Session #25

CMSC 409: Artificial Intelligence

http://www.people.vcu.edu/~mmanic/

Virginia Commonwealth University,
Fall 2015,
Dr. Milos Manic
(misko@vcu. edu)

Topics for today

- Announcements
- Previous session review
- Probabilistic classifier
    - *Univariate and multivariate classification, examples*
- Learning from Neighbors
    - Eager vs. lazy learners
    - Lazy learners
        - *K-nearest-neighbor classifier*
        - *Case based reasoning (CBR)*
        - *Distance measures*
            - *Euclidian space, coding theory, fuzzy space*
            - *Euclidian, Manhattan, Chebyshev, angle distance*

Announcements

Session # 25

- Blackboard
  - Slides, class paper instructions and template uploaded

- Assignments, update
  - Final exam: 12/04 through 12/06 (48 hour take home)
  - Pr. 4 posted (Deadline Nov. 30)

- Paper
  - *The fourth draft - due* Nov. 27, 2015
  - *In addition to previous draft, it should contain a technique (or selection thereof), you plan on using to solve the selected problem (check out the class paper instructions for the 3rd draft)*
- Subject line and signature
  - *Please use specified in syllabus*

Probabilistic classifier

- Univariate and multivariate classification
- General Bayes classifier, examples

Probabilistic Classifiers

- Probabilistic classifiers
  - Model the probabilistic distribution $P(C_i \mid x)$
  - the probability of belonging to class *Ci* given prior knowledge of *x*
  - make predictions based on probabilistic inference on this model.

- Bayesian decision theory: general framework for modeling $P(C_i \mid x)$ distribution using a Bayesian network

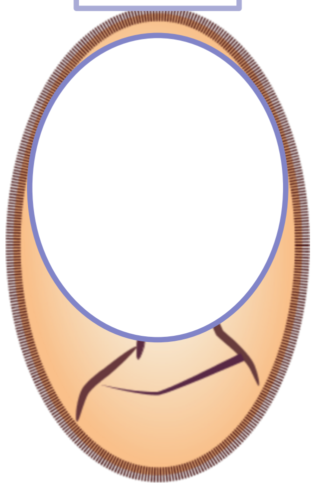
Probabilistic Classifiers

- Different probabilistic classifiers
  - Naïve Bayes:
    - application of Bayes theorem with (naïve) assumption of independence of features
    - we assume features independent from each other, samples are independent and identically distributed
  - Hidden Markov models:
    - instances in a sample not independent and the data is composed by sequences generated by a parametric random process.
    - States directly visible to the observer, hidden refers to the state sequence though which the model passes (not model parameters)
    - Viewed as simplest form of *dynamic* Bayesian network.
  - Dynamic Bayesian networks:
    - Generalization of hidden Markov models and Kalman filters
  - Kalman filters:
    - Linear quadratic estimation, applied in time series analyis

Simple example

Drew could be a name for a Male or a Female

Drew:



Is Drew a Female or a Male?

Training Dataset

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$P(male \mid drew) = \frac{P(drew \mid male)P(male)}{p(drew)}$$

$$posterior = \frac{likelihood \times prior}{marginal\ likelihood}$$

From: http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf

Simple example

$$P(male \mid drew) = \frac{P(drew \mid male)P(male)}{p(drew)}$$

Drew is more likely to be a Female

Training Dataset

Name	Sex
Drew	Male
Claudia	Female
Drew	Female
Drew	Female
Alberto	Male
Karin	Female
Nina	Female
Sergio	Male

$$P(Male \mid drew) = \frac{(1/3)(3/8)}{3/8} = \frac{0.125}{3/8}$$

$$P(Female \mid drew) = \frac{(2/5)(5/8)}{3/8} = \frac{0.250}{3/8}$$

The denominator can be ignored

$$P(male \mid drew) = \frac{P(drew \mid male)P(male)}{p(drew)}$$

$$posterior = \frac{likelihood \times prior}{marginal\ likelihood}$$

Multivariate Classification

- Probability that previously unseen sample *x* belongs to class *Ci* :

$$P(C_i \mid x) = \frac{P(x \mid C_i)P(C_i)}{P(x)} = \frac{P(x \mid C_i)P(C_i)}{\sum_{k=1}^{K} P(x \mid C_k)P(C_k)}$$

where $\sum_{k=1}^{K} P(x \mid C_k)P(C_k)$ is used for normalization

• Multiplying of lots of probabilities can result in floating point underflow. Since:

$$\log(xy) = \log x + \log y$$

so, for predicting $P(C_i \mid x)$ or belonging to class *Ci* , we can focus on the numerator (log of it) :

$$g_i(x) = \log P(x \mid C_i) + \log P(C_i)$$

Note: the values of $g_i(x)$ are **negative**

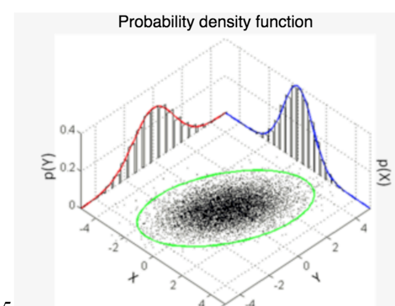
Multivariate Classification

- The maximum value of $g_i(x)$ indicates (predicts) the class C_i of sample
- One approach to get $P(x \mid C_i)$ is assume that it is drawn from a Gaussian distribution, i.e:

$$P(x \mid C_i) \sim N_d(\mu_i, \Sigma_i)$$

$$P(x \mid C_i) = \frac{1}{(2\pi)^{d/2} \left|\Sigma_i\right|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right]$$

