Computational Data Analysis

Machine Learning

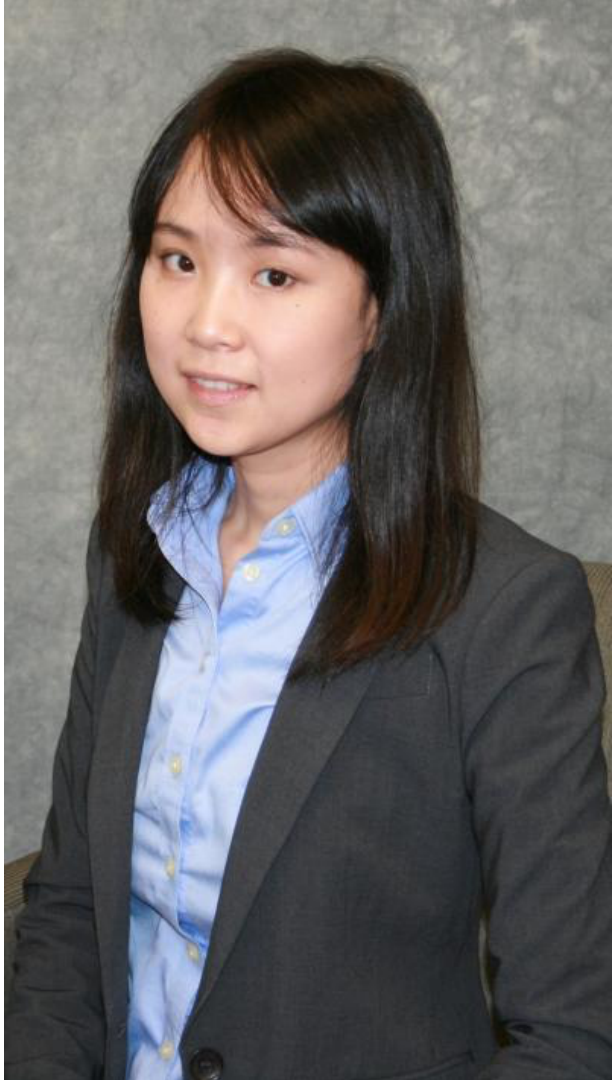
Yao Xie, Ph.D.

Associate Professor

Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering

Final Review

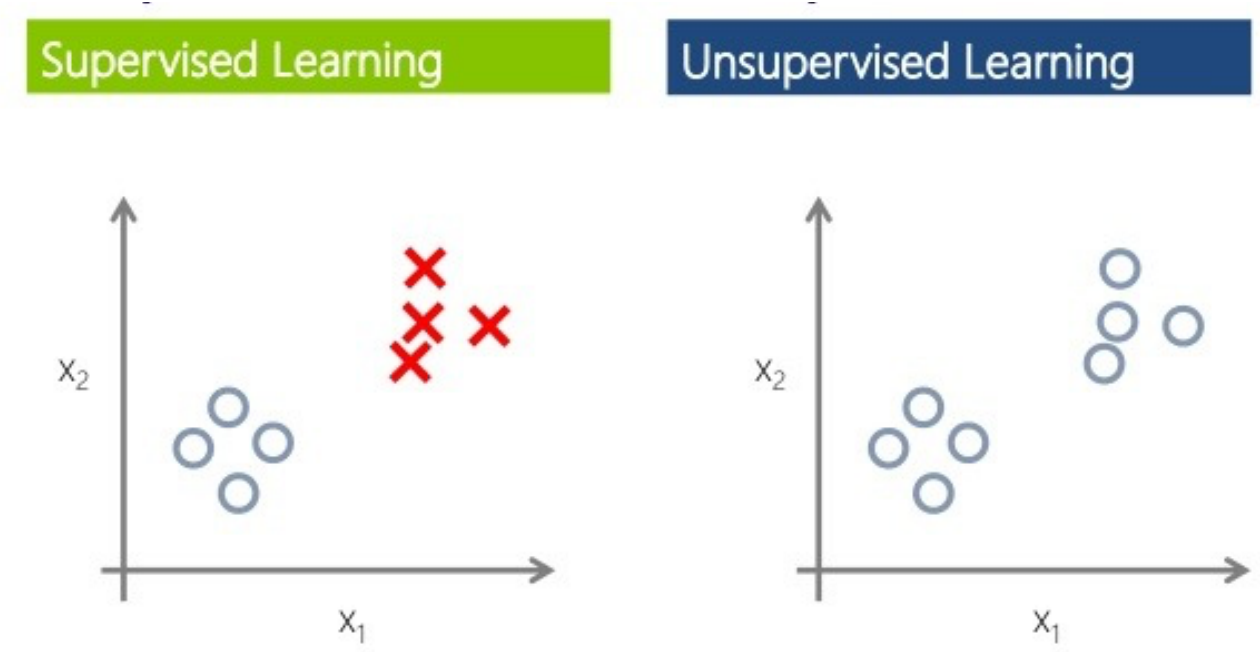


- Unsupervised learning

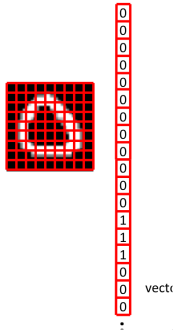
x

- Supervised learning

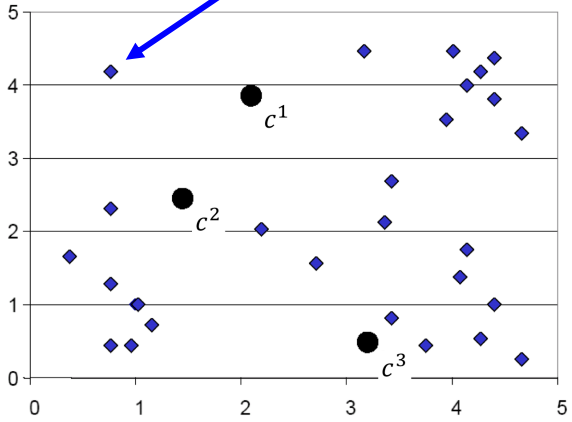
(x, y)



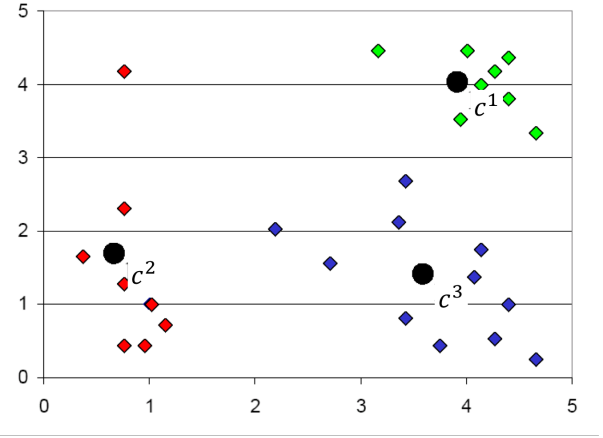
Clustering



Before clustering
Each dot is one sample (in vector space)



