Variational Inference on the Boolean Hypercube with the Quantum Entropy

Eliot Beyler
Inria, Ecole Normale Supérieure,
PSL Research University

Francis Bach
Inria, Ecole Normale Supérieure,
PSL Research University

Abstract

In this paper, we derive variational inference upper-bounds on the log-partition function of pairwise Markov random fields on the Boolean hypercube, based on quantum relaxations of the Kullback-Leibler divergence. We then propose an efficient algorithm to compute these bounds based on primal-dual optimization. An improvement of these bounds through the use of "hierarchies," similar to sum-of-squares (SoS) hierarchies is proposed, and we present a greedy algorithm to select among these relaxations. We carry extensive numerical experiments and compare with state-of-the-art methods for this inference problem.

1 INTRODUCTION

Graphical models (see, e.g., Wainwright and Jordan, 2007) are used to define structured relationships between random variables, making them useful in a variety of computational tasks such as bioinformatics (Durbin et al., 1998; Pevzner, 2000), combinatorial optimization (Goemans and Williamson, 1995), constraint satisfaction (Maneva, Mossel, and Wainwright, 2007), statistical physics (Landau and Lifshitz, 2013), language and speech processing (Blei, Ng, and Jordan, 2003), image processing (Blake, Kohli, and Rother, 2011), or any other problems formulated as a probabilistic model.

1.1 Log-partition function

Undirected graphical models (a.k.a. Markov random fields) define probability distributions up to an un-

known normalization constant, called the log-partition function, which is notoriously hard to compute and a key quantity in inference tasks.

More precisely, we are given a probability p on a finite set $\mathcal{X}$ (which we will take to be $\{-1, 1\}^d$ in most of the paper) known to be proportional to $e^{f(x)}q(x)$ where $f : \mathcal{X} \to \mathbb{R}$ is a function and q a base measure (often chosen to be the uniform measure). We want to find the normalizing constant $\Phi(f)$ such that:

$$p(x) = e^{f(x) - \Phi(f)}q(x).$$

Normalization of p gives the following formula:

$$\Phi(f) = \log \sum_{x \in \mathcal{X}} e^{f(x)}q(x). \tag{1}$$

Computing directly the sum (1) is intractable in general, as the number of terms $|\mathcal{X}|$ will grow exponentially with the number of variables d in our model. Our goal is to derive tractable approximations of this value, i.e., approximations that can be computed with polynomial complexity with respect to d. Moreover, we want to preserve properties of the log-partition function, such as its convexity in f.

1.2 MRF on the Boolean hypercube

The focus of our study is a special class of problems, namely pairwise Markov random fields (MRF) on the Boolean hypercube $\mathcal{X} = \{-1, 1\}^d$, also known as the Ising model, meaning that f is of the form:

$$f(x) = \sum_{s \in V} \theta_s x_s + \sum_{(i,j) \in E} \theta_{ij} x_i x_j,$$

where $G = (V, E)$ is an undirected graph, with vertices $V \subset \{1, \ldots, d\}$. The structure of the graph can be used to characterize independence properties of the probability distribution (Wainwright and Jordan, 2007). When the structure does not matter, we can simply assume the graph to be complete by setting $\theta_i = 0$ if $i \notin V$ and $\theta_{ij} = 0$ if $(i,j) \notin E$.

When the underlying graph has a special tree structure, the computation of the log-partition function can be carried exactly with a message-passing algorithm, known as the sum-product or belief propagation algorithm (see for example Kschischang, Frey, and Loeliger (2001)). However, this problem becomes intractable in the general case. As a consequence, research is focused on obtaining tractable approximation or bounds on the log-partition function.

1.3 Kullback-Leibler divergence

A powerful approach to finding and studying such approximation is variational inference (see, e.g., Wainwright and Jordan, 2007, and references therein). It relies on the Kullback-Leibler (KL) divergence.

For two probability measures p and q over a finite space $\mathcal{X}$, we define the KL divergence between p and q as:

$$D(p\|q) = \sum_{x\in\mathcal{X}} \log\left(\frac{p(x)}{q(x)}\right) p(x).$$

The KL divergence is non-negative, equal to zero if and only if $p = q$ and jointly convex in p and q (Cover and Thomas, 2005).

1.4 Variational inference

The idea behind variational inference is to use convex duality (through Fenchel conjugates) to express the log-partition function as an optimization problem:

$$\log\sum_{x\in\mathcal{X}} e^{f(x)}q(x) = \sup_{p\in\mathcal{P}(\mathcal{X})} \sum_{x\in\mathcal{X}} f(x)p(x) - D(p\|q),$$

where $\mathcal{P}(\mathcal{X})$ is the space of probability on $\mathcal{X}$. This equality is sometimes referred as the "Donsker-Varadhan" identity. One can then construct a tractable upper-bound (or lower-bound), by finding a tractable lower-bound (resp. upper-bound) on the KL divergence, changing the optimization variable and have outer (resp. inner) approximation of the optimization domain. Moreover, if the bound on the KL and the approximation of the optimization domain are convex, the resulting quantity is expressed as a convex optimization problem, thus can be computed efficiently.

In this framework, some methods have been developed to give upper-bounds on the log-partition function from convex lower bound on the KL[1], an important one being the tree-reweighted message passing (TRW)

algorithm (Wainwright, Jaakkola, and Willsky, 2005), another the log-determinant relaxation (Jordan and Wainwright, 2003). The TRW algorithm uses a bound on the entropy based on a combination of entropy for MRF on trees, for which we have a simple closed form formula. For the log-determinant relaxation, the entropy is bounded by the entropy of a Gaussian vector with similar covariance matrix.

1.5 Challenges

Recently, new links between sum-of-squares (SoS) optimization (Lasserre, 2010) and quantum divergences between positive semi-definite matrices used as lower bounds on entropy have been made by Bach (2023, 2024). Bach (2023, 2024) also proposes to use these divergences in variational inference, but experiments are limited to the torus and lack actual comparisons with existing alternatives. Therefore, our work attempts to further explore the use of quantum divergences in machine learning by delving deeper into the particular case of the Boolean hypercube to try the answer the following questions: can we develop practical methods of variational inference with these quantities? Can these methods be competitive with existing algorithms?

We answer these questions positively by implementing efficient optimization algorithms for variational inference methods with the quantum entropy, which can be improved using hierarchies of relaxations similar to the ones used in SoS optimization and greedy selection between theses relaxations.

1.6 Contributions

In this paper, we make the following contributions:

- We specialize the framework of Bach (2023, 2024), reviewed in section 3, to the case of a pairwise MRF on the Boolean hypercube, and propose an efficient algorithm to compute an upper-bound on the log-partition function in section 4, based on a primal-dual optimization algorithm.

- We propose in section 5 an improvement of bound with "hierarchies," similar to SoS hierarchies, and present a greedy algorithm to select among these relaxations.

- We carry in section 7 extensive numerical experiments and compare with the TRW and logdet relaxations.[2]

---

[1]Note that other commonly used methods (see, e.g., Murphy, 2022, 2023, and references therein) such as mean-field methods, have other properties (e.g., give lower bounds). These methods are not studied here.

[2]Source code to reproduce the experiments is available at `https://github.com/ebeyler/variational_inference_quatum_entropy`

2   NOTATIONS

We introduce the following notations:

- $\mathcal{X} = \{-1, 1\}^d$ the Boolean hypercube. We denote $\mathcal{P}(\mathcal{X})$ the set of probability distributions on $\mathcal{X}$. For a function f on $\mathcal{X}$ and a probability $p \in \mathcal{P}(\mathcal{X})$, we write $\mathbb{E}_p[f(x)] = \sum_{x \in \mathcal{X}} f(x)p(x)$ the expectation of $f(x)$ under p.

- $\mathbb{R}$ the space of real numbers. For a positive integer n, $\mathbb{R}^n$ the space of vector of dimension n, $\mathcal{M}_n(\mathbb{R})$ the space of square real matrices of dimension $n \times n$, $\mathbb{S}_n \subset \mathcal{M}_n(\mathbb{R})$ the space the symmetric matrices, $\mathbb{S}_n^+ \subset \mathbb{S}_n$ the positive semi-definite (PSD) matrices, and $\mathbb{S}_n^{++} \subset \mathbb{S}_n$ the positive definite matrices. For $A \in \mathbb{S}_n$, we write $A \succcurlyeq 0$ if $A \in \mathbb{S}_n^+$ and $A \preccurlyeq B$ if $B - A \succcurlyeq 0$.

- For a vector $y \in \mathbb{R}^n$ (or a matrix $A \in \mathcal{M}_n(\mathbb{R})$), we write y^T (resp. A^T) its transpose.

- For $A \in \mathcal{M}_n(\mathbb{R})$, we write $\operatorname{tr} A = \sum_i A_{ii}$, $\operatorname{diag}(A) = (A_{ii})_{i=1\ldots n} \in \mathbb{R}^n$ the diagonal of matrix A, and for $\lambda \in \mathbb{R}^n$, $\operatorname{diag}(\lambda) \in \mathcal{M}_n(\mathbb{R})$ the diagonal matrix with diagonal λ.

- For $\lambda \in \mathbb{R}^n$, $\max(\lambda) = \lambda_i$ such that $\forall j \in \{1, ..., n\}, \lambda_i \geq \lambda_j$.

- For $h : \mathbb{R} \to \mathbb{R}$, and $A \in \mathbb{S}_n$, we write $h(A) = U^T \operatorname{diag}(h(\lambda_1), \ldots, h(\lambda_n))U$ where $A = U^T \operatorname{diag}(\lambda_1, \ldots, \lambda_n)U$ is the spectral decomposition of A. The value $h(A)$ does not depend on the specific choice of orthogonal diagonalization basis and leads to a *spectral function* $h : \mathbb{S}_n \to \mathbb{S}_n$.

3   BOUNDING THE KL DIVERGENCE

The function f defined by $f(x) = \sum_i \theta_i x_i + \sum_{i,j} \theta_{ij} x_i x_j$, can be rewritten as a quadratic form $f(x) = \varphi(x)^T F \varphi(x)$ with $\varphi(x) = (1, x_1, \ldots, x_d) \in \mathbb{R}^{d+1}$, and F a $(d+1) \times (d+1)$ matrix defined by $F_{1,i+1} = \theta_i$ and $F_{i+1,j+1} = F_{j+1,i+1} = \theta_{ij}/2$. Then we can write $f(x) = \operatorname{tr} \varphi(x)\varphi(x)^T F$, leading to

$$\sum_{x \in \mathcal{X}} f(x)p(x) = \operatorname{tr} \Sigma_p F$$

with $\Sigma_p = \sum_{x \in \mathcal{X}} \varphi(x)\varphi(x)^T p(x)$ the *moment matrix*. With this in mind, we present in this section KL approximations based on moment matrices.

3.1   Moment matrix

For $x \in \mathcal{X}$ and $I \subset \{0, 1\}^d$, we define a *feature vector* indexed by I as

$$\varphi_I(x) = (x^\alpha)_{\alpha \in I},$$

where $x^\alpha = \prod_i x_i^{\alpha_i}$. We will often take $I_0 = \{\alpha, \sum_i \alpha_i \leq 1\} = \{0, e_1, \ldots, e_d\}$, where $e_i = (\mathbb{1}_{j=i})_{j \in \{1, \ldots, d\}}$, for which we denote $\varphi_0(x) = \varphi_{I_0}(x) = (1, x) = (1, x_1, \ldots, x_d)$. The x^α are called *monomials* or *features*. We denote n the cardinal of the set I. We have $\|\varphi(x)\|^2 = \sum_{i \in I} (x^{\alpha_i})^2 = n$. We index the entries of $\varphi(x)$ either by the α's, $\varphi(x)_\alpha = x^\alpha$, or by $\{1, \ldots, n\}$.

For a positive measure p on $\mathcal{X}$, we define the *moment matrix* Σ_p by:

$$\Sigma_p = \sum_{x \in \mathcal{X}} \varphi(x)\varphi(x)^T p(x) = \mathbb{E}_p[\varphi(x)\varphi(x)^T].$$

As $\varphi(x)\varphi(x)^T \in \mathbb{S}_n^+$ and p is a non-negative measure, we have $\Sigma_p \in \mathbb{S}_n^+$. Moreover, we have $\operatorname{tr} \Sigma = \sum_{x \in \mathcal{X}} \|\varphi(x)\|^2 p(x) = n$.

