Boolean Matrix Factorization and Noisy Completion via Message Passing

Siamak Ravanbakhsh MRAVANBA@CS.CMU.EDU
Barnabás Póczos BAPOCZOS@CS.CMU.EDU
Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15217 USA

Russell Greiner RGREINER@UALBERTA.CA
University of Alberta, Edmonton, AB T6G 2E8, Canada

Abstract

Boolean matrix factorization and Boolean matrix completion from noisy observations are desirable unsupervised data-analysis methods due to their interpretability, but hard to perform due to their NP-hardness. We treat these problems as maximum a posteriori inference problems in a graphical model and present a message passing approach that scales linearly with the number of observations and factors. Our empirical study demonstrates that message passing is able to recover low-rank Boolean matrices, in the boundaries of theoretically possible recovery and compares favorably with state-of-the-art in real-world applications, such collaborative filtering with large-scale Boolean data.

1. Introduction

A body of problems in machine learning, communication theory and combinatorial optimization involve the product form $Z = X \odot Y$ where $\odot$ operation corresponds to a type of matrix multiplication and

$$Z = \{Z_{m,n}\}^{M \times N}, X = \{X_{m,k}\}^{M \times K}, Y = \{Y_{k,n}\}^{K \times N}.$$

Here, often one or two components (out of three) are (partially) known and the task is to recover the unknown component(s). A subset of these problems, which are most closely related to Boolean matrix factorization and matrix completion, can be expressed over the Boolean domain – *i.e.*, $Z_{m,n},\ X_{m,k},\ Y_{k,n} \in \{\text{false, true}\} \cong \{0, 1\}$. The two most common Boolean matrix products used in such applications are

$$Z = X \bullet Y \Rightarrow Z_{m,n} = \bigvee_{k=1}^{K} X_{m,k} \wedge Y_{k,n} \tag{1a}$$

$$Z = X * Y \Rightarrow Z_{m,n} \equiv \big(\sum_{k=1}^{K} X_{m,k} \wedge Y_{k,n}\big) \bmod 2 \tag{1b}$$

where we refer to Equation (1a) simply as *Boolean product* and we distinguish Equation (1b) as *exclusive-OR (XOR) Boolean product*. One may think of Boolean product as ordinary matrix product where the values that are larger than zero in the product matrix are set to one. Alternatively, in XOR product, the odd (even) numbers are identically set to one (zero) in the product matrix.

This model can represent **Low Density Parity Check (LDPC)** coding using the XOR product, with $N = 1$. In LDPC, the objective is to transmit the data vector $Y \in \{0, 1\}^K$ though a noisy channel. For this, it is encoded by Equation (1b), where $X \in \{0, 1\}^{m \times k}$ is the *parity check matrix* and vector $Z\{0, 1\}^M$ is then sent though the channel with a noise model $p^O(O \mid Z)$, producing observation O. Message passing decoding has been able to transmit Z and recover Y from O at rates close to the theoretical capacity of the communication channel (Gallager, 1962).

LDPC codes are in turn closely related to the **compressed sensing** (Donoho, 2006) – so much so that successful binary LDPC codes (*i.e.*, matrix X) have been reused for compressed sensing (Dimakis et al., 2012). In this setting, the column-vector Y is known to be ℓ-sparse (*i.e.*, ℓ non-zero values) and *approximate message passing* (AMP; Donoho et al., 2009) is used to recover Y using few noisy measurements O – that is $M \ll K$ and similar to LDPC codes, the *measurement matrix* X is known. When the underlying domain and algebra is Boolean (*i.e.*, Equation (1a)), the compressed sensing problem reduces to the problem of (noisy) **group testing** (Du & Hwang, 1993) [1] where message passing has been successfully applied in

Proceedings of the 33^rd^ International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

[1]The intuition is that the non-zero elements of the vector Y

this setting as well (Atia & Saligrama, 2012; Sejdinovic & Johnson, 2010).

These problems over Boolean domain are special instances of the problem of Boolean factor analysis in which Z is given, but not X nor Y. Here, inspired by the success of message passing techniques in closely related problems over "real" domain, we derive message passing solutions to a graphical model for "Boolean" factorization and matrix completion (*i.e.*, XOR Boolean product is not covered here), and show that simple application of Belief Propagation (BP; Pearl, 1982) to this graphical model favorably compares with the state-of-the-art in both Boolean factorization and completion.

In the following, we briefly introduce the Boolean factorization and completion problems in Section 1.1 and Section 2 reviews the related work. Section 3 formulates both of these problems in a Bayesian framework using a graphical model. The ensuing message passing solution is introduced in Section 4. Experimental study of Section 5 demonstrates that message passing is an efficient and effective method for performing Boolean matrix factorization and noisy completion.

1.1. Boolean Factor Analysis

The umbrella term "factor analysis" refers to the unsupervised methodology of expressing a set of observations in terms of unobserved factors (McDonald, 2014).[2] In contrast to LDPC and compressed sensing, in factor analysis, only (a partial and/or distorted version of) the matrix Z is observed, and our task is then to find X and Y whose product is close to Z. When the matrix Z is partially observed, a natural approach to Boolean matrix *completion* is to find sparse and/or low-rank Boolean factors that would lead us to missing elements of Z. In the following we focus on the Boolean product of Equation (1a), noting that message passing derivation for factorization and completion using the XOR product of Equation (1b) is similar.

The "Boolean" factor analysis – including factorization and completion – has a particularly appealing form. This is because the Boolean matrix Z is simply written as disjunction of Boolean matrices of rank one – that is $Z = \bigvee_{k=1}^{K} X_{:,k} \bullet Y_{k,:}$, where $X_{:,k}$ and $Y_{k,:}$ are column vector and row vectors of X and Y respectively.

identify the presence or absence of a rare property (*e.g.*, a rare disease or manufacturing defect), therefore Y is sparse. The objective is to find these non-zero elements (*i.e.*, recover Y) by screening a few ($M \ll K$) "subsets" of elements of Y. Each of these Y-bundles corresponds to a row of X (in Equation (1a)).

[2]While some definitions restrict factor analysis to variables over continuous domain or even probabilistic models with Gaussian priors, we take a more general view.

