Text Categorization

Introduction

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Definition:

- Text categorization is the process of organizing textual data into categories, making it easier to manage, retrieve, and analyze information.

Importance:

- Facilitates quick access to relevant information.
- Enhances search capabilities and content organization.
- Improves user experience in information systems.

Applications:

- Search engines (e.g., Google categorizing search results).
- Recommendation systems (e.g., Netflix recommending movies).
- Digital libraries (e.g., categorizing books and articles).
- Content management systems (e.g., tagging blog posts).

Categorization

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Text categorization (TC) - given a set of categories (subjects, topics) and a collection of text documents, the process of finding the correct topic (or topics) for each document.

Given:

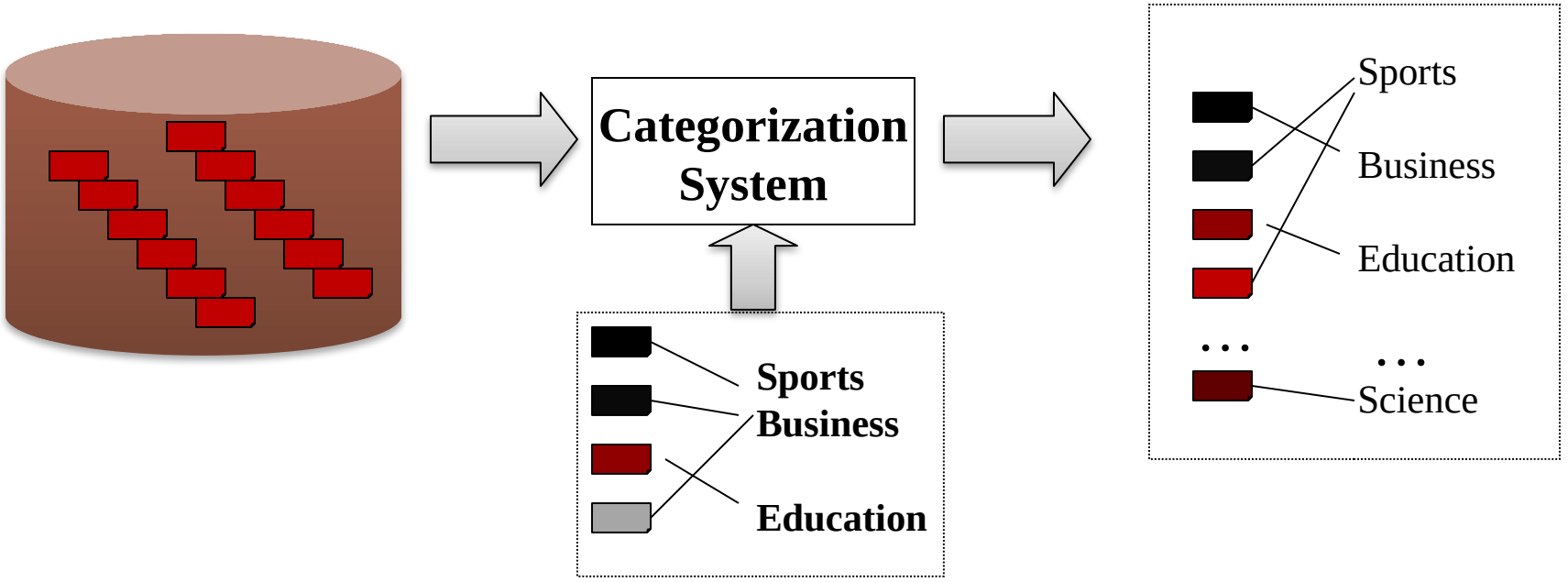
- A description of an instance, $x \in X$, where X is the instance language or instance space.
- A fixed set of categories: $C=\{c_1, c_2,\ldots c_n\}$

Determine:

- The category of x: $c(x) \in C$, where c(x) is a categorization function whose domain is X and whose range is C.

Text Categorization

- Pre-given categories and labeled document examples (Categories may form hierarchy)
- Classify new documents
- A standard classification (supervised learning) problem



Definition of Text Categorization

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- The task of approximating an unknown category assignment function $F:D \times C \rightarrow \{0,1\}$,

  Where, D - set of all possible documents and

  C - set of predefined categories.

- The value of F(d, c) is 1 if the document d belongs to the category c and 0 otherwise.
- The approximating function $M:D \times C \rightarrow \{0,1\}$ is called a classifier, and the task is to build a classifier that produces results as "close" as possible to the true category assignment function F.
- Classification Problems
  - Single-Label versus Multi-label Categorization
  - Document-Pivoted versus Category-Pivoted Categorization
  - Hard versus Soft Categorization

Classification Problems

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Single-Label versus Multi-label Categorization

- In single-label categorization, each document belongs to exactly one category.
- In multi-label categorization the categories overlap, and a document may belong to any number of categories.

Document-Pivoted versus Category-Pivoted Categorization

- For a given document, the classifier finds all categories to which the document belongs is called a document-pivoted categorization.
- Finding all documents that should be filed under a given category is called a category-pivoted categorization.

Hard versus Soft Categorization

- A particular label is explicitly assigned to the instance is called Soft/Ranking Categorization.
- A probability value is assigned to the test instance is known as Hard Categorization.

Categorization Methods

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Manual: Typically rule-based

- Does not scale up (labor-intensive, rule inconsistency)
- May be appropriate for special data on a particular domain

Automatic: Typically exploiting machine learning techniques

- Vector space model based
  - ✓ K-nearest neighbor (KNN)
  - ✓ Decision-tree (learn rules)
  - ✓ Neural Networks (learn non-linear classifier)
  - ✓ Support Vector Machines (SVM)
- Probabilistic or generative model based
  - ✓ Naïve Bayes classifier

Feature Selection

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Feature selection:

- Removes non-informative terms (irrelevant words) from documents.
- Improves classification effectiveness.
- Reduces computational complexity.

