Operation Research and Data Mining

Shouyi Wang, Yaju Fan, Wanpracha Chaovalitwongse

Abstract

Data mining (DM) and operations research (OR) are two largely independent paradigms of science. DM involves data driven methods aim to extract meaningful patterns from data instances whereas OR is based on model formulations and optimization algorithms to achieve optimal solutions for complex problems. DM and OR are also two overlapping disciplines. There is a growing interest to apply OR techniques to determine the underlying fitting structures of data samples in data mining problems; and many operations research problems have to include a data collection and analysis part to derive relevant variables in OR decision models. This paper provides a description of the most popular data mining techniques in use today in term of the basic methods and applications, and then present a review of how OR techniques can be applied to DM problems.

Keywords: data mining, operations research, optimization

1 Introduction

Data has become an essential part of today's world in the past decade, it is estimated that the amount of information in the world doubles every 20 months and the size and number of databases are increasing even faster. With this explosion in data and information, DM techniques have became ubiquitous with an explosion of interest from both academia and industry. The methods applied to extracting patterns from data have a long history for centuries, such as sorting data manually or using various hypothesis driven methods. However, it is extremely difficult to transform data into valuable knowledge by the traditional means of analysis in large volume of data in modern times. This motivates the development of modern DM methods, which are designed discover meaningful representations of data structure using pattern recognition, statistical and mathematical techniques. Generally, the analysis process of DM starts with a set of data based on which valuable knowledge is acquired. Once knowledge has been acquired this can scale up to be applied to very large databases under the assumption that they have a structure similar to the sample data.

Compared to DM, a quite young discipline, OR is a mature subject that may be traced to the early research of the optimization problems of transportation and mail system by the English mathematician Charles Babbage (1791-1871). The goal of OR is generally to achieve optimal solutions of some objective function to complex problems using mathematical modeling and optimization algorithms. Today, there are numerous well-developed OR techniques that have been used routinely to solve problems in a wide range of application areas. Basically, OR and DM are two independent fields with distinct objectives. OR tries to provide optimal solutions to a given target, while DM aims to discover unknown structures or relations to a given data set. However, the intersection between OR and DM is also quite broad. Each of them could benefit from making using of the other. For example, OR techniques could enhance the efficiency of a DM process by embedding various optimization tools, and DM could provide a concise and meaningful data space that may greatly facilitate an OR process. Recently, a growing interest in the integration of OR and DM can be observed in both of OR and DM literature. Of a particular interest, the focus of this paper is basically on the use of OR in DM. The rest of this paper is organized as follows, firstly the most popular and widely used DM methods are presented, and then the important applications of OR techniques in DM are discussed; finally the concluding remarks and discussion are given.

2 Data Mining Approaches

Data mining can be broadly categorized as either supervised or unsupervised learning. In supervised methods, the algorithm is provied with a set of training data whose class attributes are known. If class information of data are not available, the unsupervised techniques can be employed, such that clustering algorithms are designed to discover underlying groupings (or clusters) of data instances, and rule association rules aim to discover all associations and correlations among data items. In the following section, a number of most important algorithms from the two major classes of data mining will be described. Although there are many other algorithms and variations of the techniques described, the algorithms presented here are mostly basic ones that have been widely used in real world applications of DM and OR.

2.1 Supervised Learning Methods

Supervised learning approach constructs a predictive function from training data, which consists of a set of desired input-output pairs. Supervised learning algorithms first find a global mapping between inputs and their corresponding outputs to the highest possible extent, and then make predictions of future outputs to input values that it has never seen by their generalization capability. Generally, a good generalization of supervised learning requires a training data set that contains sufficiently large and representative of all cases so that a valid general mapping between outputs and inputs can be found. Supervised learning is one of the most frequently used data mining techniques, and a large number of supervised learning algorithms have been developed in the last decades. The most popular ones are discussed here.

Decision Tree: Decision tree is a hierarchical tree structure that is used to classify data classes based on attributes of data instances. In a decision tree, nodes represent classification attributes and branches represent conjunctions of attributes that lead to those classifications. Given a set of training data of attributes together with their associated classes, a decision tree can be induced in form of a sequence of rules that are used to recognize data classes. Once a decision tree is formed, it can be easily used to classify unseen data instances based on their attribute values starting at the root node.

