Delft University of Technology

Delft Center for Systems and Control

Technical report 06-027

On algorithms for a binary-real $(\max, \times)$ matrix approximation problem*

B. De Schutter, J. Schepers, and I. Van Mechelen

If you want to cite this report, please use the following reference instead:
B. De Schutter, J. Schepers, and I. Van Mechelen, "On algorithms for a binary-real $(\max, \times)$ matrix approximation problem," *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, California, pp. 5168–5173, Dec. 2006.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: `https://www.dcsc.tudelft.nl`

*This report can also be downloaded via `https://pub.deschutter.info/abs/06_027.html`

On Algorithms for a Binary-Real $(\max,\times)$ Matrix Approximation Problem

Bart De Schutter, Jan Schepers and Iven Van Mechelen

***Abstract*— We consider algorithms to solve the problem of approximating a given matrix D with the $(\max,\times)$ product of a binary (i.e., a 0-1) matrix S and a real matrix P: $\min_{S,P}\|S\odot P-D\|$. The norm to be used is the ℓ_1, ℓ_2 or ℓ_∞ norm, and the $(\max,\times)$ matrix product is constructed in the same way as the conventional matrix product, but with addition replaced by maximization. This approximation problem arises among others in data clustering applications where the maximal component instead of the sum of the components determines the final result. We propose several algorithms to address this problem. The binary-real $(\max,\times)$ matrix approximation problem can be solved exactly using mixed-integer programming, but since this approach suffers from combinatorial explosion we also propose some alternative approaches based on alternating nonlinear optimization, and a method to obtain good initial solutions. We conclude with a simulation study in which the performance and optimality of the different algorithms are compared.**

I. Introduction

We consider a matrix approximation problem in which a data matrix $D\in\mathbb{R}^{m\times n}$ should be approximated by a model matrix $M\in\mathbb{R}^{m\times n}$ with M the $(\max,\times)$ product of a binary matrix $S\in\{0,1\}^{m\times k}$ and a real matrix $P\in\mathbb{R}^{k\times n}$: $M=S\odot P$, where $\odot$ denotes the $(\max,\times)$ matrix product: $m_{ij}=\max_{l=1,\ldots,k} s_{ik}\cdot p_{kj}$ for all i,j. This results in the following problem:

> Given a matrix $D\in\mathbb{R}^{m\times n}$ and an integer k find $S\in\{0,1\}^{m\times k}$ and $P\in\mathbb{R}^{k\times n}$ such that
>
> $$\|S\odot P-D\| \text{ is minimized.} \qquad (1)$$

In Section II we will explain how this problem arises among others in data clustering applications where the maximal component instead of the sum of the components determines the final result. Problem (1) is an extension of the HICLAS problems considered in [1]–[3]. It is also related to the $(\max,+)$ matrix approximation problem considered in [4].

Remark 1.1 Typical ranges for the dimension of the matrices appearing in the applications we target are: $m\in\{20\ldots,400\}$, $n\in\{10,\ldots,50\}$, and $k\in\{2,\ldots,7\}$. Furthermore, for practical applications the entries of D are often nonnegative, and the same should then hold for M and P. □

We denote the transpose of a matrix A by A^{T}. The ith row of A is denoted by $A_{i,\bullet}$, and the jth column by $A_{\bullet,j}$. For the norm in (1) we will consider the ℓ_1, ℓ_2 or ℓ_∞ norm, which are defined as follows for a matrix $A\in\mathbb{R}^{m\times n}$ [5]:

$$\|A\|_{\ell_1}=\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}| \quad , \quad \|A\|_{\ell_2}=\left(\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}^2\right)^{\frac{1}{2}}$$

$$\|A\|_{\ell_\infty}=\max_{i=1,\ldots,m}\max_{j=1,\ldots,n}|a_{ij}| \ .$$

In Section II we show how the binary-real matrix $(\max,\times)$ approximation problem (1) arises in the context of some data clustering approaches. Next, we discuss some properties of the solutions of Problem (1) in Section III. In Section IV we will then show that the optimization problem (1) can be recast as a mixed-integer linear (for the ℓ_1 and ℓ_∞ norm) or mixed-integer quadratic programming problem (for the ℓ_2 norm). Although there exist good commercial and free solvers for mixed-integer linear and quadratic programming problems, the problem is inherently combinatorial. Therefore, we also present some alternative approaches to find a suboptimal solution in a more efficient way in Section V. These approaches are based on an alternating approach in which S is determined using an enumerative or greedy approach and P using nonlinear (real-valued) optimization. For a given S we also propose a method to determine a good initial solution for P. Finally, in Section VII we present the results of a simulation study in which the performance and optimality of the different algorithms are compared.

