Learning Kolmogorov Models for Binary Random Variables

Hadi Ghauch, Mikael Skoglund, Hossein Shokri-Ghadikolaei, Carlo Fischione, Ali Sayed

Learning Kolmogorov Models for Binary Random Variables

Hadi Ghauch [1] **Mikael Skoglund** [1] **Hossein Shokri-Ghadikolaei** [1] **Carlo Fischione** [1] **Ali Sayed** [2]

Abstract

We summarize our recent findings Authors (2017), where we proposed a framework for learning a *Kolmogorov model*, for a collection of binary random variables. More specifically, we derive conditions that *causally link* outcomes of specific random variables, and extract valuable relations from the data. We also propose an algorithm for computing the model and show its *first-order optimality*, despite the combinatorial nature of the learning problem. We apply the proposed algorithm to recommendation systems, although it is applicable to other scenarios. We believe that the work is a significant step toward interpretable machine learning.

1. Introduction

Machine learning and artificial intelligence tools have permeated a large number of areas (Marr, Sept 2016). These tools are based on *machine learning models*, which consist of learning an input-output mapping for a given dataset. Despite the plethora of models (e.g., matrix factorization (Koren et al., 2009), SVD-based models (Koren, 2008), neural networks (LeCun et al., 2015), and models inspired from physics (Stark, 2016b)), they lack *interpretability*: not offering insight about the data, nor the underlying process.

The work follows recent attempts at *interpretable machine learning* (Doshi-Velez & Kim, 2017), where the lack of interpretablity may have serious consequences in mission-critical systems, ethics, and validation of computer-aided diagnosis (Doshi-Velez & Kim, 2017). While there is no consensus around the definition of interpretability, *causality* (Lipton, 2016) is a vital component: it refers to associations within the data and information about the underlying data-generating process. We adopt the latter as our 'definition' of *interpretable model*, as one where data-to-data relations are *accurately* discovered.

We propose learning a so-called *Kolmogorov Model (KM)* associated with a set of binary Random Variables (RVs). In addition to prediction, the interpretability of the model (as defined above) enables learning *causal relations* (Agrawal et al., 1993): We derive a sufficient conditions under which the realization of one RV's outcome (deterministically) *implies* the outcome of the other. In the context of recommendation systems, causal relations identify groups of items, for which a user liking one item *implies* that he/she likes all other items in the group. In cancer detection, the same rules identify groups of samples, for which the presence of DNA methylation in the group, implies its presence in all other samples. Additionally, these rules may provide insight into the physical mechanisms underlying user preferences, and DNA methylation.

We formulate the resulting problem as a *coupled combinatorial problem*, decompose it into two subproblems using the *Block-Coordinate Descent (BCD)* method, and we obtain provably optimal solutions for both. For the first one, we exploit the structure of linear programs on the unit simplex, to propose a low-complexity (yet optimal) *Frank-Wolfe* algorithm (Frank & Wolfe, 1956). To bypass the inherent complexity of the second subproblem (combinatorial and NP-hard), we propose a semidefinite relaxation, and show its *quasi-optimality* in recovering the optimal solution of the combinatorial subproblem. Finally, we show the convergence of our algorithm to a stationary point of the original problem. We refer the reader to Authors (2017) for all the derivations/discussions.

2. System Model

Notation: We use bold upper-case letters to denote matrices, bold lower-case letters to denote vectors, and calligraphic letters to denote sets. For a given matrix $\boldsymbol{A}$, $[\boldsymbol{A}]_{i,j}$ denotes element (i,j), $\text{tr}(\boldsymbol{A})$ denotes its trace, $\|\boldsymbol{A}\|_F$ its Frobenius norm, and $\boldsymbol{A}^T$ its transpose. For a vector $\boldsymbol{a}$, $[\boldsymbol{a}]_i$ denotes element i, $[\boldsymbol{a}]_{i:j}$ elements i to j, and $\text{supp}(\boldsymbol{a})$ its support. The inequality $\boldsymbol{x} \leq \boldsymbol{y}$ holds element-wise. $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix, $\mathbf{1}$ and $\mathbf{0}$ the all-one and all-zero vectors (of appropriate dimension). $\boldsymbol{e}_n$ is the nth elementary basis

[1]School of Electrical Engineering, Royal Institute of Technology, KTH, Stockholm [2]School of Electrical, Ecole Polytechnique Federale de Lausanne . Correspondence to: Hadi Ghauch <ghauch@kth.se>.

Workshop on Modern Trends in Nonconvex Optimization for Machine Learning. The 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

vector, $\mathcal{P} = \{\boldsymbol{p} \in \mathbb{R}_+^D \mid \mathbf{1}^T\boldsymbol{p} = 1\}$ the unit probability simplex, and $\{n\} = \{1, \cdots, n\}$.

2.1. Problem Formulation

Consider a double-indexed set of binary *Random Variables (RVs)*, $X_{u,i} \in \mathcal{A} = \{1, 2\}$, with indexes from $\mathcal{D} = \{(u, i) \mid u \in \mathcal{U}, i \in \mathcal{I}\}$. The RVs are defined on a sample space Ω, consisting of elementary events $\Omega = \{\omega_d \mid 1 \leq d \leq D\}$. We denote by $\mathbb{P}[X_{u,i} = z]$, $z \in \mathcal{A}$, the probability that RV $X_{u,i}$ takes the value $z \in \mathcal{A}$. Using that $\mathcal{A} = \{1, 2\}$, we write

$$\mathbb{P}[X_{u,i} = 1] = \boldsymbol{\theta}_u^T\boldsymbol{\psi}_{i,1} \text{ and } \mathbb{P}[X_{u,i} = 2] = \boldsymbol{\theta}_u^T\boldsymbol{\psi}_{i,2}, \quad (1)$$

where $\boldsymbol{\psi}_{i,1} + \boldsymbol{\psi}_{i,2} = \mathbf{1}$. $\boldsymbol{\theta}_u$ is a *Probability Mass Function (PMF)* vector on the unit simplex, $\mathcal{P}$, and $\{\boldsymbol{\psi}_{i,1}, \boldsymbol{\psi}_{i,2}\} \in \mathbb{B}^D$ are *binary indicator vectors* representing the support of its probability measure. The model follows from established results in classical probability (Gray, 2009).Since $X_{u,i}$ is binary, it is fully characterized by considering one outcome,

$$\mathbb{P}[X_{u,i} = 1] = \boldsymbol{\theta}_u^T\boldsymbol{\psi}_i, \quad (2)$$

(1) and (2) are equivalent, and will be used interchangeably (dropping the $_z$ subscript of $\boldsymbol{\psi}_i$ without any loss in generality). Thus, each RV $X_{u,i}$ is associated with (determined by) a *PMF vector* $\boldsymbol{\theta}_u$, $u \in \mathcal{U}$, and an *indicator vector* $\boldsymbol{\psi}_i$, $i \in \mathcal{I}$. Notice that the model in (2) can approximate with arbitrarily small accuracy the measure corresponding to $\mathbb{P}[]$, given a large enough D.