After clustering
Group assigned



k-means

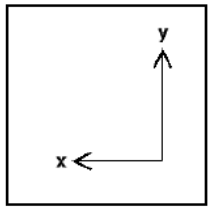
- Decide the cluster memberships of each data point, x^i, by assigning it to the nearest cluster center (cluster assignment)

$$\pi(i) = argmin_{j=1,\ldots,k} \left\| x^i - c^j \right\|^2$$

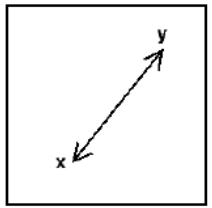
- Adjust the cluster centers (center adjustment)

$$c^j = \frac{1}{|\{i: \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

Replace this by other metrics to have k-mediod algorithm



Manhattan

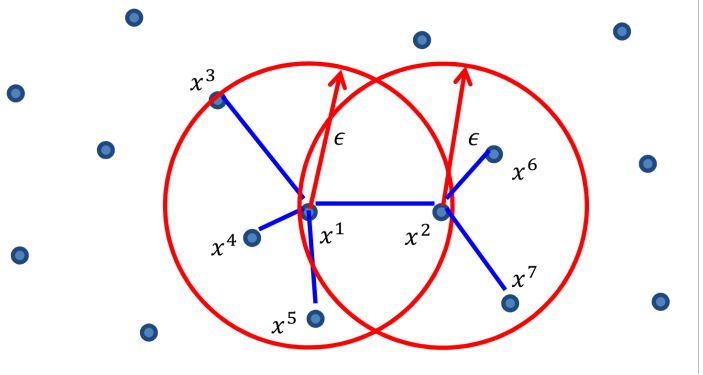


Euclidean

Spectral clustering

- Represent relationship between data using a graph

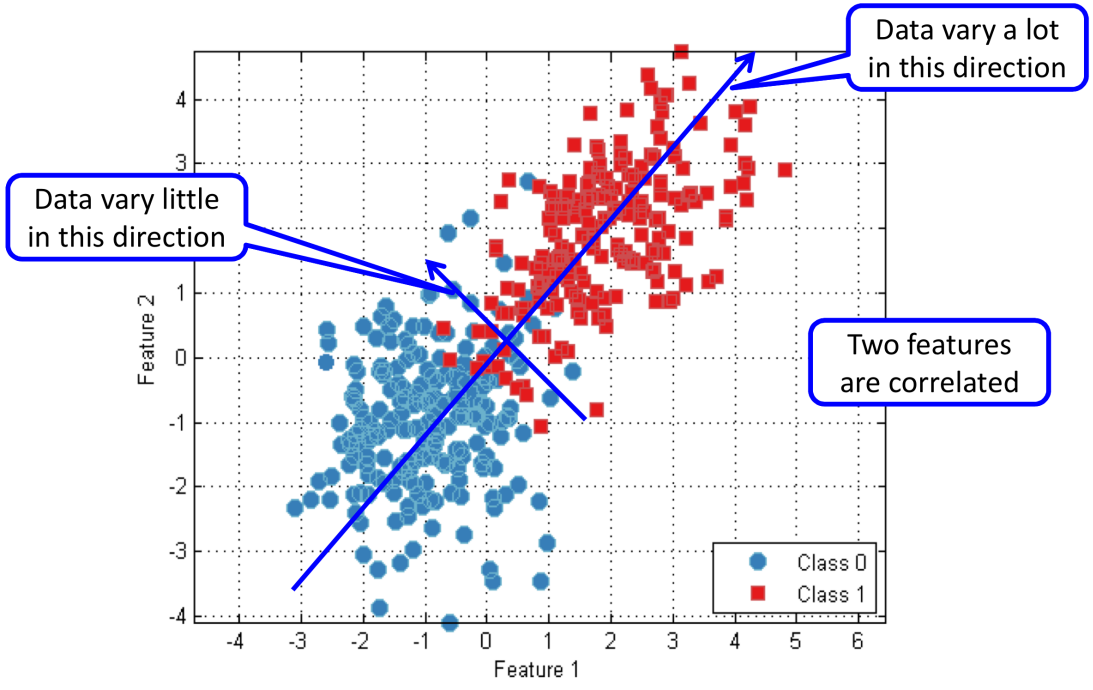
$$A^{ij} = \begin{cases} 1, \text{if } \|x^i - x^j\| \le \epsilon \\ 0, \quad \text{otherwise} \end{cases}$$



- Divide data into groups by cutting graph into several sub-graphs
- Advantage: can capture connected components

Dimensionality reduction

- Project high-dimensional data into lower dimensional components
- PCA: linear projection

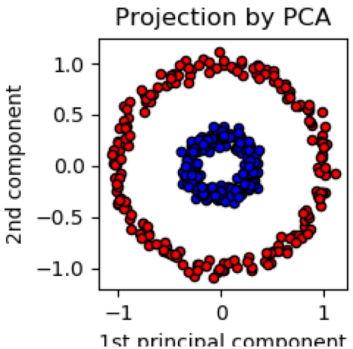


Comparison of PCA and KPCA

- PCA

$$\max_{w:\|w\|\leq 1} w^\top C w$$

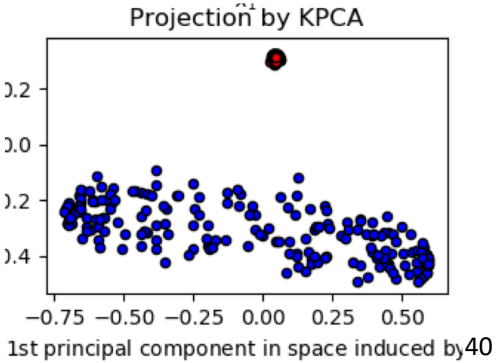
$$\underbrace{\left(\frac{1}{m}\sum_{i=1}^{m}\left(x^i-\mu\right)\left(x^i-\mu\right)^\top\right)}_{\text{covariance matrix } C}$$