II. Background of the problem

A challenge for a data analyst is to capture the structural information that is present in a given data matrix of, say, m objects by n variables. One way of achieving this goal is to search for homogeneous clusters within the objects. An example is the well-known K-means clustering technique [6], which describes an approximate (least squares) decomposition of the data into on the one hand a binary matrix which represents a partition, and on the other hand a real-valued matrix usually referred to as the cluster centroid matrix. In this paper, we will construct an approximation M to the data D, that also tries to achieve the goal of capturing the underlying structural information in D by decomposing M into a binary matrix S and a real-valued matrix P. But unlike K-means clustering, which implies a decomposition in conventional linear algebra, we consider the problem where the approximation M can be written as the $(\max,\times)$ matrix product of a binary matrix S (not necessarily a partition) and a real-valued matrix P:

$$D\approx M=S\odot P \qquad (2)$$

B. De Schutter is with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands, email: b@deschutter.info

J. Schepers and I. Van Mechelen are with the Research Group Quantitative and Personality Psychology, Faculty of Psychology and Educational Sciences, K.U.Leuven, Tiensestraat 102, B-3000 Leuven, Belgium, email: Jan.Schepers@psy.kuleuven.be, Iven.VanMechelen@psy.kuleuven.be

with $D, M \in (\mathbb{R}^+)^{m\times n}$, $S \in \{0,1\}^{m\times k}$ and $P \in (\mathbb{R}^+)^{k\times n}$, where $\mathbb{R}^+$ is the set of the nonnegative real numbers. The value of k is assumed to be known in advance.

Of course, the decomposition according to (2) implies that a specific type of structural information is captured. As an example, one can think of data pertaining to the response latencies of object recognition for a set of visually presented objects, and a set of persons. Suppose a researcher is interested in a number of latent tasks that need to be processed in parallel by the visual system in order for the objects to be recognized [7], [8]. Moreover, the researcher does not know which latent tasks need to be processed for each object or does not want to make certain a priori assumptions about this. In this case, the binary matrix S in (2) may represent the latent tasks performed by the visual system as implied by each of the objects (such as processing the information on spatial frequency, orientation, color, etc.), and the real-valued matrix P in (2) may denote the response times that each latent task requires from each persons visual system. To illustrate this, consider the following hypothetical data matrix where the rows correspond to the set of visually presented objects and the columns to the set of persons participating in the experiment:

$$D = \begin{bmatrix} 0.67 & 0.06 & 0.26 & 0.57 \\ 0.82 & 0.96 & 0.55 & 0.57 \\ 0.67 & 0.99 & 0.36 & 0.60 \\ 0.82 & 0.96 & 0.55 & 0.05 \\ 0.57 & 0.99 & 0.36 & 0.60 \end{bmatrix}.$$

This data matrix can be decomposed, according to expression (2), into a binary matrix S in which the three columns represent the latent tasks that need or need not to be processed (indicated by 1 and 0, respectively) in order for the objects to be recognized. The matrix P represents, in each column, the processing time required for each of the three tasks corresponding to that particular person. We have

$$S = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 0.57 & 0.99 & 0.36 & 0.60 \\ 0.82 & 0.96 & 0.55 & 0.05 \\ 0.67 & 0.06 & 0.26 & 0.57 \end{bmatrix}.$$

If we focus on the second object (i.e., the second row of S), we see that both the second and third latent tasks need to be processed in order for that object to be recognized. The $(\max,\times)$ algebra in (2) implies that the response latencies of recognition for that object (as represented in the second row of D) will only be affected by the most demanding of these two latent tasks. This matches up in a very natural way with the assumption of parallel processing of information by the visual system.

Of course, in practice there will always be errors and noise present in the data, and then we have to approximate D as good as possible with the model matrix $M = S \odot P$, where the error or goodness-of-fit is indicated by $\|D - S \odot P\|$. In this paper we will consider the ℓ_1, ℓ_2 and ℓ_∞ norm. This results in the optimization problem (1).

III. Properties of the solutions

A. Uniqueness

If $W \in \{0,1\}^{k\times k}$ is permutation matrix (i.e., W has exactly one 1 on each row and each column, the other elements being equal to 0), then for a given optimal solution (S^*, P^*) of Problem (1), $(S^* \cdot W, W^{\mathrm{T}} \cdot P^*)$ is also an optimal solution where $\cdot$ denotes the conventional matrix multiplication.

B. Range of the entries

Let us denote the jth column maximum and minimum of D as follows: $d_{\max,j} = \max_i d_{ij}$ and $d_{\min,j} = \min_i d_{ij}$.

Proposition 3.1: Given a data matrix D there always exists an optimal approximation (S^*, P^*) for the ℓ_1, ℓ_2 or ℓ_∞ norm such that the entries of the jth column of P^* are bounded from above by $d_{\max,j}$ and bounded from below by $d_{\min,j}$ for each $j \in \{1,\ldots,n\}$. ◇

Proof: Let $j \in \{1,\ldots,n\}$ and consider an optimal approximation (S^*, P^*). We will prove that there exists an(other) optimal solution $(S^\#, P^\#)$ for which the entries of the jth column of $P^\#$ are bounded from above by $d_{\max,j}$. The proof for the lower bound is similar.

Define $L = \{l \in \{1,\ldots,k\} \mid p^*_{lj} > d_{\max,j}\}$. If $L = \emptyset$ the upper bound part of the proposition is proved. So let us now consider the case $L \neq \emptyset$.