- Where:
  - N_d *is the Multivariate Normal Distribution (MVN)*
  - d is the dimension of the data
  - Σ_i is the covariance matrix of the Gaussian distribution
  - μ_i is the mean of the Gaussian distribution



https://en.wikipedia.org/wiki/Multivariate_normal_distribution

Multivariate Classification

- Having a training dataset $X^{(i)} \in R^{N_i \times d}$ we can estimate the values of Σ_i , μ_i and $P(C_i)$ using the following equations:

$$P(C_i) = \frac{N_i}{N}$$

N_i is the number of samples that belong to class i

N is the number of all samples

$$\mu_i = \frac{\sum_{k=1}^{Ni} x_k^{(i)}}{N_i}, \textit{where } x_k^{(i)} \textit{ is the sample } k \textit{ that belongs to the class } Ci$$

$$\boxed{\Sigma_i = \frac{\sum_k (x_k^{(i)} - \mu_i)(x_k^{(i)} - \mu_i)^T}{N_i}}$$

For multi dimensional data

$$\boxed{\Sigma_i = \sigma_i^2 = \frac{\sum_k (x_k^{(i)} - \mu_i)^2}{N_i}}$$

For 1D data

Multivariate Classification

- In summary, first calculate Σ_i, μ_i, and $P(C_i)$ using the training dataset;

- then, to predict the class of a new sample x, evaluate $P(x \mid C_i)$ for all classes C_i, and evaluate the following equation:

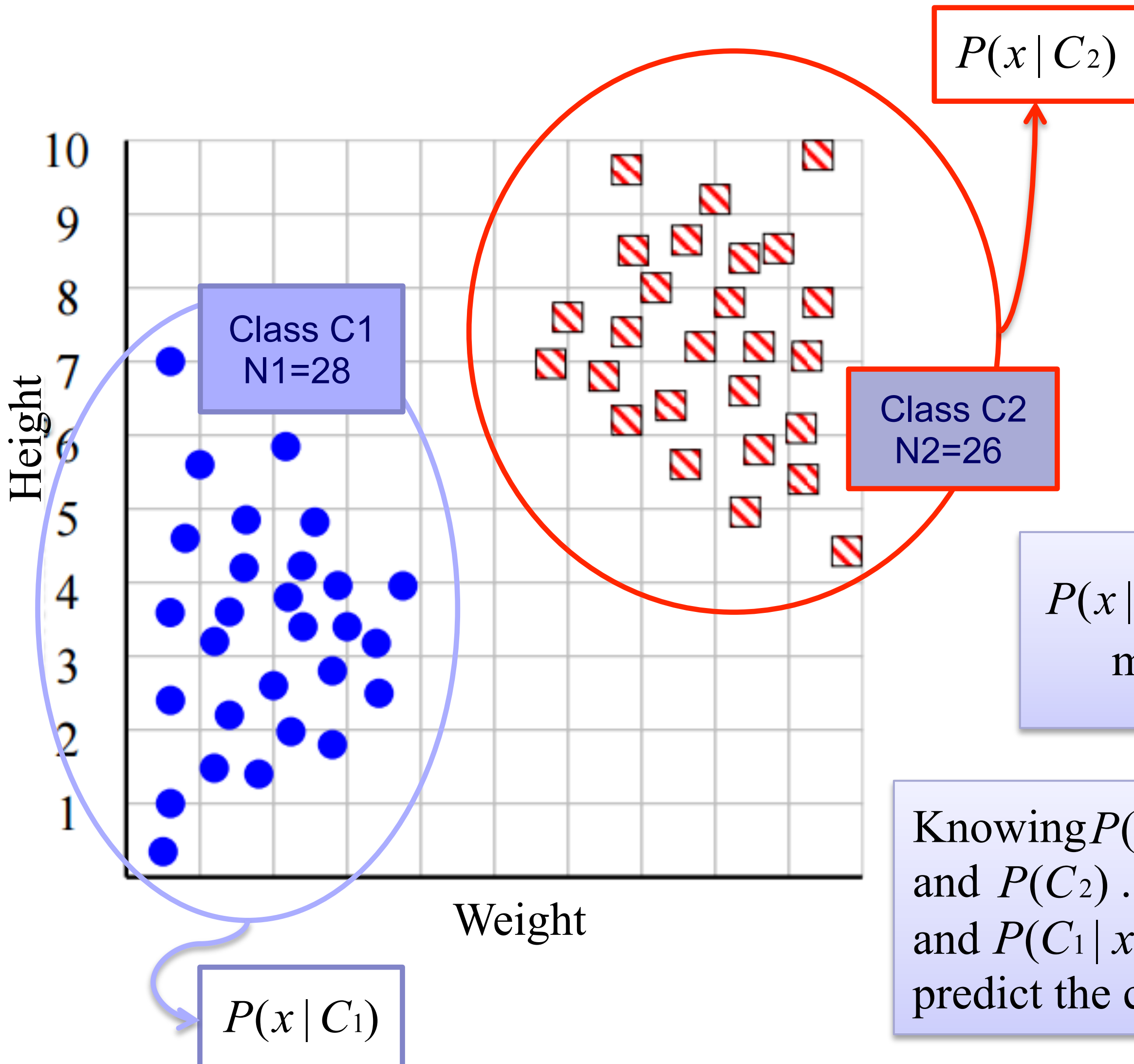
$$g_i(x) = \log P(x \mid C_i) + \log P(C_i)$$