Problem 1 (Problem Statement) Let $p_{u,i}$ denote the empirical values of $\mathbb{P}[X_{u,i} = 1]$. We assume that $\{p_{u,i}\}$ are known for elements of a *training set* $\mathcal{K} \subseteq \mathcal{D}$, where $\mathcal{K} = \{(u, i) \mid (u, i) \in \mathcal{U} \times \mathcal{I}\}$. [1] Given samples coming from the model in (2), we wish to deduce the parameters of underlying probability distribution: find parameters of the KM, i.e., $\{\boldsymbol{\psi}_i, \boldsymbol{\theta}_u\}$ that best describe $\{p_{u,i} \mid (u, i) \in \mathcal{K}\}$. The resulting problem is a *fully parametric statistical inference* task. For tractability, we address it using the *minimum mean-squared error* as a point estimator, which in turn results in minimizing $\sum_{(u,i)\in\mathcal{K}}(\mathbb{P}[X_{u,i} = 1] - p_{u,i})^2 = \sum_{(u,i)\in\mathcal{K}}(\boldsymbol{\theta}_u^T\boldsymbol{\psi}_i - p_{u,i})^2$. Once computed, the optimal KM parameters can be used for prediction on a different set, and extracting statistical relations (among the RVs in $\mathcal{K}$). The resulting optimization problem is

$$(Q)\begin{cases} \min\limits_{\{\boldsymbol{\psi}_i\},\{\boldsymbol{\theta}_u\}} \sum\limits_{(u,i)\in\mathcal{K}} \left(\boldsymbol{\theta}_u^T\boldsymbol{\psi}_i - p_{u,i}\right)^2 \triangleq \mathcal{E} \\ \text{s.t. } \boldsymbol{\theta}_u \in \mathcal{P}\ ,\ \boldsymbol{\psi}_i \in \mathbb{B}^D,\ \forall (u, i) \in \mathcal{K} \end{cases}. \quad (3)$$

Our solution to this non-convex combinatorial problem is detailed in Section 3.

[1] Note that acquiring (estimates of) the empirical probabilities can be done via training, and the specific method is application-dependent (see Appendix A.2).

2.2. Toy Example: Recommendation Systems

In this context, $X_{u,i}$ models the preference of user u for item i, $(u, i) \in \mathcal{K}$. Thus, $\mathbb{P}[X_{u,i} = 1]$ (resp. $\mathbb{P}[X_{u,i} = 2]$) models the probability that user u likes (resp. dislikes) item i. Moreover, $\boldsymbol{\theta}_u$ determines the profile/taste of user u, $\boldsymbol{\psi}_i$ is related to item i (depending on genre, price, etc.), and the elementary events denote movie genres (e.g., $\omega_1 =$ "Action", $\omega_2 =$ "SciFi", etc.). [2] Then, the corresponding empirical probability (i.e., training set) is obtained as $p_{u,i} \triangleq [\boldsymbol{R}]_{u,i}/R_{\max}$ where $[\boldsymbol{R}]_{u,i} \in \mathbb{N}$ denotes the rating that user u has provided for item i, and $R_{\max}$ the maximum rating (Stark, 2015).

Consider a 10-star "recommendation system", having 2 users and 2 items. We then find the D-dimensional ($D = 3$) KM factorization to obtain , $\{\boldsymbol{\psi}_i\}_{i=1}^2$ and $\{\boldsymbol{\theta}_u\}_{u=1}^2$.[3] To showcase the model's intuition, note that $p_{1,1}$, the probability that user 1 likes movie 1, is represented as $\boldsymbol{\psi}_1^T\boldsymbol{\theta}_1$. It is thus expressed as *convex/stochastic* mixture of *movie genres*, since elementary events are movie genres in this scenario. More generally, a KM represents a set of observed outcomes for RVs, as mixtures of elementary events. After finding the empirical probabilities from the rated entries (as above), (Q) is solved to learn $\{\boldsymbol{\psi}_i\}_{i=1}^2$ and $\{\boldsymbol{\theta}_u\}_{u=1}^2$, and an example result is shown below:

$$\underbrace{\begin{bmatrix} 0.3 & 0.5 \\ 0.1 & 0.2 \end{bmatrix}}_{\{p_{u,i}\}} = \begin{bmatrix} \boldsymbol{\theta}_1^T\{ \\ \boldsymbol{\theta}_2^T\{ \end{bmatrix} \begin{matrix} 0.2 & 0.3 & 0.5 \\ 0.1 & 0.1 & 0.8 \end{matrix} \begin{bmatrix} 0 & 1 & \}\text{Action} \\ 1 & 1 & \}\text{SciFi} \\ \underbrace{0}_{\boldsymbol{\psi}_1} & \underbrace{0}_{\boldsymbol{\psi}_2} & \}\text{Drama} \end{bmatrix}$$

2.3. Related Work

Our proposal to model binary RVs as elementary events on a Kolmogorov space (Section 2.1) is based on established results from classical probability theory. To our best knowledge, this specific formulation is novel. Because this model is rooted in probability theory, (2) defines the outcome of a RV in the strict Kolmogorov sense (see definition in Section 2.1), and the resulting causal relations (Section 4) also hold analytically. This is the reason behind the versatility of the approach. We will also show that the combinatorial aspects of (Q) are not a limitation.

The inner product in (2) is reminiscent of factorization methods such as, *Matrix Factorization (MF)* (Koren et al., 2009), *Nonnegative Matrix Factorization (NMF)* (Lee & Seung, 2001), SVD (Cai et al., 2010), and physics-inspired techniques (e.g., Nonnegative Models (NNMs) (Stark, 2016a)).

[2] The model is generic since interpreting the elementary events in context-dependent. For a coin toss, $\Omega = \{\omega_1, \omega_2\}$, the elementary events denote heads and tails, respectively.

[3] D is the size of the Kolmogorov space Ω, the number of elementary events, and the dimension of the factorization (selected via cross-validation to minimize the test error).

However, the inner product in these methods do not model RVs, in an analytical sense (Section 2.1). Our method also generalizes K-means and some of its variants. Detailed discussions of the relation between our proposed method and these prior works is in Appendix A.1.