The key to building a decision tree is to choose attributes in order to branch. The objective is to reduce impurity or uncertainty in data as much as possible. A subset of data is pure if all instances belong to the same class. There are several popular decision tree algorithms such as ID3, C4.5 and CART that perform well in tree regression. In general, the decision tree algorithms are recursive. The most well-know algorithm to generate decision trees is known as C4.5 [36]. C4.5 builds decision trees from a set of training data by using the concept of Shannon entropy [46], which is a measure of uncertainty associated with a random variable. Based on the fact that each attribute of data can be used to make a decision that splits the data into smaller subsets, C4.5 examines the relative entropy for each attribute, the attribute with the highest normalized information gain is used to make decisions. Ruggieri [40] provided an efficient version of C4.5, called EC4.5, which was claimed to be able to achieve a performance gain up to five times while compute the same decision trees as C4.5. Olcay and Onur [51] presented three parallel C4.5 algorithms which were designed to be applicable to large data sets. Baik and Bala [3] presented a distributed version of decision trees. In this agent-based approach, agents generate partial trees and communicate the temporary results among them in a collaborative way. The experimental results gave a very good performance of distributed decision trees for the data sets collected from distributed hosts.

Decision trees provide an effective method of decision making since they do not require any knowledge or parameter setting. One of the most useful characteristics of decision trees is that they are simple to understand and interpret. People can understand decision tree models after a brief explanation. The assumption made in the decision trees is that data instances belonging to different classes have different values in at least one of their features. Therefore, decision trees tend to perform better when dealing with discrete or categorical features.

Neural Network: Neural networks (NNs), inspired by the structure of biological neurons, are powerful tools that have been widely used to solve many problems of classification where there exists sufficient amount of observation data. Neural Networks have gained this popularity due to their powerful capacity to model extremely complex non linear functions and to their relatively easy use with well developed training algorithms. To train a NN, one first presents to the network with a set of training data with inputs and desired outputs, and then adjusts the weights of the NN in such a way that the error between the desired and actual outputs from the training data is minimized. The mean-squared error is the most commonly used error cost function, which is represented by:

$$E = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2, \tag{1}$$

where x_i, y_i are the input and output of training data, $f(x_i)$ is the output of NN with respect to the input x_i, and N is the number of training data.

To minimize the error cost function, one of the most widely used update rule is called perceptron learning rule, which was developed by Rosenblatt [39] in the 1950's. The basic perceptron update rule is given by:

$$\omega(j) = \omega(j) + \alpha(\delta - y)x(j), \tag{2}$$

where $x(j)$ denotes the jth item in input vector, $\omega(j)$ denotes the jth item in weight vector, y denotes the output from the neuron, δ is the expected output, α is a constant called learning rate that satisfies $0 < \alpha < 1$ and indicates the relative size of change in weights in every iteration.

Bayesian Learning: Bayesian decision theory is one of the most widely used statistical approaches to solve problems in machine learning. The basic idea of Bayesian learning is based on the estimation of probabilistic decisions [16], which can be described by the Bayes formula as follows:

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)} \tag{3}$$

where $P(H)$ is called prior probability of hypothesis H. It is the probability that H is correct before the data D was seen; $P(D|H)$ is called likelihood probability. It is the probability based on observation data D given that the hypothesis H is hold. $P(D)$ is the prior probability that the data will be observed. It is the probability of witnessing the data D under all possible hypotheses. The ratio $P(D|H)/P(D)$ is called irrelevance index. If the irrelevance index is 1, any knowledge about H is not relevant to D. Any value below 1 measures the relevancy between D and hypothesis H. $P(H|D)$ is the posterior probability. It is the probability that the hypothesis H is true, given the data and the previous state of belief about the hypothesis.

Compared to decision trees and NNs, Bayesian learning takes into account the probability of prior information. It is a supervised learning method that is studied from a probabilistic point of view. It start with a prior belief of an event and after observing the event the belief is revised to reflect the experience. The belief is represented as probabilities and the posterior probability is a product of the prior probability and likelihood. Bayesian learning has become popular in recent years largely due to this property which models the belief revision system of humans quite closely [44]. Bayesian learning approach has been widely used to solve many emerging problems in diverse domains (e.g., internet [31], epidemiology [32], robotics [48]). where *evidence* is a new observation of data instance, the *prior* reflects the current knowledge about an event before we have seen this newly observed data instance, the *likehood* is the conditional probability of seeing this new *evidence* given that the *prior* knowledge is true, and finally *posterior* is calculated by the above formula to reflect the updated knowledge about the event after we have observed the new *evidence* [16].

K-Nearest Neighbor (KNN): KNN is a supervised learning algorithm where the result of a new instance is classified based on the majority category of its K-nearest neighbor. It is a type of instance-based learning, the classifiers do not require model building or parameter estimation, and only based on the attributes of training samples. A data instance is classified by a majority vote of its K closest neighbors. The data instance is then assigned to the most common class amongst its K nearest neighbors. Any ties can be broken at random. If $K = 1$, then the instance is simply assigned to the class of its nearest neighbor.