1.1.1. Combinatorial Representation

The combinatorial representation of Boolean factorization is the **biclique cover** problem in a bipartite graph $\mathcal{G} = (\mathcal{A} \cup \mathcal{B}, \mathcal{E})$. Here a bipartite graph has two disjoint node sets $\mathcal{A}$ (s.t. $|\mathcal{A}| = M$) and $\mathcal{B}$ (s.t. $|\mathcal{B}| = N$) where the only edges are between these two sets – *i.e.*, $\mathcal{E} \subseteq \{(a,b) \mid a \in \mathcal{A},\ b \in \mathcal{B}\}$. In our notation $Z \in \{0,1\}^{M \times N}$ represents the incident matrix of $\mathcal{G}$ and the objective of factorization is to cover (only) the edges using K bicliques (*i.e.*, *complete* bipartite sub-graphs of $\mathcal{G}$). Here the k^{th} biclique is identified with a subset of $\mathcal{A}$, corresponding to $X_{:,k}$, the k^{th} column of X, and a subset of $\mathcal{B}$, $Y_{k,:}$, corresponding to the k^{th} row of Y the Boolean product of which is a Boolean matrix of rank 1. The disjunction of these rank 1 matrices is therefore a biclique covering of the incident matrix Z.

2. Applications and Related Work

Many applications of Boolean factorization are inspired by its formulation as **tiling problem** (Stockmeyer, 1975).[3] Examples include mining of Boolean databases (Geerts et al., 2004), role mining (Vaidya et al., 2007; Lu et al., 2008), bi-clustering of gene expression data (Zhang et al., 2010) and approximate lifted inference with binary evidence (Van den Broeck & Darwiche, 2013). Several of these applications are accompanied by a method for approximating the Boolean factorization problem.

The most notable of these is the **"binary" factorization**[4] of Zhang et al. (2010) that uses an alternating optimization method to repeatedly solve a penalized non-negative matrix factorization problem over real-domain, where the penalty parameters try to enforce the desired binary form. Note that a binary matrix factorization is generally a more constrained problem than Boolean factorization and therefore it also provides a valid Boolean factorization.

Among the heuristics (*e.g.*, Keprt & Snásel, 2004; Belohlavek et al., 2007) that directly apply to Boolean factorization, the best known is the **Asso** algorithm of Miettinen et al. (2006). Since Asso is incremental in K, it can efficiently use the Minimum Description Length principle to select the best rank K by incrementing its value (Miettinen & Vreeken, 2011).

An important application of Boolean matrix completion is

[3]Since rows and columns in the rank one Boolean product $X_{:,k} \bullet Y_{:,k}^{T}$ can be permuted to form a "tile" – *i.e.*, a sub-matrix where all elements are equal and different from elements outside the sub-matrix – the Boolean factorization can be seen as tiling of matrix Z with tiles of rank one.

[4] Binary factorization is different from Boolean factorization in the sense that in contrast to Boolean factorization $1 + 1 \neq 1$. Therefore the factors X and Y are further constrained to ensure that Z does not contain any values other than zeros and ones.

in collaborative filtering with Boolean (*e.g.*, like/dislike) data, where the large-scale and sparsely observed Boolean matrices in modern applications demands a scalable and accurate Boolean matrix completion method.

One of the most scalable methods for this problem is obtained by modeling the problem as a **Generalized Low Rank Model** (GLRM; Udell et al., 2014), that uses proximal gradient for optimization. Using logistic or hinge loss can enforce binary values for missing entries. Using the *hinge loss*, GLRM seeks

$$\arg\min_{X,Y} \sum_{(m,n)\in\Omega} \big(1 - (\sum_k X_{m,k} Y_{k,n})(2O_{m,n} - 1)\big)_+$$

, where $(2O_{m,n} - 1)$ changes the domain of observations to $\{-1, +1\}$ and Ω is index-set of observed elements.

In the **1-Bit matrix completion** of Davenport et al. (2014), the single bit observation $O_{m,n}$ from a hidden real-valued matrix Q is obtained by sampling from a distribution with the cumulative distribution function $f(Q_{m,n})$ – *e.g.*, $\mathsf{f}(Q_{m,n}) = (1 + \exp(-Q_{m,n}))^{-1}$. For application to Boolean completion, our desired Boolean matrix is $Z = \mathbb{I}(f(Q) \geq .5)$. 1-Bit completion then minimizes the likelihood of observed entries, while constraining the nuclear norm of Q

$$\arg\min_{Q} \sum_{(m,n)\in\Omega} \Big(O_{m,n} \log(f(Q_{m,n})) + \qquad (2)$$
$$O_{m,n} \log(1 - f(Q_{m,n}))\Big) \quad s.t.\ \|Q\|_* \leq \beta\sqrt{KMN},$$

where $\beta > 0$ is a hyper-parameter.

In another recent work, Maurus & Plant (2014) introduce a method of *ternary* matrix factorization that can handle missing data in Boolean factorization through ternary logic. In this model, the ternary matrix Z is factorized to ternary product of a binary matrix X and a ternary basis matrix Y.

3. Bayesian Formulation

Expressing factorization and completion problems as a Marginal or MAP inference problem is not new (*e.g.*, Mnih & Salakhutdinov, 2007), neither is using message passing as an inference technique for these problems (Krzakala et al., 2013; Parker et al., 2013; Kabashima et al., 2014; Matsushita & Tanaka, 2013). However, these methods operate on the real-domain matrices, where AMP assumes a Gaussian distribution for BP messages. This gives an (asymptotically exact) approximation to BP updates. Here, we apply BP to solve the "Boolean" factorization/completion problem. In this setting, exact BP remains tractable, however, one needs to define the factor-graph to enforce Boolean product; see Section 3.1.

To formalize approximate decompositions for Boolean data, we use a communication channel, where we assume that the product matrix Z is communicated through a noisy binary erasure channel (Cover & Thomas, 2012) to produce the observation $O \in \{0, 1, \text{null}\}^{M\times N}$ where $O_{m,n} = \text{null}$, means this entry was erased in the channel. This allows us to model matrix completion using the same formalism that we use for low-rank factorization.

For simplicity, we assume that each element of Z is independently transmitted (that is erased, flipped or remains intact) through the channel, meaning the following conditional probability completely defines the **noise model**:

$$p^O(O \mid Z) = \prod_{m,n} {p^O}_{m,n}(O_{m,n} \mid Z_{m,n}) \qquad (3)$$

Note that each of these conditional probabilities can be represented using six values – one value per each pair of $O_{m,n} \in \{0, 1, \text{null}\}$ and $Z_{m,n} \in \{0, 1\}$. This setting allows *the probability of erasure to depend on the value of m, n and $Z_{m,n}$.*

The objective is to recover X and Y from O. However, due to its degeneracy, recovering X and Y is only up to a $K \times K$ permutation matrix U – that is $X \bullet Y = (X \bullet U) \bullet (U^T \bullet Y)$. A Bayesian approach can resolve this ambiguity by defining non-symmetric **priors**

$$p^X(X) = \prod_{m,k} {p^X}_{m,k}(X_{m,k}) \qquad (4a)$$

$$p^Y(Y) = \prod_{k,n} {p^Y}_{k,n}(Y_{k,n}) \qquad (4b)$$

where we require the a separable product form for this prior. Using strong priors can enforce sparsity of X and/or Y, leading to well-defined factorization and completion problems where $K > M, N$.