3. Proposed Algorithm

We use the well-known Block-Coordinate Descent (BCD) framework to handle coupling in the cost function of (Q) in (3). The method essentially splits (Q) into two subproblems. We derive different methods for each subproblem, with provable accuracy, and show convergence of the algorithm. Given $\{\boldsymbol{\psi}_i^{(n)}\}$ at iteration n, we first refine the current PMF estimate $\boldsymbol{\theta}_u$, as

$$(Q_1): \boldsymbol{\theta}_u^{(n+1)} \in \underset{\boldsymbol{\theta}_u \in \mathcal{P}}{\operatorname{argmin}} f(\boldsymbol{\theta}_u) \triangleq \boldsymbol{\theta}_u^T \boldsymbol{Q}_u^{(n)} \boldsymbol{\theta}_u - 2\boldsymbol{\theta}_u^T \boldsymbol{r}_u^{(n)},$$

where

$$\boldsymbol{Q}_u^{(n)} \triangleq \sum_{i \in \mathcal{I}_K} \boldsymbol{\psi}_i^{(n)} \boldsymbol{\psi}_i^{(n)^T},\ \boldsymbol{r}_u^{(n)} \triangleq \sum_{i \in \mathcal{I}_K} \boldsymbol{\psi}_i^{(n)} p_{u,i}. \quad (e.1)$$

We then refine the current indicator vector estimate, $\boldsymbol{\psi}_i$ as

$$(Q_2): \{\boldsymbol{\psi}_i^{(n+1)}\} \in \underset{\boldsymbol{\psi}_i \in \mathbb{B}^D}{\operatorname{argmin}} g(\boldsymbol{\psi}_i) \triangleq \boldsymbol{\psi}_i^T \boldsymbol{S}_i^{(n+1)} \boldsymbol{\psi}_i - 2\boldsymbol{\psi}_i^T \boldsymbol{v}_i^{(n+1)}$$

where

$$\boldsymbol{S}_i^{(n+1)} \triangleq \sum_{u \in \mathcal{U}_K} \boldsymbol{\theta}_u^{(n+1)} \boldsymbol{\theta}_u^{(n+1)^T}, \boldsymbol{v}_i^{(n+1)} \triangleq \sum_{u \in \mathcal{U}_K} \boldsymbol{\theta}_u^{(n+1)} p_{u,i} \quad (e.2)$$

Moreover, $\mathcal{U}_K$ and $\mathcal{I}_K$ are defined as $\mathcal{K} = \{(u,i) \mid u \in \mathcal{U}_K \subset \mathcal{U}\ ,\ i \in \mathcal{I}_K \subset \mathcal{I}\}$. Next we describe the solution approach to each problem.

3.1. Refine PMF Estimate

(Q_1) is a convex quadratic problem that can be solved by a variety of tools. However, we exploit its structure to greatly reduce the computational complexity: The Frank-Wolfe (FW) algorithm (Frank & Wolfe, 1956) solves (Q_1) as a succession of Linear Programs (LPs) over the unit simplex. While LP solvers generally have similar complexity as quadratic program solvers, solving an LP reduces to searching for the minimum index, when the LP is over the unit simplex. We formalize the algorithm, focusing on the original FW (detailed in Jaggi (2013)[Algorithm 1]).

While $\boldsymbol{\theta}_u$ should have two superscripts, n for the BCD iteration and k for the FW iteration, we only use $\boldsymbol{\theta}_u^{(k)}$. We first determine the *descent direction*:

$$\boldsymbol{d}_u^{(k)} \in \operatorname{argmin}_{\boldsymbol{s}} \left(\nabla f(\boldsymbol{\theta}_u^{(k)})\right)^T \boldsymbol{s} \ \text{ s.t. } \ \boldsymbol{s} \in \mathcal{P}\ . \quad (4)$$

The constraint $\boldsymbol{s} \in \mathcal{P}$ greatly simplifies the above LP, i.e.,

$$\boldsymbol{d}_u^{(k)} = \boldsymbol{e}_{j^\star},\ \ j^\star \in \operatorname{argmin}_{1 \le j \le D} [\nabla f(\boldsymbol{\theta}_u^{(k)})]_j\ . \quad (5)$$

The solution follows from LPs over the unit probability simplex (Proposition 2). Thus, finding the descent direction reduces to searching over the D-dimensional vector $\nabla f(\boldsymbol{\theta}_u^{(k)})$ (done in $\mathcal{O}(D)$). Then, $\boldsymbol{\theta}_u^{(k)}$ is updated using a simple step size, $\alpha_u^{(k)} = k/(k+1)$ (Jaggi, 2013). Table 1 summarizes the FW procedure, and Proposition 3 shows its convergence,

Table 1. Frank-Wolfe Procedure.

procedure $[\boldsymbol{\theta}_u^\star]$ = FWA ($\boldsymbol{Q}_u, \boldsymbol{r}_u, \epsilon$)
  for $k = 1, 2, ..., I_{FW}$ **do**
    $\boldsymbol{d}_u^{(k)} = \boldsymbol{e}_{j^\star}$, where $j^\star = \underset{1 \le j \le D}{\operatorname{argmin}} [\nabla f(\boldsymbol{\theta}_u^{(k)})]_j$
    $\boldsymbol{\theta}_u^{(k+1)} = (1 - \alpha_u^{(k)})\boldsymbol{\theta}_u^{(k)} + \alpha_u^{(k)} \boldsymbol{d}_u^{(k)}$
    Stop if $\|\boldsymbol{\theta}_u^{(k+1)} - \boldsymbol{\theta}_u^{(k)}\| \le \epsilon$
  end for
end procedure

3.2. Refine Indicator Estimate

The NP-hard nature of (Q_2) implies that relaxations are the only choice for a scalable solution. We thus propose a solution based on *Semi-Definite Relaxation (SDR)* and randomization, and establish its *quasi-optimality* for (Q_2). We use the results of Ma et al. (2002)[Sec IV-C]) and a series of reformulations to rewrite (Q_2) in its equivalent form (Authors, 2017):

$$\boldsymbol{X}_i^\star \in \begin{cases} \underset{\boldsymbol{X}_i}{\operatorname{argmin}} \operatorname{tr}(\tilde{\boldsymbol{S}}_i \boldsymbol{X}_i) \\ \text{s.t. } \boldsymbol{X}_i \succeq \boldsymbol{0},\ [\boldsymbol{X}_i]_{k,k} = 1, \forall k\ ,\ \operatorname{rank}(\boldsymbol{X}_i) = 1 \end{cases}$$

where $\boldsymbol{X}_i = \boldsymbol{x}_i \boldsymbol{x}_i^T$, $\tilde{\boldsymbol{S}}_i = \begin{bmatrix} (1/4)\boldsymbol{S}_i & -\tilde{\boldsymbol{t}}_i/2 \\ -\tilde{\boldsymbol{t}}_i^T/2 & 0 \end{bmatrix}$, $\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{z}_i \\ w_i \end{bmatrix}$, $\boldsymbol{z}_i = 2\boldsymbol{\psi}_i - \boldsymbol{1}$, $w_i \in \{-1, +1\}$ is an auxiliary variable, and $\tilde{\boldsymbol{t}}_i \triangleq (\boldsymbol{v}_i - (1/2)\boldsymbol{S}_i \boldsymbol{1})$. The above problem is then relaxed into a convex SDP,

$$\boldsymbol{X}_i^{(\text{SDR})} \in \begin{cases} \underset{\boldsymbol{X}_i}{\operatorname{argmin}} \operatorname{tr}(\tilde{\boldsymbol{S}}_i \boldsymbol{X}_i)\ . \\ \text{s.t. } \boldsymbol{X}_i \succeq \boldsymbol{0},\ [\boldsymbol{X}_i]_{k,k} = 1, \forall k \end{cases} \quad (6)$$

$\boldsymbol{X}_i^{(\text{SDR})}$ may be solved using generic SDP solvers. Then, a randomization procedure (Ma et al., 2002) extracts an approximate (binary) solution $\hat{\boldsymbol{\psi}}_i$ of (Q_2); see Table 2. This evidently raises the issue of the *suboptimality gap* for SDR. We show in Proposition 4 that SDR is optimal (asymptotically in D) in recovering the *binary solution* of (Q_2). Note that the performance bound in Proposition 4 compares the quality of the approximate *binary solution* offered by SDR, against the optimal solution of (Q_2) (rather than just comparing the resulting cost functions).