- then we predict that the sample x belongs to the class i (class C_i corresponding to the maximum $g_i(x)$)

- which in turn is equivalent to the maximum of

$$P(x \mid C_i) P(C_i)$$

Multivariate Classification



N= N1+N2= 54

P(C1)=N1/N=0.52

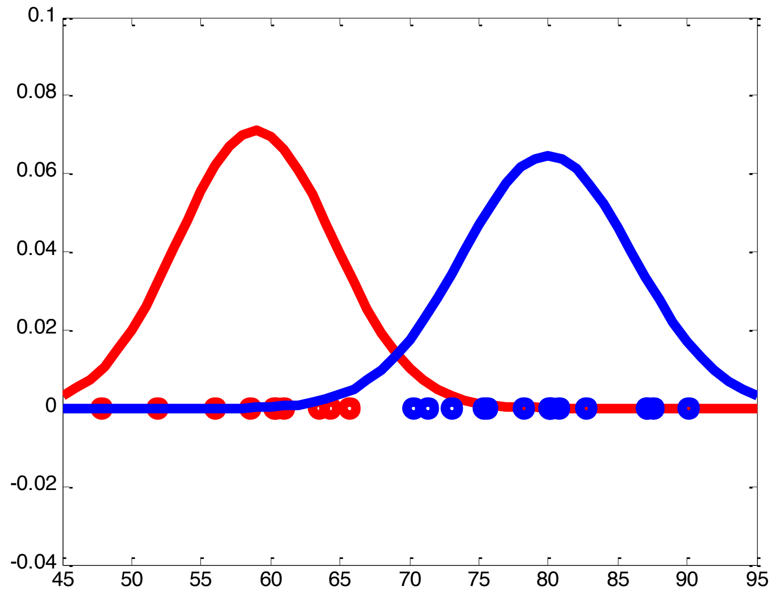
P(C2)=N2/N=0.48

$P(x \mid C_1)$ and $P(x \mid C_2)$ can be modeled as Gaussians.

Knowing $P(x \mid C_1)$, $P(x \mid C_2)$, $P(C_1)$, and $P(C_2)$. We can predict $P(C_1 \mid x)$ and $P(C_1 \mid x)$ using the Bayes rule to predict the class of new samples.

Example

Female(C1)	Male(C2)
56.0307	73.0330
64.2989	87.1268
60.3343	75.5307
51.8031	80.1888
47.8763	78.1822
58.5809	80.7478
65.7290	70.2774
60.9058	87.6195
60.2713	82.7291
63.4388	90.0496
	87.0834
	80.0574
	75.3048
	71.3055
	80.0849



When dealing with continuous data, we typically assume values from each class follow Gaussian distribution.

For. ex., training data contains continuous attribute x. We can segment data by class, then compute mean & variance for each class. Probability distribution of some value given a class C1 $P(x = v \mid C_1)$ *:*

For 1D data, the multivariate Gaussian reduces to:

$$P(x = v \mid C_1) = \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left[-\frac{(v - \mu_1)^2}{2(\sigma_1)^2} \right]$$

Example

Female(C1)	Male(C2)
56.0307	73.0330
64.2989	87.1268
60.3343	75.5307
51.8031	80.1888
47.8763	78.1822
58.5809	80.7478
65.7290	70.2774
60.9058	87.6195
60.2713	82.7291
63.4388	90.0496
	87.0834
	80.0574
	75.3048
	71.3055
	80.0849

$$P(C_i \mid x) = \frac{P(x \mid C_i)P(C_i)}{P(x)}$$

Mean Calculated from the data set

For Female	For Male
$\mu_1 = 58.92$	$\mu_2 = 79.95$

Standard deviation calculated from the dataset

For Female	For Male
$\sigma_1 = 5.334$	$\sigma_2 = 5.955$

$$\sigma = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)^2}$$

Modeled after Gaussian *pdf*
(probability density distribution)

$$P(x \mid C_1) = \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left[-\frac{(x-\mu_1)^2}{2(\sigma_1)^2}\right]$$

