Mathematical Programming for Data Mining

IOANNIS P. ANDROULAKIS[1], W. ART CHAOVALITWONGSE[2]
[1] Department of Biomedical Engineering, Rutgers University, Piscataway, NJ, USA
[2] Department of Industrial and Systems Engineering, Rutgers University, Piscataway, NJ, USA

MSC2000: TS2

Article Outline

Keywords

Mathematical programming; Data mining; Optimization; Clustering; Classification

Introduction

Progress in digital data acquisition and storage technology has resulted in the growth of huge databases. This has occurred in a variety of scientific and engineering research applications [8] as well as medical domain [19,20]. Making sense out of these rapidly growing massive data sets gave birth to a "new" scientific discipline often referred to as *Data Mining*. Defining a discipline is, however, always a controversial task. The following working definition of the area was recently proposed [9]: Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.

Clearly the term data mining if often used as a synonym for the process of extracting useful information from databases. However, the overall knowledge discovery from databases (KDD) process is far more complicated and convoluted and involves a number of additional pre and post-processing steps [6]. Therefore, in our definition data mining refers to the ensemble of new, and existing, specific algorithms for extracting structure from data [8]. The exact definition of the knowledge extraction process and the expected outcomes are very difficult to characterize. However, a number of specific tasks can be identified and, by and large, define the key subset of deliverables from a data mining activity. Two such critical activities are classification and clustering. A number of variants for these tasks can be identified and, furthermore, the specific structure of the data involved greatly impacts the methods and algorithms that are to be employed. Before we proceed with the exact definition of the tasks we need to provide working definitions of the nature and structure of the data.

Basic Definitions

For the purposes of our analysis we will assume that the data are expressed in the form of n-dimensional feature vectors $x \in X \subseteq \Re^n$. Appropriate pre-processing of the data may be required to transform the data into this form. Although in many cases this transformations can be trivial, in other cases transforming the data into a "workable" form is a highly non-trivial task. The goal of data mining is to estimate an explicit, or implicit, function that maps points of the feature vector from the input space, $X \subseteq \Re^n$, to an output space, C, given a finite sample. The concept of the finite sample is important because, in general, what we are given is a finite representative subset of the original space (training set) and we wish to make predictions on new elements of the set (testing set). The data mining tasks can thus de defined based on the nature of the mapping C and the extent to which the train set is characterized. If the predicted quantity is a categorical value and if we know the value that corresponds to each elements of the training set then the question becomes how to identify the mapping that connects the feature vector and the corresponding categorical value (class). This problem is known as the classification problem (supervised learning). If the class assignment is not known and we seek to: (*a*) identify whether a small, yet unknown, number

TS1 Please note that pagination and layout are not final.
TS2 Please provide the MSC.

Editor's or typesetter's annotations (will be removed before the final TEX run) of classes exist; (*b*) define the mapping assigning the features to classes then we have a clustering problem (unsupervised learning).

A related problem associated with superfluous information in the feature vector is the so-called feature selection problem. This is a problem closely related to over-fitting in regression. Having a minimal number of features leads to simpler models, better generalization and easier interpretation. One of the fundamental issues in data mining is therefore to identify the least number of features, sub-set of the original set of features, that best address the two issues previously defined. The concept of parsimony (Occam's razor) is often invoked to bias the search [1]: never do with more what can be done with fewer.

Although numerous methods exist for addressing these problems they will not be reviewed here. Nice reviews of classification and were recently presented in [8,9]. In this short introduction we will concentrate on solution methodologies based on reformulating the clustering, and classification questions as optimization problems.

Mathematical Programming Formulations

Classification and clustering, and for that matter most of the data mining tasks, are fundamentally optimization problems. Mathematical programming methodologies formalize the problem definition and make use of recent advances in optimization theory and applications for the efficient solution of the corresponding formulations. In fact, mathematical programming approaches, particularly linear programming, have long been used in data mining tasks.

The pioneering work presented in [13,14] demonstrated how to formulate the problem of constructing planes to separate linearly separable sets of points.

In this summary we will follow the formalism put forth in [2] since it presented one of the most comprehensive approaches to this problem. One of the major advantages of a formulation based on mathematical programming is the ease in incorporating explicit problem specific constraints. This will be discussed in greater detail later in this summary.