Now, we can express the problem of recovering X and Y as a *maximum a posteriori* (MAP) inference problem $\arg\max_{X,Y}\ p(X, Y \mid O)$, where the posterior is

$$p(X, Y \mid O) \;\propto\; p^X(X)\, p^Y(Y)\, p^O(O \mid X \bullet Y) \qquad (5)$$

Finding the maximizing assignment for Equation (5) is NP-hard (Stockmeyer, 1975). Here we introduce a graphical model to represent the posterior and use a simplified form of BP to approximate the MAP assignment.

An alternative to finding the MAP assignment is that of finding the marginal-MAP – *i.e.*,

$$\arg\max_{X_{m,k}} p(X_{m,k} \mid O) = \arg\max_{X_{m,n}} \sum_{X\backslash X_i, Y} p(X, Y \mid O).$$

While the MAP assignment is the optimal "joint" assignment to X and Y, finding the marginal-MAP corresponds

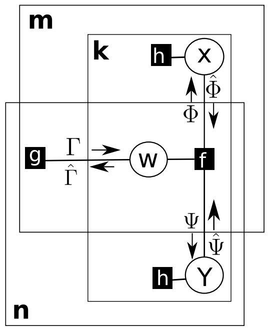

Figure 1. The factor-graph and the message exchange between variables and factors.

to optimally estimating individual assignments for each variable, while the other variable assignments are marginalized. We also provide the message passing solution to this alternative in Appendix B.

3.1. The Factor-Graph

Figure 1 shows the factor-graph (Kschischang et al., 2001) representation of the posterior Equation (5). Here, variables are circles and factors are squares. The factor-graph is a bipartite graph, connecting each factor/function to its relevant variables. This factor-graph has one variable $X_{m,k} \in \{0,1\}$ for each element of X, and a variable $Y_{k,n} \in \{0,1\}$ for each element of Y. In addition to these $K \times (M+N)$ variables, we have introduced $K \times M \times N$ auxiliary variables $W_{m,n,k} \in \{0,1\}$. For Boolean matrix completion the number of auxiliary variables is $K|\Omega|$, where $\Omega = \{(m,n)|O_{m,n} \neq \text{null}\}$ is the set of observed elements (see Section 4.1).

We use plate notation (often used with directed models) in representing this factor-graph. Figure 1 has three plates for $1 \leq m \leq M$, $1 \leq n \leq N$ and $1 \leq k \leq K$ (large transparent boxes in Figure 1). In plate notation, all variables and factors on a plate are replicated. For example, variables on the m-plate are replicated for $1 \leq m \leq M$. Variables and factors located on more than one plate are replicated for all combinations of their plates. For example, since variable X is in common between m-plate and k-plate, it refers to $M \times K$ binary variables – *i.e.*, $X_{m,k}\ \forall m,k$.

3.1.1. Variables and Factors

The auxiliary variable $W_{m,n,k}$ represents the Boolean product of $X_{m,k}$ and $Y_{k,n}$ – *i.e.*, $W_{m,n,k} = X_{m,k} \wedge Y_{k,n}$. This is achieved through $M \times N \times K$ hard constraint factors

$$\mathsf{f}_{m,n,k}(X_{m,k}, Y_{k,n}, W_{m,n,k}) = \mathbb{I}(W_{m,n,k} = X_{m,k} \wedge Y_{k,n})$$

where $\mathbb{I}(.)$ is the identity function on the inference semiring (see Ravanbakhsh & Greiner, 2014). For the max-sum inference $\mathbb{I}_{\text{max-sum}}(\text{true}) = 0$ and $\mathbb{I}_{\text{max-sum}}(\text{false}) = -\infty$.

Local factors $\mathsf{h}_{m,k}(X_{m,k}) = \log(p^X(X_{m,k}))$ and

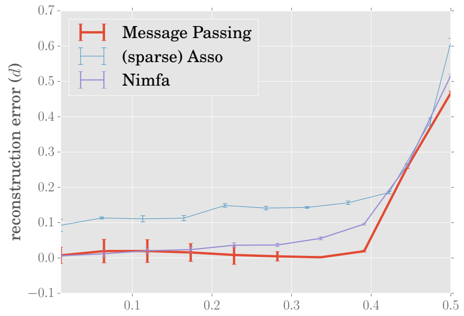

Figure 2. Comparison of message passing and NIMFA for Boolean matrix factorization

$\mathsf{h}_{k,n}(Y_{k,n}) = \log(p^Y(Y_{k,n}))$ represent the logarithm of priors over X and Y in Equation (5).

Finally, the noise model in Equation (5) is represented by $M \times N$ factors over auxiliary variables

$$\mathsf{g}_{m,n}(\{W_{m,n,k}\}_{1 \leq k \leq K}) = \log\left({p^O}_{m,n}(O_{m,n} \mid \bigvee_k W_{m,n,k})\right).$$

Although our introduction of auxiliary variables is essential in building our model, the factors of this type have been used in the past. In particular, factor g is generalized by a high-order family of factors with tractable inference, known as cardinality-based potentials (Gupta et al., 2007). This factor is also closely related to noisy-or models (Pearl, 2014; Middleton et al., 1991); where MCMC (Wood et al., 2012) and variational inference (Šingliar & Hauskrecht, 2006) has been used to solve more sophisticated probabilistic models of this nature.

The combination of the factors of type g and f, represent the term $p(O_{m,n} \mid \bigvee_{k=1}^{K} X_{m,k} \wedge Y_{k,n})$ in Equation (5) and the local factors h, represent the logarithm of the priors. It is easy to see that the sum of all the factors above, evaluates to the logarithm of the posterior

$$\log(p(X,Y \mid O) = \sum_{m,k} \mathsf{h}_{m,k}(X_{m,k}) + \sum_{k,n} \mathsf{h}_{k,n}(X_{k,n}) + \sum_{m,n} \mathsf{g}_{m,n}(\{X_{m,k} \wedge Y_{k,n}\}_{1 \leq k \leq K})$$

if $W_{m,n,k} = X_{m,k} \wedge Y_{k,n}\ \forall m,n,k$ and $-\infty$ otherwise. Therefore, maximizing the sum of these factors is equivalent to MAP inference for Equation (5).

