A NEW CLASSIFICATION TECHNIQUE BASED ON HYBRID FUZZY SOFT SET THEORY AND SUPERVISED FUZZY C-MEANS

BANA HANDAGA

**A thesis submitted in
fulfillment of the requirement for the award of the
Doctor of Philosophy**

**Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia**

AUGUST 2013

ABSTRACT

Recent advances in information technology have led to significant changes in today’s world. The generating and collecting data have been increasing rapidly. Popular use of the World Wide Web (www) as a global information system led to a tremendous amount of information, and this can be in the form of text document. This explosive growth has generated an urgent need for new techniques and automated tools that can assist us in transforming the data into more useful information and knowledge. Data mining was born for these requirements. One of the essential processes contained in the data mining is classification, which can be used to classify such text documents and utilize it in many daily useful applications. There are many classification methods, such as *Bayesian*, *K-Nearest Neighbor*, *Rocchio*, SVM classifier, and Soft Set Theory used to classify text document. Although those methods are quite successful, but accuracy and efficiency are still outstanding for text classification problem. This study is to propose a new approach on classification problem based on hybrid fuzzy soft set theory and supervised fuzzy c-means. It is called Hybrid Fuzzy Classifier (HFC). The HFC used the fuzzy soft set as data representation and then using the supervised fuzzy c-mean as classifier. To evaluate the performance of HFC, two well-known datasets are used i.e., 20 Newsgroups and Reuters-21578, and compared it with the performance of classic fuzzy soft set classifiers and classic text classifiers. The results show that the HFC outperforms up to 50.42% better as compared to classic fuzzy soft set classifier and up to 0.50% better as compare classic text classifier.

ABSTRAK

Kemajuan terkini dalam teknologi maklumat telah membawa kepada perubahan penting dalam dunia hari ini. Menjana dan mengumpul data telah meningkat dengan pesat. Penggunaan popular Jaringan Sejagat (www) sebagai sistem maklumat global membawa kepada jumlah maklumat yang sangat banyak, dan ini mungkin adalah dalam bentuk dokumen teks. Ledakan pertumbuhan ini telah menjana keperluan segera bagi teknik-teknik baru dan alatan berautomatik yang boleh membantu kita dalam mentransformasi data kepada maklumat dan pengetahuan yang lebih berguna. Perlombongan data dilahirkan bagi keperluan ini. Salah satu proses penting yang terkandung di dalam perlombongan data adalah klasifikasi, yang boleh digunakan untuk mengklasifikasikan dokumen teks tersebut dan digunakan dalam pelbagai aplikasi kehidupan seharian. Terdapat pelbagai kaedah klasifikasi, seperti *Bayesian*, *K-Nearest Neighbor, Rocchio*, pengkelas SVM, dan Soft Set Theory yang digunakan untuk mengklasifikasikan dokumen teks. Walaupun kaedah tersebut boleh dikira sebagai sukses, tetapi ketepatan dan kecekapan masih belum jelas bagi permasalahan klasifikasi teks. Kajian ini adalah untuk mencadangkan satu pendekatan baru kepada permasalahan klasifikasi berdasarkan hibrid teori set lembut kabur dan c-min berselia kabur. Ia dipanggil Pengkelas Hibrid Kabur (HFC). HFC menggunakan set lembut kabur sebagai perwakilan data dan kemudiannya menggunakan c-mean berselia kabur sebagai pengkelas. Bagi menilai prestasi HFC, dua set data yang diketahui ramai digunakan iaitu, *20 Newsgroup* dan *Reuters-21578*, dan dibandingkan dengan prestasi pengkelas klasik Fuzzy Soft Set dan pengkelas klasik teks. Dapatan menunjukkan bahawa HFC melebihi performa sehingga 50.42% lebih baik berbanding dengan pengkelas Fuzzy Soft Set klasik dan 0.50% lebih baik dibanding pengkelas teks klasik.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACC	Accuracy
DF	Document Frequency
ECOC	Error Correcting Output Coding
FCM	Fuzzy C-Means
FN	False Negative
FP	False Positive
FSSC	Fuzzy Soft Set Classifier
HFC	Hybrid Fuzzy Classifier
IDF	Inverse Document Frequency
KDD	Knowledge Dicovery from Data
k-NN	K Nearest Neighbor
NB	Naïve Bayes
SSC	Soft Set Classifier
SVM	Support Vector Machine
TC	Text Classification
TDM	Term Document Matrix
TF	Term Frequency
TF-IDF	Term Frequency – Inverse Document Frequency
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
WWW	World Wide Web

LIST OF SYMBOLS

U	:	Initial universe.
$P(U)$	:	The power set of U.
$S(U)$	:	The set of all the soft sets over U.
$F(U)$	:	The set of all the fuzzy sets over U.
$FS(U)$	:	The set of all *fs*-sets over U.
$cFS(U)$	:	The set of all cardinal sets of *fs*-sets over U.
E	:	A set of parameters, and $A \subseteq E$.
F_A	:	A soft set.
f_A	:	A soft set approximation function.
μ_X	:	A membership functions of X.
Γ_A	:	A fuzzy soft set.
γ_A	:	A fuzzy approximate functions.
$c\Gamma_A$	:	A cardinal set of fuzzy soft set Γ_A.

CHAPTER 1

INTRODUCTION

1.1 Background

Recent advances in information technology have led to significant changes in today's world. The processes of generating and collecting data have been increasing rapidly. Contributing factors that lead to this include the computerization of business, scientific, and government transactions; the widespread use of digital cameras, publication tools, and bar codes for most commercial products; and advances in data collection tools ranging from scanned text and image platforms to satellite remote sensing systems. In addition, popular use of the World Wide Web (www) as a global information system led to a tremendous amount of information. This explosive growth in stored or transient data has generated an urgent need for new techniques and automated tools that can assist us in transforming the data into more useful information and knowledge (Han & Kamber, 2011).

Data mining was born for these requirements. Data mining refers to extracting or "mining" knowledge from large amounts of data. Many people treat data mining as a synonym for another popularly used term, Knowledge Discovery from Data, or KDD (Han & Kamber, 2011). Fayyad *et al*. (1996) has another view that is KDD refers to the overall process of discovering useful knowledge from data, and data mining refers to a particular step in this process.

In computer science, data mining also called knowledge discovery in databases (KDD) is the process of discovering interesting and useful patterns and relationships in large volumes of data (Britanica, 2013).

In general, data mining tasks can be classified into two categories: descriptive and predictive (Han & Kamber, 2011). Descriptive mining tasks characterize the general properties of the data in the database. While predictive mining tasks perform inference on the current data in order to make predictions. In some cases, users may have no idea regarding what kinds of patterns in their data may be interesting, that could lead to searching for several other kinds of patterns in parallel. As such, it is important to have a system that can mine multiple kinds of patterns to accommodate different user expectations. Data mining functionalities consist of (a) concept or class description, (b) mining frequent patterns, associations, and correlations (c) classification and prediction (d) cluster analysis (e) outlier analysis and (f) evolution analysis.

1.2 Classification and Prediction

A bank officer needs analysis of her data in order to learn which loan applicants are “safe” and which are “risky” for the bank. A manager at computer shop needs data analysis to help guess whether a customer with given profile will buy a new machine. A researcher wants to analyze breast cancer data in order to predict which one of the three specific treatments a patient should receive. In all of these examples, the data analysis task is classification, where a model or classifier is constructed to predict categorical labels, such as “safe” or “risky” for the loan application data, “yes” or “no” label for the marketing data; or “treatment A”, “treatment B”, or “treatment C” for the medical data. These categories can be represented by discrete values, where the ordering among values has no meaning. For example, the value 1, 2, and 3 may be used to represent treatments A, B, and C, where there is no ordering implied among this group of treatment regimes.

Suppose that the marketing manager would like to predict how much a given customer will spend during a sale at computer shop. This data analysis task is an example of numeric prediction, where the model constructed predicts a continuous values function, or ordered value, as opposed to a categorical label. This model is a predictor. Regression analysis is a statistical methodology that is most often used for numeric prediction, hence the two terms are often used synonymously. For simplicity, when there is no ambiguity, we will use the shortened term of prediction to refer to numeric prediction.

The classification is the task of assigning objects to one of several predefined categories, and is one of the essential processes contained in the data mining. There are two forms of data analysis that can be used to extract models, whether describing data classes or to predict future data trends (Fayyad *et al.*, 1996). Databases are rich with hidden information that can be used for intelligent decision making. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Such analysis can help provide us with a better understanding of the data at large. Whereas classification predicts categorical (discrete, unordered) labels, prediction models continuous valued functions.

Basic technique for data classification consist of decision tree classifiers, Bayesian classifiers, Bayesian belief networks, rule-based classifiers, classification based on association rule mining, Back propagation classifier, support vector machine, k-nearest neighbors classifiers, case-based reasoning, genetic algorithms, rough sets, and fuzzy logic techniques. Methods for prediction, including linear regression, non-linear regression, and other regression based models.