The classification accuracy of KNN can be severely degraded by the presence of irrelevant features, or if the feature scales are not consistent with their importance. Therefore, the effects of feature standardization should be be performed and comparatively assessed before using KNN classification. The distance measure is also essential to kNN approaches. Using a distance measure that is appropriate for the data at hand is important. There are numerous distance measures, of which Euclidean distance is commonly used in KNN. One major problem of KNN is that the classes with more frequent samples tend to dominate the prediction of new instances, as they tend to come up in the K nearest neighbors due to their large populations. One way to overcome this problem is to take into account the distance of each K nearest neighbors with the new test data and predict the class of data instances based on these distances. Another disadvantage of KNN is computational expensive since we have to compute distances to all training examples. Therefore a key issue in much KNN research effort has been put into selecting or scaling features [52]. A good selection of features without redundant ones could improve classification accuracy and scale down computation time considerably.

Support Vector Machines: Support vector machine (SVM) is a widely used technique for classification and regression [14]. The key concept of SVM is to project input data instances into a higher dimensional space and divide the space with a continuous separation hyperplane while iteratively minimizing the distance of misclassified data instances from the hyperplane. In other words, SVM generally aimed at finding an optimal hyperplane that separates labeled data into two groups, say A and B. The optimal hyperplane can then be used for classifying new observations. The term “optimal” is used because a set of data of two groups may have many possible separating planes. SVM only finds one separating hyperplane that has the largest margin. The margin is defined as the minimum distance from the hyperplane to all other elements in each group. The resulting optimal hyperplane is intuitively reasonable. It is because the hyperplane has the longest distance to the data points in neighborhoods of both classes, and thus a good separation is achieved. There have been many variations of SVM models. One of the most successful models uses the idea that once a data set is transformed into a high dimensional space, which is called kernel transformation, every data instance can be classified by a separating plane if the new dimension is sufficiently high enough [10].

2.2 Unsupervised Learning Methods

Clustering: Unsupervised learning is inspired by brain's ability to extract statistical patterns and recognize complex visual scenes, sounds, and odors from sensory data. It takes root in neuroscience/psychology and is established on information theory and statistics. Unsupervised learning is used to reveal and capture unknown, but useful data groupings in a given data set autonomously. The goal of unsupervised learning is to build meaningful representations of data sets that can be used for decision making, prediction, and efficient communication. Unsupervised learning has wide applications in biology, medicine, market research, and robotics, etc.

Unsupervised learning, usually refers to clustering, deals with data that have not been pre-classified in any way, and does not need any type of supervision during the learning process. It is a learning paradigm which automatically assigns the received data into meaningful clusters based on their similarity. The similarity measures between two clusters drawn from the same feature space are essential to most unsupervised leaning algorithms. Because of the variety of similarity measures available, one must carefully choose the measures, which will highly influence the shape of clusters, as some elements may be close to one cluster according to one measure and further away according to another. There are many approaches to define similarity or distance between data instances, such as Euclidean distance, Manhattan distance, Hamming distance, Mahalanobis distance, and angular separation, etc [15]. The performance of clustering algorithms can be evaluated by the inter-relationships, namely, intra-connectivity and inter-connectivity. Intra-connectivity is a measure of the density of connections between the data instances within a single cluster. A higher intra-connectivity indicates a better clustering arrangement in a sense that the data instances in the same cluster are highly dependent on each other. Inter-connectivity is a measure of the connectivity between distinct clusters. A low degree of inter-connectivity is desirable since it indicates that the individual clusters are largely independent of each other.Clustering algorithms can be generally categorized into partitioning methods, hierarchical methods, and distributed clustering methods.

Partitioning Algorithm: The most well-known partitioning algorithm is k-means clustering. The goal of k-means clustering is to find k cluster centers that minimize a squared-error criterion function [16]. Cluster centers are represented by the gravity center of data instances; that is, the coordinates of a cluster center are the arithmetic mean for each dimension separately over all the data instances in the cluster. The k-means clustering assigns each instance to a cluster whose center is nearest to it. Since the k-means clustering generates partitions such that each pattern belongs to one and only one cluster, the obtained clusters are disjointed. In [17], a widely used clustering algorithm called fuzzy c-means (FCM) was developed to allow one data instance to belong to two or more clusters. Each data instance is associated with every cluster by a membership function, by which it has a degree of likelihood to every cluster, rather than just being assigned completely to one cluster. For example, the data instances on the edge of a cluster may be in the cluster to a lesser degree than the instances around the center of the cluster. FCM finds the most characteristic data instance in a cluster to be the ‘center’ of the cluster, and then assigns the grade of membership for each data instance in the cluster.