Measures of feature relevance:

- Relations between features and the categories.
- Feature Selection Methods
  - Document Frequency Threshold (DF)
  - Information Gain (IG)
  - χ2statistic (CHI)
  - Mutual Information (MI)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Document Frequency Threshold (DF)
  - Document frequency DocFreq(w) *is* a measure of the relevance of each feature in the document

- Information Gain (IG)
  - Measures the number of bits of information obtained for the prediction of categories by the presence or absence in a document of the feature f.
  - The probabilities are ratios of frequencies in the training data.

$$IG(w) = \sum_{c \in C \cup \overline{C}} \sum_{f \in \{w, \overline{w}\}} P(f, c).\log \frac{P(c \mid f)}{P(c)}$$

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Chi-square
  - Measures the maximal strength of dependence between the feature and the categories.

$$\chi^2_{\max}(f) = \max_{c \in C} \frac{|T_r| .(P(f,c).P(\bar{f},\bar{c}) - P(f,\bar{c}).P(\bar{f},c))^2}{P(f).P(\bar{f}).P(c).P(\bar{c})}$$

- Mutual Information

$$MI(f) = \log(\frac{1}{P(\bar{c})}) - \log(\frac{1}{P(\bar{c}|f)}) = \log(\frac{P(f,\bar{c})}{P(f)P(\bar{c})})$$

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Dimensionality Reduction by Feature Extraction

- Feature reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space
- Criterion for feature reduction can be different based on different problem settings.
  - Unsupervised setting: minimize the information loss
  - Supervised setting: maximize the class discrimination

Feature Reduction Algorithms

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Unsupervised
  - Latent Semantic Indexing (LSI): truncated SVD
  - Independent Component Analysis (ICA)
  - Principal Component Analysis (PCA)
  - Manifold learning algorithms
- Supervised
  - Linear Discriminant Analysis (LDA)
  - Canonical Correlation Analysis (CCA)
  - Partial Least Squares (PLS)
- Semi-supervised

Feature Reduction Algorithms

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Linear
  - Latent Semantic Indexing (LSI): truncated SVD
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Canonical Correlation Analysis (CCA)
  - Partial Least Squares (PLS)
- Nonlinear
  - Nonlinear feature reduction using kernels
  - Manifold learning

Machine Learning Approach to Text Categorization

- In the ML approach, the classifier is built automatically by learning the properties of categories from a set of pre-classified training documents.
- Learning process is an instance of
  - Supervised Learning - process of applying the known true category assignment function on the training set.
  - The unsupervised version of the classification task, called clustering.
- Issues in using machine learning techniques to develop an application based on text categorization:
  - Choose how to decide on the categories that will be used to classify the instances.
  - Choose how to provide a training set for each of the categories.
  - Choose how to represent each of the instances on the features.
  - Choose a learning algorithm to be used for the categorization.

Approaches to Classifier Learning

- Probabilistic Classifiers
- Bayesian Logistic Regression
- Decision Tree Classifiers
- Decision Rule Classifiers
- Regression Methods
- Neural Networks
- Example-Based Classifiers
- Support Vector Machines
- Classifier Committees: Bagging and Boosting

Probabilistic Classifiers

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Probabilistic classifiers view the categorization status value CSV(d,c),
  - The probability P(c | d) that the document d belongs to the category c and compute this probability by application of Bayes' theorem

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

  - P(c|d): Probability that document d belongs to category c.
  - P(d | c): Probability that category c would produce document d.
  - P(c): Prior probability of category c.
  - P(d): Total probability of document d (across all categories).
- The marginal probability P(d) is constant for all categories.
  - The term P(d) is the same for all categories and often treated as a constant. This is because it is the total probability of the document d, regardless of which category it belongs to.

Cont…

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- To calculate P(d | c), however, we need to make some assumptions about the structure of the document d.
- **Document Representation as a Feature Vector:**
  - Documents are often represented as feature vectors d=(w_1, w_2, . . .) where each w_i represents a word or feature in the document.

- **Naive Bayes Assumption:**
  - The **Naive Bayes** classifier assumes that all features (words) in the document are independent of each other, given the category c. This is known as the **naive** assumption. With this assumption, the probability P(d | c) can be broken down as:

$$P(d \mid c) = \prod_i P(w_i \mid c)$$

This simplifies the calculation of P(d | c) since instead of considering the entire document as a whole, you consider each word independently.

Cont…

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- The Naive Bayes classifier uses these assumptions to efficiently compute the probability P(c | d) for each category c and then assigns the document d to the category with the highest probability.

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Cont…

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Cont…

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Applying Bayes'theorem:

P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)

P(Sunny|Yes)= 3/10= 0.3

P(Sunny)= 0.35

P(Yes)=0.71

P(No)= 0.29

P(Sunny)= 0.35

So P(Yes|Sunny) = 0.3*0.71/0.35= **0.60**

P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)

P(Sunny|NO)= 2/4=0.5

So P(No|Sunny)= 0.5*0.29/0.35 = **0.41**

So as we can see from the above calculation that **P(Yes|Sunny)>P(No|Sunny)**

Hence on a Sunny day, Player can play the game.