This research focused on classification problem, and selects four basic classification techniques to compare with proposed technique, implemented in text classification problem. These four basic text classification techniques are as follows:

(i). Bayesian classifiers (Domingos & Pazzani, 1997; Duda *et al.*, 2000; Langley *et al.*, 1992; Ordonez & Pitchaimalai, 2010; Rish, 2001)

(ii). K-Nearest Neighbor classifiers (Dasarathy, 1991; Duda *et al.*, 2000; S. Jiang *et al.*, 2012; Qiao et al., 2010)

(iii). Rocchio classifier (specific for text classifier) (Miao & Kamel, 2011; Rocchio, 1971)

(iv). Support vector machines (Boser *et al.*, 1992; Cortes & Vapnik, 1995; Joachims, 1998; Pan *et al.*, 2012; Scholkopf *et al.*, 1999; Sullivan & Luke, 2007; Tong & Koller, 2002; Vapnik, 1998; Yu *et al.*, 2003)

Each technique typically suits a problem better than others (Fayyad *et al.*, 1996). Thus, there is no universal data-mining method, and choosing a particular algorithm for a particular application is something of an art. In practice, a large portion of the application effort can go into properly formulating the problem (asking the right question) rather than into optimizing the algorithmic details of a particular data-mining method (Langley & Simon, 1995).

1.3 How does classification work?

Data classification is a two-step process (learning step and classification step). The first step that is the learning step, where a classification algorithm builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels.

A tuples, X, is represented by n-dimensional attribute vector, $X = \{x_1, x_2, \dots, x_n\}$, depicting n measurements made on tuple from n database attributes, respectively, $A_{1,} A_2, \dots, A_n$. Each tuple, X, is assumed to belong to a predefined class as determined by another database attribute called the class label attribute. The class label attribute is discrete valued and unordered. It is categorical in that each value serves as a category or class. The individual tuples making up the training set are referred to as training tuples and are selected from database under analysis. In the context of classification, data tuples can be referred to as samples, examples, instances, data points, or objects.

Because of the class label of each training tuple is provided, this step is also known as **supervised learning**. It contrasts with **unsupervised learning** (or clustering), in which the class label of each training tuple is not known, and the number or set of classes to be learned may not be known in advance.

In the second step, the model is used for classification. A test set is used, made up of test tuples and their associated class labels. These tuples are randomly selected from the general data set. They are independent of the training tuples, meaning that they are not used to construct the classifier. In other word, tuples in the test set must be different from the tuples in the training set.

Classification methods can be compared and evaluated according to the following criteria, (i). Accuracy: The accuracy of a classifier refers to the ability of a given classifier to correctly predict the class label of new or previously unseen data. Similarly, the accuracy of a predictor refers to how well a given predictor can guess the value of the predicted attribute for new or previously unseen data.

(ii). Speed: This refers to the computational costs involved in generating and using the given classifier or predictor.

(iii). Robustness: This is the ability of the classifier or predictor to make correct predictions given noisy data or data with missing values.

(iv). Scalability: This refers to the ability to construct the classifier or predictor efficiently given large amounts of data.

(v). Interpretability: This refers to the level of understanding and insight that is provided by the classifier or predictor. Interpretability is subjective and therefore more difficult to assess

1.4 Problem Statement

In 1999, the concept of soft set theory as a mathematical tool for dealing with uncertainties has initiated by (D. Molodtsov, 1999), which has been further developed by (P. K. Maji *et al.*, 2003). The soft set theory is different from traditional tools for dealing with uncertainties, and further it is free from the inadequacy of the parameterization tools of those theories (D. A. Molodtsov, 2004). The soft set theory has a rich potential for applications in several directions, few of which had been shown by Molodtsov in his pioneer work (D. Molodtsov, 1999).

At present, work on the soft set theory is progressing rapidly both in theoretical models and applications. As for practical applications of soft set theory, great progress has been achieved. The soft set theory can be applied to solve the decision-making problem (F. Feng *et al.*, 2010, 2012; P. K. Maji *et al.*, 2002; Roy & Maji, 2007), parameter reduction (Herawan *et al.*, 2009; Ma et al., 2011), data clustering (Qin, Ma, Zain, *et al.*, 2012), data analysis under incomplete information (Qin, Ma, Herawan, *et al.*, 2012; Zou & Xiao, 2008), the combined forecasting (Xiao *et al.*, 2009), and association rules mining (Herawan & Deris, 2010).

An example of the application of soft set theory for classification is proposed by (Mushrif *et al.*, 2006). They used the soft set theory to classify images texture based on application soft set theory on decision-making problem. A soft set classifier based on similarity measure between the two generalized fuzzy soft sets has reported by (Majumdar & Samanta, 2010). In their work, they provided an example on how the similarity between the two generalized fuzzy soft sets used to detect whether an ill person is suffering from a certain disease.

Although both methods are quite successful for classification, low accuracy and efficiency when applied to text classification is the problem. The writing of this thesis has a purpose to propose a new approach on classification problem based on hybrid fuzzy soft set theory and supervised fuzzy c-means. This new approach is expected to improve the accuracy and the efficiency of classification in text classification problem.

1.5 Research Objectives

The objectives of this research are:

(i). To propose new classification technique based on hybrid fuzzy soft set theory and fuzzy c-means.

(ii). To develop an algorithm based on the proposed technique as in (a).

(iii). Applying the algorithm that develop in (b) on text classification problem.

(iv). To compare the algorithm with the existing algorithm based on efficiency and accuracy performance metrics.

1.6 Contributions

The main contributions of this study are in the area of data mining, the detail of these contributions is as follows:

(i). Extend the area application of soft set theory. The study has introduced a new algorithm for classification based on fuzzy soft set theory.

(ii). Introduce a new algorithm of classification for text classification problem. Applying the proposed algorithm to classify text document that has performance outperform as compare to the previous soft set classifiers and the classic text classifiers, based on efficiency and accuracy performance metrics.

(iii). Introduce a new hybrid algorithm of classification. The proposed algorithm is a hybrid fuzzy algorithm, which is consist of fuzzy soft set theory and supervised fuzzy c-means.

1.7 Research Scope

This study focus on developing the new approach to classify text document based on hybrid fuzzy soft set theory and Fuzzy C-means. Test case will be done using two well-known datasets that are the Reuter-21578 dataset for unevenly distributed dataset, and the 20 Newsgroups for evenly distributed dataset. Comparison will be done on the two groups of classifier. The first group will be used to compare the proposed algorithm with the other two soft set classifiers such as soft set classifier based on decision making-problem and soft set classifier based on similarity between two fuzzy soft sets. The second group will be used to compare the proposed algorithm with the four classic text classifiers, such as k-NN, Rocchio, Bayesian, and Support Vector Machine (SVM).

1.8 Thesis Organization

The thesis is organized into six different chapters. Chapter 1 provides the background and describes what motivated the researcher to introduce the new algorithm for text classification using soft set theory. Chapter 2 will explains the foundations of basic theory of soft set, fuzzy soft set, and text classification. Next, Chapter 3 will describes the new algorithm to classify text document based on fuzzy soft set theory and supervise fuzzy c-means. After that, Chapter 4 will reports the experimental results and discussion, which then tabulate and compare its findings to other research work. Finally, Chapter 5 will conclude and propose future work.

1.9 Chapter Summary

Recent advances in information technology have led to significant changes in today’s world. This explosive growth in stored or transient data has generated an urgent need for new techniques and automated tools that can assist us in transforming the data into more useful information and knowledge. The classification is the task of assigning objects to one of several predefined categories, and is one of the essential processes contained in the data mining. There are two forms of data analysis that can be used to extract models, whether describing data classes or to predict future data trends. Although classic methods are quite successful for classification, low accuracy and efficiency when applied to text classification is the problem. Objective of this research is to propose new classification technique based on hybrid fuzzy soft set theory and fuzzy c-means.

Some important terms related to this study include the following:

(i). **Data mining** is a process to extracting or "mining" knowledge from large amounts of data.

(ii). **Knowledge Discovery from Data** (KDD) is the overall process of discovering useful knowledge from data, and data mining refers to a particular step in this process.

(iii). **Classification** is task of assigning objects to one of several predefined categories, and is one of the essential processes contained in the data mining. There are two models of classification, (a) classification model when the model is used to predict categorical labels, (b) prediction model when the model is used to predict a numerical.

(iv). **Supervised learning** is a learning process when the class label of each training tuple is provided, otherwise is **unsupervised learning**.

(v). **Soft set theory** is as a theory proposed by Molodtsov to deal with uncertainty problem that work with binary features.

(vi). **Fuzzy soft set theory** is a extended version of soft set theory to work with fuzzy number of features.

(vii). **Fuzzy c-means** is a data mining technique to data clustering.

CHAPTER 2

CLASSIFICATION AND SOFT SET THEORY

This chapter describes some basic theories, which will be used as a basis for classification proposed in this research. This includes soft-set theory, classic classification based on soft set theory, fuzzy set theory, fuzzy soft set theory, and fuzzy C-means.

2.1 Introduction

Machine learning, knowledge discovery in databases (KDD) and data mining are three terms that often appear associated with data processing and classification. They have similarities and differences. The similarities between them relate to the two fundamental facts:

(i). All of them develop methods and procedures to process data, and

(ii). Any data processing algorithm or procedure may belong to any.

