Learning Associations

- Basket analysis:

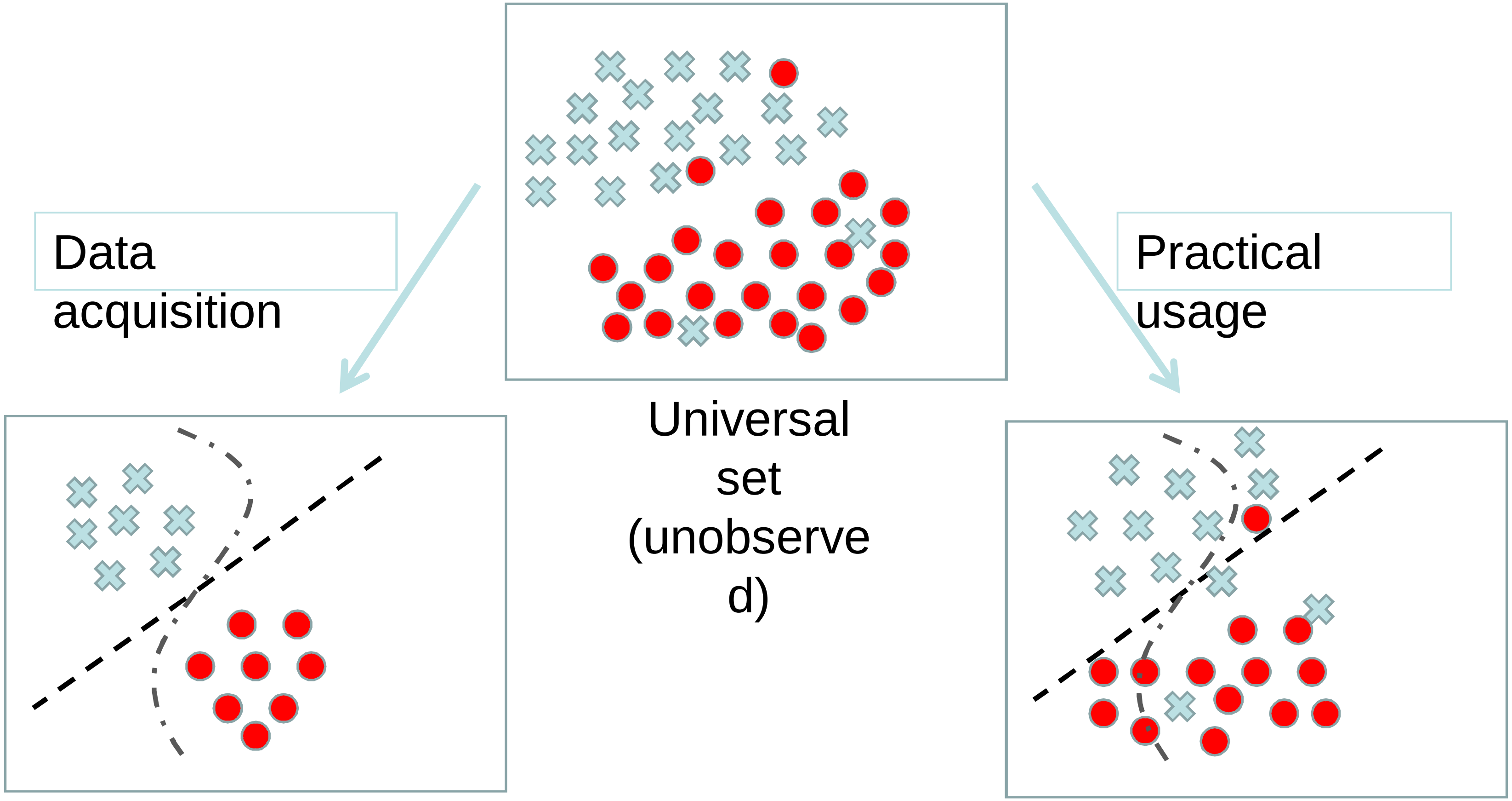
  P (*Y* | *X*) probability that somebody who buys *X* also buys *Y* where *X* and *Y* are products/services.

Market-Basket transactions

Example: *P* (chips | beer) = 0.7

TID	***Items***
1	**Bread, Milk**
2	**Bread, Diaper, Beer, Eggs**
3	**Milk, Diaper, Beer, Coke**
4	**Bread, Milk, Diaper, Beer**
5	**Bread, Milk, Diaper, Coke**

Training and testing



Data acquisition

Universal set (unobserved)

Practical usage

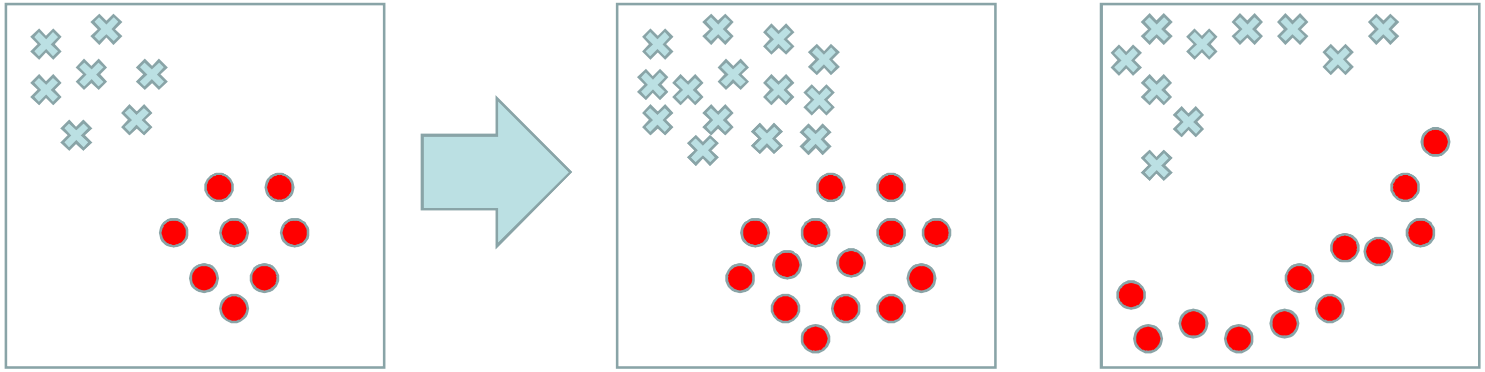
Training set (observed)

Testing set (unobserved)

Training and testing

- Training is the process of making the system able to learn.

