Discrete Integration by Decoding Binary Codes

Stefano Ermon, Carla P. Gomes {ERMONSTE,GOMES}@CS.CORNELL.EDU
Dept. of Computer Science, Cornell University, Ithaca NY 14853, U.S.A.

Ashish Sabharwal ASHISH.SABHARWAL@US.IBM.COM
IBM Watson Research Center, Yorktown Heights, NY 10598, U.S.A.

Bart Selman SELMAN@CS.CORNELL.EDU
Dept. of Computer Science, Cornell University, Ithaca NY 14853, U.S.A.

Abstract

Many probabilistic inference and learning tasks involve summations over exponentially large sets. Recently, it has been shown that these problems can be reduced to solving a polynomial number of MAP inference queries for a model augmented with randomly generated parity constraints. By exploiting a connection with max-likelihood decoding of binary codes, we show that these optimizations are computationally hard. Inspired by iterative message passing decoding algorithms, we propose an Integer Linear Programming (ILP) formulation for the problem, enhanced with new sparsification techniques to improve decoding performance. By solving the ILP through a sequence of LP relaxations, we get both lower and upper bounds on the partition function, which hold with high probability and are much tighter than those obtained with variational methods.

1. Introduction

Discrete probabilistic graphical models (Wainwright & Jordan, 2008; Koller & Friedman, 2009) are often defined up to a normalization factor involving a summation over an exponentially large combinatorial space. Computing these factors is an important problem, as they are needed, for instance, to evaluate the probability of evidence, rank two alternative models, and learn parameters from data. Similar high-dimensional discrete integrals also arise when computing expectations with respect to the model, e.g. of a sufficient statistic for parameter learning. Unfortunately, computing these discrete integrals exactly in very high dimensional spaces quickly becomes intractable, and approximation techniques are often needed. Among them, sampling and variational methods are the most popular approaches. Variational inference problems are typically solved using message passing techniques, which are often guaranteed to converge to some local minimum (Wainwright & Jordan, 2008), but without guarantees on the quality of the solution found. Markov Chain Monte Carlo (Jerrum & Sinclair, 1997; Madras, 2002; Wei & Selman, 2005) are asymptotically correct, but the number of samples required to obtain a statistically reliable estimate can grow exponentially in the worst case.

Recently, Ermon et al. (2013) introduced a new technique called WISH which comes with provable (probabilistic) guarantees on the approximation error. Their method combines combinatorial optimization techniques with the use of universal hash functions to uniformly partition a large combinatorial space, originally introduced by Valiant and Vazirani to study the Unique Satisfiability problem and later exploited by Gomes et al. (2006a;b) for solution counting. Specifically, they show that one can obtain the intractable normalization constant (partition function) of a graphical model within any desired degree of accuracy, by solving a polynomial number of MAP queries for the original graphical model augmented with randomly generated parity constraints as evidence. Although MAP inference is NP-hard and thus also intractable, this is a significant step forward as counting problems such as estimating the partition function are #-P hard, a complexity class believed to be significantly harder

Presented at the International Conference on Machine Learning (ICML) workshop on *Inferning: Interactions between Inference and Learning*, Atlanta, Georgia, USA, 2013.

than NP.

In this work, we investigate the class of MAP inference queries with random parity constraints arising from the WISH scheme. These optimization problems turn out to be intimately connected with the fundamental problem of maximum likelihood decoding of a binary code (Vardy, 1997; Berlekamp et al., 1978). We leverage this connection to show that the inference queries generated by WISH are NP-hard to solve and to approximate, even for very simple graphical models. Although generally hard in the worst case, message passing and related linear programming techniques (Feldman et al., 2005) are known to be very successful in practice in decoding certain types of codes such as low density parity check (LDPC) codes (Gallager, 1962). Inspired by the success of these methods, we formulate the MAP inference queries generated by WISH as Integer Linear Programs (ILP). Unfortunately, such queries are typically harder than traditional decoding problems because they involve more complex probabilistic models, and because universal hash functions naturally give rise to very "dense" parity constraints. To address this issue, we propose a technique to construct equivalent but sparser (and empirically easier to solve) parity constraints. Our ILP formulation with sparsification techniques provides very good lower bounds on the partition function, while at the same time providing also upper bounds based on solving a sequence of LP relaxations. These upper bounds are much tighter than those obtained by tree decomposition and convexity (Wainwright, 2003).