4. Message Update

Max-sum Belief Propagation (BP) is a message passing procedure for approximating the MAP assignment in a graphical model. In factor-graphs without loops, max-sum BP is simply an exact dynamic programming approach that leverages the distributive law. In loopy factor-graphs the

Algorithm 1: message passing for Boolean matrix factorization/completion

```
Input: 1) observed matrix $O \in \{0,1\}^{M \times N}\ \forall m,n$;
2) $K \in \mathbb{N}$;
3) priors ${p^{X}}_{m,k}, {p^{Y}}_{n,k}\ \forall m,n,k$;
4) noise model ${p^{O}}_{m,n}\ \forall m,n,k$
Output: $X \in \{0,1\}^{M \times K}$ and $Y \in \{0,1\}^{K \times N}$.
$t := 0$
init $\Phi^{(t)}_{m,n,k}, \Psi^{(t)}_{m,n,k}, \hat{\Phi}^{(t)}_{m,n,k}, \hat{\Psi}^{(t)}_{m,n,k}, \hat{\Gamma}^{(t)}_{m,n,k}$ and $\Gamma^{(t)}_{m,n,k}\ \forall m,n,k$
while $t < T_{\max}$ and not converged for all $m,n,k$ do
    $\Phi^{(t+1)}_{m,n,k} := \left(\Gamma^{(t)}_{m,n,k} + \hat{\Psi}^{(t)}_{m,n,k}\right)_+ - \left(\hat{\Psi}^{(t)}_{m,n,k}\right)_+$    (6a)
    $\Psi^{(t+1)}_{m,n,k} := \left(\Gamma^{(t)}_{m,n,k} + \hat{\Phi}^{(t)}_{m,n,k}\right)_+ - \left(\hat{\Phi}^{(t)}_{m,n,k}\right)_+$    (6b)
    $\hat{\Phi}^{(t+1)}_{m,n,k} := \log\left(\frac{{p^{X}}_{m,k}(1)}{{p^{X}}_{m,k}(0)}\right) + \sum_{n' \neq n} \Phi^{(t)}_{m,n',k}$    (6c)
    $\hat{\Psi}^{(t+1)}_{m,n,k} := \log\left(\frac{{p^{Y}}_{n,k}(1)}{{p^{Y}}_{n,k}(0)}\right) + \sum_{m' \neq m} \Psi^{(t)}_{m',n,k}$    (6d)
    $\hat{\Gamma}^{(t+1)}_{m,n,k} := \min\left\{\hat{\Phi}^{(t)}_{m,n,k} + \hat{\Psi}^{(t)}_{m,n,k}, \hat{\Phi}^{(t)}_{m,n,k}, \hat{\Psi}^{(t)}_{m,n,k}\right\}$    (6e)
    $\Gamma^{(t+1)}_{m,n,k} := \min\left\{\left(-\max_{k' \neq k} \hat{\Gamma}^{(t)}_{m,n,k'}\right)_+, \sum_{k' \neq k}\left(\hat{\Gamma}^{(t)}_{m,n,k'}\right)_+ + \log\left(\frac{{p^{O}}_{m,n}(O_{m,n} \mid 1)}{{p^{O}}_{m,n}(O_{m,n} \mid 0)}\right)\right\}$    (6f)
end
calculate log-ratio of the posterior marginals
    $\Xi_{m,k} := \log\left(\frac{{p^{X}}_{m,k}(1)}{{p^{X}}_{m,k}(0)}\right) + \sum_{n} \Phi^{(t)}_{m,n,k}$    (7a)
    $\Upsilon_{k,n} := \log\left(\frac{{p^{Y}}_{k,n}(1)}{{p^{Y}}_{k,n}(0)}\right) + \sum_{m} \Psi^{(t)}_{m,n,k}$    (7b)
calculate $X$ and $Y$
    $X_{m,k} := \begin{cases} 1, & \text{if } \Xi_{m,k} > 0 \\ 0, & \text{otherwise} \end{cases}$    (8a)
    $Y_{k,n} := \begin{cases} 1, & \text{if } \Upsilon_{k,n} > 0 \\ 0, & \text{otherwise} \end{cases}$    (8b)
return $X, Y$
```

approximations of this message passing procedure is justified by the fact that it represents the zero temperature limit to the sum-product BP, which is in turn a fixed point iteration procedure whose fixed points are the local optima of the Bethe approximation to the free energy (Yedidia et al., 2000); see also (Weiss et al., 2012). For general factor-graphs, it is known that the approximate MAP solution obtained using max-sum BP is optimal within its "neighborhood" (Weiss & Freeman, 2001).

We apply max-sum BP to approximate the MAP assignment of the factor-graph of Figure 1. This factor-graph is very densely connected and therefore, one expects BP to oscillate or fail to find a good solution. However, we report in Section 5 that BP performs surprisingly well. This can be attributed to the week influence of majority of the factors, often resulting in close-to-uniform messages. Near-optimal behavior of max-sum BP in dense factor-graph is not without precedence (*e.g.*, Frey & Dueck, 2007; Ravanbakhsh et al., 2014).

The message passing for MAP inference of Equation (5) involves message exchange between all variables and their neighboring factors in both directions. Here, each message is a Bernoulli distribution. For example $\mathfrak{m}_{X_{m,k} \to \mathsf{f}_{m,n,k}}(X_{m,n}) : \{0,1\} \to \Re^2$ is the message from variable node $X_{m,n}$ to the factor node $\mathsf{f}_{m,n,k}$. For binary variables, it is convenient to work with the log-ratio of messages – *e.g.*, we use $\hat{\Phi}_{m,n,k} = \log\left(\frac{\mathfrak{m}_{X_{m,k} \to \mathsf{f}_{m,n,k}}(1)}{\mathfrak{m}_{X_{m,k} \to \mathsf{f}_{m,n,k}}(0)}\right)$ and the log-ratio of the message is opposite direction is denoted by $\hat{\Phi}$. Messages Ψ, $\hat{\Psi}$, $\hat{\Gamma}$ and Γ in Figure 1 are defined similarly. For a review of max-sum BP and the detailed derivation of the simplified BP updates for this factor-graph, see Appendix A. In particular, a naive application of BP to obtain messages $\Gamma_{m,n}$ from the likelihood factors $\mathsf{g}_{m,n}(\{W_{m,n,k}\}_{1 \leq k \leq K})\ \forall m,n$ to the auxiliary variables $W_{m,n,k}$ has a $\mathcal{O}(2^K)$ cost. In Appendix A, we show how this can be reduced to $\mathcal{O}(K)$. Algorithm 1 summarizes the simplified message passing algorithm.

At the beginning of the Algorithm, $t = 0$, messages are initialized with some random value – *e.g.*, using $\log(U) - \log(1-U)$ where $U \sim \text{Uniform}(0,1)$. Using the short notation $(a)_+ = \max\{0, a\}$, at time $t+1$, the messages are updated using 1) the message values at the previous time step t; 2) the prior; 3) the noise model and observation O. The message updates of Equation (6) are repeated until convergence or a maximum number of iterations $T_{\max}$ is reached. A possibility that we do not explore here is in using convergent alternatives of BP. We decide the convergence based on the maximum absolute change in one of the message types *e.g.*, $\max_{m,n,k} |\Phi^{(t+1)}_{m,n,k} - \Phi^{(t)}_{m,n,k}| \overset{?}{\leq} \epsilon$.