- No free lunch rule:
  - Training set and testing set come from the same distribution
  - Need to make some assumptions or bias



Performance

- There are several factors affecting the performance:
  - **Types of training** provided
  - The form and extent of any initial **background knowledge**
  - The **type of feedback** provided
  - The **learning algorithms** used

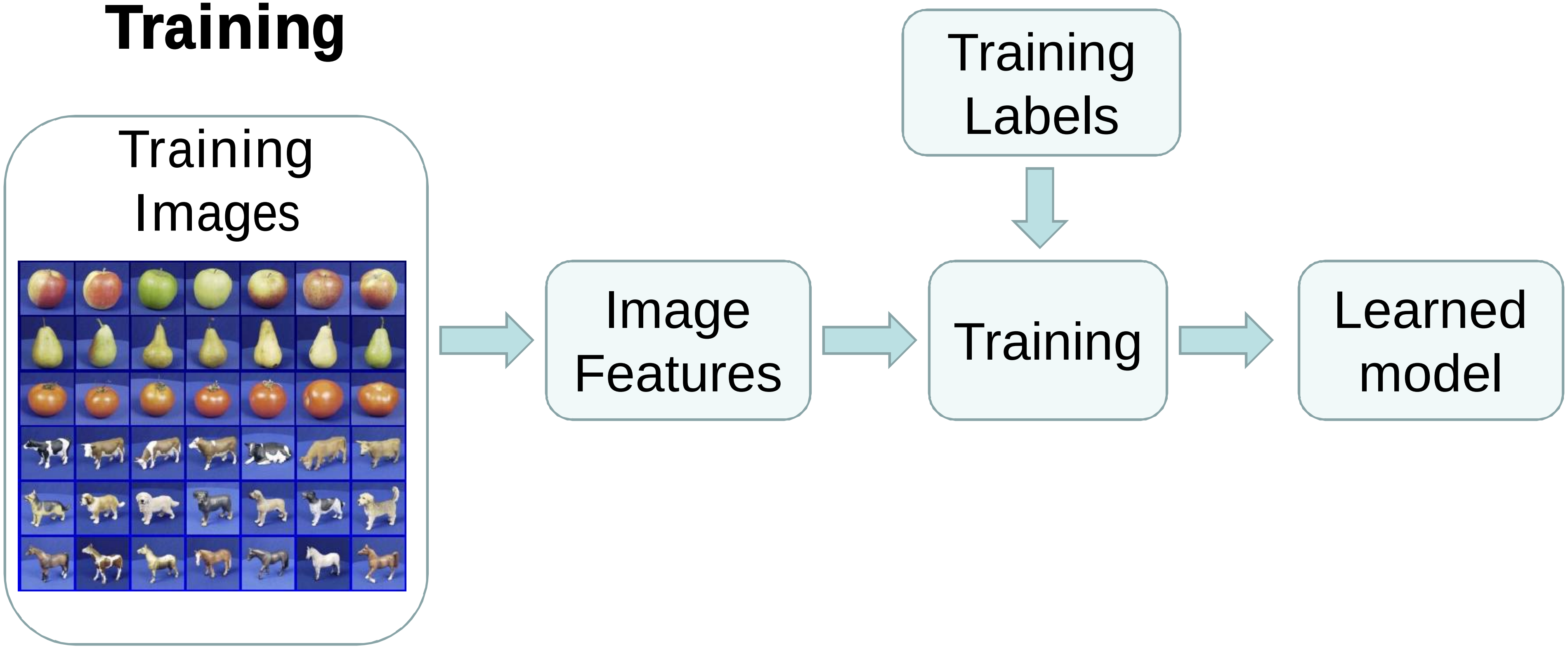
- Two important factors:
  - Modeling
  - Optimization

Classification: Applications

- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- Web Advertising: Predict if a user clicks on an ad on the Internet.

Steps

Training



Training Images

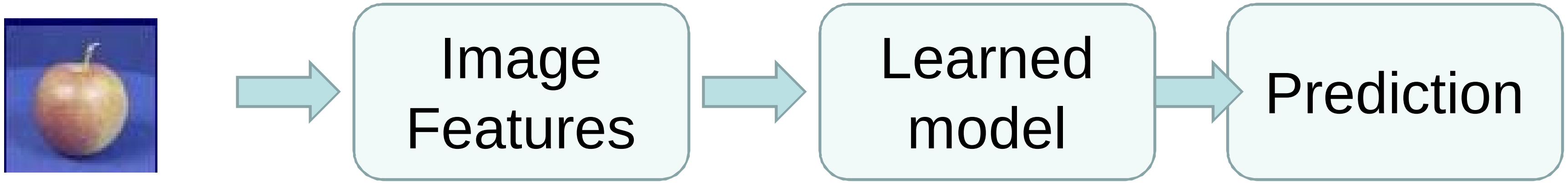
Image Features

Training Labels

Training

Learned model

Testing



Test Image

Image Features

Learned model

Prediction

Prediction: Regression

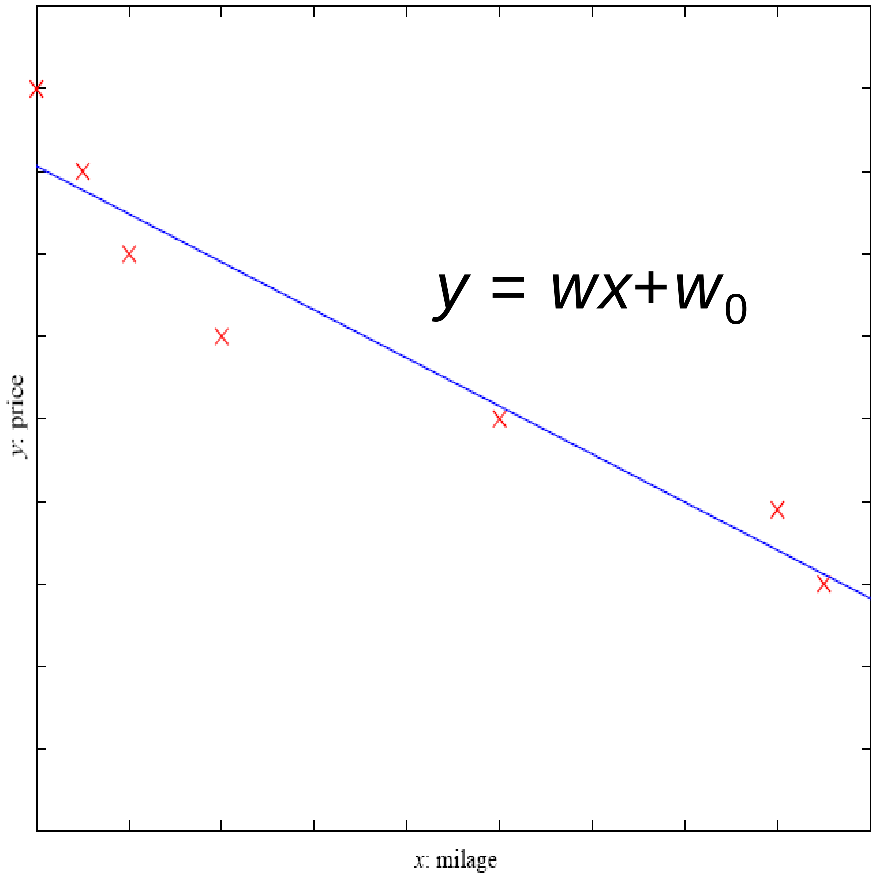
- Example: Price of a used car
- x : car attributes