The differences are in the different perspectives. The difference in perspectives does not affect the procedures but it affects the choice between them in the interpretation of concepts and results (Mirkin, 2011).

Fayyad, *et al*. (1996), the knowledge discovery in databases (KDD) field is the development of methods and techniques for making sense of data. The basic problem addressed by the KDD process is one of mapping low-level data (which are typically too voluminous to understand and digest easily) into other forms that might be more compact (for example, a short report), abstract (for example, a descriptive approximation or model of the process that generated the data), or useful (for example, a predictive model for estimating the value of future cases). At the core of the process is the application of specific data-mining methods for pattern discovery and extraction. The data-mining component of KDD currently relies heavily on known techniques from machine learning, pattern recognition, and statistics to find certain patterns from data.

Knowledge discovery as a process is depicted in Figure 2.1 and consists of an iterative sequence of the following steps (Fayyad *et al*., 1996; Han & Kamber, 2011):

(i). Data cleaning is used to remove noise and inconsistent data.

(ii). Data integration, where multiple data sources may be combined. A popular trend in the information industry is to perform data cleaning and data integration as a preprocessing step, where the resulting data are stored in a data warehouse.

(iii). Data selection, where data relevant to the analysis task are retrieved from the database.

(iv). Data transformation, where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance. Sometimes data transformation and consolidation are performed before the data selection process, particularly in the case of data warehousing. Data reduction may also be performed to obtain a smaller representation of the original data without sacrificing its integrity.

(v). Data mining, an essential process where intelligent methods (such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis) are applied in order to extract data patterns.

(vi). Pattern evaluation, to identify the truly interesting patterns representing knowledge based on some interestingness measures.

(vii). Knowledge presentation, where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

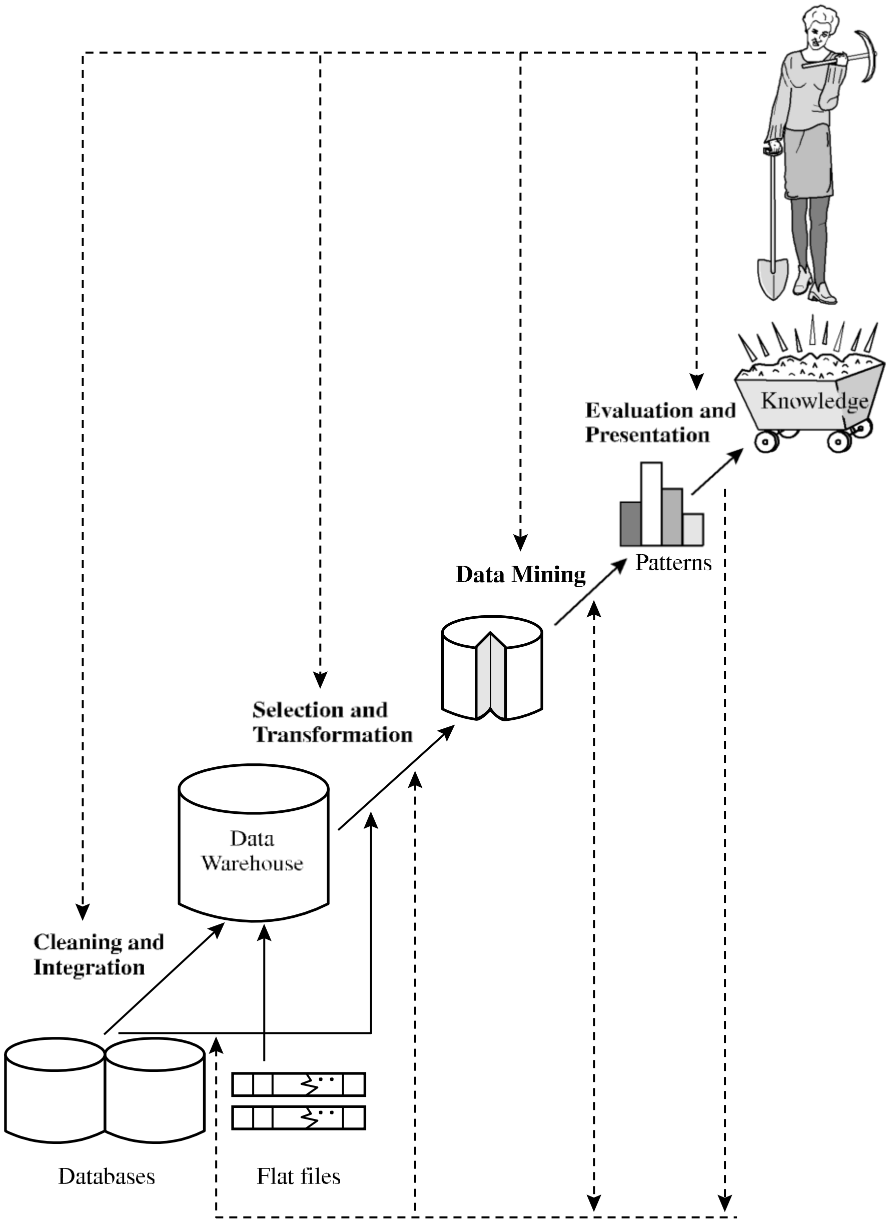


Figure 2.1: Data mining as a step in the process of knowledge discovery.

Steps 1 to 4 are different forms of data preprocessing, where the data are prepared for mining. The data mining step may interact with the user or a knowledge base. The interesting patterns are presented to the user and may be stored as new knowledge in the knowledge base. Note that according to this view, data mining is only one step in the entire process, albeit an essential one because it uncovers hidden patterns for evaluation.

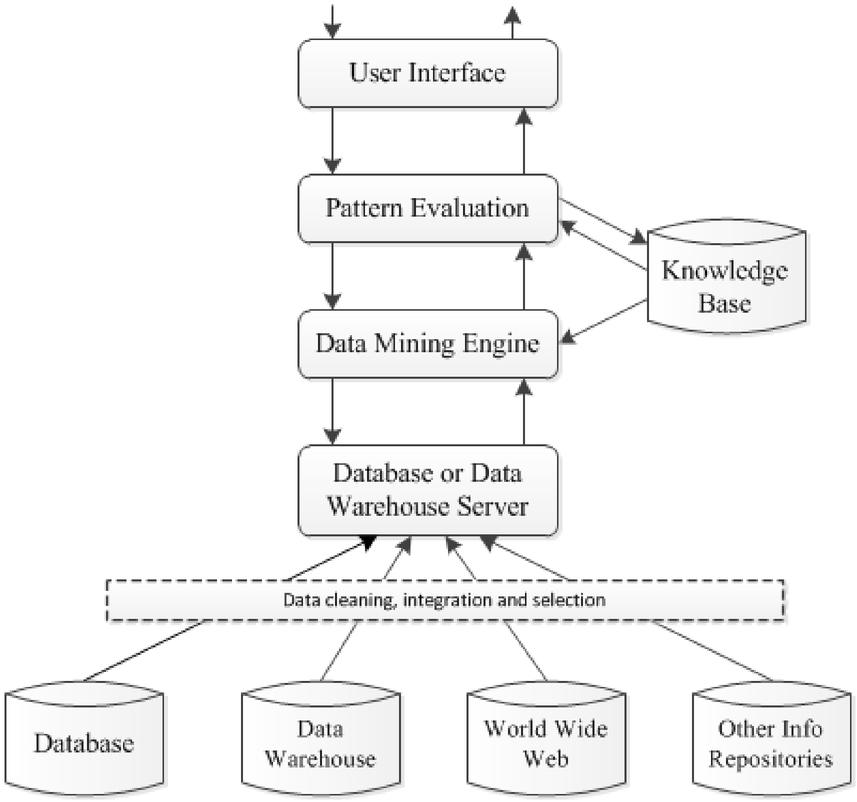


Figure 2.2: Architecture of a typical data mining system.

However, in industry, in media, and in the database research milieu, the term data mining is becoming more popular than the longer term of knowledge discovery from data. Data mining is the process of discovering interesting knowledge from large amounts of data stored in databases, data warehouses, or other information repositories (Han & Kamber, 2011). Based on this view, the architecture of a typical data mining system has the following components (Figure 2.2):

(i). Database, data warehouse, World Wide Web, or other information repository: This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.

(ii). Database or data warehouse server: The database or data warehouse server is responsible for fetching the relevant data, based on the user’s data mining request.

(iii). Knowledge base: This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns. Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction.

(iv). Data mining engine: This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.

(v). Pattern evaluation module: This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.

(vi). User interface: This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

2.2 Datasets, observations, features

Data is the word was originally the plural of datum, which means "a single fact," but it is now used as a collective singular (Douglas *et al.*, 2009). It is originally a Latin noun meaning "something given." Today, data is used in English both as a plural noun meaning "facts or pieces of information" and as a singular mass noun meaning "information." In data classification problems, a data (dataset) is a collection of records (observations). Each observation is usually represented as a vector, $\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n$. The coordinates of vectors $\mathbb{R}^n$ are called features or attributes. Suppose that the dataset contains N observations. In this case, the dataset can be represented as a collection of points $x^i = \left(x_1^i, \cdots, x_n^i\right); i = 1, \cdots, N$ or a data matrix.