2. Problem Statement and Assumptions

We consider a discrete probabilistic graphical model (Wainwright & Jordan, 2008) with $n = |V|$ random variables $\{x_i, i \in V\}$ where each random variable x_i takes values in a finite set $\mathcal{X}_i$. We consider a factor graph representation for a joint probability distribution over elements $x \in \mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ (also referred to as **configurations**)

$$p(x) = \frac{1}{Z} \prod_{\alpha \in \mathcal{I}} \psi_\alpha(\{x\}_\alpha) \tag{1}$$

This is a compact representation for $p(x)$, which is defined as the product of potentials or factors $\psi_\alpha : \{x\}_\alpha \mapsto \mathbb{R}^+$, where $\mathcal{I}$ is an index set and $\{x\}_\alpha \subseteq V$ a subset of variables the factor ψ_α depends on. Z is a normalization constant known as *partition function* ensuring the probabilities sum up to one. Formally the partition function Z is defined as

$$Z = \sum_{x \in \mathcal{X}} \prod_{\alpha \in \mathcal{I}} \psi_\alpha(\{x\}_\alpha) = \sum_{x \in \mathcal{X}} w(x) \tag{2}$$

where for compactness we have introduced a weight function $w : \mathcal{X} \to \mathbb{R}^+$ that assigns to each configuration $x \in \mathcal{X}$ its unnormalized probability, namely

$$w(x) = \prod_{\alpha \in \mathcal{I}} \psi_\alpha(\{x\}_\alpha) \tag{3}$$

Computing the partition function Z is a #-P complete, intractable problem because it generalizes #-SAT. However, the partition function is a key property of a graphical model, needed e.g. to actually evaluate the probability of a configuration x under p.

2.1. Parameter Learning

Many algorithms for maximum-likelihood learning of parameters in graphical models involve estimating the gradient of the log-likelihood. The gradient is given by the difference between the *empirical mean parameters* (according to the data) and the *mean parameters*, i.e. expected value of the sufficient statistics according to the model. Computing a *mean parameter* $\sum_x \phi(x)p(x)$ for a sufficient statistic ϕ is generally intractable because it involve a sum over an exponentially large set of items as in (2), although with a different weight function $w'(x) \triangleq \phi(x)p(x)$.

In this paper, we will focus on approximate techniques to estimate and bound weighted sums over exponentially large sets of items. For simplicity, we consider the case of binary variables where $x_i \in \mathcal{X}_i = \{0, 1\}$. The general case can be encoded using a bit representation and binary variables.

3. Background

This paper extends previous work by Ermon et al. (2013) who introduced an algorithm called WISH to estimate the partition function (2). WISH is a randomized approximation algorithm that gives a constant factor approximation of Z with high probability. It involves solving a polynomial number of MAP inference queries for the graphical model conditioned on randomly generated evidence based on hashing.

3.1. The `WISH` Algorithm for Discrete Integration

The basic idea behind WISH is to (implicitly) randomly partition the space of all possible configurations by universally hashing configurations into 2^m

buckets. This step is achieved using randomly generated *parity constraints* of the form $Ax = b \mod 2$, which may also be viewed as logical XOR operations acting on the binary variables of the problem: $A_{i1}x_1 \oplus A_{i2}x_2 \oplus \cdots \oplus A_{in}x_n = b_i$. A combinatorial optimization solver is then used to find a configuration with the largest weight within a *single* bucket. This corresponds to solving a MAP query, i.e., solving an optimization problem subject to (randomly generated) parity constraints. By varying the number of buckets and repeating the process a small number of times, this strategy provably yields an estimate of the intractable normalization factor (2) within any desired degree of accuracy, with high probability and using only a polynomial number of MAP queries. For completeness, we provide the pseudocode for WISH as Algorithm 1.

