Business Forecasting with Various Data Mining Techniques

KAVYA mANDLI, 800984226, University of North Carolina at Charlotte

Data forms the most valuable resource for making business related decisions and Big Data has become one of the popular technologies implemented in various fields. There are billions of data streams being generated in every organization. These large data streams can be useful to the organizations in providing various insights and predictions: both positive as well as negative. Many factors like customer behavior, profits, losses, market trends and product value of a company in the market can be predicted by using various data mining techniques. There are many such techniques, how is one to decide which one produces the optimal results for Business analytics. Neural networks, decision trees and random forests are some of the best in data mining and these are the ones most frequently used. But, these are limited to multi-dimensional data. Hence, we propose a new data mining method known as ‘Similarity Forests’ which can be used for prediction and forecasting. Similarity forests is an extended version of random forests. It can be used to determine the similarity between any arbitrary datasets. Similarity forests can be used with simple and easy modifications in numerous practical occasions, where, the similarity values between objects are incompletely specified. Some of the business-related classification data sets are used for the experiments which will be implemented by all major data mining Techniques, including similarity forests to determine why it is preferably the best one.

CCS Concepts: • **Data** **Computing techniques → Regression trees and classification Trees**; *Supervised learning by classification*; **→** Data analysis/ Mining;

KEYWORDS

Big Data; Data preprocessing; Business prediction/ Forecasting; k-nearest neighbor; Random forests; Support Vector Machine (SVM); Similarity forests or SimForests; Naïve Bayesian

ACM Reference format:

Ben Trovato, G.K.M. Tobin, Lars Thørvӓld, Lawrence P. Leipuner, Sean Fogarty, Charles Palmer, John Smith, and Julius P. Kumquat. 1997. SIG Paper in Word Format. *ACM J. Comput. Cult. Herit*. 9, 4, Article 39 (March 2010),4 pages.

DOI: 10.1145/1234

1 INTRODUCTION

There has always been a revolution of information availability through internet exchange. While database management systems and warehouses can be used for the collection, storage, retrieval and manipulation of this information. There is a demand for organizations to know about how their data could be used as a prime driver for predicting growth and expansion of data mining techniques. Whereas the Machine learning folks have focused on getting some kind of pattern and knowledge from this data. This is often known as Data mining or Knowledge Discovery in Databases. Data mining refers to the drawing out crucial information from massive data sets through recognition of useful patterns within data. These are currently being used to store data logs holding process execution data to recreate definite business processes. The implementation of data mining has been developed and refined to create business process-mining techniques. Business-related mining practices use execution logs of business processes.

There are a lot of methodologies in data mining that make use of classification type of algorithms such as KNN, Naïve Bayesian, random forests, support vector machine and the new edition SimForests. Using well-defined and proper classes and a training set consisting of pre-classified examples, forecasting can easily be implemented with this idea going forward. K-Nearest Neighbors is one the most elementary and fundamental classification type of algorithm that can be used for mining purpose. It comes under the supervised learning methodology and helps recognize a deep purpose in pattern identification, intrusion detection in data and data mining. It can be used for both classification as well as regression predictive problems. An interpreter, a mechanism for finding the lowest calculation time is what is needed for this to be implemented for classification purpose while interpreting various business issues. There are various disadvantages of this algorithm : large data storage for huge datasets which are used by the model for training purpose , the distance has to be calculated between their test data and also all the training data has to be calculated for every test data. Therefore, This algorithm takes up a lot of time for testing.

Another ML supervised learning method is Support Vector Machine, which is used for both classifications and regression methods. But widely, it used for classification purpose. Some advantages of this algorithm are that it works amazingly well with when it comes to a clear margin of separation. Another advantage was that it worked well when it came to high dimensional spaces. But it works best when the number of dimensions are greater than the number of samples for testing the data. Now, let’s talk about the disadvantages of this algorithm. It has a low performance when working with huge datasets as it takes a lot of time for training the datasets that are huge in size. Low performance of the datasets results in noise where the target classes are overlapping. SVM does not work well when it comes to offering for probability estimates as well which adds to another limitation of it. An expensive five-fold cross-validation method is used for estimation in this case because of the above reason.

Now, let’s look into Naïve Bayesian technique before we look into random forests. Naive Bayes classifier is a powerful and more straight forward and easy to understand than Random Forest algorithm for the [classification](https://dataaspirant.com/2016/09/24/classification-clustering-alogrithms/) task. When working with million sets of record and very few attributes and entities, Naïve Bayesian is the best recommended technique. One of the other areas where Naïve Bayesian works best is with text analysis that also includes NLP or Natural Language Processing. It works based on the Bayes theorem. Bayes theorem named after Rev. Thomas Bayes. It works on conditionalprobability which works on the concept that something would occur on the basis of what already happened. This probability is conditional to the previous condition and its occurrence majorly depends on it. Thus, Naïve Bayesian uses this to calculate the probability of various events and predict depending on the prior information and data we have on it.

The next algorithm which we are looking onto is Random Forest. It is one of the other algorithms which can be used for both prediction and classification problems. Random forest is just a combination of multiple decision trees combined to form a forest. The accuracy of the model greatly depends on the total number of trees present in the forest. The greater the number of trees, the higher will be the accuracy of the model. Some of the advantages of his algorithm are : handling of the missing values in the data, it does not overfit like other algorithms when the number of trees in the forest are typically high, and one of the major advantage is that the random forest algorithm can also fit the categorical values. Banking, , Stock Market, E-commerce, Medicine, Information Technology are some of the areas in which this algorithm is widely used. In fact, this was the best performing algorithm when it came to classification problems. Occasionally, SVM was the only algorithm that outdid random forests when it came to classification problems but either ways, the accuracy and performance of random forests is quite superior when compared to SVM any day.