Once the message update converges, at iteration T, we can use the values for $\Phi^{(T)}_{m,n,k}$ and $\Psi^{(T)}_{m,n,k}$ to recover the log-ratio of the marginals $p(X_{m,k})$ and $p(Y_{n,k})$. These log-ratios are denoted by $\Xi_{m,k}$ and $\Upsilon_{k,n}$ in Equation (7). A positive log-ratio $\Xi_{m,k} > 0$ means $p(X_{m,k} = 1) > p(X_{m,k} = 0)$ and the posterior favors $X_{m,k} = 1$. In this way the marginals are used to obtain an approximate MAP assignment to both X and Y.

For better convergence, we also use *damping* in practice. For this, one type of messages is updated to a linear combination of messages at time t and $t+1$ using a *damping parameter* $\lambda \in (0,1]$. Choosing $\hat{\Phi}$ and $\hat{\Psi}$ for this purpose, the updates of Equations (6c) and (6d) become

$$\hat{\Phi}^{(t+1)}_{m,n,k} := (1-\lambda)\hat{\Phi}^{(t)}_{m,n,k} + \lambda\Big(\log\Big(\frac{p^X{}_{m,k}(1)}{p^X{}_{m,k}(0)}\Big) + \sum_{n' \neq n} \Phi^{(t)}_{m,n',k}\Big), \tag{9}$$
$$\hat{\Psi}^{(t+1)}_{m,n,k} := (1-\lambda)\hat{\Psi}^{(t)}_{m,n,k} + \lambda\Big(\log\Big(\frac{p^Y{}_{n,k}(1)}{p^Y{}_{n,k}(0)}\Big) + \sum_{m' \neq m} \Psi^{(t)}_{m',n,k}\Big).$$

4.1. Further Simplifications

Partial knowledge. If any of the priors, $p(X_{m,k})$ and $p(Y_{n,k})$, are zero or one, it means that X and Y are partially known. The message updates of Equations (6c) and (6d) will assume $\pm\infty$ values, to reflect these hard constrains. In contrast, for uniform priors, the log-ratio terms disappear.

Matrix completion speed up. Consider the case where $\log\big(\frac{p^O(O_{m,n}|1)}{p^O(O_{m,n}|0)}\big) = 0$ in Equation (6f) – *i.e.*, the probabilities in the nominator and denominator are equal. An important case of this happens in matrix completion, when the probability of erasure is independent of the value of $Z_{m,n}$ – that is $p^O(\text{null} \mid Z_{m,n}=0) = p^O(\text{null} \mid Z_{m,n}=1) = p^O(\text{null})$ for all m and n.

It is easy to check that in such cases, $\Gamma_{m,n,k} = \min\big((-\max_{k' \neq k} \hat{\Gamma}^{(t)}_{m,n,k})_+, \sum_{k' \neq k} (\hat{\Gamma}^{(t)}_{m,n,k})_+\big)$ is always zero. This further implies that $\hat{\Phi}_{m,n,k}$ and $\hat{\Psi}_{m,n,k}$ in Equations (6c) and (6d) are also always zero and calculating $\hat{\Gamma}_{m,n,k}$ in Equation (6f) is pointless. The bottom-line is that we only need to keep track of messages where this log-ratio is non-zero. Recall that $\Omega = \{(m,n) \mid O_{m,n} \neq \text{null}\}$ denote the observed entries of O. Then in the message passing updates of Equation (6) in Algorithm 1, wherever the indices m and n appear, we may restrict them to the set Ω.

Belief update. Another trick to reduce the complexity of message updates is in calculating $\{\hat{\Phi}_{m,n,k}\}_n$ and $\{\hat{\Psi}_{m,n,k}\}_m$ in Equations (6c) and (6d). We may calculate the marginals $\Xi_{m,k}$ and $\Upsilon_{k,n}$ using Equation (7), and replace the Equation (9), the damped version of the Equations (6c) and (6d), with

$$\hat{\Phi}^{(t+1)}_{m,n,k} := (1-\lambda)\hat{\Phi}^{(t)}_{m,n,k} + \lambda\big(\Xi^{(t)}_{m,k} - \Phi^{(t)}_{m,n,k}\big) \tag{10a}$$
$$\hat{\Psi}^{(t+1)}_{m,n,k} := (1-\lambda)\hat{\Psi}^{(t)}_{m,n,k} + \lambda\big(\Upsilon^{(t)}_{k,n} - \Psi^{(t)}_{m,n,k}\big) \tag{10b}$$

where the summation over n' and m' in Equations (6c) and (6d) respectively, is now performed only once (in producing the marginal) and reused.

Recycling of the max. Finally, using one more computational trick the message passing cost is reduced to linear: in Equation (6e), the maximum of the term $\big(-\max_{k' \neq k} \hat{\Gamma}^{(t)}_{m,n,k}\big)_+$ is calculated for each of K messages $\{\Gamma_{m,n,k}\}_{k \in \{1,\ldots,K\}}$. Here, we may calculate the "two" largest values in the set $\{\hat{\Gamma}^{(t)}_{m,n,k}\}_k$ only once and reuse them in the updated for all $\{\Gamma_{m,n,k}\}_k$ – *i.e.*, if the largest value is $\hat{\Gamma}^{(t)}_{m,n,k^*}$ then we use the second largest value, only in producing Γ_{m,n,k^*}.

Computational Complexity. All of the updates in (6a,6b,6f,6e,10) have a constant computational cost. Since these are performed for $K|\Omega|$ messages, and the updates in calculating the marginals Equations (7a) and (7b) are $\mathcal{O}(K|\Omega|)$, the complexity of one iteration is $\mathcal{O}(K|\Omega|)$.

5. Experiments

We evaluated the performance of message passing on random matrices and real-world data.[5] In all experiments, message passing uses damping with $\lambda = .4$, $T = 200$ iterations and uniform priors $p^X{}_{m,k}(1) = p^Y{}_{k,n}(1) = .5$. This also means that if the channel is symmetric – that is $p^O(1 \mid 1) = p^O(0 \mid 0) > .5$ – the approximate MAP reconstruction $\widehat{Z}$ does not depend on p^O, and we could simply use $p^O{}_{m,n}(1 \mid 1) = p^O{}_{m,n}(1 \mid 1) = c$ for any $c > .5$. The only remaining hyper-parameters are rank K and maximum number of iterations T.