Other than gravity center, many clustering algorithms try to find clusters based on data density in a region. For a given radius, the neighborhood of each data instance of a cluster has to contain a minimum number of data instances. The most well known density-based clustering algorithms are Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [18], Generalized Density-Based Spatial Clustering of Applications with Noise (GDBSCAN) [42], Ordering Points To Identify the Clustering Structure (OPTICS) [2], Local Outlier Factors (LOF) [9], and Fast Density-Based Clustering (FDC)[55].

There are also some clustering algorithms that have been developed from Expectation-Maximization (EM) algorithms [30]. These EM based clustering algorithms first build a probability model to describe the probability that a data instance belongs to a certain cluster, then calculate the cluster probabilities for each data instance based on some initial guesses of model parameters. Subsequently, the obtained probabilities are in turn to verify the model parameters. The process is repeated to find the maximum likelihood estimates of the parameters in the probabilistic model. However, the two major drawbacks of this kind of algorithms are expensive computation and over-fitting.

Hierarchical Clustering: The clustering algorithms mentioned above all partition data instances directly in a single step. On the other hand, hierarchical clustering, which attempts to find successive clusters using previously established clusters, has also been extensively researched. Hierarchical clustering takes a series of partitions, which may separate the previous clusters successively into finer clusters, or proceed a series of fusions of current clusters into larger clusters. Some well developed hierarchical clustering algorithms are Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [54], Clustering Using Representatives (CURE) [21], a hierarchical clustering based on dynamic modeling called CHAMELEON [27], and an incremental hierarchical clustering algorithm called GRIN [12]. The major advantage of hierarchical clustering is that it does not require the number of clusters to be known in advance. However, these methods suffer from their inability to perform adjustments once splitting or merging decisions are made.

Distributed Clustering: Recently distributed clustering algorithms have attracted considerable attention to extracting knowledge from large databases [25], [1]. In many cases nowadays, the data are originally collected at different sites, and then brought together to extract information, these data sets are usually in huge size. Instead of being transmitted to a central site where we can analyze data by standard clustering algorithms, data can be clustered independently on different local sites. In a subsequent step, the central site tries to establish a global clustering based on the local clustering results. This approach is efficient, since local clusterings can be operated in a parallel way.

3 Operation Research in Data Mining

OR techniques can contribute to DM by developing better solution through various well-developed optimization tools. Many DM methods can incorporate optimization as a part of the DM problem or be directly formulated as an optimization problem. In the following, we will make a review of how OR techniques (such as linear programming, nonlinear optimization etc.) can significantly contribute to DM methods at three major stages: data preprocessing, DM modeling and results optimization, as shown in Figure

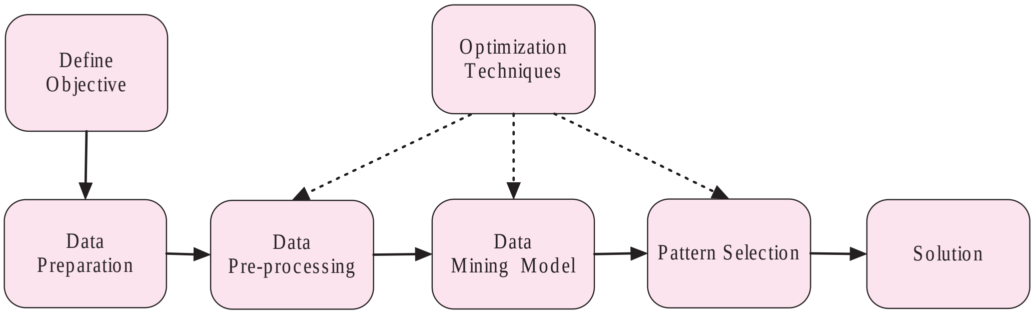

Figure 1: Block diagram of a data mining process model, and the use of OR techniques in DM at three major steps.

3.1 Data Preprocessing

The preprocessing step also refers to feature selection in DM, which aims to determine the optimal training data set from raw data before applied into a DM model. Optimization techniques can be quite useful in the dimensionality reduction by selecting optimal subset of features. A good selection of input features can significantly improve a DM algorithm in terms of modeling accuracy and fanning speed. The problem of feature selection can be formulated as a mathematical program with a parametric objective function and linear constraints. A good example of this field can be found in Chaovalitwongse et al [11] who used a binary integer programming formulation to solve the feature selection problem of massive electroencephalogram data. This formulation actually repents a more general optimization framework of feature selection which can be expressed as follows:

$$\begin{aligned} & \min f(X) \\ \text{s.t.} \quad & L \leq \sum x_i \leq U, \\ & x_i = 1 \text{ if feature i is selected, 0 otherwise,} \\ & x_i \in X \text{ is the feature vector.} \end{aligned} \tag{4}$$

where U and L are the upper and lower bound of the number of desired features, respectively.

