LEARNING KOLMOGOROV MODELS FOR BINARY RANDOM VARIABLES

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a framework for learning a Kolmogorov model, for a collection of binary random variables. More specifically, we derive conditions that link (in the sense of implications in mathematical logic) outcomes of specific random variables and extract valuable relations from the data. We also propose an efficient algorithm for computing the model and show its first-order optimality, despite the combinatorial nature of the learning problem. We exemplify our general framework to recommendation systems and gene expression data. We believe that the work is a significant step toward interpretable machine learning.

1 INTRODUCTION

Machine learning and artificial intelligence method have permeated a large number of areas (Marr, Sept 2016). These methods are based on machine learning models, which consist of learning an input-output mapping for a given dataset. Despite the plethora of such models (e.g., matrix factorization (Koren et al. 2009), SVD-based models (Koren 2008), deep neural networks (LeCun et al. 2015), and models inspired from physics (Stark 2016b)), they lack interpretability: they are not capable of offering insight about the data, nor the underlying process. The lack of interpretability 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 causal relations within the data and insight about the underlying data-generating process. We thus adopt a generalized version of causality using implication in mathematical logic as our ‘definition’ of interpretable models.

To this end, 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. We derive a sufficient condition under which the realization of one RV’s outcome (causally) 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 gene expression analysis, the same relations identify groups of DNA locations for which the expression of a gene in one of them implies its expression in all other locations. The foundation of our approach is to model binary RVs as elementary events on a Kolmogorov space, by an inner product of two vectors (formally stated in Section 2.1), which is based on established results from classical probability theory. To our best knowledge, this specific formulation is novel in the context of learning representation.

The inner product of our formulation is reminiscent of factorization methods such as, matrix factorization (Koren 2008), non-negative matrix factorization (Lee & Seung 2001), SVD (Cai et al. 2010), and physics-inspired techniques, non-negative models (Stark 2016a). It should be noted that the inner product components of these methods is usually based on strong intuition about the data, provided and validated by a human expert. In contrast, the proposed KM is a fully automated approach, deeply rooted in probability theory, which finds an interpretable model (as defined above) of the data without any human intervention. Thus, both the model and the causal relations within the data have a strong mathematical basis. Moreover, in most of the existing approaches, we may need to have one pipeline for learning the representation and another one for mining these relations. However, our approach requires only a single pipeline for both tasks. This is very appealing on a

1Hereafter, for the sake of simplicity, causal relations mean logical relations within the data.
conceptual level, and it can drastically simplify coding and validation. 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.3.

In an abstract sense, we formulate a KM learning problem as a coupled combinatorial program, decompose it into two subproblems using the Block-Coordinate Descent (BCD) method, and obtain provably optimal solutions for both. For the first one, we exploit the structure of linear programs on the unit simplex and use 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 propose a simple algorithm for mining the causal relations. All the proofs and additional discussions are available in Authors (Oct 2017).

2 System Model

Notation: Lowercase letter $a$, uppercase bold letter $A$, and calligraphic letter $A$ denote vectors, matrices, and sets, respectively. $[A]_{i,j}$ and $A^T$ denote element $(i, j)$ and the transpose of $A$. $\text{supp}(a)$ denotes the support set of $a$. The inequality $x \leq y$ holds element-wise. $I$ denotes the identity matrix, $1$ and $0$ the all-one and all-zero vectors (of appropriate dimension). $e_n$ is the $n$th elementary basis, $\mathcal{P} = \{ p \in \mathbb{R}^D_+ \mid 1^T p = 1 \}$ the unit simplex, and $\{ n \} := \{ 1, \ldots, n \}$.

2.1 Problem Formulation

Consider a double-indexed set of binary Random Variables (RVs), $X_{u,i} \in \mathcal{A} = \{ 0, 1 \}$, taken from a dataset $D = \{ (u,i) \mid (u,i) \in U \times I \}$. Each RV is defined on a sample space $\Omega$, 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 $z \in \mathcal{A}$. Since $X_{u,i}$ is binary, it is fully characterized by considering one outcome. Thus, we write the Kolmogorov Model (KM) for $X_{u,i}$ as

$$\text{KM: } \mathbb{P}[X_{u,i} = 1] = \theta_{u,i}^T \psi_i.$$  

Thus, each RV $X_{u,i}$ is associated with (characterized by) a Probability Mass Function (PMF) $\theta_{u,i}, u \in U$, and an indicator vector $\psi_i, i \in I$. The model follows from established results in classical probability theory [Gray 2009] (formalized in Appendix A.2). Notice that the model in (1) can approximate with arbitrarily small accuracy the measure corresponding to $\mathbb{P}[^1]$ for a large enough $D$.

Problem 1 (Problem Statement) Let $p_{u,i}$ denote the empirical value of $\mathbb{P}[X_{u,i} = 1]$. We assume that $\{ p_{u,i} \}$ are known for elements of a training set $\mathcal{K} \subseteq D$, where $\mathcal{K} = \{ (u,i) \mid (u,i) \in U \times I \}$.

