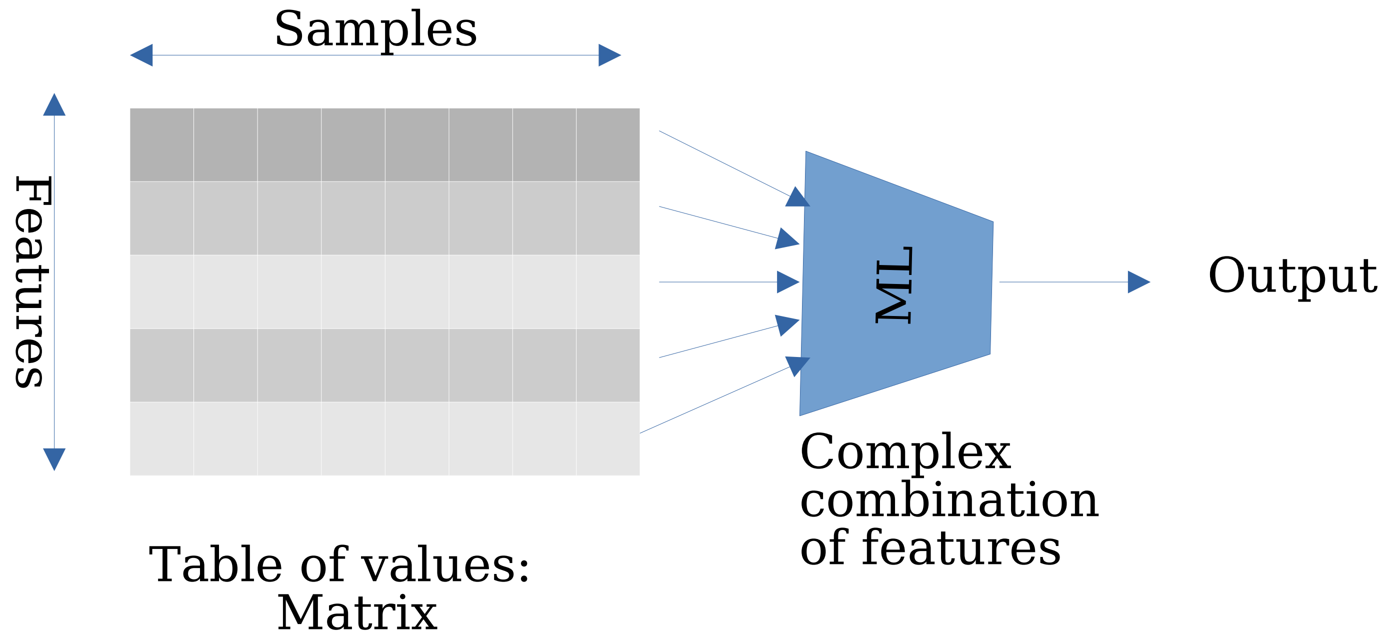
Feature selection

Chapter 5-6 of
Pattern Recognition
S. Theodoridis, K. Koutroumbas
4^{th} edition
Academic Press

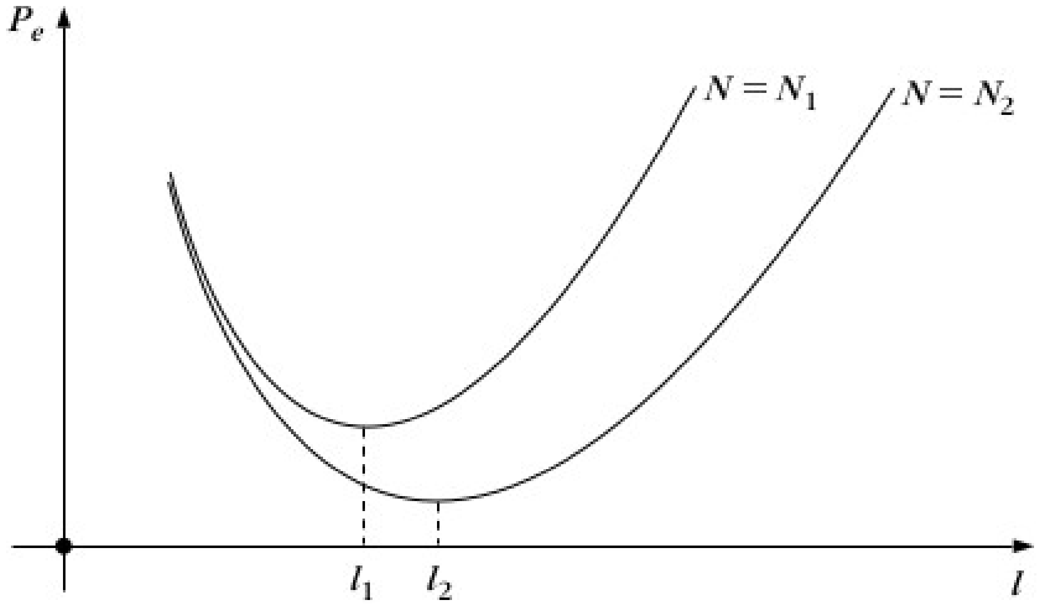
Data & Features



Goal: Make the learning easier, more robust, faster
Select among:
  Good, bad, noisy, unrelated, correlated… features

