European Journal of Operational Research 206 (2010) 1–10

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Invited review

Synergies of Operations Research and Data Mining

Stephan Meisel *, Dirk Mattfeld

Carl-Friedrich Gauss Department, University of Braunschweig, 38106 Braunschweig, Germany

ARTICLE INFO

Article history:
Received 27 August 2007
Accepted 16 October 2009
Available online 26 October 2009

Keyword:
Data Mining

ABSTRACT

In this contribution we identify the synergies of Operations Research and Data Mining. Synergies can be achieved by integration of optimization techniques into Data Mining and vice versa. In particular, we define three classes of synergies and illustrate each of them by examples. The classification is based on a generic description of aims, preconditions as well as process models of Operations Research and Data Mining. It serves as a framework for the assessment of approaches at the intersection of the two procedures.

1. Introduction

Increasing interest in the integration of Operations Research (OR) and Data Mining (DM) can be observed. Recently, a number of publications of successful approaches at the intersection of the two procedures appeared. They underline the potential for benefits from integration. However, these approaches focus either on specific application domains or on specific methods.

Criteria for the classification of existing approaches are needed for a better understanding of how far research has come. In addition, essential aims of the integration of OR and DM must be known in order to understand its long term impact. The identification of the basic synergies of OR and DM is necessary to answer both of these questions.

The purposes of the two procedures are different. Although the definition of OR is discussed until today (see e.g. [69]), its goals and methods are well known. The purpose is to optimally solve decision problems appearing in real-world applications [38]. Optimal decisions require insights into the structure of the application system under consideration. The vast body of techniques within OR provides the means for capturing this structure in terms of models. Further, OR provides the algorithmic means for deriving a decision on how to modify the application system (c.f. Fig. 1). For a detailed introduction into OR, we refer to e.g. Hillier and Lieberman [46].

In contrast to OR, DM still is quite a young discipline. Though the term "Data Mining" has been used earlier (see e.g. [64] and [31]), the field as it is known today has its origins in the mid 1990s. One of the latest introductions to the current DM methodology is given by Tan et al. [94].

A unique definition of DM has not been established yet. Nevertheless the field's subject is obvious. DM is concerned with secondary analysis of large amounts of data. Both the aspect of secondary analysis and the sheer size of data distinguish the field from common statistics [40,41]. Data is not collected based on experiments designed to answer a certain set of a priori known questions. Instead DM copes with data arising as a byproduct of either operating application systems or simulations of such systems. It aims at abstraction of information about the application system from data. (c.f. Fig. 1).

Thus, both OR and DM are application focused [102,99]. Many Data Mining approaches are within traditional OR domains like logistics [105], manufacturing [74], health care [58] or finance [57]. Further, both OR and DM are multidisciplinary. Since its origins, OR has been relying on fields such as mathematics, statistics, economics and computer science [99]. In DM, most of the current textbooks show a strong bias towards one of its founding disciplines, like database management [39], machine learning [100] or statistics [45].

Being multidisciplinary and application focused, it should be a natural step for both of the paradigms to gain synergies from integration. Some authors even suggest DM to be a natural extension of the OR problem solving methodology [92]. Recent publication of mere DM algorithms in the OR community [7,27,47,53,51,62,98] seems to strengthen this claim. Fig. 1 summarizes OR and DM in an application context, including the benefits both can gain from each other.

The remainder of this paper explains Fig. 1 in detail. We proceed as follows. In Section 2 we establish a common foundation of OR and DM from an application perspective. Process models of the two procedures are derived. In Section 3 we illustrate how OR techniques lead to increased efficiency of DM as well as how DM contributes to the effectiveness of OR. Three ways of achieving synergies of OR and DM are identified. A number of examples from

* Corresponding author. Tel.: +49 531 391 3213; fax: +49 531 391 8144.
E-mail addresses: stephan.meisel@tu-braunschweig.de (S. Meisel), d.mattfeld@tu-braunschweig.de (D. Mattfeld).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2009.10.017

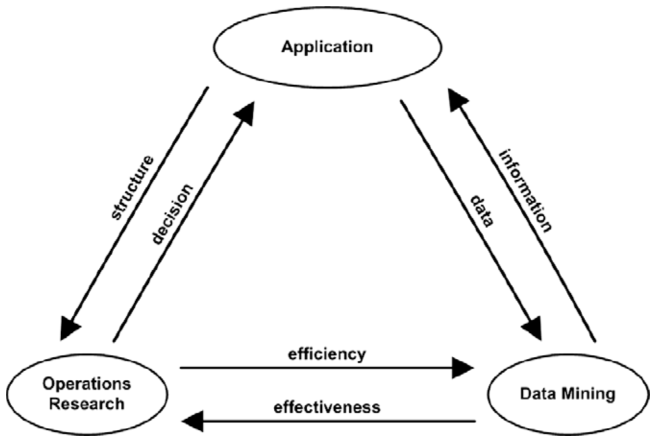

Fig. 1. Operations Research and Data Mining in an application context. OR models the structure of an application system and seeks for decisions in order to act on the application. DM transforms data into information about the application system. OR techniques increase the efficiency of DM and DM increases effectiveness of OR.

the literature are classified according to these. Section 4 concludes the paper.