Random Forest is a robust classifier and although it works well when it comes to variation of settings. It is also a better algorithm to choose when working in the absence of domain related information. Random Forest was intended to work with multi-dimensional data where it creates trees to combine into a forest. But the limitation is that this high dimensional data might not be possible for all the applications like sequences, graphs and also time series data. That is when similarity Forest comes into picture where such applications can be solved and analyzed by calculating the similarities between various objects present in the data. These are specified by the domain specialists. Since it is now evident that random forests can’t be used in such cases, it is important that random forests do well while handling similarities of the subset one pair at a time. One limitation of calculating similarities between objects is that this method can be very expensive to use. Also, this can be used furthermore to restrict in a situation where we cannot possibly calculate or state all the similarities. We’ll look more into this method in detail further into this paper.

* 1. Issues with Data mining

1. Massive datasets and high dimensionality.

2. Understanding of patterns.

3. Overfitting or underfitting the statistical elements involved.

4. Redundant data and mixed changing.

5. And, data integration and non-standard or incomplete/insignificant data

This paper will seek to understand the best data mining techniques for business forecasting by comparison between the performances of the mining techniques using business-related data sets. These are KNN, Naïve Bayesian, Support Vector Machine (SVM ) and also the latest one ‘Similarity Forest’ or ‘SimForest’ algorithm. This paper as a draft of the main study report includes the title, introduction, related work, methods used and the references as forth.

1. RELATED WORK

In order to deal with the methods containing similarity matrix between a pair of objects. Extracting the multi- dimensional embeddings from the matrix which was formed by calculating similarity distribution would be the first solution to this problem. O(n3) time is needed for this and this is not very costly. However, it also needs O(n2) space to store the similarity matrix. If we consider this carefully, space of the order of 1012 is needed for a data set with multiple break points. O (n . log(n)) time and O(n) space is what is need for for an ensemble component and this is a potential solution too. Common classification techniques are hard to use when the similarities are not fully specified or only when there is partial information regarding it. Hence, this paper deals with ‘Similarity Forests’ which is an improvement on all of them

SimForests are built on top of random forests so they can work straight forwardly by the similarity matrices. Basic principle of Similarity Forests is that it assumes that data is embedded in some multi-dimensional space. Very high quality results are obtained when it can be combined with ensemble centered technique. This is when Similarity Forest was discovered when we could build random forests without needing the embeddings. Yet, we know that Random forest technique has a much better advantage when compared to the SVM algorithm. Using Similarity Forests is advantageous where we cannot use conventional Random forest technique. It can also be extended to all the cases where all the similarity pairs between the objects are non-existing. It occurs in many cases where there are a whole lot of data points and O(n2) values do not exist. It can also be used in cases where multi-dimensional representation is available and hence this can used for creating better similarity splits. Hence, this paper goes ahead to prove that Similarity Forest is a better data mining methodology that can be used rather than other classification algorithms for all the above mentioned reasons.

Similarity comparisons are often quite expensive, and it may not be feasible to compute all pairwise similarities from a computational point of view. Therefore, it is important to be able to construct the random forests efficiently while computing an extremely small subset of the pairwise similarities. After all, the number of pairwise similarities scales up quadratically with the number of points. Therefore, any method that requires the entire similarity matrix between pairs of objects to be specified up front is bound to face computational and storage challenges. Furthermore, in cases where all pairwise similarities are not specified (or even computable), this type of approach can still be used, albeit in a limited way. More about similarity forests in detail in the methods being used section. This work on Similarity Forests algorithm was completely done and presented by Saket Sathe and Charu Aggarwal (IBM T.J.Watson Research Center) at the KDD 2017 conference.

3 METHODS USED AND UNDERSTANDING OF ALGORITHMS

3.1 Support Vector Machine Algorithm

3.1.1 Introduction to Support Vector Machine Classifier

SVMs can be used to perform binary classification and regression estimation tasks. The support vector machine algorithm has been invented by Vapnik and Chervonenkis. Initially, this was only used to detect hyperplanes for other classifiers which are linear. Later in the year 1995, both the inventors found out a way to also build a classifier for non-linear and this was done by using a kernel. Therefore, this SVM was one of the foremost algorithms to be considered as a supervised learning technique. Support Vector Machine classifier technique can be implemented for all the datasets that contained both the features as well as class labels .But SVM is considered as an unsupervised algorithm if there are no features or class labels to deal with. This leads to talking about “Support Vector Clustering” and it can be implemented in place of Support Vector Machine in the absence of features and class labels and when it is treated as an unsupervised algorithm. The major advantage working with them is that first, they minimize the expected error rather than the classification error and second, they employ the duality theory of programming to get a dual problem the admits efficient methods.

3.1.2 Support Vector Machine Classifier working

If we consider a dataset with features and labels, SVM can be used to build a model to predict the classes for all the new data points present in the provided dataset. It then allocates these new data points to the available classes. The classifier is called as the “Binary Support Vector Machine” if the dataset consists of two different classes.

There are two types of support vector machine classifiers which are as follows.