Table 2. Semidefinite Relaxation + Randomization (SDR) Proc.

procedure $[\hat{\boldsymbol{\psi}}_i]$ = SDR-wR ($\boldsymbol{S}_i, \boldsymbol{t}_i, M_{rnd}$)
  // Repeat to approximate each $\boldsymbol{\psi}_i^\star, \forall i \in \mathcal{I}_K$
  Solve (6) to find $\boldsymbol{X}_i^{(\text{SDR})}$
  Factorize as $\boldsymbol{X}_i^{(\text{SDR})} = \boldsymbol{L}_i^T \boldsymbol{L}_i$
  for $m = 1, 2, ..., M_{rnd}$ **do**
    Generate zero-mean i.i.d Gaussian vector $\boldsymbol{u}_i^{(m)}$
    Compute $\hat{\boldsymbol{u}}_i^{(m)} = \text{sign}[\ \boldsymbol{L}_i^T \boldsymbol{u}_i^{(m)}\ ]$
  end for
  Find $m^\star = \text{argmin}_{1 \le m \le D+1}\ \hat{\boldsymbol{u}}_i^{(m)^T} \tilde{\boldsymbol{S}}_i \hat{\boldsymbol{u}}_i^{(m)}$
  Compute $\hat{\boldsymbol{z}}_i = [\boldsymbol{u}_i^{(m^\star)}]_{1:D}\ [\boldsymbol{u}_i^{(m^\star)}]_{D+1}$
  Approximate $\boldsymbol{\psi}_i^\star$, as $\hat{\boldsymbol{\psi}}_i = (\hat{\boldsymbol{z}}_i + \mathbf{1})/2$
end procedure

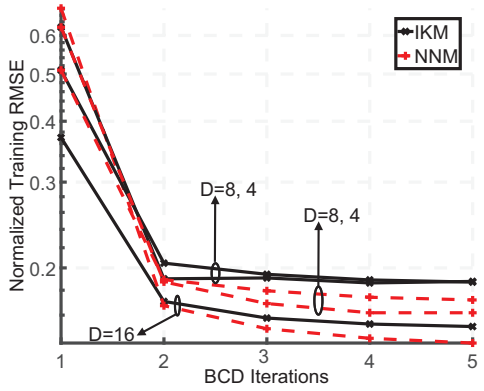

Figure 1. Training error for IKM for different values of D (ML100K dataset)

3.3. Algorithm Description

The BCD-based algorithm alternates between refining the indicator and PMF vectors (using the methods of Sec. 3.2 and Sec. 3.1; see Algorithm 1. Its convergence to a stationary point of (Q) is shown Lemma 1. Figure 1 shows the convergence behavior of Algorithm 1 for the ML100K dataset. The numerical setup and further results are provided in Appendix A.7 (due to the lack of space).

4. Interpretability via Causal Relations

Once a KM is found, we derive causal relations that emerge from (1) and (2).

4.1. Causal Relations

Proposition 1 (Inclusion of Support Set) Consider two random variables $X_{u,i}$ and $X_{u,j}$ (belonging to the training set), whose KM are given by (1). If $\text{supp}(\boldsymbol{\psi}_j) \subseteq \text{supp}(\boldsymbol{\psi}_i)$,

Algorithm 1 Iterative computation of KMs (IKM)

// Randomly Initialize $\{\boldsymbol{\theta}_u^{(1)} \in \mathcal{P}\}$
for $n = 1, 2, ...$ **do**
  Compute $\boldsymbol{S}_i^{(n)}$ and $\boldsymbol{t}_i^{(n)}$ from $(e.2)$
  Update $\hat{\boldsymbol{\psi}}_i^{(n)}$ = SDR-wR($\boldsymbol{S}_i^{(n)}, \boldsymbol{t}_i^{(n)}, M_{rnd}$), $\forall i \in \mathcal{I}_K$
  Compute $\boldsymbol{Q}_u^{(n)}$ and $\boldsymbol{r}_u^{(n)}$ from $(e.1)$
  // Initialize FWA with $\{\boldsymbol{\theta}_u^{(n-1)}\}$*, from prev iter*
  Update $\boldsymbol{\theta}_u^{(n)^\star}$ = FWA($\boldsymbol{Q}_u^{(n)}, \boldsymbol{r}_u^{(n)}, \epsilon$), for all $u \in \mathcal{U}_K$
end for

then the following *causal relations* hold:

$$X_{u,i} = 1 \text{ implies } X_{u,j} = 1 \tag{7}$$

$$X_{u,j} = 2 \text{ implies } X_{u,i} = 2\,. \tag{8}$$

For the toy example of Section 2.2, note that $\text{supp}(\boldsymbol{\psi}_1) \subseteq \text{supp}(\boldsymbol{\psi}_2)$. Then, Proposition 1 yields: if user 1 (or user 2) likes movie 2 implies he/she also likes movie 1.

Proposition 1 motivates us to look for cases where the support set condition trivially holds: when $\boldsymbol{\psi}_i = \mathbf{1}$, then $\text{supp}(\boldsymbol{\psi}_i) = \{D\}$, and $\text{supp}(\boldsymbol{\psi}_j) \subseteq \text{supp}(\boldsymbol{\psi}_i)$ holds, for any choice of $\boldsymbol{\psi}_j \forall\ j \in \mathcal{I}_K$, where $\mathcal{I}_K$ defined in Section 3.

Corollary 1 (Maximally Influential RVs) Let $\{\boldsymbol{\psi}_i,\ \boldsymbol{\theta}_u\}_{(u,i)\in\mathcal{K}}$ denote the KM associated with the outcome 1 for $X_{u,i}$, i.e., $\{X_{u,i} = 1\}_{(u,i)\in\mathcal{K}}$. We define $\mathcal{M} = \{i \mid \boldsymbol{\psi}_i = \mathbf{1}\}$ as the set of RV outcomes with maximum support. Then, the condition $\text{supp}(\boldsymbol{\psi}_j) \subseteq \text{supp}(\boldsymbol{\psi}_i)$ (Proposition 1) holds trivially $\forall\ j \in \mathcal{I}_K$. It follows that the causal relations in (7) hold, for each $i \in \mathcal{M}$.