Consider $l \in L$ such that $l = \arg\max_u p^*_{uj}$, and define $I_l = \{i \mid s_{il} = 1\}$. Now we distinguish between two cases: $I_l = \emptyset$ and $I_l \neq \emptyset$. If $I_l = \emptyset$, then we can change the value of p^*_{lj} without changing the value of $S^* \odot P^*$ and thus also of $\|S^* \odot P^* - D\|$. Hence, we can always set $p^*_{lj} = d_{\max,j}$ in this case.

Now we consider the case $I_l \neq \emptyset$ and we show by contradiction that this case will never occur. We define $S^\# = S^*$, $P^\# \in \mathbb{R}^{k\times n}$ such that $p^\#_{uv} = p^*_{uv}$ for all $(u,v) \in \{1,\ldots,k\} \times \{1,\ldots,n\}$ with $(u,v) \neq (l,j)$ and such that $p^\#_{lj} = d_{\max,j}$. Define $E^* = S^* \odot P^* - D$, and $E^\# = S^\# \odot P^\# - D$. Then for any $i \notin I_l$ we have $e^\#_{ij} = e^*_{ij}$ and for any $i \in I_l$, we have $e^\#_{ij} = \max_u(s^\#_{iu} p^\#_{uj}) - d_{ij} = s^\#_{il} p^\#_{lj} - d_{ij} = p^\#_{lj} - d_{ij} = d_{\max,j} - d_{ij} \geqslant 0$. Furthermore, as in addition $p^*_{lj} > d_{\max,j}$ we have $e^*_{ij} > e^\#_{ij}$ for each $i \in I_l$. So $0 \leqslant e^\#_{ij} < e^*_{ij}$ for each $i \in I_l$ and $e^\#_{ij} = e^*_{ij}$ for each $i \notin I_l$. As a consequence, we have $\|E^\#\| < \|E^*\|$ for the ℓ_1, ℓ_2, and ℓ_∞ norm. As (S^*, P^*) is an optimal approximation, we thus obtain a contradiction. Hence, the initial assumption that $I_l \neq \emptyset$ is not valid.

By repeating the reasoning above we can show for each $l \in L$ we have $I_l = \emptyset$. As in this case we can replace $p^*_{lj} > d_{\max,j}$ by $d_{\max,j}$ without changing $S^* \odot P^*$, there always exists an optimal approximation (S^*, P^*) for which the entries of the jth column of P^* are bounded from above by $d_{\max,j}$. ■

IV. An optimal approach based on mixed-integer programming

We will show that by introducing some auxiliary variables Problem (1) can be recast into a mixed integer quadratic programming (MIQP) problem (for the ℓ_2 norm), or into a mixed integer linear programming (MILP) problem (for the ℓ_1 and the ℓ_∞ norms).

In general, an MILP problem is defined as follows:

$$\min_{x\in\mathbb{R}^n} f^{\mathrm{T}}x$$
$$\text{subject to } Ax \leqslant b \text{ and } x_i \in \mathbb{Z} \text{ for } i \in I_{\text{int}},$$

and an MIQP problem as follows:

$$\min_{x\in\mathbb{R}^n} \tfrac{1}{2}x^{\mathrm{T}}Hx + f^{\mathrm{T}}x$$
$$\text{subject to } Ax \leqslant b \text{ and } x_i \in \mathbb{Z} \text{ for } i \in I_{\text{int}},$$

for appropriately sized matrices H and A, and vectors f and b, and where $I_{\text{int}} \subseteq \{1,\ldots,n\}$ and $\mathbb{Z}$ is the set of the integers. MILP and MIQP problems can be solved using branch-and-bound methods [9].

Consider Problem (1) and introduce the auxiliary matrix variable $E \in \mathbb{R}^{m\times n}$ such that $E = S\odot P - D$. Then we have $e_{ij} = \max_{l=1,\ldots,k}(s_{il}p_{lj}) - d_{ij}$ for each i,j. This equation can be rewritten as

$$\max_{l=1,\ldots,k}(s_{il}p_{lj} - d_{ij} - e_{ij}) = 0 \ . \tag{3}$$

Now we introduce additional boolean variables δ_{ilj} that indicate whether or not $s_{il}p_{lj} - d_{ij} - e_{ij}$ is different from 0. Let B_{ilj} be a lower bound for $s_{il}p_{lj} - d_{ij} - e_{ij}$ (see equation (15) below) and consider the equations

$$B_{ilj}\delta_{ilj} \leqslant s_{il}p_{lj} - d_{ij} - e_{ij} \leqslant 0 \tag{4}$$
$$\sum_{l=1}^{k} \delta_{ilj} \leqslant k-1 \tag{5}$$
$$\delta_{ilj} \in \{0,1\} \ . \tag{6}$$