* Linear Support Vector Machine Classifier

Every point of data shall be treated as a p-dimensional vector in the linear support Vector machine classification model. These vectors are usually a list of p-numbers. The separation of these data points is done by the usage of p-1 dimensional hyperplane. Even though this data can be split by many number of hyperplanes in a linear fashion, it is recommended to use the one which maximizes the margin. This nothing but the distance between hyperplane and nearest data point in either of the two classes. This would be the best one and is called the “Maximum margin hyperplane” and it computed by the data point which is the nearest to it. These data points are called as support vectors in the classifier

* Non-Linear Support Vector Machine Classifier

Usually, it not possible to separate the p dimensional space non linearly. Hence, in order to resolve this problem, mapping the p-dimensional space to a higher dimensional space was proposed by one of the inventors of SVM. Each of the kernels are told to hold a non-linear kernel functions and by using this little idea, the non-linear hyperplanes can be drawn. A high dimensional feature space can be constructed by using this phenomena. The kernels can be of various types such as : polynomial homogeneous kernel, polynomial non-homogeneous kernel, radial basis function kernel and all this can be used in Non linear SVM classifiers.

3.1.3 Support Vector Machine Hyperplanes

We can see below that the two parallel hyperplanes can be used in order to separate the classes of data present and the ares that is present in between these hyperplanes is called as “Maximum Margin Hyperplane

‾wxi – b >= 1 if θi = 1

‾wxi– b <= 1 if θi = -1

|‾w | = normal vector to hyperplane

θi = classes

Xi is used for features.

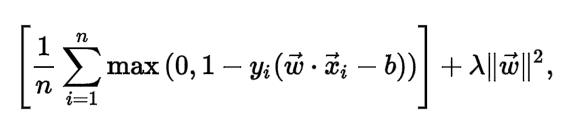
Distance between both the hyperplanes is given by 2 / |‾w |

|‾w | should be minimized to make the distance max.

We can develop a joint equation as follows for a proper equation which is as below:

|‾w | min for θi (‾wxi – b ) >= 1 ∀ i = 1,2,3,4 …..., n

In order to construct a classifier for non-linear data, the below equation must be minimised



In this equation, if Xi is on the correct side of the margin, then max() method will result in a zero. Function’s value is proportional to the distance from the margin if the data is present on opposite side of the given margin.

λ is tradeoff between increasing margin size

Xi here represents the correct side of the margin.

3.1.4 Advantages of Support Vector Machine Classifier

1. It is effective in high dimensional spaces.
2. SVM is memory efficient as it uses a large subset of training points.
3. Support vector machines are efficient when the number of features is very large. This type of classifier is effective when the number of features is greater than the number of samples.
4. Kernel trick is used to build hyperplanes which can be used to separate the linear and non-linear data.
5. This r is a robust model in solving prediction issues because it maximizes margin.
6. Clear margin of separation is needed for it to work well

3.1.5 Disadvantages of Support Vector Machine Classifier

1. It is very complex and uses large storage requirements.
2. Leads to poor performance and overfitting when the number of samples are high.
3. Probability estimates are not specified.
4. It has good generalization performance. However, it can be very slow in the test phase.
5. Selecting a wrong kernel leads to a rise in error percentage.
6. It uses quadratic programming and hence large computation time.

3.1.6 Applications of Support Vector Machine Classifier

A few applications of support vector machine are as follows.

1. Handwritten digit recognition: Isolated handwritten digits which are optically scanned using this technique and are recognized accurately.
2. Facial expression classification: These are used in classifying facial expressions. Statistical models of shape and support vector machines are also widely used
3. Text Categorization: SVM plays an important role in information retrieval.
4. Speech Recognition: Can be used to distinguish speech by differentiating between the keywords and non-keywords by using the vectors of the words.

3.2 K-Nearest Neighbor Algorithm

3.2.1 Introduction to k-nearest neighbor classifier

As we already know, both classification and regression predictive problems can be solved by using the K nearest Neighbor. But, in the industry it is mainly used for the purpose of classification problems. It is a non-parametric supervised leanring classifier that can be used for both classification and regression methodologies. K-nearest neighbors was first discovered by Fix & Hodges in the year1951 to identify pattern tasks. If a classification associated task has to be performed, then K nearest neighbor is the one of the best choice. It calculates the target variable label by going through the nearest neighbor class varible. Distance measures like Manhattan distance and Euclidean distance can be sued for finding out the nearest class.

3.2.2 Pseudocode for K-nearest Neighbor.

Problem Statement:

Let (Ai, Bi) where i = 1, 2,3,4,5and so on till n be the data points. Ai indicates the feature values and Bi indicates the labels for Ai for each i. Now, let’s assume that the number of classes as ‘c’, such that Ci ∈ {1, 2, 3,4,5…, c} for all the values of i. A be a data point whose class label is not known and it would be determined by applying KNN.

Pseudocode:

Step 1 : Compute Dist (A, Ai) where Dist represents the Euclidean distance between the data points and i = 1, 2, ……, n.

Step 2 : Arranging the computed n Euclidean distances in non-decreasing order.

Step 3 : Let k be a positive integer, select the first k distances from this non-decreasing order list.

Step 4 : Determining the k-points corresponding to the k-distances.

Step 5 : Let ki denote the number of data points belonging to the i class among k points, which is, k ≥ 0

Step 6 : If ki > kl ∀ i ≠ l then put A in class i.

3.2.3 How to choose the value of K?

The optimum K will always vary depending on the dataset. It should be as big such that the noises won't affect the prediction highly. If in case the value of k is minimal. It will result in noise having a higher effect on the output results. And as low such that one factor won't dominate another factor. The higher value of k, it makes it costly computationally which opposes the fundamental idea of KNN (the data points that are close might belong to similar classes). The most important task to do in KNN algorithm is to choose the value of K. There are a number of techniques. While the most accurate one is to try many K values and do Cross-Validation to see which K value is giving the best result. However, this is not feasible all the time. And it leads to a very high probability of overfitting. The simplest method to choose the value of k would be by using the formula k = n (1/2) where n is the number of features.