As for the vector $\varphi(x)$, the entries of the matrix Σ can be indexed either by $i, j \in \{1, \ldots, n\}$, or by $\alpha, \beta \in I$. Moreover, we have:

$$\begin{aligned}(\Sigma_p)_{\alpha,\beta} &= \mathbb{E}_p[\varphi(x)_\alpha \varphi(x)_\beta] \\ &= \mathbb{E}_p[\prod_i x_i^{\alpha_i + \beta_i}] = \mathbb{E}_p[x^{\alpha \triangle \beta}],\end{aligned} \quad (2)$$

where $\triangle$ is the XOR or symmetric difference, $\alpha \triangle \beta = (\mathbb{1}_{\alpha_i \neq \beta_i})_{i=1,\ldots,d}$.

We denote $\mathcal{V}$ the linear span of the $(\varphi(x)\varphi(x)^T)_{x \in \mathcal{X}}$, with the following description:

$$\begin{aligned}\mathcal{V} = \{M : |I| \times |I| \text{ matrix such that} \\ M_{\alpha,\beta} = M_{\alpha',\beta'} \text{ for all } \alpha \triangle \beta = \alpha' \triangle \beta'\}, \quad (3)\end{aligned}$$

where the inclusion $\subset$ comes directly from (2) and the reserve inclusion $\supset$ comes from the fact that the family of functions $(x \mapsto x^\alpha)_{\alpha \in \{0,1\}^d}$ forms a basis of $\mathbb{R}^{\mathcal{X}}$, the space of functions from $\mathcal{X}$ to $\mathbb{R}$. For q the uniform measure on $\mathcal{X}$, the random variable $x_1, \ldots, x_d$ are independent, hence $\mathbb{E}_q[x^\alpha] = \prod_i \mathbb{E}_p[x_i^{\alpha_i}]$. $\mathbb{E}_q[x_i^{\alpha_i}]$ is equal to 0 if $\alpha_i = 1$ and 1 otherwise. With equation (2), it leads to $\Sigma_q = I_n$.

3.2   Quantum lower bound on the KL

We want to define a relaxation of the KL based on moment matrices. Hence it will be defined on $\mathbb{S}_n^+$.

For $A, B \in \mathbb{S}_n^+$, the quantum relaxation of the KL (Bach, 2024; Matsumoto, 2018) is :

$$D_V^{\mathrm{QT}}(A, B) = \operatorname{tr} V B^{1/2} h(B^{-1/2} A B^{-1/2}) B^{1/2},$$

where $h(t) = t \log t - t + 1$ defines a spectral function on $\mathbb{S}_n^+$ and $V \in \mathbb{S}_n^+$ is such that $\varphi(x)^T V \varphi(x) \leq 1$.

This quantity, among a series of relaxation of the KL,

is thoroughly studied by Bach (2024)[3], from which we learn that D_V^{QT} is jointly convex in A and B, $D_V^{\mathrm{QT}}(A,B) \geq 0$. Moreover, it is a lower bound on the KL: for $p,q \in \mathcal{P}(\mathcal{X})$, $D_V^{\mathrm{QT}}(\Sigma_p \| \Sigma_q) \leq D(p \| q)$.

On the boolean hypercube, for any $x \in \mathcal{X}$, $\|\varphi(x)\|^2 = \sum_{\alpha \in I}(x^\alpha)^2 = n$, therefore $V = I/n$ verifies $V \succcurlyeq 0$ and $\forall x \in \mathcal{X}, \varphi(x)^T V \varphi(x) \leq 1$.[4] For $V = I/n$, we write:

$$D^{\mathrm{QT}}(A,B) = D_{I/n}^{\mathrm{QT}}(A,B)$$
$$= \frac{1}{n}\left(\operatorname{tr} A \log B^{-1}A - \operatorname{tr} A + \operatorname{tr} B\right). \quad (4)$$

Taking $A = \Sigma_p$ for some probability p and $B = \Sigma_q = I$ for q the uniform distribution, we define:

$$H(\Sigma_p) = D^{\mathrm{QT}}(\Sigma_p, \Sigma_q) = \frac{1}{n}\operatorname{tr}\Sigma_p \log \Sigma_p,$$

which corresponds to the von Neumann entropy (see for example Araki and Lieb (1970)).

4 TRACTABLE UPPER BOUND ON THE LOG-PARTITION FUNCTION

4.1 Upper bound on the log-partition function

We start from the Donsker-Varadhan identity. To be more general (and make a link with optimization problems later), a temperature parameter $\varepsilon > 0$ is added by taking $f \leftarrow f/\varepsilon$ and multiplying the equality by ε:

$$\Phi_\varepsilon(f) = \varepsilon \log \sum_{x \in \mathcal{X}} e^{f(x)/\varepsilon} q(x)$$
$$= \sup_{p \in \mathcal{P}(\mathcal{X})} \sum_{x \in \mathcal{X}} f(x)p(x) - \varepsilon D(p\|q). \quad (5)$$

We then get an upper bound on $\Phi_\varepsilon(f)$ by replacing $D(p\|q)$ by its lower-bound $D^{\mathrm{QT}}(\Sigma_p\|\Sigma_q)$:

$$\sup_{p \in P(\mathcal{X})} \sum_{x \in \mathcal{X}} f(x)p(x) - \varepsilon D^{\mathrm{QT}}(\Sigma_p\|\Sigma_q). \quad (6)$$

To define our first quantum relaxation of the log-partition function (see section 5 for more refined relaxations), we take the feature vector $\varphi(x) = \varphi_0(x) = (1,x)$, for which we have $f(x) = \varphi(x)^T F \varphi(x)$ with $F \in \mathbb{S}_n$ defined by $F_{1,i+1} = \theta_i$ and $F_{i+1,j+1} = F_{j+1,i+1} =$

$\theta_{ij}/2$. Then, for $p \in \mathcal{P}(\mathcal{X})$, $\sum_{x \in \mathcal{X}} f(x)p(x) = \operatorname{tr}\Sigma_p F$. Equation (6) then becomes:

$$\sup_{p \in P(\mathcal{X}), \Sigma = \Sigma_p} \operatorname{tr} F\Sigma - \varepsilon D^{\mathrm{QT}}(\Sigma\|\Sigma_q). \quad (7)$$

The set

$$\mathcal{K} = \{\Sigma = \Sigma_p, p \in P(\mathcal{X})\}$$

is too complex to lead to a tractable relaxation, therefore we need to find a proper outer approximation. For $\Sigma \in \mathcal{K}$, there exists a probability p such that $\Sigma = \Sigma_p = \sum_{x \in \mathcal{X}} \varphi(x)\varphi(x)^T p(x)$. In particular, we have $\Sigma \succcurlyeq 0$, $\Sigma \in \mathcal{V}$ and $\operatorname{tr}\Sigma = n$. Hence the choice of the outer-approximation

$$\mathcal{K}' = \{\Sigma \succcurlyeq 0, \operatorname{tr}(\Sigma) = n, \Sigma \in \mathcal{V}\}.$$

From (7), we can then define our quantum relaxation of the log-partition function:

$$a_\varepsilon(F) = \sup_{\Sigma \in \mathcal{K}'} \operatorname{tr}\Sigma F - \varepsilon D^{\mathrm{QT}}(\Sigma, \Sigma_q),$$

which verifies $a_\varepsilon(F) \geq \Phi_\varepsilon(f)$.

Using the expression of $D^{\mathrm{QT}}(\Sigma, \Sigma_q)$ given by (4), we get:

$$a_\varepsilon(F) = \sup_{\Sigma \in \mathcal{K}'} \operatorname{tr}\Sigma F - \frac{\varepsilon}{n}\left(\operatorname{tr}\Sigma \log \Sigma - \operatorname{tr}\Sigma + n\right),$$

and because of the constraint $\operatorname{tr}(\Sigma) = n$, it is also equivalent to:

$$a_\varepsilon(F) = \sup_{\Sigma \in \mathcal{K}'} \operatorname{tr}\Sigma F - \frac{\varepsilon}{n}\operatorname{tr}\Sigma \log \Sigma.$$

Assuming that the feature vector is of size polynomial in d, this relaxation is tractable as it amounts to manipulating matrices of size polynomial in d, and solving a convex optimization problem, which can be done in polynomial time.

4.2 Computing the bound

We used Chambolle-Pock algorithm (Chambolle and Pock, 2011) to solve the problem defining the quantum relaxation. This algorithm solves a problem of the form:

$$\min_{x \in X} F(x) + G(x), \quad (8)$$

where X is a finite-dimensional real vector space with inner product $\langle \cdot, \cdot \rangle$, $G : X \to [0, +\infty]$ and $F : X \to [0, +\infty]$ are proper, convex, lower-semicontinuous functions.[5] Note that this primal problem is equivalent to its dual:

$$\max_{y \in X} -(G^*(-y) + F^*(y)). \quad (9)$$

---

[3]Note that more information theory properties of a different but very similar quantity were also studied in Bach (2023). Moreover, statistical and geometric properties of the quantum KL have been studied recently by Chazal, Korba, and Bach (2024).

[4]Instead of taking a fixed matrix V, we proposed a method to optimize the bound over a set of such matrices in the supplementary materials C.

[5]For more details on the convexity results used in this section, see for example Bauschke and Combettes (2011).

where $G^* : X \to [0, +\infty]$, $F^* : X \to [0, +\infty]$ are the Fenchel conjugates of F and G.

Here we take $X = \mathbb{S}_n$ and:

$$G : x = \Sigma \mapsto \frac{\varepsilon}{n} \left(\operatorname{tr} \Sigma \log \Sigma - \operatorname{tr} \Sigma + n \right) - \operatorname{tr} \Sigma F$$

$$F : x = \Sigma \mapsto \iota_{\mathcal{V} \cap H}(\Sigma),$$

with $H = \{\Sigma, \operatorname{tr} \Sigma = n\}$ an affine hyperplane, $\mathcal{V}$ is defined by (3) and $\iota_{\mathcal{V} \cap H} : \mathbb{S}_n \to \{0, +\infty\}$ the indicator function of $\mathcal{V} \cap H$.

The proximal operators are defined, for $\tau > 0$ and $\sigma > 0$, by:

$$(I + \tau \partial F)^{-1}(\Sigma_0)$$
$$= \operatorname{argmin}_\Sigma \left(F(\Sigma) + \frac{\|\Sigma - \Sigma_0\|^2}{2\tau} \right)$$
$$= \operatorname{proj}_{\mathcal{V} \cap H}(\Sigma_0)$$
$$= \operatorname{proj}_{\mathcal{V}}(\Sigma_0) + \left(1 - \frac{\operatorname{tr}(\operatorname{proj}_{\mathcal{V}}(\Sigma_0))}{n} \right) I,$$

with $\operatorname{proj}_{\mathcal{V} \cap H}$ and $\operatorname{proj}_{\mathcal{V}}$ the orthogonal projections on the affine spaces $\mathcal{V} \cap H$ and $\mathcal{V}$. We get $(I + \sigma \partial F^*)^{-1}$ through Moreau's identity:

$$(I + \sigma \partial F^*)^{-1}(\Sigma_0) = \Sigma_0 - \sigma \left(I + (1/\sigma) \partial F \right)^{-1} (\Sigma/\sigma)$$
$$= \Sigma_0 - \operatorname{proj}_{\mathcal{V}}(\Sigma_0) - \left(\sigma - \frac{\operatorname{tr}(\operatorname{proj}_{\mathcal{V}}(\Sigma_0))}{n} \right) I.$$

And for G:

$$(I + \tau \partial G)^{-1} = \operatorname{argmin}_\Sigma \left(G(\Sigma) + \frac{\|\Sigma - \Sigma_0\|^2}{2\tau} \right),$$

the first order condition can be rewritten:

$$\log \Sigma + \frac{n}{\varepsilon \tau} \Sigma = \frac{n}{\varepsilon} \left(F + \frac{\Sigma_0}{\tau} \right).$$

Defining the real function $g_\lambda : t \in \mathbb{R}^{++} \mapsto \log t + \frac{t}{\lambda} \in \mathbb{R}$, we can express this first order condition with a spectral function $g_{\frac{\tau \varepsilon}{n}}(\Sigma) = \frac{n}{\varepsilon} \left(F + \frac{\Sigma_0}{\tau} \right)$, hence the solution:

$$(I + \tau \partial G)^{-1} = g_{\frac{\tau \varepsilon}{n}}^{-1} \left(\frac{n}{\varepsilon} \left(F + \frac{\Sigma_0}{\tau} \right) \right).$$

Moreover, $g_\lambda^{-1}(t) = \lambda \Omega(t - \log \lambda)$ where Ω is the Wright Omega function, for which we have fast and reliable implementation (Lawrence, Corless, and Jeffrey, 2012).