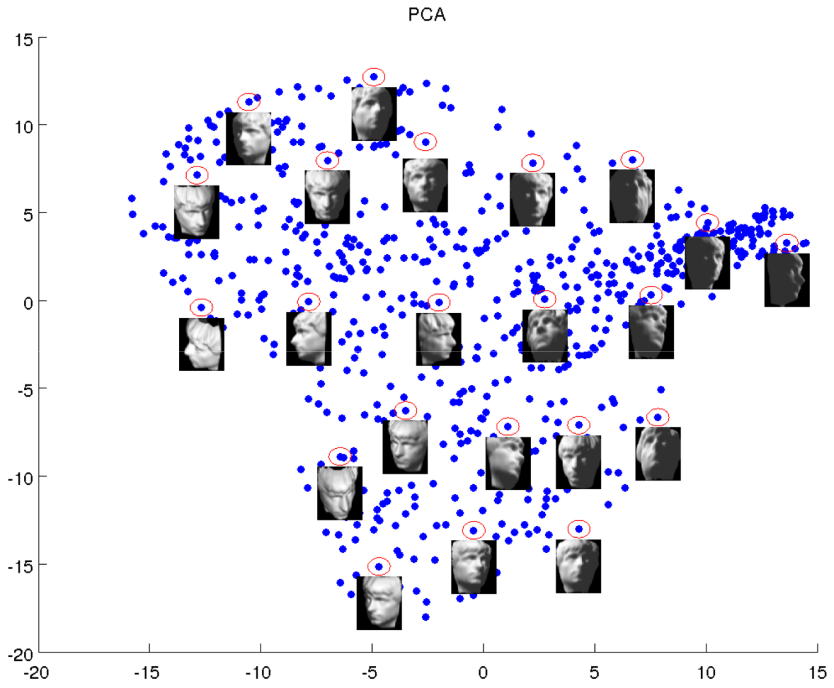
- Kernel PCA
  - Form Gram matrix K using kernel
  - Find leading eigenvector

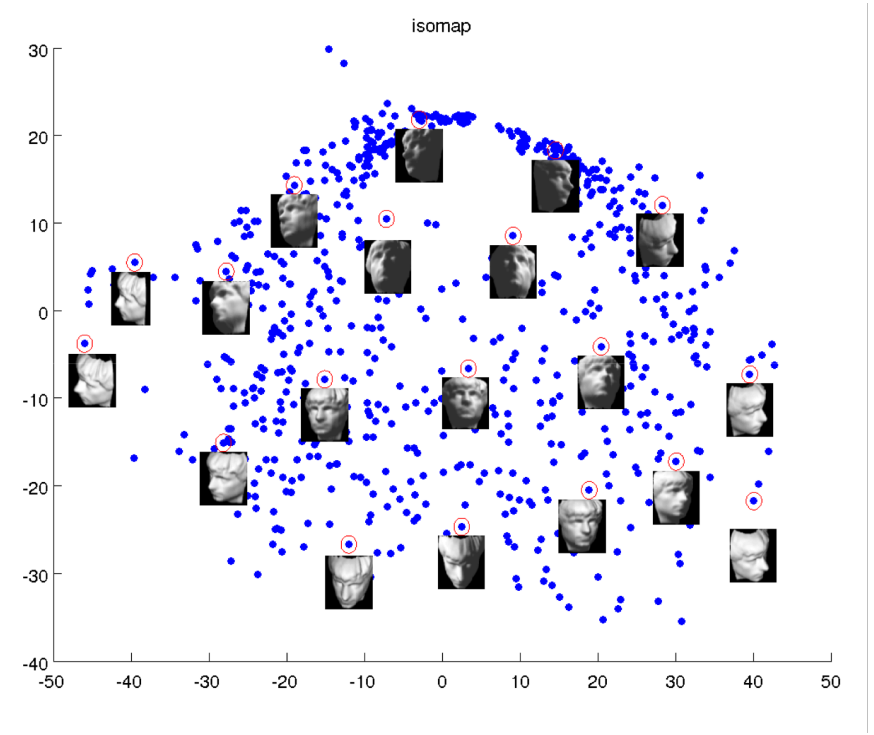
$$\max_{a:\|a\|\leq 1} a^T K a$$

$$w = \sum_i \alpha_k x^i = X\alpha$$



Isomap versus PCA





More interpretable

Density estimation

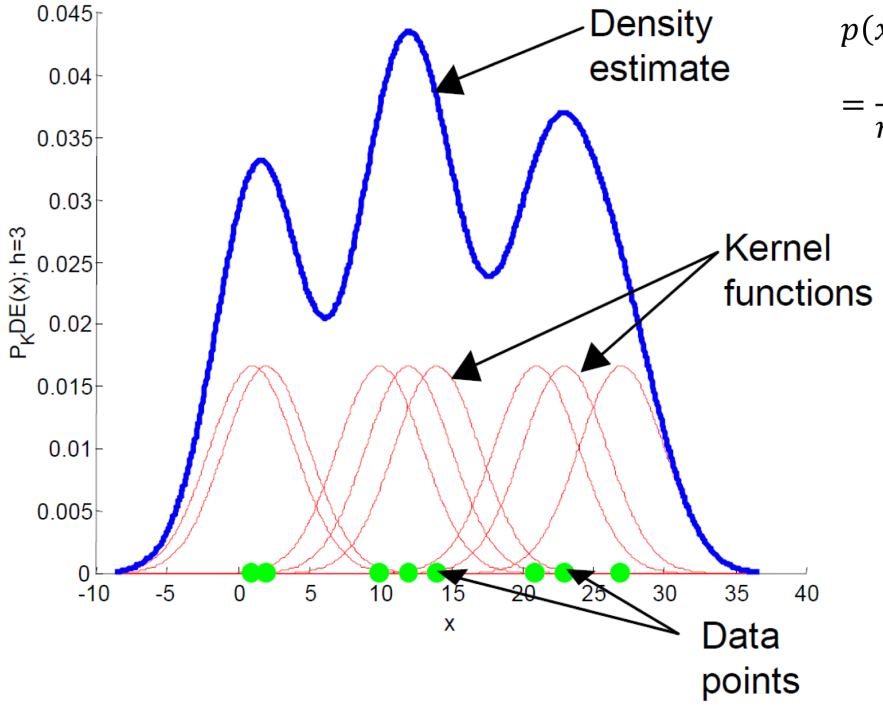
- Learn shape of the data distribution
- Kernel density estimator

$$p(x) = \frac{1}{m}\sum_{i}^{m}\frac{1}{h}K\left(\frac{x^i - x}{h}\right)$$

- Smoothing kernel function
  - $K(u) \geq 0,$
  - $\int K(u)du = 1,$
  - $\int uK(u) = 0,$
  - $\int u^2K(u)du \leq \infty$