To improve the output results of this, we use the cross-validation method. This can be used for testing purposes to test the k-nearest neighboring algorithm with as many as k values who has the patience to try. Therefore, the model with good accuracy can be used an ideal method. Best way to go about with this is by going through each feasible values of k and testing the result accordingly. But again, it is distinct for different cases.

3.2.4 Condensed Nearest Neighbor Data Reduction Rule

When working on a large dataset, the tasks might be very costly and expensive. The main thing to do is clean the data and then sort it depending on the key observations of the data when it comes to condensed Nearest Neighbor rule. This results in lowering of the execution time needed but the disadvantage is that it also lowers the accuracy of the prediction model.

The steps to reduce and to divide data points

1. Outliers: Outliers are the observations which lie at a far-away distance from the rest of the data points. They are usually considered as extreme values. In order to increase the accuracy of the model, outliers have to be removed.
2. Prototypes: The minimum number of points in the training set needed to recognize the non-outlier points are called as prototypes.
3. Absorbed points: The points that are precisely known to be the non-outlier points are the absorbed points.

3.2.5 Advantages of K-Nearest Neighbor Algorithm

1. Training set can trained even without knowing the structure of data.
2. It works best with datasets which have small number of rows and columns, basically small datasets.
3. Re-training of the model is not required even though a new pattern is added to the already trained KNN model
4. The performance of K nearest neighbor asymptotically approaches that of the Bayes classifier.

3.2.6. Disadvantages of K-Nearest Neighbor Algorithm

1. A lot of space for storage purposes might be required for very large datasets
2. For each test data, the distance should be determined between test data and all the training data, thus it takes a lot of processing time.
   1. Naïve Bayesian Classifier
      1. Introduction to Naïve Bayesian

The Bayes theorem is used by the Naïve Bayesian classifier. Naïve Bayesian classifier is used to predict membership probabilities such that for each class, the probability that given record or data point belongs to the same class.  The most likely class is the one with the highest probability. This is also known as Maximum A Posteriori (MAP).

The MAP for a hypothesis is: MAP(H)  
= max( P(H|E) )  
=  max( (P(E|H)\*P(H))/P(E))  
= max(P(E|H)\*P(H))

P(E) is evidence probability, and it is used to normalize the result. It remains same so, removing it won’t affect.

3.3.2 Example of Naïve Bayesian

Let’s consider that the Path Lab is performing a Test of disease say “Di” with two results “P” & “N.” where P stands for “positive” and N stands for “Negative” This guarantees that their test result is about 99percent accurate: It means that if you have the disease, 99% of the time, your test will be P. If you don’t have the disease, 99% of the time, the test will be N. If about 3% of all the people have the disease and this test gives “P” result, what will be the probability that you actually have the disease?

In order to solve the above problem, conditional probability has to be used.  
The probability of people suffering from Disease Di, P(Di) = 0.03 = 3%  
The probability that test gives “P” result and patient have the disease, P(Pos | Di) = 0.99 =99%

The probability of people not suffering from Disease Di, P(~Di) = 0.97 = 97%  
The probability that test gives “P” result and patient does have the disease, P(Pos | ~Di) = 0.01 =1%

To calculate the probability that the patient actually have the disease i.e, P( Di | Pos) we will use Bayes theorem:

\textrm{P(D \textbar Pos) = }  \frac{\textrm{ P(Pos \textbar D) * P(D)}} {\textrm{P(Pos)}}

The various values of numerator but we need to calculate P(Pos):  
P(Pos) = P(D, pos) + P( ~D, pos)  
= P(pos|D)\*P(D) + P(pos|~D)\*P(~D)  
= 0.99 \* 0.03 + 0.01 \* 0.97  
= 0.0297 + 0.0097  
= 0.0394

Let’s calculate, P( D | Pos) = (P(Pos | D) \* P(D)) / P(Pos)  
= (0.99 \* 0.03) / 0.0394  
= 0.753807107

So, Approximately 75% chances are there that the patient is actually suffering from disease.

* + 1. Advantages of Naïve Bayesian Classifier

The advantages of the Naïve Bayesian Classifier are as follows

1. Naive Bayes Algorithm is a fast, highly scalable algorithm.
2. Naive Bayes can be use for Binary and Multiclass classification. It provides different types of Naive Bayes Algorithms like GaussianNB, MultinomialNB, BernoulliNB.
3. It is a simple algorithm that depends on doing a bunch of counts.
4. Great choice for Text Classification problems. It’s a popular choice for spam email classification.
5. It can be easily train on small dataset.
   * 1. Disadvantages of Naïve Bayesian Classifier
6. It considers all the features to be unrelated, so it cannot learn the relationship between features.
   * 1. Applications of Naïve Bayesian Classifier
7. It can be used for  classification for credit scoring in e-lending platform.
8. Check a piece of text expressing positive emotions, or negative emotions and in sentiment analysis.
9. To mark an email as spam or not a spam, this classifier can be used.

3.4 Random Forests Algorithm

3.4.1 Introduction to Random forests algorithm

As we have already seen in the introduction section, the random forests are a category of supervised learning and it build the forest comprising of several decision trees. The accuracy of the model always depends on the number of decisions in the forest. The higher the number of trees, the more will be the accuracy of the random forest model. Before we go forward with random forest, it is essential to see how decision trees work and comprise of the forest. So, in decision trees, nodes and rules are formulated by using the gini index, information gain and co-efficient computations. But, when it comes to random forest, implementing the root node and diving the feature nodes takes place randomly.