  y : price

  $$y = g\,(x \mid \theta\,)$$

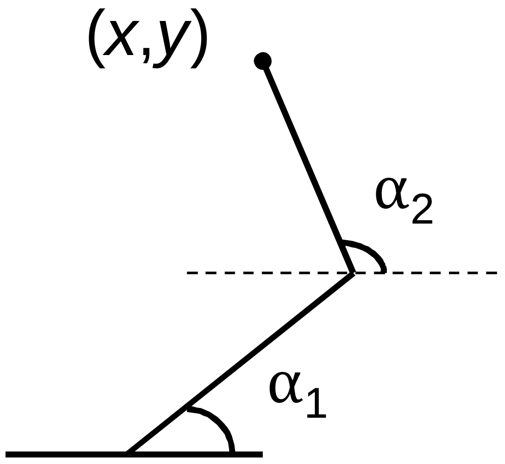
  g ( ) model,

  θ parameters



Regression Applications

- Navigating a car: Angle of the steering wheel (CMU NavLab)
- Kinematics of a robot arm



$\alpha_1 = g_1(x,y)$

$\alpha_2 = g_2(x,y)$

Inductive Learning

- **Given** examples of a function *(X, F(X))*
- **Predict** function *F(X)* for new examples *X*
  - Discrete *F(X)*: Classification
  - Continuous *F(X)*: Regression
  - *F(X)* = Probability(*X*): Probability estimation

Supervised Learning: Uses

Example: decision trees tools that create rules

- Prediction of future cases: Use the rule to predict the output for future inputs
- Knowledge extraction: The rule is easy to understand
- Compression: The rule is simpler than the data it explains
- Outlier detection: Exceptions that are not covered by the rule, e.g., fraud

Algorithms

- The success of machine learning system also depends on the algorithms.

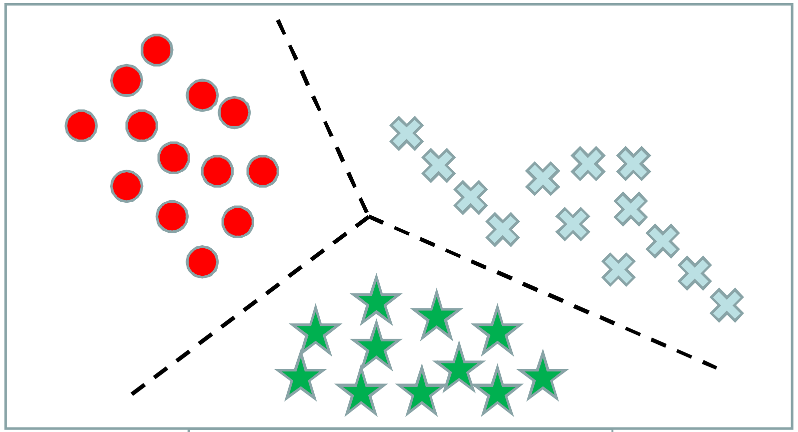
- The algorithms control the search to find and build the knowledge structures.

- The learning algorithms should extract useful information from training examples.

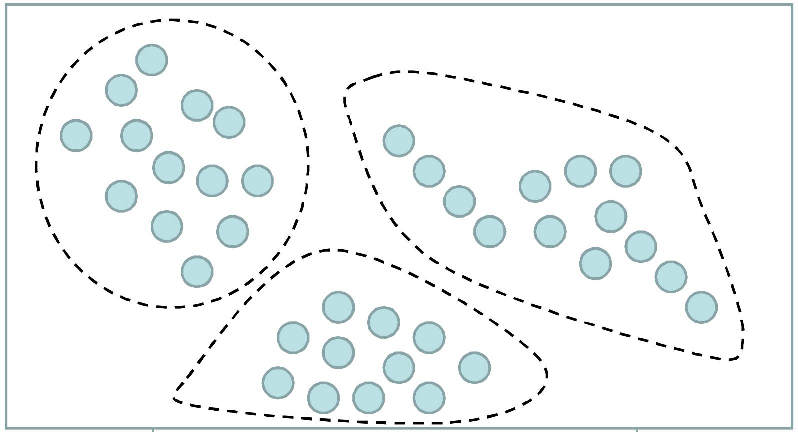
Algorithms

- **Supervised learning** ($\{x_n \in R^d, y_n \in R\}_{n=1}^N$)
  - Prediction
  - Classification (discrete labels), Regression (real values)
- **Unsupervised learning** ($\{x_n \in R^d\}_{n=1}^N$)
  - Clustering
  - Probability distribution estimation
  - Finding association (in features)
  - Dimension reduction
- **Semi-supervised learning**
- **Reinforcement learning**
  - Decision making (robot, chess machine)

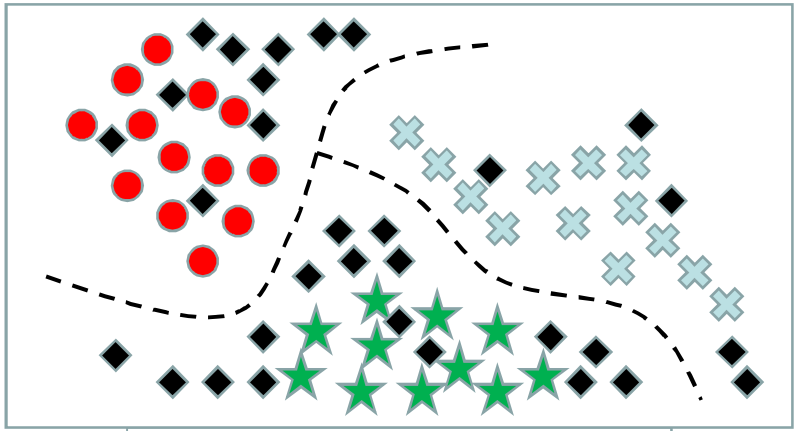
Algorithms



Supervised learning



Unsupervised learning



Semi-supervised learning

What are we seeking?

