Machine Learning using MATLAB

Ankur Kulhari
Raju Pal
Himanshu Mittal

Department of Computer Science
Jaypee Institute of Information Technology, Noida

Outline

Machine Learning

- **Herbert Alexander Simon:** "Learning is any process by which a system improves performance from experience."
- "Machine Learning is concerned with computer programs that automatically improve their performance through experience."

Why Machine Learning?

- Discover new knowledge from large databases (data mining).
  -Market basket analysis (e.g. diapers and beer)
- Develop systems that can automatically adapt and customize themselves to individual users.
  -Personalized news or mail filter

Outline

ML Applications

Outline

The Learning Process

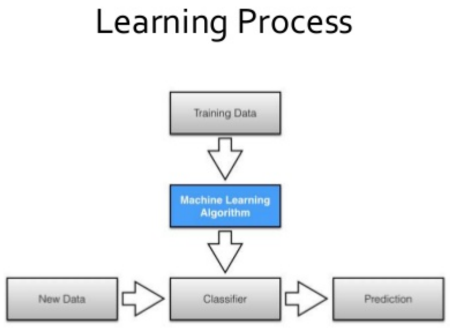


Figure: Supervised Learning Process

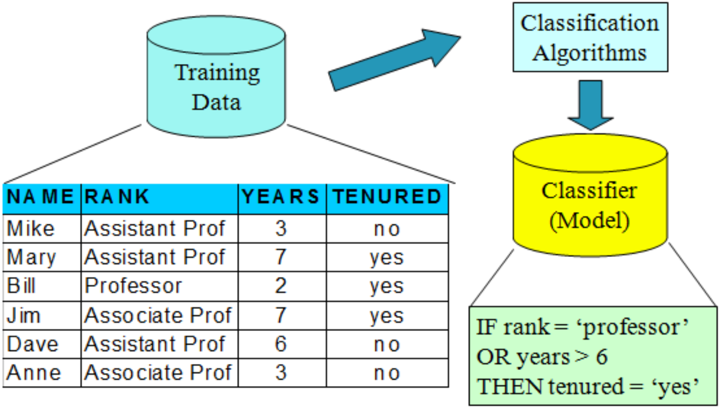
ML Categories

- Supervised Learning
  - Classification (discrete labels)
  - Regression (real values)
- Unsupervised Learning
  - Clustering
  - Dimension reduction
- Semi-Supervised Learning

Outline

Classification

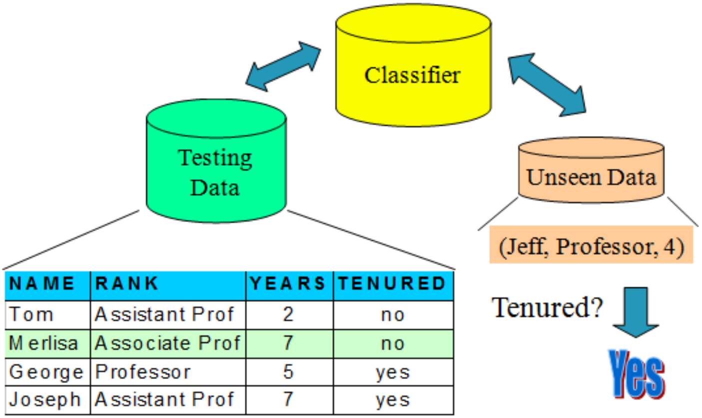
Model Construction



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classification

Using the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

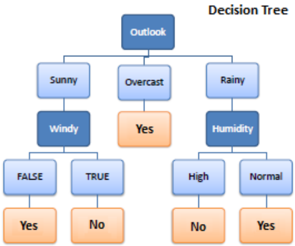
Outline

Classification

Decision Tree Induction

- Decision tree builds classification or regression models in the form of a tree structure.
- Leaf node represents a classification or decision.
- The topmost decision node in a tree which corresponds to the best predictor called root node.
- Decision trees can handle both categorical and numerical data.

