Low-Rank Boolean Matrix Approximation by Integer Programming

Réka Á. Kovács[*]
Oxford Mathematical Institute
reka.kovacs@maths.ox.ac.uk

Oktay Gunluk
IBM Research
gunluk@us.ibm.com

Raphael A. Hauser[†]
Oxford Mathematical Institute
hauser@maths.ox.ac.uk

Abstract

Low-rank approximations of data matrices are an important dimensionality reduction tool in machine learning and regression analysis. We consider the case of categorical variables, where it can be formulated as the problem of finding low-rank approximations to Boolean matrices. In this paper we give what is to the best of our knowledge the first integer programming formulation that relies on only polynomially many variables and constraints, we discuss how to solve it computationally and report numerical tests on synthetic and real-world data.

1 Introduction

A common problem in machine learning and regression analysis is to predict the value of an as of yet unobserved output variable of interest as a function of m observed input variables called features. The functional dependence between the input and output variables has to be learned from a set of n samples, or items, each of whose inputs and outputs are known. When n is small in relation to m, this task may be affected by overfitting [HTF13], but in applications it is typically observed that the data is inherently well approximated by a lower dimensional representation, and that reducing the dimensionality dramatically reduces the overfitting. A classical technique to achieve dimensionality reduction is linear factor analysis [Jol10, GV89, Mul09]: Given a data matrix $X \in \mathbb{R}^{n\times m}$ whose rows correspond to n items and columns to m features, compute $C \in \mathbb{R}^{n\times k}$ and $R \in \mathbb{R}^{k\times m}$ such that the Frobenius norm $\|X - CR\|_F$ of the approximation error is minimal for some fixed rank $k \in \mathbb{N}$. The rank-k approximation CR describes the data matrix using only k implicit features: the rows of R specify how the observed variables relate to the implicit features, while the rows of C show how the observed variables of each item can be (approximately) expressed as a linear combination of the k implicit features.

Many practical data sets contain a mixture of different data types. In this paper we concentrate on categorical variables. For example, in the data set of congressional votes discussed in the numerical section of this paper, items correspond to 435 members of congress, and features to votes on 16 different bills. The voting behavior of each member can be represented by two Boolean variables per bill, a first variable taking the value 1 if the member voted "yes", and a second variable that takes the value 1 if they voted "no". Note that in case of abstentions, both variables take the value 0, indicating an absence of both categorical features. Such an expansion of categorical variables into Boolean variables proportional to the number of different categories is both typical and necessary because of an asymmetry of treating 1s and 0s under Boolean arithmetic.

For Boolean data matrices $X \in \{0,1\}^{n\times m}$ it is natural to require that the factor matrices C and R are Boolean as well. This requirement introduces an intrinsic difficulty into the problem because real arithmetic is replaced by arithmetic over the Boolean semiring in which $1 + 1 = 1$ holds. Boolean matrix multiplication is defined as $X = C \circ R \iff x_{i,j} = \bigvee_{\ell=1}^{k} c_{i,\ell} \wedge r_{\ell,j}$ for some

[*]R.K. was supported by the Robin & Nadine Wells Scholarship from St Cross College Oxford
[†]This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Boolean matrices $X \in \{0,1\}^{n\times m}, C \in \{0,1\}^{n\times k}, R \in \{0,1\}^{k\times m}$. Note that for Boolean vectors $a, b \in \{0,1\}^k$, it holds that $\bigvee_{\ell=1}^k a_\ell \wedge b_\ell = \min\{1, \sum_{\ell=1}^k a_\ell b_\ell\}$. An optimal rank-$k$ Boolean matrix approximation for $X \in \{0,1\}^{n\times m}$ and $k \in \mathbb{N}$ is then given by $C \in \{0,1\}^{n\times k}$ and $R \in \{0,1\}^{k\times m}$ for which $\|X - C \circ R\|_F^2$ is minimal. The Boolean rank of X is defined as the smallest k for which the approximation error is zero [Kim82]. Note that the Boolean rank of a Boolean matrix X may differ from its linear algebraic rank. In the following example, inspired by [Mie13], let $X = \left[\begin{smallmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right]$ be a data matrix of Boolean variables $x_{i,j}$ that indicate if worker i has access to room j. The Boolean factorization $X = C \circ R = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{smallmatrix}\right] \circ \left[\begin{smallmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{smallmatrix}\right]$ is of exact Boolean rank 2 and reveals that there are two different roles, one requiring access to rooms 1 and 2, and the other requiring access to rooms 2 and 3, and that worker 2 serves in both roles, whereas workers 1 and 3 serve only in one. In contrast, treating X as a real matrix renders it of linear algebraic rank 3, and the best rank-2 approximation $X \approx \left[\begin{smallmatrix} 1.207 & 0.707 \\ 1.207 & 0 \\ 1.207 & -0.707 \end{smallmatrix}\right] \left[\begin{smallmatrix} 0.5 & 0.707 & 0.5 \\ 0.707 & 0 & -0.707 \end{smallmatrix}\right]$ fails to reveal a clear interpretation.