- Supervised: Low E-out or maximize probabilistic terms

$$error = \frac{1}{N}\sum_{n=1}^{N}[y_n \neq g(x_n)]$$

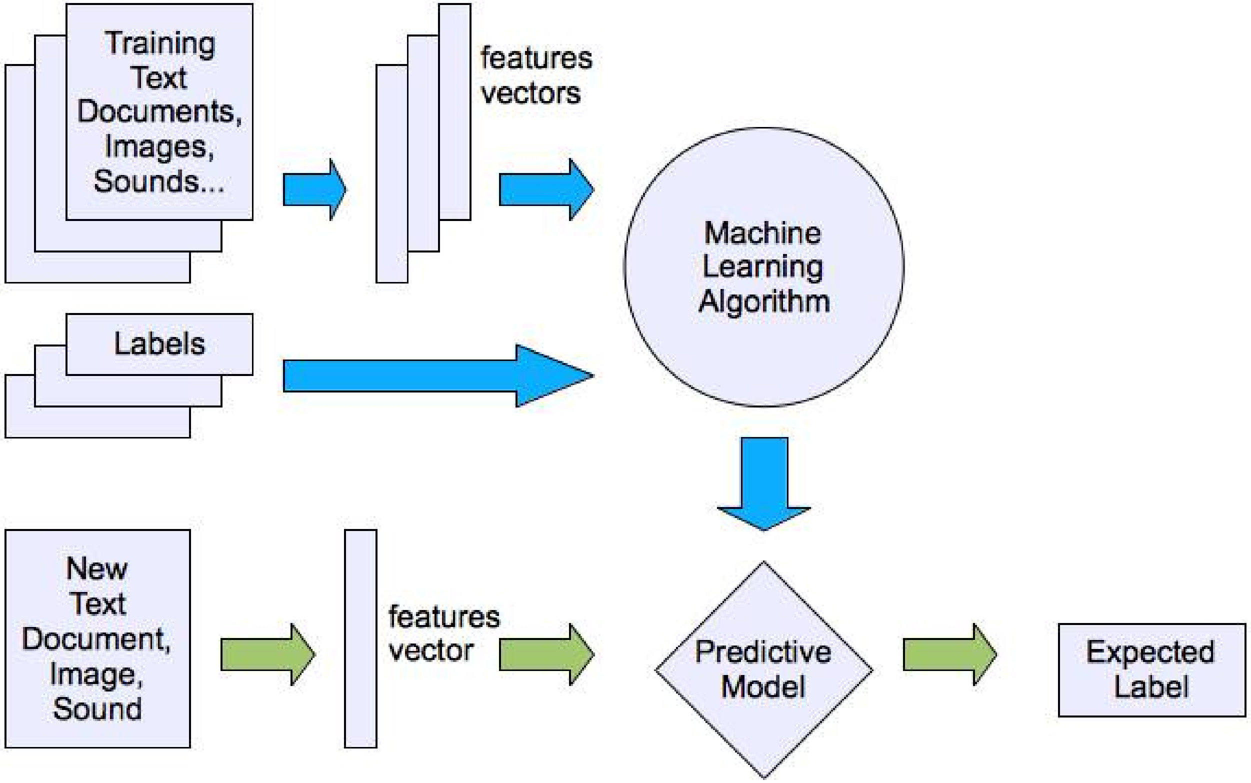
E-in: for training set
E-out: for testing

$$Eout(g) \leq Ein(g) \pm O\left(\sqrt{\frac{d_{VC}}{N} \ln N}\right)$$

- Unsupervised: Minimum quantization error, Minimum distance, MAP, MLE(maximum likelihood estimation)

Machine learning structure

- Supervised learning

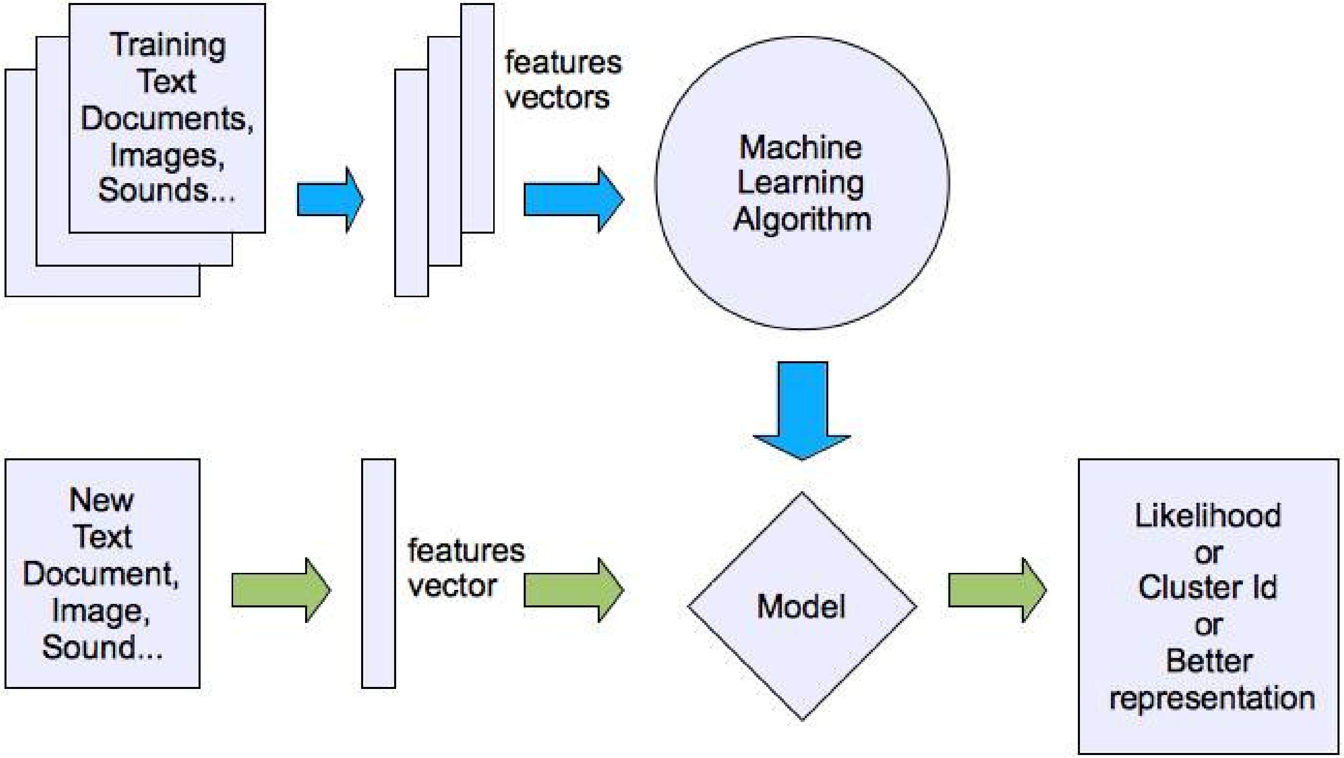

Training Text Documents, Images, Sounds...
features vectors
Machine Learning Algorithm
Labels
New Text Document, Image, Sound
features vector
Predictive Model
Expected Label