Features and information

Probability
of error



$N_1 < N_2$
Nb of
samples

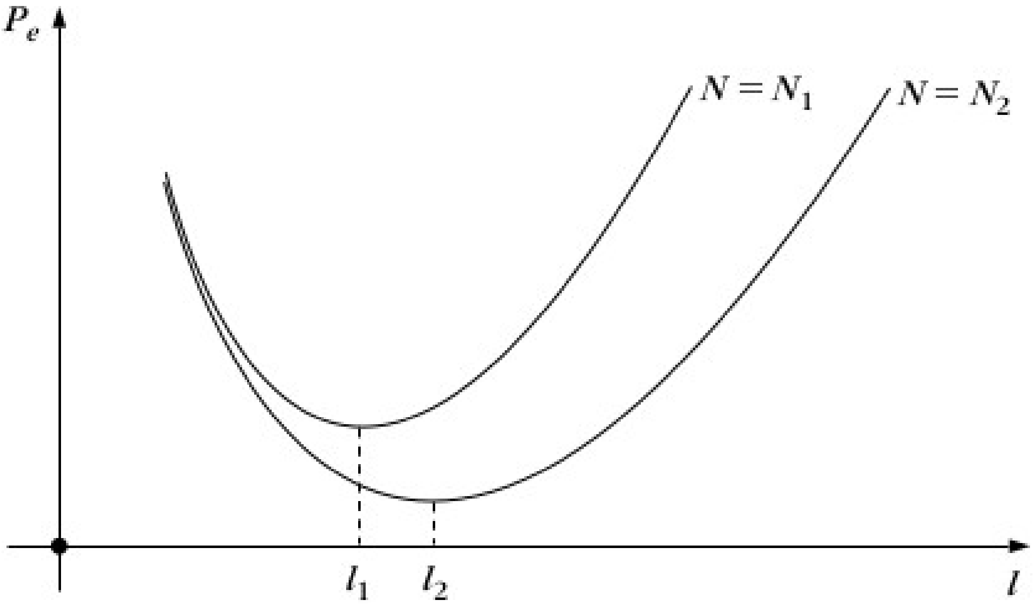
Nb of
features

Increase of
information

Overwhelming
Information masked,
Coincidences appear

Features and information

Probability
of error



Nb of
features

Solution: use expert knowledge & a-priori information
“inductive bias”

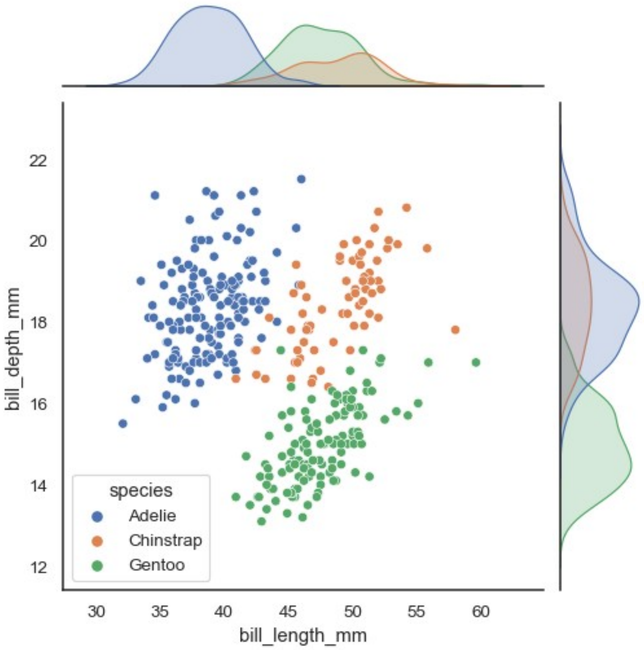
Data & Features

Samples

$$x = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \end{pmatrix} = \begin{pmatrix} x_1(1) & x_1(2) & \dots & x_1(N) \\ x_2(1) & x_2(2) & \dots & . \\ . & . & \dots & . \\ . & . & \dots & x_K(N) \end{pmatrix}$$

Features

E: expectation, sum over the samples / N



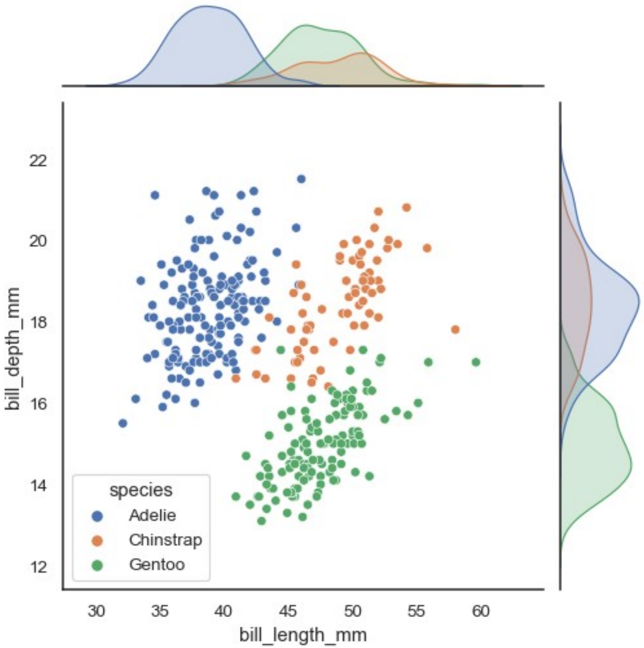
Mathematical model:
Sample: one realisation of a random variable

Data & Features

Samples

Features

$$x = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \end{pmatrix} = \begin{pmatrix} x_1(1) & x_1(2) & \dots & x_1(N) \\ x_2(1) & x_2(2) & \dots & . \\ . & . & \dots & . \\ . & . & \dots & x_K(N) \end{pmatrix}$$



!! what about categorical data ??
Not treated here. :-(
→ One-hot-encoding, embeddings

Pre-processing

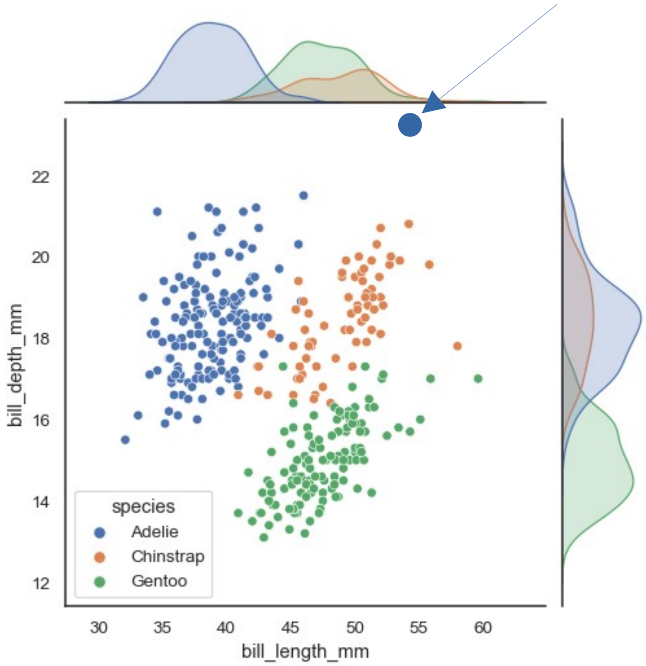
- Remove outliers
- Normalize data
- Fill in missing values
- More advanced transformations