For maximally influential RVs, the realization of one outcome, $X_{u,i} = 1$, determines that of *all RVs of the set* $\{X_{u,j} = 1 \mid \forall j \in \mathcal{I}_K\}$. It is illustrated in Figure A.1. For a recommendation system, maximally influential RVs are items for which a user liking an item implies that he/she like all other items, in the training set. We underline that, unlike other probabilistic causal relations, our approach provides deterministic rules. In Appendix A.3, we have provided efficient algorithmic approach to automatically mine these rules based on the adjacency matrix and influence score. Simulation results confirm the usefulness of our causal relations ; see Appendix A.7.

5. Conclusion

We have proposed a framework for learning a Kolmogorov model, associated with a collection of binary RVs. Interpretability of the model (as defined by causality) was harnessed by deriving causal relations, i.e., by finding sufficient conditions that bind outcomes of certain random variables. We also proposed an algorithm for computing a Kolmogorov model, a combinatorial non-convex problem, and showed

its convergence to a stationary point of the problem, using block-coordinate descent. The combinatorial nature of the problem was addressed using a semi-definite relaxation, where we showed that it yields asymptomatically optimal solutions. Our results suggest that increased interpretability and improved prediction, do not cause a significant increase in complexity.

References

Agrawal, Rakesh, Imieliński, Tomasz, and Swami, Arun. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, pp. 207–216, New York, NY, USA, 1993. ACM. ISBN 0-89791-592-5.

Authors, Anonymous. Learning elementary representations of random variables. *Journal of Machine Learning Research (submitted)*, 2017.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer-Verlag, Secaucus, NJ, USA, 2006. ISBN 0387310738.

Cai, Jian-Feng, Candès, Emmanuel J., and Shen, Zuowei. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010. doi: 10.1137/080738970.

Davenport, Mark and Romberg, Justin. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10 (4):608–622, June 2016. ISSN 1932-4553. doi: 10.1109/JSTSP.2016.2539100.

Doshi-Velez, Finale and Kim, Been. Towards a rigorous science of interpretable machine learning. *arXiv*, 2017.

Frank, Marguerite and Wolfe, Philip. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. ISSN 1931-9193.

Gantner, Zeno, Rendle, Steffen, Freudenthaler, Christoph, and Schmidt-Thieme, Lars. Mymedialite: A free recommender system library. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pp. 305–308. ACM, 2011. ISBN 978-1-4503-0683-6.

Gray, Robert M. *Probability, Random Processes, and Ergodic Properties*. Springer, 2nd edition, 2009. ISBN 1441910891, 9781441910899.

Houseman, Eugene Andres, Accomando, William P., Koestler, Devin C., Christensen, Brock C., Marsit, Carmen J., Nelson, Heather H., Wiencke, John K., and Kelsey, Karl T. DNA methylation arrays as surrogate measures of cell mixture distribution. *BMC Bioinformatics*, 13(1):86, 2012. ISSN 1471-2105.

Jaggi, Martin. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pp. 427–435, 2013.

Koren, Y., Bell, Robert, and Volinsky, Chris. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.263.

Koren, Yehuda. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining*, KDD, pp. 426–434. ACM, 2008. ISBN 978-1-60558-193-4.

LeCun, Yann, Bengio, Yoshua, and Hinton, Geoffrey. Deep learning. *Nature*, 521, 2015. doi: 10.1038/nature14539.

Lee, Daniel and Seung, Sebastian. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 556–562. MIT Press, 2001.

Lipton, Zachary Chase. The mythos of model interpretability. *CoRR*, abs/1606.03490, 2016.

Lloyd, Stuart. Least squares quantization in PCM. *IEEE Trans. Inf. Theor.*, 28(2):129–137, September 2006. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489.

Luo, Zhi-Quan, Ma, Wing-Kin, So, A.M.-C., Ye, Yinyu, and Zhang, Shuzhong. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine,*, 27(3):20–34, May 2010. ISSN 1053-5888. doi: 10.1109/MSP.2010.936019.

Ma, Wing-Kin, Davidson, T. N., Wong, Kon Max, Luo, Zhi-Quan, and Ching, Pak-Chung. Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA. *IEEE Transactions on Signal Processing*, 50(4):912–922, April 2002.

Marr, Bill. The top 10 AI and machine learning use cases everyone should know about. *Forbes Magazine*, Sept 2016.

Slawski, Martin, Hein, Matthias, and Lutsik, Pavlo. Matrix factorization with binary components. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3210–3218, 2013.

Stark, Cyril J. Expressive recommender systems through normalized nonnegative models. *CoRR*, abs/1511.04775, 2015.

Stark, Cyril J. Expressive recommender systems through normalized nonnegative models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1081–1087, 2016a.

Stark, Cyril J. Recommender systems inspired by the structure of quantum theory. *CoRR*, abs/1601.06035, 2016b.

Tan, Peng Hui and Rasmussen, L. K. The application of semidefinite programming for detection in CDMA. *IEEE Journal on Selected Areas in Communications*, 19(8): 1442–1449, Aug 2001. ISSN 0733-8716. doi: 10.1109/49.942507.

Whang, Joyce Jiyoung, Dhillon, Inderjit S., and Gleich, David F. Non-exhaustive, overlapping k-means. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pp. 936–944, 2015. doi: 10.1137/1.9781611974010.105.

A. Supplementary Material for Learning Kolmogorov Models for Binary Random Variables

A.1. Related Work

We position our work against other approaches (focusing on recommendation systems).

Factorization Methods: Note that, (Q) can be re-written as a low-rank matrix factorization problem, over the set of binary and stochastic matrices (Authors, 2017)[Sec. 9.2]. Thus, the proposed approach is connected to factorization methods: *Matrix Factorization (MF)* (Koren et al., 2009), *Nonnegative Matrix Factorization (NMF)* (Lee & Seung, 2001), SVD (Cai et al., 2010) (and their many variants/extensions) have gained widespread applicability, covering areas in sound processing, (medical) image reconstruction, recommendation systems and prediction problems (Davenport & Romberg, 2016). These techniques assume that each element in $\mathcal{K}$ is the inner product of two *arbitrary vectors*. Thus, the model in (2) does not represent a RV (in a mathematical sense), when viewed in the context of the proposed model (Section 2.1). Consequently, the analytical guarantees of Section 4, that yield the causal relations, do not hold for general factorization methods: Though causal relations may still be extracted, they are not as rooted in Kolmogorov probability theory, and lead to different statistical relations. Naturally, we wish the explore the causal relations that arise from the proposed model.