We also compute the following expressions for the Fenchel conjugates of F and G:

$$F^*(Y) = \operatorname{tr} Y + \iota_{\mathcal{V}^* + \mathbb{R} \cdot I}(Y),$$
$$G^*(Y) = \frac{\varepsilon}{n} \operatorname{tr} \left(\exp \left(\frac{n}{\varepsilon}(Y + F) \right) \right).$$

The algorithm requires parameters $\tau, \sigma > 0$, $\theta \in [0, 1]$ and initial point $x_0 \in X$, the updates are defined in (Alg. 1). We take parameter values $\theta = 1$, $\tau \sigma < 1$, for which Theorem 1 of Chambolle and Pock (2011) guaranties convergence of the iterates. We experimentally found that it was reasonable to take fixed values of τ and σ, which are $\tau = 3$ and $\sigma = 0.3$.

Algorithm 1 Chambolle-Pock algorithm

Initialization: Choose $\tau, \sigma > 0$, $\theta \in [0, 1]$ $x_0, y_0 \in X$ and set $\bar{x}_0 = x_0$

Iterations ($n \geq 0$): Update $x_n, y_n, \bar{x}_n$ as follows:

$$\begin{cases} y_{n+1} & = (I + \sigma \partial F^*)^{-1}(y_n + \sigma \bar{x}_n) \\ x_{n+1} & = (I + \tau \partial G)^{-1}(x_n - \tau y_{n+1}) \\ \bar{x}_{n+1} & = x_{n+1} + \theta(x_{n+1} - x_n) \end{cases}$$

Using the primal problem (8) and the dual (9), one can compute a duality gap and define a stopping criterion guarantying some tolerance tol > 0. While the dual points given by the iterates are always feasible, it is not the case for the primal points. We construct a feasible point by first projecting an iterate Σ onto $\mathcal{V} \cap H$, then adding the identity matrix to ensure that eigenvalues are non-negative:

$$\Sigma_{\text{feasible}} = (1 - u) \cdot \operatorname{proj}_{\mathcal{V} \cap H}(\Sigma) + u \cdot I,$$

with $u = \begin{cases} -\frac{\lambda}{1-\lambda} & \text{if } \lambda = \lambda_{\min}(\operatorname{proj}_{\mathcal{V} \cap H}(\Sigma)) < 0 \\ 0 & \text{otherwise.} \end{cases}$

4.3 Computational complexity

Our algorithm involves manipulating matrices of size $n \times n$. We do not need to keep any history of the iterates, hence a space complexity of $O(n^2)$. Assuming that the size n of the feature vector is $O(d)$ ($n = d + 1$ for $\varphi(x) = (1, x)$), we get a space complexity $O(d^2)$.

The projection onto the space $\mathcal{V}$ (3) can be computed in $O(n^2)$, because we only need to average entries that are equal for matrices in $\mathcal{V}$. All other steps can be cast by summing and multiplying matrices by a scalar $O(n^2)$, computing the eigenvalue decomposition of a matrix $O(n^3)$, applying a scalar function to a vector of eigenvalues $O(n)$, and multiplying matrices $O(n^3)$. Hence each step of Chambolle-Pock algorithm as a time complexity in $O(n^3)$.

Chambolle and Pock (2011) give a theoretical bound on the number of iterations to reach a tolerance of δ in $O(\frac{1}{\delta})$, with a constant that can depend on the objective function, hence in n and model parameters θ.

Empirically (see section E.6 in supplementary material), for feature vector $\varphi(x) = (1, x)$, we found out that we rather had linear convergence in δ, i.e., in

$O(\log(\frac{1}{\delta}))$, and that the constant was linear in d for reasonable choices of models parameters. Hence a total time complexity in $O(\log(\frac{1}{\delta})d^4)$.

For a more general feature vector, we do not observe this linear convergence in δ, but rather the theoretical rate in $O(\frac{1}{\delta})$, with a constant in $O(n^2)$, leading to a time complexity in $O(\frac{1}{\delta}n^5)$.

5 IMPROVING THE BOUND

5.1 Tightness

The "smallest" feature vector we can take in order to be sure to be able to write $\sum_{x\in\mathcal{X}} f(x)p(x) = \operatorname{tr}\Sigma_p F$ is $\varphi_0(x) = (1,x)$. Could adding more features make the bound tighter ?

For the feature vector $\varphi(x) = (x^\alpha)_{\alpha\in\{0,1\}^d}$, i.e., taking all possible features, the bound $D^{\mathrm{QT}}(\Sigma_p\|\Sigma_q) \leq D(p\|q)$ is tight. Indeed, the vectors $\varphi(x)/2^d$ form an orthogonal basis of $\mathbb{R}^{2^d}$ as for $x, x' \in \mathcal{X}$:

$$\varphi(x)^T \varphi(x') = \sum_{\alpha\in\{0,1\}^d} x^\alpha x'^\alpha = \prod_i (1 + x_i x_i').$$

Therefore, the eigenvectors and eigenvalues of Σ_p and Σ_q are exactly the $\varphi(x)/2^d$ and respectively $2^d p(x)$ and $2^d q(x)$. It follows that:

$$D^{\mathrm{QT}}(\Sigma_p, \Sigma_q) = \frac{1}{2^d} \operatorname{tr}\left(\Sigma_p \log \Sigma_q^{-1}\Sigma_p - \Sigma_p + \Sigma_q\right)$$
$$= \frac{1}{2^d}\left(\sum_{x\in\mathcal{X}} 2^d p(x) \log \frac{2^d p(x)}{2^d q(x)} - 2^d q(x) + 2^d q(x)\right)$$
$$= \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = D(p\|q).$$

However, as the number of features $n = 2^d$ is exponential in d, it is not tractable.

5.2 Hierarchies

What about just adding some extra features? Adding extra features defines hierarchies of relaxations, classical in sum-of-squares optimization (Lasserre, 2010).

Suppose that a feature vector φ defines a quantum relaxation of the log-partition function, with some matrix F such that $f(x) = \varphi(x)^T F \varphi(x)$ as above. If this feature vector is embedded in a larger feature vector $\widetilde{\varphi} = (\varphi, \varphi_+)$, one can write $f(x) = \widetilde{\varphi}(x)^T \widetilde{F}\widetilde{\varphi}(x)$ with

$$\widetilde{F} = \begin{pmatrix} F & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{S}_{\widetilde{n}},$$

leading to a relaxation of the log-partition function, defined as an optimization problem over $\widetilde{\mathcal{K}} = \{\widetilde{\Sigma} \succcurlyeq$

$0, \operatorname{tr}(\widetilde{\Sigma}) = \widetilde{n}, \Sigma \in \widetilde{\mathcal{V}}\}$. A matrix $\widetilde{\Sigma} \in \mathbb{S}_{\widetilde{n}}$ can be written

$$\widetilde{\Sigma} = \begin{pmatrix} \Sigma & \Sigma_+ \\ \Sigma_+^T & \Sigma_{++} \end{pmatrix},$$

with $\Sigma \in \mathbb{S}_n$. Having $\widetilde{\Sigma} \in \widetilde{\mathcal{K}}'$ implies that $\Sigma \in \mathcal{K}'$, meaning that adding extra feature, like in classical SoS, leads to additional constraints on the optimization domain and an improved value of the upper-bound on the log partition function.

5.3 Greedy algorithm

Adding additional features can make the bound tighter, however it makes the problem more computationally complex. We explore the idea of adding additional features one by one with a greedy algorithm similar to orthogonal matching pursuit (Mallat, 2009).

From a feature vector $\varphi_I(x) = (x^\alpha)_{\alpha\in I}$, we compute the set of features at "distance one" from I: $\widetilde{I} = \{\alpha \,\triangle\, e_i, \forall i \in \{1,\dots,d\}, \forall \alpha \in I\}\backslash I$, where $e_i = (\mathbb{1}_{j=i})_{j\in\{1,\dots,d\}}$ and $\triangle$ is the XOR or symmetric difference. We then select the feature $x^{\hat\alpha} \in \widetilde{I}$ such that the value of $\widetilde{a}_\varepsilon(F)$ computed with feature vector $\widetilde{\varphi}(x) = (x^\alpha)_{\alpha\in I\cup\{\hat\alpha\}}$ is the smallest. The procedure is summarized in Alg. 2.

Starting from feature vector $\varphi_0 = (1,x)$, we select k features with this procedure. As $|\widetilde{I}| \leq d|I|$, we compute the value of $a_\varepsilon(F)$ at most $O(k \times d \times (d+k))$ times with feature vectors of size at most $d + k + 1$, therefore this algorithm remains polynomial in d.

In practice, to select the best $\hat\alpha$ we don't need a high precision on the value of the $\widetilde{a}_\varepsilon(F)$'s, so we used a higher tolerance during the selection selection process, and then a low tolerance to compute the final value once the features have been chosen.

Algorithm 2 Greedy selection of features

Initialization: F matrix of parameters , k number of features to select.
Set: $I = I_0 = \{0, e_1, \dots, e_d\}$ and $\varphi(x) = \varphi_I(x) = (1, x_1, \dots, x_d)$.
For $i = 1, \dots, k$**:**
   – Compute $\widetilde{I} = \{\alpha \,\triangle\, e_i, \forall i \in \{1,\dots,d\}, \forall \alpha \in I\}\backslash I$.
   – For $\hat\alpha \in \widetilde{I}$: compute $a(\hat\alpha) = \widetilde{a}_\varepsilon(F)$ for feature vector $\widetilde{\varphi}(x) = (x^\alpha)_{\alpha\in I\cup\{\hat\alpha\}}$.
   – Select $\hat{\alpha_0} = \operatorname{argmin}_{\hat\alpha} a(\hat\alpha)$.
   – Update $I \leftarrow I \cup \{\hat{\alpha_0}\}$.

---

6 LOW TEMPERATURE LIMIT

As $D(p\|q)$ and $D^{\mathrm{QT}}(\Sigma\|\Sigma_q)$ are non-negative, $\Phi_\varepsilon(f)$ and $a_\varepsilon(F)$ are non-decreasing with respect to ε. What are the limit of these quantities when $\varepsilon \to 0$?

From the dual formulation (5), as $D(p\|q)$ is finite for all $p \in \mathcal{P}(\mathcal{X})$ and $\mathcal{P}(\mathcal{X})$ is bounded, it follows that:

$$\Phi_\varepsilon(f) \xrightarrow[\varepsilon \to 0]{} \sup_{p \in \mathcal{P}(\mathcal{X})} \sum_{x \in \mathcal{X}} f(x)p(x) = \max_{x \in \mathcal{X}} f(x).$$

Moreover, $D^{\mathrm{QT}}(\Sigma\|\Sigma_q)$ is finite for all $\Sigma \in \mathcal{K}'$ with $\mathcal{K}'$ bounded, so we also have:

$$a_\varepsilon(F) \xrightarrow[\varepsilon \to 0]{} a_0^1(F) = \sup_{\Sigma \in \mathcal{K}'} \operatorname{tr} \Sigma F.$$

$a_0^1(F)$ corresponds to classical tractable SOS relaxation of the optimization problem $\max_{x \in \mathcal{X}} f(x)$ (Lasserre, 2010). Here, in the special case of the Boolean hypercube, these relaxations are linked to the famous semidefinite relaxations of the MAXCUT problem (Goemans and Williamson, 1995) and nonconvex quadratic optimization (Nesterov, 1998).