5.1. Random Matrices

Matrix Factorization. We compared our method against binary matrix factorization method of (Zhang et al., 2007), which was implemented by NIMFA (Zitnik & Zupan, 2012) as well as (sparse) Asso of Miettinen et al. (2006). Here, all methods receive the correct K as input.

Figure 2 compares the reconstruction error of different methods at different noise levels. The results are for 1000×1000 random matrices of rank $K = 5$ where X and Y were uniformly sampled from binary matrices. The results for different K show a similar trend.[6] The *reconstruction error* is

$$d(Z, \widehat{Z}) \overset{\text{def}}{=} \frac{1}{MN} \sum_{m,n} |Z_{m,n} - \widehat{Z}_{m,n}|. \tag{11}$$

The results suggests that message passing and NIMFA are competitive, with message passing performing better

[5]The Python implementation is available at `https://github.com/mravanba/BooleanFactorization`

[6]Both message passing and NIMFA use the same number of iterations $T = 200$. For NIMFA we use the default parameters of $\lambda_h = \lambda_w = 1.1$ and initialize the matrices using SVD. For Asso we report the result for the best threshold hyper-parameter $\tau \in \{.10, .31, .52, .74, .95\}$.

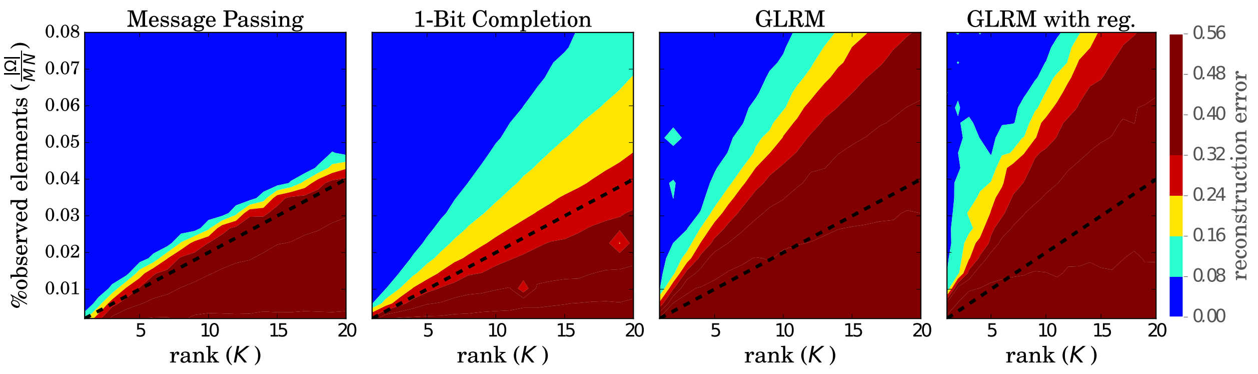

Figure 3. The matrix completion error for Message Passing, 1-Bit matrix completion and GLRM (with and without regularization) as a function of matrix rank and portion of observed elements $|\Omega|$ for $M = N = 1000$. The dashed black line indicates the tentative information bottleneck.

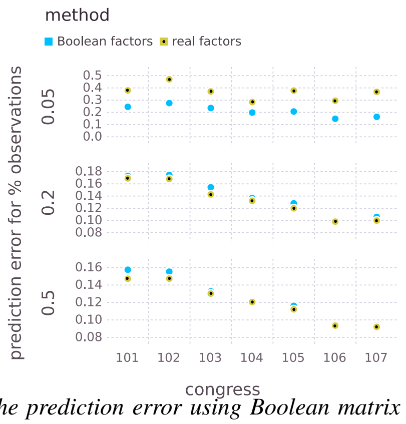

Figure 4. The prediction error using Boolean matrix completion (by message passing) versus using GLRM with hinge loss for binary matrix completion using real factors. Each panel has a different observed percentage of entries $\frac{|\Omega|}{MN} \in \{.05, .2, .5\}$. Here the horizontal axis identifies senator×issue matrices and the y-axis is the average error in prediction of the unobserved portion of the (yes/no) votes.

at higher noise levels. The experiments were repeated 10 times for each point. The small variance of message passing performance at low noise-levels is due to the multiplicity of symmetric MAP solutions, and could be resolved by performing decimation, albeit at a computational cost. We speculate that the symmetry breaking of higher noise levels help message passing choose a fixed point, which results in lower variance. Typical running times for a single matrix in this setting are 2, 15 and 20 seconds for NIMFA, message passing and sparse Asso respectively.[7]

Despite being densely connected, at lower levels of noise, BP often converges within the maximum number of iterations. The surprisingly good performance of BP, despite the large number of loops, is because most factors have a weak influence on many of their neighboring variables. This effectively limits the number of influential loops in the factor-graph; see Appendix C for more.

[7]Since sparse Asso is repeated 5 times for different hyper-parameters, its overall run-time is 100 seconds.

Matrix Completion. The advantage of message passing to its competition is more evident in matrix "completion" problem, where the complexity of BP grows with the number of observed elements, rather than the size of matrix Z. We can "approximate" a lower-bound on the *number of observed entries* $|\Omega| = MN(1 - p^O(\text{null}))$ required for recovering Z by

$$|\Omega| > K(M + N - \log(K) + 1) + \mathcal{O}(\log(K)). \quad (12)$$

To derive this approximation, we briefly sketch an information theoretic argument. Note that the total number of ways to define a Boolean matrix $Z \in \{0,1\}^{M\times N}$ of rank K is $\frac{2^{K(M+N)}}{K!}$, where the nominator is the number of different X and Y matrices and $K!$ is the irrelevant degree of freedom in choosing the permutation matrix U, such that $Z = (X \bullet U) \bullet (U^T \bullet Y)$. The logarithm of this number, using Sterling's approximation, is the r.h.s. of Equation (12), lower-bounding the number of bits required to recover Z, in the absence of any noise. Note that this is assuming that any other degrees of freedom in producing Z grows sub-exponentially with K – *i.e.*, is absorbed in the additive term $\mathcal{O}(\log(K))$. This approximation also resembles the $\mathcal{O}(KN\text{polylog}(N))$ sample complexity for various real-domain matrix completion tasks (*e.g.*, Candes & Plan, 2010; Keshavan et al., 2010).

Figure 3 compares message passing against GLRM and 1-Bit matrix completion. In all panels of Figure 3, each point represents the average reconstruction error for random 1000×1000 Boolean matrices. For each choice of observation percentage $\frac{|\Omega|}{MN}$ and rank K, the experiments were repeated 10 times.[8] The dashed black line is the information theoretic approximate lower-bound of Equation (12). This result suggests that message passing outper-

[8]This means each figure summarizes $20\,(\text{rank}) \times 20\,(\text{number of observations}) \times 10\,(\text{repeats}) = 4000$ experiments. The exception is 1-Bit matrix completion, where due to its longer run-time the number of repetition was limited to two. The results for 1-Bit completion are for best $\beta \in \{.1, 1, 10\}$.