  Use your knowledge of the data
- Future directions*: Self-supervised learning

* not in the book

Pre-processing

- Remove outliers



Pre-processing

- Normalize data

$$y_i = \frac{x_i - \mu_i}{\sigma_i}$$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \end{pmatrix} = \begin{pmatrix} x_1(1) & x_1(2) & \dots & x_1(N) \\ x_2(1) & x_2(2) & \dots & . \\ . & . & \dots & . \\ . & . & \dots & x_K(N) \end{pmatrix}$$

x1000 + 1000

→ Impact on the gradient and gradient step

Pre-processing

- Fill in missing values

Pre-processing

- More advanced transformations

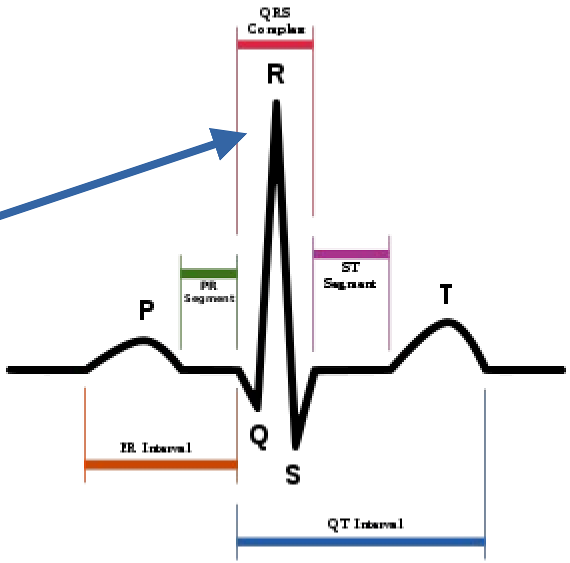
  Features are related together (time-series, image)
- Fourier transform, filtering, smoothing
- Use the time/space structure

Samples

$$x=\begin{pmatrix} x_1 \\ x_2 \\ . \\ . \end{pmatrix}=\begin{pmatrix} x_1(1) & x_1(2) & \dots & x_1(N) \\ x_2(1) & x_2(2) & \dots & . \\ . & . & \dots & . \\ . & . & \dots & x_K(N) \end{pmatrix}$$

