

Generality of pairwise, binary, and planar factor graphs for probabilistic inference

Frederik H. Eaton
frederik@ofb.net

Zoubin Ghahramani
zoubin@eng.cam.ac.uk

September 14, 2011

Keywords: Approximate inference, Machine learning, Graphical models, Boolean satisfiability, Maximum a posteriori, Computational complexity

Abstract

Many probabilistic inference algorithms apply only to restricted classes of factor graphs. To clarify the generality of these algorithms, we investigate the expressive power of three such subclasses and their intersections: discrete graphs with binary variables, with pairwise factors, and with planar topology. We consider whether it is possible to reduce the problem of inferring marginal probabilities in general factor graphs to that in each of these classes. We show in particular that binary pairwise factor graphs are only “universal” for marginal inference in strictly positive models. We then conduct experiments to assess the performance of available approximate probabilistic inference algorithms on the models produced by our reductions.

1 Introduction

We are interested in the problem of calculating the marginal probabilities and conditioned marginal probabilities of variables in a statistical model. Although this problem is perhaps the most basic and common form of “statistical inference” or “probabilistic inference”, we shall refer to it here as “marginal inference” (MI) to avoid confusion with other tasks that are sometimes also called by these names. For the purpose of MI and other statistical

inference tasks, statistical models are often specified using a structure called a *factor graph*, as described in more detail in section 2.3. Factor graphs are a simple yet flexible structure for defining probabilistic models. They generalize richer procedural representations such as causal networks and acyclically directed mixed graphs, as well as physical models such as the Ising model. Various algorithms and theoretical results have been derived which apply only to restricted classes of factor graphs. Such restricted subclasses are defined by imposing constraints on aspects of the model variables or connectivity. For the purpose of this paper, we assume that variables have finite domain¹; we usually say that variables with this property are “discrete”.

We then consider the following three subclasses and their intersections: models having only *binary variables*; or only *pairwise factors*; or whose graphical structure is topologically *planar*. In this paper we present a handful of theorems which provide a foundation for describing the reduction properties of these classes of factor graphs with respect to the MI task: we say that MI on a particular class of factor graphs can be *reduced* to MI on another class if a solution to the second problem can be easily used to solve the first, in a sense which is made precise in section 3.1. Although the feasibility of solving problems such as maximum a posteriori (MAP) and Boolean satisfiability (SAT) on general inputs by reduction to analogous members of these three classes is fairly well understood, the corresponding results for marginal inference are apparently not widely known. We find that all three of the above classes and their intersections can reduce MI on general factor graphs, although sometimes only in a partial sense whose nature has not been previously described.

In particular, we show that binary pairwise factor graphs are not in general able to reduce models containing states with zero probability, an important fact which has not been recognized by other authors. We hope that our results will help to clarify the relative usefulness of existing and future work on this and other subclasses.

The following section provides some background for our later results. We open by reviewing the known facts on reductions in MAP and SAT in section 2.1. Then in section 2.2 we give an overview of some of the existing algorithms and decompositions for marginal inference which have been defined on restricted types of factor graphs. In section 2.3, we review the definition

¹Our notion of reduction does not allow infinite models to be reduced to finite ones, which we feel are the most interesting targets for reduction.

of factor graphs and marginal inference.

The main theoretical contributions appear in section 3. This opens with our definition of reduction for MI (section 3.1), which is motivated by the traditional notion of polynomial-time reducibility, and which forms the basis for the rest of the section. The results are presented in sections 3.2 to 3.4 and summarized in section 3.5.

Finally, in section 4, we exhibit data from numerical experiments in order to give a sense of the performance of existing marginal inference algorithms on models produced by our reductions.

2 Background

Readers who are unfamiliar with the usual probabilistic inference terminology may wish to read section 2.3 before continuing.

2.1 Reductions in SAT and MAP

We now review the existing work on reductions for the SAT and MAP problems. The Boolean satisfiability problem, or SAT, is the problem of determining whether a given Boolean formula can be satisfied by assigning certain values to each of the variables. If this decision problem can be solved, then by iteratively testing the satisfiability of a modified formula, in which additional clauses have been conjoined to force certain variables to be true or false, we can obtain a full satisfying assignment - so it is also reasonable to think of SAT as the problem of finding such an assignment. As a decision problem, SAT is in NP, the class of decision problems whose solutions have “correctness proofs” the size of which is bounded by a polynomial in the size of the input. A standard definition of reducibility exists for problems in NP: we say that a problem A is “polynomial-time reducible” (or simply “reducible”) to a problem B if, given an oracle (an idealized subroutine) which solves problem B in constant time, we can solve problem A in polynomial time. According to the Cook-Levin theorem, any problem in NP is reducible to SAT. SAT and other problems which share this property are said to be “NP-complete”. There is a very large body of research whose object is to identify and classify NP-complete problems.