The above data matrix represents information about N objects (one observation for each object). Each object is described by n characteristics (in our research we call them features or attributes). For example, if the objects are people the characteristics could be name, age, marital status, original citizenship, occupation etc. Numerical measure of each characteristic for a given object is presented in the corresponding observation at the corresponding attribute. In general, four different types of attributes can be underlined (Stevens, 1946; P. N. Tan *et al*., 2006).

(i). A nominal attribute is an attribute that has two or more categories, but there is no ordering to the categories. For example, gender is a nominal attribute having two categories (male and female). The Marital status is a nominal attribute having five categories (single, married, de facto, divorced, widowed). Each category can be numerically represented. Any order can be used (for example, their appearance in the proposed list of categories, alphabetical orders etc.). We can use this numerical code to refer to the corresponding categorical attribute. In gender attribute, we can use integers 1 and 2 (or 0 and 1) for the numerical code, in marital status attribute we can use integers 1,2,3,4, and 5 or 0,1,2,3, and 4. If a nominal feature has two categories, the term "binary" feature is also used: see gender attribute.

(ii). An ordinal attribute is similar to a nominal one, but for the ordinal attributes, a meaningful order can be arranged. For example, marks for assignments: excellent, good, satisfactory, and failed (1-4 ordered categories) or age groups at hospitals: children, teenagers, adults (1-3 ordered categories). The difference between two values for ordinal attributes is not necessarily meaningful. In the example with age groups, the age difference within the third group ("adults") can be much larger than the difference between representatives from the first category and the second one. This type of attribute can be ordered, but differences between values cannot be quantified.

(iii). An interval-scaled attribute is a real number attribute without base point (zero point). For this type of attributes, order and differences between attribute values are meaningful, but ratios are not. For example, if we measure temperature (°C or °F) we can state that the difference between $20°\mathrm{C}$ and $25°\mathrm{C}$ is the same as between $30°\mathrm{C}$ and $35°\mathrm{C}$, but $15°\mathrm{C}$ is not three times "warmer" than $5°\mathrm{C}$. It happens because for this type of attributes there is no base point (meaningful zero point, natural zero), the point which presents a total absence of the property being measured (the total absence of heat in this example).

(iv). A ratio-scaled attribute is a real number attribute with base point. For this type of attributes order, differences between attribute values and ratios are meaningful. Most physical measurements (weight, length) are ratio-scaled. The Celsius and Kelvin temperature are both interval scaled. Zero on the Kelvin scale means absolute zero, the case in which all motion stops. For the Celsius scale, zero does not have the same meaning (motion is possible). Therefore, Kelvin temperature is also ratio-scaled but Celsius temperature is not.

The nominal and ordinal attributes can be classified as a categorical attribute, while the interval-scaled and ratio-scaled attributes can be classified as a numerical attribute. The numerical attribute can be any integer (discrete), binary (0 or 1), or real numbers (continue). In data matrices nominal, ordinal, and continuous attributes can appear. The data matrix is mixed (different kinds of attributes in one data matrix) or unmixed (only one kind of attributes).

2.3 Structured and Unstructured Data

Data can be designated as structured or unstructured data for classification within an organization. The term structured data refers to data that is identifiable because it is organized in a structure (Webopedia, 2012). The most common form of structured data is a database where specific information is stored based on a methodology of columns and rows. Structured data is also searchable by data type within content. Structured data is understood by computers and is efficiently organized for human readers. In contrast, unstructured data has no identifiable structure.

The term unstructured data refers to any data, that has no identifiable structure. For example, images, videos, email, documents, and text are all considered to be unstructured data within a dataset. While each individual document may contain its own specific structure or formatting that based on the software program used to create the data, unstructured data may also be considered “loosely structured data” because the data sources do have a structure but all data within a dataset, will not contain the same structure.

The studies of data mining have focused on structured data. However, in reality, a substantial portion of the available information is stored in text databases (or document databases), which is unstructured data and consist of large collections of documents from various sources, such as news articles, research papers, books, digital libraries, e-mail messages, and Web pages. Text databases are rapidly growing due to the increasing amount of information available in electronic form, such as electronic publications, various kinds of electronic documents, e-mail, and the World Wide Web (which can also be viewed as a huge, interconnected, dynamic text database). Nowadays most of the information in government, industry, business, and other institutions are stored electronically, in the form of text databases.

Some text databases are semi-structured data, in that they are neither completely unstructured nor completely structured. For example, a document may contain a few structured fields, such as title, authors, publication date, and category, but also contain some largely unstructured text components, such as abstract and contents. Text mining is a study of data mining that focused on text databases, whether on unstructured or semi-structured data (Han & Kamber, 2011). One important activity in text mining is the classification of text documents (text classification).

So far, we have explained the stages of data processing in data mining, and the kind of data that can be mine based on structure and type of data. In this study, we will use data originate from unstructured data i.e., textual data, and then transforms into structured data with attribute only consist of continue number. In the following section, we will explain the most important method in data mining. It is call classification.

2.4 Classification

Classification is the task of learning a target function *f* that maps each attribute *x* to one of the predefined class labels *y* (P. N. Tan *et al*., 2006). Given a collection of records (training set), each record contains a set of attributes, one of the attributes is the class. Find a model for class attribute as a function of the values of other attributes. Goal, previously unseen records should be assigned a class as accurately as possible. A test set is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, which training set used to build the model and test set used to validate it.

Table 2-1 The vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
Human	Warm-blooded	Hair	Yes	No	No	Yes	No	mammal
Python	Cold-blooded	Scales	No	No	No	No	Yes	Reptile
Salmon	Cold-blooded	Scales	No	Yes	No	No	No	Fish
Whale	Warm-blooded	Hair	Yes	Yes	No	No	No	Mammal
Frog	Cold-blooded	None	No	Yes	No	Yes	Yes	Amphibian
Komodo	Cold-blooded	Scales	No	No	No	Yes	No	Reptile
Bat	Warm-blooded	Hair	Yes	No	Yes	Yes	Yes	Mammal
Pigeon	Warm-blooded	Feathers	No	No	Yes	Yes	No	Birrrd
Cat	Warm-blooded	Fur	Yes	No	No	Yes	No	Mammal
Leopard	Cold-blooded	Scales	Yes	Yes	No	Yes	No	Fish
Turtle	Cold-blooded	Scales	No	Yes	No	Yes	No	Reptile
Penguin	Warm-blooded	Feathers	No	Yes	No	Yes	No	Bird
Porcupine	Warm-blooded	Quills	Yes	No	No	Yes	Yes	Mammal
Eel	Cold-blooded	Scales	No	Yes	No	No	No	Fish
Salamander	Cold-blooded	None	No	Yes	No	Yes	Yes	amphibian

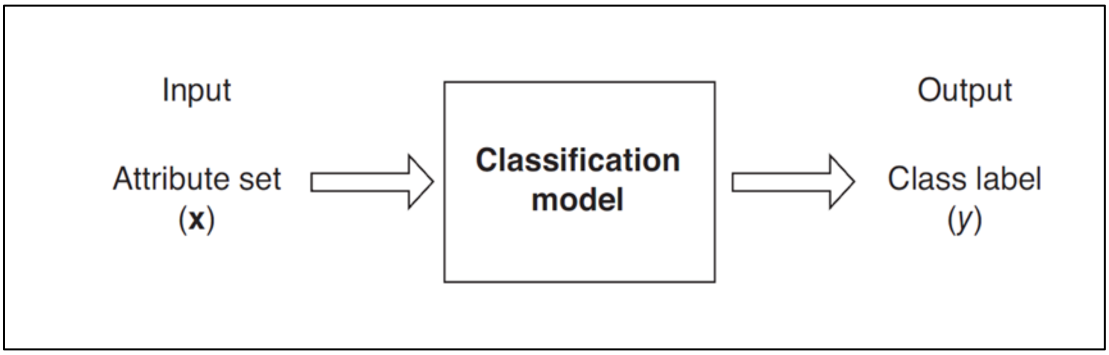


Figure 2.3 Classification as the task of mapping an input attribute set *x* into its class label *y*.

The input data for classification task is a collection of records. Each record, also known as an instance or example, is characterized by a tuple (*x*, *y*), where *x* is the attribute set, and *y* is a special attribute, designated as the class label (also known as category or target attribute). Table 2-1 shows a sample data set used for classifying vertebrates into one of the following categories: mammal, bird, fish, reptile, or amphibian. The attribute set includes properties of a vertebrate such as its body temperature, skin cover, method of reproduction, ability to fly, and ability to live in water. Although the attributes presented in Table 2-1 are mostly discrete, the attribute set can also contain continuous features. The class label, on the other hand, must be a discrete attribute. This is a key characteristic that distinguishes classification from regression, a predictive modeling task in which *y* is a continuous attribute.

The target function also known informally as a classification model. A classification model is useful for the following purpose.

(i). Descriptive modeling, a classification model can serve as an explanatory tool to distinguish between objects of different classes. For example, it would be useful, for biologists and others, to have a descriptive model that summarizes the data shown in Table 2-1 and explains what features define a vertebrate as a mammal, reptile, bird, fish, or amphibian.

