Sigma_m w}{w^T (\Sigma_c) w}$$

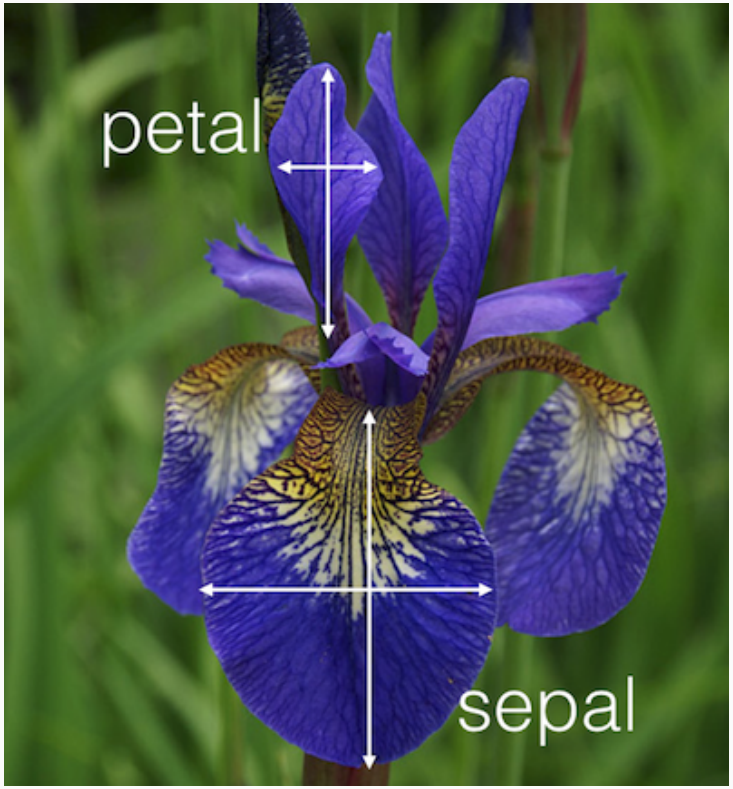
This takes the form of a generalized Rayleigh Quotient, so any w that maximizes this equation must satisfy

$$\Sigma_m w = \lambda \Sigma_c w = \Sigma_c \lambda w$$

$$\Sigma_c^{-1} \Sigma_m w = \lambda w$$

To transform the data into k dimensions, select the eigenvectors of $\Sigma_c^{-1} \Sigma_m$ that correspond to the first k dimensions as the projection basis

LDA Steps and Visualization on Iris



Linear vs Nonlinear Dimensionality Reduction

ML@B

Both PCA and LDA are linear methods of dimensionality reduction

- The resulting axes after projection are linear combinations of the initial axes
- Can kernelize PCA and LDA for a nonlinear transformation to lower dimensional subspace