Table 1. Matrix completion performance for MovieLense dataset.

		time (sec)	binary	**observed** percentage of available ratings					
		min-max	input?	1%	5%	10%	20%	50%	95%
1M-dataset	message passing	2-43	Y	**56%**	65%	67%	69%	71%	71%
	GLRM (ordinal hinge)	2-141	N	48%	65%	68%	70%	71%	72%
	GLRM (logistic)	4-90	Y	46%	63%	63%	63%	63%	62%
100K-dataset	message passing	0-2	Y	**52%**	**60%**	**63%**	65%	67%	70%
	GLRM (ordinal hinge)	0-2	N	48%	58%	63%	67%	69%	70%
	GLRM (logistic)	0-2	Y	45%	50%	62%	63%	62%	67%
	1-bit completion	**30-500**	Y	50%	53%	61%	65%	70%	72%

forms both of these methods and remains effective close to this bound. Figure 3 also suggests that, when using message passing, the transition from recoverability to non-recoverability is sharp. Indeed the variance of the reconstruction error is always close to zero, but in a small neighborhood of the dashed black line.[9]

5.2. Real-World Applications

This section evaluates message passing on two real-world applications. While there is no reason to believe that the real-world matrices must necessarily decompose into low-rank Boolean factors, we see that Boolean completion using message passing performs well in comparison with other methods that assume Real factors.

MovieLens Dataset. We applied our message passing method to MovieLens-1M and MovieLens-100K dataset[10] as an application in *collaborative filtering*. The Movie-Lense-1M dataset contains 1 million ordinals (1-5) ratings from 6000 users on 4000 movies (*i.e.*, $1/24$ of all the ratings are available). Here we say a user is "interested" in the movie iff her rating is above the global average of ratings. The task is to predict this single bit by observing a random subset of the available user$\times$movie rating matrix. For this, we use $\alpha \in (0, 1)$ portion of the 10^6 ratings to predict the one-bit interest level for the remaining ($1 - \alpha$ portion of the) data-points. Note that here $|\Omega| = \frac{\alpha M N}{24}$. The same procedure is applied to the smaller Movie-Lens-100K dataset. The reason for including this dataset was to compare message passing performance with 1-Bit matrix completion that does not scale as well. We report the results using GLRM with logistic and *ordinal* hinge loss (Rennie & Srebro, 2005) and quadratic regularization of the factors. [11] Here, only GLRM with ordinal hinge loss uses actual ratings (non-binary) to predict the ordinal ratings which are then thresholded.

[9]The sparsity of Z is not apparent in Figure 3. Here, if we generate X and Y uniformly at random, as K grows, the matrix $Z = X \bullet Y$ becomes all ones. To avoid this degeneracy, we choose $p^X{}_{m,k}(X_{m,k})$ and $p^Y{}_{k,n}(Y_{k,n})$ so as to enforce $p(Z = 1) \approx p(Z = 0)$. It is easy to check that $p^X{}_{m,k}(1) = p^Y{}_{k,n}(1) = \sqrt{1 - \sqrt[K]{.5}}$ produces this desirable outcome. Note that these probabilities are only used for random matrix "generation" and the message passing algorithm is using uniform priors.

[10]`http://grouplens.org/datasets/movielens/`

[11]The results reported for 1-Bit matrix completion are for best $\beta \in \{.1, 1, 10\}$ (see Equation (2)). The results for GLRM are for the regularization parameter in $\{.01, .1, 1, 10\}$ with the best test error.

Table 1 reports the run-time and test error of all methods for $K = 2$, using different $\alpha \in \{.01, .05, .1, .2, .5, .95\}$ portion of the available ratings. It is surprising that only using one bit of information per rating, message passing and 1-bit completion are competitive with ordinal hinge loss that benefits from the full range of ordinal values. The results also suggest that when only few observations are available (*e.g.*, $\alpha = .01$), message passing performs better than all other methods. With larger number of binary observations, 1-bit completion performs slightly better than message passing, but it is orders of magnitude slower. Here, the variance in the range of reported times in Table 1 is due to variance in the number of observed entries – *i.e.*, $\alpha = .01$ often has the smallest run-time.

Reconstructing Senate Voting Records. We applied our noisy completion method to predict the (yes/no) senate votes during 1989-2003 by observing a randomly selected subset of votes.[12] This dataset has 7 Boolean matrices (corresponding to voting sessions for $101^{st} - 107^{th}$ congress), where a small portion of entries are missing. For example, the first matrix is a 634×103 Boolean matrix recording the vote of 102 senators on 634 topics plus the outcome of the vote (which we ignore). Figure 4 compares the reconstruction error of message passing and GLRM (with hinge loss or binary predictions) for the best choice of $K \in \{1, \ldots, 10\}$ on each of 7 matrices. [13] In each case we report the prediction accuracy on the unobserved entries, after observing $\frac{|\Omega|}{MN} \in \{5\%, 20\%, 50\%\}$ of the votes. For sparse observations ($\frac{|\Omega|}{MN} = .05$), the message passing error is almost always half of the error when we use real factors. With larger number of observations, the methods are comparable, with GLRM performing slightly better.

Conclusion

This paper introduced a simple message passing technique for approximate Boolean factorization and noisy matrix completion. While having a linear time complexity, this procedure favorably compares with the state-of-the-art in Boolean matrix factorization and completion. In particular, for matrix completion with few entries, message passing significantly outperforms the existing methods that use real factors. This makes message passing a useful candidate for collaborative filtering in modern applications involving large datasets of sparse Boolean observations.

[12]The senate data was obtained from `http://www.stat.columbia.edu/~jakulin/Politics/senate-data.zip` prepared by Jakulin et al. (2009).

[13]GLRM is using quadratic regularization while message passing is using uniform priors.

Acknowledgements

We would like to thank Pauli Miettinen for sharing his code and pointers to relevant works on the topic. Also, we thank anonymous reviewers for their constructive feedback.

References

Atia, George K and Saligrama, Venkatesh. Boolean compressed sensing and noisy group testing. *Information Theory, IEEE Transactions on*, 58(3):1880–1901, 2012.

Belohlavek, Radim, Dvořák, Jiří, and Outrata, Jan. Fast factorization by similarity in formal concept analysis of data with fuzzy attributes. *Journal of Computer and System Sciences*, 73(6):1012–1022, 2007.

Candes, Emmanuel J and Plan, Yaniv. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Cover, Thomas M and Thomas, Joy A. *Elements of information theory*. John Wiley & Sons, 2012.

Davenport, Mark A, Plan, Yaniv, van den Berg, Ewout, and Wootters, Mary. 1-bit matrix completion. *Information and Inference*, 3(3):189–223, 2014.