Exact Factorization: Ideally, it is desirable to solve (Q) exactly, i.e., find $\boldsymbol{\theta}_u, \boldsymbol{\psi}_i$ satisfying $p_{u,i} = \boldsymbol{\theta}_u^T \boldsymbol{\psi}_i,\ \forall (u,i) \in \mathcal{K}$, over the training set $\mathcal{K}$. While we are unaware of such results, we highlight a related variant where the factorization is solved exactly over the entire dataset $\mathcal{D}$, i.e, $p_{u,i} = \boldsymbol{\theta}_u^T \boldsymbol{\psi}_i,\ \forall (u,i) \in \mathcal{D}$, using binary MF (Slawski et al., 2013): It is not applicable when factorizing a subset of $\mathcal{D}$, e.g., the training set $\mathcal{K}$. Consequently, binary MF is unfit for prediction tasks.

KMs as a generalization of K-Means: Consider a special case of (Q), where $\boldsymbol{\psi}_i$ is constrained to have one non-zero element. The resulting problem becomes the well-known *K-means clustering* (Lloyd, 2006). The K-means algorithms (and its variants K-medoids, fuzzy K-means and K-SVD), have become pervasive in an abundance of applications such as clustering, classification, image segmentation, DNA analysis, online dictionary learning, source coding, etc. Our approach *generalizes K-means*, by allowing for overlapping clusters. While a similar generalization of the classical K-means algorithm was considered in (Whang et al., 2015), the number of points per cluster is determined explicitly. In our approach however, the number of points per cluster is optimized within the algorithm.

Nonnegative Models: *Non-Negative Models (NNMs)* (Stark, 2016a) are recent attempts at interpretable models. For reasons of computational tractability (Stark, 2016a), NNMs are defined by relaxing $\boldsymbol{\psi}_i$ in (2), to the unit hypercube. However, this relaxation *impairs* the highly interpretable nature of the original model in (2), making causal relations *less accurate*. Moreover, the relaxation implies that (2) no longer models the outcome of a random variable, thus *limiting* its applicability to (many) problems where KMs are applicable.

A.2. Applications

We briefly mention other applications.

Outage Prediction in Wireless Communication: Consider a network with several transmitters and receivers. In this setting, $X_{u,i}$ represents the state of the communication link, between transmitter u and receiver i (link (u,i)), and $\mathbb{P}[X_{u,i} = 1]$ (resp. $\mathbb{P}[X_{u,i} = 2]$) denotes the probability that it is "good" (resp. in outage). Then the corresponding KM, computed from $\mathcal{K}$, can be used to predict the state of other links, in a different set. Moreover, the causal relations in Section 4 identify links in the network, where link (u,i) good (resp. in outage) *implies* that link (u,j) good (resp. in outage). Thus the interpretability of KM provides valuable information on the network.

DNA methylation for Cancer Detection: Recent investigations have suggested that DNA methylation, chemical changes in the DNA structure, may act as a cancer detection mechanism (Houseman et al., 2012). In this context, $p_{u,i}$ denotes the measured methylation level for location i on the DNA, and sample u. DNA methylation expresses $p_{u,i} = \boldsymbol{\psi}_i^T \boldsymbol{\theta}_u$, where $\boldsymbol{\psi}_i$ is a binary vector indicating the presence or absence of DNA methylation at location i, and $\boldsymbol{\theta}_u$ is a PMF vector modeling the weight assigned to each location (Slawski et al., 2013). From the perspective of KMs, $\boldsymbol{\psi}_i^T \boldsymbol{\theta}_u$ is the *probability* that location i and sample u is methylated. Moreover, the causal relations can identify groups of DNA locations for which the presence (resp. absence) of methylation in one location, implies its presence (or absence) for all other locations in the group. Note that, the above insights are not possible using conventional methylation analysis.

A.3. Interpretable Aspects

Here we detail additional aspects of KMs, related to interpretability (via causal relations).

Adjacency Matrix and Influence Score: The causal relations of Proposition 1 can be modeled using the so-called

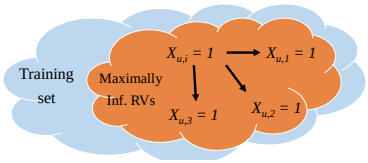

Figure A.1. causal relations for maximally influential RVs

adjacency matrix $\boldsymbol{A} \in \mathbb{B}^{|\mathcal{I}_K| \times |\mathcal{I}_K|}$, define as

$$[\boldsymbol{A}]_{i,j} = a_{i,j} = \begin{cases} 1, & \text{if } \text{supp}(\psi_j) \subseteq \text{supp}(\psi_i) \\ 0, & \text{otherwise} \end{cases} . \quad \text{(A.1)}$$

$[\boldsymbol{A}]_{i,j} = 1$ denotes the inclusion of $X_{u,j}$ in $X_{u,i}$. In this case, the first outcome of $X_{u,i}$ implies the same outcome for $X_{u,j}$, and the second outcome for $X_{u,j}$ implies the second one for $X_{u,i}$ (as stated in Proposition 1), thereby implying *coupling and mutual influence* among them (since $X_{u,i}$ influences $X_{u,j}$ and vice-versa). This raises the natural question of quantifying this coupling. We define an *influence score* essentially counting (and normalizing) the number of pairs $X_{u,i}$ and $X_{u,j}$, satisfying the support set condition,

$$\beta_i = \frac{1}{|\mathcal{I}_K|} \sum_{\substack{j \in \mathcal{I}_K \\ j \neq i}} a_{i,j} \, . \quad \text{(A.2)}$$

Thus, we provide the following method for automatically mining these rules (presented in the context of recommendation systems).

- Check the support set condition, via a pairwise search to check for pairs $\boldsymbol{\psi}_i$ and $\boldsymbol{\psi}_j$ satisfying $\text{supp}(\psi_j) \subseteq \text{supp}(\psi_i)$, $\forall (i,j) \in \mathcal{I}_K \times \mathcal{I}_K$, $i \neq j$.
- Build the adjacency matrix $\boldsymbol{A}$, in (A.1) , and compute the influence score β_i
- Find all pairs (i,j) such that $a_{i,j} = 1$: for each of these pairs the following holds (from Proposition 1),

$$\begin{cases} [u \text{ likes } i] & \text{implies } [u \text{ likes } j] \\ [u \text{ dislikes } j] & \text{implies } [u \text{ dislikes } i] \end{cases} \quad \text{(A.3)}$$

- Identify, if possible, maximally influential RVs (Corollary 1), having the all-one indicator vector, i.e., $\mathcal{M} = \{i \mid \boldsymbol{\psi}_i = \mathbf{1}\}$. For each of them, the relations in (A.3) hold for *all other items in the collection*

Practical Issues Regarding Interpretability We recall that the proposed SDR method was shown to be quasi-optimal in providing approximate *binary solutions* to (Q_2). Thus, the relaxation does not affect the interpretability, in the sense that Proposition 1 and Corollary 1 still hold. However, another remark is in order. While the derivations pertaining to causal relations (Section 4) assume globally optimal solutions to (Q) - an NP-hard problem, Algorithm 1 guarantees locally optimal ones. Thus, a bound on the gap between these solutions is needed. We highlight this issue as an interesting topic for further investigation.