Unsupervised Learning

- Learning “what normally happens”
- No output
- Clustering: Grouping similar instances
- Other applications: Summarization, Association Analysis
- Example applications
  - Customer segmentation in CRM
  - Image compression: Color quantization
  - Bioinformatics: Learning motifs

Machine learning structure

- Unsupervised learning

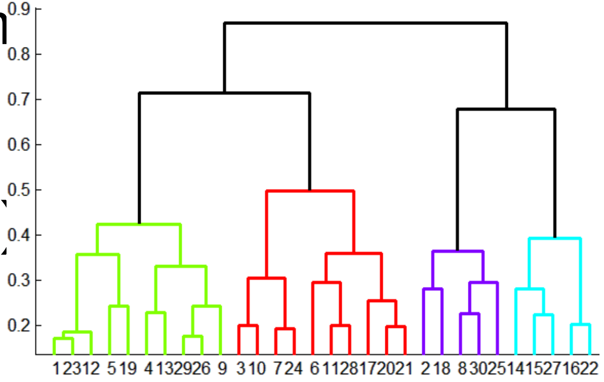


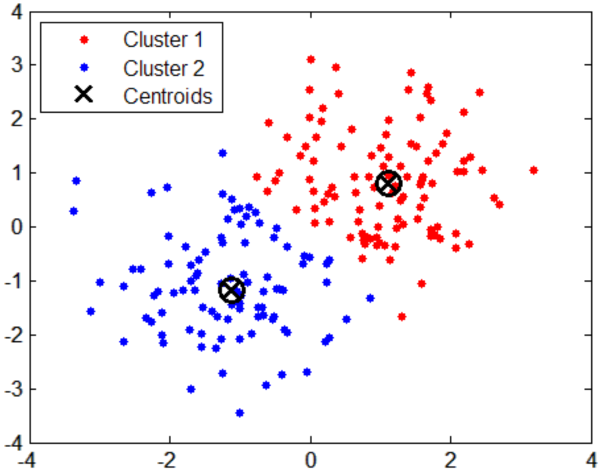
Clustering Analysis

- Definition

  Grouping unlabeled data into clusters, for the purpose of inference of hidden structures or inform

- Dissimilarity measurement
  - Distance : Euclidean(L_2), Manhattan(L_1
  - Angle : Inner product, …
  - Non-metric : Rank, Intensity, …
- Types of Clustering
  - Hierarchical
    - Agglomerative or divisive
  - Partitioning
    - K-means, VQ, MDS, …





(Matlab helppage)

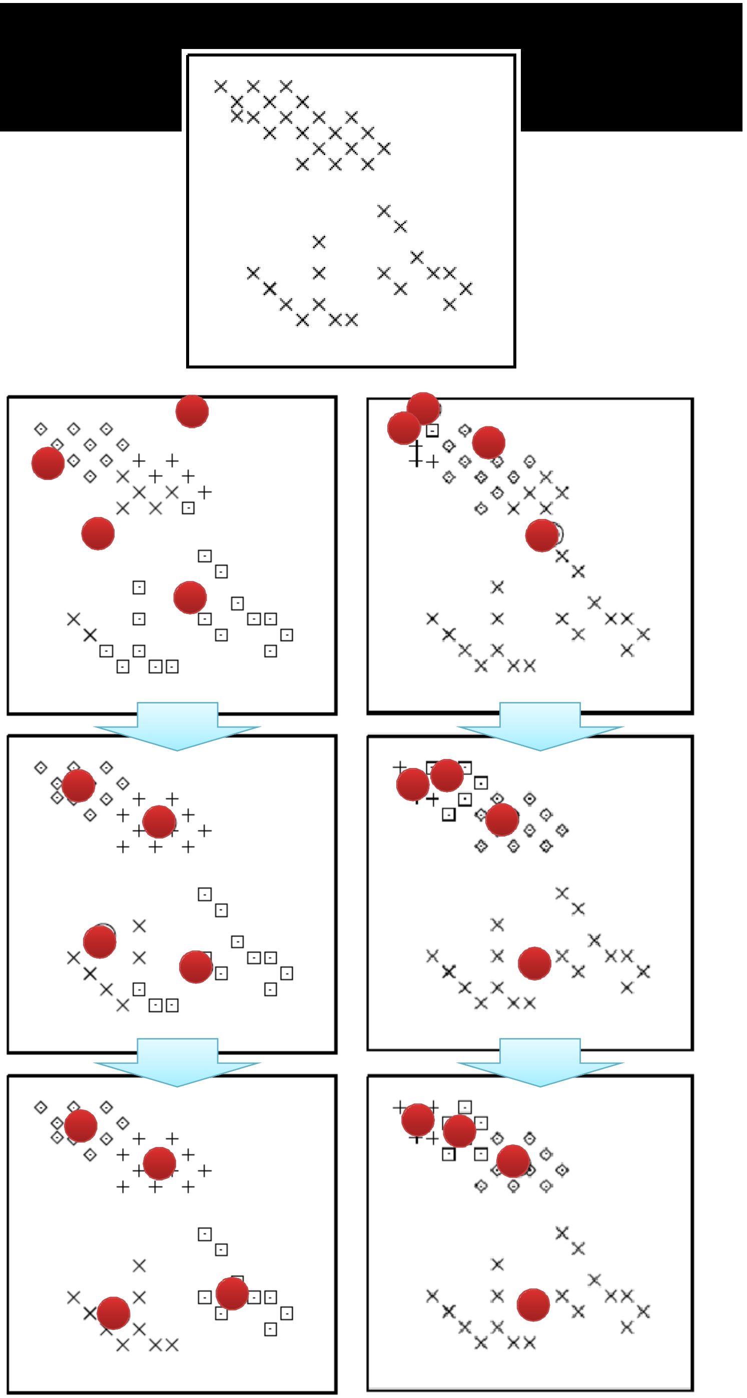
K-Means

- Find K partitions with the total intra-cluster variance minimized

$$E = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{y}_i)^2$$

- Iterative method
  - Initialization : Randomized ***y***$_i$
  - Assignment of ***x*** (***y***$_i$ fixed)
  - Update of ***y***$_i$ (***x*** fixed)