Let us now show that the system (4)–(6) is equivalent to equation (3). Note that (5) together with (6) implies that at least one δ_{ilj} is equal to 0. If $\delta_{ilj} = 0$ then (4) implies that $s_{il}p_{lj} - d_{ij} - e_{ij} = 0$, whereas $\delta_{ilj} = 1$ implies that $s_{il}p_{lj} - d_{ij} - e_{ij} \leqslant 0$. Hence, (4)–(6) is indeed equivalent to (3). In order to obtain linear equations we now introduce additional (real-valued) auxiliary variables z_{ilj} such that $z_{ilj} = s_{il}p_{lj}$. Now we use the following result taken from [10]:

Lemma 4.1: Consider the function $f : \mathscr{X} \to \mathbb{R} : x \mapsto f(x)$ with $\mathscr{X}$ a bounded subset of $\mathbb{R}$. Consider two real numbers M_f and m_f with $M_f \geqslant \sup_{x\in\mathscr{X}} f(x)$ and $m_f \leqslant \inf_{x\in\mathscr{X}} f(x)$. Then the expression $y = \delta f(x)$ with δ a boolean variable is equivalent to the system

$$m_f\delta \leqslant y \leqslant M_f\delta$$
$$f(x) - M_f(1-\delta) \leqslant y \leqslant f(x) - m_f(1-\delta) \ . \qquad \diamond$$

Recall that it follows from Proposition 3.1 that $d_{\min,j} \leqslant p_{ij} \leqslant d_{\max,j}$ for all i,j. Hence, by Lemma 4.1 we can replace the expression $z_{ilj} = s_{il}p_{lj}$ by the system

$$d_{\min,j}s_{il} \leqslant z_{ilj} \leqslant d_{\max,j}s_{il}$$
$$p_{lj} - d_{\max,j}(1-s_{il}) \leqslant z_{ilj} \leqslant p_{lj} - d_{\min,j}(1-s_{il}) \ .$$

So summarizing, the equation $E = S\odot P - D$ with S a boolean matrix can be recast into the following system of *mixed integer linear* inequalities

$$s_{il} \in \{0,1\} \tag{7}$$
$$B_{ilj}\delta_{ilj} \leqslant z_{ilj} - d_{ij} - e_{ij} \leqslant 0 \tag{8}$$
$$\sum_{l=1}^{k} \delta_{ilj} \leqslant k-1 \tag{9}$$
$$\delta_{ilj} \in \{0,1\} \tag{10}$$
$$d_{\min,j}s_{il} \leqslant z_{ilj} \leqslant d_{\max,j}s_{il} \tag{11}$$
$$p_{lj} - d_{\max,j}(1-s_{il}) \leqslant z_{ilj} \leqslant p_{lj} - d_{\min,j}(1-s_{il}) \tag{12}$$

for $i = 1,\ldots,m$, $l = 1,\ldots,k$ and $j = 1,\ldots,n$.

Now we have to minimize $\|E\|$ over the system (7)–(12). Clearly, for the ℓ_2 norm this implies that we have to minimize a quadratic objective function[1] $\sum_{i,j} e_{ij}^2$ over (7)–(12), i.e., we have to solve an MIQP problem.

For the ℓ_∞ norm we introduce an additional auxiliary variable t such that $|e_{ij}| \leqslant t$ or equivalently

$$-t \leqslant e_{ij} \leqslant t \qquad \text{for all } i,j, \tag{13}$$

and then we minimize t over the system (7)–(13), which yields an MILP problem. It is easy to verify that for the optimal solution we have $t^* = \max_{i,j} |e_{ij}^*|$, i.e., in the optimum the condition $|e_{ij}| \leqslant t$ holds with equality.

For the ℓ_1 norm we take a similar approach: we introduce additional auxiliary variables t_{ij} for $i = 1,\ldots,m$ and $j = 1,\ldots,n$ such that $|e_{ij}| \leqslant t_{ij}$ or equivalently

$$-t_{ij} \leqslant e_{ij} \leqslant t_{ij} \qquad \text{for all } i,j, \tag{14}$$

and we minimize $\sum_{i,j} t_{ij}$ over the system (7)–(12) and (14), which also yields an MILP problem. It is easy to verify that for the optimal solution we have $t_{ij}^* = |e_{ij}^*|$.

Determination of B_{ilj}

Let us now determine a lower bound B_{ilj} for $s_{il}p_{lj} - d_{ij} - e_{ij}$. From Proposition 3.1 it follows that we may assume that for an optimal solution we have $d_{\min,j} \leqslant p_{ij} \leqslant d_{\max,j}$ for all i,j, and thus $\min(d_{\min,j},0) \leqslant s_{il}p_{lj} \leqslant \max(d_{\max,j},0)$ for all i,l,j. Furthermore, $e_{ij} = \max_{l=1,\ldots,k}(s_{il}p_{lj}) - d_{ij}$ and thus $e_{ij} + d_{ij} = \max_{l=1,\ldots,k}(s_{il}p_{lj}) \leqslant \max(d_{\max,j},0)$ for all i,j. This implies that

$$\begin{aligned} B_{ilj} &\stackrel{\text{def}}{=} \min(d_{\min,j},0) - \max(d_{\max,j},0) \\ &\leqslant s_{il}p_{lj} - d_{ij} - e_{ij} \quad \text{for all } i,l,j. \end{aligned} \tag{15}$$

The approach to determine S and P presented in this section will in principle always yield the optimal solution to the Problem (1) (provided that the MILP or MIQP algorithm is allowed to run until the exact solution is found). However, as the number of variables in the MILP or MIQP increases (i.e., as n, m, and/or k increase), the running time and the memory requirements of the MILP or MIQP algorithm will in general increase (exponentially) as MILP and MIQP problems are NP-hard [11], [12]. Therefore, in the next section we will propose some alternative algorithms that will require less computing time and memory (but in general they will not always provide the optimal solution).