Pendharkar [33] formulated the feature selection problem as a set of binary variable knapsack optimization framework. The optimization problem was solved by using a hybrid heuristic based on simulated annealing and gradient-descent artificial neural network. Yang et al. [49] provide a combinatorial formulation of feature selection problem. A metaheuristic called the nested partitions method was applied. Genetic algorithm (GA) was applied to their problem as a comparison. Their results shown that the metaheuristic method was effective for the feature selection problem, and was slightly better than the GA method. Janssens et al. [24] formulate the problem as a shortest path network based on the discretization of the raw continuous input data. They applied integer programming to select optimal boundary positions which minimize the misclassification costs.

Siedlecki et al. [35] applied GA to performed feature selection for a KNN classifier. Each feature was encoded as a candidate for GA chromosome, and a GA algorithm was applied to assign optimal weight to each feature. Prior applied to a KNN classification framework, the values of each feature are multiplied by normalized values of GA-identified weights. Their later works expand this weight assignment approach to a feature selection structure which was designed to select an ideal set of feature weights [35]. The predictive accuracy of the KNN classifier was considerably improve by searching for optimal feature weights or optimal feature set.

Another distinct OR method applied in DM problems is called Logical Analysis of Data (LAD), which tries to find minimal sets of features necessary for explaining the classification results. These combination of feature values form logical patterns that can be used for further classification and can be explained by human experts. The original LAD technique is used only for binary data proposed in [13]. To cope with numerical data, a binarizing method is proposed in [8, 7]. As a result, LAD deals with numerical data that have been transformed into binary values. The pattern characteristics found by LAD can be easily explained, and therefore it becomes a useful technique in practice, such as medical diagnosis [22]

3.2 DM Modeling

OR methods can be formulated as an important component of a DM model in many studies. In other words, optimization-based algorithms can directly contribute to the structure of a DM model and may generate new DM algorithms when combined with various optimization techniques. Actually, many data ming process are fundamentally optimization problems, such as SVM. Optimization techniques have been widely applied to the construction data mining in both supervised classification and unsupervised clustering models. For example, linear programming formulations are used to find a classification hyperplane in support vector machine (SVM) [29], nonlinear optimization with convex objective function and linear constrains is used in various classification/regression model [53], combinatorial optimization models are used in logical analysis of data (LAD) [13, 8, 7]. In the following, how optimization methods can be applied to DM modeling are discussed, some of the most recent developed techniques are reviewed.

3.2.1 Classification

Neural Networks: NNs are a group methods that have great ability of classification. The key part of NNs based methods is to choose an appropriate weight update rule. Rumelhart et al. [41] proposed a pioneer work to apply gradient descent based optimization method to update the weights of a multilayer NN. The this method was later on developed into the most well-known and widely used NNs update algorithm backpropagation (BP). BP employs error derivatives of NN's weights during training process. In other words, it calculates how error changes as each weight is increased or decreased slightly. The update rule of BP is given by:

$$\omega(j) = \omega(j) - \alpha \frac{\partial E}{\partial \omega(j)}, \tag{5}$$

where α is the learning rate, and $\frac{\partial E}{\partial \omega(j)}$ is the partial derivative of error cost function E with respect to weight $\omega(j)$. BP neural networks have become popular in practice since it can often find a good set of weights in a reasonable amount of time. There have been many successful applications of BP in science, engineering, finance and other disciplines.

However, since BP training is a gradient descending process, it is often trapped in a local minima and is very inefficient in searching for global minimum in a NN's weight space. In the recent years, genetic algorithm (GAs), as a powerful global optimization tool, has been growing rapidly in optimizing the weights of NN [50]. GAs have a potential to produce a global minimum in weight space and thereby avoid local minima. They are also very useful to the problems where gradient information is either not available or costly to obtain [47].

Simulated annealing techniques are also applied to gradient descent methods to move out the local optima. However, one of the problems is that the the search algorithm may get trapped into some cycles by doing so. For this problem, heuristic search optimization techniques can be quite useful. An excellent example is the use of tabu search introduced by Glover [20] in 1989. This method applies an iterative greedy search algorithm using memory of past points visited. The tabu search imposes a "tabu" on some subset of searching space so as to avoid making mistakes a second time. Battiti et al. [4] successfully applied the concept of tabu search to train the weights of NNs, their results demonstrated that the method was effective in continuing the search after local minima and the results were robust to random initial conditions.