A.4. Variations and Special Cases

Learning RVs with Common Support: We underline some interesting special case of the proposed approach, namely, when all the RVs have the same support, i.e., $\boldsymbol{\psi}_1 = \cdots = \boldsymbol{\psi}_D \triangleq \boldsymbol{\psi}$. This reduces to learning KMs, for RVs having common support.

Learning a sequence of RVs: Consider a special case of Sec. 2.1, where we learn a KM for a *sequence* of binary RVs, $\{X_u \mid u \in \mathcal{U}\}$, from observing samples from the training set, $\{p_u \mid u \in \mathcal{U}_K\}$. The KM in (2) reduces to $\mathbb{P}[X_u = 1] = \boldsymbol{\theta}_u^T \boldsymbol{\psi}$, and resulting optimization becomes:

$$\begin{cases} \min\limits_{\{\boldsymbol{\theta}_u\}, \boldsymbol{\psi}} \sum_{u \in \mathcal{U}_K} (\boldsymbol{\theta}_u^T \boldsymbol{\psi} - p_u)^2 \\ \text{s.t. } \boldsymbol{\theta}_u \in \mathcal{P} \, , \ \forall u \in \mathcal{U}_K, \ \boldsymbol{\psi} \in \mathbb{B}^D \end{cases} \quad \text{(A.4)}$$

The BCD-based solution approach is still applicable in this case, though many simplifications are possible.

A.5. Practical Aspects

Including Regularization Parameters: Regularization parameters for $\boldsymbol{\theta}_u$ and $\boldsymbol{\psi}_i$, are needed for prediction to avoid over-fitting (Bishop, 2006)[Sec. 1.1]. They can be included without any changes to the solution method. An ℓ_2-regularization can be included in (Q_1):

$$f(\boldsymbol{\theta}_u) = \boldsymbol{\theta}_u^T (\boldsymbol{Q}_u + \lambda_u \boldsymbol{I}_D) \boldsymbol{\theta}_u - 2 \boldsymbol{\theta}_u^T \boldsymbol{r}_u + \gamma_u \, , \quad \text{(A.5)}$$

where the regularizer $\lambda_u \geq 0$ is absorbed into a "new" matrix $(\boldsymbol{Q}_u + \lambda_u \boldsymbol{I}_D)$. While desirable, an ℓ_1-regularization for $\boldsymbol{\theta}_u$ would not work, since $\boldsymbol{\theta}_u \in \mathcal{P}$. Similarly, an ℓ_1-regularization for (Q_2) is,

$$g(\boldsymbol{\psi}_i) = \boldsymbol{\psi}_i^T \boldsymbol{S}_i \boldsymbol{\psi}_i - 2(\boldsymbol{v}_i - (\mu_i/2)\mathbf{1})^T \boldsymbol{\psi}_i + \gamma_i \, , \quad \text{(A.6)}$$

where the regularizer μ_i is absorbed into the linear term, since $\mu_i \|\boldsymbol{\psi}_i\|_1 = \mu_i \mathbf{1}^T \boldsymbol{\psi}_i$, for $\boldsymbol{\psi}_i$ binary.

Computational Complexity: The computational complexity of Algorithm 1 is dominated by the SDP solution in (6), $\approx \mathcal{O}(D^{4.5})$ for medium accuracy solutions (keeping in mind the negligible cost of the FW method). Thus, the total cost (per iteration) of Algorithm 1 is $\mathcal{C}_{\text{KM}} \approx \mathcal{O}(D^{4.5})$. The added complexity compared to MF, NMF, NNM and SVD++ ($\approx \mathcal{O}(D^3)$) is not significant, keeping in mind that $D \ll \min(|\mathcal{I}_K| \, , |\mathcal{U}_K|)$. Moreover, complexity reduction techniques (for the SDP solution) can be investigated. Finally, proposed method yields problems that decouple , thereby significantly speeding up the computation due to *parallelization*.

Non-stationary distributions: The proposed method assumes that distributions of the RVs (in the training set) are stationary: Indeed, scenarios with *time-varying distributions* are a limitation (and interesting future directions). However, in learning it is quite common to assume that the data-generating distribution is stationary.

A.6. Main Results

Below, we summarized the results used in the paper; see Authors (2017) for the proofs.

LPs over the Unit Probability Simplex: We use following known result to find the descent direction for the FW method (the proof is known).

Proposition 2 *Consider the following Linear Program (LP),*

$$(P_{PS}) \;\; \boldsymbol{x}^\star = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}} \; \boldsymbol{c}^T\boldsymbol{x}, \; s.t. \; \mathbf{1}^T\boldsymbol{x} = 1, \; \boldsymbol{x} \geq \mathbf{0}$$

Its optimal solution is given by

$$\boldsymbol{x}^\star = \boldsymbol{e}_{j^\star} \text{ , where } j^\star = \operatorname{argmin}_{1\leq j\leq n} \boldsymbol{c}^T\boldsymbol{e}_j$$

Thus, the solution reduces to searching over the vector $\boldsymbol{c}$.□

Convergence of FW algorithm: We show the convergence of the FW algorithm (Table 1).

Proposition 3 *Let* $\boldsymbol{\theta}_u^\star$ *be the optimal solution to* (Q_1). *Then the sequence of iterates* $\{\boldsymbol{\theta}_u^{(k)}\}$ *satisfies (Jaggi, 2013)[Theorem 1],*

$$\|f(\boldsymbol{\theta}_u^{(k+1)}) - f(\boldsymbol{\theta}_u^\star)\|_2 \leq \mathcal{O}(1/k), \; k = 1, 2, \cdots \square$$

Proof: The linear convergence rate for all FW variants, was proved in Jaggi (2013)[Theorem 1].

Quasi-optimality of SDR: The question was studied extensively in the context of binary detection for multi-antenna communication (Tan & Rasmussen, 2001). Interestingly, (Q_2) can be recast as a noiseless binary detection problem, where SDR has been to be optimal. The results is formalized below.

Proposition 4 *Let* $g(\boldsymbol{\psi}_i^\star)$ *and* $g(\hat{\boldsymbol{\psi}}_i)$ *denote the optimal solutions to the binary QP in* (Q_2), *and its SDR after randomization (Table 2), respectively. The approximation quality is defined as (Luo et al., 2010),*

$$\eta \leq \; g(\boldsymbol{\psi}_i^\star)/g(\hat{\boldsymbol{\psi}}_i) \; \leq 1. \qquad \text{(A.7)}$$

It holds that $\eta = 1$, *with probability* $1 - \exp^{-\mathcal{O}(D)}$, *asymptotically in* D. *Thus, the relaxation is quasi-optimal.* □

Proof: See (Authors, 2017).