Interpreting X as the node-node incidence matrix of a bipartite graph G, the problem of finding the Boolean rank of X has an interpretation as a minimum edge covering of G by bi-cliques, which is a well known NP-complete problem [Orl77]. Correspondingly finding the best Boolean rank-k approximation of X has an interpretation of minimizing the number of errors in approximate coverings of G by k bi-cliques.

Boolean rank-k approximation is a problem that is generally solved via heuristics that may [FHMP07, FHP16] or may not yield a Boolean factorization [dL06, SSU03, TT06, UHZB16]. The method developed in [BV07, BV10] can be used to find exact Boolean rank-k decompositions, but not approximations. The method of [GGM04] produces Boolean rank-k approximations but treat the errors in 1 and 0 asymmetrically. [MMG$^+$06] presents a powerful heuristic for the correct error term and returns genuinely Boolean factors. Building on methods of [VAW06], the authors of [LVA08] presented an integer programming model that is closely related to low-rank Boolean matrix approximation but relies on mining an initial set of patterns from which the approximating factors are composed. They also provide an integer programming model with an exponential number of variables and constraints for low-rank Boolean matrix approximation, but do not detail its solution. Our paper further contributes to this discussion by introducing an integer programming model that relies on only a polynomial number of variables and constraints that can be solved by CPLEX [CPL17] for problem sizes that are realistic in applications.

2 Problem Formulation

For a given Boolean matrix $X \in \{0,1\}^{n\times m}$ and integer parameter k, we next describe how to construct two Boolean matrices $C \in \{0,1\}^{n\times k}$ and $R \in \{0,1\}^{k\times m}$ so as to minimize the approximation error $\|X - C \circ R\|_F^2 = \sum_{i\in N, j\in M} |x_{i,j} - C_i \circ R_j|$, where C_i and R_j denote the i-th row of C and j-th column of R, and $N := \{1,\dots,n\}$, $M := \{1,\dots,m\}$ and $K := \{1,\dots,k\}$. In the IP formulation below we denote the McCormick envelope [Mcc76] of $a, b \in [0,1]$ by $MC(a,b) := \{y \in \mathbb{R} : a \geq y,\ b \geq y,\ y \geq a + b - 1,\ y \geq 0\} \subseteq [0,1]$. Note that when $a, b \in \{0,1\}$, then $MC(a,b)$ only contains the point $ab \in \{0,1\}$, allowing us to express the non-linear relationship $y = ab$ in terms of linear constraints only. Optimal factors C, R may now be computed by solving the following binary integer program,

$$
\begin{aligned}
\text{(BP)} \quad & \min_{\xi,z,c,r,y} \sum_{i=1}^{n}\sum_{j=1}^{m} \xi_{i,j}, \\
& \text{s.t.} \quad x_{i,j} - z_{i,j} \leq \xi_{i,j}, \qquad z_{i,j} - x_{i,j} \leq \xi_{i,j}, && \forall i \in N;\ j \in M, && (1)\\
& \qquad z_{i,j} \leq \sum_{\ell=1}^{k} y_{i,\ell,j}, \qquad y_{i,\ell,j} \leq z_{i,j}, && \forall i \in N;\ j \in M;\ \ell \in K, && (2)\\
& \qquad y_{i,\ell,j} \in MC(c_{i,\ell}, r_{\ell,j}), && \forall i \in N;\ j \in M;\ \ell \in K, && (3)\\
& \quad \xi_{i,j},\ z_{i,j},\ c_{i,\ell},\ r_{\ell,j},\ y_{i,\ell,j} \in \{0,1\}, && \forall i \in N;\ j \in M;\ \ell \in K, && (4)
\end{aligned}
$$

where variables $c_{i,\ell}$ and $r_{\ell,j}$ denote the coefficients of C and R respectively, variables $z_{i,j}$ the coefficients of $Z = C \circ R$, $\xi_{i,j}$ the elements of $\Xi = |X - Z|$, and $y_{i,\ell,j}$ the product of the variables $c_{i,\ell}$ and $r_{\ell,j}$. Constraints (4) ensure that all variables take $\{0,1\}$ values in any feasible solution.