[1] Note that minimizing $\|E\|_{\ell_2}$ is equivalent to minimizing $\|E\|_{\ell_2}^2$.

V. Suboptimal algorithms

A. A suboptimal alternating approach for S and P

We propose an approach which is based on alternately determining S and P. Note that in general the convergence of this approach to an optimal solution cannot be guaranteed (see also Section VII). This algorithm works as follows:

- Given: D, k, a maximum number of iteration steps N, and a termination tolerance τ
- Initialization: Compute an initial guess S_0 and P_0 (cf. Section VI), and set $l := 0$
- Loop: **while** $l \leqslant N$ **and** $\|D - S_l \odot P_l\| \geqslant \tau$ **do**
  - Determine a (sub)optimal P^* for $S = S_l$ (cf. Section V-B) and set $P_{l+1} := P^*$
  - Determine a (sub)optimal S^* for $P = P_{l+1}$ (cf. Section V-C) and set $S_{l+1} := S^*$
  - Set $l := l + 1$
- Output: $S = S_l$ and $P = P_l$

In the next sections we propose methods to determine a (sub)optimal P for a given S, a (sub)optimal S for a given P, and appropriate initial guesses for S and P.

Note that instead of the alternating approach we could also use, e.g., simulated annealing [13] on S and P.

B. Algorithms to determine P for a given S

For a given S the problem to be solved becomes

$$\min_{P \in \mathbb{R}^{k \times n}} \|S \odot P - D\| \ . \tag{16}$$

Note that for the ℓ_1, ℓ_2, and ℓ_∞ norm this problem can be solved column by column as the jth column of P only influences the jth column of the product $S \odot P$. Problem (16) is in general not convex. To solve this problem we can use a multi-start unconstrained optimization method, a multi-start nonlinear least-squares method, or if we also include the bounds as given in Proposition 3.1 a multi-start constrained optimization method. An alternative approach would be to use simulated annealing. We refer to [13], [14] for more information on these nonlinear optimization methods.

C. Algorithms to determine S for a given P

For a given P the problem to be solved becomes

$$\min_{S \in \{0,1\}^{m \times k}} \|S \odot P - D\| \tag{17}$$

For the ℓ_1, ℓ_2, and ℓ_∞ norm this problem can be solved row by row as the ith row of S only influences the ith row of $S \odot P$. Note that now we have the additional condition that $s_{ij} \in \{0,1\}$ for all i, j, which makes this problem combinatorial. Some approaches to determine S are enumeration[2], tabu search [15], simulated annealing [13], a genetic algorithm [16], a mixed integer programming approach similar to that of Section IV to simultaneously determine optimal values for S and P, or the following greedy approach:

- Given: D, P, and a termination tolerance τ
- **for** each row $S_{i,\bullet}$ of S **do**
  - Initialize $S_{i,\bullet}$ to $\begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}$
  - Set $\mathscr{F} := \{1, \dots, n\}$ ($\mathscr{F}$ contains the indices of the "free" entries of $S_{i,\bullet}$)
  - **while** $\mathscr{F} \neq \emptyset$ **and** $\|S_{i,\bullet} \odot P - D_{i,\bullet}\| \geqslant \tau$ **do**
    - Define $e_0 = \|S_{i,\bullet} \odot P - D_{i,\bullet}\|$
    - For each index $f \in \mathscr{F}$ flip the fth entry of $S_{i,\bullet}$ to 1, determine the corresponding value $e_f = \|S_{i,\bullet} \odot P - D_{i,\bullet}\|$, and flip the entry back to 0
    - Select the index f^* for which e_f is the smallest
    - **if** ${e_f}^* < e_0$ **then**
      Remove f^* from $\mathscr{F}$ and permanently fix S_{if^*} to 1
      else
      Set $\mathscr{F} = \emptyset$ (no further improvement possible)
- Output: S

Many of the suboptimal approaches for S and P presented above require an initial starting point. Therefore, in the next section we present a method to determine initial solutions, first, for S, and next, for P (for a given S).

VI. Initial solutions for S and P

A. Initial solution for S

To determine an initial S we could consider a random binary matrix of size m by k, or use a heuristic initial solution constructed as follows: Assuming that $k \leqslant n$ (which generally is the case), we select the k first columns of D and we determine the average value of the entries of this m by k submatrix. Next we replace all entries that are larger than or equal to this average value by 1 and the other entries by 0.