- An example: Gaussian kernel $K(u) = \frac{1}{\sqrt{2\pi}}e^{-u^2/2}$

Example



$$p(x) = \frac{1}{m}\sum_{i}^{m}\frac{1}{h}K\left(\frac{x^i - x}{h}\right)$$

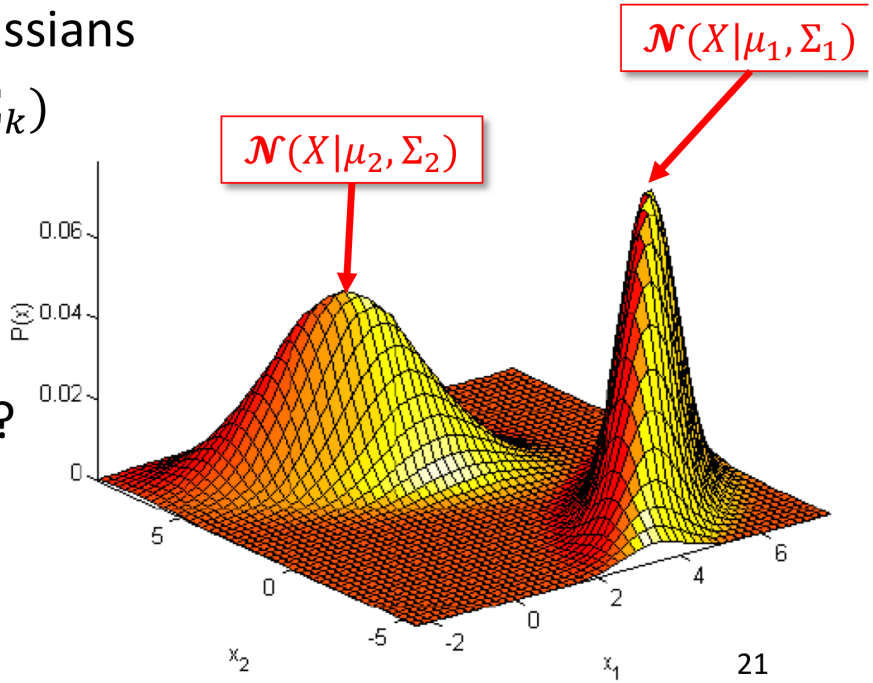
Gaussian mixture model

- Consider a mixture of K Gaussians
  - $p(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X|\mu_k, \Sigma_k)$

mixing proportion

mixture Component

- Parametric or noparametric?
- Learn $\pi_k \in (0,1), \mu_k, \Sigma_k$;



Iterate between two steps in EM

- E-step (estimate the probability that the sample comes from each of K components)

$$\tau_k^i = p\left(z_k^i = 1 \middle| D, \mu, \Sigma\right) = \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum_{k\prime=1}^{K} \pi_{k\prime} \mathcal{N}(x_i|\mu_{k\prime}, \Sigma_{k\prime})}$$

$$(k = 1 \dots K, i = 1 \dots m)$$

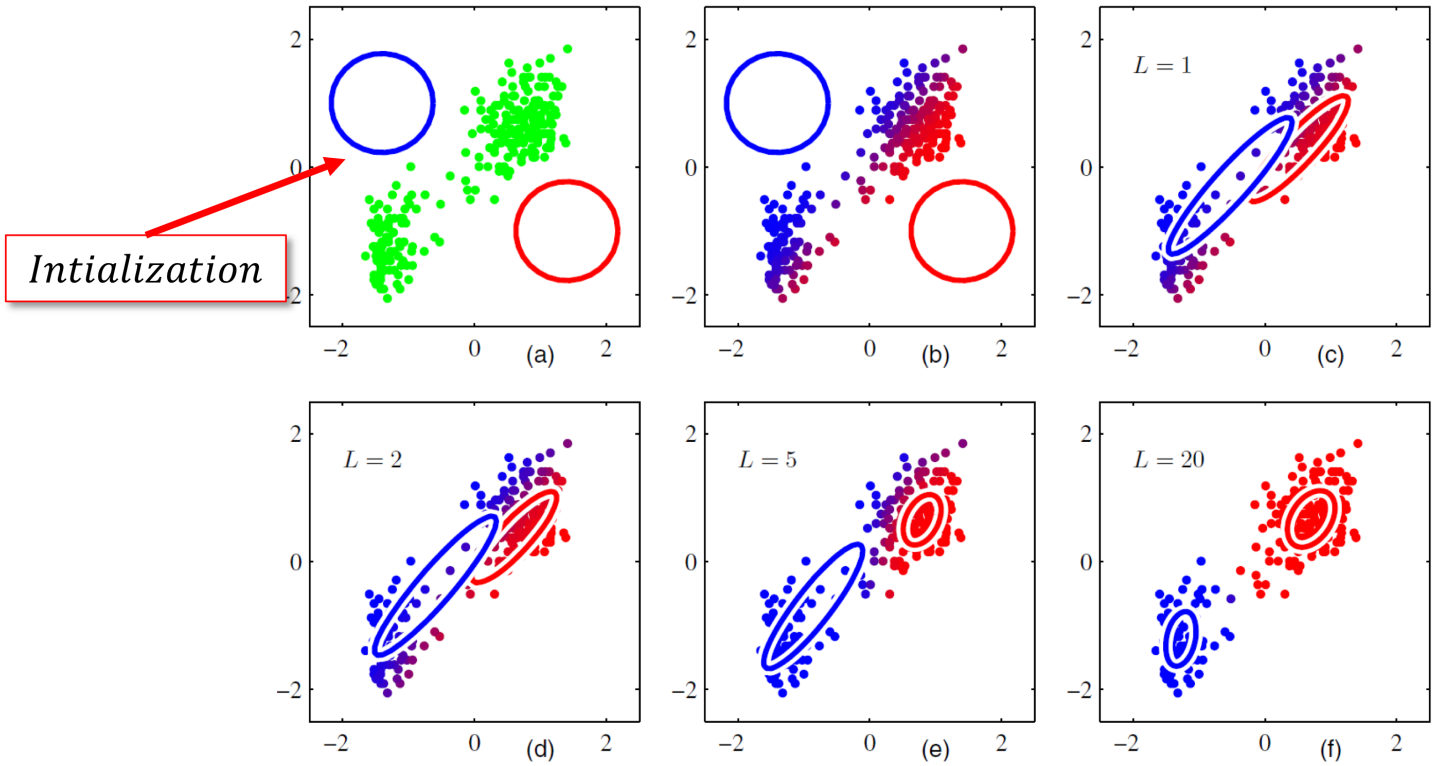
- M step (re-estimate parameters, using τ_k^i as "mixing" proportion)