Constraints (1) imply that $|x_{i,j} - z_{i,j}| \leq \xi_{i,j}$ for all $i \in N$, $j \in M$ and due to the objective function, it is easy to see that $|x_{i,j} - z_{i,j}| = \xi_{i,j}$ in any optimal solution, as desired. Furthermore, any integral solution satisfies $y_{i,\ell,j} = c_{i,\ell} r_{\ell,j}$ for all $i \in N$, $j \in M$, $\ell \in K$ due to constraints (3) and therefore constraints (2) imply that $z_{i,j} = \min\{1, \sum_{\ell=1}^{k} y_{i,\ell,j}\} = \min\{1, \sum_{\ell=1}^{k} c_{i,\ell} r_{\ell,j}\}$, as desired.

2.1 Improved Formulation

Note that (BP) uses $O(kmn)$ constraints and variables. To the best of our knowledge, previous IP models for the Boolean rank-k approximation problem from the literature all required an exponential number of constraints [LVA08]. We remark that the second set of constraints (2), $y_{i,\ell,j} \leq z_{i,j}$, may be summed and replaced by constraints $\sum_{\ell} y_{i,\ell,j} \leq k z_{i,j}$ $(i \in N, j \in M)$ without changing the feasible set. Even though this formulation has fewer constraints, we found it to be less effective in computations because it caused greater branching in the code. However, the formulation (BP) can be significantly improved in other ways to make more efficacious use of the computational power of commercial solvers such as CPLEX [CPL17]. We next describe these ideas.

Relaxation of Integrality Constraints. First note that, due to the nature of McCormick envelopes, the variables $y_{i,\ell,j}$ are guaranteed to take integer values when $c_{i,\ell}, r_{\ell,j} \in \{0,1\}$. Consequently, the integrality constraint on y may be relaxed. Further note that if all y variables are integral, then for any fixed $i \in N$, $j \in M$, (2) either implies that $z_{i,j} = 0$ (if $y_{i,\ell,j} = 0$ for all $\ell \in K$) or that $z_{i,j} = 1$ (if at least one $y_{i,\ell,j} = 1$ for some $\ell \in K$). Therefore, z-variables may also be treated as continuous. Finally, since $|x_{i,j} - z_{i,j}| = \xi_{i,j}$ holds at all optimal solutions, the integrality of the ξ variables can also be relaxed. Consequently, only the r and c variables need to be declared integral which leads to a mixed integer programming formulation with $k(n+m)$ binary variables only, which is obtained by replacing constraints (4) by

$$\xi_{i,j},\ z_{i,j},\ y_{i,\ell,j} \in [0,1],\ c_{i,\ell},\ r_{\ell,j} \in \{0,1\}, \quad \forall i \in N;\ j \in M;\ \ell \in K. \tag{5}$$

Deleting Redundant Constraints. Note that for any $i \in N$ and $j \in M$ the variable $z_{i,j}$ takes a binary value in any feasible solution, and since the input data parameter $x_{i,j}$ is binary as well, one of the two constraints (1) is redundant. Specifically, we may replace (1) by

$$\begin{cases} z_{i,j} = \xi_{i,j}, & \text{if } x_{i,j} = 0, \\ 1 - z_{i,j} = \xi_{i,j}, & \text{if } x_{i,j} = 1. \end{cases} \tag{6}$$

This in turn implies that only one of the constraints (2) that involve $z_{i,j}$ is non-redundant, and that (2) may be replaced by

$$\begin{cases} y_{i,\ell,j} \leq z_{i,j}, & \text{if } x_{i,j} = 0, \\ z_{i,j} \leq \sum_{\ell=1}^{k} y_{i,\ell,j}, & \text{if } x_{i,j} = 1. \end{cases} \tag{7}$$

Using the same reasoning, one finds that half the constraints that define $MC(c_{i,\ell}, r_{\ell,j})$ are redundant, so that (3) may be replaced by

$$\begin{cases} y_{i,\ell,j} \geq r_{\ell,j} + c_{i,\ell} - 1,\ y_{i,\ell,j} \geq 0, & \text{if } x_{i,j} = 0, \\ r_{\ell,j} \geq y_{i,\ell,j},\ c_{i,\ell} \geq y_{i,\ell,j}, & \text{if } x_{i,j} = 1. \end{cases} \tag{8}$$

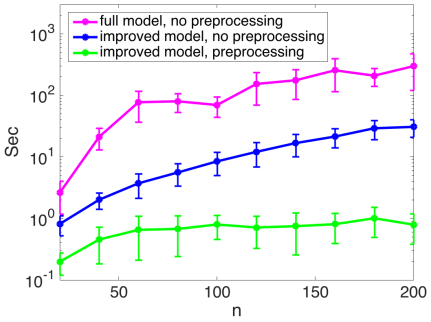
Consequently, approximately half of the constraints in the original formulation can be deleted.

