Journal of Indian Research (ISSN: 2321-4155)
Vol.3, No.4, October-December, 2015, 107-120

Computer Science

DATA MINING METHODS AND THEIR APPLICATION IN RECOMMENDER SYSTEMS WITH 'R' PROGRAMMING LANGUAGE

Mudassir Makhdoomi *
Rafi Ahmad Khan**
Bharat Bhushan Sagar***
Sebiha Rehman****

ABSTRACT

Recommender Systems are processing tools which provide recommendations to people on various products. In this paper we aim to study some common data mining methods that have been successfully used in the Recommender Systems and simultaneously illustrate the methods by plotting them using various packages of R statistical programming language. Our focus will be on some commonly used classification methods: Entropy and Information Gain for selecting the most informative attribute(s) of the given data set, Naïve Bayesian Classifiers for predicting the class label when the attributes of the data set are independent of each other and Support Vector Machines, a geometric classification method.

Keywords: Classification, Entropy, Naïve Bayesian, Recommender Systems, Similarity Measures and Data reduction, SVM.

INTRODUCTION

Recommender Systems (RS) are processing tools which provide recommendations to people on various products like books, movies, music and several other shopping products [1]. They are

***Mudassir Makhdoomi** is a PhD Research Scholar at the Department of Computer Science, Mewar University, Chittorgarh, Rajasthan. He is also Assistant Professor at the Department of Computer Applications, Islamia College of Science & Commerce, Srinagar, J&K.

****Rafi Ahmad Khan** teaches at the Department of Business School, University of Kashmir, Srinagar.

*****Bharat Bhushan Sagar** is Assistant Professor at the Department of Computer Science, BIT Mesra, Noida Campus, Noida, U.P., India.

******Sebiha Rehman** works with the Directorate of School Education, Government of J&K, Srinagar.

simply software tools which provide suggestions to customers which suit their needs most. RS work on two strategies: *content filtering:* RS creates a separate profile for each customer, reflecting his nature and *collaborative filtering:* RS uses the past transactions done by the customer to provide recommendations [1] [2] [3].

The paper is organized as follows: After a brief overview of Recommender Systems and their workings, the paper will describe some important data pre-processing methods focusing on use of similarity measures in Recommender Systems (Section A) and data reduction techniques (Section B). While describing data reduction strategies, we'll focus on one of the most important data reduction strategies: Dimensionality Reduction (Discrete Wavelet Transforms (Section B.1) and Principal Components Analysis or *Karhunen-Loeve Method* (Section B.2)). After that, various classification data mining techniques used in Recommender Systems are described (Section C). The classification techniques explained are *Entropy and Information Gain* (Section C.1), *Naïve Bayesian Classifiers* (Section C.2) and finally *Support Vector Machines* (Section C.3). Finally in the last section, Section D, we conclude the paper with contours for the future work.

DATA PREPROCESSING METHODS

Data mining deals with large volumes of data [1]. There are various kinds of data that can be mined: database data, transactional data, time-related or sequence data, data streams, spatial data, etc. [1]. In all kinds of data, a datum or a data object represents an entity. A datum is described by a set of attributes or characteristics. An attribute of a data object represents its characteristic or a particular feature. Ideally, all the attributes of all the data objects are expected to have all the corresponding values. However, the real-world data is incomplete and messy. It contains a lot of noise and needs to be preprocessed before it can be used in data mining and machine learning algorithms [1], [2]. In this section, we discuss three issues that are important for designing a Recommender System. First, we discuss similarity or proximity measures; next we take up sampling of the data in case the data set is very large and finally we discuss dome of the data reductions techniques.

Similarity or Proximity Measures

In Recommender Systems, we need to know how similar or alike or how dissimilar or different the data object are with respect to one another. Similarity measures are used to determine how similar or dissimilar the data objects are in comparison to one another [2]. Nature of similarity measure to be used depends on the type of data under consideration. For data objects with numeric attributes, the most commonly used similarity measure is the *Euclidean Distance*:

$$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}=\|\mathbf{x}-\mathbf{y}\|_2$$

In the above equation, x and y are the data objects with n attributes, x_i and y_i are the i^{th} attributes of the two data objects [1], [2]. Another well-known similarity measure is the *Manhattan or City Block distance*:

$$d(x,y) = \sum_{i=1}^{n} | x_i - y_i | = | \mathbf{x} - \mathbf{y} |_1$$

Yet another similarity measure is the *Minkowski Distance*. It is actually the generalization of the Euclidean and Manhattan distance. It is given by:

$$d(x,y) = \sqrt[r]{\sum_{i=1}^{n} | x_i - y_i |^r} \text{ Where } r \in R^+$$

Above, r is called the degree of the distance [2]. If we substitute $r = 1$, we get the *Manhattan Distance* and if $r \quad 2$ we get the *Euclidean Distance*. This distance measure is also called $L_r Norm$. Hence, *Manhattan Distance* is also called $L_1 Norm$ and *Euclidean Distance* is also called $L_2 Norm$ [3]. The distance between data objects with binary attributes is measured using a different metrics. Data objects can be viewed as sets of features or attributes or characteristics. This is exactly the approach taken by another similarity measure called *Jaccard distance* or *co-efficient*. Two common operations on sets are the *union* and *intersection* of the sets. Suppose there are two data objects **X** and **Y.** Viewing the two data objects as two sets, the cardinal numbers of the union and intersection of **X** and **Y** are given by $|\mathbf{X} \cup \mathbf{Y}|$ and $|\mathbf{X} \cap \mathbf{Y}|$, respectively. The *Jaccard distance* gives the proportion of all the attributes or characteristics that are shared by the two data objects [3]. It is calculated as:

$$d_{jaccard}(x,y) \quad 1 - \frac{|\mathbf{X} \cap \mathbf{Y}|}{|\mathbf{X} \cup \mathbf{Y}|}$$

The above equation is one of the many forms of the *Jaccard Distance*. Consider the following; let *A01 = the number of attributes where x is 0 and y is 1, A10 = the number of attributes where x is 1 and y is 0, A11 = the number of attributes where both x and y are 1 and A00 = the number of attributes where both x and y are 0.* Then the following metrics are available for calculating the similarity between data objects having binary attributes: *Simple Matching coefficient* (SMC), *The Jaccard coefficient (*JC*)* and *The Extended Jaccard coefficient (or Tanimoto coefficient) (TC)* [1], [2].

$$SMC = \frac{A11 + A00}{A01 + A10 + A11 + A00},$$

$$JS = \frac{A11}{A01 + A10 + A11}, \text{ and}$$

$$TC = \frac{x \cdot y}{\| x \|^2 + \| y \|^2 - x \cdot y}$$

Where *TC* is the vector dot products of two sets of attributes possessed by the data objects [2]. The matrix whose elements are the distance values of the set of data object pairs from the given data set is called a *Distance Matrix.* In order to visualize a Distance Matrix, a special diagram is used called *Voronoi Diagram* which divides a plane, containing n points, into cell, edges and vertices [4]. Figure 1 shows a *Voronoi Diagram* of 10 numeric data points with 10 attributes each. The data points were generated using a uniform distribution. The diagram was generated using "*deldir*" package of *R statistical programming language* [5].

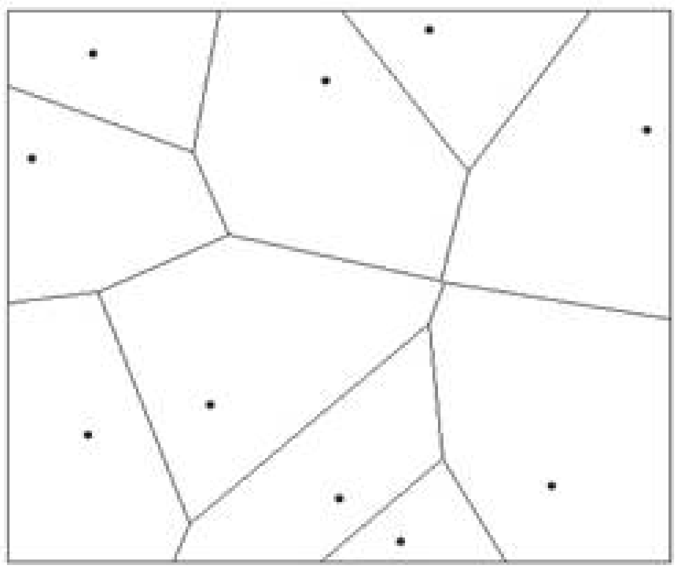

Figure 1: *Voronoi Diagram* of 10 numeric data points with 10 attributes each.

[The 2-D plane has been divided into 10 regions]

DATA REDUCTION TECHNIQUES

High-traffic e-commerce websites like amazon.com, eBay.com or Walmart.com or social networking sites like Facebook.com or Twitter.com (which use Recommender Systems for suggesting friends, pages, followers and ads) generate a huge amount of data. It will take a long time to perform data analysis and mining on such amounts of data [1], [6]. Hence, methods need to be developed that can be used to represent data in a much more compact way, yet convey the same meaning as the original data. In other words, the data needs to be reduced in volume without any loss of information. Data reduction techniques are used for the above stated purposes. A commonly used data reduction is called *Dimensionality.* The method of reducing the number of attributes or features of the data objects under consideration is called *Dimensionality Reduction*. Dimensionality of the data refers to the number of attributes of the given dataset. If we consider our data set represented as a 2-dimensional table with columns representing the values of a particular attribute and the rows representing a specific data point/object, then replacing some columns of the data set with a few or even just one column is called *Dimensionality Reduction* [7]. Popular dimensionality reduction techniques include *Discrete Wavelet Transforms (DWT)* and *Principle Component Analysis (PCA)*.

Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform (DWT) comes from the field of signal processing. However, it has been used widely in many statistical applications and also in data mining fields. In Recommender Systems, DWTs can be used as a data reduction technique. DWT is linear signal processing technique. When a DWT is applied to a given data object/vector (represented as a vector of features or attributes) it produces a "cardinally" equivalent, but numerically different data vector of *wavelet coefficients* [1]. Albeit, the two data vector are cardinally (lengthwise) equal, the usefulness of DWT arises in the fact that the wavelet transformed data can be trimmed by storing only a fraction of the strongest wavelet coefficients using some threshold value. Once the wavelet coefficients are trimmed, the original data can be *approximated* by applying the inverse of the used DWT [1], [8]. Popular DWTs include *Haar-2, Daubechies-4 and Daubechies-6* [1]. Figure 2 shows a *Haar Discrete Wavelet Transform* applied to 1024 random numbers generated from a lognormal distribution using *"wavelets"* package of *R statistical programming language* [9].

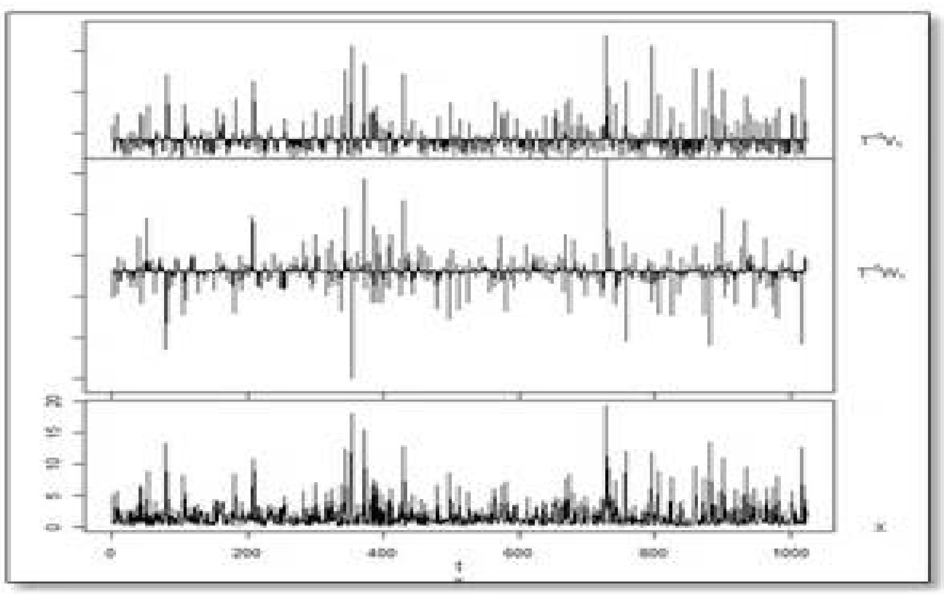

Figure 2: A *Haar Discrete Wavelet Transform* applied to 1024 random numbers generated from a lognormal distribution

Principle Component Analysis or Karhunen-Loeve or K-L Method

In order to understand Principle Component Analysis (PCA) at an intuitive level, we will proceed first by using an example from our everyday life. People devise concepts like "he is a 'good' student", but we can't directly measure the concept of "goodness" or how "good" is someone. This, however, means that we internally reduce many attributes of someone to just one attribute defining them all - "good" [10]. This is exactly how PCA works. PCA or K-L Method is the principle technique for dimensionality reduction in multivariate problems like recommending movies to a customer [11]. For recommending books, we've to consider many different attributes of the customer including his past history of movies. PCA works by finding k attributes among the n attributes of the given data objects, where k n such that the k attributes best represent the given data objects [1]. This way, the given data are interpolated to a much smaller dimensionality space. First of all, the given data are "normalized" to within a common range, such that each data fall within the chosen range. Next k orthonormal vectors are computed. These vectors are called *principle components.* These components essentially provide us with new axes for the given data. If these principle components are sorted in non-increasing order, they provide important information about the data variance. We used R programming language to do a PCA on a given data set of *"Arrests per 100,000 residents in 50 US states in 1973"* which has 4 attributes (Murder, Assault, Urban Population, and Rape). Figure 3 shows a Screen plot and Figure 4 shows a Biplot of the given data set.

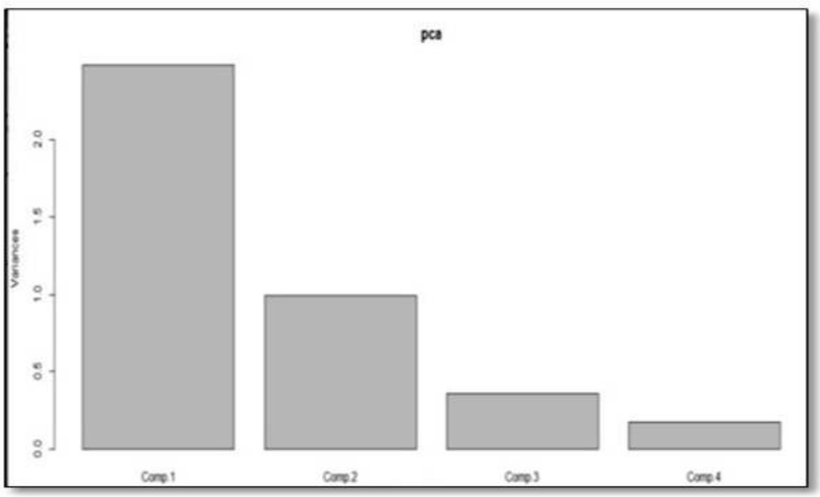

Figure 3: Screen plot of PCA of the *Arrests per 100,000 residents* which has 4 attributes; each bar corresponding to each attribute.