We will sometimes use the term “universal” to describe problems which are able to reduce more general classes, by analogy to the universality of

Turing machines. In this sense, NP-complete problems like SAT are universal for NP. Such a term could also be applied to “NP-hard” problems, which are those problems that are able to reduce NP but are not themselves in NP, either because they are more difficult or because they are not decision problems.

A Boolean formula is typically defined by applying the connectives \wedge (“and”, or “conjunction”) and \vee (“or”, or “disjunction”) as well as the unary operator \neg (“not”, or “negation”) to an arbitrary set of variables in any order. Such formulae can be transformed into a number of “normal forms”, possibly by adding extra variables, which preserve the satisfying assignments of the original variables and thus can be used to solve the original problem. For example, any satisfying assignment of

$$a \vee b \vee c \vee d \tag{1}$$

can be extended to a satisfying assignment of

$$(a \vee b \vee x) \wedge (\neg x \vee c \vee d) \tag{2}$$

by choosing an appropriate value for x ; and any satisfying assignment of the second formula also satisfies the first. This idea can be generalized to show that a Boolean satisfiability problem may be reduced in polynomial time into the problem of solving one or more formulae in k -CNF (conjunctive normal form), which look like a conjunction of disjunctive clauses:

$$\bigwedge_c \left(\left(\bigvee_{i \in c^+} v_i \right) \vee \left(\bigvee_{i \in c^-} \neg v_i \right) \right) \tag{3}$$

where c ranges over a set of disjoint “clauses” $c \equiv c^+ \cup c^-$, each of which may contain a variable (in c^+) or its negation (in c^-). This is a standard result which holds when $k \geq 3$ (otherwise we cannot fit enough auxiliary variables in our clauses). The problem of finding a satisfying assignment for a formula in k -CNF is called k -SAT. In other words SAT is reducible to k -SAT (making the latter NP-complete) for $k \geq 3$. On the other hand, 2-SAT is in P, a fact which will be used later in Theorem 4.

There is a straightforward analogy between k -CNF formulae and factor graphs with binary variables and k -ary factors. By introducing a factor for each clause, and specifying that it take the value 1 when that clause is satisfied and 0 otherwise, we can get a distribution which partitions all of its

probability equally among each of the formula’s satisfying assignments. This distribution can be used to reduce SAT to the problem of marginal inference (MI), implying that MI is NP-hard, provided that some weak guarantees are made about the accuracy of the output marginals. The counterpart of binary-pairwise factor graphs under this analogy is formulae in the class 2-CNF which, as mentioned earlier, are not very expressive for representing general SAT instances. One result of this paper is that binary-pairwise factor graphs, while suffering in part from similar difficulties, are still able to represent general factor graphs in a “limit” (section 3.3).

Relaxing the “pairwise” ($k = 2$) condition, we can alternatively consider Boolean formulae which are in some sense planar. Planarity in this context is defined as the property of being able to embed in a plane the bipartite graph relating clauses and variables. It turns out that SAT can be reduced to “planar 3-SAT”, or in other words the latter is NP-complete (Lichtenstein, 1982).

Recall that we are considering three classes of restriction: binary variables, pairwise factors, and planar topology. For SAT, in which the variables are already “binary”, we find that only the “pairwise” class and its subclasses are not universal.

We now turn to the Maximum a Posteriori problem (MAP), an optimization problem whose goal is to find the state with maximum probability in a statistical model (which may be defined by a factor graph). MAP with binary variables is equivalent to weighted k -SAT, in which weights are assigned to each clause and the goal is to find a variable assignment where the sum of weights of satisfied clauses is maximized.

It is known that MAP may also be reduced to the maximum weight independent set problem (MWIS) (Sanghavi et al., 2009), in which weights are assigned to the vertices of a graph and the goal is to find a set of vertices of maximum weight subject to the constraint that no two vertices are connected by an edge. MWIS can in turn be straightforwardly reduced to MAP on a binary-pairwise graph (ibid.). Thus, binary pairwise graphs are universal in MAP. This is also known as the statement that optimization of pseudo-Boolean functions can be reduced to optimization of quadratic pseudo-Boolean functions (Boros and Hammer, 2002). We are not yet aware of the status of MAP on planar graphs, although the maximum independent set problem (without weights) on planar graphs is known to be NP-complete (Baker, 1994; Garey and Johnson, 1979).

We consider the task of marginal inference (MI) in the body of this paper.

Recall that we define MI as the problem of computing marginal probabilities in a model (defined by a factor graph), or more generally computing the partition function, or equivalently computing the probability of an assignment of values to some subset of the model’s variables. Some reductions have already been defined for MI. The Cluster Variational Method and Junction Tree algorithms are based on the construction of a new graph whose variables represent combinations of variables of an input graph, typically so that the problem of MI on a loopy graph may be reduced to that of MI on a tree. We do not consider this type of reduction, which is exponentially complex in general. The simple reduction to pairwise form, which we do consider, was previously defined by Yedidia et al. (2001). The details of this and the other MI reductions are presented in section 3.