3.4.2 Example of Random forest algorithm

We have the following sample dataset to see how Decision Tree works. A random forest is a collection of decision trees.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Hair** | **Height** | **Weight** | **Lotion** | **Sunburn** |
| X1 | blonde | average | light | No | yes |
| X2 | blonde | tall | average | Yes | no |
| X3 | brown | short | average | Yes | no |
| X4 | blonde | short | average | No | yes |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X5 | red | average | heavy | No | yes |
| X6 | brown | tall | heavy | No | no |
| X7 | brown | average | heavy | No | no |
| X8 | blonde | short | light | Yes | no |

Fig 2: Sunburn Dataset

We have the following rules corresponding to the tree.

If Hair=blonde and Lotion=no then Sunburn=yes;

If Hair=blonde and Lotion=yes then Sunburn=no;

If Hair=red then Sunburn=yes;

If Hair=brown then Sunburn=no;

This is one such tree corresponding to an entire forest. A Decision tree is the base of a Random forest. It is a classification method which generates a tree and a set of rules, representing the model of different classes. It also helps in visualizing the data as we can see below.

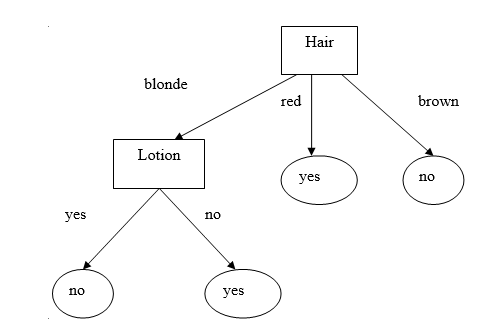


Fig 3 : Decision Tree for the Sunburn Dataset

The pseudocode for this algorithm can be divided into two steps as follows.

Step 1: Random forest creation pseudocode

1. Selection of random “k” features from total “m” features where k < < m.
2. Among the “k” features, calculate the node “d” using the best split point.
3. Divide the node into daughter nodes using the best split. Repeat 1 to 3 steps until “l” number of nodes has been reached.
4. Construct forest by repeating steps 1 to 4 for “n” number of times to create “n” number of trees.

Step 2: Random forest prediction pseudocode

1. The algorithm takes the input as test features and applies the rules of each randomly built decision tree to predict the target result. Also, it stores the target result.
2. Computes the votes for each target result.
3. Consider the high voted target result as the final prediction from the random forest algorithm.

In order to perform the prediction using the trained random forest algorithm, all the test features should go through the rules of each randomly built decision trees. The important feature here is if we use a hundred decision trees to build a random forest, each one will result in a different target result. By predicting this, each target of the predicting votes can be computed. But, if the number of random decision trees are predicted using some random unique targets say i, j, k then the votes of ‘i’ is equal to the number of trees whose prediction is equal to ‘j’ in the hundred random DTs. Similarly, the same procedure is followed by the other two targets y and z. When ‘x’ is getting most of the votes, say, 60 out of 100 decision trees are predicting the target as ‘x’, then the final random forest returns the predicted target as ‘x’. This is known as Majority Voting.

3.4.3 Application of Random Forest algorithm

1. Random Forests are widely and hugely used in banking sectors by predicting various things like if the customers are profitable to a particular bank or not.
2. It is also used in the medical fields to detect the right number of components needed for the medicine.
3. In the stock market trading, random forests are used to identify the stock behavior and predict the expected profit or loss of the stock.
4. One of the major applications of Random forests is that it can be used in E-commerce in order to build a recommender systems for different customers and different sectors.

3.4.4 Advantages of Random Forest algorithm

1. Random Forests never undergo overfitting issues and hence can be used for any classification type of algorithm.
2. But apart from that, it can also be sued for regression.
3. Feature engineering can be performed by Random Forests, which means detecting the most significant features from the available set of features in the training set

3.4.5 Disadvantages of Random Forest algorithm

1. This algorithm over fits some of the datasets in case of noisy classification and regression cases.
2. It is more biased towards the attributes which have more levels
3. The significant scores of variables are not reliable for this kind of data.

3.5 Similarity Forests Algorithm

3.5.1 Introduction to Similarity Forests algorithm

In this “paper, we are introducing a newly developed algorithm named as “Similarity Forests” by Saket Sathe and Charu C. Aggarwal. In short, this algorithm is also called as ‘SimForest’. The Similarity Forests algorithm is nothing but the construction of random forests with similarity matrices exclusive of assuming access to a precise multidimensional representation of data. Let’s think that the data is set in a certain imaginary theoretical multidimensional space and then the coordinates of the data points are calculated along one-dimensional projections by using the similarities. Such projections are built by sampling pairs of data points which should be held by different classes, and then project the remaining data points for each pair in parallel to the theoretical multidimensional direction representing the line joining the pair. This projection can be implemented through using pairwise similarities.

Consider, a collection of ‘n’ objects represented by O1, O2, …, On. These ‘n’ “objects can be of any data type. However, it is not important to know the specific object type if we can calculate a similarity value between them. Let, Sij represents the similarity between the objects Oi and Oj. Similarity need not be available readily to the algorithm. For example, a domain-specific similarity function might be used in the time series domain. In this type of situations, the similarities between the objects are determined immediately as required by the algorithm. The algorithm design needs to minimize the calculations. Assume that, the labels are derived from a collection of ‘c’ different possibilities represented by” L = {1 … c}. Also, suppose that the ith object in the training data is labeled as li ∈ L.

Problem Statement: Given, a collection of ‘n’ objects O1” … On , which are labeled as l1 … ln , build a random forest with the help of only calculated pairwise similarities between these objects and without access to the multidimensional representations of these objects if they are present.

The final model is utilized for classification of objects with not known labels.

3.5.2 SimForest Algorithm

Here, the embeddings be ‾X1 ...‾Xn. Assuming that the the directions in the data space refers to theoretical embeddings. The approach builds each decision tree in the similarity forests in a top-down fashion with the help of the many recurring splits. It is very much like the conventional random forests approach, in which it samples features for splitting. Likewise, the SimForest algorithm samples directions from the original space.