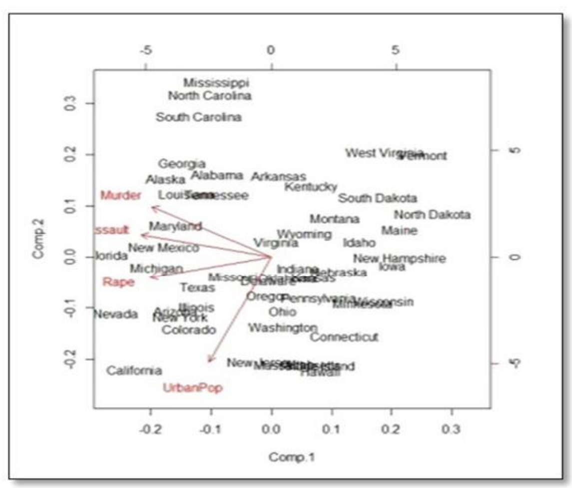

Figure 4: Bi plot of PCA of the *Arrests per 100,000 residents* which has 4 attributes. (Notice the orthonormal vectors, principle components of this data set.)

CLASSIFICATION

Classification is a data analysis technique which classifies the given data set into various similar sub-classes. It creates a model called *classifier* which is used to predict (predictive modeling) the *class label*. Classification is a stepwise process which starts by creating a model (classifier) from the given data. The data used in this step is called *training set.* Training set has a number of tuples; with each tuple having the form $X = (x_1, x_2, \ldots, x_n)$. This step is called *learning* or *training step.* The classifier "learns" from the training set which is made of up tuples and their associated class label(s). The associated class labels are called *target attributes.* The next step uses this model to predict the target attribute for the data instance which is to be classified. If the target attribute is available in the training set, the learning is called *Supervised learning* otherwise it's called *unsupervised learning* [1], [3], [7], [10], [2].

ENTROPY AND INFORMATION GAIN

In Supervised Classification problems, it is necessary to select the most valuable attribute(s). In other words, for predictive modeling, a set of attribute(s), among the given attributes, needs to be selected such that the selected set contains attributes providing important information about the target variable [3]. The selected set may contain a *single attribute* or it may contain *multiple attributes.* After selecting the informative attribute(s), we can divide or "classify" or "segment" the given data set into groups, in such a way that the resulting groups are distinguished on the basis of the target variable. It is preferable that the resulting groups be as pure or "homogenous with respect to the target variable" as possible [3]. For real-world data, it is, usually, difficult and, sometimes, impossible to find such informative variables which result in pure groups. A method or technique which enables us to find such a set of attribute(s) is called a *purity measure.*

An important purity measure, called *Entropy*, borrowed from Information Theory, pioneered by Claude Shannon, is widely used as a purity measure of the resulting groups [12]. Entropy is the measure of disorder of a system. In present case, the system is a particular group formed from the given data set. Each group member (data instance or data point) will have a collection of attributes. In supervised classification, these properties link with the values of the target variable. Entropy of this group means how mixed or "impure" this group is with respect to target variables. Entropy can be defined as:

$$entropy, E \quad -p_1 \log(p_1) - p_2 \log(p_2) - \ldots$$

In the above equation p_i is the probability (the relative percentage) of attribute i within the group. The probability, p_i ranges from 1 (when all group members have same values for attribute i and 0 (when no group members have same values for attribute i) [3]. Once we've calculated the entropies of all the "child" groups, we can know how "informative" an attribute is with respect to the target variable. Toward this end, we calculate a metric called *Information Gain* which measures the amount of "improvement" in entropy due to the attribute used in grouping the data set. In order words, it measures the amount of decrease of entropy of a group or how much information is added to a group due to the data set partitioning attribute. Information Gain (IG) can be calculated as:

$$\text{IG} = E(original) \quad [p(g_1) \times E(g_1) + p(g_2) \times E(g_2) + \ldots]$$

In the above equation, *E(original)* is the entropy of the original data set. $E(g_i)$ is the entropy of the i^{th} group and, $P(g_i)$ is the fraction/proportion of data instances belonging to that group [3] [10].

Bayesian classification

Bayesian Classifiers take a probabilistic approach to classification by predicting the class label of a given data objects or tuple [11]. The basic working idea of the Bayesian classification is Bayes' Theorem. The probability calculated to a Bayesian classifier is the conditional probability which is based on some background information [13].