It is instructive to contrast the three problems, SAT, MAP, and MI. One point of difference is in the role of auxiliary variables in reductions. In MI, whose notion of reduction we define in more detail in section 3.1, an auxiliary variable is introduced in such a way that the desired distribution is obtained from “marginalizing out” the new variable. Thus, the values of the distribution at each setting of the auxiliary variable must add up exactly to the correct quantity. In MAP, on the other hand, only one value of an auxiliary variable typically plays a role in the “solution” state, although the reduction must presumably be constructed so that the variable can switch values as appropriate to preserve the original optimization landscape. For example, given binary variables taking values in $\{0, 1\}$, we can use an auxiliary variable to turn a degree-three term into four terms of degree one or two:

$$\max_y (f(y) + y_1 y_2 y_3) = \max_{y,z} (f(y) - 2z + y_1 z + y_2 z + y_3 z) \quad (4)$$

In the new maximization problem, the auxiliary variable z is usually forced to take a single value, although if exactly two of the y_i ’s are 1 then z can take either value. In SAT, similar situations occur: many satisfying assignments in the input formula appear in the transformed formula with auxiliary variables forced to take only one value, but situations where both values are allowed may also arise. Consider the previous example (equation 2): assignments in which both $a \vee b$ and $c \vee d$ are true are “duplicated” in the new model, with one copy for each possible value of x .

A more abstract way of viewing the distinction between SAT, MAP, and MI is in terms of the methods available to us for verifying a solution to prob-

lems in each class. For SAT, it is simple to check that an assignment is indeed satisfying - and given such an assignment, we can immediately conclude that a formula is satisfiable (proving that a formula is not satisfiable can be more demanding). For MAP, given a state of the model there is no easy way in general to tell whether that state is an optimum. However, given two states, we can say which one is “better” by calculating and comparing the unnormalized joint probabilities at each state. Thus, in MAP there is a tractable total ordering of possible solutions in which the “true” solution is maximal. For MI, given two sets of univariate marginals, there is apparently no easy way to tell which one is better. But if we are given two approximate MI algorithms, each of which can be queried for conditioned marginals, then we may create a “score” as described in Eaton, F. (2011) which gives a rough indication of the better approximation. Although such a score is deterministic and is guaranteed to favor exact inference, the ordering it induces on approximations may contain cycles. This complication is absent from the simpler MAP setting. Thus we see that solution verifiability becomes progressively more difficult in the SAT, MAP, and MI problems, respectively.

2.2 Marginal inference on specialized graphs

A number of algorithms for marginal inference have been defined on special subclasses of factor graphs. The original Belief Propagation (BP) algorithm of Gallager (Gallager, 1963) was defined on binary factor graphs with parity-check factors. The BP formulation of Pearl (Pearl, 1982) originally used tree-structured causal (Bayesian) networks, and later loopy causal networks (Pearl, 1988, 1995). However, BP is easily generalized to arbitrary factor graphs (Kschischang et al., 2001).

The BP algorithm is sometimes specified on pairwise factor graphs, for pedagogical reasons (Yedidia et al., 2001) or for suitability to a parent algorithm (Wainwright et al., 2002). It is straightforward to convert general factor graphs to pairwise form (Theorem 2) and BP is actually invariant under such transformations.

Algorithms and results for binary graphs usually assume pairwise connectivity. One exception is the loop decomposition of Chertkov and Chernyak (2006) which is defined on binary n -wise factor graphs. This decomposition has been used to lower-bound the partition function of a binary pairwise fac-

tor graph (BPFPG) with “attractive”² potentials (Sudderth et al., 2008) and related theoretical results are often defined on BPFPGs (Watanabe and Fukumizu, 2011). MI algorithms for BPFPGs include Belief Optimization (Welling and Teh, 2001) and the self-avoiding-walk (SAW) tree expansion of Weitz (2006) (see also Jung and Shah, 2006). The algorithm of Montanari and Rizzo (2005) was defined on BPFPGs but is easily generalized (Mooij et al., 2007). It is not easy to see how one might adapt the interesting SAW-tree expansion to general factor graphs.

As for planar BPFPGs, we only know of one theoretical result for MI on this class, which makes the additional assumption of “pure” interaction potentials (assigning equal probability to a state and its complement): Globerson and Jaakkola’s algorithm for upper-bounding Z and calculating marginals (Globerson and Jaakkola, 2007), based on the Fisher-Kasteleyn-Temperley algorithm of statistical physics (Fisher, 1966; Kasteleyn, 1963).