When the graph underlying the MRF is a tree, this relaxation is tight as soon as we take the feature vector $\varphi(x) = (\varphi_0(x), \varphi_+(x))$, where $\varphi_+(x) = (x_i x_j)_{ij \in E}$. See supplementary material B for more details.

7 EXPERIMENTS

7.1 Competing algorithms

The state-of-the-art algorithms for the problem of computing upper-bounds on the log-partition function for the Ising model are the log-determinant relaxation from Jordan and Wainwright (2003) and tree-reweighted message passing (TRW) from Wainwright, Jaakkola, and Willsky (2005). We give a brief overview of these methods in the supplementary material A.

We implemented our algorithms, as well as the competing algorithms, in Python, writing the optimization schemes using NumPy for our algorithms and TRW, and using CVXPY (Diamond and Boyd, 2016) for the log-determinant. We chose to prioritize simplicity and reusability of the code over fully optimized code. Therefore, we do not include an extensive numerical comparison of runtimes as it would only be scientifically sound if all methods were implemented using the same framework (e.g., with first-order methods).

7.2 Set-up

Parameters. To evaluate the relaxations, we need to choose the value of the vector of the parameters θ in $f(x) = \sum_i \theta_i x_i + \sum_{ij} \theta_{ij} x_i x_j$.

First, we take $\theta \sim \mathcal{N}(0, I)$, and we call this setting *Gaussian parameters*.

We also take the setting of Jordan and Wainwright (2003), which we call *log-det parameters*. It corresponds to $\theta_i \sim \mathcal{U}(-0.25, +0.25)$ independently, and

distinguishes three cases for the couplings terms: $\theta_{ij} \sim \mathcal{U}(0, 2w)$ for *attractive coupling*, $\theta_{ij} \sim \mathcal{U}(-w, w)$ for *mixed coupling* and $\theta_{ij} \sim \mathcal{U}(-2w, 0)$ for *repulsive coupling*, where $w \geq 0$ is the *coupling strength*.

Graph. We also need to choose the graph $G = (V, E)$ underling the MRF. We take $V = \{1, \ldots, d\}$ and distinguish three cases for the edges E:

- independent variable: $E = \varnothing$, i.e., $\theta_{ij} = 0$ for all i, j.
- tree: $G = (V, E)$ is a tree, i.e., $\theta_{ij} \neq 0$ only on edges of this tree
- complete graph: $E = \{(i, j)$ for all $i, j\}$, i.e., all $\theta_{ij} \neq 0$.

Metrics. To evaluation the error in the upper-bound on the log-partition function compare to the true value, we follow Wainwright, Jaakkola, and Willsky (2005) and define the *normalized error in bound* as

$$error\_bound = (bound - true\_value)/d.$$

As in Jordan and Wainwright (2003) and Wainwright, Jaakkola, and Willsky (2005), the approximated marginals $\hat{p}$ is evaluated with a l_1 error,

$$l1\_error = \tfrac{1}{d} \sum_{i \in \{1, \ldots, d\}} |\hat{p}(x_i = 1) - p(x_i = 1)|.$$

7.3 Hierarchies

We start by comparing our different relaxations for different level of hierarchies. We take Gaussian parameters, and $d = 5$, so that we can take up to all possible features. We plot the value of our relaxations versus the size of the feature vector, starting with $\varphi_0(x) = (1, x)$ of size 6. For the quantum relaxation, we take features α with increasing numbers of non-zero coordinate, while for the quantum relaxation we select features with the greedy procedure presented in section 5.3.

From Fig. 1, we see that the greedy procedure allows to select interesting features quickly, and allows to reduce the gap with less features than adding features in arbitrary order, in particular for independent variable or MRF on a tree, while the gains for the complete graph are less pronounced.

7.4 Comparison with competing algorithms

We compare the relaxations on the log-det parameters for varying coupling strength.[6] We limit ourselves

---

[6]Comparisons for other choices of parameters are done in supplementary materials E.2 and E.3.

to $d = 5$ and coupling strength $w < 0.5$, as we observed convergence issues with TRW algorithm. For the greedy algorithm, we take $k = 3$ extra features.[7]

In Fig. 2, we use the log-det parameters on the complete graph. Our relaxations give better bound on the log-partition function than the log-determinant relaxation for all coupling strengths and interaction types. They are competitive with the TRW relaxation for low coupling strength. For attractive coupling and high coupling strength, TRW is better. However, for mixed couplings, the greedy relaxation stays competitive with TRW even at higher coupling strength, and for repulsive coupling, both relaxations become better than the TRW relaxation at high coupling strength. For the *l1_error*, our relaxations exhibit higher error than both the TRW and log-determinant relaxations.

We also carried out additional experiments with a larger value of d, $d = 16$. As we observed convergence issues with the message passing scheme of the TRW relaxation, leading to imprecise results and causing the optimization of ρ to fail, we decided to use a fixed uniform value for ρ, $\rho_{ij} = 2/d$ (such that $\sum_{ij} \rho_{ij} = d - 1$), and to stop message passing after a maximum of 10000 iterations.[8]

In Fig. 3, we use the log-det parameters on the complete graph. For attractive coupling, our relaxations are not as good as competing algorithms as soon as the coupling strength increases, but stay competitive with the log-determinant relaxation for mixed coupling and repulsive coupling, while the performance of the TRW relaxation deteriorates.

8   CONCLUSION

In this work, we developed from Bach (2023, 2024), and studied relaxations of the KL divergence and the log-partition function on the Boolean hypercube $\mathcal{X} = \{-1, 1\}^d$. The relaxations we propose are competitive with the best competing algorithms for the same problem, and even better in some settings.

They are several avenues to be explored to extend this work. As these methods rely on a feature vector, they can be adapted on other spaces $\mathcal{X}$, paving the way to a universal framework for variational inference in a wide variety of setting beyond the Boolean hypercube. We could also try to improve our algorithms by using recent improvements in SDP and SoS optimization (see, e.g., Huang et al., 2022; Jiang, Natura, and Weinstein, 2023). It would be interesting to get theoretical guaranties on our upper-bounds of the log-partition function, we could try using recent convergence results of SoS optimization for the hypercube (Baldi and Slot, 2024) or take inspiration from works on other variational inference methods (see, e.g., Jain, Koehler, and Risteski, 2019; Risteski, 2016). Moreover, there are strong links to be made with sum-of-squares optimization, allowing to tackle complex ML pipelines involving both inference and optimization. Finally, the algorithms we developed do not take into account the special structure of the graphical model. Future work could be dedicated to investigate how our methods can gain from incorporating more knowledge on the problem structure.

Acknowledgements

---

[7]We take other values of k in the supplementary material E.4, showing even stronger bounds at the expense of more computations.

[8]Note moreover that in the absence of convergence, there is no more guaranty that the output is an upper-bound on the true value.

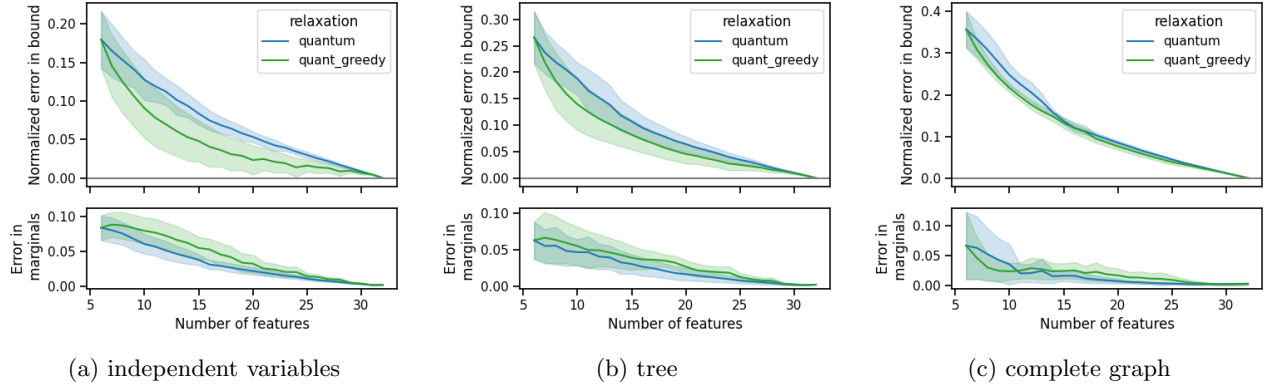(a) independent variables   (b) tree   (c) complete graph

Figure 1: Comparison of the different quantum relaxations of the KL divergence. $d = 5$, Gaussian parameters. x-axis corresponds to the number of features in the feature vector. Mean over 10 draws of parameters, ± 1 standard deviation.
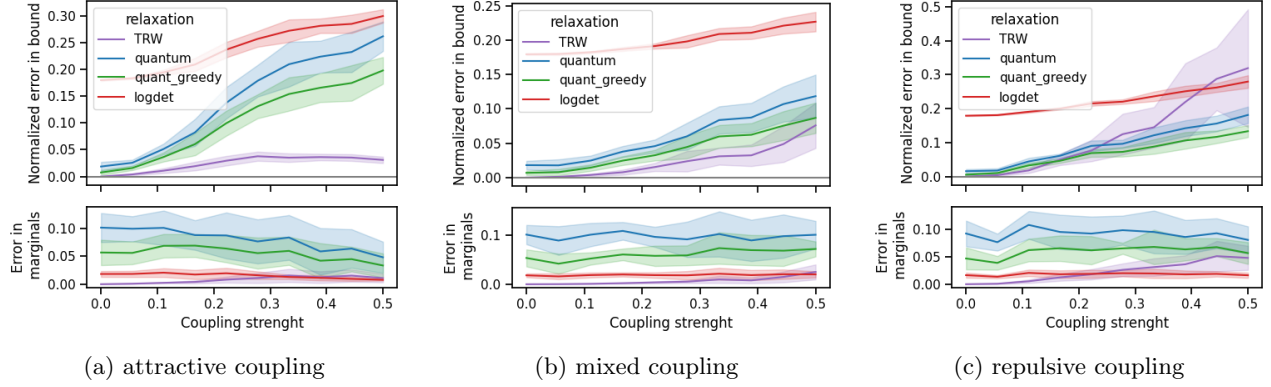

(a) attractive coupling   (b) mixed coupling   (c) repulsive coupling

Figure 2: Comparison with competing algorithm for log-det parameters and varying coupling strength. $d = 5$, complete graph. Mean over 10 draws of parameters, ± 1 standard deviation.

(a) attractive coupling   (b) mixed coupling   (c) repulsive coupling
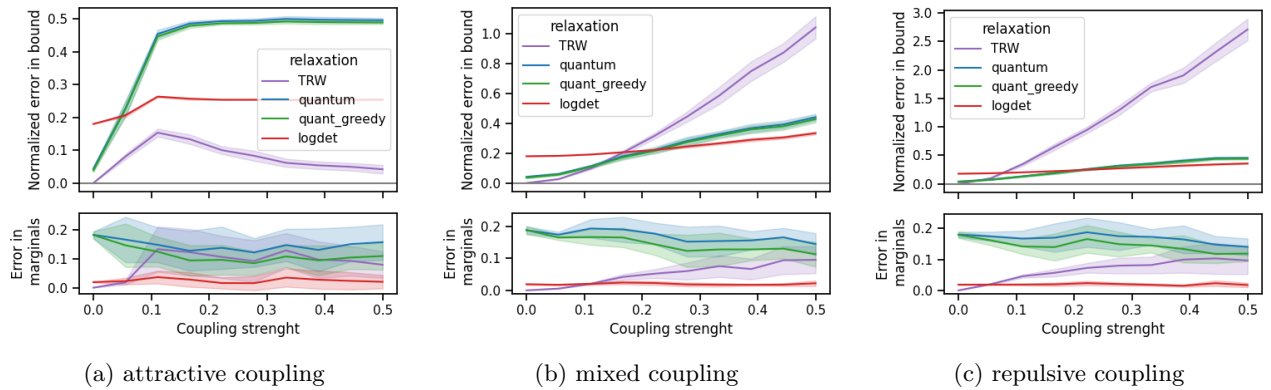
Figure 3: Comparison with competing algorithm for log-det parameters and varying coupling strength. $d = 16$, complete graph. Mean over 10 draws of parameters, ± 1 standard deviation.