Bayes' theorem is based on conditional probability. A conditional probability is a probability that is based on some background information [13]. Using the provided information, we can calculate the probability of some other event. If *A* is the required background information for calculating some other event *B*, then the "conditional" probability of event *B* given *A* is written as $P(B|A)$. The probability $P(B|A)$ is calculated using Bayes' Theorem as:

$$P(B|A) = \frac{P(A|B).P(B)}{P(A)}$$

$P(A|B)$, is *posterior probability* of A conditioned on B. Let $\boldsymbol{X}$ be data tuple described by the observations made on the set of n attributes. We can write $\boldsymbol{X}$ as $\boldsymbol{X} = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$, where x_i is the value of i^{th} attribute of the give data tuple. A Bayesian classifier calculates the probability of the class label, *C,* to which the tuple, ***X***, belongs to. Since ***X*** is the given background information, we can use, Bayes' Theorem to calculate $P(C|\boldsymbol{X})$ as:

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)}$$

The conditional probability in the numerator of the above equation $P(\boldsymbol{X}|C)$ can be calculated empirically from the given training set. It is simply the frequency with which $\boldsymbol{X}$ occurs among the tuples/instances belonging to class *C.* This probability is really hard to compute in practice. Even if each attribute is only symmetrical binary (both the values 0 and 1 are equally important), the number of combinations for $\boldsymbol{X}$ is 2^n and grows with number of values an attribute can take [11]. In order to solve this problem, a simplification is done where it is assumed that all the attributes are independent of each other that $P(\boldsymbol{X}|C)$ can be calculated as:

$$P(\boldsymbol{X}|C) = P(\{x_1, x_2, \ldots, x_i, \ldots, x_n\}|C) = P(x_1|C).P(x_2|C) \ldots . P(x_n|C)$$

This is the basis of a Bayesian classifier called Naïve Bayesian Classifier [14].

The following algorithm summarizes how a Naïve Bayesian Classifier works [1].

Let T be the set of training tuples and their associated class labels.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a training tuple with n attributes $\{A_1, A_2, \ldots, A_m\}$ and $C_1, C_2, \ldots, C_m$ be m class labels.

a) Calculate $P(X)$ which is constant for all classes.

Calculate

$$P(C_i) = \frac{|C_{i,T}|}{|T|}$$

Where $|C_{i,T}|$ is number of training tuples of class C_i in T.

b) If attribute A_k is a categorical attribute, calculate,

$$P(x_k|C_i) = \frac{|x_{k,T}|}{|C_{k,T}|}$$

Otherwise, if attribute A_k is continuous-valued, it is assumed to have a Gaussian distribution with a mean μ and standard deviation σ. Calculate:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

So that $P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$.

Calculate

$P(X|C_i) = \prod_{t=1}^{n} P(x_t|C_i)$.

This assumes all the attributes are independent of each other. This is called *class-conditional independence.*

c) Calculate

d) $$P(C_i|X) = \frac{P(X|C_i).P(C_i)}{P(X)}$$

Such that $P(C_i|X) > P(C_j)$

$for\ 1 \leq j \leq m, j \neq i$

This maximizes $P(C_i|X)$.

Figure 5 shows a plot, using *R programming language,* of the famous dataset called *Fisher's Iris data set* [15] also called *Anderson's Iris data set* [16]. The Iris dataset contains 50 samples of three species of Iris (*Iris setosa, Iris virginica and Iris versicolor*) along with observations of 4 attributes for each species: petal length, petal width, sepal width and sepal length. Although the data set is not related to Recommender Systems, it clearly illustrates how a Naïve Bayesian Classifier is used to predict the class label. The aim here is to predict the species (target class label) given the attribute values (a data tuple). For achieving this, we used "*e1071*" package of *R statistical programming language* [17]. After training our Naïve Bayesian Classifier with the Iris data set, we predicted the species using the 4 attributes