(ii). Predictive modeling, a classification model can also be used to predict the class label of unknown records. As shown in Figure 2.3, a classification model can be treated as a black box that automatically assigns a class label when presented with the attribute set of an unknown record. Supposed we are given the following characteristics of a creature known as a Gila monster (Table 2-2):

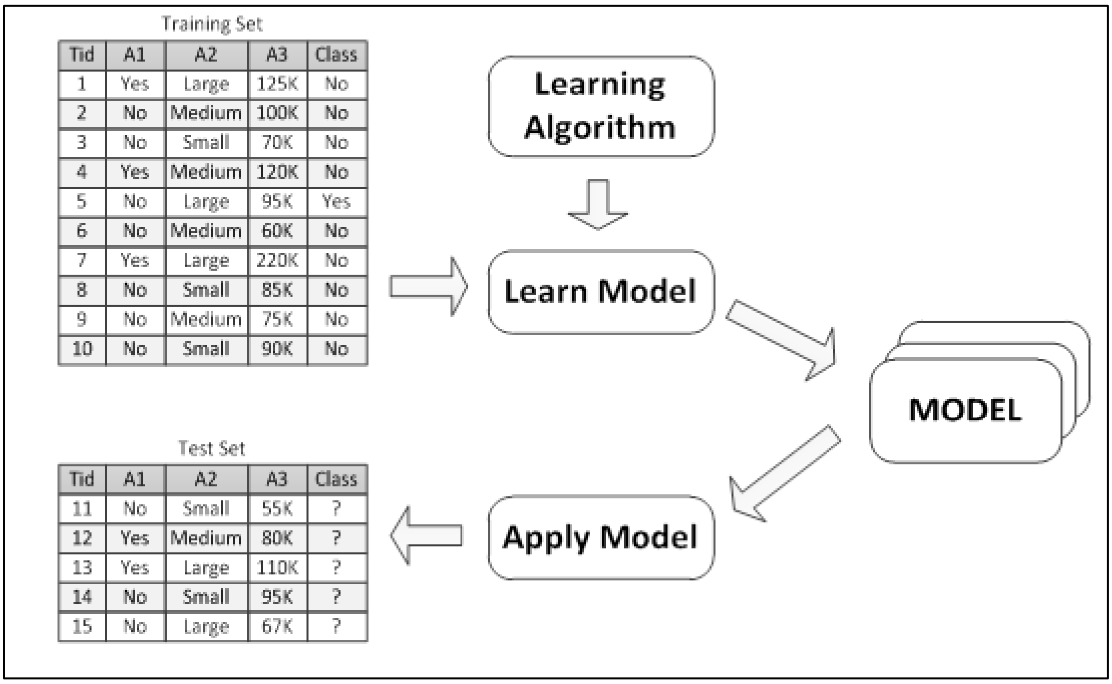
Table 2-2 The unseen vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class Label
Gila monster	Cold-blooded	Scales	No	No	No	Yes	Yes	?

Using Table 2-1 we determine the class to which the creature belongs.

P. N. Tan et. al. (2006) state that classification techniques are most suited for predicting or describing data sets with binary or nominal categories (multiclass). They are less effective for ordinal categories (e.g. to classify a person as a member of high-, medium-, or low-income group) because they do not consider the implicit order among the categories. Other forms of relationships, such as the subclass-superclass relationship among categories (e.g. humans and apes are primates, which in turn, is a subclass of mammals) are also ignored.

A classifier is a systematic approach to building classification models from an input data set. Examples include k-Nearest Neighbor, Bayesian, and Support Vector Machine classifier. Each technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before. Therefore, a key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records.



Training Set

Tid	A1	A2	A3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	No
9	No	Medium	75K	No
10	No	Small	90K	No

Test Set

Tid	A1	A2	A3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Figure 2.4 General approach for building a classification model

Figure 2.4 shows a general approach for solving classification problems. Firstly, a training set consisting of records whose class labels are known must be provided. Secondly, the training set used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels.

Based on the data type of class label, there are three cases of classification, i.e. binary, multi-class, and multi-label classification case. In the following section, we will discuss about those three cases of classification.

2.4.1 Binary Classification

The single-label classification is concerned with learning from a set of examples that are associated with a single label from a set of disjoint labels L, |L| > 1. If |L|=2, then the learning problem is called a binary classification problem, while if |L| > 2, then it is called a multiclass classification (multinomial classification) problem (Tsoumakas

& Katakis, 2009). The vertebrate data set, Table 2-1, is an example for multiclass label; because the class label column has a value of more than two classes.

2.4.2 Multi-class Classification

The multi-class classification problem can be decomposed into several binary classification tasks using binary classifiers (Aly, 2005). The idea is similar to that of using ‘code words’ for each class and then using number binary classifiers in solving several binary classification problems, whose results can determine the class label for new data. Several methods have been proposed for such decomposition such as one-against-all, all-against-all, and Error-Correcting Output-Coding (ECOC).

2.4.2.1 One-Against-All

The simplest approach is to reduce the problem of classifying among L classes into L binary problems (Allwein *et al.*, 2001; Rifkin & Klautau, 2004; Vapnik, 1998; Yukinawa *et al.*, 2009), where each problem discriminates a given class from the other L−1 classes. For this approach, we require $N = L$ binary classifiers, where the 1^{th} classifier is trained with positive examples belonging to class l and negative examples belonging to the other $L - 1$ classes. When testing an unknown example, the classifier producing the maximum output is considered the winner, and this class label is assigned to that example. Rifkin and Klautau (2004) state, that this approach, although simple, provides performance that is comparable to other more complicated approaches when the binary classifier is tuned well.

2.4.2.2 All-Against-All

In this approach, each class is compared to each other class (Friedman, 1996; Yukinawa *et al.*, 2009). A binary classifier is built to discriminate between each pair of classes, while discarding the rest of the classes. This requires building L (L−1)/2 binary classifiers. When testing a new example, a voting is performed among the classifiers and the class with the maximum number of votes wins. Results (Allwein *et al*., 2001; Hsu & Lin, 2002) show that this approach is in general better than the one-versus-all approach.

Table 2-3 ECOC example.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7
Class 1	0	0	0	0	0	0	0
Class 2	0	1	1	0	0	1	1
Class 3	0	1	1	1	1	0	0
Class 4	1	0	1	1	0	1	0
Class 5	1	1	0	1	0	0	1

2.4.2.3 Error-Correcting Output-Coding (ECOC)

This approach works by training *N* binary classifiers to distinguish between the *L* different classes. Each class is given a codeword of length *N* according to a binary matrix *M*. Each row of *M* corresponds to a certain class (Dietterich & Bakiri, 1995; Yukinawa *et al*., 2009). Table 2-3 shows an example for *K* = 5 classes and *N* = 7 bit codewords. Each class is given a row of the matrix. Each column is used to train a distinct binary classifier. When testing an unseen example, the output codeword from the *N* classifiers is compared to the given *K* codewords, and the one with the minimum hamming distance is considered the class label for that example.

So far, in Section 2.4.2, we had discussed one case of classification where the class label consists of more than two labels, and for each example, they can only have one class label, it is called multi-class classification. Following section is the last case in classification problems, where class label consists of more than two labels and each example may have more than one class labels. As Rifkin and Klautau (2004) state, that this approach, although simple, provides performance that is comparable to other more complicated approaches when the binary classifier is tuned well.

2.4.3 Multi-label Classification

In contrast to multi-class classification, alternatives in multi-label classification are not assumed to be mutually exclusive: multiple labels may be associated with a single example, in other words, each example can be a member of more than one class. In multi-label classification, the examples are associated with a set of labels $Y \subseteq L$, labels in the set *Y* are called relevant, while the labels in the set *L\Y* are irrelevant for a given example (Katakis *et al*., 2008; Madjarov *et al*., 2012; Tsoumakas & Katakis, 2009; Tsoumakas *et al*., 2010). Text documents usually belong to more than one conceptual class. Similarly, in medical diagnosis, a patient may be suffering, for example, from diabetes and prostate cancer at the same time.

Multi-label learning introduces the concept of multi-label ranking (Tsoumakas *et al*., 2010). Multi-label ranking can be considered as a generalization of multi-class classification, where instead of predicting only a single label (the top label); it predicts the ranking of all labels. In other words, multi-label ranking is understood as learning a model that associates a query example x both with a ranking of the complete label set and a bipartition of this set in to relevant and irrelevant labels.

Nowadays, many different approaches have been developed to solving multi-label learning problems. Tsoumakas and Katakis (2010) summarize them in to two main categories:

(i). Algorithm adaptation methods and

(ii). Problem transformation methods (Tsoumakas *et al*., 2010).

The first of methods extend specific learning algorithms in order to handle multi-label data directly. The second methods are algorithm independent. They transform the learning task into one or more single-label classification tasks, for which a large bibliography of learning algorithms exists.

Problem transformation methods can be grouped in to three categories: binary relevance, label power-set and pair-wise methods (Madjarov *et al*., 2012).

(i). Binary relevance methods: The simplest strategy for problem transformation is to use the one-against-all strategy to convert the multi-label problem in to several binary classification problems.