The whole idea of utilizing multivariate directions in the data space for splitting is even used in random forests. But, only for a similarity-based approach, SimForest method uses an entirely different result by stating the random directions based on splitting. These pairs of objects are sampled from the objects existing at the node, which is to be divided. An interesting discovery is that a proper sampling of pairs of objects even improves the performance of multidimensional random forests. This means that even if actual multi-dimensional representations of data points are given, it is more accurate to implement this approach over the random forests. In specific, ‘r’ pairs of objects are sampled to define ‘r’ different random directions at each node.

There are two ways to sample these pairs of objects as follows.

1. Choosing the pairs randomly without giving importance to the class label.
2. Choosing the pairs randomly to fit into various classes. This means that the selecting the first object in the pair is done randomly, whereas choosing the second object is done by selecting it from the objects belonging to a different class.

The second way leads to more biased splits and thus reduces the height of the tree built. This method offers better results; hence, this is the best choice to be selected. The ‘r’ value should be small so that the diversity can be maintained. Also, it is valuable to maintain the method’s efficiency by using the small values of ‘r’ and the accuracy can be improved by increasing the number of trees in the forest. Through the experiments, the best thing is to set the ‘r’ value to 1.

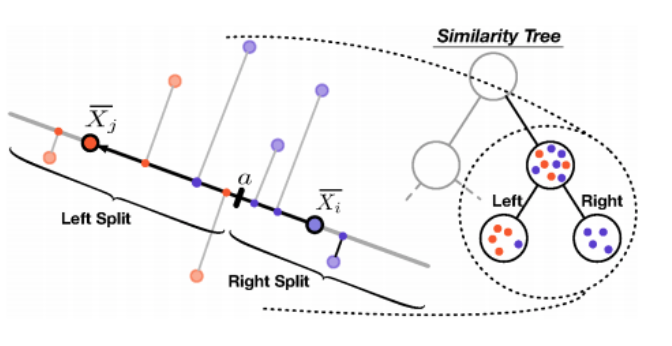


Fig 4 : SimForest splits

As we can see in the image “below, it explains the conceptual illustration of splitting for similarity tree construction. In the similarity forest, each decision tree is built recursively until the leaf nodes fit into a specific class. Though, the building of a decision tree most of the times has a pruning phase, in which, the lower nodes of the tree are detached to reduce the overfitting, this cannot be done in all possible types of random forests including the similarity forest. Hence, we shall build the tree to full height without pruning, until each node just consists of instances fitting into a specific class.

After” the decision tree has been built, the appropriate pairs of objects and their split points are stored at each node. Both are needed for the testing phase. It is significant to remember that the pair of objects (Oi, Oj) are sorted for consistency with the testing phase because the projection (Sij – Sii) of object Oi being always negative and the projection (Sjj – Sji) of object Oj being always positive. It is always true that the absolute magnitude of self-similarity Sii is greater than the absolute magnitude of any cross-object similarity Sij for the most rational similarity functions. This type of sign convention had begun by the fact that the direction of the projection vector is from Oi to Oj.

In the testing phase, it uses similar kind of approach to project the test points on the line defined by the pairs of training points at each node. Consider, (Oi, Oj) is the defining pair of objects for a given node, then the value of Skj – Ski is determined for the test object Ok. The stored split point ‘a’ in the image below is used to compute which path to follow in the decision tree based on whether Skj – Ski ≤ a”. This step is implemented for each node on the path of the tree until the object Ok is allocated to a leaf node. The label of the leaf node is conveyed as “the prediction.

3.5.3 Advantages of SimForest Algorithm

1. This can be naturally extended to circumstances even if all the pairs of similarity values are absent.
2. The bias characteristics of the individual components of the similarity forests are mostly favorable and through sampling pairs of points fitting into various classes, we end up in choosing biased directions at least in a randomized sense most of the time.
3. For domains such as sequences, graphs, times series data where random forests cannot be used, Similarity Forests makes a huge advantage
4. More time and apace is saved by this algorithm.
5. Furthermore, it extends Random forest and SVM and it can used in areas where the other two algorithms show limitations.

3.5.4 Applications of SimForest Algorithm

1. Healthcare sector
2. Stock markets
3. Banking sector
4. E-commerce
5. Medical sector.

And many more.

4. EXPERIMENTAL RESULTS

We show the experimental results comparing Sim Forest algorithm with other competitive algorithms using business-related datasets in this section. We shall begin by explaining the datasets in the next sub section. The description of evaluation methodology in section 4.2. Then, we test the SimForest’s resistance to noisy similarities in section 4.3.

4.1  Experimental Datasets

We use the Adults dataset for this experiment. Some of the important characteristics of these datasets are given in the table below. This data originates from the U.S.A. census bureau department database and contains 48842 instances with 14 attributes like age, work class, education, etc. In the original application of the data, the attributes are to be used in order to predict the income level of all the individuals. I’ve added the attribute “income” with levels such as small and large, representing an income of less than or equal to 50,000 USD and greater than 50,000 USD respectively. This data is included in arules as the dataset of the package AdultUCI.

* + 1. Data Preparation

Two attributes fnlwgt and education-num have been removed. The 1st attribute is a weight calculated by the creators of the data set from control data provided by the Population Division of the U.S.A. census bureau. The second attribute which ahs been removed is just a numeric representation of the attribute “education” which is part of the dataset. All the four numeric variables are added to bin (age, hours-per-week, capital-gain and capital-loss) to categorical attributes by building suitable categories. We divide the attributes age and hours-per-week into suitable categories using knowledge about typical age groups and working hours. For the two capital related attributes, we create a category called None for cases which have no gains/losses. Then we further divide the group with gains/losses at their median into the two categories Low and High.