Classification

As discussed earlier the main goal in classification is to predict a categorical variable (class) based on the values of the feature vector. The general families of methods for addressing this problem include [9]:

i) Estimation of the conditional probability of observing class C given the feature vector x.
ii) Analysis of various proximity metrics and based the decision of class assignment based on proximity.
iii) Recursive input space partitioning to maximize a score of class purity (tree-based methods).

The two-class classification problem can be formulated as the search of a function that assigns a given input vector x into two disjoint point sets A and B. The data are represented in the form of matrices. Assuming that the set A has m elements and the set B has k elements, then $A \in \Re^{m \times n}$, $B \in \Re^{k \times n}$, describe the two sets respectively. The discrimination in based on the derivation of hyperplane

$$P = \{x | x \in \Re^n, x^T \omega = \gamma\}$$

with normal and distance from the origin $\frac{|\gamma|}{\|\omega\|_2}$. The optimization problem then becomes to determine ω and γ such that the separating hyperplane P defines two open half spaces

$$\begin{aligned}&\{x | x \in \Re^n, x^T \omega < \gamma\}\\&\{x | x \in \Re^n, x^T \omega > \gamma\}\end{aligned}$$

containing mostly points in A and B respectively. Unless A and B are disjoint the separation can only be satisfied within some error. Minimization of the average violations provides a possible approximation of the separating hyperplane [2]:

$$\min_{\omega,\gamma} \frac{1}{m} \|(-A\omega + e\gamma + e)_+\|_1 + \frac{1}{k} \|(-B\omega + e\gamma + e)_+\|_1$$

In [2] a number of linear programming reformulations are discussed exploring the properties of the structure of the optimization problem. In particular an effective robust linear programming (RLP) reformulation was suggested making possible the solution of large-scale problems:

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

In [17] it was demonstrated how the above formulation can be applied repeatedly to produce complex space partitions similar to those obtained by the application of standard decision tree methods such as C4.5 [21] or CART [4].

Clustering

The goal of clustering is the segmentation of the raw data into groups that share a common, yet unknown, characteristic property. Similarity is therefore a key property in any clustering task. The difficulty arises from the fact that the process is unsupervised. That is neither the property nor the expected number of groups (clusters) are known ahead of time. The search for the optimal number of clusters is parametric in nature and the optimal point in an "error" vs. "number of clusters" curve is usually identified by a combined objective the weighs appropriately accuracy and number of clusters. Conceptually a number of approaches can be developed for addressing clustering problems:

i) Distance-based methods, by far the most commonly used, that attempt to identify the best k-way partition of the data by minimizing the distance of the points assigned to cluster k from the center of the cluster.

ii) Model-based methods assume the functional form of a model that describes each of the clusters and then search for the best parameter fit that models each cluster by minimizing some appropriate likelihood measure.

There are two different types of clustering: (1) hard clustering; (2) fuzzy clustering. The former assigns a data point to *exactly* one cluster while the latter assigns a data point to one of more clusters along with the likelihood of the data point belonging to one of those clusters.

The standard formulation of the hard clustering problem is:

$$\min_{c} \sum_{i=1}^{m} \min_{l} \|x^i - c^l\|_n$$

That is given m points, x, in an n-dimensional space, and a fixed number of cluster, k, determine the centers of the cluster, c, such that the sum of the distances of each point to a nearest cluster center is minimized. It was shown in [3] that this general non convex problem can be reformulated such that we minimize a bilinear functions over a polyhedral set by introducing a selection variable t_{il}:

$$\begin{aligned}
&\min_{c,d,t} \sum_{i=1}^{m} \sum_{i=1}^{k} t_{il}(e^T d_{il}) \\
&\text{s.t.} - d_{il} \le x^i - c^l \le d_{il} \\
&\qquad \sum_{l=1}^{k} t_{il} = 1 \\
&\qquad t_{il} \ge 0 \\
&\qquad i = 1, \ldots, m, l = 1, \ldots, k.
\end{aligned}$$

d is a dummy variable used to bound the components of the difference $x - c$. In the above formulation the 1-norm is selected [3].