Although MAP inference itself is an NP-hard problem, this strategy is still desirable considering that computing Z is a #P-hard problem, a complexity class believed to be even harder than NP. In practice, Ermon et al. (2013) showed that the resulting MAP inference can be solved reasonably well using a state-of-the-art MAP inference engine called Toulbar (Allouche et al., 2010), which was extended with custom propagators for parity constraints.

Algorithm 1 WISH ($w : \Sigma \to \mathbb{R}^+, n = \log_2 |\Sigma|, \delta, \alpha$)

$T \leftarrow \left\lceil \frac{\ln(n/\delta)}{\alpha} \right\rceil$
for $i = 0, \cdots, n$ **do**
  for $t = 1, \cdots, T$ **do**
    Sample hash function $h^i_{A,b} : \Sigma \to \{0,1\}^i$, i.e. sample uniformly $A \in \{0,1\}^{i \times n}$, $b \in \{0,1\}^i$
    $w_i^t \leftarrow \max_\sigma w(\sigma)$ subject to $A\sigma = b \mod 2$
  end for
  $M_i \leftarrow \text{Median}(w_i^1, \cdots, w_i^T)$
end for
Return $M_0 + \sum_{i=0}^{n-1} M_{i+1} 2^i$

Theorem 1 ((Ermon et al., 2013)). *For any $\delta > 0$, positive constant $\alpha \leq 0.0042$, WISH makes $\Theta(n \ln n/\delta)$ MAP queries and, with probability at least $(1-\delta)$, outputs a 16-approximation of $Z = \sum_{\sigma \in \mathcal{X}} w(\sigma)$.*

Further, even if the MAP instances in the inner loop of Algorithm 1 are not solved to optimality, the output of the algorithm using suboptimal MAP solutions is an *approximate lower bound* for Z (specifically, no more than $16Z$) with probability at least $(1 - \delta)$. If suboptimal solutions are within a constant factor L of the optimal, then the output is a $16L$-approximation of Z with probability at least $(1 - \delta)$ (Ermon et al., 2013). Similarly, if one has access to upper bounds to the values of the MAP instances, the output of the algorithm using these upper bounds is an *approximate upper bound* (specifically, at least $1/16Z$) for Z with probability at least $(1 - \delta)$.

4. Connections with Coding Theory

For a problem with n binary variables, WISH requires solving $\Theta(n \log n)$ optimization instances. If these optimizations could be approximated (within a constant factor of the true optimal value) in polynomial time, this would give rise to a polynomial time algorithm that gives, with high probability, a constant factor approximation for the original counting problem. Note that this is a reasonable assumption, because perhaps the most interesting #-P complete counting problems are those whose corresponding decision problem are easy, e.g. counting weighted matchings in a graph (computing the permanent). A natural question arises: *are there interesting counting problems for which we can approximate $\max_\sigma w(\sigma)$ subject to $A\sigma = b \mod 2$ in polynomial time?*

To shed some light on this question, we show a connection with a decision problem arising in coding theory:

Definition 1 (MAXIMUM-LIKELIHOOD DECODING)**.** Given a binary $m \times n$ matrix A, a vector $b \in \{0,1\}^m$, and an integer $w > 0$, is there a vector $z \in \{0,1\}^n$ of Hamming weight $||z||_1 \leq w$, such that $Az = b \mod 2$?

As noted in (Vardy, 1997), Berlekamp, McEliece, and van Tilborg (Berlekamp et al., 1978) showed that this problem is NP-complete with a reduction from 3-DIMENSIONAL MATCHING. Further, Stern (Stern, 1993) and Arora, Babai, Stern, Sweedyk (Arora et al., 1993) proved that approximating within any constant factor the solution to MAXIMUM-LIKELIHOOD DECODING is also NP-hard.