References

Araki, Huzihiro and Elliott H. Lieb (June 1970). "Entropy Inequalities". In: *Communications in Mathematical Physics* 18.2, pp. 160–170.

Bach, Francis (Feb. 2023). "Information Theory With Kernel Methods". In: *IEEE Transactions on Information Theory* 69.2, pp. 752–775.

– (Apr. 2024). "Sum-of-Squares Relaxations for Information Theory and Variational Inference". In: *Foundations of Computational Mathematics*.

Baldi, Lorenzo and Lucas Slot (Mar. 2024). "Degree Bounds for Putinar's Positivstellensatz on the Hypercube". In: *SIAM Journal on Applied Algebra and Geometry* 8.1, pp. 1–25.

Bauschke, Heinz H. and Patrick L. Combettes (2011). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. New York, NY: Springer.

Blake, Andrew, Pushmeet Kohli, and Carsten Rother (2011). *Markov Random Fields for Vision and Image Processing*. MIT press.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent Dirichlet Allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Chambolle, Antonin and Thomas Pock (May 2011). "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging". In: *Journal of Mathematical Imaging and Vision* 40.1, pp. 120–145.

Chazal, Clémentine, Anna Korba, and Francis Bach (2024). "Statistical and Geometrical Properties of Regularized Kernel Kullback-Leibler Divergence". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Cover, Thomas and Joy A Thomas (Apr. 2005). *Elements of Information Theory*. Wiley.

Diamond, Steven and Stephen Boyd (2016). "CVXPY: A Python-embedded Modeling Language for Convex Optimization". In: *Journal of Machine Learning Research* 17.83, pp. 1–5.

Durbin, Richard et al. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Frank, Marguerite and Philip Wolfe (1956). "An Algorithm for Quadratic Programming". In: *Naval Research Logistics Quarterly* 3.1-2, pp. 95–110.

Goemans, Michel X. and David P. Williamson (Nov. 1995). "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming". In: *Journal of the ACM* 42.6, pp. 1115–1145.

Huang, Baihe et al. (Oct. 2022). "Solving SDP Faster: A Robust IPM Framework and Efficient Implementation". In: *Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 233–244.

Jain, Vishesh, Frederic Koehler, and Andrej Risteski (June 2019). "Mean-Field Approximation, Convex Hierarchies, and the Optimality of Correlation Rounding: A Unified Perspective". In: *Symposium on Theory of Computing*. STOC 2019, pp. 1226–1236.

Jiang, Shunhua, Bento Natura, and Omri Weinstein (Sept. 2023). "A Faster Interior-Point Method for Sum-of-Squares Optimization". In: *Algorithmica* 85.9, pp. 2843–2884.

Jordan, Michael and Martin J Wainwright (2003). "Semidefinite Relaxations for Approximate Inference on Graphs with Cycles". In: *Advances in Neural Information Processing Systems*. Vol. 16.

Kato, Tosio (1966). *Perturbation Theory for Linear Operators*. Berlin, Heidelberg: Springer.

Kelley Jr., J. E. (Dec. 1960). "The Cutting-Plane Method for Solving Convex Programs". In: *Journal of the Society for Industrial and Applied Mathematics* 8.4, pp. 703–712.

Kschischang, F.R., B.J. Frey, and H.-A. Loeliger (Feb. 2001). "Factor Graphs and the Sum-Product Algorithm". In: *IEEE Transactions on Information Theory* 47.2, pp. 498–519.

Landau, Lev Davidovich and Evgenii Mikhailovich Lifshitz (2013). *Statistical Physics: Volume 5*. Vol. 5. Elsevier.

Lasserre, Jean-Bernard (2010). *Moments, Positive Polynomials and Their Applications*. Imperial College Press Optimization Series Vol. 1. London: Imperial College Press.

Lawrence, Piers W., Robert M. Corless, and David J. Jeffrey (Apr. 2012). "Algorithm 917: Complex Double-Precision Evaluation of the Wright ω Function". In: *ACM Trans. Math. Softw.* 38.3, 20:1–20:17.

Mallat, Stéphane (2009). *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd ed. Amsterdam Boston: Elsevier/Academic Press.

Maneva, Elitza, Elchanan Mossel, and Martin J Wainwright (2007). "A New Look at Survey Propagation and Its Generalizations". In: *Journal of the ACM (JACM)* 54.4, 17–es.

Matsumoto, Keiji (2018). "A New Quantum Version of F-Divergence". In: *Reality and Measurement in Algebraic Quantum Theory*. Ed. by Masanao Ozawa et al. Vol. 261. Singapore: Springer Singapore, pp. 229–273.

Murphy, Kevin P. (2022). *Probabilistic Machine Learning: An Introduction*. MIT Press.

– (2023). *Probabilistic Machine Learning: Advanced Topics*. MIT Press.

Nesterov, Yu (Jan. 1998). "Semidefinite Relaxation and Nonconvex Quadratic Optimization". In: *Optimization Methods and Software* 9.1-3, pp. 141–160.

Pevzner, Pavel (2000). *Computational Molecular Biology: An Algorithmic Approach*. MIT press.

Risteski, Andrej (June 2016). "How to Calculate Partition Functions Using Convex Programming Hierarchies: Provable Bounds for Variational Methods". In: *Conference on Learning Theory*, pp. 1402–1416.

Wainwright, M.J., T.S. Jaakkola, and A.S. Willsky (July 2005). "A New Class of Upper Bounds on the Log Partition Function". In: *IEEE Transactions on Information Theory* 51.7, pp. 2313–2335.

Wainwright, Martin J. and Michael I. Jordan (2007). "Graphical Models, Exponential Families, and Variational Inference". In: *Foundations and Trends® in Machine Learning* 1.1–2, pp. 1–305.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Yes]

   (b) Complete proofs of all theoretical results. [Yes]

   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Not Applicable]

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

   (d) Information about consent from data providers/curators. [Not Applicable]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A COMPETING ALGORITHMS

A.1 The log-determinant relaxation

Jordan and Wainwright (2003) use the same variational inference principle as presented above, but with the following convex bound on entropy:

$$H(p) \leq \frac{1}{2} \log \det \left[\Sigma_p + \frac{1}{3} \begin{pmatrix} I_d & 0 \\ 0 & 0 \end{pmatrix} \right] + \frac{d}{2} \log \left(\frac{\pi e}{2} \right),$$

where Σ_p is the $d+1 \times d+1$ moment matrix for the feature vector $\varphi_0(x) = (1, x)$. It corresponds to bounding by the entropy of a Gaussian vector with similar covariance matrix. For the outer approximation of $\mathcal{K}$, they also take $\mathcal{K}'$, but they also propose to add additional linear constraints to have consistency with pairwise marginals. More precisely, it consists in enforcing the following inequalities, for all $i, j \in \{1, \ldots, n\}$ and for all $a, b \in \{-1, 1\}$:

$$1 + a\mu_i + b\mu_j + ab\mu_{ij} \geq 0,$$

where $\mu_i = \Sigma_{1,i+1}$ (corresponding to $\mathbb{E}_p[x_i]$ for Σ_p) and $\mu_{ij} = \Sigma_{i+1,j+1}$ (corresponding to $\mathbb{E}_p[x_i x_j]$ for Σ_p).

Note that this upper bound on entropy is not tight, even for independent variables, by construction, as we upper bound the entropy of a non Gaussian distribution by the one of a Gaussian with matching covariance matrix. Even for the uniform distribution, it is not tight. Indeed, for p the uniform distribution :

$$H(p) = d \log 2 < d \log \left(\sqrt{\frac{2\pi e}{3}} \right)$$

with $\sqrt{\frac{2\pi e}{3}} \approx 2.386$.

A.2 Tree-reweighted Message Passing (TRW)

Wainwright, Jaakkola, and Willsky (2005) reformulate the problem Eq. 5 in terms of moment vector $\mu = (\mu_i, \mu_{ij})$, where $\mu_i = \mathbb{E}_p[x_i]$ for $i \in \{1, \ldots, n\}$ and $\mu_{ij} = \mathbb{E}_p[x_i x_j]$, for $i, j \in \{1, \ldots, n\}$.

To construct a bound on entropy, they use the spanning tree polytope, which is the set of vectors $\rho = (\rho_{ij}) \in \mathbb{R}^{d(d-1)/2}$, also called edge appearance probabilities, that verify:

$$\rho_{ij} = \mathbb{E}_{\mathbb{P}}[\mathbb{1}_{(i,j) \in T}]$$

where $\mathbb{P}$ is a probability on spanning trees T of the MRF graph $G = (V, E)$ and $\mathbb{1}_{(i,j) \in T} \in \{0, 1\}$ indicates whether (i, j) is an edge of T. Then for such a vector ρ, they introduce the following bound on entropy:

$$H(\mu) \leq \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} \rho_{st} I_{st}(\mu_{st}),$$

and deduce an upper bound on the log-partition function as done above. For fixed ρ, they solve the variational problem with a message-passing scheme, and then they optimize over ρ with the conditional gradient algorithm (Frank and Wolfe, 1956).

B TIGHTNESS OF THE RELAXATION FOR ZERO TEMPERATURE AND SPECIAL STRUCTURE IN THE PARAMETERS

B.1 Independent variables

Taking $f(x) = \sum_i \theta_i x_i$ leads to the variable $x_1, \ldots, x_d$ being independent under the corresponding probability p.

Already with feature vector $\varphi_0(x) = (1, x)$, the constraints $\Sigma \in \mathcal{K}'$ implies the positive definiteness of the matrices:

$$\begin{pmatrix} 1 & \mu_i \\ \mu_i & 1 \end{pmatrix},$$

with $\mu_i = \Sigma_{1,i+1}$. It follows that $\forall i, -1 \leq \mu_i \leq 1$. In particular, there is a probability p' such that $\forall i, \mathbb{E}_{p'}[x_i] = \mu_i$. However, the other entries of Σ might not match with Σ_p.

In the case of optimization $\varepsilon = 0$, this does not matter as the matrix F that parameterizes a probability with independent variables is non-zero only for the entries $F_{1,i+1} = F_{i+1,1}$, and therefore

$$\sup_{\Sigma \in \mathcal{K}'} \operatorname{tr}(\Sigma F) = \sup_p \operatorname{tr}(\Sigma_p F).$$

B.2 Probability that factorizes over a tree

We now consider an MRF that factorizes over a tree, meaning that $f(x) = \sum_i \theta_i x_i + \sum_{(i,j) \in E} \theta_{ij} x_i x_j$ for some tree $T = (V, E)$.

Now taking feature vector $\varphi(x) = (\varphi_0(x), \varphi_+(x))$, where $\varphi_+(x) = (x_i x_j)_{ij \in E}$, the constraints $\Sigma \in \mathcal{K}'$ implies the positive definiteness of the matrices:

$$\begin{pmatrix} 1 & \mu_i \\ \mu_i & 1 \end{pmatrix}, \begin{pmatrix} 1 & \mu_i & \mu_j & \mu_{ij} \\ \mu_i & 1 & \mu_{ij} & \mu_j \\ \mu_j & \mu_{ij} & 1 & \mu_i \\ \mu_{ij} & \mu_j & \mu_i & 1 \end{pmatrix}.$$

with $\mu_i = \Sigma_{1,i+1}$, and $\mu_{ij} = \Sigma_{i+1,j+1}$, hence:

$$\mu_i \in [-1, 1],$$

$$\frac{1}{4}[1 + a\mu_i + b\mu_j + ab\mu_{ij}] \geq 0, \forall (i,j) \in E, \forall a, b \in \{-1, 1\}.$$

The μ_i's and μ_{ij}'s are pairwise consistent marginals over the edges and vertices of a tree, which implies that there is a probability p' over $\mathcal{X}$ such that $\mathbb{E}_{p'}[x_i] = \mu_i$ and $\mathbb{E}_{p'}[x_i x_j] = \mu_{ij}$ (see for example Proposition 2.1. of Wainwright and Jordan (2007)). As previously, this implies tightness of the bounds in the case of optimization $\varepsilon = 0$, but not in general.