The fuzzy clustering problem can be formulated as follows [5]:

$$\begin{aligned}
&\min_{w} \sum_{i=1}^{m} \sum_{l=1}^{k} w_{il}^2 \|x^i - c^l\|^2 \\
&\text{s.t.} \sum_{l=1}^{k} w_{il} = 1 \\
&\qquad w_{il} \ge 1,
\end{aligned}$$

where $x^i, i = 1, \ldots, m$ is the location descriptor for the data point, $c^l, l = 1, \ldots, k$ is the center of the cluster, w_{il} is the likelihood of a data point i being assigned to cluster l.

Support Vector Machines

This optimization formalism bares significance resemblance to the Support Vector Machines (SVM) framework [25]. SVM incorporate 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. This can be achieved by augmenting the objective of the RLP formulation earlier presented by an appropriately weighted measure of the separation between the two classes as $(1 - \lambda)(e^T y + e^T z) + \frac{\lambda}{2}\|\omega\|_2^2$.

In [6] the concept of SVM is extended by introducing the Proximal support vector machines which classify points based on proximity to one of two parallel planes that are pushed as far apart as possible. Nonlinear transformations were also introduced in [6] to enable the derivation of non-linear boundaries in classifiers.

Multi-Class Support Vector Machines

Support vector machines were originally designed for binary classification. Extending to multi-class problems is still an open research area [10].
The earliest multi-class implementation is the *one against all* [22] by constructing k SVM models, where k is the number of classes. The ith SVM is classifies the examples of class i against all the other samples in all other classes. Another alternative builds *one against one* [12] classifiers by building $\frac{k(k-1)}{2}$ models where each is trained on data from two classes. The emphasis of current research is on novel methods for generating all the decision functions through the solution of a single, but much larger, optimization problem [10].

Data Mining in the Presence of Constraints

Prior knowledge about a system is often omitted in data mining applications because most algorithms do not have adequate provisions for incorporating explicitly such types of constrains. Prior knowledge can either encodes explicit and/or implicit relations among the features or models the existence of "obstacles" in the feature space [24].
One of the major advantages of a mathematical programming framework for performing data mining tasks is that prior knowledge can be incorporated in the definition of the various tasks in the form of (non)linear constraints. Efficient incorporation of prior knowledge in the form of nonlinear inequalities within the SVM framework was recently proposed by [15]. Reformulations of the original linear and nonlinear SVM classifiers to accommodate prior knowledge about the problem were presented in [7] in the context of approximation and in [16] in the context of classifiers.

Data Mining and Integer Optimization

Data mining tasks involve, fundamentally, discrete decisions:

- How many clusters are there?
- Which class does a record belong to?
- Which features are most informative?
- Which samples capture the essential information?

Implicit enumeration techniques such as branch-and-bound were used early on to address the problem of feature selection [18].
Mathematical programming inspired by algorithms for addressing various data mining problems are now being revisited and cast as integer optimization problems. Representative formulations include feature selection using Mixed-Integer Linear Programs [11] and in [23] integer optimization models are used to address the problem of classification and regression.

Research Challenges

Numerous issues can of course be raised. However, we would like to focus on three critical aspects

i) Scalability and the curse of dimensionality. Databases are growing extremely fast and problems of practical interest are routinely composed of millions of records and thousands of features. The computational complexity is therefore expected to grow beyond what is currently reasonable and tractable. Hardware advances alone will not address this problem either as the increase in computational complexity outgrows the increase in computational speed. The challenge is therefore two-fold: either improve the algorithms and the implementation of the algorithms or explore sampling and dimensionality reduction techniques.

ii) Noise and infrequent events. Noise and uncertainty in the data is a given. Therefore, data mining algorithms in general and mathematical programming formulations in particular have to account for the presence of noise. Issues from robustness and uncertainty propagation have to be incorporated. However, an interesting issue emerges: how do we distinguish between noise and an infrequent, albeit interesting observation? This in fact maybe a question with no answer.

iii) Interpretation and visualization. The ultimate goal of data mining is understanding the data and developing actionable strategies based on the conclusions. We need to improve not only the interpretation of the derived models but also the knowledge delivery methods based on the derived models. Optimization and mathematical programming needs to provide not just the optimal solution but also some way of interpreting the implications of a particular solution including the quantification of potential crucial sensitivities.