(ii). Label power-set methods: A second problem transformation method is the label combination method, or label power-set method (LP). The basis of these methods is to combine entire label sets in to atomic (single) labels to form a single-label problem (single-class classification problem). For the single-label problem, the set of possible single labels represents all distinct label subsets from the original multi-label representation.

(iii). Pair-wise methods: A third problem transformation approach to solving the multi-label learning problem is pair-wise or round robin classification with binary classifiers. The basic idea here is to use *Q* (*Q*-1)/2 classifiers covering all pairs of labels. Each classifier is trained using the samples of the first label as positive examples and the samples of the second label as negative examples. To combine these classifiers, the pairwise classification method naturally adopts the majority-voting algorithm. Given a test example, each classifier predicts one of the two labels. After the evaluation of all *Q* (Q-1)/2 classifiers, the labels are ordered according to their sum of votes. A label-ranking algorithm is used to predict the relevant labels for each example.

In this study, we use two kinds of dataset in the text classification problems; both of them have more than two labels in the column of class label. First dataset consists of documents that belong to only one class label (multi-class) and evenly distribute, and second dataset consists of documents that may belong to more than one class label (multi-label) and unevenly distribute. For both of datasets we will use one-against-all strategy to classify each document in the dataset.

2.5 The basic techniques of classification

In this section we will discuss all these basic classification techniques shortly, adopt from (Han & Kamber, 2011), consist of decision tree classifiers, Bayesian classifiers, rule-based classifiers, classification based on association rule mining, back propagation classifier, support vector machines, k-nearest-neighbors classifiers, genetic algorithms, rough sets, and fuzzy logic techniques.

Four basic techniques of classification have selected to compare with proposed technique, implemented in text classification problem. These four basic text classification techniques are as follows, Bayesian, support vector machines, k-nearest-neighbor, and Rocchio classifier.

2.6 Decision tree classifier

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree as a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node holds a class label. The topmost node in a tree is the root node. A typical decision tree is shown in Figure 2.5. It represents the concept buys computer, that is, it predicts whether a costumer at computer shop is likely to purchase a computer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals. Some decision tree algorithms produce only binary trees, whereas others can produced non binary trees.

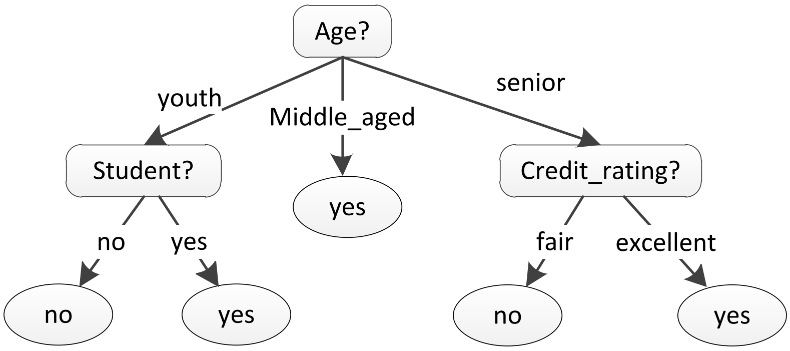


Figure 2.5. A decision tree for the concept buys computer

Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision tree can easily be converted to classification rules.

The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans. The learning and classification steps of decision tree induction are simple and fast. In general, decision tree classifiers have good accuracy. However, successful use may depend on the data at hand. Decision tree induction algorithms have been used for classification in many application areas, such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology. Decision tree are the basis of several commercial rule induction systems.

REFERENCES

Airoldi, E. M., Cohen, W. W., & Feinberg, S. E. (2004). Bayesian models for frequent terms in text. *In Proceedings of the CSNA & INTERFACE Annual Meetings*.

Aktas, H., & Cagman, N. (2007). Soft sets and soft groups. *Information Sciences*, *177*(13), 2726–2735.

Ali, M. I., Feng, F., Liu, X., Min, W. K., & Shabir, M. (2009). On some new operations in soft set theory. *Computers and Mathematics with Applications*, *57*(9), 1547–1553. Retrieved from http://www.sciencedirect.com/science/article/pii/S0898122108006871

Ali, S., & Smith, K. A. (2006). On learning algorithm selection for classification. *Applied Soft Computing*, *6*(2), 119–138. Amsterdam, The Netherlands, The Netherlands: Elsevier Science Publishers B. V.

Allwein, E. L., Schapire, R. E., & Singer, Y. (2001). Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, *1*, 113–141. JMLR.org. Retrieved from http://dx.doi.org/10.1162/15324430152733133

Aly, M. (2005). *Survey on Multiclass Classification Methods*. Technical report, California Institute of Technology, California, USA.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. (Be. Baeza-Yates, R A and Ribeiro-Neto, Ed.) (p. 513). New York: Addison Wesley. Retrieved from http://web.simmons.edu/~benoit/LIS466/Baeza-Yateschap01.pdf

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, *10*(2-3), 191–203. Retrieved from http://www.sciencedirect.com/science/article/pii/0098300484900207

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144–152). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/130385.130401

Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Jackel, L. D., LeCun, Y., *et al*. (1994). Comparison of classifier methods: a case study in handwritten digit recognition. *Pattern Recognition, 1994. Vol. 2 - Conference B: Computer*

Vision Image Processing., Proceedings of the 12th IAPR International. Conference on (Vol. 2, pp. 77 –82 vol.2).

Britanica, C. (2013). Data mining. *Encyclopedia Britannica Online*. Retrieved January 10, 2013, from http://global.britannica.com/EBchecked/topic/1056150/data-mining

Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining Knowledge Discovery*, *2*(2), 121–167. Hingham, MA, USA: Kluwer Academic Publishers. Retrieved from http://dx.doi.org/10.1023/A:1009715923555

Cagman, N, Enginoglu, S., & Citak, F. (2011). Fuzzy Soft Set Theory and Its Applications. *Iranian Journal of Fuzzy Systems*, *8*(3), 137–147. Retrieved from http://journals.usb.ac.ir/Fuzzy/en-us/JournalNumbers/Articles58/

Cagman, Naim, & Enginoglu, S. (2010a). Soft set theory and uni-int decision making. *European Journal of Operational Research*, *207*(2), 848–855. Retrieved from http://www.sciencedirect.com/science/article/pii/S0377221710003589

Cagman, Naim, & Enginoglu, S. (2010b). Soft matrix theory and its decision making. *Computers and Mathematics with Applications*, *59*(10), 3308–3314. Retrieved from http://www.sciencedirect.com/science/article/pii/S0898122110001914

De Campos, L. M., & Romero, A. E. (2009). Bayesian network models for hierarchical text classification from a thesaurus. *International Journal of Approximate Reasoning*, *50*(7), 932–944. Retrieved from http://www.sciencedirect.com/science/article/pii/S0888613X08001746

Chen, D., Tsang, E. C. C., Yeung, D. S., & Wang, X. (2005). The parameterization reduction of soft sets and its applications. *Computers and Mathematics with Applications*, *49*(5-6), 757–763. Tarrytown, NY, USA: Pergamon Press, Inc.

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naive Bayes. *Expert Systems with Applications*, *36*(3, Part 1), 5432–5435. Retrieved from http://www.sciencedirect.com/science/article/pii/S0957417408003564

Connor, M., & Kumar, P. (2010). Fast Construction of k-Nearest Neighbor Graphs for Point Clouds. *Visualization and Computer Graphics, IEEE Transactions on*, *16*(4), 599–608.

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. Hingham, MA, USA: Kluwer Academic Publishers. Retrieved from http://dx.doi.org/10.1023/A:1022627411411

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, *13*(1), 21–27.

Cunningham, P., & Delany, S. J. (2007). *k-Nearest Neighbour Classifiers*. University College Dublin and Dublin Institute of Technology.

Dasarathy, B. V. (1991). *Nearest neighbor (NN) norms: nn pattern classification techniques*. (B. V Dasarathy, Ed.). IEEE Computer Society Press.

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res.*, *2*(1), 263–286. USA: AI Access Foundation. Retrieved from http://dl.acm.org/citation.cfm?id=1622826.1622834

Domingos, P., & Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, *29*(2-3), 103–130. Hingham, MA, USA: Kluwer Academic Publishers. Retrieved from http://dx.doi.org/10.1023/A:1007413511361

Douglas, D. A., Covington, M. A., Covington, M. M., & Covington, C. A. (2009). *Dictionary of Computer and Internet Terms, Tenth Edition* (10th ed.). New York: Barron's Educational Series, Inc.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience.

Dumais, S. T., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In G. Gardarin, J. C. French, N. Pissinou, K. Makki, & L. Bouganim (Eds.), *Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management* (pp. 148–155). Bethesda, US: ACM Press, New York, US. Retrieved from http://robotics.stanford.edu/users/sahami/papers-dir/cikm98.pdf

Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, *3*, 32–57.

Eyheramendy, S., Lewis, D. D., & Madigan, D. (2003). On the Naive Bayes Model for Text Categorization. *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*.