Given samples coming from the model in (1), we wish to infer the parameters of underlying probability distribution: find parameters of the KM, i.e., $\{ \psi_i, \theta_{u} \}$ 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. The corresponding optimization problem is

$$\text{(Q): } \{ \psi_i^*, \theta_{u}^* \} \in \arg \min_{\{ \psi_i, \theta_{u}(\theta_{u}) \} \in \mathcal{K}} \left( \theta_{u}^T \psi_i - p_{u,i} \right)^2 \triangleq \mathcal{E} .$$

s. t. $\theta_{u} \in \mathcal{P}, \psi_i \in \mathbb{B}^D, \forall (u,i) \in \mathcal{K}$.

(2)

We present our solution to this coupled non-convex conspiratorial optimization in the next section. The obtained solution to (Q) can be used for prediction on a test set, as well as extracting causality structures within training set (see Section 4).

Proposed Approach: While the proposal to model binary RVs as elementary events on a Kolmogorov space is based on established results, the specific learning formulation (Problem 1) is novel. Because this model is rooted in probability theory, (1) defines the outcome of a RV in the strict Kolmogorov sense, and the resulting causal relations (Section 4) also hold from a strict analytical perspective. While causal relations (or association rules) may still be extracted using existing methods, they lead to different relations (as they are based on different heuristics). Additionally, unlike existing methods, the prediction and causal relations mining are done in ’one-shot’, thereby simplifying the implementation/validation; see Appendix A.3 for detailed discussions.

The method for acquiring (estimates of) the empirical probabilities is application-dependent (see Section 5).
2.2 Illustrative Example: Recommendation Systems

In this context, \( \mathcal{U} \) and \( \mathcal{I} \) denote the set of users and items respectively, and \( X_{u,i} \) models the preference of user \( u \) for item \( i \), \((u,i) \in \mathcal{K} \). Thus, \( \mathbb{P}[X_{u,i} = 1] \) (or \( \mathbb{P}[X_{u,i} = 2] \)) is the probability that user \( u \) likes (or dislikes) item \( i \). Moreover, \( \theta_u \) determines the profile/taste of user \( u \), \( \psi_i \) 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.). The training set, consisting of an empirical probability that user \( u \) likes item \( i \), \( p_{u,i} \), is obtained from the rating that user \( u \) has provided for item \( i \). As a concrete illustrative example, consider a 10-star “recommendation system”, having 2 users and 2 items. We then find the \( D \)-dimensional (\( D = 3 \)) KM factorization to obtain \( \{ \psi_1 \}_{i=1}^2 \) and \( \{ \theta_u \}_{u=1}^2 \). \( D \) is the size of the Kolmogorov space \( \Omega \), the number of elementary events, and the dimension of the factorization (selected via cross-validation to minimize the test error). Solving \((Q)\) results in finding \( \{ \psi_1 \}_{i=1}^2 \) and \( \{ \theta_u \}_{u=1}^2 \). An example result is given below:

\[
\begin{bmatrix}
0.3 & 1 \\
0.1 & 1 \\
(p_{u,i})
\end{bmatrix} =
\begin{bmatrix}
\theta_1^T & \theta_2^T \\
\theta_1 & \theta_2 \\
\end{bmatrix}
\begin{bmatrix}
0.2 & 0.3 & 0.5 \\
0.1 & 0.8 & 0.1 \\
\end{bmatrix}
\begin{bmatrix}
0 & 1 \\
1 & 1 \\
\psi_1 & \psi_2 \\
\end{bmatrix}
\]

To showcase the model’s intuition, note that \( p_{1,1} \), the probability that user 1 likes movie 1, is represented as \( \psi_1^T \theta_1 \). It is thus expressed as convex/stochastic mixture of movie genres, since elementary events are movie genres in this scenario. We underline that this high degree of interpretability is not artificially enforced but rather context-dependent. More generally, a KM represents a set of observed outcomes for RVs as (context-dependent) mixtures of elementary events.

3 Proposed Algorithm

To approach a solution for problem (2), we use the block-coordinate descent (BCD) method to split \((Q)\) into two sub-problems. Here, we derive our solution approach to each. We first refine the current PMF estimation \( \theta_u \), and then that of the indicator \( \psi_i \). Given \( \{ \psi_i^{(n)} \} \) at iteration \( n \), we pose the PMF refinement, \( \theta \)-step, as

\[
(Q_1): \theta_u^{(n+1)} \in \underset{\theta_u \in \mathcal{P}}{\text{argmin}} \; f(\theta_u) \triangleq \theta_u^T \begin{bmatrix} \psi & Q_{u}^{(n)} \end{bmatrix} \begin{bmatrix} \theta_u - 2\theta_u^T \psi_i \psi_i^T \psi \begin{bmatrix} \psi & Q_{u}^{(n)} \end{bmatrix} \theta_u \end{bmatrix} =: \sum_{i \in \mathcal{I}_K} \psi_i^{(n)} \psi_i^{(n)T} \theta_u \psi_i^{(n)} p_{u,i}.
\]

We then pose the indicator vector refinement, \( \psi \)-step, as

\[
(Q_2): \psi_i^{(n+1)} \in \underset{\psi_i \in \mathcal{B}^D}{\text{argmin}} \; g(\psi_i) \triangleq \psi_i^T \begin{bmatrix} S_i^{(n+1)} \psi_i \end{bmatrix} \begin{bmatrix} \psi_i - 2\psi_i^T \psi_i \psi \begin{bmatrix} S_i^{(n+1)} \psi_i \end{bmatrix} \psi_i \end{bmatrix} =: \sum_{u \in \mathcal{U}_K} \theta_u^{(n+1)} \theta_u^{(n+1)T} \psi_i \psi_i^{(n+1)} p_{u,i}.
\]

Moreover, \( \mathcal{U}_K \) and \( \mathcal{I}_K \) are defined as \( \mathcal{K} = \{ (u, i) \mid u \in \mathcal{U}_K \subseteq \mathcal{U}, \; i \in \mathcal{I}_K \subseteq \mathcal{I} \} \). Recall that having globally optimal solutions for both \((Q_1)\) and \((Q_2)\) is necessary to show the convergence of BCD (Tseng [2001]) - a challenging task due to the NP-hardness of \((Q_2)\).