Analysis of Naïve Bayes Algorithm

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Advantages:

- Work well on numeric and textual data
- Easy to implement and computation comparing with other algorithms

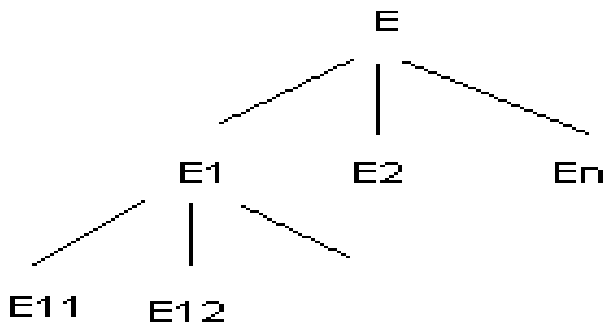
Disadvantages:

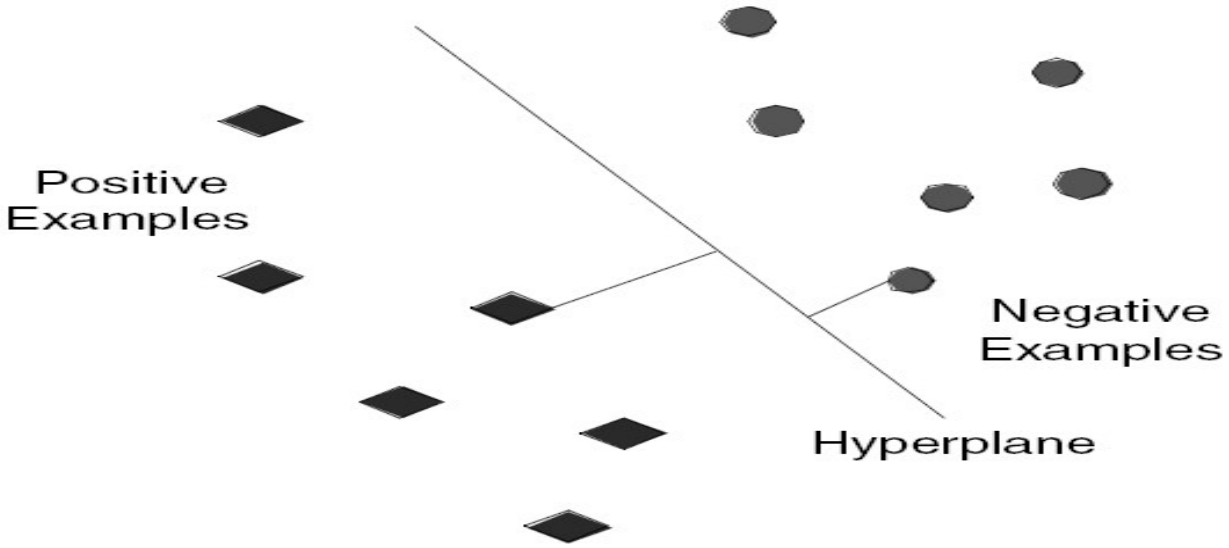
- Conditional independence assumption is violated by real-world data, perform very poorly when features are highly correlated

Decision Tree Classifiers

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- A decision tree (DT) classifier is a tree in which the internal nodes are labeled by the features, the edges leaving a node are labeled by tests on the feature's weight, and the leaves are labeled by categories.
- Decision tree associated with document:
  - Root node contains all documents
  - Each internal node is subset of documents separated according to one attribute
  - Each arc is labeled with predicate which can be applied to attribute at parent
  - Each leaf node is labeled with a class





THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Fig: A Decision Tree Classifier

The tree that corresponds to the CONSTRUE rule

- Performance of a DT classifier is mixed but is inferior to the top-ranking classifiers.
- Recursive partition procedure from root node
- Set of documents separated into subsets according to an attribute
- Use the most discriminative attribute first (highest IG)
- Pruning to deal with over-fitting

- Hierarchical classifier compares the data sequentially with carefully selected features.
- Features are determined from the spectral distributions or separability of the classes.
- There is no general procedure. Each decision tree or set of rules is custom-designed.
- A decision tree that provides only two outcomes at each stage is called a “binary decision tree” (BDT) classifier

Example

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Attributes				Classes
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Algorithm

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Calculate the Entropy of every attribute using the data set.
- Split the set into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum).
- Make a decision tree node containing that attribute.
- Recurse on subsets using remaining attributes.

Entropy

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- In order to define Information Gain precisely, we need to discuss Entropy first.
- A formula to calculate the homogeneity of a sample.
- A completely homogeneous sample has entropy of 0 (leaf node).
- An equally divided sample has entropy of 1.
- The formula for entropy is:

Entropy(S)

- where p(I) is the proportion of S belonging to class I. ∑ is over total outcomes.
- Example 1

If S is a collection of 14 examples with 9 YES and 5 NO examples then

Entropy(S) = - (9/14) Log2 (9/14) - (5/14) Log2 (5/14)

= 0.940

Information gain

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- The information gain is based on the decrease in entropy after a dataset is split on an attribute.
- The formula for calculating information gain is:

$$Gain(S, A) = Entropy(S) - ((|S_v| / |S|) * Entropy(S_v))$$

Where, S_v = subset of S for which attribute A has value v

- $|S_v|$ = number of elements in S_v
- $|S|$ = number of elements in S

Procedure

- First the entropy of the total dataset is calculated.
- The dataset is then split on the different attributes.
- The entropy for each branch is calculated.
- Then it is added proportionally, to get total entropy for the split.
- The resulting entropy is subtracted from the entropy before the split.
- The result is the Information Gain, or decrease in entropy.
- The attribute that yields the largest IG is chosen for the decision node.

Name	Gender	Car Ownership	Travel Cost	Income Level	Transportation
Abhi	Male	1	Standard	High	?
Pavi	Male	0	Cheap	Medium	?
Ammu	Female	1	Cheap	High	?

Attributes				Classes
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Calculate the entropy of the total dataset

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- First compute the Entropy of given training set.