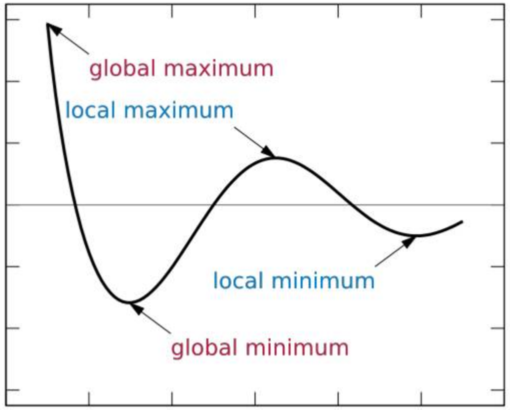
- Problem?
  ➔ Trap in local minima



(MacKay, 2003)

Deterministic Annealing (DA)

- Deterministically avoid local minima
  - No stochastic process (random walk)
  - Tracing the global solution by changing level of randomness



(Maxima and Minima, Wikipedia)

- Statistical Mechanics
  - Gibbs distribution

$$P(E_x) = \exp\left(-E_x/T\right)/Z_x \qquad Z_x = \sum_{x\in\Omega} \exp\left(-E_x/T\right)$$

  - Helmholtz free energy F = D – TS
    - Average Energy D = <∑ E_x>
    - Entropy S = - *P*(E_x) ln *P*(E_x)
    - F = – T ln Z
- In DA, we make F minimized

Deterministic Annealing (DA)

- Analogy to physical annealing process
  - Control energy (randomness) by temperature (high → low)
  - Starting with high temperature (T = 1)
    - Soft (or fuzzy) association probability
    - Smooth cost function with one global minimum
  - Lowering the temperature (T ! 0)
    - Hard association
    - Revealing full complexity, clusters are emerged
- Minimization of F, using E(***x***, ***y***$_j$) = ||***x***-***y***$_j$||^2

$$\frac{\partial}{\partial \mathbf{y}_j} F = 0 \Longleftrightarrow -T \sum_{\mathbf{x}} \frac{d(Z_{\mathbf{x}})}{Z_{\mathbf{x}}} = 0 \Longleftrightarrow \mathbf{y}_j = \frac{\sum_{\mathbf{x}} \mathbf{x} P(\mathbf{y}_j|\mathbf{x})}{\sum_{\mathbf{x}} P(\mathbf{y}_j|\mathbf{x})}$$

Iteratively,

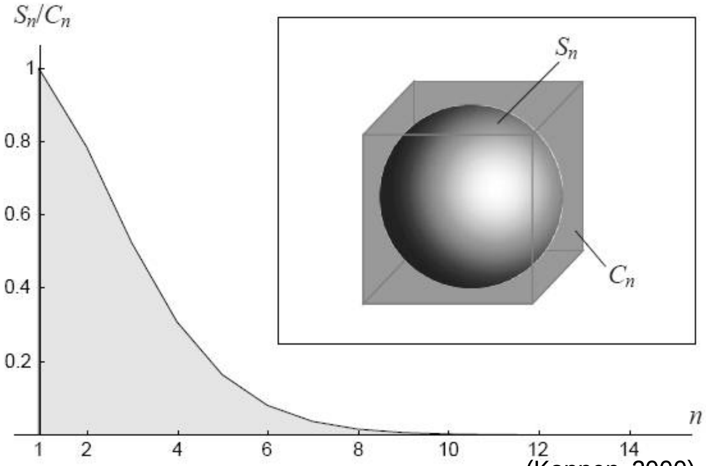
$$\mathbf{y}_j^{(n+1)} = f\left(\mathbf{y}_j^{(n)}\right)$$

Dimension Reduction

- Definition

Process to transform high-dimensional data into low-dimensional ones for improving accuracy, understanding, or removing noises.

- Curse of dimensionality
  - Complexity grows exponentially in volume by adding extra dimensions



(Koppen, 2000)

- Types
  - Feature selection : Choose representatives (e.g., filter,…)
  - Feature extraction : Map to lower dim. (e.g., PCA, MDS, …)

Machine Learning in a Nutshell

- Tens of thousands of machine learning algorithms
- Hundreds new every year
- Every machine learning algorithm has three components:
  - **Representation**
  - **Evaluation**
  - **Optimization**

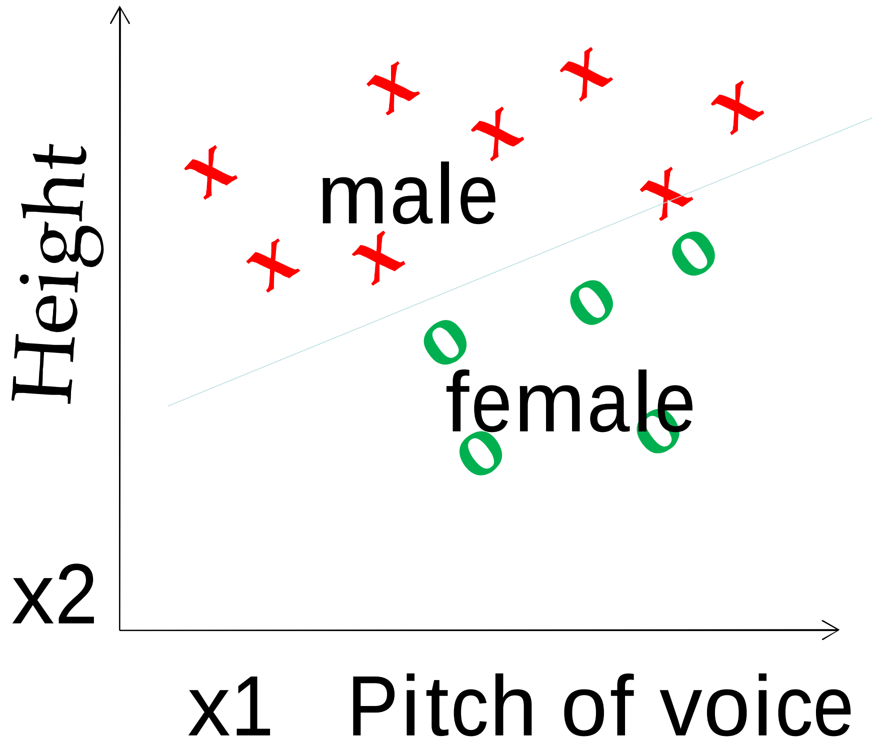
Generative vs. Discriminative Classifiers