Fayyad, U., Piatetsky-shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, *17*(3), 37–54. American Association for Artificial Intelligence. Retrieved from http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1230

Feng, F., Jun, Y. B., Liu, X., & Li, L. (2010). An adjustable approach to fuzzy soft set based decision making. *Journal of Computational and Applied Mathematics*, *234*(1), 10–20.

Feng, F., Li, Y., & Cagman, N. (2012). Generalized uni-int decision making schemes based on choice value soft sets. *European Journal of Operational Research*, *220*(1), 162–170. Retrieved from http://www.sciencedirect.com/science/article/pii/S0377221712000355

Feng, G., Guo, J., Jing, B.-Y., & Hao, L. (2012). A Bayesian feature selection paradigm for text classification. *Information Processing and Management*, *48*(2), 283–302. Retrieved from http://www.sciencedirect.com/science/article/pii/S0306457311000811

Friedman, J. H. (1996). *Another approach to polychotomous classification*. Retrieved from http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z

Garcia, V., Debreuve, E., & Barlaud, M. (2008). Fast k nearest neighbor search using GPU. *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW ’08. IEEE Computer Society Conference on* (pp. 1–6).

Han, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Hechenbichler, K., & Schliep, K. (2006). Weighted k-nearest-neighbor techniques and ordinal classification. *Discussion Paper 399, SFB 386*.

Herawan, T., & Deris, M. M. (2010). A soft set approach for association rules mining. *Knowledge-Based Systems, In Press*, , -.

Herawan, T., Rose, A. N. M., & Mat Deris, M. (2009). Soft Set Theoretic Approach for Dimensionality Reduction. In D. Slezak, T. Kim, Y. Zhang, J. Ma, & K. Chung (Eds.), *Database Theory and Application* (Vol. 64, pp. 171–178). Springer Berlin Heidelberg.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, *13*(2), 415–425.

Hu, R. (2011). *Active Learning for Text Classification*. Dublin Institute of Technology.

Jiang, J.-Y. (2011). *Feature Reduction and Multi-label Classification Approaches for Document Data*. National Sun Yat-sen University.

Jiang, S., Pang, G., Wu, M., & Kuang, L. (2012). An improved K-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, *39*(1), 1503–1509. Retrieved September 26, 2012, from http://www.sciencedirect.com/science/article/pii/S0957417411011511

Joachims, T. (1998). Text Categorization with Support Vector Machine. *Proceedings of the European Conference on Machine Learning*. Springer-Verlag.

Katakis, I., Tsoumakas, G., & Vlahavas, I. (2008). Multilabel Text Classification for Automated Tag Suggestion. *Proceedings of the ECML/PKDD 2008 Discovery Challenge*.

Kim, S.-B., Han, K.-S., Rim, H.-C., & Myaeng, S. H. (2006). Some Effective Techniques for Naive Bayes Text Classification. *Knowledge and Data Engineering, IEEE Transactions on*, *18*(11), 1457–1466.

Koller, D., & Sahami, M. (1997). Hierarchically Classifying Documents Using Very Few Words. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 170–178). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=645526.657130

Kong, Z., Gao, L., & Wang, L. (2009). Comment on “A fuzzy soft set theoretic approach to decision making problems”. *Journal of Computational and Applied Mathematics*, *223*(2), 540–542. Retrieved from http://www.sciencedirect.com/science/article/pii/S0377042708000162

Kong, Z., Gao, L., Wang, L., & Li, S. (2008). The normal parameter reduction of soft sets and its algorithm. *Computers and Mathematics with Applications*, *56*(12), 3029–3037. Retrieved from http://www.sciencedirect.com/science/article/pii/S0898122108004483

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. *Proceedings of the tenth national conference on Artificial intelligence* (pp. 223–228). San Jose, California: AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=1867135.1867170

Langley, P., & Simon, H. A. (1995). Applications of machine learning and rule induction. *Communication ACM*, *38*(11), 54–64. New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/219717.219768

Larkey, L. S., & Croft, W. B. (1996). Combining classifiers in text categorization. *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 289–297). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/243199.243276

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Nédellec & C. Rouveirol (Eds.), *Proceedings of ECML-98, 10th European Conference on Machine Learning* (pp. 4–15). Chemnitz, DE: Springer Verlag, Heidelberg, DE. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.8397

Li, N., & Wu, D. D. (2010). Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decision Support Systems*, *48*(2), 354–368. Retrieved from http://www.sciencedirect.com/science/article/pii/S0167923609002097

Lopes, C., Cortez, P., Sousa, P., Rocha, M., & Rio, M. (2011). Symbiotic filtering for spam email detection. *Expert Systems with Applications*, *38*(8), 9365–9372. Retrieved from http://www.sciencedirect.com/science/article/pii/S0957417411003228

Ma, X., Sulaiman, N., Qin, H., Herawan, T., & Zain, J. M. (2011). A new efficient normal parameter reduction algorithm of soft sets. *Computers and Mathematics with Applications*, *62*(2), 588–598. Retrieved from http://www.sciencedirect.com/science/article/pii/S0898122111004366

Madjarov, G., Gjorgjevikj, D., & Deroski, S. (2012). Two stage architecture for multi-label learning. *Pattern Recognition*, *45*(3), 1019–1034. Retrieved from http://www.sciencedirect.com/science/article/pii/S0031320311003487

Maji, P., Biswas, R., & Roy, A. (2001). Fuzzy soft sets. *Journal of Fuzzy Mathematics*, *9(3),* 589–602.

Maji, P. K., Biswas, R., & Roy, A. R. (2003). Soft set theory. *Computers and Mathematics with Applications*, *45*(4-5), 555–562. Retrieved from http://www.sciencedirect.com/science/article/pii/S0898122103000166

Maji, P. K., Roy, A. R., & Biswas, R. (2002). An application of soft sets in a decision making problem. *Computers and Mathematics with Applications*, *44*(8-9), 1077–1083. Retrieved from http://www.sciencedirect.com/science/article/pii/S089812210200216X

Majumdar, P., & Samanta, S. K. (2008). Similarity Measure Of Soft Sets. *New Mathematics and Natural Computation (NMNC)*, *4*(01), 1–12.

Majumdar, P., & Samanta, S. K. (2010). Generalised fuzzy soft sets. *Computers & Mathematics with Applications*, *59*(4), 1425–1432. Tarrytown, NY, USA: Pergamon Press, Inc.

Majumdar, P., & Samanta, S. K. (2011). On Similarity Measures of Fuzzy Soft Sets. *International Journal Advance Soft Compututing Application*, *3*(2). Retrieved from http://www.i-csrs.org/Volumes/ijasca/vol.3/vol.3.2.4.July.11.pdf

Manning, C. D., Raghavan, P., & Schutze, H. (2009). *An Introduction to Information Retrieval (on line edition)*. Cambridge University Press, Cambridge, England.

McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. *Workshop on Learning for Text Categorization* (pp. 41–48). AAAI Press.

Miao, Y., & Kamel, M. (2011). Pairwise optimized Rocchio algorithm for text categorization. *Pattern Recognition Letters, 32*(2), 375–382. Retrieved September 26, 2012, from http://www.sciencedirect.com/science/article/pii/S0167865510003223

Mirkin, B. (2011). Data analysis, mathematical statistics, machine learning, data mining: Similarities and differences. *Advanced Computer Science and Information System (ICACSIS), 2011 International Conference on* (pp. 1–8).

Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill.

Molodtsov, D. (1999). Soft set theory: First results. *Computers and Mathematics with Applications, 37*(4-5), 19–31. Retrieved from http://www.sciencedirect.com/science/article/pii/S0898122199000565

Molodtsov, D. A. (2004). *The theory of soft sets*. Moscow: URSS Publishers.

Morelos-Zaragoza, R. H. (2006). *The art of error correcting coding*. Wiley Interscience.

Mushrif, M., Sengupta, S., & Ray, A. (2006). Texture Classification Using a Novel, Soft-Set Theory Based Classification Algorithm. In P. Narayanan, S. Nayar, & H.-Y. Shum (Eds.), *Computer Vision - ACCV 2006* (Vol. 3851, pp. 246–254). Springer Berlin / Heidelberg.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Mach. Learn.*, *39*(2-3), 103–134. Hingham, MA, USA: Kluwer Academic Publishers. Retrieved from http://dx.doi.org/10.1023/A:1007692713085

Nogueira, T. M., Rezende, S. O., & Camargo, H. A. (2010). On the use of fuzzy rules to text document classification. *Hybrid Intelligent Systems (HIS), 2010 10th International Conference on* (pp. 19–24).

Olson, D. L., & Delen, D. (2008). *Advanced Data Mining Techniques* (p. 180). Springer-Verlag Berlin Heidelberg.

Ordonez, C., & Pitchaimalai, S. K. (2010). Bayesian Classifiers Programmed in SQL. *Knowledge and Data Engineering, IEEE Transactions on*, *22*(1), 139–144.

Pan, S., Iplikci, S., Warwick, K., & Aziz, T. Z. (2012). Parkinson's Disease tremor classification: A comparison between Support Vector Machines and neural networks. *Expert Systems with Applications*, *39*(12), 10764–10771. Retrieved from http://www.sciencedirect.com/science/article/pii/S0957417412004629

Pawlak, Z. (1982). Rough sets. *International Journal of Parallel Programming*, *11*(5), 341–356. Springer Netherlands. Retrieved from http://dx.doi.org/10.1007/BF01001956

Pei, D., & Miao, D. (2005). From soft sets to information systems. *Granular Computing, 2005 IEEE International Conference on* (Vol. 2, pp. 617 – 621 Vol. 2).