$$\pi_k = \frac{\sum_i \tau_k^i}{m}, \qquad \mu_k = \frac{\sum_i \tau_k^i x^i}{\sum_i \tau_k^i}$$

$$\Sigma_k = \frac{\sum_i \tau_k^i \left(x^i - \mu_k\right)\left(x^i - \mu_k\right)^T}{\sum_i \tau_k^i}$$

$$(k = 1 \dots K)$$

EM is ”soft” k-means

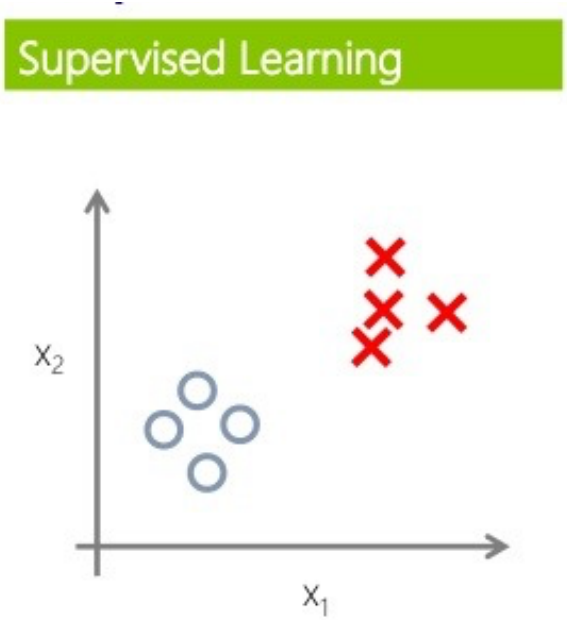


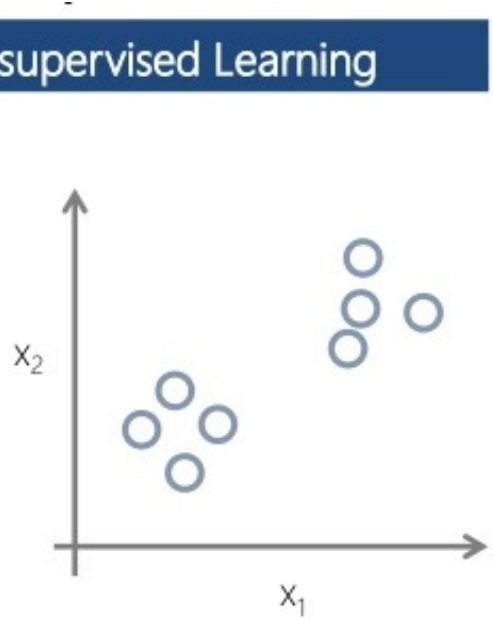
- Unsupervised learning

x

- Supervised learning

(x, y)

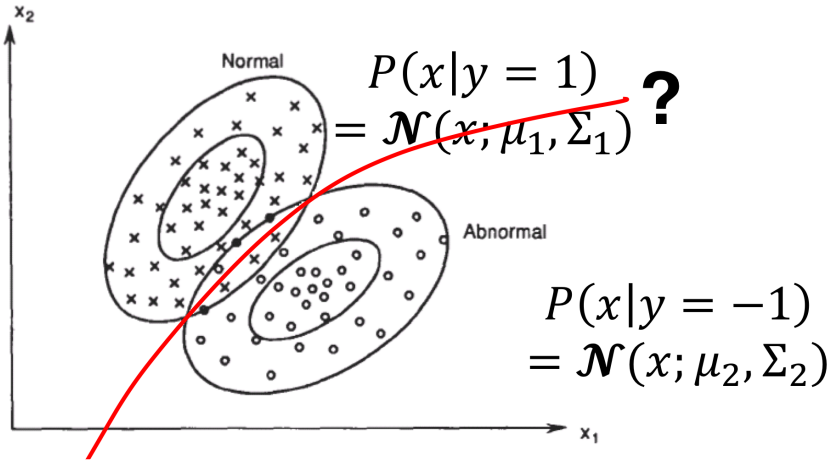




Know the label information

Naïve Bayes

- LDA and QDA?



- The poster probability of a test point

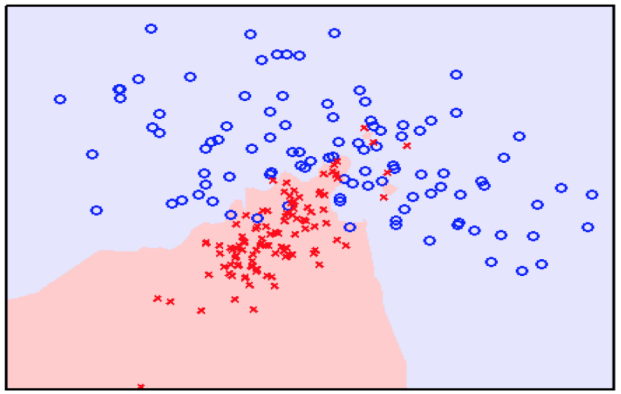
$$q_i(x) := P(y=i|x) = \frac{P(x|y)P(y)}{P(x)}$$

- Bayes decision rule:
  - If $q_i(x) > q_j(x)$, then $y = i$, otherwise $y = j$

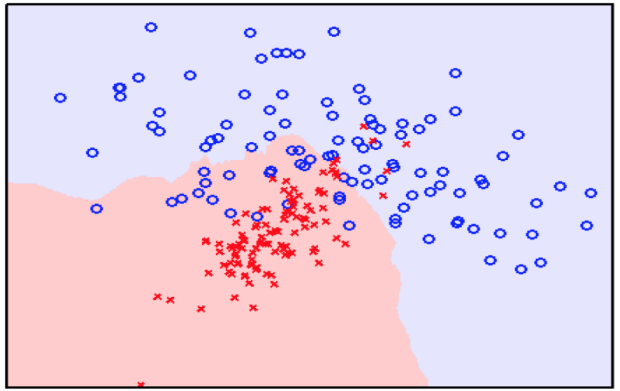
K-nearest neighbors

- Assign x.a label by taking the majority vote over k nearest training points that are closest to x