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



Attribute Selection Measure

Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Expected information (entropy) needed to classify a tuple in D

$$Info(D) = -\sum p_i \log_2(p_i) \tag{1}$$

- Information needed (after using A to split D into v partitions) to classify D

$$Info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times info(D_j) \tag{2}$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D) \tag{3}$$

Decision Tree Induction

Data set

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Attribute Selection Measure

(Information Gain)

Class P: buys_computer = "yes"

Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

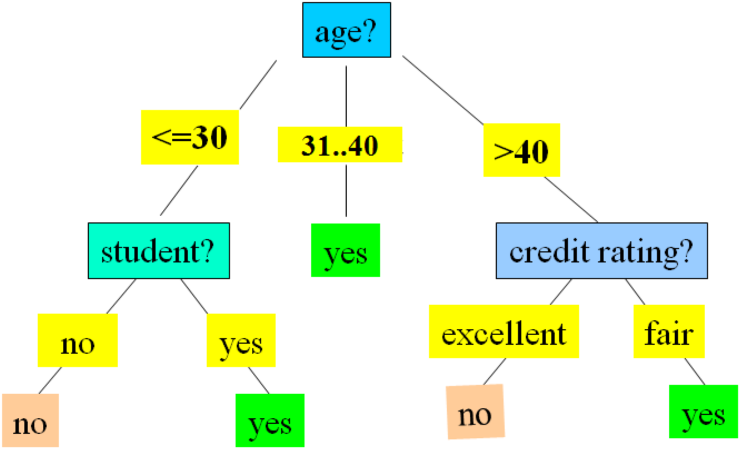
Similarly,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

Decision Tree Induction



Classification using Decision Tree

Predict Labels Using a Classification Tree

- **Load data set (Fisher's iris)**
  - load fisheriris
  - X = meas;
  - Y = species;
- **Partition the data into training (50%) and validation (50%) sets**
  - n = size(meas,1);
  - idxTrn = false(n,1);
- **Training set logical indices**
  - idxTrn(randsample(n,round(0.5*n))) = true;
- **Find the logical indices of validation set**
  - idxVal = idxTrn == false;

Classification using Decision Tree

Predict Labels Using a Classification Tree

- **Construct a classification tree using the training set**
  - Mdl = fitctree(meas(idxTrn,:), species(idxTrn));
- **Predict labels for the validation data**
  - label = predict(Mdl,meas(idxVal,:));
- **Display predicted labels for some random samples**
  - label(randsample(numel(label),5))
- **Count the number of miss-classified observations**
  - numMisclass = sum(strcmp(label,species(idxVal)))
- **Display Accuracy**
  -Accuracy=(size(ytest)-numMisclass)*100/size(ytest)

Outline

Classification

Naive Bayesian classifier

- Based on Bayes Theorem.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|\mathrm{X}) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

- A statistical classifier that performs probabilistic prediction
- Each training example can incrementally increase/decrease the probability that a hypothesis is correct.
- Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can't.

Classification using Naive Bayesian classifier

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Classification using Naive Bayesian classifier

age	income	studen	redit_ratin	_com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643

P(buys_computer = "no") = 5/14 = 0.357

Compute $P(X|C_i)$ for each class

P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222

P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6

P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444

P(income = "medium" | buys_computer = "no") = 2/5 = 0.4

P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667

P(student = "yes" | buys_computer = "no") = 1/5 = 0.2

P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667

P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044

P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028

P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Classification using Naive Bayesian classifier

- **Load data set (Fisher's iris)**
  - load fisheriris
  - X = meas; (Predictors)
  - Y = species; (Response)
- **Train the naive Bayes classifier and specify holdout 30% of the data for test sample**
  - CVMdl = fitcnb(X,Y,'Holdout',0.30, ... 'ClassNames','setosa','versicolor','virginica');
- **Extract trained, compact classifier**
  - CMdl = CVMdl.Trained1;
- **Extract the test indices**
  - testIdx = test(CVMdl.Partition);
  - XTest = X(testIdx,:);
  - YTest = Y(testIdx);

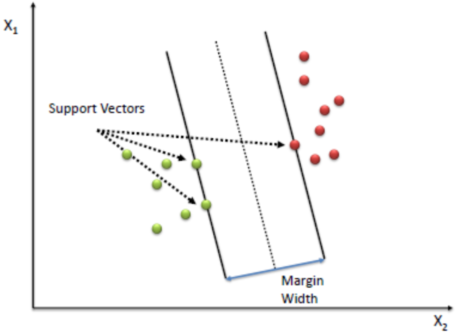
Classification using Naive Bayesian classifier

- **Label the test sample observations**
  - idx = randsample(sum(testIdx),10);
  - label = predict(CMdl,XTest);
- **Display the results for a random set of 10 observations in the test sample**
  - table(YTest(idx),label(idx),'VariableNames',... 'TrueLabel','PredictedLabel')
- **Count the number of miss-classified observations**
  - numMisclass = sum(strcmp(label,YTest));
- **Display Accuracy**
  - Accuracy=(size(YTest)-numMisclass)*100/size(YTest)
- **For Training Naive bayes on a data set and testing on a new sample like [1 2 3 4]**
  - Mdl = fitcnb(X,Y)
  - label = predict(Mdl,X);

Outline

Classification using Support Vector Machine

- A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors (cases) that define the hyperplane are the support vectors.