Generative Models	Discriminative Models
• Represent both the data and the labels	• Learn to directly predict the labels from the data
• Often, makes use of conditional independence and priors	• Often, assume a simple boundary (e.g., linear)
• Examples	• Examples
– Naïve Bayes classifier	– Logistic regression
– Bayesian network	– SVM
	– Boosted decision trees
• Models of data may apply to future prediction problems	• Often easier to predict a label from the data than to model the data

Slide credit: D. Hoiem

Classifiers: Logistic Regression

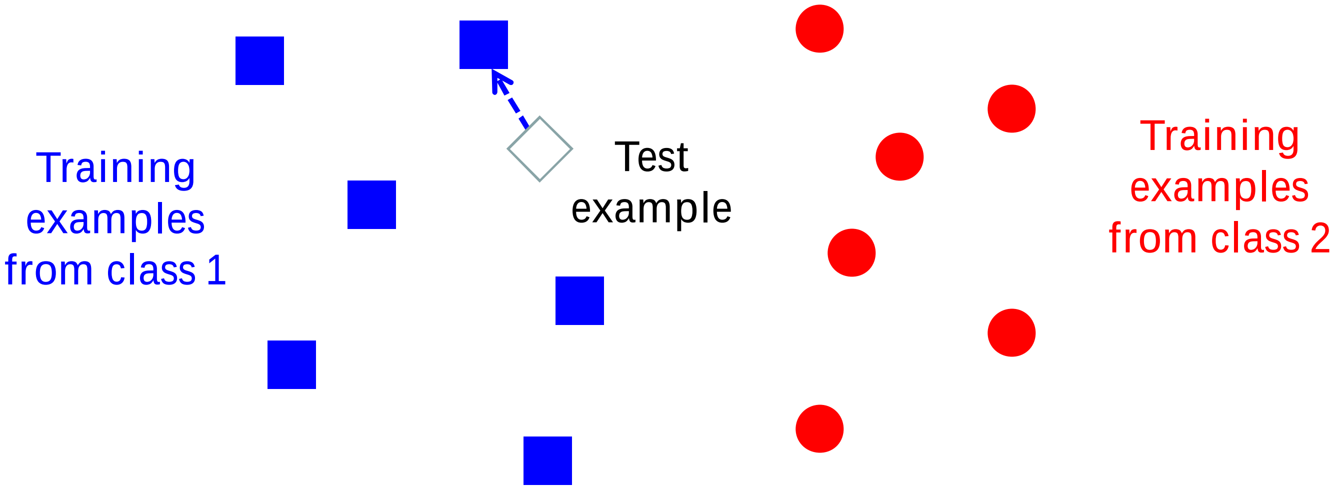
Maximize likelihood of label given data, assuming a log-linear model



$$\log \frac{P(x_1, x_2 \mid y=1)}{P(x_1, x_2 \mid y=-1)} = \mathbf{w}^T \mathbf{x}$$

$$P(y=1 \mid x_1, x_2) = 1/\left(1 + \exp\left(-\mathbf{w}^T \mathbf{x}\right)\right)$$

Classifiers: Nearest neighbor



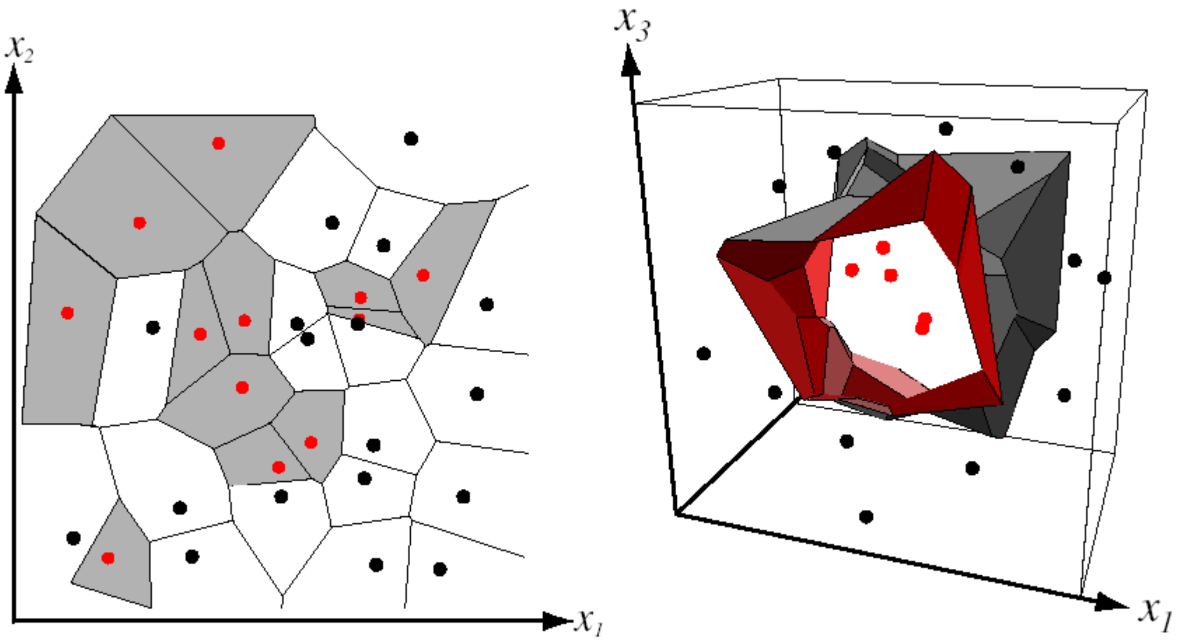
f($\mathbf{x}$) = label of the training example nearest to $\mathbf{x}$

All we need is a distance function for our inputs
No training required!