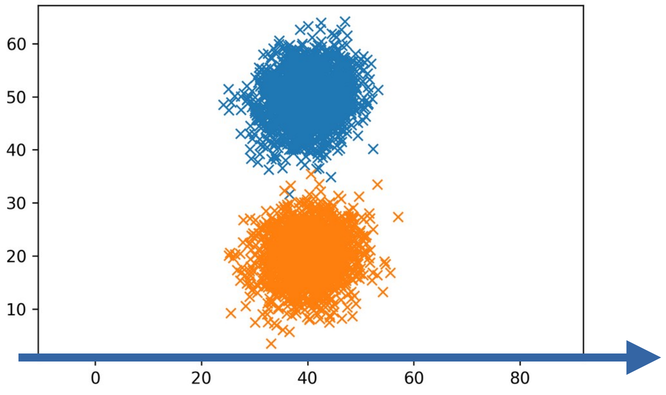
Features

May not be an outlier

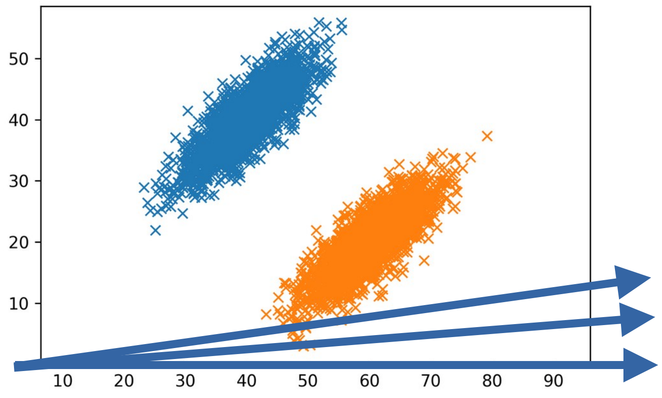


Feature selection

- Statistics on the features



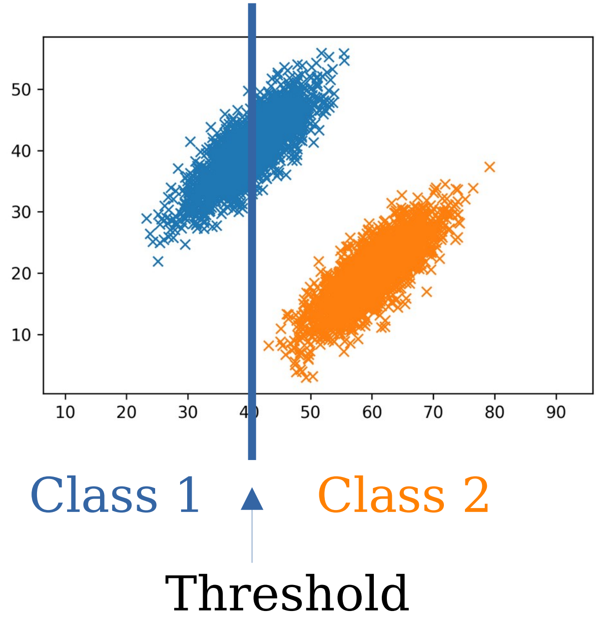
Useless feature

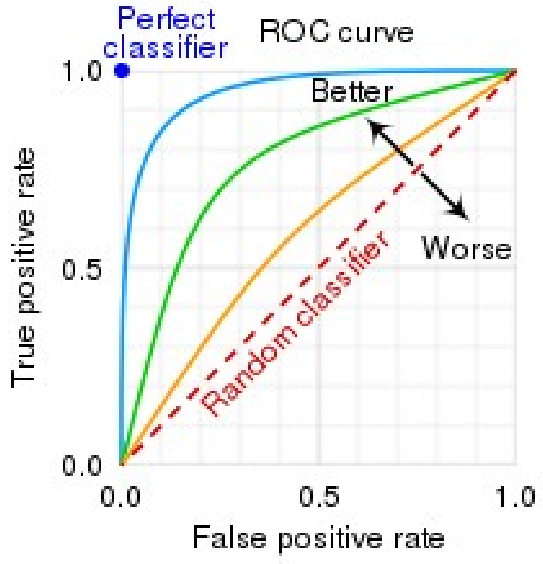


Correlated features

The ROC curve and AUC score

- Receiver Operating Characteristic





AUC Area Under Curve

→ Used to evaluate classifiers

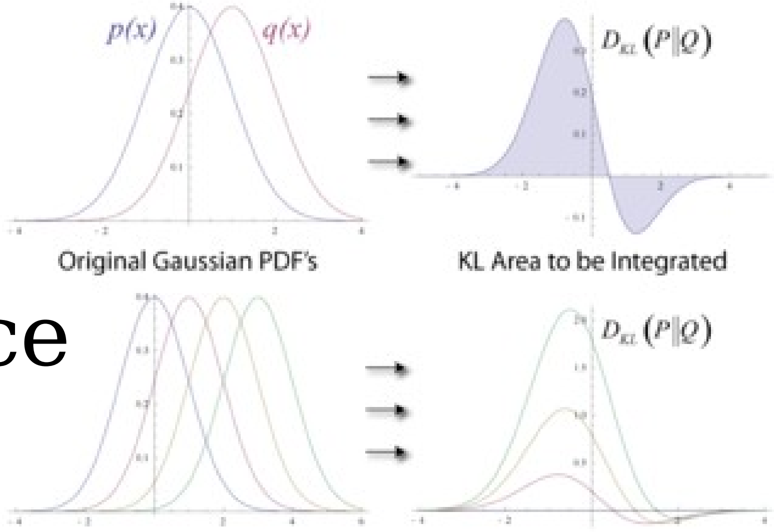
Divergence

- Comparing probability distributions
- 2 distributions p and q close if

$$D_{pq}(x) = \ln \frac{p(x)}{q(x)}$$

  is small
- Kullback-Leibler divergence

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

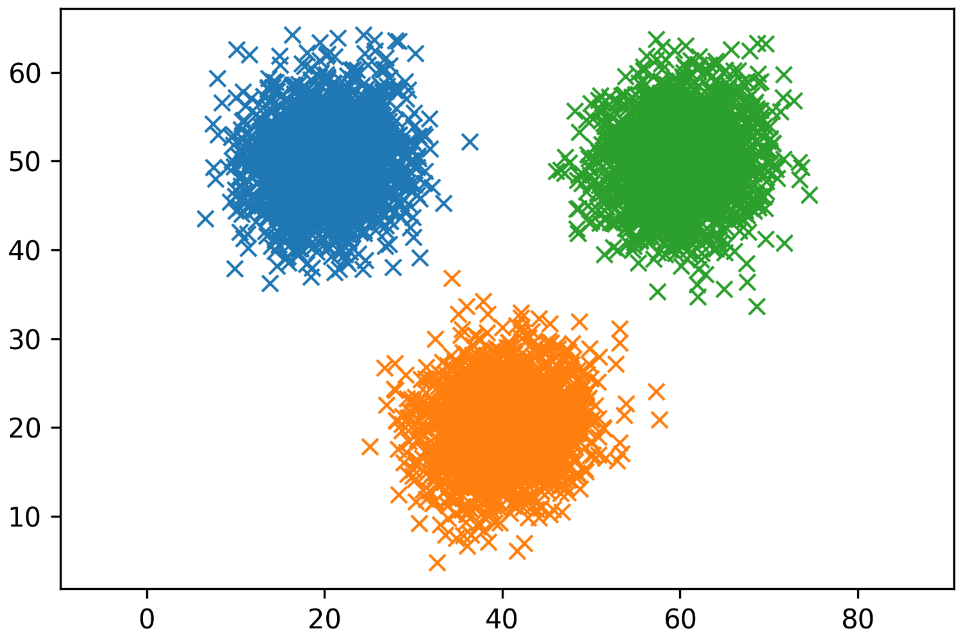


From Wikipedia

Important in machine learning! (loss functions)

Global separability of classes

- Scatter matrices & covariance matrix



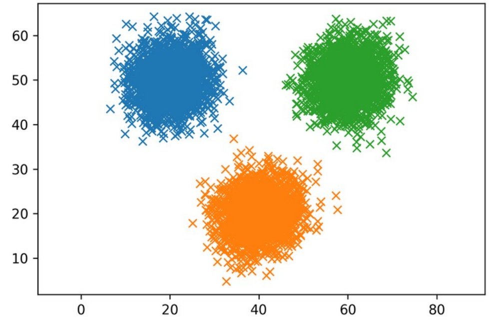
Notation

$$xx^T = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \end{pmatrix} \begin{pmatrix} x_1 & x_2 & . & . \end{pmatrix} = \begin{pmatrix} x_1 x_1 & x_1 x_2 & ... & x_1 x_K \\ x_2 x_1 & x_2 x_2 & ... & . \\ . & . & ... & . \\ . & . & ... & x_K x_K \end{pmatrix}$$

E: expectation, sum over the samples / N

Mathematical model:
Sample: one realisation of a random variable

Scattering



Class i

Covariance matrix

$$\Sigma_i = E_i[(x-\mu_i)(x-\mu_i)^T]$$

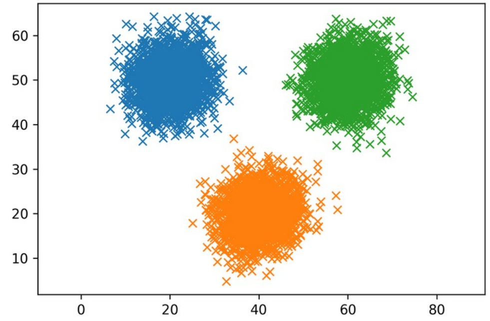
$$\Sigma_i = \begin{pmatrix} E_i[(x_1-\mu_i)(x_1-\mu_i)^T] & E_i[(x_1-\mu_i)(x_2-\mu_i)^T] & \dots \\ E_i[(x_2-\mu_i)(x_1-\mu_i)^T] & E_i[(x_2-\mu_i)(x_2-\mu_i)^T] & \dots \\ . & . & \dots \\ . & . & \dots \end{pmatrix}$$