3.1 \( \theta \)-Step: Refine PMF Estimate

We use the Frank-Wolfe (FW) algorithm [Frank & Wolfe [1956]] to solve \((Q_1)\) 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 - which is computationally efficient, when the LP is over the unit simplex. We summarized this FW variant for solving \((Q_1)\); see also Fang [2013] [Algorithm 1]. Here, we drop the BCD iteration index, \( n \), and just keep the FW iteration number, \( k \), for notation simplicity. We first determine the descent direction: \( \mathbf{d}_u^{(k)} \in \underset{s \in \mathcal{P}}{\text{argmin}} \; \nabla f(\mathbf{d}_u^{(k)})^T s \). The constraint \( s \in \mathcal{P} \) greatly simplifies the above LP and yields: \( \mathbf{d}_u^{(k)} = \nabla f(\theta_u^{(k)})^T \mathbf{1} \). Proposition 2 (Appendix A.5). The solution follows from LPs over the unit probability simplex. Thus, finding the descent direction reduces to searching over the \( D \)-dimensional gradient vector (done in \( O(D) \)). Then, the current value is updated using a simple step size rule, \( \alpha_u^{(k)} = k/(k+1) \). Table 1 shows the \( \theta \)-step solution, and Proposition 3 in Appendix A.5 characterizes its convergence.
We highlight that the performance bound in Proposition 4 compares the quality of both the approximation algorithms.

Algorithm 1: Iterative computation of KM

```plaintext
function \( \{ \theta_u^n \} = \text{FW} (Q_u, r_u, \epsilon) 
for k = 1, 2, \ldots, I_{FW} do
    d_{kj} = e_j, j^* = \arg\min [\nabla f(\theta_u^k)]_{j}
    \theta_u^{k+1} = (1 - \alpha_u^k)\theta_u^k + \alpha_u^kd_u^k
    \text{Stop if } ||\theta_u^{k+1} - \theta_u^k|| \leq \epsilon
end for

\text{end function}
```

Table 1: \( \theta \)-step solution using FW

3.2 \( \psi \)-step: Refine Indicator Estimate

To address the NP-hard nature of (Q2), we propose a solution based on Semi-Definite Relaxation and Randomization (SDR), and establish its quasi-optimality. We use the results of Ma et al. (2002) and a series of reformulations to rewrite (Q2) in the following equivalent form (see Authors: Oct 2017) for all the derivations:

\[
X_i^* \in \arg\min_{X_i} \text{tr}(\hat{S}_iX_i), \text{ s.t. } X_i \succeq 0, [X_i]_{k,k} = 1, \forall k, \text{ rank}(X_i) = 1
\]

where \( X_i = x_i^Tx_i^T, \hat{S}_i = \begin{bmatrix} (1/4)S_i & -\hat{t}_i/2 \\ -\hat{t}_i^T/2 & 0 \end{bmatrix}, x_i = \begin{bmatrix} z_i \\ w_i \end{bmatrix}, z_i = 2\psi_i - 1, w_i \in \{-1, 1\} \) is an auxiliary variable, and \( \hat{t}_i \equiv (v_i - (1/2)S_i1) \). The problem is then relaxed into a convex program,

\[
X_i^{(SDR)} \in \arg\min_{X_i} \text{tr}(\hat{S}_iX_i), \text{ s.t. } X_i \succeq 0, [X_i]_{k,k} = 1, \forall k
\]

\( X_i^{(SDR)} \) may be solved using generic solvers. Then, a randomization procedure extracts an approximate (binary) solution for (Q2); see Table 2. This evidently raised the issue of the sub-optimality gap for SDR. Based on the results of Tan & Rasmussen (2001) and Luo et al. (2010), we show in Proposition 4 (see Appendix A.5) that SDR (Table 2) is optimal (asymptotically in D) in recovering the binary solution of (Q2).

We highlight that the performance bound in Proposition 4 compares the quality of both the approximate binary solution offered by SDR (with respect to the optimal binary solution of (Q2)), as well as their respective cost functions. The asymptotic optimality is empirically validated in Appendix A.6.

3.3 Algorithm Description

The BCD-based algorithm alternates between refining the indicator and PMF vectors; see Algorithm 1. Lemma 2 (Appendix A.5) shows its convergence to a stationary point of (Q).