2. Operations Research and Data Mining

In this section we describe the roles of OR and DM with respect to an application system (cf. Fig. 1). First, we specify both preconditions (structure and data) and aims (decisions and information) of the two procedures (Section 2.1). Then we illustrate the basic steps bridging the gap between preconditions and aims, leading to process models of OR and DM, respectively (Sections 2.2 and 2.3).

2.1. Preconditions and aims

A common point of reference for OR and DM is given by an application system. Each application system is naturally tied to a specific application domain. However, a generic view of system elements is necessary to understand the general roles of OR and DM as well as the synergies of the two.

We start from the basic definition of a system as a set of interrelated objects, with an object being either a real entity or a mere concept. Each object consists of a set of attributes, leading to a more precise definition of a system as a relation among attributes of objects. (For a thorough discussion of this definition, see [68].)

A system's appearance is given by the values its constituting attributes hold as the system is observed. A mutable system object comprises at least one attribute whose value may change while the system is considered. Changing values of single attributes result in a modification of the system's appearance. A sequence of appearances of a system is defined as system behavior. Clearly, system behavior is restricted by the domains of the values of the system attributes. Note that system behavior does not necessarily comprise interaction of subsequent appearances (see Example 2) The following two examples illustrate the definitions given up to now:

Example 1 (*Fleet surveillance*). An example system from the application domain of logistics is given by a number of vehicles, a number of potential customers and a road network as system objects. Mutable attributes are e.g. 'current location' and 'current load' of each vehicle, 'location' and 'quantity currently demanded' of each customer as well as 'estimated travel time' for each link of the road network.

An instance of such a system for the surveillance of a same-day courier service can be found in Attanasio et al. [6]. The courier service they describe constantly records data about vehicles, customers and road conditions, i.e. the company records the values of the attributes constituting the system's appearance. System behavior is e.g. due to vehicle movement operations, due to new customer orders or due to changing road conditions.

Example 2 (*Market basket analysis*). In the application domain of marketing, a system may be consisting of a number of product types sold by a company. In addition to immutable system attributes like 'product name' or 'manufacturer', a common mutable attribute for each product is 'quantity currently sold', where "currently" refers to the moment the system is observed.

Such a system has been considered many times in the literature on market basket analysis (e.g. [4,21]). In the standard market basket analysis approach a system appearance is identified with a sales transaction. Hence, subsequent system appearances will usually be to a large degree independent of each other.

A basic model of a system S consisting of a number n of attributes is given as a subset of the Cartesian product of these attributes' domains A_i, i.e.

$$S \subseteq A_1 \times \cdots \times A_n.$$

A system appearance $s \in S$ is given by an n-tuple of the current values $a_i \in A_i$ of the system attributes.

A special case occurs if a system consists of an application in combination with a number of agents deliberately affecting this application. In this case, two specific types of attributes must be considered. On the one hand, we have attributes representing the agents' decisions affecting the application. On the other hand, we have attributes representing the agents' measures for the decisions. The domain of one of the former is denoted by D_j. The domain of one of the latter attributes is denoted by V_k. For a system comprising n regular system attributes as well as m decision attributes and k measurement attributes the resulting extended system model is

$$S \subseteq A_1 \times \cdots \times A_n \times D_1 \times \cdots \times D_m \times V_1 \times \cdots \times V_l.$$

S is a true subset of the Cartesian product due to the structural determinants of the system. System structure is identified by means of the following types of relations:

- *Static relations* represent constraints on the values of single attributes as well as mutual dependencies of the values of attributes of an appearance.
- *Dynamic relations* specify dependencies between the values of (possibly different) attributes with respect to subsequent system appearances. They characterize the change of the values of these attributes over a subsequent number of system appearances. Dynamic relations do not exist if subsequent appearances are independent of each other.

The following example illustrates the concept of system structure based on the extended system model.

Example 3 (*Fleet management*). A natural extension of our fleet surveillance system (Example 1) is due to an agent taking decisions on the operation of the single vehicles. The real-time fleet management system developed in [6] is an example of such a system. In that case, decision attributes represent the assignment of customer requests to vehicles as well as the repositioning of the vehicles. In addition, measures for evaluation of these decisions are derived in terms of courier efficiency and service times. We give some examples of static relations:

The domain of a single attribute may be bounded by definition. E.g. a vehicle's 'current load' attribute is bound to a value between