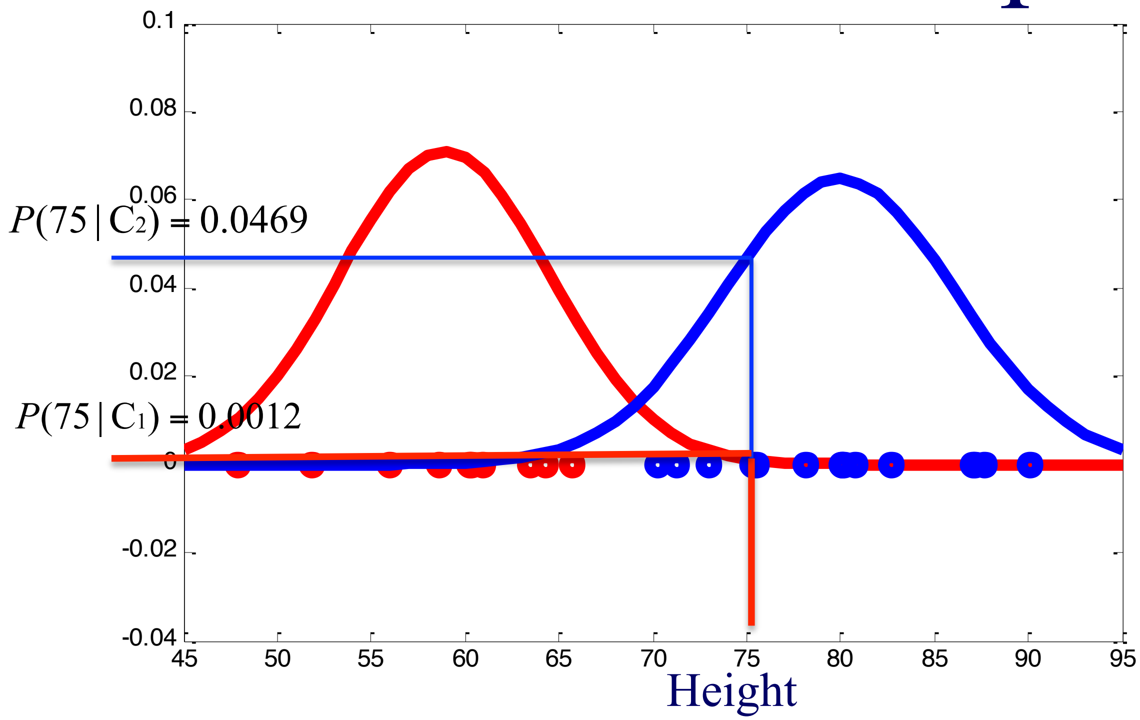
$$P(x \mid C_2) = \frac{1}{\sigma_2 \sqrt{2\pi}} \exp\left[-\frac{(x-\mu_2)^2}{2(\sigma_2)^2}\right]$$

$$P(C_1) = \frac{10}{25}$$

$$P(C_2) = \frac{15}{25}$$

To predict the class of a new sample *x*, we just evaluate *P(x|C1)P(C1)* and *P(x|C2)P(C2)*; the one which is larger corresponds to the predicted class

Example



$$P(x | C_1) = \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left[-\frac{(x-\mu_1)^2}{2(\sigma_1)^2}\right]$$

$$P(x | C_2) = \frac{1}{\sigma_2 \sqrt{2\pi}} \exp\left[-\frac{(x-\mu_2)^2}{2(\sigma_2)^2}\right]$$