Algorithm 1: Iterative computation of KM

```plaintext
// Randomly Initialize \( \{ \theta_u^{(1)} \} \in P \)
for n = 1, 2, \ldots do
    Compute \( S_i^n \) and \( t_i^n \) using \( \{ \}
    Call \( \psi_i^n = \text{SDR}(S_i^n, t_i^n, M_{rnd}) \), \( \forall i \in I_K \)
    Compute \( Q_u^n \) and \( r_u^n \) using \( \{ \}
    // Initialize FW with \( \{ \theta_u^{(n-1)} \} \), from previous iteration
    Call \( \theta_u^{(n)} = \text{FW}(Q_u^n, r_u^n, \epsilon) \), for all \( u \in U_K \)
end for
```

Table 2: \( \psi \)-step solution using SDR
4 INTERPRETABILITY VIA CAUSAL RELATION

4.1 CAUSAL RELATIONS

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

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 the model in (1). If $\text{supp}(\psi_j) \subseteq \text{supp}(\psi_i)$, then the following two causal relations hold:

$$ X_{u,i} = 1 \implies X_{u,j} = 1, \text{ and } X_{u,j} = 2 \implies X_{u,i} = 2. $$

Proof: See [Authors] (Oct 2017) for the proof.

Stated plainly, when the support set condition holds, 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 $X_{u,i}$, thereby implying a mutual causal relation among them (since $X_{u,i}$ influences $X_{u,j}$ and vice-versa). Note that our above definition of causality and causal relations is different than conventional ones in Pearl (2009) [Chap 2.8]. Moreover, our definition is distinct from Granger causality, due to the mutual coupling among the RVs in question.

We present next a special case of Proposition 1. When $\psi_1 = 1$, then $\text{supp}(\psi_1) = \{D\}$, and $\text{supp}(\psi_2) \subseteq \text{supp}(\psi_1)$ holds, for any choice of $\psi_j, \forall j \in I_K$, where $I_K$ defined in Section 3.

Corollary 1 (Maximally Supported RVs) Let $\{\psi_1, \theta_u\}_{(u,i) \in K}$ denote the KM associated with $\{P[X_{u,i} = 1]\}_{(u,i) \in K}$. We define $M$ as set of RVs for which the support set of the indicator vector is one, i.e., $M = \{i \mid \psi_i = 1\}$. Then, the condition $\text{supp}(\psi_j) \subseteq \text{supp}(\psi_i)$ (Proposition 1) holds trivially $\forall j \in I_K$. Thus, the causal relations in (6) hold, for every $i \in M$.

For maximally supported RVs, the realization of one outcome, $X_{u,i} = 1$, determines that of all RVs of the set $\{X_{u,j} = 1 \mid \forall j \in I_K\}$.

Example 1 (Causal Relations in Recommendation Systems) In addition to prediction, recommendation systems are designed to accurately mine for association rules: if a user likes item $i$ will he/she like another item $j$? Thus, the causal relations of Proposition 1 and Corollary 1 will be quite powerful, as we shall see. In the example of Section 2.2, note that $\text{supp}(\psi_1) \subseteq \text{supp}(\psi_2)$. Then, Proposition 1 yields: if user 1 (or user 2) likes movie 2 implies he/she also likes movie 1. Moreover, $X_{2,2}$ and $X_{3,2}$ are maximally supported RVs, since $\psi_2 = 1$. Thus, Corollary 1 reads: If any user likes item 2, then this causally implies that he/she likes all other items in the training set.

Thus, our approach provides association rules that follow from causal relations between different RVs. Consequently, these relations are stricter (as they are rooted in Kolmogorov probability) than other methods for mining association rules, which are not based on causality.

4.2 CAUSAL RELATIONS MINING (CRM)

In this section, we provide an efficient algorithmic approach to automatically mine the above relations. The above causal relations can be modeled using the adjacency matrix $A \in \mathbb{B}^{|I_K| \times |I_K|}$, defined as

$$ [A]_{i,j} = 1, \text{ if } \text{supp}(\psi_j) \subseteq \text{supp}(\psi_i), \text{ and } 0, \text{ otherwise, } \forall i \neq j. $$

$[A]_{i,j} = 1$ denotes the inclusion of $X_{u,j}$ in $X_{u,i}$. Note that when $A$ is sparse (resp. dense) is an indication for a dataset in which few (resp. many) causal relations exist. To quantify the sparsity level, we define an influence score, $\beta_i = |I_K|^{-1} \sum_{j \in I_K} [A]_{i,j}$, which measures the normalized number of pairs $X_{u,i}$ and $X_{u,j}$, satisfying the support set condition. These steps are summarized in the Algorithm 2 (illustrated in Figure 1). Moreover, its complexity is dominated by the pairwise search over $I_K$ (see Step 1), which comes at $O(|I_K|^2)$ operations.

5 SPECIAL CASES AND APPLICATIONS

We highlight relevant special cases and applications of our approach.
Algorithm 2 Causal Relations Mining (CRM)

1. Check the support set condition, via a pairwise search to check for pairs $\psi_i$ and $\psi_j$ satisfying supp($\psi_i$) $\subseteq$ supp($\psi_j$), $\forall (i, j) \in \mathcal{I}_K \times \mathcal{I}_K$, $i \neq j$ (done over the training set $\mathcal{K}$).
2. Build the adjacency matrix $A_i$ in (7), and compute the influence score $\beta_i$, $\forall i \in \mathcal{I}_K$.
3. Find all pairs $(i, j)$ such that $a_{i,j} = 1$. For each of these pairs it holds that (Proposition 1), $X_{u,i}$ and $X_{u,j}$ are causally related, i.e.,
   \[ X_{u,i} = 1 \implies X_{u,j} = 1, \quad \text{and} \quad X_{u,j} = 1 \implies X_{u,i} = 1 \quad \text{(8)} \]
4. Identify (if possible) maximally supported RVs (Corollary 1), $\mathcal{M} = \{ i \mid \psi_i = 1 \}$. For each of them, the relations in (8) hold for all $i \in \mathcal{I}_K$.