Probability
Bus : 4/10 = 0.4
Train: 3/10 = 0.3
Car:3/10 = 0.3

$E(S) = -P(I) \log_2 P(I)$

$E(S) = -(0.4) \log_2 (0.4) – (0.3)\log_2 (0.3) – (0.3)\log_2 (0.3) = 1.571$

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Attributes				Classes
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Attributes	Classes
Gender	Transportation
Male	Bus
Male	Bus
Female	Train
Female	Bus
Male	Bus
Male	Train
Female	Train
Female	Car
Male	Car
Female	Car

Split the dataset on 'Gender' attribute

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Attributes	Classes
Gender	Transportation
Male	Bus
Male	Bus
Male	Bus
Male	Train
Male	Car

Probability
Bus : 3/5 = 0.6
Train: 1/5 = 0.2
Car:1/5 = 0.2

Attributes	Classes
Gender	Transportation
Female	Train
Female	Bus
Female	Train
Female	Car
Female	Car

Probability
Bus : 1/5 = 0.2
Train: 2/5 = 0.4
Car:2/5 = 0.4

Gain(S,A) = E(S) – I(S,A)
*I(S,A) = 1.522 *(5/10) + 1.371*(5*

Gain(S,A) = 1.571 – 1.447
= 0.12

$E(S_v) = -0.6 \log_2(0.6) - 0.2 \log_2(0.2) - 0.2 \log_2(0.2)$
$= 1.522$

$E(S_v) = -0.2 \log_2(0.2) - 0.4 \log_2(0.4) - 0.4 \log_2(0.4)$
$= 1.371$

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Attributes				Classes
Gender	Car Ownership	Travel Cost	Income Level	Transportation
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Attributes	Classes
Car Ownership	Transportation
0	Bus
1	Bus
1	Train
0	Bus
1	Bus
0	Train
1	Train
1	Car
2	Car
2	Car

Split the dataset on 'ownership' attribute

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Attributes	Classes
Car Ownership	Transportation
0	Bus
0	Bus
0	Train

Attributes	Classes
Car Ownership	Transportation
1	Bus
1	Train
1	Bus
1	Train
1	Car

Attributes	Classes
Car Ownership	Transportation
2	Car
2	Car

Gain(S,A) = E(S) – I(S,A)
*I(S,A) = 0.918 *(3/10) + 1.522*(5/10) + 0*(2/10)*

Gain(S,A) = 1.571 – 1.0364 = 0.534

Probability
Bus : 2/3= 0.6
Train: 1/3 = 0.3
Car:0/3 = 0
Entropy = 0.918

Probability
Bus : 2/5 = 0.4
Train: 2/5 = 0.4
Car:1/5 = 0.2
Entropy = 1.522

Probability
Bus : 0/2 = 0
Train: 0/2 = 0
Car:2/2 = 1
Entropy = 0

- If we choose Travel Cost as splitting attribute,

-Entropy for Cheap = 0.722
Standard = 0
Expensive = 0
IG = 1.21

- If we choose Income Level as splitting attribute,

-Entropy for Low = 0
Medium = 1.459
High = 0
IG = 0.695

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Attribute	Information Gain
Gender	0.125
Car Ownership	0.534
Travel Cost	1.21
Income Level	0.695

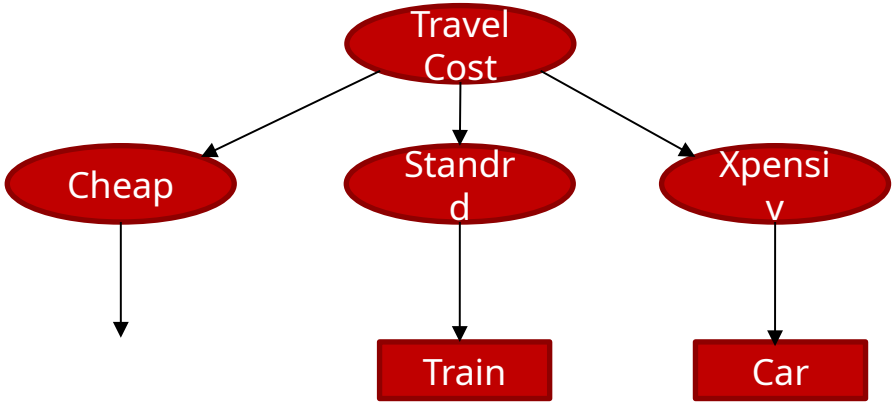
Attributes				Classes
Travel Cost	Gender	Car Ownership	Income Level	Transportation
Cheap	Male	0	Low	Bus
Cheap	Male	1	Medium	Bus
Cheap	Female	1	Medium	Train
Cheap	Female	0	Low	Bus
Cheap	Male	1	Medium	Bus

Attributes				Classes
Travel Cost	Gender	Car Ownership	Income Level	Transportation
Standard	Male	0	Medium	Train
Standard	Female	1	Medium	Train

Attributes				Classes
Travel Cost	Gender	Car Ownership	Income Level	Transportation
Expensive	Female	1	High	Car
Expensive	Male	2	Medium	Car
Expensive	Female	2	High	Car

Diagram : Decision Tree

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)



?

Iteration on Subset of Training Set

Attributes				Classes
Travel Cost	Gender	Car Ownership	Income Level	Transportation
Cheap	Male	0	Low	Bus
Cheap	Male	1	Medium	Bus
Cheap	Female	1	Medium	Train
Cheap	Female	0	Low	Bus
Cheap	Male	1	Medium	Bus