$$f_k(x) \coloneqq sign\left(\sum_{i \in I_k(x)} y_i\right)$$



K = 3



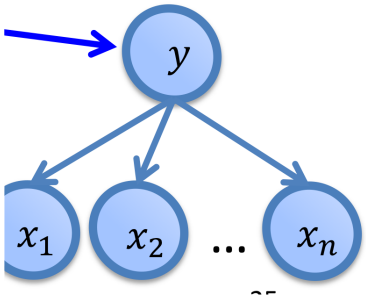
K = 51

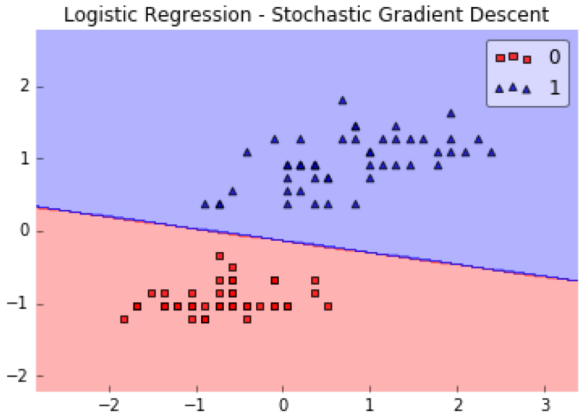
Logistic regression

- Logistic function

$$p(y = 1|x, \theta) = \frac{1}{1 + \exp(-\theta^T x)}$$

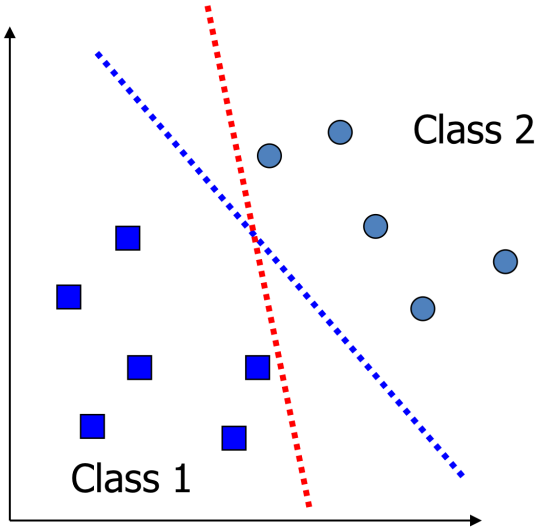
- Assume label y is a Bernoulli random variable with probability determined by features x
- Can be viewed as a one layer neural networks
- Logistic regression gives linear decision boundary





Support vector machine

- Optimal linear decision boundary to have largest margin



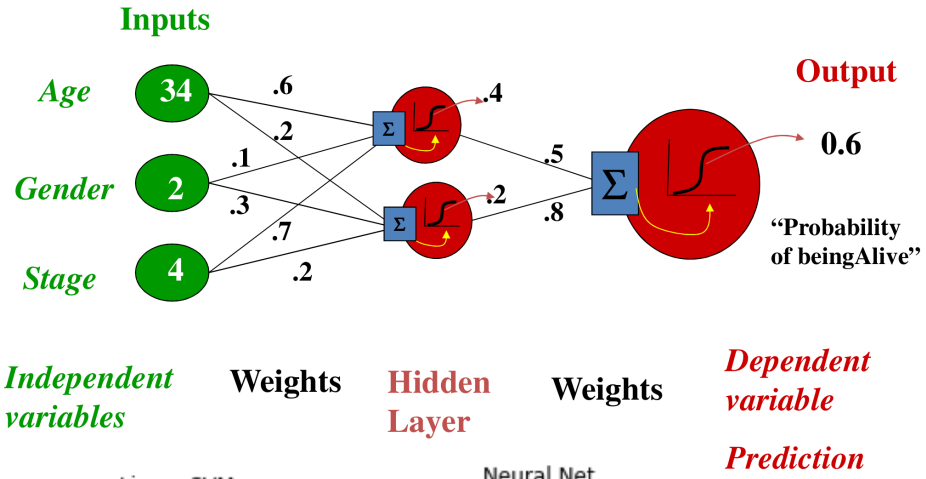
$$\min_{w,b} \|w\|^2$$
$$s.t.\ y^i\left(w^\top x^i + b\right) \geq 1, \forall i$$

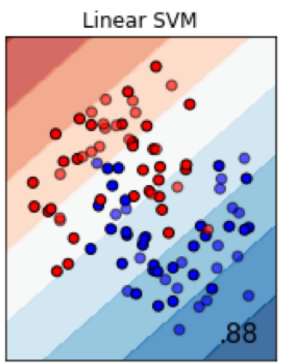
Dual formulation

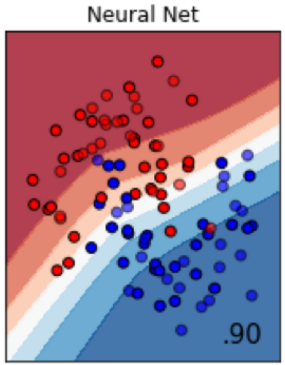
$$L(w,\alpha,\beta) = \sum_i^m \alpha_i - \frac{1}{2}\sum_{i,j}^m \alpha_i\alpha_j y^i y^j \left(x^{i\top} x^j\right)$$
$$s.t.\ \alpha_i \geq 0, i = 1,\dots,m$$
$$\sum_i^m \alpha_i y^i = 0$$

Neural networks

- Build complex non-linear decision boundary

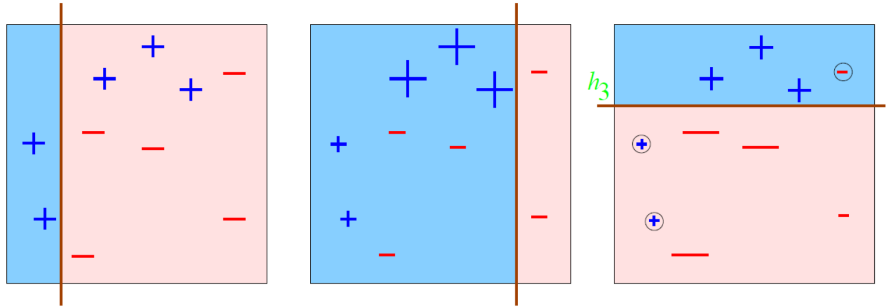


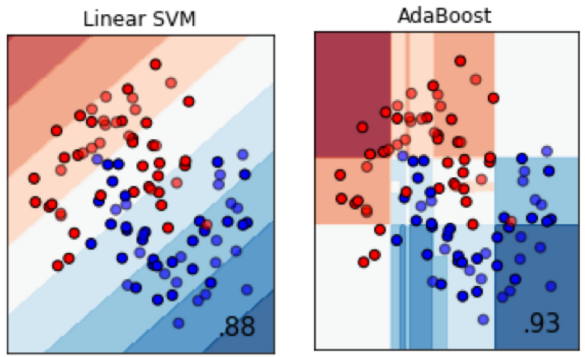




Ada-boosting

- Build complex decision boundary from simple decision stumps
- Each decision stump only focus on one dimension
- Weights to combine decision stumps are calculated automatically