![Algorithm for CRM (toy example)](image)

Special Case: Unsupervised Learning Setting. Note that Algorithm 1 is equally applicable to an unsupervised learning task (no prediction needed). Moreover, if the training set has no missing data, i.e., $\mathcal{K} = \mathcal{D}$, then an alternate solution to (Q) may be obtained using the binary matrix factorization (MF) method. However, it is not applicable when factorizing a training set $\mathcal{K}$ (where $\mathcal{K} \subset \mathcal{D}$). Unlike Algorithm 1, binary MF does not offer prediction.

Application: Analysis of Gene Expression. The ability to analyze gene expression data is critical to DNA research (Zhang et al., 2009). This task can be formulated in our proposed framework as follows: $\Pr[X_{u,i} = 1]$ denotes the probability that the genes being studied are expressed in sample $u \in \mathcal{U}$ at location $i \in \mathcal{I}$ of the DNA. In this setting, the set Kolmogorov elementary events, $\{\omega_1, \ldots, \omega_D\}$, represents the various genes that might be involved. Consequently, our approach models the probability that a set of genes is expressed as a convex mixture of $D$ various genes. We recall that this intuition follows from our model and is not enforced explicitly. More importantly, the causal relations can identify groups of DNA locations for which the expression of a gene (or its absence) in a given location implies its presence (or its absence) for all other locations in the group. We will numerically show the usefulness of these relations to DNA analysis, in Section 7.2.

6 Practical Considerations

Overfitting: Regularization parameters (to mitigate overfitting) can be included without any changes to the solution method. An $\ell_2$–regularization can be included in (Q1):
   \[ f(\theta_u) = \theta_u^T (Q_u + I) \theta_u - 2 \theta_u^T r_u + \gamma_u \]
   where the regularizer $\lambda_u \geq 0$ is absorbed into a “new” matrix $(Q_u + \lambda_u I)$. Note that an $\ell_1$–regularization for $\theta_u$ would not work, since $\theta_u \in \mathcal{P}$. Similarly, an $\ell_1$-regularization for (Q2) is:
   \[ g(\psi_i) = \psi_i^T S \psi_i - 2 \psi_i - (\mu_i / 2) 1^T \psi_i + \gamma_i \]
   where the regularizer $\mu_i \geq 0$ is absorbed into the linear term, since $\mu_i \| \psi_i \|_1 = \mu_i 1^T \psi_i$, for $\psi_i$ binary.

Optimality Gap: We recall that the proposed SDR method was shown to be quasi-optimal in providing approximate binary solutions to (Q). Thus, the relaxation does not affect the interpretability, in the sense that Proposition 1 and Corollary 1 still hold. 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.

Computational Complexity: The computational complexity of Algorithm 1 is dominated by the solution in [5], which is $\approx O(D^{1.5})$ operations per iteration of Algorithm 1 (recalling the negligible cost of the FW method). The added complexity compared to matrix factorization ($\approx O(D^3)$) and its extensions is not significant, keeping in mind that $D \ll \min(|\mathcal{I}_K|, |\mathcal{U}_K|)$. Moreover, the computational complexity of Algorithm 2 consists mainly of step 1, which has $\approx O(|\mathcal{I}_K|^2)$ operations.

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 inter-
An artificial training dataset, where the training set consists of $U=20$, $I=40$, and $N_{\text{rand}}=10$. The training set is i.i.d. and uniformly chosen on the unit interval. We benchmark against a variant on Algorithm 1 where the $\psi$-step for $Q_2$ is replaced by an exhaustive search. As the data is artificial (unsupervised learning setting), we include the Binary MF in [Sawski et al. 2013, Algorithm 2].

Setting 2: The training set $\mathcal{K}$ is chosen as the MovieLens 100K, with $U=943$ users and $I=1682$ items, split into 80% for training and 20% for testing. Let $\{\hat{\psi}_u\}, \{\hat{\theta}_i\}$ be the output of Algorithm 1 after 5 iterations. We benchmark against matrix factorization (MF) [Koren et al., 2009], non-negative MF (NNM) [Lee & Seung 2001], SVD++ [Koren, 2008] (ensuring the dimension of the factorization, $k$, is close to $D$), non-negative models (NNM) [Stark, 2015], and the K-means (K-M) algorithm. The implementation and results use the MyMediaLite package [Gantner et al., 2011].

Training Performance for Unsupervised Learning (Setting 1): Fig 2a shows the resulting normalized training RMSE $\frac{\sum_{(u,i)\in \mathcal{K}}|p_{u,i} - \hat{\theta}_i^T \hat{\psi}_u|^2}{|\mathcal{K}|}\frac{1}{2}$. We observe that the monotone convergence in Lemma 2 is validated numerically, and that the training error decays with increasing model size, $D$. Note that the training performance of Algorithm 1 is indistinguishable from its exhaustive search variant. Moreover, Algorithm 1 converges to solution whose performance is similar to Binary MF, with a few iterations (except for $D=8$ where Algorithm 1 outperforms Binary MF).