Support Vector Machine: SVM is in principle an optimization based DM technique. The optimization formalism in SVM framework incorporates the concept of structural risk minimization by determining a separating hyperplane that maximizes not only a quantity measuring the misclassification error but also maximizing the margin separating the two classes. One can define a hyperplane with normal $\omega \in \mathbb{R}^d$ and express the plane as

$$x^{\mathsf{T}}\omega = \gamma,$$

where d is total number of features used to represent data cases, and $\gamma \in \mathbb{R}$ is a scalar. The objective is to decide values of (ω,γ), which reach the maximum margin. Denote the set A by the matrix $A \in \mathbb{R}^{m\times d}$ and the set B by the matrix $B \in \mathbb{R}^{k\times d}$. m and k are the number of cases which belong to groups A and B, respectively. The two sets, A and B, separately fall in two open half spaces. The set A lie in $\{x|x \in \Re^n, x^T\omega < \gamma\}$ and the set B lie in $\{x|x \in \Re^n, x^T\omega > \gamma\}$. Let e denote a vector of ones with arbitrary dimension. Then the following constraints must be satisfied:

$$A\omega > e\gamma, \quad B\omega < e\gamma.$$

Variables (ω,γ) can be rescaled to obtain non-strict inequalities because strict inequality constraints are not valid in linear programming formulations. To scale them, variables (ω,γ) can be divided by the positive value of $\min\limits_{i=1,\ldots,m,j=1,\ldots,k}\{A_i\omega-\gamma, -B_j\omega+\gamma\}$.

Without loss of generality, the equivalent inequalities can be written as

$$A\omega \geq e\gamma + e, \quad B\omega \leq e\gamma - e. \tag{6}$$

By this construction, the objective is to maximize the margin, $\frac{2}{\|\omega\|}$. In practice, most data sets are not perfectly separable. Hence, the assumption of having perfectly separable data sets for SVM is violated, and there exists no solution of (ω,γ) such that the inequalities (6) hold. For this reason, one tries to approximate the goal of maximizing margin by minimizing an average sum of violations. This leads to the development of robust linear programming formulation by Bennett and Mangasarian (1992) [5]. The model is given by

$$\begin{aligned} \min_{\omega,\gamma,y,z} \quad & \frac{e^Ty}{m} + \frac{e^Tz}{k} \\ \text{s.t.} \quad & A\omega - e\gamma - e \geq y, \\ & -B\omega + e\gamma - e \geq z, \\ & y \geq 0, z \geq 0. \end{aligned} \tag{7}$$

The variables y and z in the constraints of this problem satisfy the conditions:

$$y \geq \max\{0, -(A\omega - e\gamma - e)\}$$

and

$$z \geq \max\{0, -(B\omega + e\gamma - e)\}.$$

Hence, y and z are vectors containing violations of constraints (6). Minimizing the objective function of (8) leads to the minimum average violations.

The training optimization problem of SVM reaches a global minimum instead of a local minimum, which may happen in other algorithms such as NNs. SVMs have been applied to many real life problems including handwritten digit recognition [45], object recognition [6], speaker identification [43], face detection in images [32], and text categorization [26].

3.2.2 Clustering:

The goal of clustering is to group a given data set based on a measure of similarity. Optimization techniques also play an important role in various clustering methods, since the problem is fundamentally to find the optimal groupings or relations from a given data set. A comprehensive review of clustering and and optimization formulations can be found in Hansen et al. [23]. In particular, Rao [37] was one of the pioneers to apply linear and nonlinear programming formulations to solve clustering problems in a efficient way. Another excellent example of this area is given by Bradley et al. [34] who formulated a concave minimization problem for finding optimal clusters. Given m points in a n-dimensional Euclidean space R^n, a fixed number of cluster k, the centers of the cluster c is determined such that the sum of the distances of each point to a nearest cluster center is minimized. The general nonconvex optimization formulation can be expressed as follows:

$$\begin{aligned} \min_{C,D} \quad & \textstyle\sum_{i=1}^{m} \min e^T d_{il} \\ \text{s.t.} \quad & -d_{il} \le x_i - c_l \le d_{il}, \\ & i = 1, \ldots, m, \\ & l = 1, \ldots, k. \end{aligned} \tag{8}$$

where x_i is the i-th point out of m, c_l is the center of l-th cluster out of k, $d_{il} \in R^n$ is the dummy variable used to bound the components of the difference between point x_i and the center c_l, e is the unity vector. This general nonconvex problem was further reformulated into an optimization structure which minimizes a bilinear function using a k-median algorithm.

3.3 Result Optimization

Due to the large size of database, the results of many DM methods usually generate numerous patterns or models, which are still hard to interpret and applicable to solve target problems. In this perspective, OR techniques could play a valuable role in mining result optimization and interpretation. Optimization formulations can be built to select the best patters and models generated by a DM algorithm.