These hardness results restrict the kind of problems we can hope to solve in our setting, which is more general. In fact, we can define a graphical model with factors $\psi_i(x_i) = \exp(-1)$ if $x_i = 1$, $\psi_i(x_i) = 1$ if $x_i = 0$. Let $\mathcal{S} = \{x \in \{0,1\}^n : Ax = b \bmod 2\}$. Then

$$\max_{x \in \mathcal{S}} w(x) = \max_{x \in \mathcal{S}} \prod_{i=1}^n \psi_i(x_i) = \exp\left(\max_{x \in \mathcal{S}} \sum_{i=1}^n \log \psi_i(x_i)\right) = \exp\left(\max_{x \in \mathcal{S}} -H(x)\right) = \exp\left(-\min_{x \in \mathcal{S}} H(x)\right)$$

where $H(x)$ is the Hamming weight of x. Thus, MAXIMUM-LIKELIHOOD DECODING of a binary code is a special case of MAP inferences subject parity constraints, but on a simple (disconnected) factor graph with potentials acting only on single variable nodes. Intuitively, in the context of coding theory, there is a variable for each transmitted bit, and factors capture the probability of a transmission error on each bit. Thus there are no interactions between the variables, except for the ones introduced by the parity constraints $Ax = b \mod 2$, while in our context we allow for more complex probabilistic dependencies between variables specified as in Eq. (1). We therefore have the following theorem:

Theorem 2. *Given $A \in \{0,1\}^{m\times n}$, $b \in \{0,1\}^m$, and $w(x)$ as in Eqn. (3), the optimization problem*

$$\max_{x\in\{0,1\}^n} \log w(x) \text{ subject to } Ax = b \mod 2$$

is NP-hard to solve and to approximate within any constant factor.

5. Integer Programming Formulation

The NP-hard combinatorial optimization problem $\max_\sigma w(\sigma)$ subject to $A\sigma = b \mod 2$ can be formulated as an Integer Program (Bertsimas & Tsitsiklis, 1997). This is a promising approach because Integer Linear Programs and related Linear programming (LP) relaxations have been shown to be a very effective at decoding binary codes by Feldman et. al (Feldman et al., 2005). Further, the empirically successful iterative message-passing decoding algorithms are closely related to LP relaxations of certain Integer Programs, either because they are directly trying to solve an LP or its dual like the MPLP and TRWBP (Globerson & Jaakkola, 2007; Sontag et al., 2008; Wainwright, 2003), or attempting to approximately solve a variational problem over the same polytope like Loopy Belief Propagation (Wainwright & Jordan, 2008).

5.1. MAP Inference as an ILP

For simplicity, we consider the case of binary factors (pairwise interactions between variables), where equation (3) simplifies to $w(x) = \prod_{i\in V} \psi_i(x_i) \prod_{(i,j)\in E} \psi_{ij}(x_i, x_j)$ for some edge set E. Rewriting in terms of the logarithms $\theta = \log \psi$, the unconstrained MAP inference problem can be stated as $\max_{x\in\{0,1\}^n} \sum_{i\in V} \theta_i(x_i) + \sum_{(i,j)\in E} \theta_{ij}(x_i, x_j)$ which can be written as an Integer Linear Program using binary indicator variables $\{\mu_i, i \in V\}$ and $\{\mu_{ij}(x_i, x_j), (i, j) \in E, x_i \in \{0, 1\}, x_j \in \{0, 1\}\}$ as follows (Wainwright & Jordan, 2008):

$$\max_{\mu_i,\mu_{ij}(x_i,x_j)} \sum_{i\in V} \theta_i(1)\mu_i + \theta_i(0)(1 - \mu_i) + \sum_{(i,j)\in E} \sum_{x_i,x_j} \theta_{ij}(x_i, x_j)\mu_{i,j}(x_i, x_j)$$