$$P(C_1) = \frac{10}{25} \qquad P(C_2) = \frac{15}{25}$$

Input: 75

$$P(75 | C_1) = 0.0012$$

$$P(75 | C_2) = 0.0469$$

$$P(75 | C_1) P(C_1) = 4.8000\text{e-}04$$

$$P(75 | C_2) P(C_2) = 0.0281$$

We predict that the subject with height 75 belongs to class C2, because

$$P(75 | C_1) P(C_1) < P(75 | C_2) P(C_2)$$

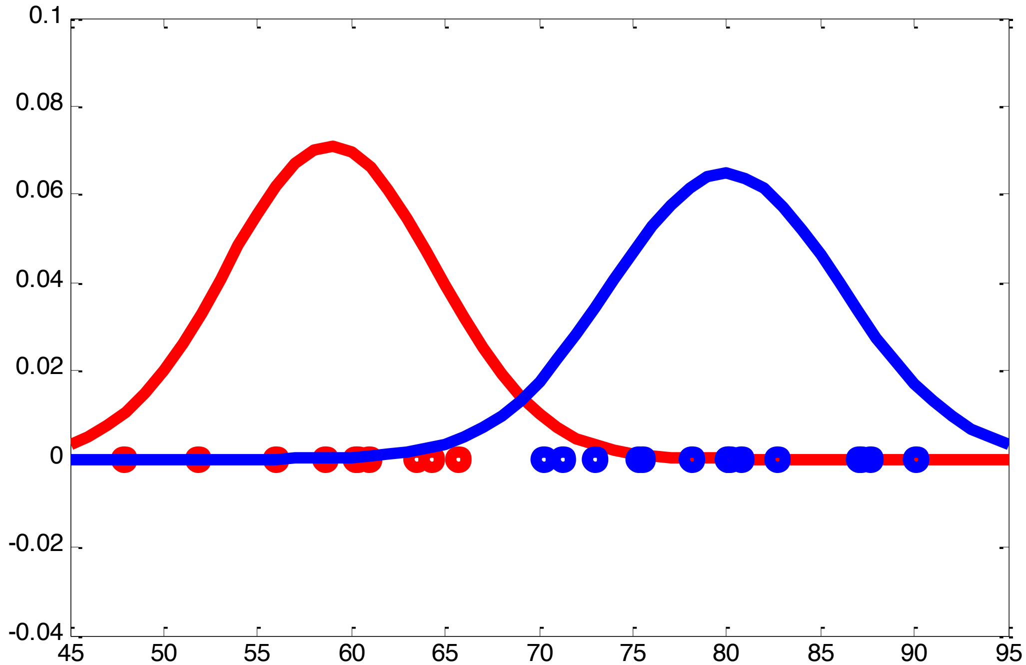
$$g_1(75) = \log(P(C_1 | x)) + \log(P(C_1))$$