One good example comes from the research of Quinlan et al. [36] who proposed a method of selecting the best decision tree using the Minimum Description Length (MDL) principle [38]. A decision tree was first built using the standard DM algorithm C4.5, and the optimal subtree structure was selected by the MDL, which select a solution that minimizes the total number of bits needed to encode the tree and the description of the data given by the tree. The optimization formulation of MDL offered a way to prune the designed tree structure and thus avoid the problem of overfitting. Kennedy et al. [28] provide an example for applying GA to select the best decision tree generated by a decision tree induction model. In particular, they encoded decision trees as chromosomes, and the prediction accuracy was defined as the fitness function of GA. Fu et al [19] also applied GA to determine the best choice for multiple decision trees along the research line of Kennedy et al. [28], and a high prediction accuracy was reported according to their studies.

4 Conclusions

In this paper, we first provide an overview of the most popular DM algorithms and then pay a particular attention on the survey of how optimization techniques can be applied to solve DM problems. In general, optimization can significantly contribute to DM in three major steps:

- Optimal feature selection in data preprocessing step, which can significantly reduce the data dimensionality and search space for DM models.
- Construct optimization formulations directly in DM models, develop efficient optimization-based DM algorithms.
- Pick up the best patterns among a large number of candidates generated by DM models.

The use of optimization techniques can considerably improve the performance of DM in terms of efficiency, applicability and accuracy. We provide the most distinct examples of OR applications in DM at each step, and demonstrate how optimization frameworks can be formulated in DM structures. We believe that optimization techniques can play a critical role in providing better and faster solutions to DM problems. As a commentary part, DM methods can also contribute to OR problems in building efficient decision models by providing concise and meaningful representations for a given data set. It is noted that this area is relatively less studied compared to the numerous applications of OR in DM. The integration of OR and DM in a closely complementary manner constitutes an future field of research for OR and DM researchers.

References

[1] Panagiotis D. Alevizos, Dimitris K. Tasoulis, and Michael N. Vrahatis. Parallelizing the unsupervised k-windows clustering algorithm. In *Lecture Notes in Computer Science*, pages 225–232. Springer-Verlag, 2004.

[2] M. Ankerst, M. Breunig, H. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Philadelphia PA, 1999.

[3] S. Baik and J. Bala. A decision tree algorithm for distributed data mining: Towards network intrusion detection. In *Proceedings of International Conference of Computational Science and Its Applications*, volume 3046, pages 206–212, 2004.

[4] R. Battiti and G. Tecchiolli. Training neural nets with the reactive tabu search. *IEEE Transactions on Neural Networks*, 6(5).

[5] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[6] V. Blanz, B. Scholkopf, H. Bulthoff, C. Burges, V. Vapnik, and T. Vetter. Comparison of view-based object recognition algorithms using realistic 3d models. In *Lecture Notes in Computer Science*, volume 1112, pages 251–256. Springer, 1996.

[7] E. Bores, P.L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik. An implementation of logical analysis of data. *Knowledge and Data Engineering, IEEE Transactions on*, 12(2):292–306, Mar/Apr 2000.

[8] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, and Alexander Kogan. Logical analysis of numerical data. *Math. Program.*, 79(1-3):163–190, 1997.

[9] M. Breuning, H. Kriegel, R.T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2000.

[10] C. Burges. Tutorial on support vector machines for pattern reccognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[11] W. Chaovalitwongse, Y.J. Fan, and R.C. Sachdeo. Novel optimization models for abnormal brain activity classification. *Operations Research*, 56(6), 2008.

[12] C.Y. Chen, S.C. Hwang, and Y.J. Oyang. An incremental hierarchical data clustering algorithm based on gravity theory gravity theory. In *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference*, volume 2336, pages 237–250, May 2002.

[13] Yves Crama, Peter L. Hammer, and Toshihide Ibaraki. Cause-effect relationships and partially defined boolean functions. *Annals of Operations Research*, 16(1):299–325, December 1988.

[14] N. Cristianini and J.S. Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, 2000.

[15] E. Deza and M.M. Deza. *Dictionary of Distances*. Elsevier, 2006.

[16] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2 edition, 2001.

[17] J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973.

[18] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial data sets with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, 1996.

[19] Z. Fu, B. Golden, S. Lele, S. Raghavan, and E. Wasil. A genetic algorithm-based approach for building accurate decision trees. *INFORMS Journal Computing*, 15(1), 2003.

[20] E. Glover. Tabu search. *Part I, ORSA Journal on Computing*, 1:190–206, 1989.

[21] S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large data sets. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 73–84, Seattle, Washington, USA, Jun. 1998.