Convergence of IKM:

Lemma 1 *Let* $t_n \triangleq \mathcal{E}(\{\boldsymbol{\psi}_i^{(n)}\}, \{\boldsymbol{\theta}_u^{(n)}\}), \; n = 1, 2, \ldots$ *be the sequence of iterates, resulting from the updates in IKM. Then,* $\{t_n\}$ *is non-increasing in* n, *and converges to a stationary point of* (Q) *in* (3), *almost surely.* □

Proof: The convergence is shown in (Authors, 2017).

A.7. Numerical Results

Experimental Setup: The *training set* $\mathcal{K}$, is chosen as the MovieLens 100K (ML100K), with $U = 943$ users and $I = 1682$ items, split into 80% for training and 20% for testing. Let $\{\hat{\boldsymbol{\psi}}_i\}, \{\hat{\boldsymbol{\theta}}_u\}$ the output of Algorithm 1, after 5 iterations (used to predict $p_{u,i}$ over the test set). For benchmarking, we factorize the rating matrix using MF (Koren et al., 2009), NMF (Lee & Seung, 2001), SVD++(Koren, 2008) (ensuring the dimension of the factorization, k, is close to D). The implementation and results use the MyMediaLite package (Gantner et al., 2011), and the corresponding performance results are available `http://www.mymedialite.net/examples/datasets.html`. We also benchmark against the NNM algorithm in Stark (2015), and the classical K-means (K-M) algorithm.

Training Performance: We first evaluate the performance of Algorithm 1 on artificial training data, i.e., $p_{u,i} \in \mathcal{K} = \{U = 20\} \times \{I = 40\}$ where $\{p_{u,i}\}$ are i.i.d. and uniformly chosen on the unit interval. We benchmark against a variant on Algorithm 1, where the SDR solution for Q_2 is replaced by an *exhaustive search*. As the data is artificial, the resulting matrix does not have any missing entries: we also include the *binary matrix factorization (BMF)* in Slawski et al. (2013)[Algorithm 2]. Table (3) is a numerical validation of Proposition 4 where we computed the error rate of SDR (compared to the exhaustive search), aggregated over all iterations. We observe that the approximation error decreases, with increasing D (following Proposition 4).

Table 3. Error rate for SDR ($U = 20, I = 40$).

	$D = 4$	$D = 8$	$D = 10$
SDR Accur. $\times 10^{-3}$	7.5	4.4	4.0

Following the same setup and benchmarks, Fig A.2 shows the resulting normalized training RMSE training error, RMSE $= (\sum_{(u,i)\in\mathcal{K}} |p_{u,i} - \hat{\boldsymbol{\theta}}_u^T \hat{\boldsymbol{\psi}}_i|^2/|\mathcal{K}|)^{1/2}$, for several values of D. We observe that the monotone convergence in Lemma 1 is validated numerically, and that the training error decays with increasing model size, D. While the performance of IKM (first-order optimality guarantee) is indistinguishable from its exhaustive search variant, there is large gap compared to BMF (globally optimal). Unfortunately, BMF does not work with missing data, and is inapplicable to prediction (Slawski et al., 2013). The same conclusions hold when testing Algorithm 1 on the ML100K (Fig. 1).

Interpretability of KMs: We numerically evaluate the method for finding causal relations (Section A.3), on the ML100K dataset, with the resulting influence scores shown in Fig. A.3. We first iden-

Table 4. Normalized RMSE values for test set (ML100K). The dimension of factorization for MF/SVD++, k, is equal to D (unless stated in the corresponding entry).

	$D=4$	$D=8$	$D=16$	$D=24$
KM	0.199	**0.2013**	**0.1900**	**0.1861**
NNM	**0.194**	0.2255	0.2057	0.2118
MF	0.229	$0.228(k=10)$	–	$0.226(k=40)$
SVD++	0.228	$0.227(k=10)$	$0.227(k=20)$	$0.226(k=50)$
K-M	0.210	.2096	0.2105	0.2105
NMF	–	–	–	$0.192(k=100)$

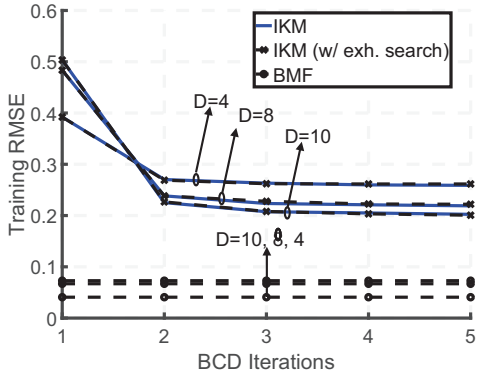

Figure A.2. Training error vs number of iterations ($U=20, I=40$)

tify the set of maximally influential items, $\mathcal{M} = \{119, 814, 1188, 1190, 1290, 1393, 1462, 1486, 1494, 1530, 1590, 1638\}$. For each of these items, a user liking one given item, implies he/she likes all other items in the training set. Interestingly, these results remain the same when $D=24$, thereby suggesting that procedure for mining causal relations is quite stable.

Prediction Performance: Since the range of the predicted variable is different for MF/NMF/SVD++, and KM/NNM, we use the normalized test RMSE, i.e., NRMSE $= \eta(\sum_{(u,i)\in\bar{\mathcal{K}}} |[\boldsymbol{R}]_{(u,i)} - \hat{R}_{u,i}|^2/|\bar{\mathcal{K}}|)^{1/2}$ where $\bar{\mathcal{K}}$ is the test set, and $\eta = (R_{\max} - R_{\min})^{-1} = 1/4$ is the normalization for MF/NMF/SVD++. For KMs/NNMs the same metric reduces to NRMSE $= \left(\sum_{(u,i)\in\bar{\mathcal{K}}} |[\boldsymbol{R}]_{(u,i)}/R_{\max} - \hat{\boldsymbol{\theta}}_u^T \hat{\boldsymbol{\psi}}_i|^2/|\bar{\mathcal{K}}|\right)^{1/2}$. The best values for λ_u and μ_i, were picked from a coarse two-dimensional grid by cross-validation, using a held-out validation set. The Normalized RMSE results are shown in Table 4. We observe a significant gap between KMs, and well known collaborative filtering methods, especially as D increases. Moreover, the drop in performance for NNMs for increasing D may be due to over-fitting.

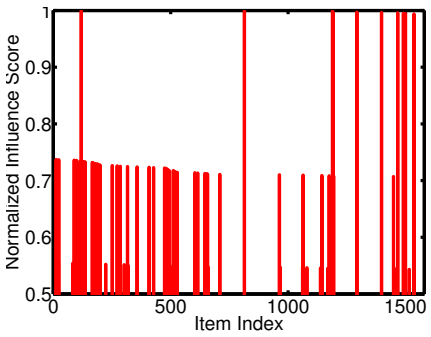

Figure A.3. Influence score for items having $\beta_i \geq 0.5$ ($D=8$, ML100K dataset)