Discrete Optimization 13 (2014) 36–48

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

A combinatorial branch-and-bound algorithm for box search

Q. Louveaux, S. Mathieu *

Institut Montefiore (B28), University of Liège, Grande Traverse 10, B-4000 Liège, Belgium

ARTICLE INFO

Article history:
Received 19 July 2013
Received in revised form 29 January 2014
Accepted 19 May 2014
Available online 2 June 2014

MSC:
90C27
90C57
90C11
97R50

Keywords:
Combinatorial optimization
Branch-and-bound
Data mining
Mixed integer programming

ABSTRACT

Considering a set of points in a multi-dimensional space with an associated real value for each point, we want to find the box with the maximum sum of the values of the included points. This problem has applications in data mining and can be formulated as a mixed-integer linear program. We propose a branch-and-bound algorithm where the bounding is obtained by combinatorial arguments instead of the traditional linear relaxation. Computational experiments show that this approach competes with current state of the art mixed-integer solvers. The algorithm proposed in this paper may be seen as a simple and dependence-free method to solve the box search problem.

1. Introduction

Data mining [1] has been a popular field of research recently since it is possible to generate big databases. It consists in finding relevant information in a large set of data that is usually automatically generated. The most well-known example is when a company collects information about its customers in order to provide a better individualized service. Together with this new trend, companies also try to automatically collect as much data as possible. An important question is therefore to understand how to handle this new information correctly. In this paper, we are interested in a particular problem that arises in this context and that may be of interest for very complex industrial processes. In particular, we consider a process for which we automatically retrieve a number of variables that might influence a given output variable. For example, we may consider a line creating some products and the output variable could be an estimation of the quality of the product. It is clear that a large number of variables may influence the quality of the production and we assume that most of them can be retrieved along the way for each produced item. The question that we ask is to find a set of rules, i.e. intervals on the variables, for which the average output is maximized. This can be of interest in order to find a setup that works well in average. On the other hand, if we find a set of rules for which the average output is minimized, this could also potentially explain why a number of objects is of bad quality. Both questions are of interest in the context of understanding an industrial process and we try to address them in this paper.

More specifically, we assume that each produced item i is represented by a D-dimensional vector x^i of the input variables for which we consider a normalized value in [0, 1]. The output value of the item i is given by $c^i \in \mathbb{R}$ and might be either

* Corresponding author. Tel.: +32 4 366 4814.
E-mail addresses: q.louveaux@ulg.ac.be (Q. Louveaux), sebastien.mathieu@ulg.ac.be (S. Mathieu).

http://dx.doi.org/10.1016/j.disopt.2014.05.001
1572-5286/

positive or negative. The problem addressed in this paper is to find a box, i.e. an interval $[l_t, u_t]$ for each dimension t, which maximizes the sum of the values c^i of each point i included in the box. We name this problem the *box search*. The purpose of a box is its simplicity and robustness.

A way to handle data mining is by using machine learning strategies. In [2], Friedman et al. introduce the heuristic PRIM (Patient Rule Induction Method) for bump-hunting in high-dimensional data. The procedure starts with a box including all points. At each iteration, PRIM peels α points of the box by restricting the box on one dimension. The procedure stops when the number of points included in the box drops below a fraction β of the total number of points. Several improvements of this technique have been proposed as PRIM2 [3] or f-PRIM [4]. Other alternatives in this area are suggested like subgroup discovery [5] or a genetic algorithm [6].

A common drawback of these approaches is that they give no guarantee on the quality of the solution. One may however require finding the best box and provide a certificate of optimality. Unfortunately, this kind of problem is known to be NP-Hard. It can be trivially solved in $M^{O(D)}$ operations where M is the number of points and D their dimension [7]. Eckstein et al. introduce in [8] the *maximum box problem* where we seek a box maximizing the weighted sum over a set of points with a positive objective value while not intersecting a set of given points. They prove that the *maximum box problem* is NP-Hard in the general case and polynomial if we fix the dimension. The maximum box problem trivially reduces to the box search which implies that it is NP-hard as well. This polynomial complexity is particularly interesting for the two-dimensional case where algorithms of low complexity can be implemented [9,10].

A classic approach to solve NP-Hard problems to global optimality is the branch-and-bound algorithm [11]. It has been applied successfully in [12] to find logical patterns where the branching decision is made on the inclusion or the exclusion of one logical element. A combinatorial branch-and-bound algorithm has been proposed to solve the maximum box problem where every child is created by dividing the space along each dimension to exclude a point which must not be intersected by the box [8]. However the algorithm proposed in [8] creates 2^D branches at each node in the worst case which may be prohibitive for high dimension. The main difference between the box search and [8] is that we do not restrict ourselves to homogeneous boxes of positive points. Whereas in [8], the goal is to find the box with a maximum number of points with a positive value without any point from an excluded set, we find the box with the maximum weighted sum of points with any value, positive or negative.

We show that the computation of the box with the maximum sum can be formulated as a mixed-integer linear program. We propose a combinatorial branch-and-bound approach to tackle the problem. The branching decision is not made on the variables of the linear model but on a decision of including or excluding a point from a candidate box. The bounding is obtained by combinatorial arguments instead of the traditional linear relaxation. We investigate common branching strategies such as strong branching [13,14] and reliability branching [15] as well as one problem specific strategy. Computational experiments show that this approach compete with state of the art mixed-integer linear programming (MILP) softwares for this particular problem.

The outline is as follows. Section 2 proposes a MILP formulation of the box search problem. We present our combinatorial branch-and-bound algorithm in Section 3. We compare the performance of the proposed algorithm with a standard MILP software in Section 4 and discuss the performance of the proposed algorithm.

2. Integer programming models

In this section we formulate the box search problem as an integer program. We propose two formulations for the problem. In the first formulation, we propose to use the bounds of the box as continuous variables of the problem. The second formulation is purely binary.

We explore a D-dimensional space with M points normalized with linear scaling transform [16]. The coordinates of a point i are denoted $\mathbf{x}^i$ with $\mathbf{x}^i \in [0, 1]^D$. The value of a point i is $c^i \in \mathbb{R}$. Those points are partitioned in two sets: the set of positive points $\mathcal{P} = \{i \in \{1, \ldots, M\} | c^i > 0\}$ and the set of negative points $\mathcal{N} = \{i \in \{1, \ldots, M\} | c^i < 0\}$. Points such that $c^i = 0$ can be ignored.

To improve the robustness of the models, we apply the following transformation to the coordinates of the points. For each dimension $t \in \{0, \ldots, D\}$, the points are sorted with respect to their coordinate $\{x_t^i, \ \forall i \in \{1, \ldots, M\}\}$. Due to some points having the same coordinates on dimension t, we observe in this sorted list K_t distinct values, $K_t \leq M$. We replace the coordinate of the point i, x_t^i by its order in the sorted list of points divided by the number of elements K_t. The index of i in this sorted list is denoted r_t^i.

2.1. Continuous box bounds

In this formulation, a box is defined by its two opposite corners: $\mathbf{l} \in [0, 1]^D$ and $\mathbf{u} \in [0, 1]^D$. We introduce one binary variable z^i per point where $z^i = 1$ if the point i is included in the box and 0 otherwise. The box search can be formulated as the following optimization problem:

$$\max f = \sum_{i=1}^{M} c^i z^i \tag{1}$$