Probability
Bus : 4/5 = 0.8
Train: 1/5 = 0.2
Car:0/5 = 0

$E(S) = -P(I) \log_2 P(I)$

$E(S) = -(0.8) \log_2 (0.8) - (0.2) \log_2 (0.2) = 0.722$

- If we choose Gender as splitting attribute,

-Entropy for Male = 0

Female = 1

IG = 0.322

- If we choose Car Ownership as splitting attribute,

-Entropy for 0 = 0

1 = 0.918

IG = 0.171

- If we choose Income Level as splitting attribute,

-Entropy for Low = 0

Medium = 0.918

IG = 0.171

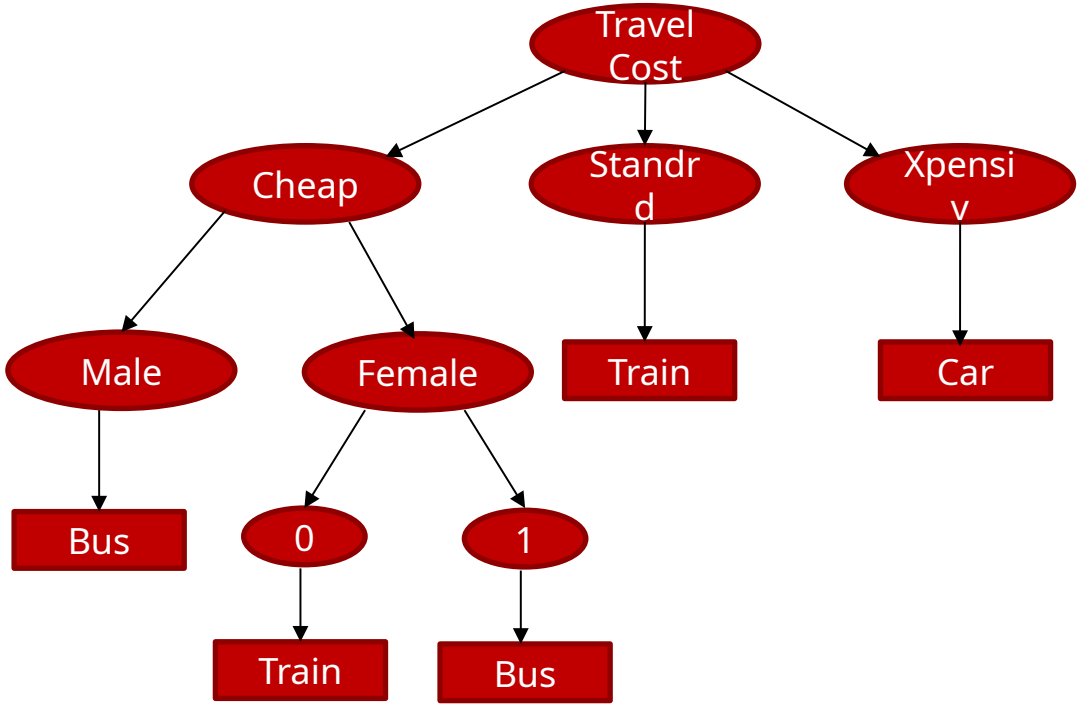
THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Attributes	Information Gain
Gender	0.322
Car Ownership	0.171
Income Level	0.171

Attributes			Classes
Gender	Car Ownership	Income Level	Transportation
Male	0	Low	Bus
Male	1	Medium	Bus
Male	1	Medium	Bus

Attributes			Classes
Gender	Car Ownership	Income Level	Transportation
Female	1	Medium	Train
Female	0	Low	Bus

Diagram : Decision Tree



Solution to Our Problem :

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Name	Gender	Car Ownership	Travel Cost	Income Level	Transportation
Abhi	Male	1	Standard	High	Train
Pavi	Male	0	Cheap	Medium	Bus
Ammu	Female	1	Cheap	High	Bus

Analysis of Decision Tree Algorithm

Advantages:

- Easy to understand
- Easy to generate rules
- Reduce problem complexity

Disadvantages:

- Training time is relatively expensive
- A document is only connected with one branch
- Once a mistake is made at a higher level, any sub-tree is wrong
- Does not handle continuous variable well
- May suffer from over-fitting

Decision Rule Classifiers

- Decision rule (DR) classifiers are also symbolic like decision trees and are built from the training collection using Inductive Rule Learning.
- DNF(Disjunctive Normal Form) rules are built in a bottom-up fashion.
- DNF rules are logical formulas that consist of a disjunction (OR) of conjunctions (AND). In simple terms, a DNF rule looks like a series of "AND" conditions linked by "OR" operators.
- DR classifiers are built in a bottom-up fashion, meaning they start by creating rules that are very specific and then generalize them over time.
- The process begins by viewing each document in the training set as a clause, which is a simple "AND" condition combining the document's features.

$$d_1 \wedge d_2 \wedge \ldots \wedge d_n \rightarrow c$$

Where, d_i - features of the document and c its category.

- After the initial specific rules are created, the learner applies **generalizations** to these rules.
- The goal is to **maximize the compactness** of the rules while ensuring they maintain their ability to correctly classify documents (referred to as the **covering property**).
  - **Compactness:** Making the rules simpler by removing unnecessary terms from the clauses.
  - **Merging Rules:** Combining similar rules to create more general rules.
- At the end of the process, a pruning step similar to the DT pruning is applied that trades covering for more generality.
  - Pruning in DR classifiers is similar to pruning in decision trees. It involves removing parts of the rule that may cause overfitting, making the rule more general but still effective in classification.

- Rule learners vary widely in their specific methods, heuristics, and optimality criteria.
- Example:
  - RIPPER (Repeated Incremental Pruning to Produce Error Reduction) (Cohen 1995a; Cohen 1995b; Cohen and Singer 1996).
  - Ripper builds a rule set by
    - ✓ Adding new rules until all positive category instances are covered and
    - ✓ Adding conditions to the rules until no negative instance is covered.
  - Features of Ripper is its ability to bias the performance toward higher precision or higher recall.
  - Determined by the loss ratio parameter, which measures the relative cost of “false negative” and “false positive” errors.

Example

Neural Networks

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Neural network (NN) can be built to perform text categorization.
  - The input nodes of the network receive the feature values,
  - The output nodes produce the categorization status values, and
  - The link weights represent dependence relations.
- For classifying a document,
  - Its feature weights are loaded into the input nodes;
  - The activation of the nodes is propagated forward through the network, and
  - The final values on output nodes determine the categorization decisions.