subject to $\forall i \in V, (i, j) \in E, \sum_{x_j\in\{0,1\}} \mu_{i,j}(0, x_j) = 1 - \mu_i$; $\forall i \in V, (i, j) \in E, \sum_{x_j\in\{0,1\}} \mu_{i,j}(1, x_j) = \mu_i$; $\forall i \in V, (i, j) \in E, \sum_{x_i\in\{0,1\}} \mu_{i,j}(x_i, 0) = 1 - \mu_j$; $\forall i \in V, (i, j) \in E, \sum_{x_i\in\{0,1\}} \mu_{i,j}(x_i, 1) = \mu_j$.

5.2. Parity Constraints

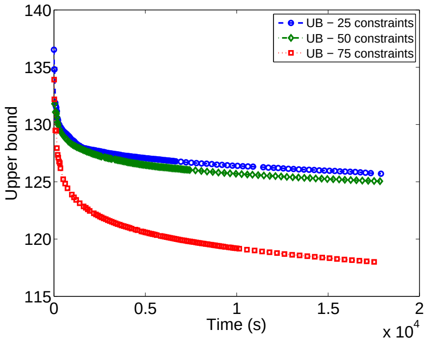
We now present an encoding for the parity constraints $A\sigma = b \mod 2$, defining the so called **parity polytope** over $\sigma \in \mathbb{R}^n$. Let $\mathcal{J}$ be the set of parity constraints (one entry per row of A). Let $\mathcal{N}(j)$ be the set of variables the j-th parity constraint depends on, namely the indexes of the non-zero columns of the j-th row of A . We'll refer to $|\mathcal{N}(j)|$ as the length of the j-th XOR.

Yannakakis (Yannakakis, 1991) introduced the following compact representation which requires only $O(n^3)$ variables and constraints, where n is the number of variables. For each constraint j, define $T_j = \{0, 2, \cdots, 2\lfloor|\mathcal{N}(j)|/2\rfloor\}$ as the set of even numbers between 0 and $|\mathcal{N}(j)|$. For all $j \in \mathcal{J}$ and for all $k \in T_j$ we have a binary variable $\alpha_{j,k} \in \{0, 1\}$. For all $j \in \mathcal{J}$ and for all $k \in T_j$ and for all $i \in \mathcal{N}(j)$ we have a binary variable $z_{i,j,k} \in \{0, 1\}$, $0 \le z_{i,j,k} \le \alpha_{j,k}$. Then the following constraints are enforced: $\forall i \in V, j \in \mathcal{N}(i), \mu_i = \sum_{k\in T_j} z_{i,j,k}$; $\forall j \in \mathcal{J}, \sum_{k\in T_j} \alpha_{j,k} = 1$; $\forall j \in \mathcal{J}, \forall k \in T_j, \sum_{i\in\mathcal{N}(j)} z_{i,j,k} = k\alpha_{j,k}$.

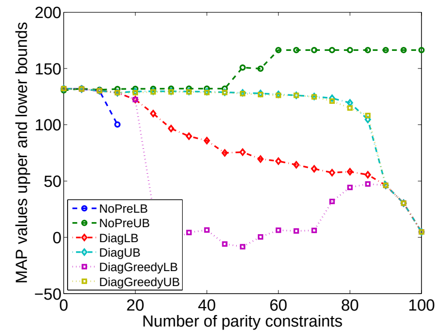
5.3. Solving Integer Programs

Solving ILPs typically relies on solving a sequence of Linear Programming (LP) relaxations obtained by by relaxing the integrality constraints. The solution to the relaxation provides an upper bound to the original integer maximization problem. Since LP can be solved in polynomial time, using Theorem 1 and following remarks we have a **polynomial time** method to obtain approximate upper bounds on the partition function which hold with high probability, although without tightness guarantees.