E_i : Class i

Within class scattering:

$$S_w = \sum_{i=1}^{M} P_i \Sigma_i \qquad P_i = n_i / N$$

Scattering



Between-class

$$S_b = \sum_{i=1}^{M} P_i (\mu_i - \mu_0)(\mu_i - \mu_0)^T$$

$$\mu_0 = \begin{pmatrix} \mu_{01} \\ \mu_{02} \\ \vdots \\ \mu_{0l} \end{pmatrix}$$

$$J_3 = Trace(S_w^{-1} S_b)$$

$$S_w = \sum_{i=1}^{M} P_i \Sigma_i$$

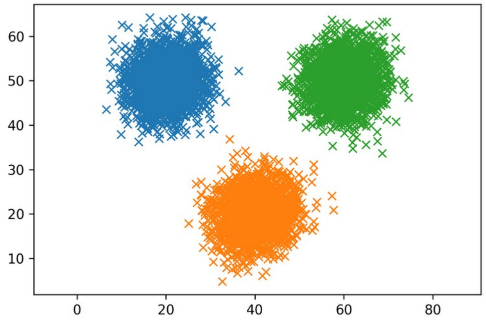
Trace: sum of the diagonal terms

Fisher's discriminant ratio

$$S_b = \sum_{i=1}^{M} P_i (\mu_i - \mu_0)(\mu_i - \mu_0)^T$$

$$S_w = \sum_{i=1}^{M} P_i E_i[(x - \mu_i)(x - \mu_i)^T]$$

$$J_3 = Trace(S_w^{-1} S_b)$$



$$FDR = \sum_{i=1}^{M} \sum_{j \neq i}^{M} \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2}$$

Linear discriminant Analysis

New features are linear combination of initial features

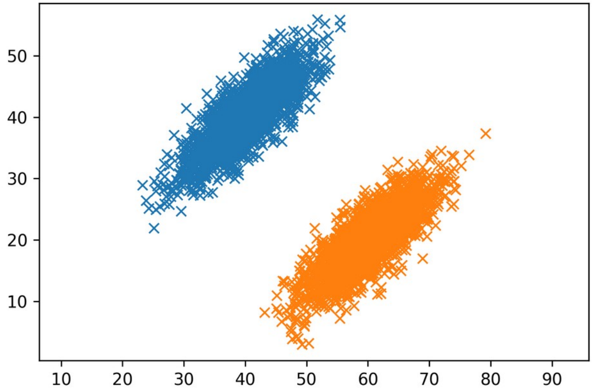
$$y = A^T x$$

A is rectangular, m x l with l<m

With y maximizing the FDR

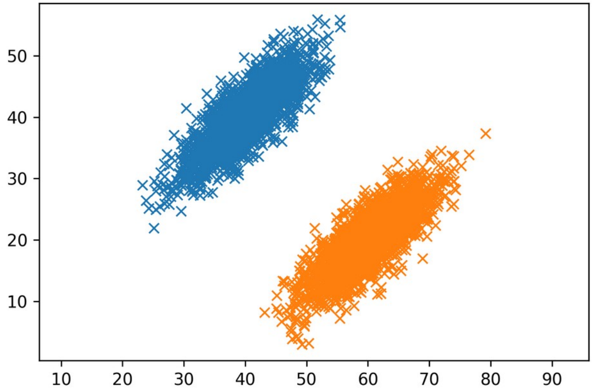
$$J_3 = Trace\left(S_w^{-1} S_b\right)$$

$$J_3(y) = Trace\left(\left(A^T S_w A\right)^{-1}\left(A^T S_b A\right)\right)$$



Linear discriminant Analysis

New features are linear combination of initial features



$$y = A^T x$$

Maximum: derivative is zero

$$J_{3y} = Trace\left(\left(A^T S_w A\right)^{-1}\left(A^T S_b A\right)\right)$$

$$\frac{\partial}{\partial A} J_{3y} = -2 S_w A \left(A^T S_w A\right)^{-1}\left(A^T S_b A\right)\left(A^T S_w A\right)^{-1} + 2 S_b A \left(A^T S_w A\right)^{-1} = 0$$

$$= -2 S_w A \left(A^T S_w A\right)^{-1}\left(A^T S_b A\right) + 2 S_b A$$

$$S_{yw} = A^T S_w A \qquad S_{yb} = A^T S_b A$$

$$S_{xw}^{-1} S_{xb} A = A S_{yw}^{-1} S_{yb}$$

Linear discriminant Analysis

$$y = A^T x$$

$$S_{xw}^{-1} S_{xb} A = A S_{yw}^{-1} S_{yb}$$

$$\hat{y} = B^T A^T x$$

$$J_{3y} = Trace\left(\left(B^{-T} S_w B\right)^{-1} \left(B^{-T} S_b B\right)\right) = Trace\left(B^{-1} S_w^{-1} S_b B\right) = Trace\left(S_w^{-1} S_b B B^{-1}\right)$$

$$= J_{3x}$$

Such that matrix B diagonalize the scatter matrices:

$B^{-1} S_{yw} B = I$ $\quad$ $B^{-1} S_{yb} B = D$ $\quad$ D diagonal

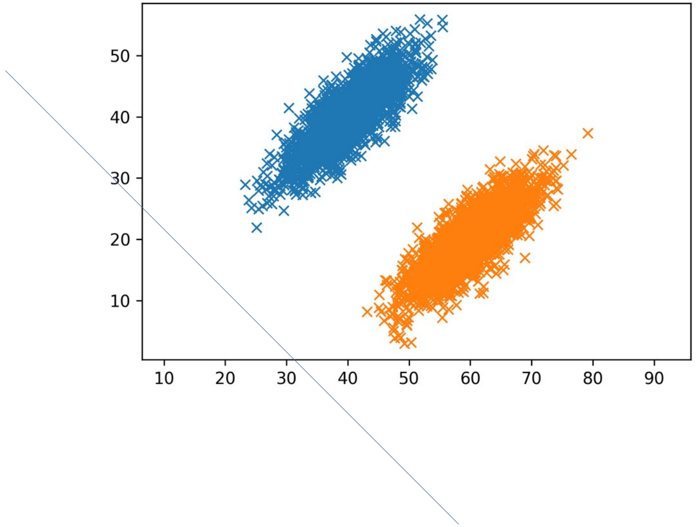
$$S_{xw}^{-1} S_{xb} AB = ABD \qquad S_{xw}^{-1} S_{xb} C = CD$$