C METRIC LEARNING

C.1 Quantum relaxation with metric learning

The quantum relaxation of the KL is given by:

$$D_V^{\mathrm{QT}}(A, B) = \operatorname{tr} V B^{1/2} h(B^{-1/2} A B^{-1/2}) B^{1/2},$$

with V such that $\varphi(x)^* V \varphi(x) \leq 1$. Can we optimize the bound with respect to V? In fact it corresponds to optimizing over linear transformation of the feature vector $\widetilde{\varphi}(x) = T\varphi(x)$ through $V = T^T T$ (Bach, 2024).

We limit ourselves to what we call *diagonal metric learning*, with $V = \operatorname{diag}(\eta), \eta \succcurlyeq 0, \sum_i \eta_i = 1$, for which $\varphi(x)^T V \varphi(x) = \sum_{\alpha \in I} \eta_\alpha (x^\alpha)^2 = 1$. It leads to define:

$$D_{\mathrm{diag}}^{\mathrm{QT}}(A, B) = \sup_{\eta \succcurlyeq 0, \sum_i \eta_i = 1} \operatorname{tr} \operatorname{diag}(\eta)(A \log B^{-1} A - A + B) = \max(\operatorname{diag}(A \log B^{-1} A - A + B)),$$

and

$$a_\varepsilon^{\text{diag}}(F) = \sup_{\Sigma \in \mathcal{K}'} \operatorname{tr} \Sigma F - \varepsilon \max(\operatorname{diag}(\Sigma \log(\Sigma) - \Sigma + I). \tag{10}$$

The convex problem defining $a_\varepsilon^{\text{diag}}(F)$ can be optimized with Kelley's method (Kelley, 1960). For more details, see the following subsection.

We experimentally found that the gain on the bound of the log-partition function were not very important albeit increased computational complexity. Therefore, it seems more interesting to simply take the quantum relaxation and add features with the greedy algorithm.

C.2   Optimization algorithm

C.2.1   Overview of the Kelley's method

Kelley's method (Kelley, 1960) solves optimization problems of the form:

$$\min \quad f(x)$$
$$\text{s.t.} \quad x \in \mathcal{X}.$$

We assume that we can compute subgradients of the convex function f for any $x \in \mathcal{X}$, and that we can solve linear optimization problems over $\mathcal{X}$ (in particular $\mathcal{X}$ must be compact).

From iterates $x_0, \ldots, x_t \in \mathcal{X}$, convexity of f gives that $\forall i \in \{0, \ldots, t\}, \forall x \in X, f(x) \geq f(x_i) + g_i \cdot (x - x_i)$, where $g_i \in \partial f(x_i)$, hence $\forall x \in X, f(x) \geq h_t(x)$ where $h_t(x) = \sup_{i=0,\ldots,t} f(x_i) + g_i \cdot (x - x_i)$. We construct next iterate x_{t+1} by solving the linear program $\min_{x \in \mathcal{X}} h_t(x)$, which gives a lower-bound on the optimal value.

Denoting $\hat{f}_t = \min_{x \in \mathcal{X}} h_{t-1}(x)$ and $\widetilde{f}_t = \min_{i=0,\ldots,t} f(x_i)$, $\hat{f}_t$ is a lower-bound on the optimal value, increasing with t and $\widetilde{f}_t$ is an upper-bound on the optimal value, decreasing with t, hence the natural stopping criterion $\widetilde{f}_t - \hat{f}_t < \text{tol}$.

C.2.2   Use in quantum relaxation with metric learning

To solve (10) with Kelley's method, we take $f : x = \Sigma \mapsto \operatorname{tr} \Sigma F - \varepsilon \max(\operatorname{diag}(\Sigma \log(\Sigma) - \Sigma + I)$ and constraints $\Sigma \in \mathcal{K}'$.

Solving the inner linear program and dealing with the constraints can be done with any optimization framework. In this work, we used CVXPY (Diamond and Boyd, 2016). We also need to find the expression of a subgradient of the function $g : \Sigma \mapsto \max(\operatorname{diag}(\Sigma \log(\Sigma) - \Sigma + I) = \max_i g_i(\Sigma)$, with $g_i(\Sigma) = e_i^T(\Sigma \log(\Sigma) - \Sigma + I)e_i = \operatorname{tr}(e_i e_i^T(\Sigma \log(\Sigma) - \Sigma + I))$ and $(e_i)_i$ is the canonical basis of $\mathbb{R}^n$.

Using result form section D, we have:

$$\nabla g_i(\Sigma) = \sum_{kl} \frac{h(\lambda_i) - h(\lambda_j)}{\lambda_i - \lambda_j} u_l^T e_i e_i^T u_k u_l u_k^T = \sum_{kl} (u_k)_i (u_l)_i \frac{h(\lambda_k) - h(\lambda_l)}{\lambda_k - \lambda_l} u_l u_k^T.$$

with $\Sigma = \lambda_i u_i u_i^T$ the orthogonal decomposition of sigma and $h : t \mapsto t \log t - t + 1$.

Then, with theorem 18.5 of Bauschke and Combettes (2011), it follows that $\partial g(\Sigma) = \operatorname{conv}(\nabla g_i(\Sigma))_{i \in \widetilde{I}}$, the convex envelope of $(\nabla g_i(\Sigma))_{i \in \widetilde{I}}$, where $\widetilde{I} = \{i : h(\Sigma)_{ii} = \max(\operatorname{diag}(h(\Sigma)))\}$.

D   DIFFERENTIATING SPECTRAL FUNCTIONS

To solve the optimization problems of C, we need to compute gradient of a function involving terms of the form $A \mapsto f(A)$. To achieve this, we use a trick based on Cauchy's residue formula due to Kato (1966).

Let A be a symmetric matrix and $f : \mathbb{C} \to \mathbb{C}$ a holomorphic function. We write $A = \sum_i \lambda_i u_i u_i^T$. With Cauchy's

residue formula, we get:

$$f(A) = \sum_i f(\lambda_i) u_i u_i^T$$
$$= \frac{1}{2i\pi} \oint_\gamma f(z)(zI - A)^{-1} dz,$$

where γ is a contour enclosing all eigenvalues. For a small perturbation Δ, the eigenvalues of $A + \Delta$ remain enclosed by γ, and:

$$f(A + \Delta) - f(A) = \frac{1}{2i\pi} \oint_\gamma f(z) \left[(zI - A - \Delta)^{-1} - (zI - A)^{-1} \right] dz$$
$$= \frac{1}{2i\pi} \oint_\gamma f(z)(zI - A)^{-1} \Delta (zI - A - \Delta)^{-1} dz$$
$$= \frac{1}{2i\pi} \oint_\gamma f(z)(zI - A)^{-1} \Delta (zI - A)^{-1} dz + o(\Delta)$$
$$= \frac{1}{2i\pi} \oint_\gamma \sum_{ij} \frac{f(z)}{(z - \lambda_i)(z - \lambda_j)} u_i u_i^T \Delta u_j u_j^T dz + o(\Delta)$$
$$= \sum_{ij} \frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j} u_i u_i^T \Delta u_j u_j^T + o(\Delta),$$

as $\frac{1}{(z - \lambda_i)(z - \lambda_j)} = \frac{1}{\lambda_i - \lambda_j} \left(\frac{1}{z - \lambda_i} - \frac{1}{z - \lambda_j} \right)$ for $\lambda_i \neq \lambda_j$, and denoting $\frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j} = f'(\lambda_i)$ if $\lambda_i = \lambda_j$. This leads to the following expression of the differential:

$$D_f(A) : \Delta \mapsto \sum_{ij} \frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j} u_i u_i^T \Delta u_j u_j^T.$$

For $g : A \mapsto \operatorname{tr} B f(A)$, where $B \in M_n$ the chain rule gives

$$D_g(A) : \Delta \mapsto \operatorname{tr}(B D_f(A)(\Delta))$$
$$= \operatorname{tr} \left(\left(\sum_{ij} \frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j} u_j^T B u_i u_j u_i^T \right) \Delta \right),$$

hence,

$$\nabla g(A) = \sum_{ij} \frac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j} u_j^T B u_i u_j u_i^T.$$

In particular, for $B = I$, the formula becomes $\nabla g(A) = f'(A)$.

E   ADDITIONAL EXPERIMENTS

E.1   Hierarchies on the KL divergence

Here we compare our different relaxations on the KL divergence for different level of hierarchies. We take Gaussian parameters, and $d = 5$, so that we can take up to all possible features. We plot the value of our relaxations of the KL divergence versus the size of the feature vector, starting with $\varphi_0(x) = (1, x)$ of size 6. For the quantum relaxation, we take features α with increasing numbers of non-zero coordinate, while for the quantum relaxation we select features with the greedy procedure presented in section 5.3.

From Fig. 4, we see that the greedy procedure allows to select interesting features quickly, and allows to reduce the gap with less features than adding features in arbitrary order. For the feature vector $\varphi(x) = (x^\alpha)_{\alpha \in \{0,1\}^d}$, the relaxation is tight.

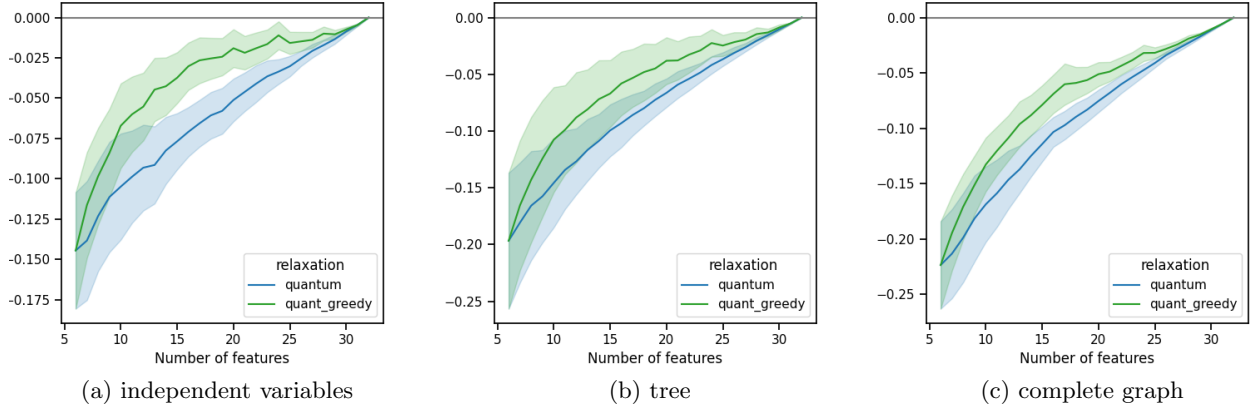(a) independent variables   (b) tree   (c) complete graph

Figure 4: Comparison of the different quantum relaxations of the KL divergence. $d = 5$, Gaussian parameters. x-axis corresponds to the number of features in the feature vector. Mean over 10 draws of parameters, ± 1 standard deviation.

E.2 TRW parameters

We make additional experiments in the setting of Wainwright, Jaakkola, and Willsky (2005), we call *TRW parameters*. It corresponds to $\theta_i \sim \mathcal{U}(-0.05, +0.05)$ independently, and distinguishes two cases for the couplings terms: $\theta_{ij} \sim \mathcal{U}(0, w)$ for *attractive coupling* and $\theta_{ij} \sim \mathcal{U}(-w, w)$ for *mixed coupling*, where w is called the *coupling strength*.

In Fig. 5, we use the TRW parameters on the complete graph. Our relaxations on the log-partition function are competitive with the TRW relaxation for low coupling strength. For mixed couplings, the greedy relaxation stays competitive with TRW even at higher coupling strength. The log-determinant relaxation exhibits a much higher error on the log-partition function. For the *l1_error*, our relaxations exhibit higher error than both the TRW and log-determinant relaxations.

We also carried out additional experiments with a larger value of d, $d = 16$. In this case, for the TRW relaxation, we use a fixed uniform value for ρ, ($\rho_{ij} = 2/d$) and stop message passing after a maximum of 10000 iterations.