IP solvers such as IBM CPLEX solve a sequence of LP relaxations based on branching on the problems's variables, iteratively improving the upper bound and keeping track of the best integer solution found, until lower and upper bounds match. Thus, one advantage of using an IP solver over standard Message Passing techniques is that the upper and lower bounds improve over time, and it is guaranteed to eventually provide an optimal solution for the original integer problem. In Figure 1(a) we plot the upper bound reported by CPLEX as a function of runtime for a random 10×10 Ising model with mixed interactions. It's clear that there is quickly a dramatic improvement over the value



(a) MAP upper bound as a function of runtime.



(b) MAP Upper and lower bounds with and without sparsification.

Figure 1. Bounds on MAP inference subject to parity constrains obtained from the ILP formulation.

of the basic LP relaxation, which is the value reported by CPLEX around time zero.

5.4. Inducing Sparsity

As we have shown, solving MAP inference queries subject to parity constraints is hard in general. However, adding parity constraints can sometimes makes the optimization easier. For example, when A is the identity matrix, enforcing $A\sigma = b \mod 2$ corresponds to fixing the values of all variables and leads to a trivial optimization problem. Empirically, **sparse** constraints, such as the ones used in low density parity check (LDPC) codes from Gallager (Gallager, 1962), tend to be much easier to solve. Unfortunately, constructions to create pairwise independent hash functions require constraints of average length $n/2$.

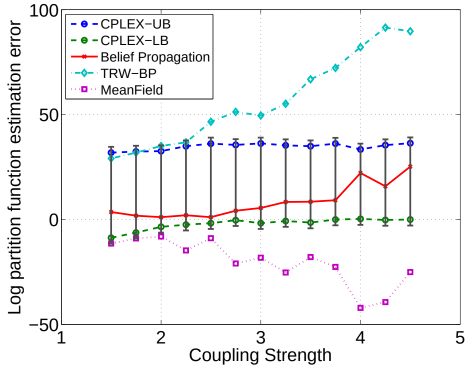
In this paper we propose to rewrite the constraints in a form that is equivalent, i.e. defines the same set of solutions, but is easier to solve. Specifically, given a a set of parity constraints specified through matrices A, b we look for matrices A', b' that define the same set of solutions, namely $\{x \in \{0,1\}^n : Ax = b\} = \{x \in \{0,1\}^n : A'x = b'\}$ but are much sparser. We propose to use two approaches:
1) Perform Gauss-Jordan elimination on $[A|b]$ to convert $[A|b]$ to **reduced row echelon form**.
2) Try all combinations of up to k rows $r_1, \cdots, r_k$ of $[A|b]$, and if their sum $r_1 \oplus \cdots \oplus r_k$ is sparser than any of the r_i, substitute r_i with $r_1 \oplus \cdots \oplus r_k$.
Both techniques are based on **elementary row operations** and therefore are guaranteed to maintain the solution set $\mathcal{S}$ and to improve sparsity.

In Figure 1(b) we show the median upper and lower bounds found by CPLEX for several randomly generated constraints on a random 10×10 Ising grid model with mixed interactions. We run CPLEX for 10 minutes with and without sparsification, reporting the best upper and lower bounds found. We see that without any preprocessing (NoPre) CPLEX fails at finding any integer solution when there are more than 15 parity constraints. Performing Gauss-Jordan elimination (Diag) significantly improves both the upper bound and the lower bound. The effect is particularly significant for a large number of constraints, when the reduced row echelon form of A is close to the identity matrix. Adding the additional greedy substitution step (DiagGreedy, looking at all combinations of up to $k = 4$ rows) slightly improves the quality of the upper bound, but the lower bound significantly degrades. Therefore, for the rest of the paper we will use only Gauss-Jordan elimination preprocessing.