References

1. Blumer A, Ehrenfeucht A, Haussler D, Warmuth MK (1987) Occam's razor. Inf Process Lett 24:377–380
2. Bradley PS, Fayyad U, Mangasarian OL (1999) Mathematical programming for data mining: Formulations and challenges. INFORMS J Comput 11:217–238
3. Bradley PS, Mangasarian OL, Street WN (1997) Clustering via concave minimization. In: Mozer MC, Jordan MI, Petsche T (eds) Advances in Neural Information Processing Systems, MIT Press, pp 368–374 CE3
4. Breiman L, Friedman J, Olsen R, Stone C (1993) Classification and Regression Trees. Wadsworth Inc., CE3
5. Dunn JC (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J Cybern 3:32–57
6. Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery: An overview In: Fayyad UM, Piatetsky-Shapiro G, Smyth P, Uthursamy R (eds) Advances in knowledge discovery and data mining. AAAI Press, pp 229–248 CE3
7. Fung GM, Mangasarian OL (2001) Proximal Support Vector Machine Classifiers In: Provost F, Srikant R (eds) Proceedings KDD-2001: Knowledge Discovery and Data Mining, San Francisco, August 26–29 2001, Asscociation for Computing Machinery, pp 77–86, ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-02.ps CE4
8. Grossman RL, Kamath C, Kegelmeyer P, Kumar V, Namburu EE (2001) Data mining for scientific and engineering applications. Kluwer Academic Publishers, CE3
9. Hand DJ, Mannila H, Smyth P (2001) Principle of data mining. Bradford Books, CE3
10. Hsu C-W, Lin C-J (2002) A comparison of methods multiclass support vector machines. IEEE Trans Neural Netw 13:415–425
11. Iannatilli FJ, Rubin PA (2003) Feature selection for multiclass discrimination via mized-integer linear programming. IEEE Trans Pattern Anal Mach Learn 25:779–783
12. Krebel U (1999) Pairwise classification and support vector machines: Advances in Kernel Methods – Support Vector Learning. MIT Press CE3
13. Mangasarian OL (1965) Linear and nonlinear separation of pattern by linear programming. Oper Res 31:445–453
14. Mangasarian OL (1968) Multisurface method for pattern separation. IEEE Trans Inf Theory IT-14:801–807
15. Mangasarian OL, Shavlik JW, Wild EW (2003) Knowledge-based kernel approximations. Tech. rep., Data Mining Institute, University of Wisconsin CE3
16. Mangasarian OL, Shavlik JW, Wild EW (2004) Knowledge-based kernel approximation. J Mach Learn Res 5:1127–1141
17. Mangasarian OL, Street WN, Wolberg WH (1995) Breast cancer diagnosis and prognosis via Linear Programming. Oper Res 43:570–577
18. Narendra P, Fukunaga K (1977) A branch and bound algorithm for feature subset selection. IEEE Trans Comput 26:917–922
19. Pardalos PM, Boginski V, Vazakopoulos A (2007) Data Mining in Biomedicine. Springer CE3
20. Pardalos PM, Principe J (2002) Biocomputing, Kluwer Academic Publishers CE3
21. Quinlan JR (1993) C4.5: Programs for Machine Learning. Morgan Kaufmann CE3
22. Scholkopf B, Burges C, Vapnik V (1995) Extracting support data for a given task: Proc. First International Conference on Knowledge Discovery and Data Mining. AAAI Press, pp 252–257 CE3
23. Shioda R (2003) Integer Optimization in Data Mining. PhD thesis, MIT CE3
24. Tung AK, Hou J, Han J (2001) Spatial clustering in the presence of obstacles In: Proceedings ICDE-2001: 17th International Conference on Data Engineering, pp 359–367 CE3
25. Vapnik VN (1995) The nature of statistical learning, Springer CE3

CE3 Please provide publisher location.
CE4 Please provide access date.

Editor's or typesetter's annotations (will be removed before the final TEX run)