Training Performance for Supervised Learning (Setting 2): Recall that Binary MF is not applicable here, due to the supervised learning setting. The same conclusions hold when testing Algorithm 1 on the ML100K (Figure 2b). We underline that while NNMs yield better training performance over Algorithm 1 the latter will have better test performance (since NNMs are indeed defined by relaxing KMs). In Table 5 (Appendix A.6), we empirically verify that the test performance of the proposed method outperforms the benchmarks in Setting 2.

Interpretability via Causal Relations (Setting 2): We numerically evaluate the relations of Algorithm 2. The influence score for each item in the training set, $\beta_i$, is shown in Figure 2c, where we displayed items with ‘high’ influence score, $\beta_i \geq 0.5$. Note that each of these high influence items is causally related to at least half of the items in the training set. This confirms the effectiveness of Algorithm 2 for finding causal relations, and that a sparse adjacency matrix is uncommon. We next identify the set of items corresponding to maximally supported RVs, $\mathcal{M} = \{119, 814, 1188, 1190, 1290, 1393, 1462, 1486, 1494, 1530, 1590, 1638\}$. 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.
7.2 Application to Gene Expression

Experimental Setup: Following the problem statement in Section 5, we show the usefulness of Algorithm 2 for gene expression. We first define another setting.

Setting 3: We use the REGED0 dataset. Element \((u, i)\) in the input matrix, \(|L|_{u,i}\), represents the level of gene expression for sample \(u\) at location \(i\) of the DNA. We obtain the training set as \(p_{u,i} := |L|_{u,i}/L_{\text{max}}\), where \(p_{u,i}\) denotes the (empirical) probability that the gene is expressed in sample \(u\) at location \(i\). After running Algorithm 1, we use Algorithm 2 to mine causal relations. As this is unsupervised learning setting, we include the binary MF (Slawski et al., 2013) method.

Results: While Figure 3a shows the training performance for several values of \(D\), Figure 3b plots the corresponding influence score that is obtained with our method. Figure 3a reveals a huge gap (≈ 3× less) between the training error of Algorithm 1 and Binary MF (unlike Figure 2a where both algorithms yield similar performance). This drastic degradation in the performance of binary MF compared to the small artificial of Figure 2a may be attributed to increasing the data/problem size (though we were unable to empirically verify this claim). Thus, we opted to use the solution of Algorithm 1 as a basis for finding causal relations. We identified 10 DNA locations corresponding to the highest \(\beta_i\) as: \{813, 250, 774, 706, 380, 49, 477, 162, 740, 702\}. For instance, the highest influence score of .1612 at DNA location 813 allowed to identify a set, \(S\), of 161 different locations which are causally related to DNA location 813. More specifically, the expression (or absence) of a given gene in location 813 implies its expression (or absence) in all other DNA locations in \(S\). While similar relations are possible using gene analysis methods, the above causal relations follow from our rigorous mathematical framework.

8 Conclusion

We have proposed a framework for learning a Kolmogorov model, associated with a collection of binary random variables. Interpretability of the model (as defined by logical implication and 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 results from block-coordinate descent. The combinatorial nature of the problem was addressed using a semi-definite relaxation. We also proposed an efficient algorithm to mine for the causal relations inherent to our model. Our results suggest that increased interpretability and improved prediction, do not cause a significant increase in complexity.

http://www.causality.inf.ethz.ch/data/REGED.html
REFERENCES


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


A SUPPLEMENTARY MATERIAL FOR LEARNING KOLMOGOROV MODELS FOR BINARY RANDOM VARIABLES

A.1 Definitions

A.2 Kolmogorov Model for a Random Variable

Let \((\Omega, \mathcal{M}, \mu)\) be a finite probability space. Here \(\Omega\) is the sample space, the event class \(\mathcal{M}\) is the power set of \(\Omega\), and \(\mu\) assigns probabilities to the sets in \(\mathcal{M}\). Let \(X_{u,i}\) denote a doubly-indexed set of random variables on the given probability space, having output alphabet in \(\mathbb{A}\), where elements of \(\mathcal{A}\) are indexed by \(z\), i.e., \(\mathcal{A}(z)\) represents the \(z\)th element in \(\mathcal{A}\). Also, let \(P[X_{u,i} = \mathcal{A}(z)] \in [0, 1]\) denote the probability that outcome \(\mathcal{A}(z)\) occurs, for \(1 \leq z \leq |\mathcal{A}|\). We let \(\Omega = \{\omega_d \mid 1 \leq d \leq D\}\), where \(\{\omega_d\}\) is the set of all \(D\)-elementary events. Since \(X_{u,i}\) is binary, i.e., \(\mathcal{A} = \{1, 2\}\), we write the KM for \(X_{u,i}\) as,

\[
\begin{align*}
P[X_{u,i} = 1] &= \theta_u^T \psi_{i,1} \\
P[X_{u,i} = 2] &= \theta_u^T \psi_{i,2} = 1 - \theta_u^T \psi_{i,1},
\end{align*}
\]