$$g_1(75) = \log 0.0012 + \log \frac{10}{25} = -7.6417$$

$$g_2(75) = \log(P(C_2 | x)) + \log(P(C_2))$$

$$g_2(75) = \log 0.0469 + \log \frac{15}{25} = -3.5706$$

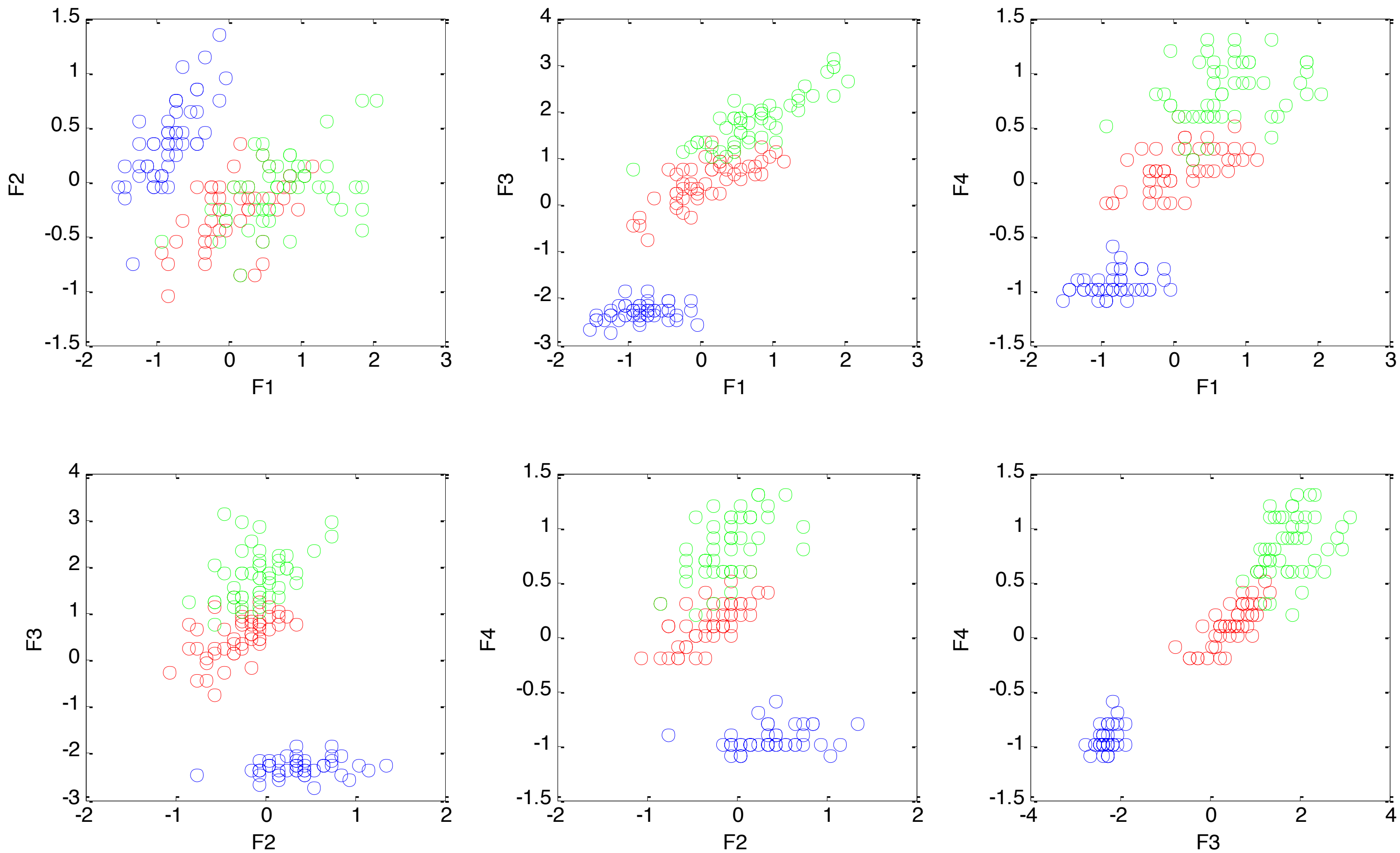
Example



$$P(C_1 \mid 75) = \frac{0.0012\,\frac{10}{25}}{\left(0.0012\,\frac{10}{25}\right)\left(0.0469\,\frac{15}{25}\right)} = 0.0168$$

$$P(C_2 \mid 75) = \frac{0.0469\,\frac{15}{25}}{\left(0.0012\,\frac{10}{25}\right)\left(0.0469\,\frac{15}{25}\right)} = 0.9832$$

Example #1: Iris data set



https://archive.ics.uci.edu/ml/datasets/Iris

Learning from Neighbors

- Eager vs. lazy learners
- Lazy learners
  - *K-nearest-neighbor classifier*
  - *Case based reasoning (CBR)*
  - *Distance measures*
    - *Euclidian space, coding theory, fuzzy space*
    - *Euclidian, Manhattan, Chebyshev, angle distance*

Lazy Learners – Learning from Neighbors

Eager vs. lazy learners

- Eager learners
  - *Decision trees, ANNs, SVMs, association rules*
  - *The model is defined before unseen patterns arrive (eager to classify new patterns)*
  - *Essential part of the work done in training phase*

- Lazy learners
  - *Store training pattern and waits until testing pattern arrives to cluster/ predict…*
  - *Storage/computation expensive, good fit for parallel execution*
  - *Incremental learning, learning by analogy*
  - *Essential part of the work done essentially in testing phase*

Lazy Learners – k-Nearest-Neighbor Classifier

k-Nearest-Neighbor Classifier

Algorithm

- Searches for *k* training patterns most similar to testing pattern
- For *k*=1: unknown pattern assigned to the closest single pattern's class
- For *k*=n: classifies unknown pattern as belonging to
  - a major class of neighbors
  - average of *k* similar patterns
- Both classification and prediction

Similarity

- *Similarity based on certain similarity measure or distance metrics*
- *Various distance measures*
  - *Hamming, Euclidean, Manhatan*

Normalization

- *If pattern attributes of significantly different ranges – normalize*

Lazy Learners – k-Nearest-Neighbor Classifier

k-Nearest-Neighbor Classifier

Distance for categorical attributes

- Such as color (e.g. distance between blue and green, black and white)
- Hamming distance (1 or 0), or grade (black & white maps to [0,1] range)

Missing attributes

- If both comparable attribute from two patterns are missing , *dist.*=1
- If one missing, then *dist.*=|*attrib*-1|

Determining *k*

- *Heuristics*
- *k=1, 2, 3,…until satisfies error criterion (min error)*
- *Cases:*

$$N_{patterns} \rightarrow \infty,\ \ k = 1$$

$$N_{patterns} \rightarrow \infty,\ \ k = \infty$$

Lazy Learners – k-Nearest-Neighbor Classifier

***k*-Nearest-Neighbor Classifier**

Attribute weighting

- *Each attribute carries same importance*
- *Better - lower weighting of irrelevant attributes*
- *Pruning of irrelevant patterns*

Complexity

- For a DB of D patterns and k=1, $O(D)$
- If patterns organized in search tree, then $O(log(D))$,
  - *growth of a decision tree is $O(log(n))$ when there are n leaves*
- Parallelization reduces O (up to $O(1)$)

Partial distance

- Distance between n attributes only (if these prove to be above threshold, remaining attributes are not checked)

Editing

- Pruning of irrelevant, redundant training patterns

Lazy Learners – Case-Based Reasoning

Case-Based Reasoning (CBR)

Algorithm

- Based on a DB of problem solutions (cases)
- (in *k*-nearest-neighbors, patterns are stored)
- E.g. case based law, medical case based treatments and diagnosis, engineering diagnostic problems (tech help)
- When unseen case is to be classified, a DB of similar cases is searched
- If identical training case is found, the accompanying solution is returned
- If no identical case is found,
  - the closest (neighboring) solution is returned
  - E.g. for solutions as graphs – subgraphs that are similar are searched for
- Problems
  - More training cases
  - Accuracy vs. efficiency