Fig 5: Glimpse of the dataset

The following are some of the tasks performed for data preparation and training:

* It is worthwhile to note that any factor variables in predictors can have a maximum of 32 levels, so consider regrouping if your have more than 32 levels. All the categorical variables with more than 32 levels have been removed.
* The data is split in the ratio of 80 and 20, 80 being the train data and 20 being the test data.
* Now, models have been built for the prediction if a person’s income will be more than 50k per year or not
* Let’s construct the Random Forest, K-nearest neighbor, Support Vector Machine, Naïve Bayesian and also Similarity Forest models to compare the accuracy of their predictions.

4.2 Evaluation Strategy

We perform an 80-20 separation of data for each experiment. 80% of the data is used for training the classifier, 20% of the data is used for testing the performance of the classifier. This training data is divided into portions for model constructions and parameter tuning most of the times (ten percent is used for tuning). The accuracy of the model is always calculated in percentage and it is measured by the number of data points that were predicted rightly.

4.3 Screenshots and Results:

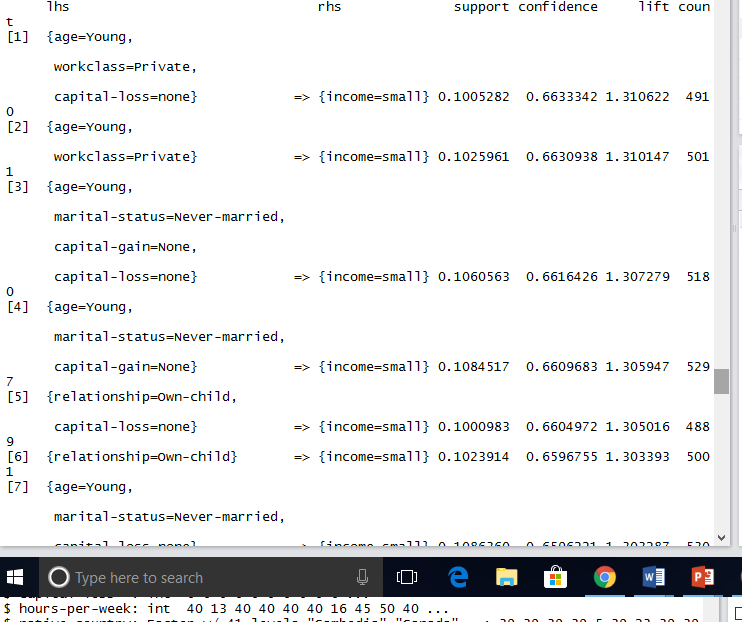


Fig 6: Screenshot of the redundant rules

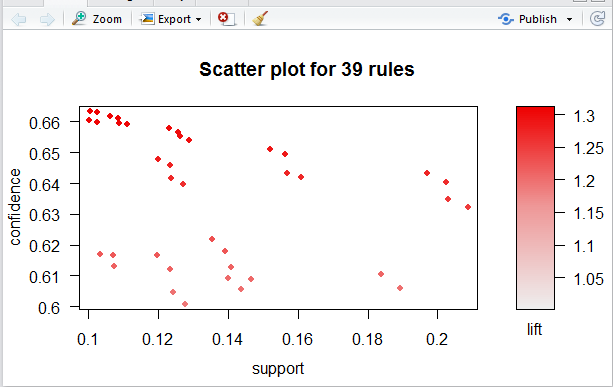


Fig 7:Scatter plot for redundant rules.

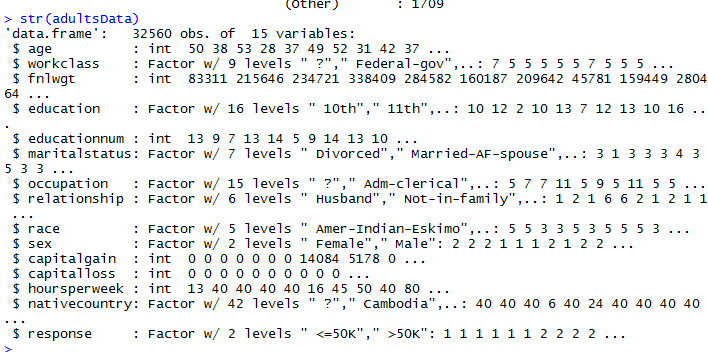


Fig 8: Attributes “naïve country” has about 42 levels of predictors, hence dropping the attribute.

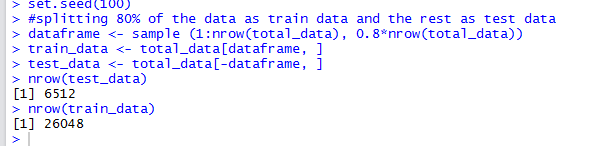


Fig 9 : Splitting the train and test data with 80:20 ratio

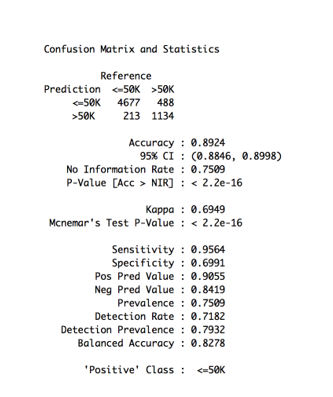


Fig 10 : Accuracy obtained for Random Forest Classifier

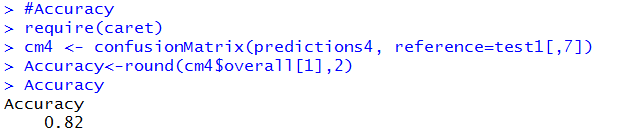


Fig 11 : Accuracy obtained for KNN classifier

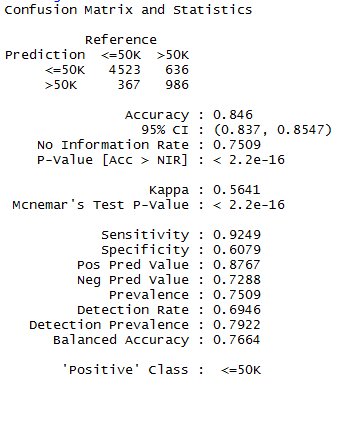


Fig 12 : Accuracy obtained by SVM.