(A.1)

where \(\theta_u\) is a probability mass function (PMF) vector on the unit simplex, \(\mathcal{P}\), and \(\{\psi_{1,1}, \psi_{1,2}\} \in \mathbb{R}^D\) are binary indicator vectors representing the support of its probability measure. Moreover, the last equality follows from \(\psi_{1,1} + \psi_{1,2} = 1\), which in turn follows from the outcomes of each RV summing to one. Since \(X_{u,i}\) is binary, it is fully characterized by considering one outcome,

\[
P[X_{u,i} = 1] = \theta_u^T \psi_i,
\]

(A.2)

A.3 Related Work

Matrix Factorization Methods: Note that, \((Q)\) can be re-written as a low-rank matrix factorization problem, over the set of binary and stochastic matrices (see Appendix A.4). Thus, the proposed approach is connected to factorization methods: Matrix Factorization (MF) (Koren et al., 2009), Nonnegative Matrix Factorization (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 model elements of the training set as inner product of two arbitrary vectors, and consequently, this inner product does not represent a RV (in a mathematical sense), when viewed in the context of the proposed model (see Section 2.1). Consequently, the analytical guarantees of Section 4, which underpin the causal relations, do not hold for general factorization methods.

Binary MF: We underline that most MF methods also rely on BCD methods (alternating minimization), for which globally optimal solutions to each subproblem are needed for convergence. Thus, these methods cannot be directly applied to \((Q)\), due to the binary constraints on \(\psi_i\). The authors are unaware of any generic solution approaches for \((Q)\). Nonetheless, the Binary MF method (Slawski et al., 2013) can solve \((Q)\) when the problem is feasible and the training set spans the entire data set, i.e., \(K = D\). However, the method operates in the unsupervised learning setting only, without providing prediction (see Section 5). This indeed limits the applicability of binary MF to practical scenarios, since real-world data will have missing/erroneous data.

Clustering Methods: Consider a special case of \((Q)\), where \(\psi_i\) 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.

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 \(\psi_i\) in \((Q)\). However, this relaxation impairs the highly interpretable nature of the model in \((Q)\), making causal relations less accurate.

A.1 Definitions

A.2 Kolmogorov Model for a Random Variable

Let \((\Omega, \mathcal{M}, \mu)\) be a finite probability space. Here \(\Omega\) is the sample space, the event class \(\mathcal{M}\) is the power set of \(\Omega\), and \(\mu\) assigns probabilities to the sets in \(\mathcal{M}\). Let \(X_{u,i}\) denote a doubly-indexed set of random variables on the given probability space, having output alphabet in \(\mathbb{A}\), where elements of \(\mathcal{A}\) are indexed by \(z\), i.e., \(\mathcal{A}(z)\) represents the \(z\)th element in \(\mathcal{A}\). Also, let \(P[X_{u,i} = \mathcal{A}(z)] \in [0, 1]\) denote the probability that outcome \(\mathcal{A}(z)\) occurs, for \(1 \leq z \leq |\mathcal{A}|\). We let \(\Omega = \{\omega_d \mid 1 \leq d \leq D\}\), where \(\{\omega_d\}\) is the set of all \(D\)-elementary events. Since \(X_{u,i}\) is binary, i.e., \(\mathcal{A} = \{1, 2\}\), we write the KM for \(X_{u,i}\) as,

\[
\begin{align*}
P[X_{u,i} = 1] &= \theta_u^T \psi_{i,1} \\
P[X_{u,i} = 2] &= \theta_u^T \psi_{i,2} = 1 - \theta_u^T \psi_{i,1},
\end{align*}
\]

(A.1)

where \(\theta_u\) is a probability mass function (PMF) vector on the unit simplex, \(\mathcal{P}\), and \(\{\psi_{1,1}, \psi_{1,2}\} \in \mathbb{R}^D\) are binary indicator vectors representing the support of its probability measure. Moreover, the last equality follows from \(\psi_{1,1} + \psi_{1,2} = 1\), which in turn follows from the outcomes of each RV summing to one. Since \(X_{u,i}\) is binary, it is fully characterized by considering one outcome,

\[
P[X_{u,i} = 1] = \theta_u^T \psi_i,
\]

(A.2)

A.3 Related Work

Matrix Factorization Methods: Note that, \((Q)\) can be re-written as a low-rank matrix factorization problem, over the set of binary and stochastic matrices (see Appendix A.4). Thus, the proposed approach is connected to factorization methods: Matrix Factorization (MF) (Koren et al., 2009), Nonnegative Matrix Factorization (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 model elements of the training set as inner product of two arbitrary vectors, and consequently, this inner product does not represent a RV (in a mathematical sense), when viewed in the context of the proposed model (see Section 2.1). Consequently, the analytical guarantees of Section 4, which underpin the causal relations, do not hold for general factorization methods.

Binary MF: We underline that most MF methods also rely on BCD methods (alternating minimization), for which globally optimal solutions to each subproblem are needed for convergence. Thus, these methods cannot be directly applied to \((Q)\), due to the binary constraints on \(\psi_i\). The authors are unaware of any generic solution approaches for \((Q)\). Nonetheless, the Binary MF method (Slawski et al., 2013) can solve \((Q)\) when the problem is feasible and the training set spans the entire data set, i.e., \(K = D\). However, the method operates in the unsupervised learning setting only, without providing prediction (see Section 5). This indeed limits the applicability of binary MF to practical scenarios, since real-world data will have missing/erroneous data.