- The neural networks are trained by back propagation, whereby the training documents are loaded into the input nodes.
- If a misclassification occurs, the error is propagated back through the network, modifying the link weights in order to minimize the error.
- The simplest kind of a neural network is a perceptron
  - It has only two layers - the input and the output nodes.
  - Such network is equivalent to a linear classifier.
- More complex networks contain one or more hidden layers between the input and output layers.
- The experiments (Schutze, Hull, and Pederson 1995; Wiener 1995) shows very small or no improvement of nonlinear networks over their linear counterparts in the text categorization task.

One Neuron as a Network

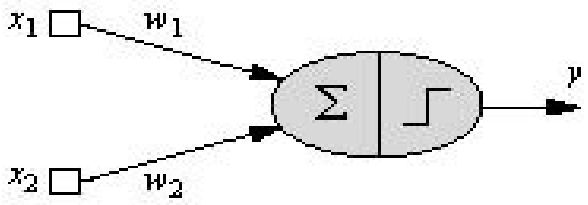


Fig1: an artificial neuron

- Here x1 and x2 are normalized attribute value of data.
- y is the output of the neuron , i.e the class label.
- x1 and x2 values multiplied by weight values w1 and w2 are input to the neuron x.
- Value of x1 is multiplied by a weight w1 and values of x2 is multiplied by a weight w2.
- Given that
  - w1 = 0.5 and w2 = 0.5
  - Say value of x1 is 0.3 and value of x2 is 0.8,
  - So, weighted sum is :
  - sum= w1 x x1 + w2 x x2 = 0.5 x 0.3 + 0.5 x 0.8 = 0.55

One Neuron as a Network

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- **The neuron receives the weighted sum as input and calculates the output as a function of input as follows :**
- **y = f(x) , where f(x) is defined as**
- **f(x) = 0 { when x< 0.5 }**
- **f(x) = 1 { when x >= 0.5 }**
- **For our example, x (weighted sum) is 0.55, so y = 1 ,**
- That means corresponding input attribute values are classified in class 1.
- **If for another input values , x = 0.45 , then f(x) = 0,**
- **so we could conclude that input values are classified to class 0.**

Analysis of NN Algorithm

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Advantages:

- Produce good results in complex domains
- Suitable for both discrete and continuous data (especially better for the continuous domain)
- Testing is very fast

Disadvantages:

- Training is relatively slow
- Learned results are difficult for users to interpret than learned rules (comparing with DT)
- Empirical Risk Minimization (ERM) makes NN try to minimize training error, may lead to over-fitting

Example-Based Classifiers

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

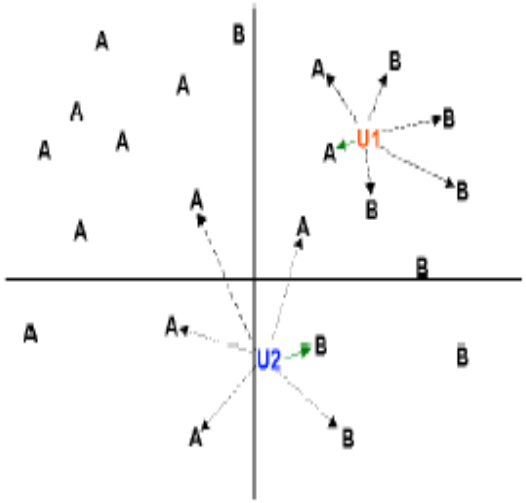
- Rely on computing the similarity between the document to be classified and the training documents.
- These methods are called lazy learners because they defer the decision on how to generalize beyond the training data until each new query instance is encountered.
- "Training" for such classifiers consists of simply storing the representations of the training documents together with their category labels.
- The most prominent example of an example-based classifier is kNN (k-nearest neighbor).
  - To decide whether a document d belongs to the category c, kNN checks whether the k training documents most similar to d belong to c
  - If the answer is positive for a sufficiently large proportion of them, a positive decision is made; otherwise, the decision is negative.

K-Nearest-Neighbor Algorithm

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

- Principle: points (documents) that are close in the space belong to the same class

  - Calculate similarity between test document and each neighbor
  - Select k nearest neighbors of a test document among training examples
  - Assign test document to the class which contains most of the neighbors



Analysis of k-NN Algorithm

Advantages:

- Effective
- one of the best-performing text classifiers
- Non-parametric
- Robust in terms of not requiring the categories to be linearly separated
- More local characteristics of document are considered comparing with Rocchio

Disadvantages:

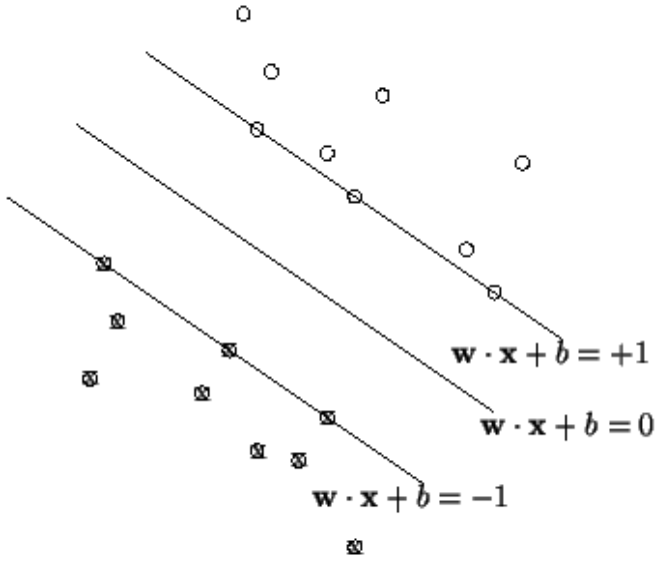
- High computational cost
- Classification time is long
- Difficult to find optimal value of k

Support Vector Machines

The support vector machine (SVM) algorithm is very fast and effective for text classification problems.