Dimakis, Alexandros G, Smarandache, Roxana, and Vontobel, Pascal O. Ldpc codes for compressed sensing. *Information Theory, IEEE Transactions on*, 58(5):3093–3114, 2012.

Donoho, David L. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

Donoho, David L, Maleki, Arian, and Montanari, Andrea. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

Du, Ding-Zhu and Hwang, Frank K. *Combinatorial group testing and its applications*. World Scientific, 1993.

Frey, Brendan J and Dueck, Delbert. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

Gallager, Robert G. Low-density parity-check codes. *Information Theory, IRE Transactions on*, 8(1):21–28, 1962.

Geerts, Floris, Goethals, Bart, and Mielikäinen, Taneli. Tiling databases. In *Discovery science*, pp. 278–289. Springer, 2004.

Gupta, Rahul, Diwan, Ajit A, and Sarawagi, Sunita. Efficient inference with cardinality-based clique potentials. In *Proceedings of the 24th international conference on Machine learning*, pp. 329–336. ACM, 2007.

Jakulin, Aleks, Buntine, Wray, La Pira, Timothy M, and Brasher, Holly. Analyzing the us senate in 2003: Similarities, clusters, and blocs. *Political Analysis*, pp. mpp006, 2009.

Kabashima, Yoshiyuki, Krzakala, Florent, Mézard, Marc, Sakata, Ayaka, and Zdeborová, Lenka. Phase transitions and sample complexity in bayes-optimal matrix factorization. *arXiv preprint arXiv:1402.1298*, 2014.

Keprt, Ales and Snásel, Václav. Binary factor analysis with help of formal concepts. In *CLA*, volume 110, pp. 90–101, 2004.

Keshavan, Raghunandan H, Montanari, Andrea, and Oh, Sewoong. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.

Krzakala, Florent, Mézard, Marc, and Zdeborová, Lenka. Phase diagram and approximate message passing for blind calibration and dictionary learning. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pp. 659–663. IEEE, 2013.

Kschischang, Frank R, Frey, Brendan J, and Loeliger, Hans-Andrea. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47 (2):498–519, 2001.

Lu, Haibing, Vaidya, Jaideep, and Atluri, Vijayalakshmi. Optimal boolean matrix decomposition: Application to role engineering. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pp. 297–306. IEEE, 2008.

Matsushita, Ryosuke and Tanaka, Toshiyuki. Low-rank matrix reconstruction and clustering via approximate message passing. In *Advances in Neural Information Processing Systems*, pp. 917–925, 2013.

Maurus, Samuel and Plant, Claudia. Ternary matrix factorization. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pp. 400–409. IEEE, 2014.

McDonald, Roderick P. *Factor analysis and related methods*. Psychology Press, 2014.

Middleton, Blackford, Shwe, Michael, Heckerman, David, Henrion, Max, Horvitz, Eric, Lehmann, Harold, and Cooper, Gregory. Probabilistic diagnosis using a reformulation of the internist-1/qmr knowledge base. *Medicine*, 30:241–255, 1991.

Miettinen, Pauli and Vreeken, Jilles. Model order selection for boolean matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 51–59. ACM, 2011.

Miettinen, Pauli, Mielikäinen, Taneli, Gionis, Aristides, Das, Gautam, and Mannila, Heikki. The discrete basis problem. In *Knowledge Discovery in Databases: PKDD 2006*, pp. 335–346. Springer, 2006.

Mnih, Andriy and Salakhutdinov, Ruslan. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pp. 1257–1264, 2007.

Parker, Jason T, Schniter, Philip, and Cevher, Volkan. Bilinear generalized approximate message passing. *arXiv preprint arXiv:1310.2632*, 2013.

Pearl, Judea. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI*, pp. 133–136, 1982.

Pearl, Judea. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

Ravanbakhsh, Siamak and Greiner, Russell. Revisiting algebra and complexity of inference in graphical models. *arXiv preprint arXiv:1409.7410*, 2014.

Ravanbakhsh, Siamak, Rabbany, Reihaneh, and Greiner, Russell. Augmentative message passing for traveling salesman problem and graph partitioning. In *Advances in Neural Information Processing Systems*, pp. 289–297, 2014.

Rennie, Jason DM and Srebro, Nathan. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*, pp. 180–186. Kluwer Norwell, MA, 2005.

Sejdinovic, Dino and Johnson, Oliver. Note on noisy group testing: asymptotic bounds and belief propagation reconstruction. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pp. 998–1003. IEEE, 2010.

Šingliar, Tomáš and Hauskrecht, Miloš. Noisy-or component analysis and its application to link analysis. *The Journal of Machine Learning Research*, 7:2189–2213, 2006.

Stockmeyer, Larry J. *The set basis problem is NP-complete*. IBM Thomas J. Watson Research Division, 1975.

Udell, Madeleine, Horn, Corinne, Zadeh, Reza, and Boyd, Stephen. Generalized low rank models. *arXiv preprint arXiv:1410.0342*, 2014.

Vaidya, Jaideep, Atluri, Vijayalakshmi, and Guo, Qi. The role mining problem: finding a minimal descriptive set of roles. In *Proceedings of the 12th ACM symposium on Access control models and technologies*, pp. 175–184. ACM, 2007.

Van den Broeck, Guy and Darwiche, Adnan. On the complexity and approximation of binary evidence in lifted inference. In *Advances in Neural Information Processing Systems*, pp. 2868–2876, 2013.

Weiss, Yair and Freeman, William T. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *Information Theory, IEEE Transactions on*, 47(2):736–744, 2001.

Weiss, Yair, Yanover, Chen, and Meltzer, Talya. Map estimation, linear programming and belief propagation with convex free energies. *arXiv preprint arXiv:1206.5286*, 2012.

Wood, Frank, Griffiths, Thomas, and Ghahramani, Zoubin. A non-parametric bayesian method for inferring hidden causes. *arXiv preprint arXiv:1206.6865*, 2012.

Yedidia, Jonathan S, Freeman, William T, Weiss, Yair, et al. Generalized belief propagation. In *NIPS*, volume 13, pp. 689–695, 2000.

Zhang, Zhong-Yuan, Li, Tao, Ding, Chris, Ren, Xian-Wen, and Zhang, Xiang-Sun. Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery*, 20(1):28–52, 2010.

Zhang, Zhongyuan, Ding, Chris, Li, Tao, and Zhang, Xiangsun. Binary matrix factorization with applications. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pp. 391–400. IEEE, 2007.

Zitnik, Marinka and Zupan, Blaz. Nimfa: A python library for nonnegative matrix factorization. *Journal of Machine Learning Research*, 13:849–853, 2012.