Linear discriminant Analysis

$$\hat{y} = C^T x$$

$$S_{xw}^{-1} S_{xb} C_i = \lambda_i C_i$$

- Projection on the eigenvectors
- J_3 maximal with largest eigenvalues

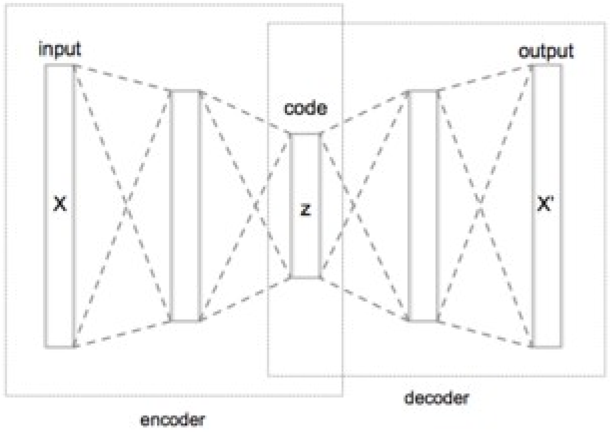


More recent feature selection / generation

Self-supervised learning

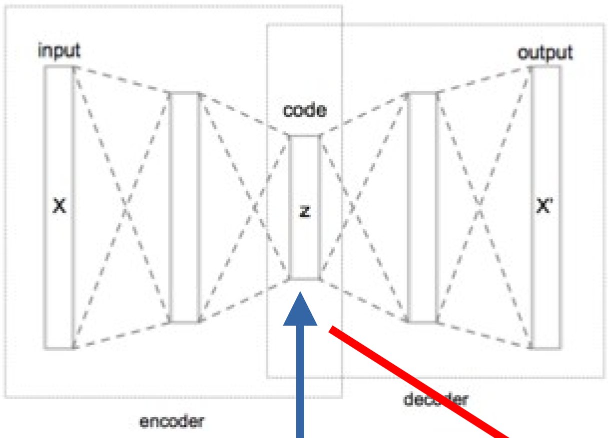
- Examples:
- Autoencoders
- BERT (text and language)
- Other tasks https://ai.googleblog.com/2021/09/discovering-anomalous-data-with-self.html

Principle: Modify the data and train to detect the modification or reconstruct the original data



Self-supervised learning

Auto encoder:
Reconstruct the input



Embedding space
Or latent space

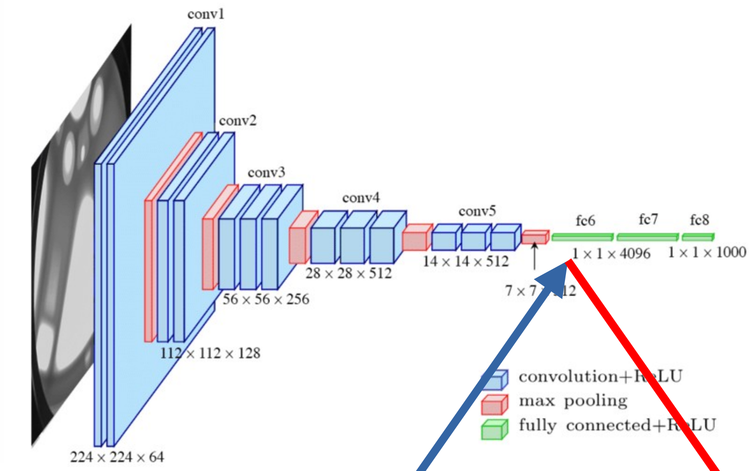
Feature vector!