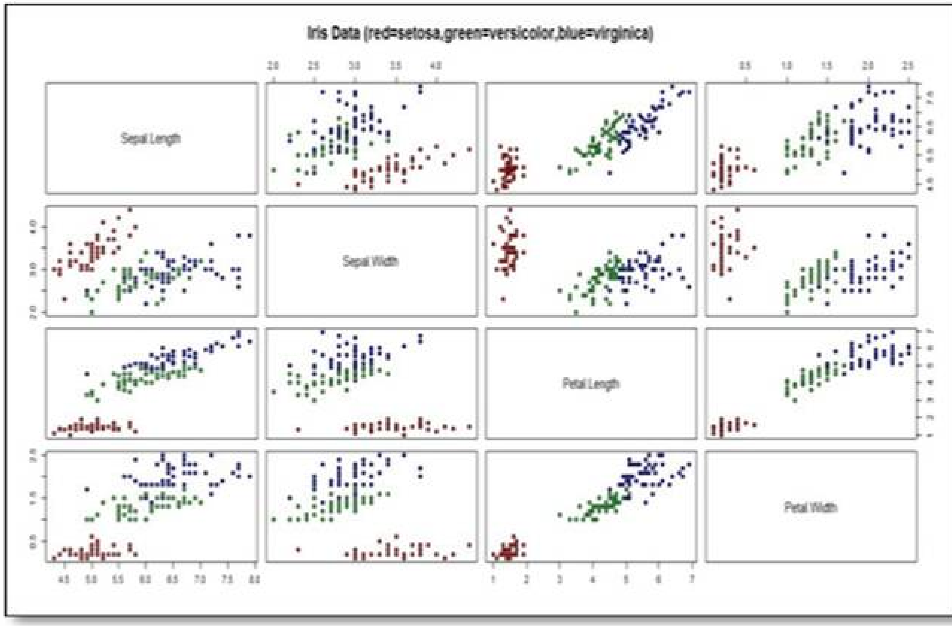

Figure 5: Plot of the Iris dataset. Red is for Setosa, green for Versicolor and blue for Virginica.

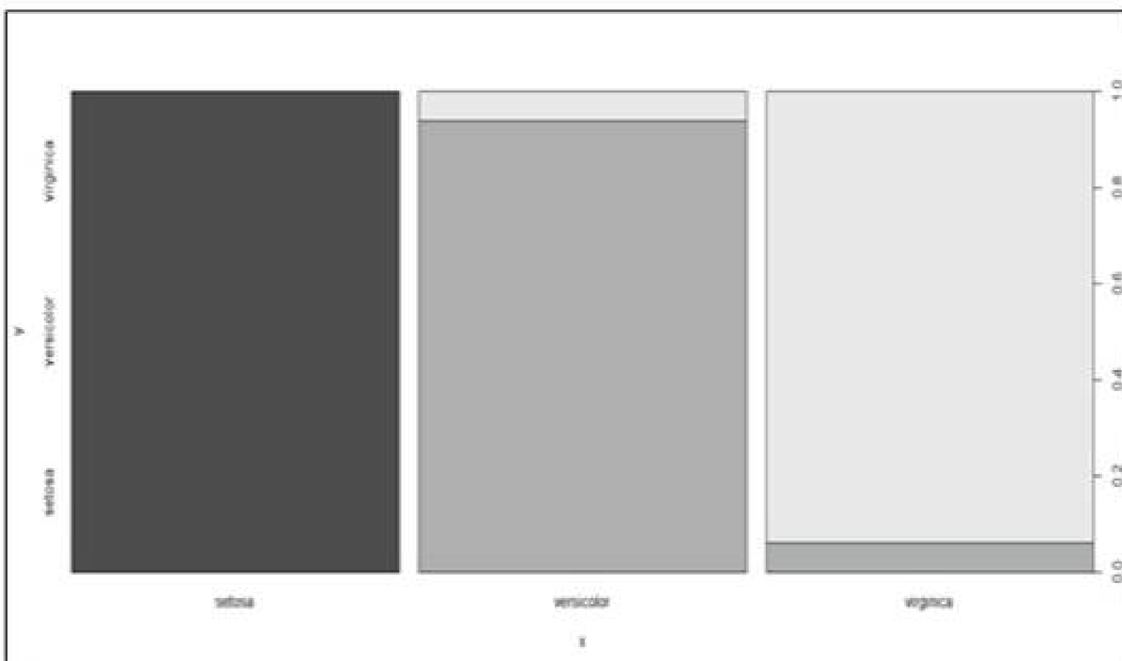

Figure 6: Iris species classified into 3 respective classes, each represented by a variable shade of grey color.

Support Vector Machines (SVMs)

An SVM classifier is a geometrical method which can be used to classify both linear as well as non-linear data. An SVM classifier uses a non-linear function or mapping to transform a given data set into some other higher dimension where it, optimally, searches for a linear hyper-plane or a decision boundary. This hyper-plane is then used to classify (separate) the data set into classes. If we consider a simple linearly-separable 2-D two class classification problem, as shown in Figure 7 and 8, it can be observed that several hyper-planes are possible. Notice that each hyper-plane has an associated margin. The goal of an SVM is to choose the hyper-plane that maximizes the margin. This optimal hyper-plane ensures misclassification is kept to minimum, if not completely eliminated [2]. For finding such maximum margin hyper-planes, a dot product between the two vectors is calculated. However, it is not always possible to linearly separate the given data set. Sometimes the data instances overlap in such a way that it is not possible to draw a straight hyper-plane. For such data sets, we can either let the stray data instances to be misclassified upto a certain tolerance error rate. This is done by introducing *Slack variables*. These variables associate a cost with each misclassified instance. Other solution is to draw a curved hyperplane, instead of a straight line. This is achieved, usually, by using *Kernalization* or "*kernel trick*". The main idea is to replace the dot product by a more general function.