Preprocessing the Input Data. In practice, the matrix $X \in \{0,1\}^{n \times m}$ of input data may contain rows (or columns) of all zeros. Deleting these rows (or columns) leads to an equivalent problem whose solution C and R can easily be translated to a solution for the original problem by inserting a row of zeros to C (respectively a column of zeros to R) in the corresponding place. In addition, X may contain duplicate rows (or columns). In this case it is sufficient to keep only one copy of each, solve the problem on the reduced data, and to reinsert the relevant copies of rows of C (respectively columns of R) in the optimal factors C and R found for the reduced data. In order for this process to yield the correct result, the objective function of the reduced problem must correspond to the approximation error of the original problem. Therefore, if row i of X is repeated α_i times, and column j is repeated β_j times, then the variable $\xi_{i,j}$ that corresponds to the remaining copy of these rows and columns must be multiplied by $\alpha_i \beta_j$ in the objective function of the reduced problem.

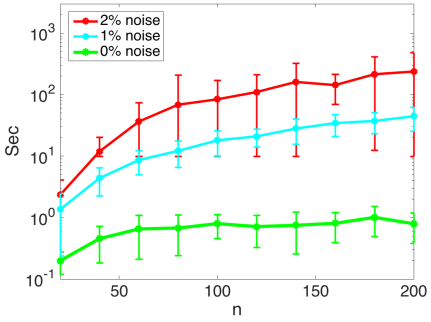
3 Computational Experiments

In this section we report numerical tests of the model (BP) on artificial and real-world datasets. The model was solved by CPLEX [CPL17] in each instance on a 2015 MacBook Pro with a 3.1 GHz Intel Core i7 processor and 16 GB of memory. The experiments with artificial data reveal that the improved formulation leads to significant speed-up.

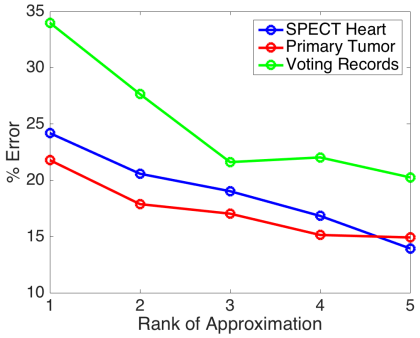
Artificial Datasets. Artificial datasets were generated by sampling matrices $C \in \{0,1\}^{n\times\kappa}$ and $R \in \{0,1\}^{\kappa\times m}$ with i.i.d. random coefficients for fixed parameters n, m, κ. The data matrix X was computed by forming the Boolean product $C \circ R$, which is a matrix with exact Boolean rank κ, and by randomly flipping $\mu\%$ of the coefficients. The sparsity of X was controlled so that $x_{i,j} = 1$ with probability $1/2$. The results of Figure 1b show how noise in the data affects the complexity of the problem. Each experiment was repeated 50 times, and the plot shows averaged running times and error bars. We observe that the complexity of the problem grows rapidly with the level of noise, an effect that occurs partly, but not solely, because the preprocessing step is less powerful at reducing the dimension of noisy matrices. This is confirmed by Figure 1a, which shows the effect of the improved formulation and the preprocessing steps. Each randomly generated input was used in the full model, the improved model, and the improved model with a preprocessing step, and the running times were averaged over 50 repeats of the experiment.



(a) Synthetic exact rank-5 data, $m = 20$.



(b) Synthetic noisy rank-5 data, $m = 20$.



(c) Percentage of approximation error as a function of the approximating rank.

Figure 1: Computational Experiments.

Real-World Datasets. The *SPECT Heart dataset* [CK] describes cardiac Single Proton Emission Computed Tomography images of 267 patients by 22 binary feature patterns, providing us a Boolean data matrix of dimension 267×22. The *Primary Tumor dataset* [KC88] contains observations of 17 categorical variables for 339 patients. 4 of the variables were non-Boolean and were converted to 11 Boolean variables, resulting in an all Boolean representation of the data matrix in dimension 339×24. The *1984 United States Congressional Voting Records dataset* [Sch87] includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The 16 categorical variables taking values of "voted for", "voted against" or "did not vote", are converted into 32 Boolean variables. The resulting Boolean data matrix is of dimension 435×32. Rank-k approximations were computed for all three datasets with $k = 1, \ldots, 5$ separately. Each run was limited to a budget of 2h, after which the incumbent solution was used as an approximation and the error reported in Figure 1c. We observe that even a suboptimal rank-5 approximation correctly reconstructs at least 80% of the data in each case. If the problems were solved to optimality, the approximation error would decrease in k, but since the figure shows the approximation errors of suboptimal solutions the curves can be non-monotonic.