ML model 2

* unsupervised
* semi-supervised
* supervised

Transfer learning

Train on task 1
Supervised



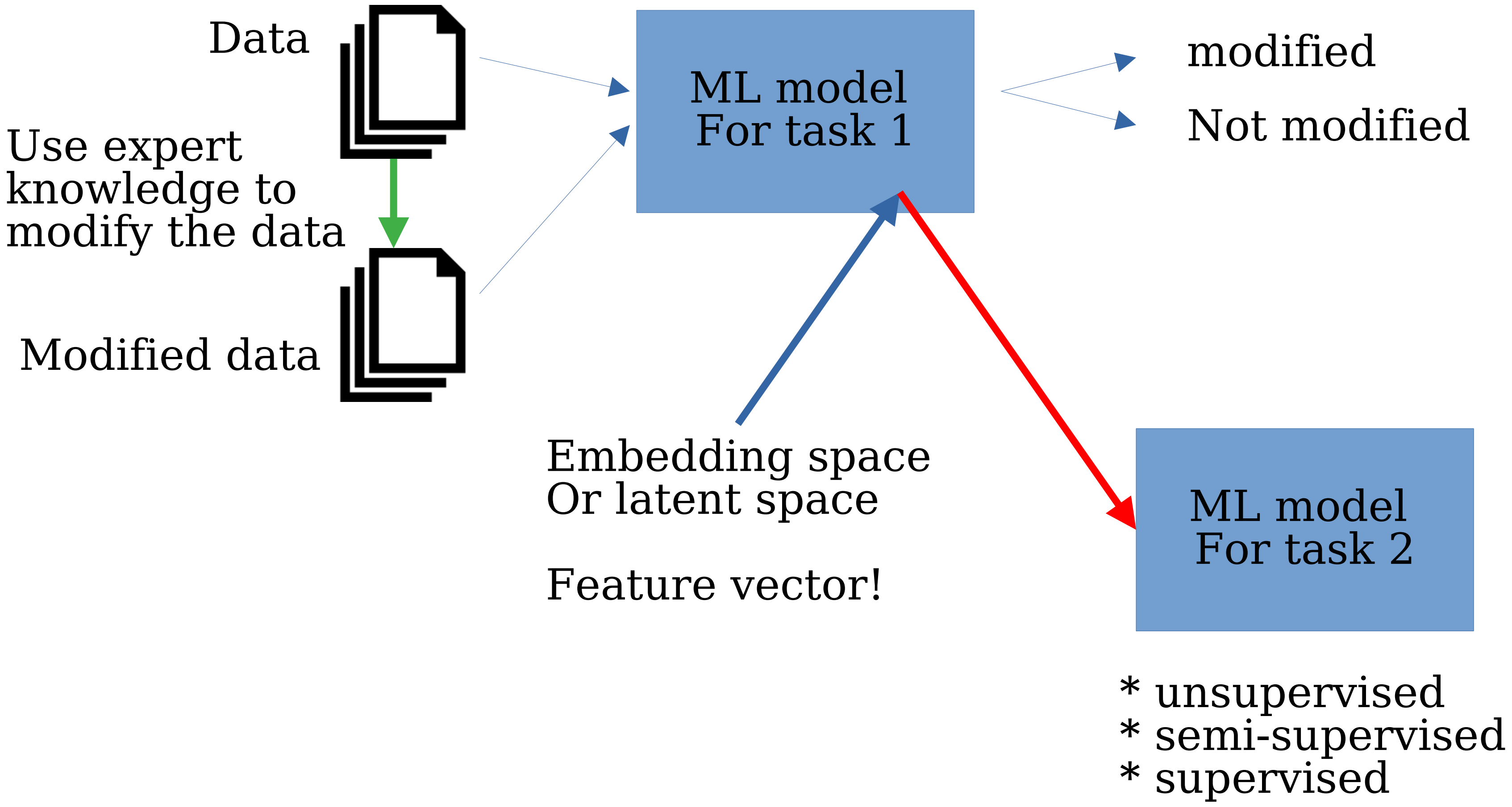
Embedding space
Or latent space

Feature vector!

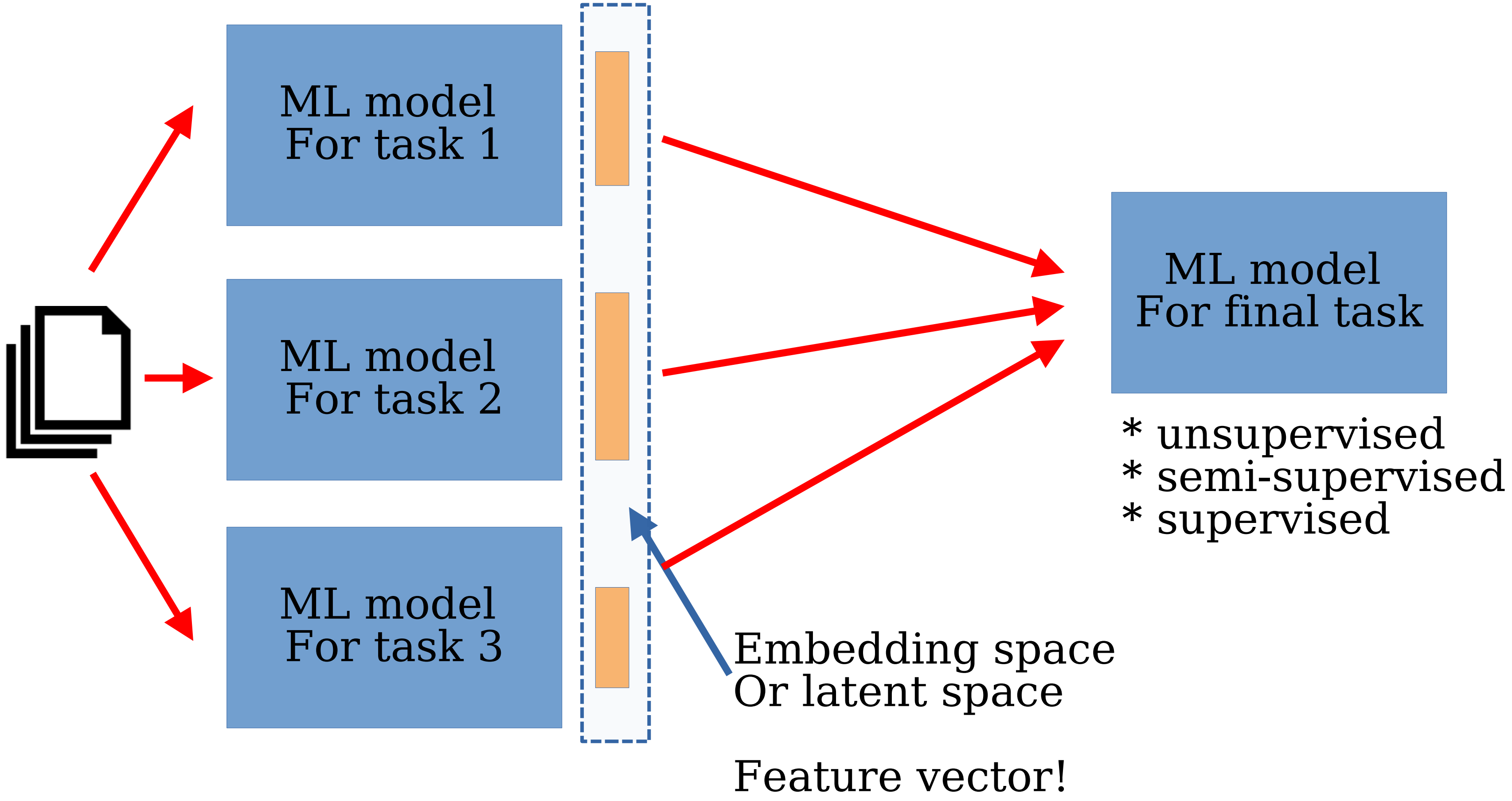
ML model
For task 2

* unsupervised
* semi-supervised
* supervised

Use expert knowledge



Meta-learning

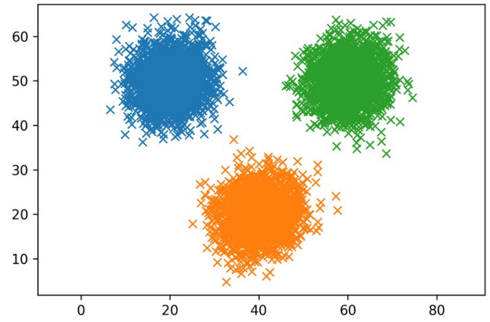


PCA

Samples

$$x = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \end{pmatrix} = \begin{pmatrix} x_1(1) & x_1(2) & \dots & x_1(N) \\ x_2(1) & x_2(2) & \dots & . \\ . & . & \dots & . \\ . & . & \dots & x_K(N) \end{pmatrix}$$