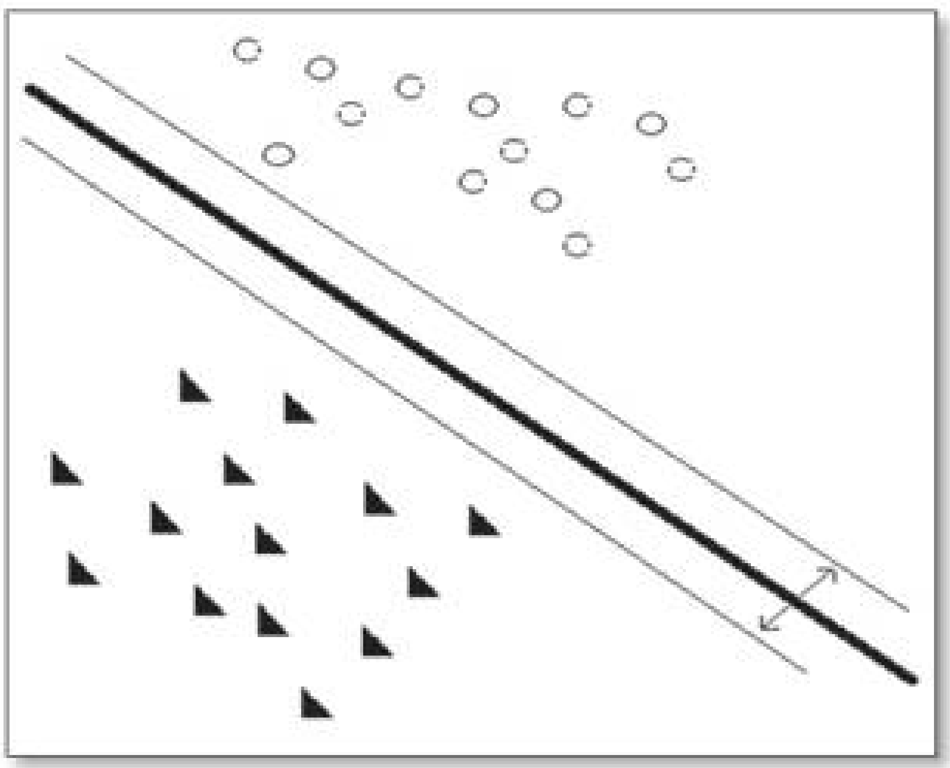

Figure 7: Linearly separable two class classification with SVM. The hyper-plane is associated with a small margin.

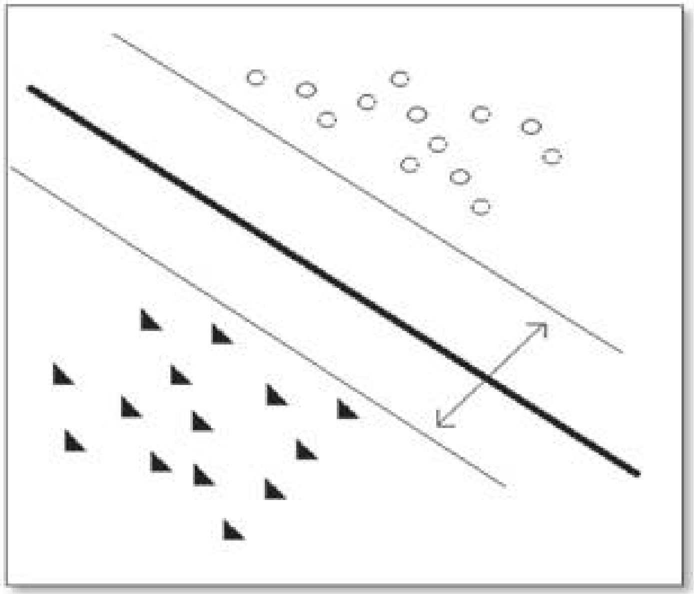

Figure 8: Linearly separable two-class classification with SVM. The hyper-plane is associated with a large margin.

Some commonly used kernel functions are Polynomial, Sigmoid and a family of Radial Basis Function (RBF) [1] [2] [11]. We will use a data set called "cats" for illustrating SVMs in R. The data set contains 144 rows and 3 columns [Sex, Bwt (Body Weight in kg) and Hwt (Heart Weight in kg)]. 144 adult cats were used for experiments with the drug *digitalis.* Their heart and body weights were recorded. 97 of the cats were male and 47 were female [18]. We will use the same"*e1071*" package of *R statistical programming language* [17]. The results are shown in Figure 9.