4 Conclusions

To the best of our knowledge, the MIP formulation of the optimal low-rank Boolean matrix approximation problem discussed in this paper is the first model that relies on only polynomially many variables and constraints and constitutes the first exact method that is viable for realistic problem sizes. Our preliminary computational experiments suggest that our technique is applicable to real world data sets that were hitherto only approachable via heuristics [MMG+06].

References

[BV07] Radim Belohlávek and Vilém Vychodil. Formal concepts as optimal factors in boolean factor analysis: Implications and experiments. In *CLA*, 2007.

[BV10] Radim Belohlavek and Vilem Vychodil. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences*, 76(1):3 – 20, 2010.

[CK] Krzysztof J. Cios and Lukasz A. Kurgan. Uci machine learning repository: Spect heart data.

[CPL17] CPLEX Optimization, Inc., Incline Village, NV. *Using the CPLEX Callable Library, V.12.6*, 2017.

[dL06] Jan de Leeuw. Principal component analysis of binary data by iterated singular value decomposition. *Computational Statistics & Data Analysis*, 50(1):21 – 39, 2006. 2nd Special issue on Matrix Computations and Statistics.

[FHMP07] A. A. Frolov, D. Husek, I. P. Muraviev, and P. Y. Polyakov. Boolean factor analysis by attractor neural network. *IEEE Transactions on Neural Networks*, 18(3):698–707, May 2007.

[FHP16] A. A. Frolov, D. Húsek, and P. Y. Polyakov. Comparison of seven methods for boolean factor analysis and their evaluation by information gain. *IEEE Transactions on Neural Networks and Learning Systems*, 27(3):538–550, March 2016.

[GGM04] Floris Geerts, Bart Goethals, and Taneli Mielikäinen. *Tiling Databases*, pages 278–289. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[GV89] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins U. Press, Baltimore, 1989.

[HTF13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013.

[Jol10] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2010.

[KC88] Igor Kononenko and Bojan Cestnik. Uci mach. learn. rep.: Primary tumor domain, 1988.

[Kim82] K.H. Kim. *Boolean matrix theory and applications*. Monographs and textbooks in pure and applied mathematics. Dekker, 1982.

[LVA08] Haibing Lu, Jaideep Vaidya, and Vijayalakshmi Atluri. Optimal boolean matrix decomposition: Application to role engineering. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 297–306, Washington, DC, USA, 2008. IEEE Computer Society.

[Mcc76] Garth P. Mccormick. Computability of global solutions to factorable nonconvex programs: Part i – convex underestimating problems. *Math. Program.*, 10(1):147–175, December 1976.

[Mie13] Pauli Miettinen. Boolean matrix factorization in data mining and elsewhere, 2013.

[MMG+06] Pauli Miettinen, Taneli Mielikäinen, Aristides Gionis, Gautam Das, and Heikki Mannila. The discrete basis problem. In *Proc. of 10th Eur. Conf. on Principles and Practice of Knowledge Discovery in Databases*, ECMLPKDD'06, pages 335–346, Berlin, 2006. Springer-Verlag.

[Mul09] Stanley A Mulaik. *Foundations of Factor Analysis, Second Edition*. Chapman and Hall/CRC Statistics in the Social and Behavioral Sciences. Chapman and Hall/CRC, 2 edition, 2009.

[Orl77] James Orlin. Contentment in graph theory: covering graphs with cliques. In *Indagationes Mathematicae (Proceedings)*, volume 80, pages 406–424. North-Holland, 1977.

[Sch87] Jeff Schlimmer. Uci machine learning repository: 1984 US Cong. Voting Records Database, 1987.

[SSU03] Andrew I. Schein, Lawrence K. Saul, and Lyle H. Ungar. A generalized linear model for principal component analysis of binary data. page 546431, 2003.

[TT06] F. Tang and H. Tao. Binary principal component analysis. In *BMVC 2006 - Proceedings of the British Machine Vision Conference 2006*, pages 377–386, 2006.

[UHZB16] M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118, 2016.

[VAW06] Jaideep Vaidya, Vijayalakshmi Atluri, and Janice Warner. Roleminer: Mining roles using subset enumeration. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 144–153, New York, NY, USA, 2006. ACM.