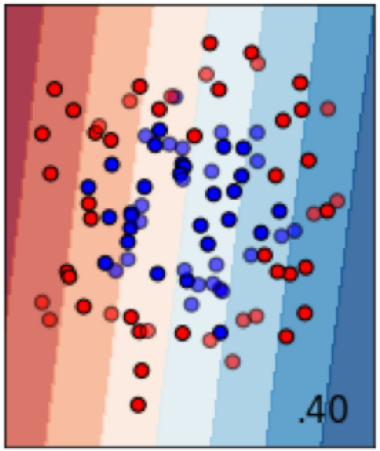




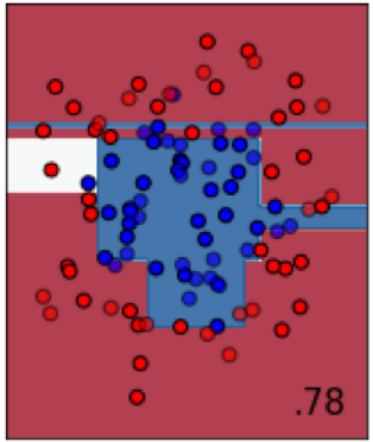
Random forest

- Regression (classification) tree: partition feature space into a set of rectangles, and fit a simple regression (classification) model in each region
- Random forest is “average” (majority vote) of a bunch of trees

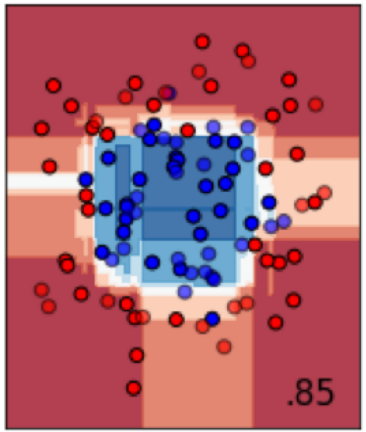
SVM



Decision tree



Random forest

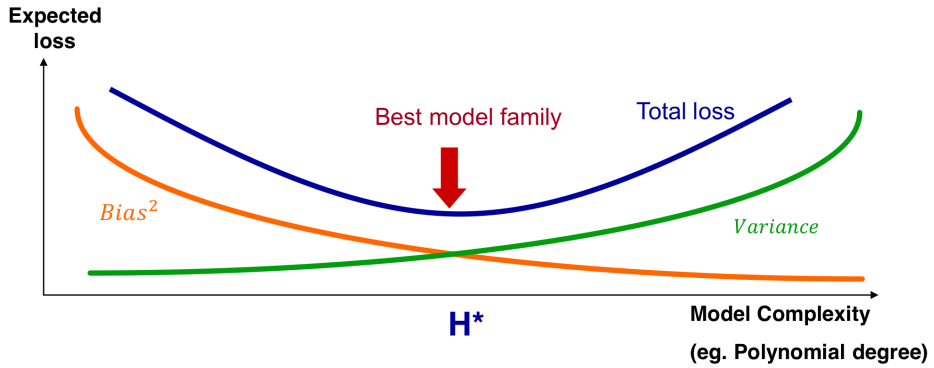


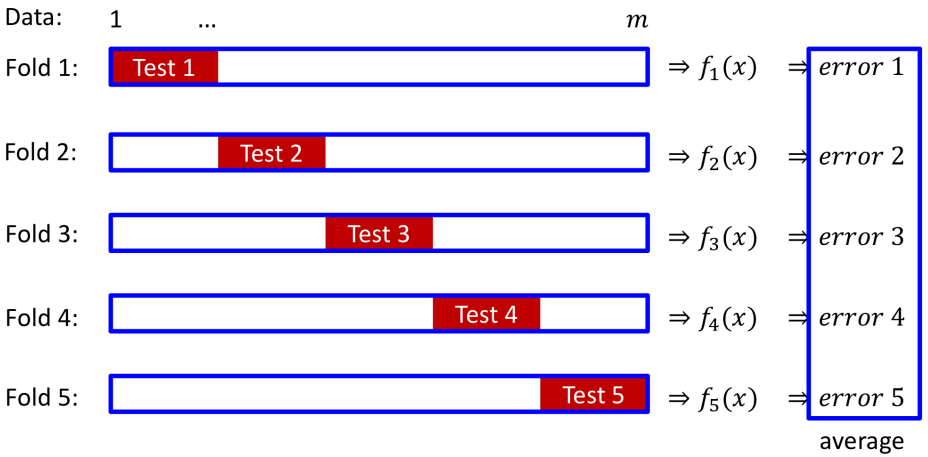
Pros and cons

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲ *= good,* ◆ *=fair, and* ▼ *=poor.*

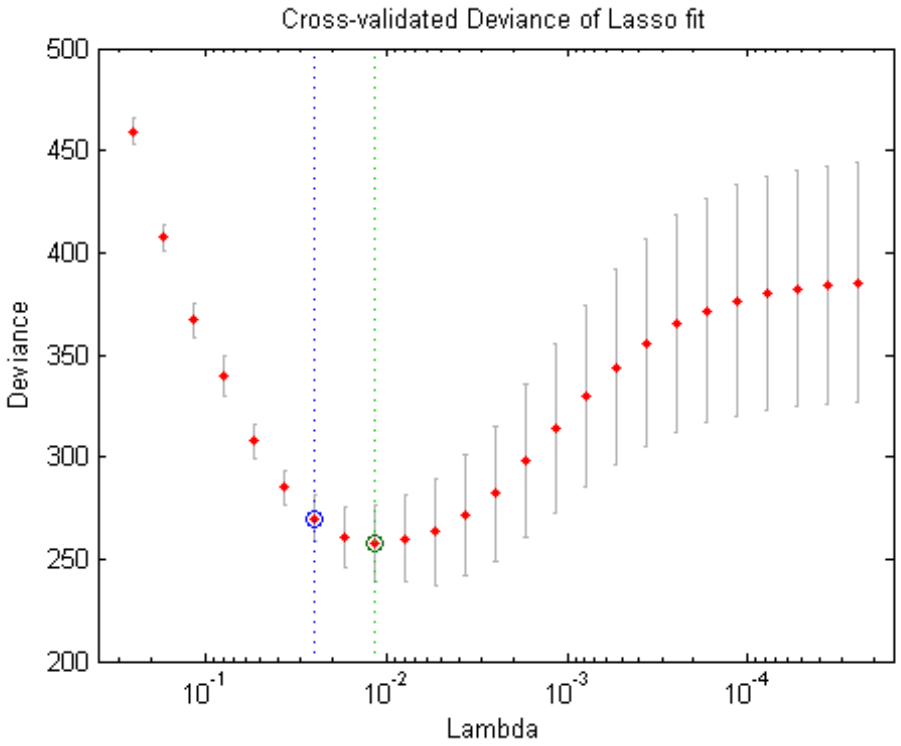
Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