B. A suboptimal initial solution for P based on the largest subsolution

Note that if S contains a zero row, the corresponding row of $S \odot P$ is also a zero row, so these rows cannot be influenced by changing P. Hence, we now assume that the zero rows of S and the corresponding rows of D have been removed. Furthermore, for the sake of simplicity we assume that all entries of D are nonnegative.

The proposed approach is based on a two-step procedure:

- First we compute the "largest subsolution" of $S \odot P \leqslant D$, i.e., we solve the multi-objective optimization problem

$$\begin{aligned} &\max_P P \\ &\text{subject to } S \odot P \leqslant D \ . \end{aligned} \tag{18}$$

  This problem can be solved analytically as will be shown below. Furthermore, as S is assumed to have no zero rows, the solution of Problem (18) is always finite.
- Next, for each column j of P we add a constant α_j to the column such that $\|D_{\bullet,j} - S \odot (P_{\bullet,j} + \alpha_j)\|$ is minimized. We will show that this problem can also be solved analytically for the ℓ_1, ℓ_2, and ℓ_∞ norm.

Note that this approach yields the *exact* solution for a given S if there exists a P such that (S, P) is the optimal solution of Problem (1). These solutions can also be used as an initial

[2]Note that for practical problems k ranges from 2 to 7 (cf. Remark 1.1). So enumeration over all possible 2^k values for each row is still feasible.

solution for the approaches used to determine both S and P (cf. Section V-A), or P for a given S (cf. Section V-B).

Step 1: Consider Problem (18) and define $I_l = \{i \mid s_{il} = 1\}$ for $l = 1, \ldots, k$. Note that due to our assumption that S has no zero rows, we always have $I_l \neq \emptyset$ for each $l \in \{1, \ldots, k\}$. For a feasible solution of Problem (18) we should have $s_{il} p_{lj} \leqslant d_{ij}$ for all i, l, j, and thus $p_{lj} \leqslant \min_{i \in I_l} d_{ij}$ as $I_l \neq \emptyset$ and as we have assumed that $d_{ij} \geqslant 0$ for all i, j. Hence, the entries of the optimal solution P_{ls} of Problem (18) are given by

$$(P_{\text{ls}})_{lj} = \min_{i \in I_l} d_{ij} \ . \tag{19}$$

It is easy to verify that the following property holds:

$$\begin{aligned} &\text{for each } j \in \{1, \ldots, n\} \text{ there exists an index } i_j \\ &\text{such that } (S \odot P_{\text{ls}})_{i_j j} = d_{i_j j} \ . \end{aligned} \tag{20}$$

Also note that if all the entries of D are nonnegative this also holds for the entries of P_{ls}.

Step 2: Let $j \in \{1, \ldots, n\}$ and now consider the following optimization problem for the given matrix S and for the matrix P_{ls} given by (19):

$$\min_{\alpha_j} \| D_{\bullet,j} - S \odot ((P_{\text{ls}})_{\bullet,j} + \alpha_j) \| \ .$$

In order to simplify the notation we define $\alpha = \alpha_j$, $p = (P_{\text{ls}})_{\bullet,j}$, $d = D_{\bullet,j}$, and we consider the problem

$$\min_{\alpha} \| d - S \odot (p + \alpha) \| \ . \tag{21}$$

If we define $e(\alpha) = d - S \odot (p + \alpha)$ and $\mu = S \odot p$ (note that by construction $\mu \leqslant d$), then we have

$$\begin{aligned} e_i(\alpha) &= d_i - \max_{l=1,\ldots,k} s_{il}(p_l + \alpha) \\ &= d_i - \max_{l=1,\ldots,k} (s_{il} p_l + s_{il} \alpha) \\ &= d_i - \max_{l=1,\ldots,k} (s_{il} p_l) - \alpha \qquad \text{(as there is at least one index } l \text{ with } s_{il} \neq 0) \\ &= d_i - \mu_i - \alpha \ . \end{aligned}$$

Note that $e(0) \geqslant 0$ and that e is a decreasing function of α. This implies that the optimal α for Problem (21) will satisfy the condition $\alpha \geqslant 0$. Now we consider the solution of Problem (21) for the ℓ_1, ℓ_2, and ℓ_∞ norm respectively:

- ℓ_1 **norm**:
  In this case the optimization problem becomes: $\min_\alpha \sum_{i=1}^m |d_i - \mu_i - \alpha|$. Using generalized gradients, the optimal value for α is then given by the solution of the equation $\sum_{i=1}^m \operatorname{sgn}(d_i - \mu_i - \alpha) = 0$, which loosely speaking implies that the optimal value of α is such that for half of the indices i the sign is positive and for the other half the sign is negative, i.e., α^* is the median of the set $\{d_1 - \mu_1, d_2 - \mu_2, \ldots, d_m - \mu_m\}$.
- ℓ_2 **norm**:
  In this case the optimization problem results in $\min_\alpha \sum_{i=1}^m (d_i - \mu_i - \alpha)^2$. The optimal value for α is then given by the solution of the equation $\sum_{i=1}^m 2(d_i - \mu_i - \alpha)(-1) = 0$, i.e., $\alpha^* = \frac{1}{m} \sum_{i=1}^m (d_i - \mu_i)$.
- ℓ_∞ **norm**:
  In this case the optimization problem becomes:

$$\begin{aligned} &\min_\alpha \max_{i=1,\ldots,m} |d_i - \mu_i - \alpha| \\ \Leftrightarrow\ &\min_\alpha \max_{i=1,\ldots,m} \max(d_i - \mu_i - \alpha, \alpha + \mu_i - d_i)) \\ \Leftrightarrow\ &\min_\alpha \max \big(\max_{i=1,\ldots,m} (d_i - \mu_i - \alpha), \\ &\qquad \max_{i=1,\ldots,m} (\alpha - (d_i - \mu_i)) \big) \\ \Leftrightarrow\ &\min_\alpha \max \big(\max_{i=1,\ldots,m} (d_i - \mu_i) - \alpha, \\ &\qquad \alpha - \min_{i=1,\ldots,m} (d_i - \mu_i) \big) \\ \Leftrightarrow\ &\min_\alpha \max(\delta_{\max} - \alpha, \alpha) \end{aligned}$$

  with $\delta_{\max} = \max_{i=1,\ldots,m} (d_i - \mu_i)$ and where we have taken into account that $d \geqslant \mu$ and that there exists at least one index i such that $d_i = (S \odot p)_i = \mu_i$ (cf. (20)), which implies that $\min_{i=1,\ldots,m}(d_i - \mu_i) = 0$. As both arguments of the max function are affine in the scalar variable α, it is easy to verify that the optimal value of α is obtained when $\delta_{\max} - \alpha = \alpha$. So $\alpha^* = \frac{\delta_{\max}}{2}$.

VII. Comparison of the algorithms

In this section we present the results of a comparison of the algorithms proposed above. We consider the ℓ_2 norm, and we define a set of 250 random test matrices, in which all data matrices can be represented exactly as $S \odot P$. The fact that an exact solution exists allows us to assess the optimality of an algorithm when applied to the given data matrix.

For constructing a test matrix $D^{(i)}$ for which an exact approximation exists, we first select a random integer $m^{(i)} \in \{2, \ldots, 9\}$, and we define $n^{(i)} = \max\left(2, \left\lfloor \frac{2}{3} m^{(i)} \right\rfloor\right)$, $k^{(i)} = \max\left(2, \min\left(n^{(i)} - 1, \left\lfloor \frac{1}{2} m^{(i)} \right\rfloor\right)\right)$, where $\lfloor x \rfloor$ with x a real number denotes the largest integer less than or equal to x. Next, we create a random non-zero binary matrix $S^{(i)} \in \{0,1\}^{m^{(i)} \times k^{(i)}}$, and a random integer matrix $P^{(i)} \in \{0, \ldots, 10\}^{k^{(i)} \times n^{(i)}}$, and we construct $D^{(i)} = S^{(i)} \odot P^{(i)}$.

We have applied the following 5 approaches:

1) the MIQP-based approach of Section IV,
2) a heuristic approach using the initial solution for S of Section VI-A and the corresponding suboptimal solution for P of Section VI-B,
3) an alternating approach (cf. Section V-A) that uses an enumerative approach for (the rows of) S and an multi-start SQP-based approach for (the columns of) P,
4) an alternating approach that uses the greedy approach of Section V-C for (the rows of) S and an multi-start SQP-based approach for (the columns of) P,
5) a simulated annealing approach for S and P jointly.

The algorithms were implemented in Matlab (except for the MIQP solver, for which we used the TOMLAB Matlab interface to CPLEX). The tuning parameters for each of the approaches above (such as the number of initial starting points for the multi-start SQP-based approach, the initial temperature, annealing period and rate for the simulated

annealing approach, etc.) have been set to heuristically determined "optimal" values. As the (first) starting point for the alternating approaches and for the simulated annealing approach we have used the initial solutions presented in Sections VI-A and VI-B. For the MIQP-based approach we took the default initial value of CPLEX.

In Figure 1 we plot the percentage of optimal solutions, the average CPU time[3], and the average relative error values $\|S \odot P - D\|_{\ell_2} / \|D\|_{\ell_2}$ as a function of the row dimension m of the $(\max, \times)$ matrix product for each of the solution approaches. From these plots we conclude that the MIQP approach indeed always retrieves the exact solution, but at the cost of higher computation times. We note that the alternating approaches perform better than the heuristic approach and the simulated annealing approach. Although for the alternating approaches the number of times the exact optimal solution is retrieved drops below 50 % for larger values of m (see leftmost plot), the relative error is still sufficiently small (see rightmost plot), i.e., we obtain a suboptimal solution. So we could say that the alternating approaches offer a reasonable trade-off between speed and optimality.