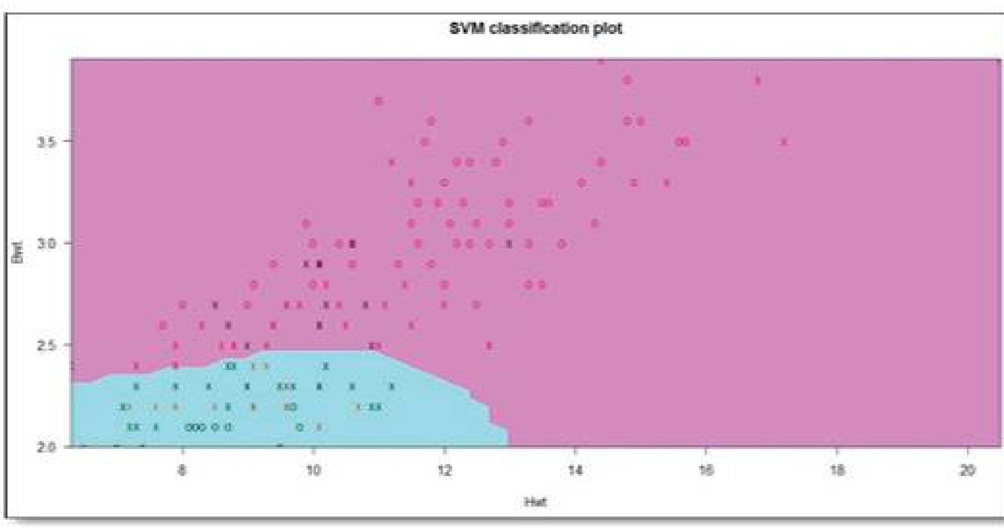

Figure 9: SVM classification of "cats" dataset with *Hwt* as x-axis and *Bwt* as y-axis. The pink area represents the male cats and the sky-bluish color represents the female cats. The boundary between the two colors is the hyperplane with RBF used as kernel function.

CONCLUSION AND FUTURE WORK

The paper only enumerated and explained some of the commonly used data mining methods used in RS. There are many other methods and techniques which we have not covered. Bayesian Belief Networks, Decision Tree Induction, k-nearest neighbor, Artificial Neural Networks, Association Rule mining and many clustering techniques have an excellent scope for future study. There is also a future scope for evaluating the efficiency, performance and accuracy of various mining methods with relative advantages and disadvantages.

REFERENCES

1. Han, J. ; M. Kamber & J. Pei(2012). *Data Mining: Concepts and Techniques*, MA, USA: Morgan Kaufman.
2. Ricci, F.; L. Rokach, B. Shapira & P. B. Kantor(2011). *Recommender Systems Handbook,* New York, USA: Springer.
3. Provost,F. & T. Fawcett(2014). *Data Science for Business,* Sebastopol, CA: O'Reilly.
4. Aurenhammer,F. & R. Klein(2000). *Handbook of Computational Geometry*, Amsterdam, Netherlands: North-Holland.
5. Turner,R.(2014). "*deldir: Delaunay Triangulation and Dirichlet (Voronoi) Tessellation*," 2 Febuary 2014. [Online]. Available at following link: http://cran.rproject.org/web/packages/deldir/index.html. [Accessed 2 July 2014].
6. Berkhin,P.(2006). *Grouping Multidimensional Data*, Springer Berlin Heidelberg.
7. Conway,D. & J. M. White(2012). *Machine Learning for Hackers*, Sebastopol, CA: O'Reilly, 2012.
8. Vidakovic,B. & P. Mueller(1991). "*WAVELETS FOR KIDS: A Tutorial Introduction*," Institute of Statistics and Decision Sciences, Duke University, Durham.
9. Aldrich,E.(2013). "*Wavelets: A package of functions for computing wavelet filters, wavelet transforms and multiresolution analyses*," 18 December 2013. [Online]. Available at http://cran.r-project.org/web/packages/wavelets/wavelets.pdf. [Accessed 5 July 2014].
10. Schutt,R. & C. O'Neil(2013). *Doing Data Science - Straight talk from the frontline*, Sebastopol, CA: O'Reilly Media.
11. Janert, P. K. (2012). *Data Analysis with Open Source Tools*, Sebastopol, CA: O'Reilly Media.
12. Shannon, C. E. (1948). "A Mathematical Theory of Communication," *The Bell System Technical Journal,* vol. **27**, pp. 379-423.
13. Downey, A. B. (2013).*Think Bayes,* Sebastopol, CA: O'Reilly Media.
14. Murphy, K. P. (2006). *Naive Bayes classifiers,* University of British Columbia.
15. Fisher, R. A. (1936). " The use of multiple measurements in taxonomic problems," *Annals of Eugenics,* Vol. **7**(2):179–188.

16. Anderson, E. (1936). “The species problem in Iris,” *Annals of the Missouri Botanical Garden* , Vol. **23**(3): 457–509.
17. Meyer, D. ; E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch, C.-C. Chang & C.-C. Lin(2014). “*e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*,” 17 March 2014. [Online]. Available:http://cran.rproject.org/web/packages/e1071/index.html. [Accessed 10 July 2014].
18. “R:Anatomical Data from Domestic Cats,” [Online]. Available: http://astrostatistics.psu.edu/su07/R/html/MASS/html/cats.html. [Accessed 2014 July 2014 based upon R. A. Fisher (1947). The analysis of covariance method for the relation between a part and the whole, *Biometrics* **3**, 65–68.