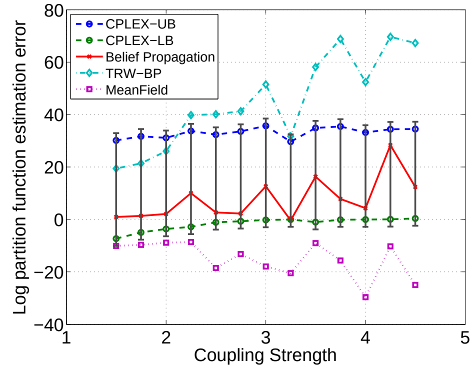
6. Experiments

We evaluate the performance of WISH augmented with CPLEX to solve the ILP formulation of the MAP queries. All the optimization instances are solved in parallel on a compute cluster, with a timeout of 10 minutes on Intel Xeon 5670 3GHz machines with 48GB RAM. We use Gauss-Jordan elimination to preprocess the parity constraints to improve the quality of the LP relaxations. We evaluate the lower bound and upper bound estimates for the partition functions of $M \times M$ grid Ising models for $M \in \{10, 15\}$, with random interactions (positive and negative) and external field $f \in \{0.1, 1.0\}$. Specifically, there are M^2 binary variables, with single node potentials $\psi_i(x_i) = \exp(f_i x_i)$ and pairwise interactions $\psi_{ij}(x_i, x_j) = \exp(w_{ij} x_i x_j)$, where $w_{ij} \in_R [-w, w]$ and $f_i \in_R [-f, f]$.

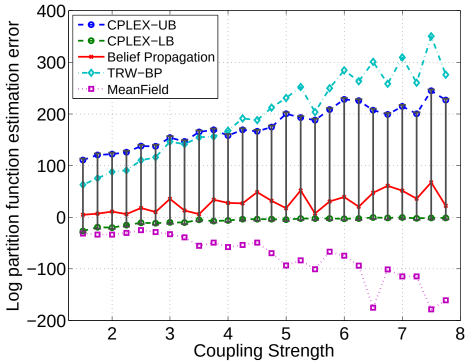
We compare with Loopy BP (Murphy et al., 1999) which estimates Z, Tree Reweighted BP (Wainwright, 2003) which gives a provable upper bound, and the Mean Field approach (Wainwright & Jordan, 2008)



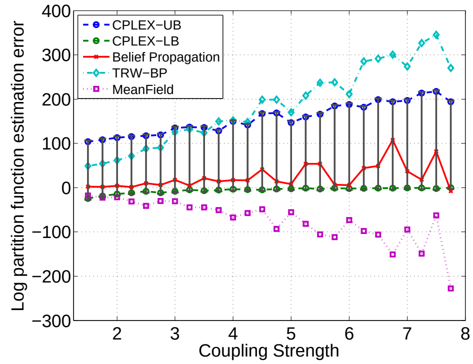
(a) Mixed 10×10. Field 0.1.



(b) Mixed 10×10. Field 1.0.



(c) Mixed 15×15. Field 0.1.



(d) Mixed 15×15. Field 1.0.

Figure 2. Results on spin glasses grids.

which gives a provable lower bound. We use the implementations in the LibDAI library (Mooij, 2010) and compare with ground truth obtained using the Junction Tree method (Lauritzen & Spiegelhalter, 1988).

Figure 2 shows the error in the resulting estimates, together with the upper and lower bounds obtained with WISH using CPLEX. We immediately see that our *lower bounds* are highly accurate (error close to 0), which means that the lower bounds provided by CPLEX for the ILPs must be close to optimality. Similarly good lower bounds can also be obtained using the original WISH algorithm (Ermon et al., 2013). However, the original WISH (without LP relaxations) does not provide *upper bound* guarantees, only the TRWBP approach does. Specifically, the original WISH algorithm with Toulbar (Allouche et al., 2010) provides an upper bound only upon proving optimality for all optimization instances in the inner loop. In contrast, the ILP formulation provides us with **anytime** and gradually improving upper bounds based on LP relaxations (cf. Figure 1(a)), often well before it can actually solve the problems to optimality (which might not be possible on larger instances) or, in principle, even before it can find a feasible solution. Figure 2 shows that our upper bounds are significantly tighter than the ones obtained using TRWBP in the hard weights region. Further, our ILP approach is guaranteed to eventually give an accurate answer, within a constant factor, given enough time. In contrast, message passing techniques are usually quite fast (if they converge) but do not provide better results with more runtime.