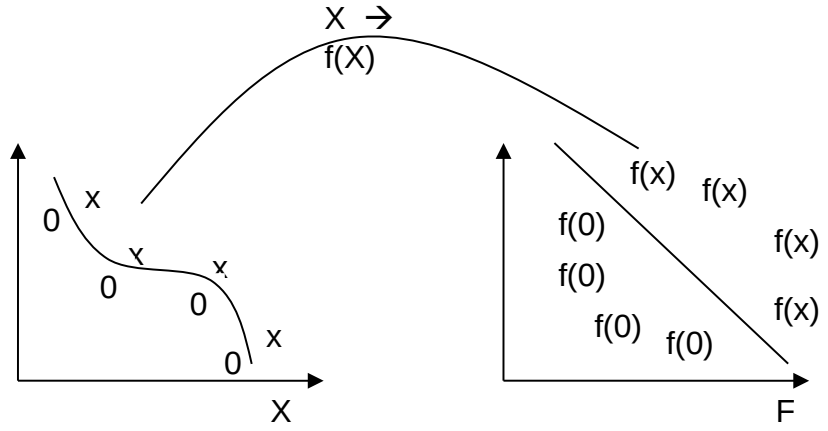
- Main idea of SVMs

Find out the linear separating hyperplane which maximize the margin, i.e., the optimal separating hyperplane (OSH)



- Nonlinear separable case

Kernel function and Hilbert space



SVM classification

Maximizing the margin is equivalent to:

$$\underset{w,\, b,\, \zeta_i}{\text{minimize}} \quad \frac{1}{2} w^T w + C\left(\sum_{i=1}^{N} \zeta_i\right)$$

$$\text{subject to} \quad y_i (w^T x_i - b) + \zeta_i - 1 \geq 0, \quad 1 \leq i \leq N$$

$$\zeta_i \geq 0, \quad 1 \leq i \leq N$$

Introducing Lagrange multipliers α, β , the Lagrangian is:

$$\begin{aligned} \ell(w, b, \zeta_i; \alpha, \beta) &= \frac{1}{2} w^T w + C \sum_{i=1}^{N} \zeta_i \\ &\quad - \sum_{i=1}^{N} \alpha_i [y_i (w^T x_i - b) + \zeta_i - 1] - \sum_{i=1}^{N} \mu_i \zeta_i \\ &= \frac{1}{2} w^T w + \sum_{i=1}^{N} (C - \alpha_i - \mu_i) \zeta_i \\ &\quad - \left(\sum_{i=1}^{N} \alpha_i y_i x_i^T \right) w - \left(\sum_{i=1}^{N} \alpha_i y_i \right) b + \sum_{i=1}^{N} \alpha_i \end{aligned}$$

Dual problem:

$$\underset{\alpha}{\text{maximize}} \ell_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

subject to:

$$0 \leq \alpha_i \leq C, \qquad \sum_i \alpha_i y_i = 0.$$

The solution is given by:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \qquad b = y_j - w \cdot x_j$$

The problem of classifying a new data point x is now simply solved by looking at the sigh of

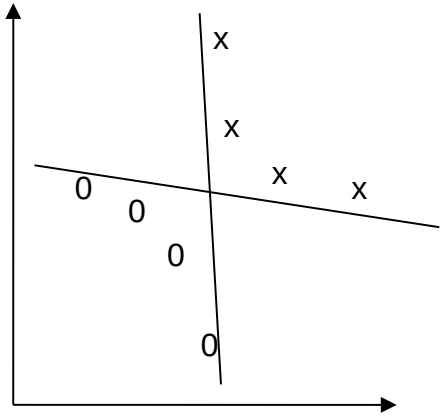
$$w \cdot x + b$$

Analysis of SVM Algorithm

Advantages:

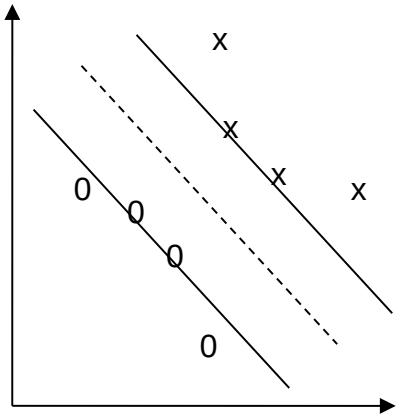
- Comparing with NN, SVM capture the inherent characteristics of the data better
- Embedding the Structural Risk Minimization (SRM) principle, which minimizes the upper bound on the generalization error

  (better than the Empirical Risk Minimization principle)
- Ability to learn can be independent of the dimensionality of the feature space
- Global minima vs. Local minima



Disadvantage:

- Parameter Tuning
- Kernel Selection



Bayesian Logistic Regression

- BLR is a statistical approach often applied to text categorization (TC) problems, and it tends to exhibit very high performance in these tasks.
- Let categorization be binary, then the logistic regression model has the form,

$$P(c \mid d) = \varphi(\beta . d) = \varphi(\sum_i \beta_i d_i)$$

where ,

c = ±1 - category membership value,

d = (d1, d2, . . .) - document representation in the feature space,

β = (β1, β2, . . .) - model parameters vector, and

ϕ - logistic link function.

$$\varphi(x) = \frac{\exp(x)}{1 + \exp(x)} = \frac{1}{1 + \exp(-x)}$$

- The Bayesian approach is to use a prior distribution for the parameter vector β. Different priors are possible,
  - Gaussian and
  - Laplace priors.
- Gaussian prior with zero mean and variance τ

$$p(\beta_i \mid \tau) = N(0, \tau) = \frac{1}{\sqrt{2\pi\tau}} \exp\left(\frac{-\beta_i^2}{2\tau} \right)$$