[22] Peter L. Hammer and Tiberius O. Bonates. Logical analysis of data - an overview: From combinatorial optimization to medical applications. *Annals of Operations Research*, 148(1):203–225, November 2006.

[23] Pierre Hansen, , and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 71:191–215, 1997.

[24] D. Janssens, T. Brijs andK. Vanhoof, and G. Wets. Evaluating the performance of cost-based discretization versus entropyand error-based discretization. *Computers & Operations Research*, 33(11):3107–3123, 2006.

[25] E. Januzaj, H.P. Kriegel, and M. Pfeifle. Towards effective and efficient distributed clustering. In *In Workshop on Clustering Large Data Sets (ICDM)*, pages 49–58, 2003.

[26] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of 10th European Conference on Machine Learning*, volume 1398, pages 137–142, Chemnitz, Germany, Apr. 1998.

[27] G. Karypis, E.H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *Computer*, 32(8):68–75, 1999.

[28] H. Kennedy, C. Chinniah, P. Bradbeer, and L. Morss. The construction and evaluation of decision trees: A comparison of evolutionary and concept learning methods. *Evolutionary Computing*, 1305:147–161, 1997.

[29] O. L. Mangasarian. Data mining via support vector machines. In *IFIP Conference on System Modelling and Optimization*, pages 23–27. Kluwer Academic Publishers, 2001.

[30] G.J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 1997.

[31] P. Myllymaki, T. Silander, H. Tirri, and P. Uronen. Bayesian data mining on the web with b-course. In *Proceedings of the 1st IEEE International Conference on Data Mining*, pages 626–629, 2001.

[32] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 130–136, 1997.

[33] P. Pendharkar. A data mining-constraint satisfaction optimization problem for cost effective calssification. *Computers & Operations Research*, 33(11):3124C3135, 2006.

[34] O.L. Mangasarian P.S. Bradley and W.N. Street. Clustering via concave minimization. *Advances in Neural Information Processing Systems*, 9:368–374, 1997.

[35] W. F. Punch, E. D. Goodman, M. Pei, L. Chia-Shun, P. Hovland, and R. Enbody. Further research on feature selection and classification using genetic algorithms. *In Preecdings of International Conference on Gentic Algorithms*, 93:557–564, 1993.

[36] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA, 1993.

[37] M. Rao. Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, 66:622–626, 1971.

[38] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[39] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, Nov. 1958.

[40] S. Ruggieri. Efficient c4.5. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):438–444, 2001.

[41] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. *Parallel Data Processing*, 1:318–362, 1986.

[42] J.O. Sander, M. Ester, H.P. Kriegel, and X. Xu. Density-based clustering in spatial data sets: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2:169–194, 1998.

[43] M. Schmidt. Identifying speaker with support vector networks. In *Proceedings of Interface*, Sydney, 1996.

[44] S. Schocken and P.R. Kleindorfer. Artificial intelligence dialects of the bayesian belief revision language. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1106–1121, 1989.

[45] B. Scholkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, pages 252–257. AAAI Press, 1995.

[46] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, Jul. 1948.

[47] M.N.H. Siddique and M.O. Tokhi. Training neural networks: Backpropagation vs. genetic algorithms. In *IEEE International Joint Conference on Neural Networks*, volume 4, pages 2673–2678, 2001.

[48] J. Ting, A. D'Souza, S. Vijayakumar, and S. Schaal. A bayesian approach to empirical local linearization for robotics. In *International Conference on Robotics and Automation*, May 2008.

[49] Jaekyung Yang and Sigurdur Olafsson. Optimization-based feature selection with adaptive instance sampling. *Computers & Operations Research*, 33(11):3088–3106, 2006.

[50] G.G. Yen and H. Lu. Hierarchical genetic algorithm based neural network design. In *IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, pages 168–175, 2000.

[51] O.T. Yildiz and O. Dikmen. Parallel univariate decision trees. *Pattern Recognition Letters*, 28:825–832, May 2007.

[52] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.

[53] Hao Zhang, Hao Helen Zhang, Grace Wahba, Grace Wahba, Yi Lin, Yi Lin, Meta Voelker, Meta Voelker, Michael Ferris, Michael Ferris, Ronald Klein, Ronald Klein, Barbara Klein, and Barbara Klein. Variable selection and model building via likelihood basis pursuit. *Journal of the American Statistical Association*, 99(467):659–672, 2004.

[54] T. Zhang, R. Ramakrishnan, and M. Linvy. Birch: An efficient data clustering method for very large data sets. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.

[55] B. Zhou, D.W. Cheung, and B. Kao. A fast algorithm for density-based clustering in large database. In *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, pages 338–349, 1999.