Lazy Learners – Distance Measures

Distances - Similarity measures

- Similarity based on certain similarity or distance metrics
- Various distance measures
  - **Euclidian space**
    - *Manhattan (1-norm)*
    - *Euclidean (2-norm)*
    - *Minkowski (p-norm)*
    - *Infinity-norm*
  - **Coding theory**
    - *Hamming*
  - **Fuzzy space**
    - *Fuzzy measures*

$$Dist_{Manhattan(p=1)} = \sum_{i=1}^{n} |x_i - y_i|$$

$$Dist_{Euclidian(p=2)} = \sqrt[2]{\sum_{i=1}^{n} |x_i - y_i|^2}$$

$$l^p(X,Y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

$$\lim_{p\to\infty}\left(l^p(X,Y)\right) = \lim_{p\to\infty}\left(\sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}\right) =$$

$$= \max(|x_1 - y_1|, |x_2 - y_2|, \ldots, |x_n - y_n|)$$

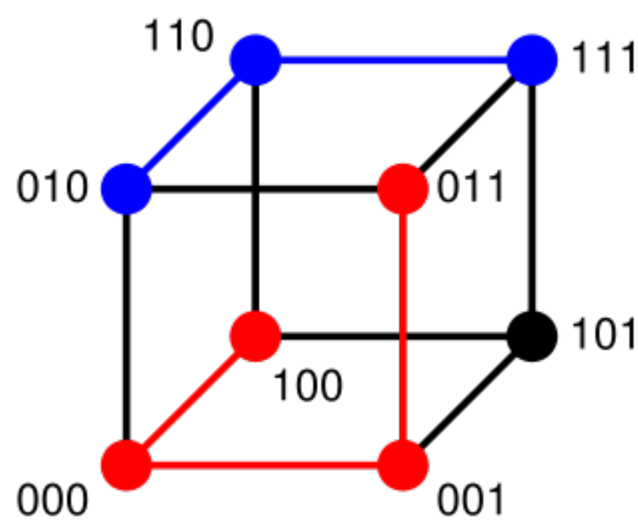
$$Dist_{Hamming(p=1)} = \left(\sum_{i=1}^{n} x_i \cdot y_i \;|\; x_i, y_i \in (0,1)\right)$$

Lazy Learners – Distance Measures

Distances - ***Hamming distance***

- *For two **strings** of equal length, HD is the number of positions for which the corresponding symbols are **different**.*
- *For binary strings: metric space for n-length binary strings - Hamming cube; HD= number of ones in a xor b*

- *E.g.*
  - **100->011 has distance 3 (red path)**
  - **010->111 has distance 2 (blue path)**
  - **HD=2**
    - **1011101**
    - **1001001**
  - **HD=3**
    - **2143896**
    - **2233796**
  - **HD=3**
    - **"toned"**
    - **"roses"**

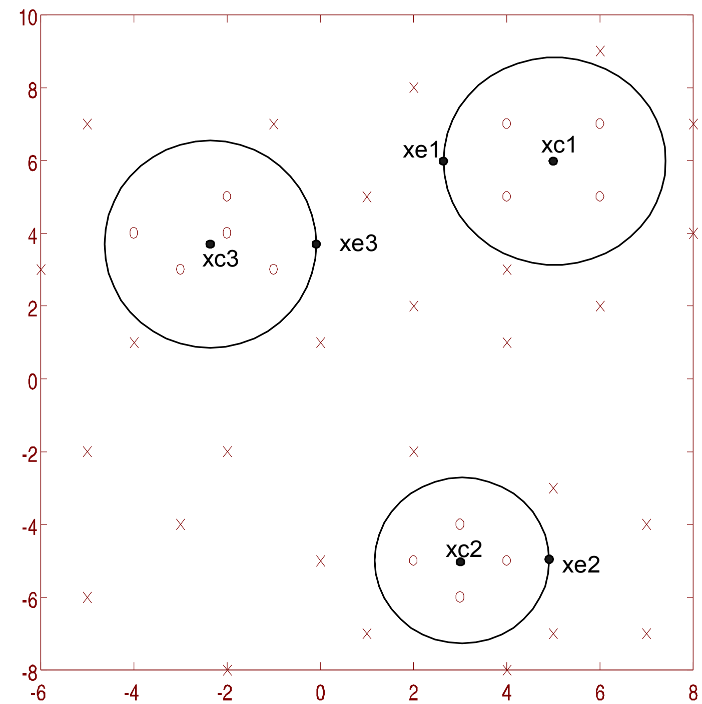


Lazy Learners – Distance Measures

Distances – ***Eucledian distance***

- ***Euclidian (Pythagorean) geometry****, considered by the Greek mathematician Euclid (300 BC)*
- *"ordinary" distance that can be measured by ruler*
- *Based on Pythagorean theorem*
- *2-norm distance*