Features



Normalize the samples and write the covariance matrix $y_i = x_i - \mu_i$

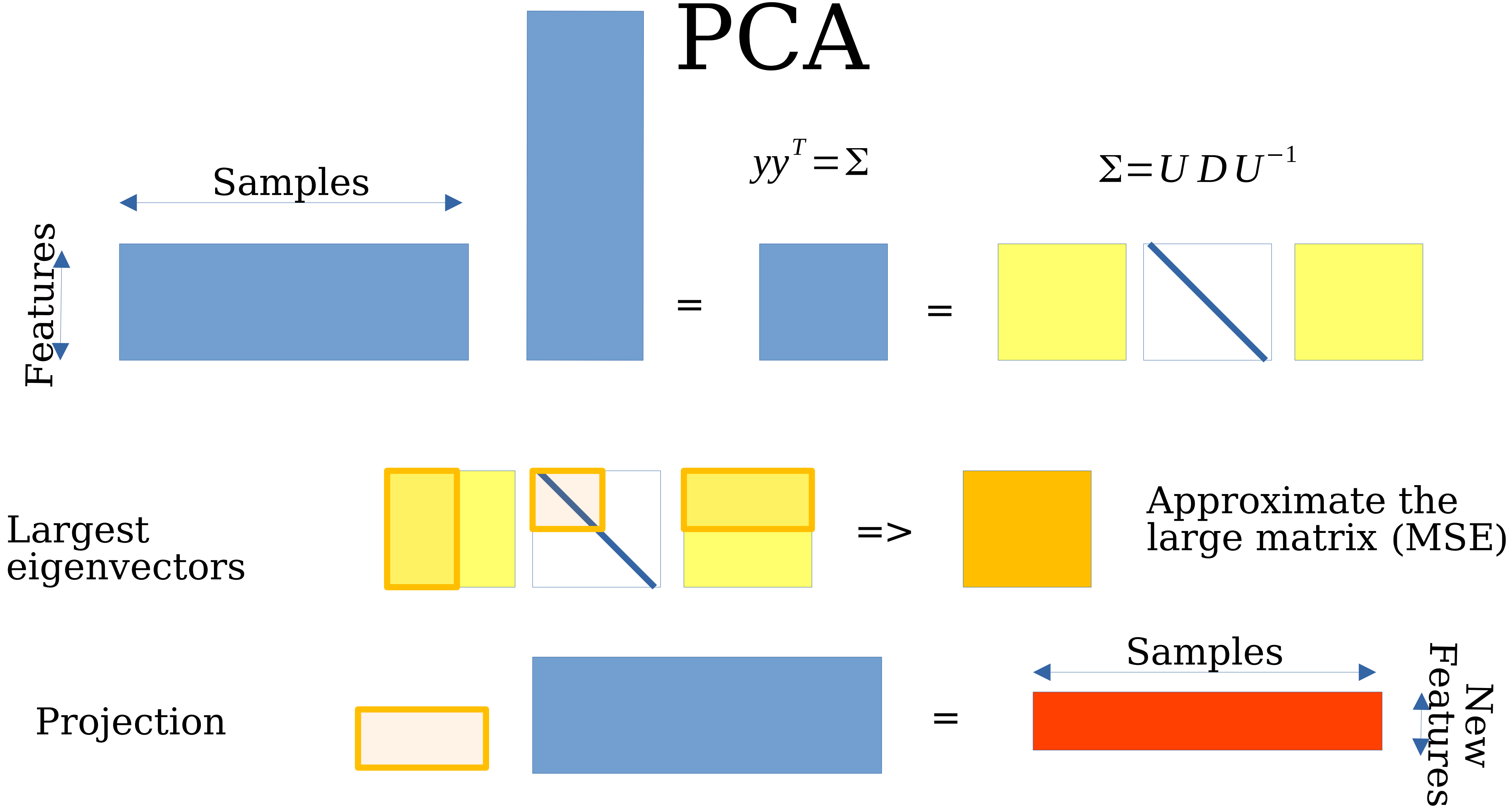
$$yy^T = \Sigma$$

Diagonalize the covariance matrix

$$\Sigma = U\,D\,U^{-1}$$

Note: no label information used

PCA



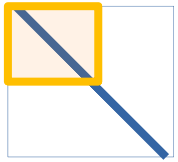
Samples
Features
$yy^T = \Sigma$
$\Sigma = U D U^{-1}$
=
=
Largest eigenvectors
=>
Approximate the large matrix (MSE)
Projection
=
Samples
New Features

PCA

$$Y = A^T X$$

Maximizing the covariance

$$YY^T = \Sigma_y = A^T XX^T A = A^T \Sigma_x A$$

If $A = U$ Matrix of eigenvectors, $\Sigma_y =$ 

- Zero covariance,
- maximal variance with largest eigenvalues

Note: $\underset{w}{argmax} \dfrac{w^T A w}{w^T w}$ Solution: w eigenvector with largest eigenvalue

PCA

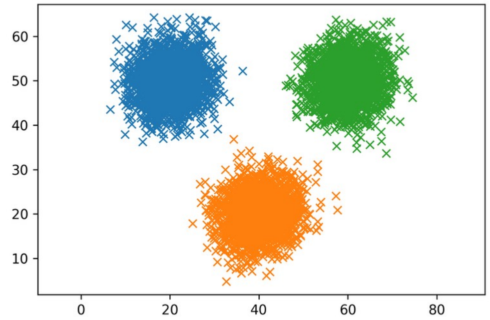
- Some visual examples

LDA & PCA

- Remarks: complex mix of initial features, can loose interpretability / explainability

  what the important features?

Kernel PCA



Laplacian eigenmaps