VIII. Conclusions and future research

We have considered a binary-real $(\max, \times)$ matrix approximation problem that arises in the context of some data clustering applications. We have shown that the exact solution of this problem can be obtained by solving a mixed-integer linear or quadratic programming problem. However, as these mixed-integer programming problems are in general combinatorial, we have proposed some alternative suboptimal solution approaches for the binary-real $(\max, \times)$ matrix approximation problem. We have discussed how good initial solutions can be obtained. Finally, we have made a comparison of the proposed algorithms. These experiments show that the mixed-integer approach indeed always retrieves the optimal solution, and that the alternating approaches offer a reasonable trade-off between speed and optimality.

Topics for future research include: Further analysis of the properties of the binary-real $(\max, \times)$ matrix approximation problem and its optimal solutions, further tuning and improvement of the proposed algorithms, development of tuning guidelines, investigating other approximations (e.g., semidefinite programming [17]) and a more extensive comparison and assessment (also for real data sets).

Acknowledgments

Research partially funded by Fund for Scientific Research — Flanders project G.0146.06 (awarded to I. Van Mechelen).

References

[1] I. Leenen and I. Van Mechelen, "An evaluation of two algorithms for hierarchical classes analysis," *Journal of Classification*, pp. 57–80, 2001.

[2] P. De Boeck and S. Rosenberg, "Hierarchical classes: Model and data analysis," *Psychometrika*, vol. 53, pp. 361–381, 1988.

[3] I. Van Mechelen, P. De Boeck, and S. Rosenberg, "The conjunctive model of hierarchical classes," *Psychometrika*, vol. 60, pp. 505–521, 1995.

[4] B. De Schutter and B. De Moor, "Matrix factorization and minimal state space realization in the max-plus algebra," in *Proceedings of the 1997 American Control Conference*, Albuquerque, New Mexico, June 1997, pp. 3136–3140.

[5] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, United Kingdom: Cambridge University Press, 1985.

[6] J. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.

[7] B. McElree and M. Carrasco, "The temporal dynamics of visual search: Evidence for parallel processing in feature and conjunction searches," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 25, pp. 1517–1539, 1999.

[8] G. Rousselet, M. Fabre-Thorpe, and S. Thorpe, "Parallel processing in high-level categorization of natural images," *Nature Neuroscience*, vol. 5, pp. 629–630, 2002.

[9] R. Fletcher and S. Leyffer, "Numerical experience with lower bounds for MIQP branch-and-bound," *SIAM Journal on Optimization*, vol. 8, no. 2, pp. 604–616, May 1998.

[10] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, Mar. 1999.

[11] M. Garey and D. Johnson, ""Strong" NP-completeness results: Motivation, examples, and implications," *Journal of the Association for Computing Machinery*, vol. 25, no. 3, pp. 499—508, July 1978.

[12] A. Schrijver, *Theory of Linear and Integer Programming*. Chichester, UK: John Wiley & Sons, 1986.

[13] R. Eglese, "Simulated annealing: A tool for operations research," *European Journal of Operational Research*, vol. 46, pp. 271–281, 1990.

[14] P. Pardalos and M. Resende, Eds., *Handbook of Applied Optimization*. Oxford, UK: Oxford University Press, 2002.

[15] F. Glover and M. Laguna, *Tabu Search*. Boston: Kluwer Academic Publishers, 1997.

[16] L. Davis, Ed., *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.

[17] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, ser. MPS/SIAM Series on Optimization. Philadelphia, Pennsylvania: SIAM, 2001, vol. 1.

[3]The experiments where performed on a 2.4 GHz Pentium 4 PC with 512 MB of RAM and running Linux.

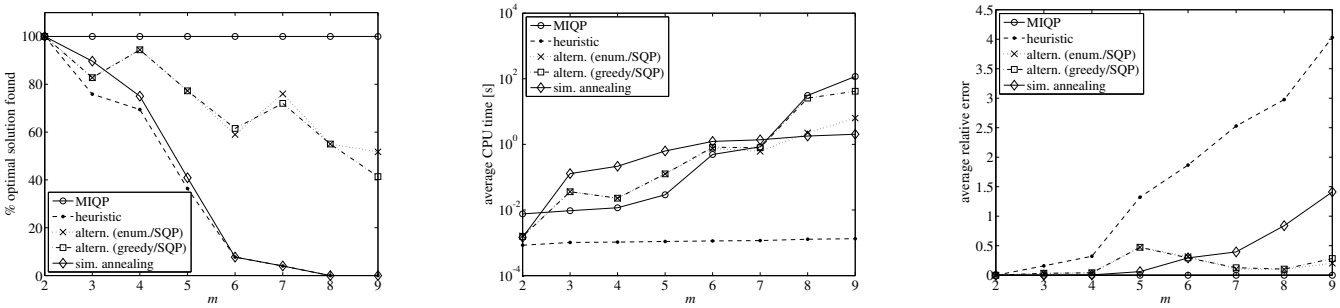

Fig. 1. Percentage of optimal solutions, average CPU time, and average relative error values as a function of m for the data matrices of the test set.