Slide credit: L. Lazebnik

Nearest Neighbor Classifier

- Assign label of nearest training data point to each test data point



partitioning of feature space for two-category 2D and 3D data

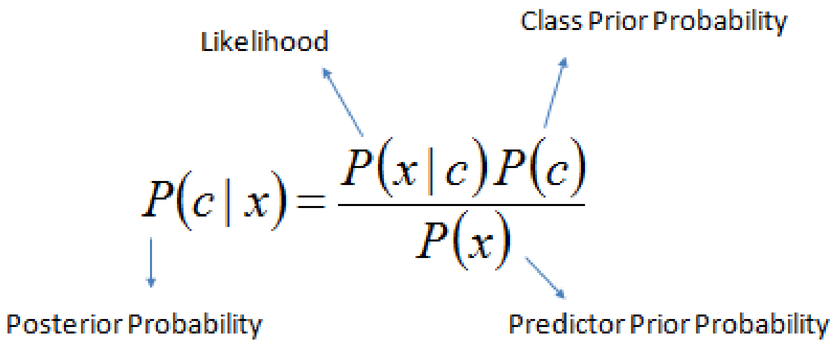
Source: D. Lowe

K-nearest neighbor

- It can be used for both classification and regression problems.
- However, it is more widely used in classification problems in the industry.
- K nearest neighbours is a simple algorithm
  - stores all available cases and
  - classifies new cases by a majority vote of its k neighbours.
  - The case being assigned to the class is most common amongst its ***K nearest neighbours*** measured by a distance function.
  - These distance functions can be ***Euclidean, Manhattan, Minkowski and Hamming distance.***
    - First three functions are used for continuous function and
    - Fourth one (Hamming) for categorical variables.
  - If ***K* = *1*,** then the case is simply assigned to the class of its nearest neighbour.
  - At times, choosing ***K*** turns out to be a challenge while performing ***KNN modelling.***

Naïve Bayes

- Bayes theorem provides a way of calculate
  - posterior probability P(c|x) from P(c), P(x) and P(x|c).
- Look at the equation belo



$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

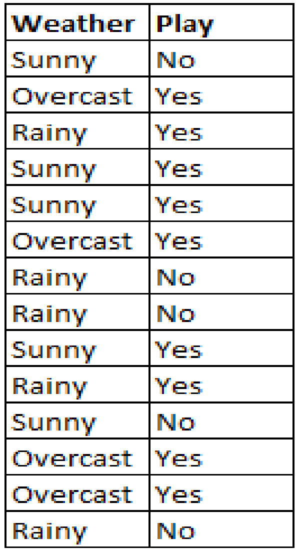
$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

  - Here,
  - *P*(*c*|*x*) is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
  - *P*(*c*) is the prior probability of *class*.
  - *P*(*x*|*c*) is the likelihood which is the probability of *predictor* given *class*.
  - *P*(*x*) is the prior probability of *predictor*

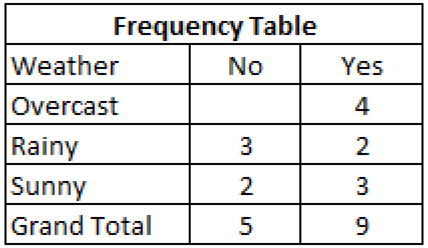
Naïve Bayes Example

- Let’s understand it using an example.
  - Have a training data set of weather and corresponding target variable ‘Play’.
  - Now, we need to classify whether players will play or not based on weather condition.
  - Let’s follow the below steps to perform it.
    - Step 1: Convert the data set to frequency table
    - Step 2: Create Likelihood table by finding the probabilities like
      - Overcast probability = 0.29 and

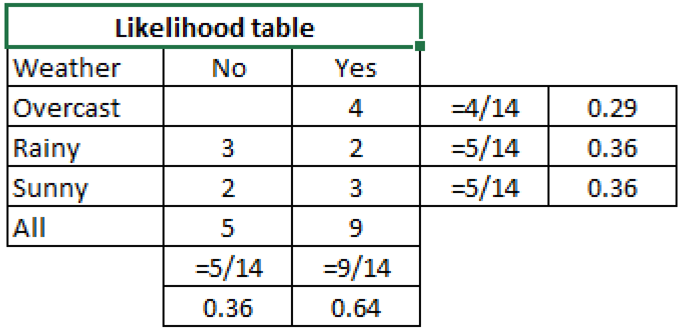
bability of playing is 0.64.



Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No



Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9



Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Naïve Bayes

- Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class.
  - The class with the highest posterior probability is the outcome of prediction.

- **Problem:**
  - Players will pay if weather is sunny, is this statement is correct?
  - We can solve it using above discussed method,
    - so P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)
    - Here we have, P (Sunny |Yes) = 3/9 = 0.33,
      P(Sunny) = 5/14 = 0.36,
      P(Yes)= 9/14 = 0.64
    - Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60,
    - which has higher probability.
- Naive Bayes uses a similar method to
  - predict the probability of different class based on various attributes.
  - This algorithm is mostly used in text classification and
  - with problems having multiple classes.

EM algorithm

- Problems in ML estimation
  - Observation X is often not complete
  - Latent (hidden) variable Z exists
  - Hard to explore whole parameter space

- Expectation-Maximization algorithm
  - Object : To find ML, over latent distribution $P(Z|X,\theta)$
  - Steps

    0. Init – Choose a random θ^{old}

    1. E-step – Expectation $P(Z|X, \theta^{old})$

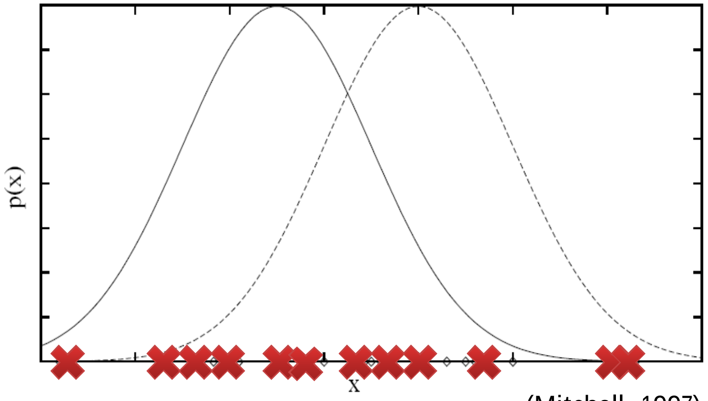
    2. M-step – Find θ^{new} which maximize likelihood.

    3. Go to step 1 after updating θ^{old} Ã θ^{new}

Maximum Likelihood (ML) Estimation

- Problem

  Estimate hidden parameters (θ={μ, σ}) from the given data extracted from k Gaussian distributions



(Mitchell , 1997)

- Gaussian distribution

$$\mathcal{N}(x|\mu,\sigma) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- Maximum Likelihood

$$\theta_{\mathrm{ML}} = \underset{\theta\in\Theta}{\operatorname{argmax}} P(\mathcal{X}|\theta) = \underset{\theta\in\Theta}{\operatorname{argmax}} \prod_{i=1}^{m} P(x_i|\theta) = \underset{\theta\in\Theta}{\operatorname{argmax}} \prod_{i=1}^{m} \ln\{P(x_i|\theta)\}$$

  - With Gaussian (P = N), $\theta_{\mathrm{ML}} = \underset{\{\mu\}\in\Theta}{\operatorname{argmin}} \sum_{i=1}^{m} (x_i-\mu)^2$
  - Solve either brute-force or numeric method