7. Conclusion

Leveraging a connection with max-likelihood decoding of binary codes, we showed that the MAP inference queries generated by WISH are in general not polynomial time solvable or even approximable. On the positive side, this led to the use of an ILP formulation for the problem, inspired by iterative message passing decoding. To increase the practicality of the ILP approach, we sparsified parity constraints while preserving their desirable properties. Finally, we showed that by solving a sequence of LP relaxations we can obtain not only very accurate lower bounds but also upper bounds that are much tighter than the ones provided by tree decomposition and convexity.

References

Allouche, D., de Givry, S., and Schiex, T. Toulbar2, an open source exact cost function network solver. Technical report, INRIA, 2010.

Arora, Sanjeev, Babai, László, Stern, Jacques, and Sweedyk, Z. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pp. 724–733. IEEE, 1993.

Berlekamp, E., McEliece, R., and Van Tilborg, H. On the inherent intractability of certain coding problems. *Information Theory, IEEE Transactions on*, 24(3):384–386, 1978.

Bertsimas, Dimitris and Tsitsiklis, John N. Introduction to linear optimization. 1997.

Ermon, Stefano, Gomes, Carla, Sabharwal, Ashish, and Selman, Bart. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *ICML (To appear)*, 2013.

Feldman, Jon, Wainwright, Martin J, and Karger, David R. Using linear programming to decode binary linear codes. *Information Theory, IEEE Transactions on*, 51(3):954–972, 2005.

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

Globerson, Amir and Jaakkola, Tommi. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. *Advances in Neural Information Processing Systems*, 21(1.6), 2007.

Gomes, C.P., Sabharwal, A., and Selman, B. Near-uniform sampling of combinatorial spaces using XOR constraints. *Advances In Neural Information Processing Systems*, 19: 481–488, 2006a.

Gomes, C.P., Sabharwal, A., and Selman, B. Model counting: A new strategy for obtaining good bounds. In *AAAI*, pp. 54–61, 2006b.

Jerrum, M. and Sinclair, A. The Markov chain Monte Carlo method: an approach to approximate counting and integration. *Approximation algorithms for NP-hard problems*, pp. 482–520, 1997.

Koller, Daphne and Friedman, Nir. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

Lauritzen, Steffen L and Spiegelhalter, David J. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 157–224, 1988.

Madras, N.N. *Lectures on Monte Carlo Methods.* American Mathematical Society, 2002. ISBN 0821829785.

Mooij, J.M. libDAI: A free and open source c++ library for discrete approximate inference in graphical models. *JMLR*, 11:2169–2173, 2010.

Murphy, K.P., Weiss, Y., and Jordan, M.I. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, 1999.

Sontag, David, Meltzer, Talya, Globerson, Amir, Jaakkola, Tommi, and Weiss, Yair. Tightening LP relaxations for MAP using message passing. In *UAI*, 2008.

Stern, Jacques. Approximating the number of error locations within a constant ratio is np-complete. In *Proceedings of the 10th International Symposium on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pp. 325–331. Springer-Verlag, 1993.

Vardy, Alexander. Algorithmic complexity in coding theory and the minimum distance problem. In *STOC*, 1997.

Wainwright, M.J. Tree-reweighted belief propagation algorithms and approximate ML estimation via pseudo-moment matching. In *AISTATS*, 2003.

Wainwright, M.J. and Jordan, M.I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

Wei, W. and Selman, B. A new approach to model counting. In *Theory and Applications of Satisfiability Testing (SAT)*, pp. 324–339, 2005.

Yannakakis, Mihalis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991.