Classification using Support Vector Machine

- **Load data set (Fisher's iris)**
  - load fisheriris
  - X = meas; (Predictors)
  - Y = categorical(species); (Response)
- **Train an ECOC model using SVM binary classifiers and specify holdout 30% of the data for test sample**
  - CVMdl = fitcecoc(X,Y,'Holdout',0.30);
- **Extract trained, compact classifier**
  - CMdl = CVMdl.Trained1;
- **Extract the test indices**
  - testIdx = test(CVMdl.Partition);
  - XTest = X(testIdx,:);
  - YTest = Y(testIdx,:);

Classification using Support Vector Machine

- **Predict the test-sample labels**
  - labels = predict(CMdl,XTest);
  - idx = randsample(sum(testInds),10);
- **Print a random subset of true and predicted labels**
  - table(YTest(idx),labels(idx),'VariableNames',...
  'TrueLabels','PredictedLabels')
- **Display Confusion Matrix**
  - [C,order] = confusionmat(YTest,labels)
- **Count the number of miss-classified observations**
  - numMisclass = sum(strcmp(labels,YTest));
- **Display Accuracy**
  - Accuracy=(size(YTest)-numMisclass)*100/size(YTest)

Outline

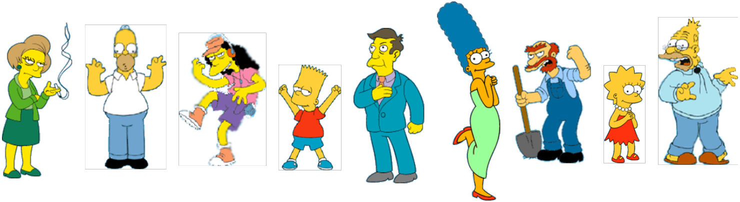
What is Clustering?

- Organizing data into groups or classes such that there is
  - high intra-class similarity
  - low inter-class similarity

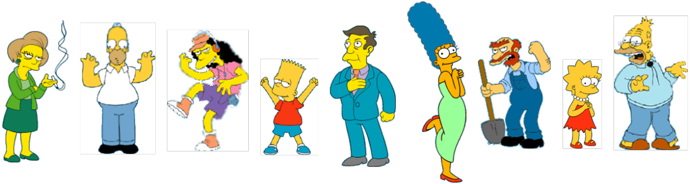
- Finding the class labels and the number of classes directly from the data
  (in contrast to classification)

- More informally, finding natural groupings among objects.

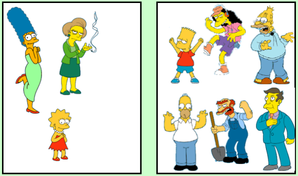
What is a natural grouping among these objects?



What is a natural grouping among these objects?



Clustering is subjective



Females Males

K-Means

- Widely used algorithm for performing clustering
- k-means minimizes within-cluster point scatter
- Dissimilarity measure: Euclidean distance metric
- **Algorithm:**
  1. Randomly choose ***k*** data items from dataset X as initial centroids.
  2. Repeat until the convergence criteria is met.
     - Assign each data point to the cluster with closet centroid based on the distance measure.
     - Update cluster centroids using the mean of data items.
- Number of clusters to be created needs to be known inadvance
- Different initial partitions result in different final clusters

Clustering using k-Means

- Eg; identify the cluster for newdata (1.2,0.5) using the 'fisher's iris' dataset.
  - Load 'Fisher's iris' data set.

    load fisheriris;
  - Use the petal lengths and widths as predictors.

    x = meas(:,3:4);
  - Clustering the data with $k = 3$ clusters

    [idx,C] = kmeans(x,3);

    where, idx is a column vector of predicted cluster indices for each observation and C is matrix of centroid locations.
  - Load the data of cluster to be identified

    newData=[1.2 0.5]
  - Calculate the distance b/w newData and identified centroids

    dis=pdist2(C,newData)
  - Find the cluster with minimum distance

    [M,I] = min(dis)

    where, M represents the distance of newData from identified cluster and I represents the cluster number

Machine Learning Excercise

Compare the performance of naive bayes and support vector machine classifier over pima-indians-diabetes dataset.

-Use 80% sample for training and 20% for testing.

Dataset link :

https://data.world/data-society/pima-indians-diabetes-database

For Further Reading I

Ethem Alpaydin.
Introduction to Machine Learning.
MIT Press.

Jiawei Han, Micheline Kamber, and Jian Pei
Data Mining: Concepts and Techniques ,
The Morgan Kaufmann