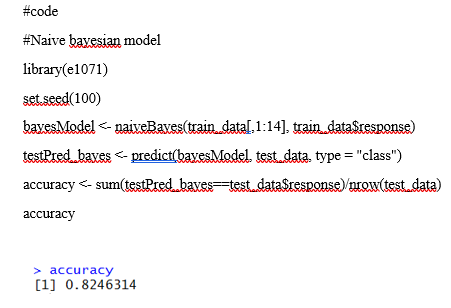


Fig 13: Accuracy obtained by Naïve Bayesian.

4.4 Performance

The performances of the models are as below according to their accuracy metrics.

|  |  |
| --- | --- |
| Model | Accuracy |
| Random Forest | 0.89 |
| Support Vector Machine | 0.86 |
| Naïve Bayesian | 0.79 |
| K nearest Neighbour | 0.82 |
| SimForest | ~0.99 |

Fig 14 : Comparison

From the above experimental results, we can see that Random Forest and SimForest perform equally. The basic condition behind SimForest is the one in which only the similarities between objects are present and the objects need not be multidimensional. In such cases, the random forests cannot be implemented. Instead, kernel support vector machine is implemented. Similarity matrices are noisy in most of the real-time cases. Thus, the errors occur naturally in the similarity. But, the SimForest is very much resistant towards noisy similarities when compared with other classifiers. An important note is in the presence of noisy similarity matrices and testing the effect of a specific level of noise. Hence, the approach is to utilize the multidimensional data to build the similarity matrices with added noise, however, do not to provide access to the multidimensional features for constructing the SimForest model. This can be accomplished by building two types of similarity matrices with respect to the cosine similarity and the Gaussian RBF similarity. Both the matrices are naturally used as kernels in support vector machines. Hence, the support vector machines approach is well suited to such types input similarity matrices.

5. CONCLUSION

Data Mining deals with extracting useful patterns from historical data. There are a lot of other data mining techniques and each of them have further more of them. It is difficult to say that one single technique can be suitable for all types of domains and the data concerning them. Sometimes, even hybrid techniques are proven to work better rather than using one single technique.

Applications of Data Mining

1. Market-Basket analysis is based on data mining
2. Prediction and analysis of useful patterns can be done by using various data mining techniques.
3. For the sake of telecommunication data analysis, visualization tools can be used.
4. intrusion detection can also be done by the usage of data mining.
5. Association and correlation analysis and also aggregation can be done in order to help the selection and to build discriminating attributes.

And many more

6 ACKNOWLEDGMENTS

The work on Similarity Forests was completely done and presented by Saket Sathe (IBM T.J. Watson Research Center) and Charu Aggarwal (IBM T.J. Watson Research Center) at the KDD 2017 conference.

7 REFERENCES

|  |  |
| --- | --- |
| [1] | Similarity Forests by Saket Sathe (IBM T.J. Watson Research Center) and Charu Aggarwal (IBM T.J. Watson Research Center) in the KDD 2017 conference: http://www.kdd.org/kdd2017/papers/view/similarity-forests |
| [2] | C. Aggarwal. Data Classification: Algorithms and Applications, CRC Press, 2014. |
| [3] | L. Brieman. Random Forests. Journal Machine Learning archive, 45(1), pp. 5–32, 2001 |
| [4] | M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? e-Journal of Machine Learning Research, 15(1), pp. 3133–3181, 2014 |
| [5] | A. Graf, A. Smola, and S. Borer. Classification in a normalized feature space using support vector machines. IEEE Transactions on Neural Networks, 14(3), 2003. |
| [6] | Ashutosh Tiwari, Chris Turner, A review of business process mining: State-of-art and future trends in Business Process Management Journal, February 2008 |
| [7] | https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/ |
| [8] | http://www.geeksforgeeks.org/k-nearest-neighbours/ |

[9] https://www.analyticsvidhya.com/blog/2014/06/introduction-random-forest-simplified/

[10] http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/

[11] E. Scornet. Random forests and kernel methods. IEEE Transactions on Information Theory, 62(3), pp. 1485-1500, 2016.

[12] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[13] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[14] http://dataaspirant.com/2016/12/23/k-nearest-neighbor-classifier-intro/

[15] http://www.scholarpedia.org/article/K-nearest\_neighbor

[16] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[17] https://en.wikipedia.org/wiki/Support\_vector\_machine

[18] http://dataaspirant.com/2017/01/13/support-vector-machine-algorithm/

[19] http://www.support-vector-machines.org/SVM\_soft.html

[20] <https://core.ac.uk/download/pdf/6302770.pdf>

[21] <http://www.iasri.res.in/ebook/expertsystem/datamining.pdf>

[22] <http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/>

[23] <https://machinelearningmastery.com/what-is-data-mining-and-kdd/>

[24] <https://cran.r-project.org/web/packages/arules/vignettes/arules.pdf>