Perez-Diaz, N., Ruano-Ordas, D., Mendez, J. R., Galvez, J. F., & Fdez-Riverola, F. (2012). Rough sets for spam filtering: Selecting appropriate decision rules for boundary e-mail classification. *Applied Soft Computing*, *12*(11), 3671–3682. Retrieved from http://www.sciencedirect.com/science/article/pii/S1568494612002748

Porter, M. F. (1997). Readings in information retrieval. In K. Sparck Jones & P. Willett (Eds.), (pp. 313–316). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=275537.275705

Porter, M. F. (2001). Snowball: A language for stemming algorithms. Retrieved from http://snowball.tartarus.org/texts/introduction.html

Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, *3*(2), 143–157. Retrieved from http://www.sciencedirect.com/science/article/pii/S1751157709000108

Qiao, Y.-L., Lu, Z.-M., Pan, J.-S., & Sun, S.-H. (2010). Fast k-nearest neighbor search algorithm based on pyramid structure of wavelet transform and its application to texture classification. *Digital Signal Processing*, *20*(3), 837–845. Retrieved from http://www.sciencedirect.com/science/article/pii/S1051200409001894

Qin, H., Ma, X., Herawan, T., & Zain, J. M. (2012). DFIS: A novel data filling approach for an incomplete soft set. *International Journal of Applied Mathematics and Computer Science*, *22*(4), 817–828. Retrieved from http://www.amcs.uz.zgora.pl/?action=paper&paper=651

Qin, H., Ma, X., Zain, J. M., & Herawan, T. (2012). A novel soft set approach in selecting clustering attribute. *Knowledge-Based Systems*, (0), -. Retrieved from http://www.sciencedirect.com/science/article/pii/S0950705112001712

Raghavan, V. V, & Wong, S. K. M. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, *37*(5), 279–287.

Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. R. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *In Proceedings of the Twentieth International Conference on Machine Learning* (pp. 616–623).

Rifkin, R., & Klautau, A. (2004). In Defense of One-Vs-All Classification. *J. Mach. Learn. Res.*, *5*, 101–141. JMLR.org. Retrieved from http://dl.acm.org/citation.cfm?id=1005332.1005336

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI-01 workshop on "Empirical Methods in AI"*. Retrieved from http://www.intellektik.informatik.tu-darmstadt.de/~tom/IJCAI01/Rish.pdf

Rocchio, J. (1971). Relevance feedback in information retrieval. In Gerard Salton (Ed.), *The Smart Retrieval System-Experiments in Automatic Document Processing* (pp. 313–323). Prentice Hall.

Roy, A. R., & Maji, P. K. (2007). A fuzzy soft set theoretic approach to decision making problems. *Journal of Computational and Applied Mathematics*, *203*(2), 412–418. Retrieved from http://www.sciencedirect.com/science/article/pii/S0377042706002160

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian Approach to Filtering Junk E-Mail. *Learning for Text Categorization: Papers from the 1998 Workshop*. Madison, Wisconsin: AAAI Technical Report WS-98-05. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.1254

Salton, G, Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communication ACM*, *18*(11), 613–620. New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/361219.361220

Salton, Gerard, & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, *24*(5), 513–523. Tarrytown, NY, USA: Pergamon Press, Inc. Retrieved from http://dx.doi.org/10.1016/0306-4573(88)90021-0

Schneider, K.-M. (2003). A comparison of event models for Naive Bayes anti-spam e-mail filtering. *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1* (pp. 307–314). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dx.doi.org/10.3115/1067807.1067848

Schneider, K.-M. (2004). On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification. In J. Vicedo, P. Martinez-Barco, R. Munoz, & M. Saiz Noeda (Eds.), *Advances in Natural Language Processing* (Vol. 3230, pp. 474–485). Springer Berlin / Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-540-30228-5_42

Scholkopf, B., Bartlett, P., Smola, A., & Williamson, R. (1999). Shrinking the tube: a new support vector regression algorithm. *Proceedings of the 1998 conference on Advances in neural information processing systems II* (pp. 330–336). Cambridge, MA, USA: MIT Press. Retrieved from http://dl.acm.org/citation.cfm?id=340534.340663

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47. New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/505282.505283

Singh, S. R., Murthy, H. A., Gonsalves, T. A., Liu, H., Motoda, H., Setiono, R., & Zhao, Z. (2010). Feature Selection for Text Classification Based on Gini Coefficient of Inequality. *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, June 21st, 2010, Hyderabad, India* (Vol. 10, pp. 76–85).

Soucy, P., & Mineau, G. W. (2001). A Simple KNN Algorithm for Text Categorization. In N. Cercone, T. Y. Lin, & X. Wu (Eds.), *Proceedings of ICDM01 IEEE International Conference on Data Mining* (pp. 647–648). IEEE Computer Society Press, Los Alamitos, US. Retrieved from http://portal.acm.org/citation.cfm?id=645496.757723&coll=GUIDE&dl=GUIDE&CFID=91696714&CFTOKEN=59964605#

Stevens, S. S. (1946). On the Theory of Scales of Measurement. *Science*, *103*(2684), 677–680.

Sullivan, K. M., & Luke, S. (2007). Evolving kernels for support vector machine classification. *Proceedings of the 9th annual conference on Genetic and* *evolutionary computation* (pp. 1702–1707). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1276958.1277292

Tan, P. N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining*. Addison-Wesley Companion Book Site.

Tan, S. (2005). Neighbor-weighted K-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, *28*(4), 667–671. Retrieved from http://www.sciencedirect.com/science/article/pii/S0957417404001708

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, *2*, 45–66. JMLR.org. Retrieved from http://dx.doi.org/10.1162/153244302760185243

Tsoumakas, G., & Katakis, I. (2009). Multi-Label Classification: An Overview. *Database Technologies: Concepts, Methodologies, Tools, and Applications,* 309–319.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Mining multi-label data. *In Data Mining and Knowledge Discovery Handbook* (pp. 667–685).

Vapnik, V. N. (1998). *Statistical Learning Theory*. New York: Wiley.

Vidhya, K. A., & Aghila, G. (2010). A Survey of Naive Bayes Machine Learning approach in Text Document Classification. *International Journal of Computer Science and Information Security*, *7*(2), 206–211.

Wan, C. H., Lee, L. H., Rajkumar, R., & Isa, D. (2012). A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbor and support vector machine. *Expert Systems with Applications*, *39*(15), 11880–11888. Retrieved from http://www.sciencedirect.com/science/article/pii/S0957417412003120

Webopedia. (2012). Structure Data. *Webopedia*. Retrieved December 12, 2012, from http://www.webopedia.com/TERM/S/structured_data.html

Widyantoro, D. H., & Yen, J. (2000). A fuzzy similarity approach in text classification task. *Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on* (Vol. 2, pp. 653 –658 vol.2).

Wu, D., Vapnik, V. N., & Bank, R. (1998). Support Vector Machine for Text Categorization. *Learning*, 1–16. State University of New York at Buffalo. Retrieved from http://portal.acm.org/citation.cfm?id=1048295

Xiao, Z., Gong, K., & Zou, Y. (2009). A combined forecasting approach based on fuzzy soft sets. *Journal of Computational and Applied Mathematics, 228*(1), 326–333. Retrieved from http://www.sciencedirect.com/science/article/pii/S0377042708005001

Yang, X., Lin, T. Y., Yang, J., Li, Y., & Yu, D. (2009). Combination of interval-valued fuzzy set and soft set. *Computers and Mathematics with Applications, 58*(3), 521–527. Retrieved from http://www.sciencedirect.com/science/article/pii/S0898122109003228

Yang, X., Yu, D., Yang, J., & Wu, C. (2007). Generalization of Soft Set Theory: From Crisp to Fuzzy Case. In B.-Y. Cao (Ed.), *Fuzzy Information and Engineering* (pp. 345–354). Springer Berlin Heidelberg.

Yang, Y., & Liu, X. (1999). A re-examination of text categorization methods. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 42–49). New York, NY, USA: ACM.

Yang, Y., & Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 412–420). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=645526.657137

Yu, H., Yang, J., & Han, J. (2003). Classifying large data sets using SVMs with hierarchical clusters. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 306–315). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/956750.956786

Yukinawa, N., Oba, S., Kato, K., & Ishii, S. (2009). Optimal Aggregation of Binary Classifiers for Multiclass Cancer Diagnosis Using Gene Expression Profiles. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, *6*(2), 333–343. Los Alamitos, CA, USA: IEEE Computer Society Press. Retrieved from http://dx.doi.org/10.1109/TCBB.2007.70239

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338–353.

Zou, Y., & Xiao, Z. (2008). Data analysis approaches of soft sets under incomplete information. *Knowledge-Based Systems*, *21*(8), 941–945. Retrieved from http://www.sciencedirect.com/science/article/pii/S0950705108001056