In Fig. 6, we use the TRW parameters on the complete graph. For attractive coupling, our relaxations are not as good as competing algorithms as soon as the coupling strength increases. However, they stay competitive with the log-determinant relaxation for mixed coupling, while the TRW relaxation becomes significantly worse at high coupling strength. For the *l1_error*, our relaxations still exhibit higher error than both the TRW and log-determinant relaxations.

E.3 Gaussian parameters and varying temperature

We compare with competing algorithms on the Gaussian parameters for varying temperature. We take $d = 10$, and for the TRW relaxation, we use a fixed uniform value for ρ, ($\rho_{ij} = 2/d$) and stop message passing after a maximum of 10000 iterations.

In Fig. 7, we see that our relaxations are competitive with the log-determinant relaxation at low-temperature, and as the temperature increases it reaches a plateau whereas the error on the log-determinant relaxation grows linearly. They are also better than the TRW relaxation at low temperature and stay competitive as the temperature increase. For the *l1_error*, our relaxation are better than the log-determinant and TRW relaxation at low temperature, but worse at high temperature.
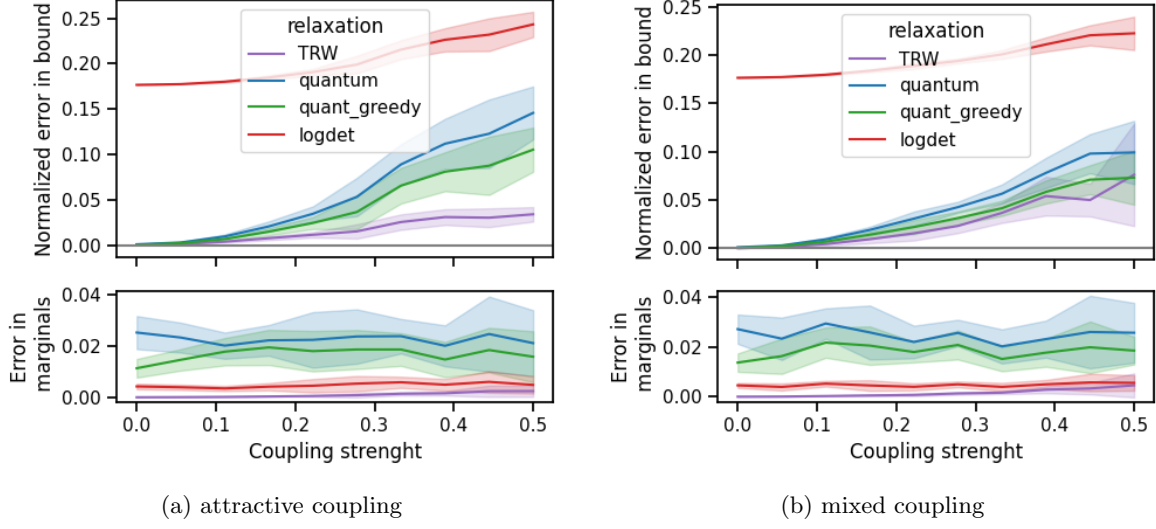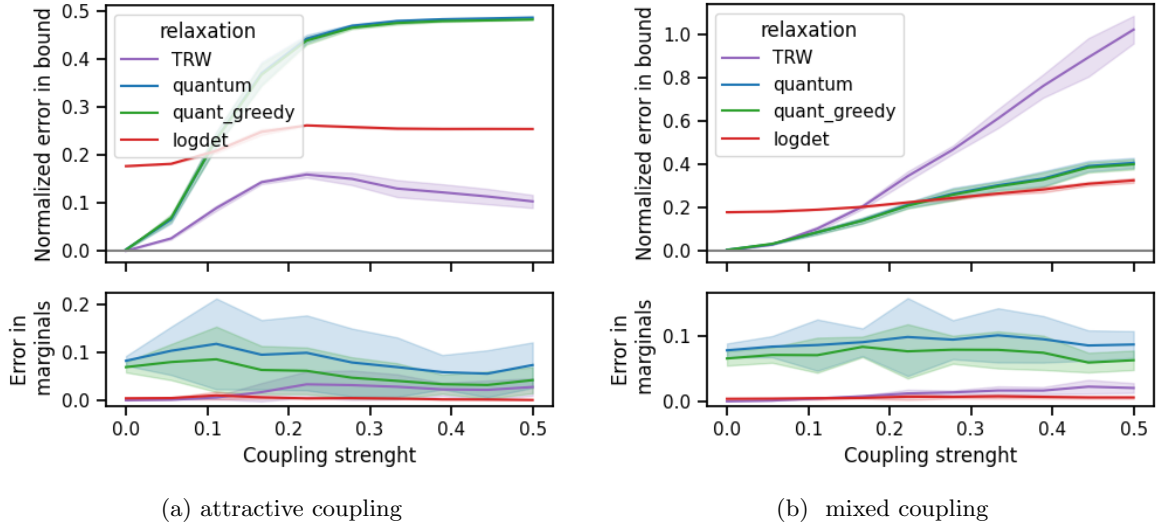
(a) attractive coupling

(b) mixed coupling

Figure 5: Comparison with competing algorithm for TRW parameters and varying coupling strength. $d = 5$, complete graph. Mean over 10 draws of parameters, ± 1 standard deviation.
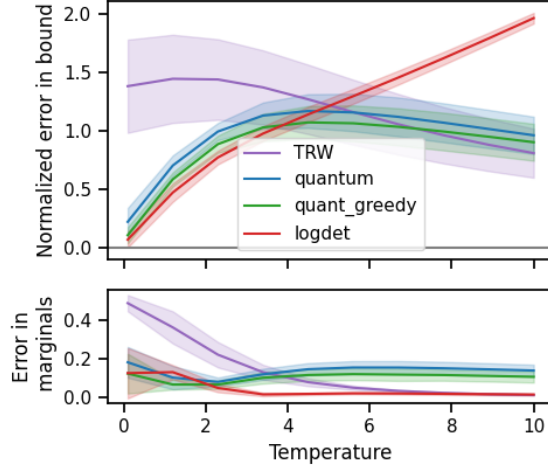

(a) attractive coupling

(b) mixed coupling

Figure 6: Comparison with competing algorithm for TRW parameters and varying coupling strength. $d = 16$, complete graph. Mean over 10 draws of parameters, ± 1 standard deviation.

Figure 7: Comparison with competing algorithms for Gaussian parameters and varying temperature. $d = 10$, complete graph. Mean over 10 draws of parameters, ± 1 standard deviation.

E.4 Number of added greedy features

We're now taking a closer look at the impact of the number of features added by the greedy procedure. To this end, we introduce the normalized relative gain in bound, defined by:

$$gain\_bound = \frac{bound - quantum\_bound}{d},$$

where $bound$ is the value of the bound with added features and $quantum\_bound$ is the bound for the regular quantum relaxation with $\varphi(x) = (1, x)$.

We take $d = 10$, and for the TRW relaxation, use a fixed uniform value for ρ, $(\rho_{ij} = 2/d)$, and stop message passing after a maximum of 10000 iterations.

Experiments with Gaussian parameters are plotted in Fig. 8. In plot (a), we observe that additional features increase the normalized relative gain in bound over the whole range of temperature. In plot (b), we see the gains for $k = 3$ and $k = 10$ when compared with the log-determinant and TRW relaxations. Adding 10 extras features makes the relaxation better than the two competing algorithms.

Experiments with log-det parameters are plotted in Fig. 9 and Fig. 10. In Fig. 9, we observe that additional features increase the normalized relative gain in bound over the whole range of coupling strength, and type of couplings, with a larger gain at high coupling strength. In Fig. 10, we see the gains for $k = 3$ and $k = 10$ when compared with the log-determinant and TRW relaxations. This gains allow to beat the log-determinant relaxation at high coupling strength for both mixed coupling and repulsive coupling, and mitigate the gap between our relaxations and competing algorithm for attractive coupling.

E.5 Higher dimension

We carried additional experiments with higher value of d. As the complexity of the brute force solution growth exponentially fast with d, we can't compute the exact value of the log-partition function and marginals. Therefore, we compare our bound to the log-determinant relaxation, by defining:

$$relative\_error\_bound = \frac{bound - bound\_log\_det}{d}.$$

When this value is negative, relaxation is better than log-determinant relaxation, when it is positive, relaxation is worse.

We take $d = 30$, and for the TRW relaxation, we use a fixed uniform value for ρ, $\rho_{ij} = 2/d$ (such that $\sum_{ij} \rho_{ij} = d - 1$), and stop message passing after a maximum of 10000 iterations.

(a) Normalized gain in bound
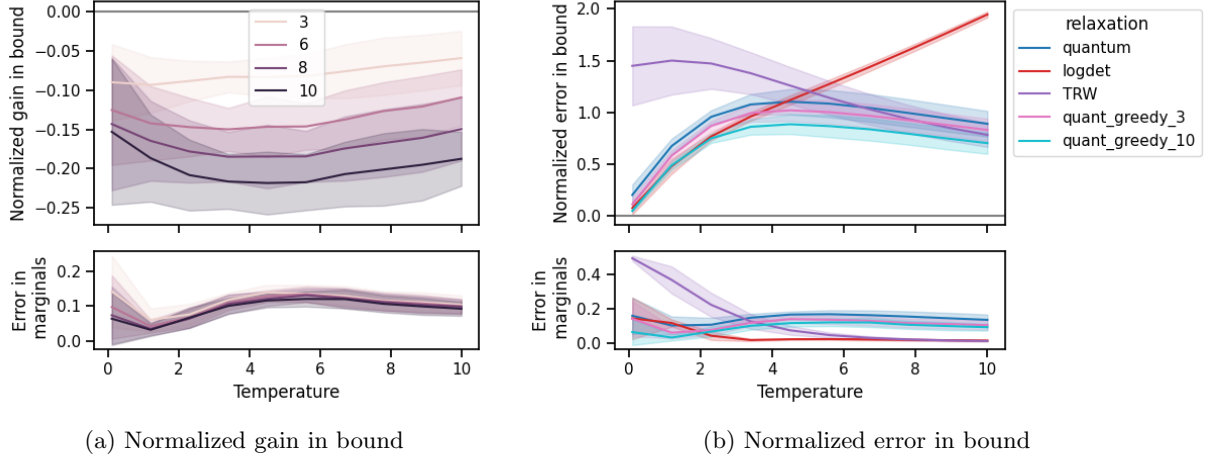
(b) Normalized error in bound