Bias-variance tradeoff and cross-validation



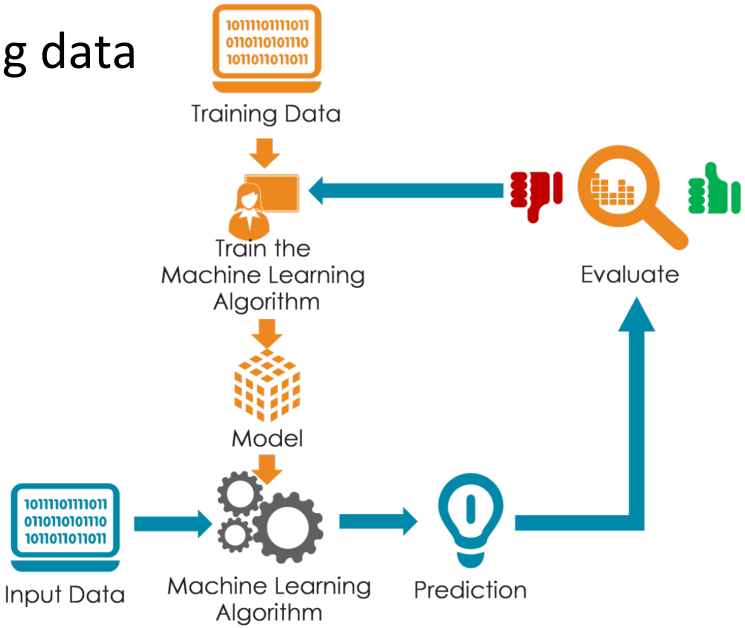


e.g. lasso


Cross-validated Deviance of Lasso fit
Deviance
500
450
400
350
300
250
200
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Lambda

Machine learning

- Artificial intelligence is goal; machine learning is the fuel to get us there.
- Big data



TensorFlow

scikits learn
machine learning in Python

- Computer science + statistics + optimization + numerical linear algebra + domain knowledge …

Used everywhere

Top 10 Use Cases for Data Science & Machine Learning

HEALTHCARE:
Patient Diagnosis

FINANCE:
Fraud Detection

MANUFACTURING:
Anomaly Detection

RETAIL:
Inventory Optimization

GOVERNMENT:
Smarter Services

TRANSPORTATION:
Demand Forecasting

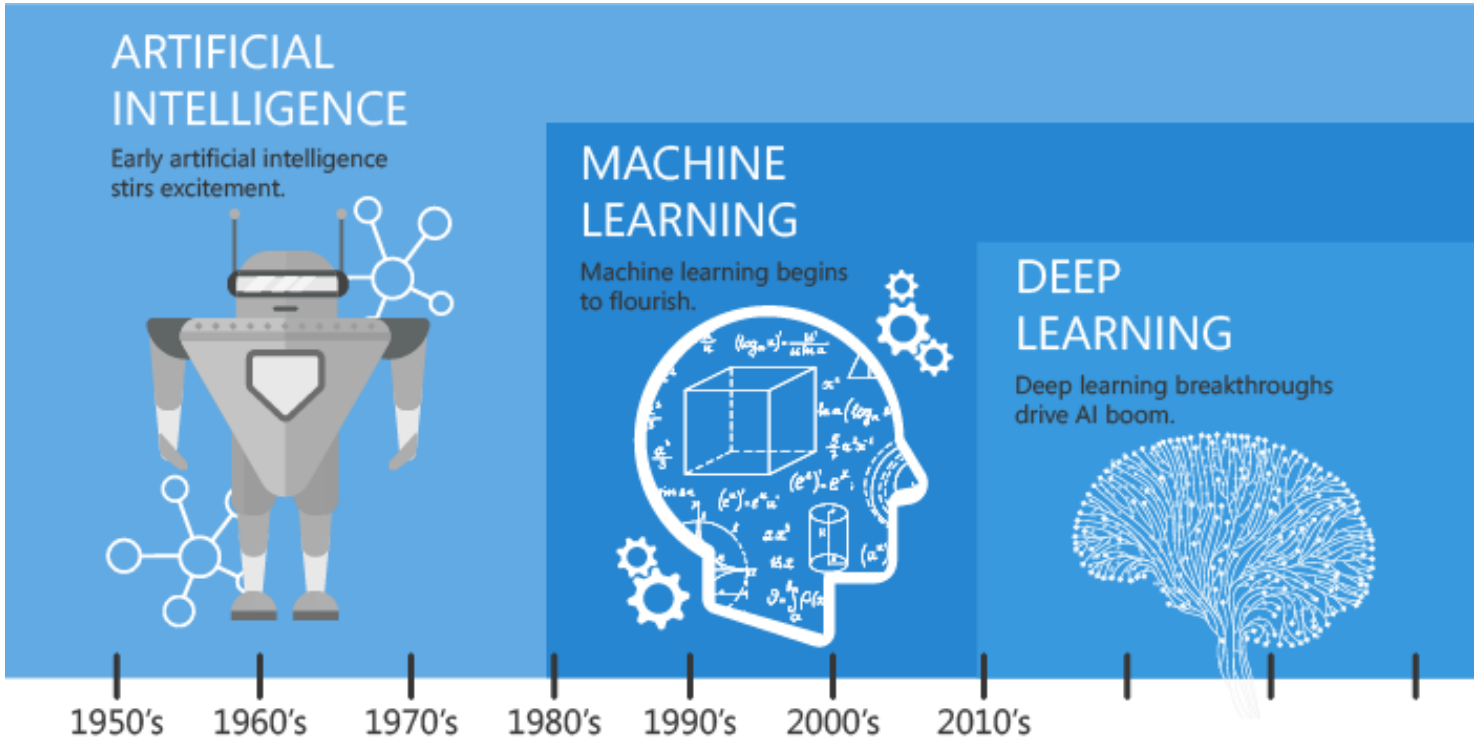
NETWORKS:
Intrusion Detection

E-COMMERCE:
Recommender Systems

MEDIA:
Interaction & Speed

EDUCATION:
Research Insight

Where are we going to?



Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.


SUMMARY