- The maximum a posteriori (MAP) estimate of β is equivalent to ridge regression for the logistic model.
- Disadvantage of the Gaussian prior in the TC problem,
  - The MAP estimates of the parameters will rarely be exactly zero; thus, the model will not be sparse.
- Laplace prior does achieve sparseness

$$p(\beta_i \mid \lambda) = \frac{\lambda}{2} \exp(-\lambda \mid \beta_i \mid)$$

- The log-posterior distribution of β is

$$l(\beta) = p(\beta \mid D) = -\left(\sum_{(d,c)\in D} \ln(\exp(-c\beta.d) + 1 \right) + \ln\ p(\beta)$$

- where, D= {(d1, c1), (d2, c2) . . .} is the set of training documents d_i and their true category membership values $c_i = \pm 1$, and p(β) is the chosen prior:

- In $p(\beta) = -\left(\sum_i \left(\ln \sqrt{\tau} + \frac{\ln 2\pi}{2} + \frac{\beta_i^2}{\tau} \right) \right)$ for Gaussian prior

- In $p(\beta) = -\left(\sum_i (\ln 2 - \ln \lambda + \lambda \mid \beta_i \mid) \right)$ for Laplace prior

- The MAP estimate of β is arg $\max_\beta l(\beta)$, which can be computed by any convex optimization algorithm

Classifier Committees: Bagging and Boosting

- Committees of classifiers stems from the intuition that a team of experts, by combining their knowledge, may produce better results than a single expert alone.
- Principle: using multiple evidence (multiple poor classifiers => single good classifier)
  - Generate some base classifiers
  - Combine them to make the final decision
- Bagging Method (classifiers are trained in parallel manner)
- Boosting Method (classifiers are trained sequentially)

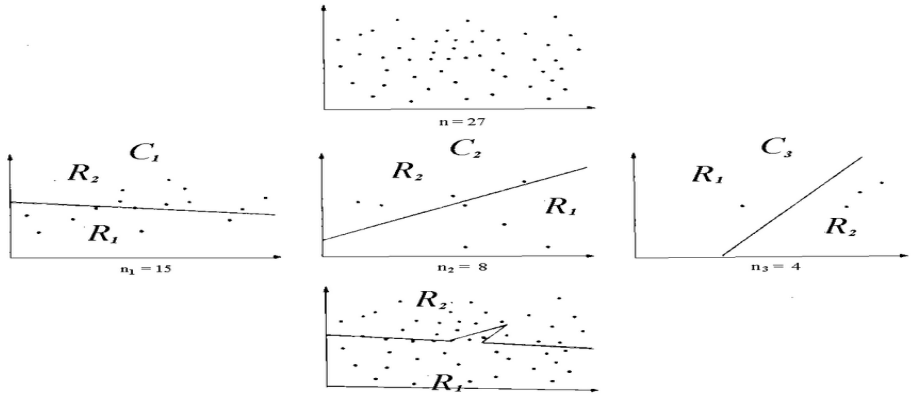
Bagging Algorithm

- Use multiple versions of a training set D of size N, each created by resampling N examples from D with bootstrap
- Each of data sets is used to train a base classifier, the final classification decision is made by the majority voting of these classifiers.

Boosting Algorithm: AdaBoost

Main idea:

- Maintain a distribution or set of weights over the training set.
- Initially, all weights are set equally, but in each iteration the weights of incorrectly classified examples are increased so that the base classifier is forced to focus on the ‘hard’ examples in the training set.
- For those correctly classified examples, their weights are decreased so that they are less important in next iteration.



Why ensembles can improve performance:

- Uncorrelated errors made by the individual classifiers can be removed by voting.
- Hypothesis space *H* may not contain the true function *f*. Instead, *H* may include several equally good approximations to *f*.
- By taking weighted combinations of these approximations, we may be able to represent classifiers that lie outside of *H*.

AdaBoost Algorithm

Given: *m* examples $(x_1, y_1),...,(x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1,+1\}$

Initialize $D_1(i) = 1/m$ for all $i = 1 \ldots m$

For t = 1,…,T:

- Train base classifier using distribution D_t.
- Get a hypothesis $h_t : X \to \{-1,+1\}$ with error $\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i:h_t(x_i) \neq y_i} D_t(i).$
- Choose $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.
- Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Analysis of Voting Algorithms

Advantages:

- Surprisingly effective
- Robust to noise
- Decrease the over-fitting effect

Disadvantages:

- Require more calculation and memory

Using Unlabeled Data to Improve Classification

Two ways of incorporating knowledge from unlabeled documents:

Expectation Maximization (EM)

- EM works with probabilistic generative classifiers. The idea is to find the most probable model given both labeled and unlabeled documents.
- The EM algorithm performs the optimization:
  - First, the model is trained over the labeled documents.
  - Then the following steps are iterated until convergence in a local maximum occurs:
    - E-step: the unlabeled documents are classified by the current model.
    - M-step: the model is trained over the combined corpus.

Cotraining

- The cotraining is a bootstrapping strategy in which the unlabeled documents classified by means of one of the views are then used for training the classifier using the other view, and vice versa.

Performance Measure

Performance of algorithm:

- Training time
- Testing time
- Classification accuracy
  - Precision, Recall
  - F-score
  - Statistical analysis

Goal: High classification quality and computation efficiency

Comparison among Classifiers

- SVM, AdaBoost, k-NN, and Regression are showed good performance
- NB and Rocchio showed relatively poor performance among the ML classifiers, but both are often used as baseline classifiers.
- There are mixed results regarding the Neural Networks and Decision Tree classifiers.