Figure 8: (a) Normalized gain in bound compared to quantum relaxation for different number of features added with the greedy procedure. (b) Comparison with competing algorithms. $d = 10$, complete graph, Gaussian parameters. Mean over 10 draws of parameters, ± 1 standard deviation.
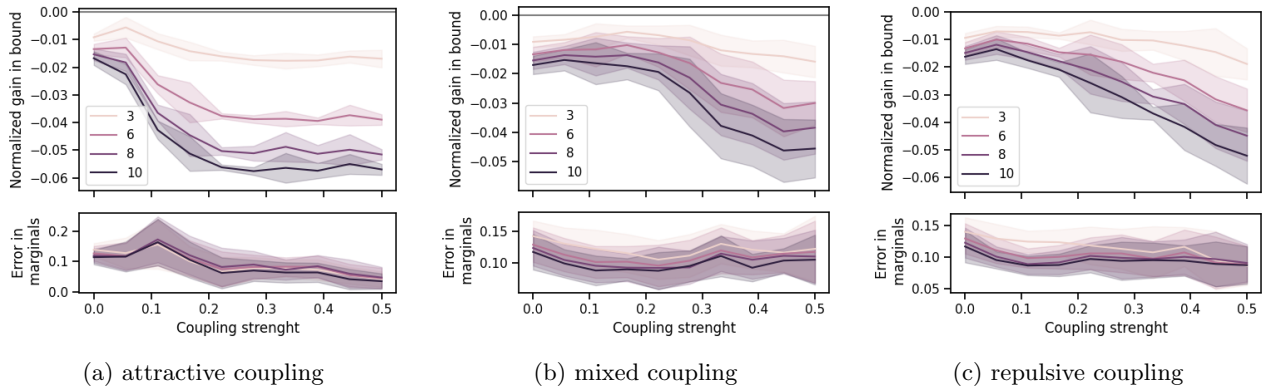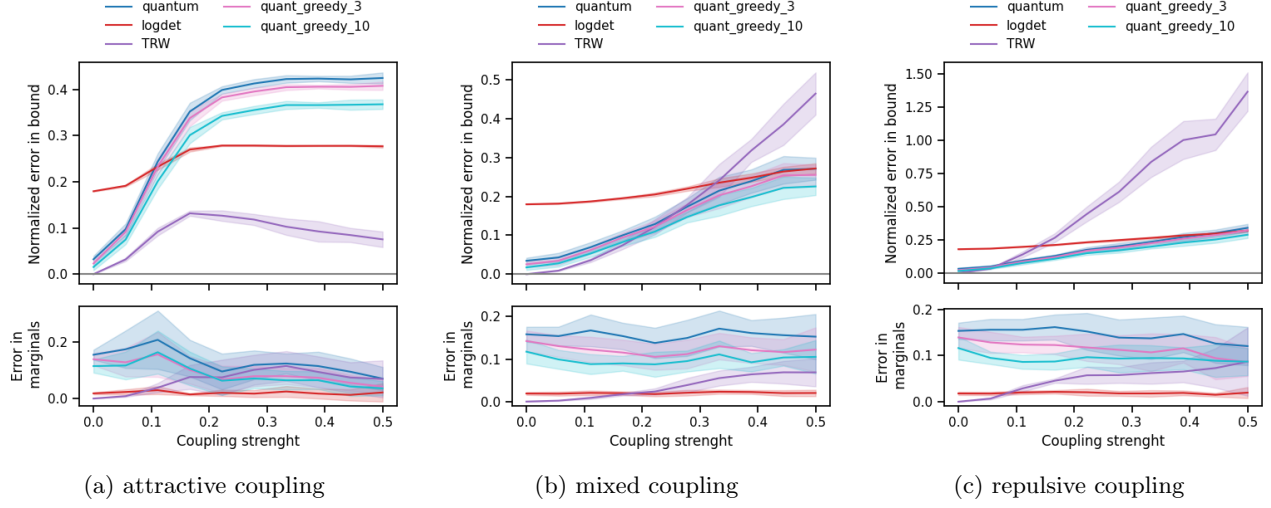

(a) attractive coupling

(b) mixed coupling

(c) repulsive coupling

Figure 9: Normalized gain in bound compared to quantum relaxation for different number of features added with the greedy procedure. $d = 10$, complete graph, log-det parameters. Mean over 10 draws of parameters, ± 1 standard deviation.

Figure 10: Comparison with competing algorithms. $d = 10$, complete graph, log-det parameters. Mean over 10 draws of parameters, ± 1 standard deviation.
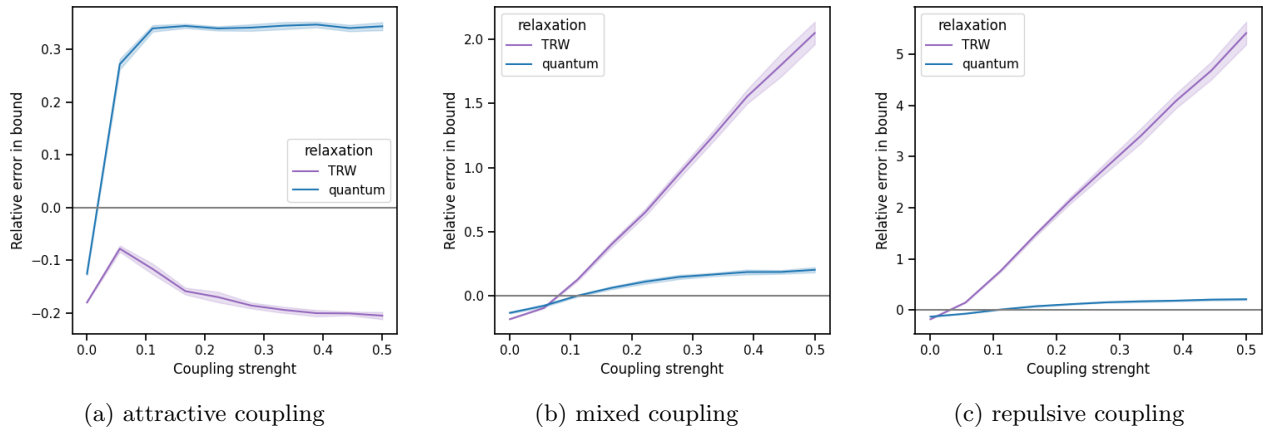

Figure 11: Comparison with competing algorithm for log-det parameters and varying coupling strength. $d = 30$, complete graph. Mean over 10 draws of parameters, ± 1 standard deviation.
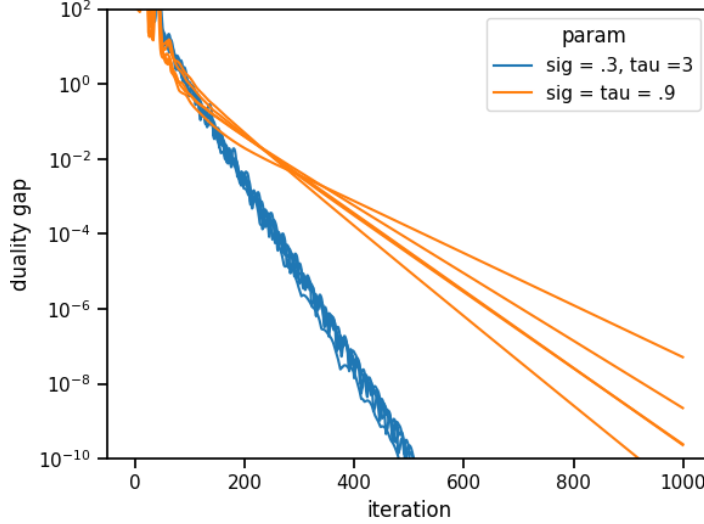
Figure 12: Duality gap along iterations of Chambolle-Pock algorithm, for 5 draws of models parameters and different values of τ and σ. $d = 50$, Gaussian parameters, complete graph, $\varphi(x) = (1, x)$. y-axis is logarithmic scale.

In Fig. 11, we observe results that are similar to Fig. 3. For attractive coupling, our relaxations are not as good as competing algorithms as soon as the coupling strength increase, but stay competitive with the log-determinant relaxation for mixed coupling and repulsive coupling, while the performance of the TRW relaxation deteriorates.

E.6   Convergence of Chambolle-Pock algorithm

We studied empirically the number of iterations of Chambolle-Pock algorithm with respect to d and the tolerance on the duality gap δ.

For feature vector $\varphi(x) = (1, x)$, in Fig. 12, we observe that the duality gap decrease exponentially fast with the number of iterations t. Hence the number of iterations is $O(\log(\frac{1}{\delta}))$.

We then study the dependence of the constant with regard to d, for $\varphi(x) = (1, x)$. We took random samples of the models parameters according to a normal distribution $\theta \sim \mathcal{N}(0, I)$, such that the scale of the individuals θ_i, θ_{ij} does not vary with d : $\theta_i, \theta_{ij} = O(1)$. In Fig. 13, we observe that the number of iterations to reached a fixed tolerance of 10^{-8} grows linearly with d.

In Fig. 14, we fixed $d = 10$, and take a feature vector of size $n = 50$ (features are chosen in arbitrary order, by increasing degree). For this more general feature vector, we do not observe this linear convergence in δ, and can only hope to get the theoretical rate in $O(\frac{1}{\delta})$. Reaching arbitrary small tolerance becomes much more time consuming than in the case of $\varphi(x) = (1, x)$, but we observe that we can still reach lower precision in reasonable time. In Fig. 15, we fixed $d = 10$, and we observe that the number of iterations to reached a fixed tolerance of 10^{-2} for various values of n. We observed a general growth in $O(n^2)$, though for some draws and values of n, the number of iterations can be much bigger.
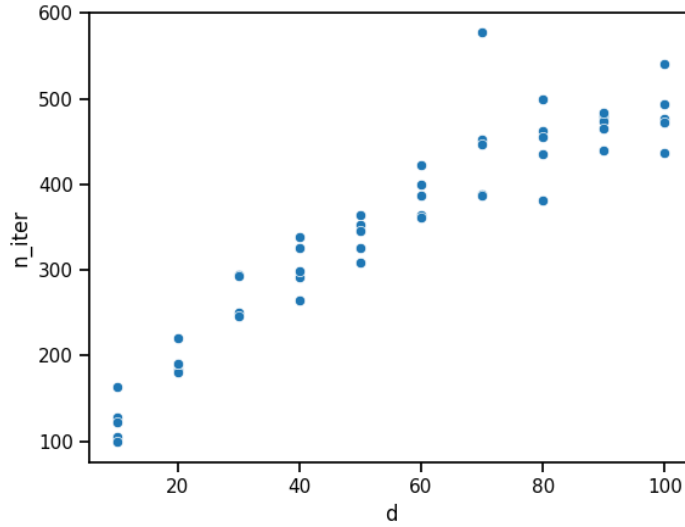
Figure 13: Number of iterations to reach a duality gap $< 10^{-8}$, for various values of d. 5 draws of models parameters at each value d, Gaussian parameters, complete graph, $\varphi(x) = (1, x)$.
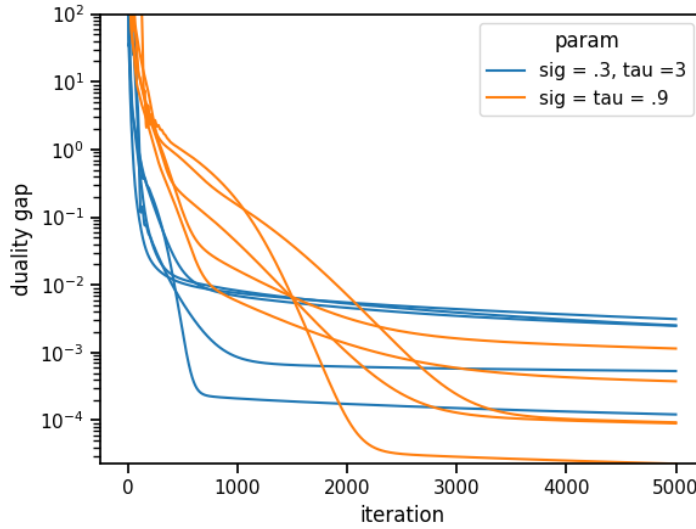

Figure 14: Duality gap along iterations of Chambolle-Pock algorithm, for 5 draws of models parameters and different values of τ and σ. $d = 50$, Gaussian parameters, complete graph, $\varphi(x) = (1, x)$. y-axis is logarithmic scale.
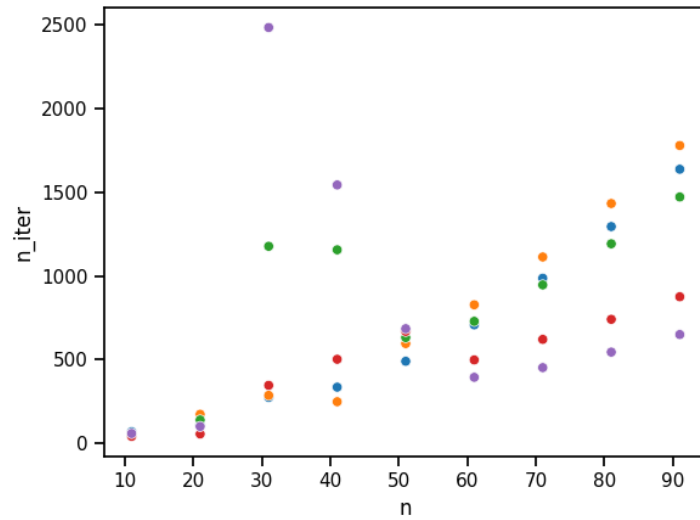
Figure 15: Number of iterations to reach a duality gap $< 10^{-2}$, for various values of n, with fixed d. 5 draws of models parameters, $d = 10$, Gaussian parameters, complete graph, $\varphi(x) = (1, x)$.