Clustering Methods: Consider a special case of \((Q)\), where \(\psi_i\) 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.

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 \(\psi_i\) in \((Q)\). However, this relaxation impairs the highly interpretable nature of the model in \((Q)\), making causal relations less accurate.
A.4 Problem Formulation in Matrix Form

Let $\Psi = [\psi_1, \ldots, \psi_I]$, $\Psi \in B^{D \times I}$ denote the aggregate indicator matrix (containing all the individual indicator vectors), $\Theta = [\theta_1, \ldots, \theta_U]$, $\Theta \in B^{D \times U}$ the aggregate matrix of PMF vectors, and $[P]_{(u, i)} = p_{(u, i)}, \forall (u, i) \in U \times I, P \in R^{U \times I}$ the aggregate matrix of known probabilities. Then, $(Q)$ can be written in equivalent matrix form,

$$
\begin{align*}
\min_{\Psi, \Theta} E(\Psi, \Theta) = \| M \circ (\Theta^T \Psi - P) \|_F^2 \\
\text{s.t.} \quad \Psi \in B^{D \times I}, \Theta \in R^{D \times U}, \Theta^T 1 = 1.
\end{align*}
$$

where $\circ$ denotes the Hadamard product, and $M \in B^{U \times I}$ is a mask matrix having $M_{u,i} = 1$, $\forall (u, i) \in K$.

A.5 Main Results

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

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),

$$(P_{PS}) \quad x^* = \arg\min_{x \in R^n} c^T x, \quad \text{s.t.} \quad 1^T x = 1, x \geq 0$$

Its optimal solution is given by

$$x^* = e_{j^*}, \quad \text{where} \quad j^* = \arg\min_{1 \leq j \leq n} c^T e_j$$

Thus, the solution reduces to searching over the vector $c$. □

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

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

$$\|f(\theta_u^{(k+1)}) - f(\theta_u^*)\|_2 \leq O(1/k), \quad k = 1, 2, \ldots$$

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 $g(\psi^*_i)$ and $g(\hat{\psi}_i)$ 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),

$$\eta \leq g(\psi^*_i)/g(\hat{\psi}_i) \leq 1.$$  \hspace{1cm} (A.4)

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

Proof: See Authors (Oct 2017).

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

Proof: The convergence is shown in Authors (Oct 2017).
A.6 ADDITIONAL NUMERICAL RESULTS

Prediction Performance (Setting 2): Since the range of the predicted variable is different for MF/NMF/SVD++, and KM/NNM, we use the normalized test RMSE, i.e., $\text{NRMSE} = \eta \left( \sum_{(u,i) \in \mathcal{K}} \left[ |R_{(u,i)} - \hat{R}_{u,i}|^2 / |\mathcal{K}| \right]^{1/2} \right)$ where $\mathcal{K}$ is the test set, and $\eta = (R_{\text{max}} - R_{\text{min}})^{-1} = 1/4$ is the normalization for MF/NMF/SVD++. For KMs/NNMs the same metric reduces to $\text{NRMSE} = \left( \sum_{(u,i) \in \mathcal{K}} \left[ |R_{(u,i)} / R_{\text{max}} - \hat{\theta}_u^T \hat{\psi}_i|^2 / |\mathcal{K}| \right]^{1/2} \right)$. The best values for $\lambda_u$ and $\mu_i$, 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 3. 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.

Table 3: Normalized Test RMSE (Setting 2). The dimension of factorization for MF/SVD++, $k$, is equal to $D$ (unless stated in the corresponding entry). The normalized test RMSE for all other methods are taken from the following repository: [http://www.mymedialite.net/examples/datasets.html](http://www.mymedialite.net/examples/datasets.html) (*- indicates the unavailability of the correspond test RMSE from the repository.

<table>
<thead>
<tr>
<th>$D$</th>
<th>KM</th>
<th>NNM</th>
<th>MF</th>
<th>SVD++</th>
<th>K-M</th>
<th>NMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.199</td>
<td><strong>0.194</strong></td>
<td>0.229</td>
<td>0.228</td>
<td>0.210</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>0.2013</td>
<td>0.2255</td>
<td>0.228</td>
<td>0.227</td>
<td>0.2096</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td><strong>0.1900</strong></td>
<td>0.2057</td>
<td>0.227</td>
<td>0.227</td>
<td>0.2105</td>
<td>-</td>
</tr>
<tr>
<td>24</td>
<td><strong>0.1861</strong></td>
<td>0.2118</td>
<td>-</td>
<td>0.226(k = 40)</td>
<td>0.2105</td>
<td>-</td>
</tr>
</tbody>
</table>

Asymptotic optimality of SDR for $\psi$-step solution: Table 4 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 4: Error rate for SDR (Artificial Dataset, $U = 20$, $I = 40$).

<table>
<thead>
<tr>
<th>$D$</th>
<th>SDR Accuracy $\times 10^{-3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>7.5</td>
</tr>
<tr>
<td>8</td>
<td>4.4</td>
</tr>
<tr>
<td>10</td>
<td>4.0</td>
</tr>
</tbody>
</table>