$$Dist_{Euclidian(p=2)} = \sqrt[2]{\sum_{i=1}^{n} \left|x_i - y_i\right|^2}$$

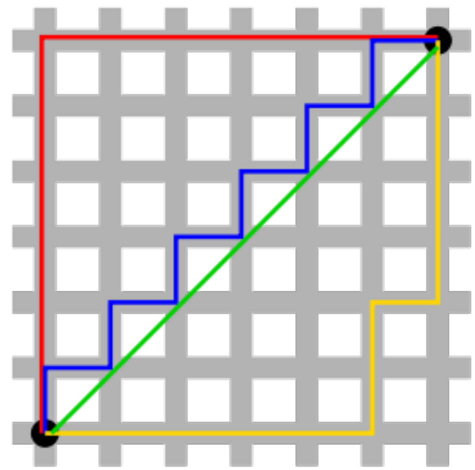


Lazy Learners – Distance Measures

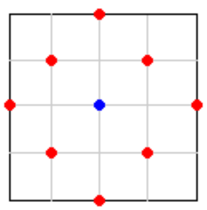
Distances – ***Manhattan or Citiblock distance***

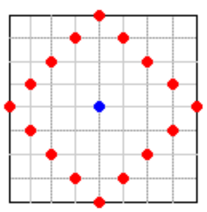
- ***Taxicab geometry****, considered by Hermann Minkowski in the 19th century*
- *Unlike Euclidean D., distances not squared (large difference in one dimension less likely to dominate the total distance)*

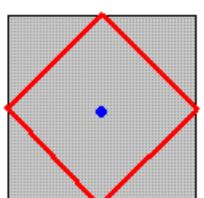
- *E.g.*
  - **red**, **blue**, **yellow = 12**
  - **green line** $\sqrt{6^2+6^2} \approx 8.48$



(L_1 metric (norm)







	Taxicab geometry	Euclidian geom.
shape	Circle	square
one side length	2r	r*sqr(2)
circumference	4*2r=8r	2*r*pi

Figures taken from
http://en.wikipedia.org/wiki/Taxicab_geometry

Lazy Learners – Distance Measures

Distances – ***Chebyshev (Tchebychev) distance***

- *Russian mathematician (18th century)*
- *Greatest distance of differences between two vectors along any coordinate dimension.*
- *In 2-dim space -* ***chessboard distance*** *(for a king)*
- *Infinity-norm distance*

$$Dist_{Chebyshev} = \max_i(|x_i - y_i|) = \lim_{k \to \infty} \sqrt[k]{\sum_{i=1}^{n} |x_i - y_i|^k}$$

(L_∞ metric)

number of moves a king requires

Figure taken from http://en.wikipedia.org/wiki/Chebyshev_distance

Lazy Learners – Distance Measures

Distances in chess

- *Distance between squares on the chessboard*
  - *To reach from one square to another, only kings require the number of moves equal to the distance; rooks, queens and bishops require one or two moves*

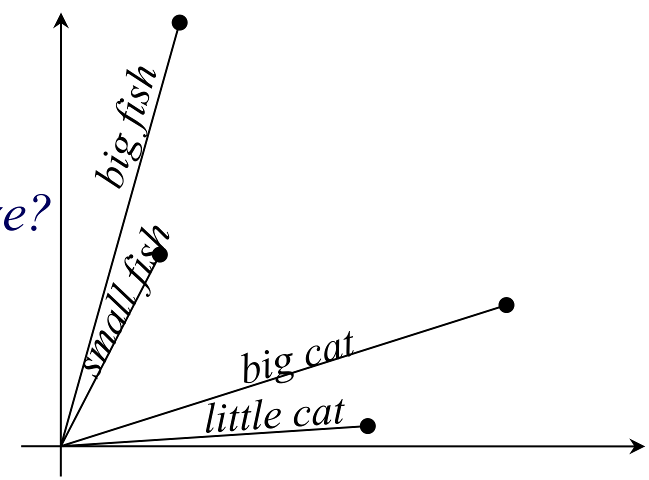
  - *For **rooks & bishops** (same color only) measured in Manhattan distance;*
  - *For **kings and queens** in Chebyshev distance*

pawn, rook, knight, bishop, queen, and king

Lazy Learners – Distance Measures

Distances – ***Angle distance***

- *Similarities in the way the fields within each record are related*
- E.g.
  - ***Species***
    - ***Sardines*** *should go with salmon, sardines, code, tuna, catfish*
    - ***Kitten*** *should go with lions, tigers, cougars*
  - ***Size*** *(kitten with sardines*
  - ***Whiskers*** *(kitten with catfish)*
- *How about the length of tail, body length, claw size?*
  - ***Single points*** *vs.* ***ratios of lengths*** *in each species!*



Angle

- ***Sine angle*** *rather than* ***magnitude***
- *Sine – relation*
  - *(0 & 180 different by constant factor -1);*
- *Cosine – correlation*
  - *(0 for orthogonal, 1 for parallel vectors)*