2.3 Definitions

We understand a statistical model to be a probability distribution over some set of random variables, here taken to be discrete: $x \in \prod_{i=1\dots n} \mathcal{X}_i$, where the \mathcal{X}_i are finite sets. This distribution is often defined in terms of a *factor graph*, and we will assume such a representation in all of the material that follows. A factor graph is a collection \mathcal{F} of *factors*, each of which is associated with a set α of variables and a function (its “potential” or “local function”) from the domains of such variables to the non-negative real numbers:

$$\psi_\alpha : \mathcal{X}_\alpha \rightarrow \mathbb{R}_+ \tag{5}$$

where $\mathcal{X}_\alpha \equiv \prod_{i \in \alpha} \mathcal{X}_i$. These functions are multiplied together and normalized to induce a distribution over the variables:

$$P(x) = \frac{1}{Z} \prod_{\alpha \in \mathcal{F}} \psi_\alpha(x_\alpha) \tag{6}$$

A less flexible but essentially identical representation, called a *Markov random field* (MRF), is sometimes used to specify a distribution. In an MRF the product consists of potential functions whose domains correspond to *cliques*

²Sudderth and Wainwright define a binary pairwise model to be “attractive” if for every edge potential, the relation $\psi_{ij}(0,0)\psi_{ij}(1,1) \geq \psi_{ij}(0,1)\psi_{ij}(1,0)$ holds.

of a graph rather than arbitrary sets of variables as above. In this paper we use the factor graph representation and terminology.

The problem of *marginal inference* (MI) (also called probabilistic inference or Bayesian statistical inference) is to calculate marginals

$$P(x_i) \equiv \sum_{x_{\setminus i}} P(x) \quad (7)$$

Here $x_{\setminus i}$ represents the set of all x variables excluding x_i , i.e. $x_{1\dots i-1, i+1\dots n}$. When such calculation is only approximate, then we say that “approximate marginal inference” (or when there is no room for confusion, “approximate inference”) is being done. When the calculation is exact (to machine precision) or the resulting approximation is required to be accurate to within some bound, then for general factor graphs MI is known to be NP-hard (Cooper, 1990; Barahona, 1982). In this paper, we are not too concerned with the accuracy guarantees, if any, of MI algorithms which might be applied to a representation or transformation of a factor graph, since in each case the representation itself is either exact or can be made arbitrarily precise. In other words, any errors will be faithfully propagated through the transformation, rather than being introduced or magnified by it. Our results will apply to both exact and approximate forms of MI, and we shall refer to both variants as simply “MI”.

By introducing a factor which is a delta function, we can constrain a variable to take a given value. The resulting distribution is equal to a conditioned version of the original distribution:

$$P(x|x_i = x_i^*) = \frac{1}{Z'} \delta(x_i, x_i^*) \prod_{\alpha} \psi_{\alpha}(x_{\alpha}) \quad (8)$$

MI in the conditioned model gives conditioned marginals, and these can be combined with unconditioned marginals to compute the probability of arbitrarily many variables:

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \quad (9)$$

In fact, many inference algorithms, for example BP, produce estimates of the partition function, allowing such “multi-variable marginal” probabilities to be computed in one step; if r is a set of variables then we have

$$P(x_r) = \frac{Z'}{Z} \equiv \frac{\sum_{x'} \prod_{i \in r} \delta(x'_i, x_i) \prod_{\alpha} \psi_{\alpha}(x'_{\alpha})}{\sum_x \prod_{\alpha} \psi_{\alpha}(x_{\alpha})} \quad (10)$$

We shall call x_r a *partial assignment* (PA) and shall consider the problem of “weighing” or calculating the probability mass of multi-variable PAs as equivalent to MI. When more notational precision is needed, we shall write the variables and the assigned values of the PA separately, e.g. $x_r = x_r^*$, as in $P(x_r = x_r^*)$. Where we have a superset $r' \supseteq r$ and $(x_{r'})_r = x_r^*$ then we say the PA $\{x_{r'} = x_{r'}^*\}$ is an extension of the PA $\{x_r = x_r^*\}$. Since PAs are a special kind of “event” in a σ -algebra, we also use terminology from this domain, and speak accordingly of the “union” or “intersection” of PAs. One may check that PAs are closed under intersection but not union.

Note that we can easily apply MAP to a conditioned model using the constraint technique above. But applying MAP to a model in which some variables have been summed over or “marginalized out” is not straightforward. Given an algorithm which computes the most probable assignment of all the variables in the model, there is no general way to adapt it to compute the most probable assignment of only some of the variables, summing over the others.

The terms “marginal inference” and “partial assignment” are not common. Our emphasis on the broader view of MI as the task of calculating, in addition to marginals, the probabilities of PAs and conditionals is also somewhat new. The rest of our terminology and notation is fairly standard. Factor graphs were defined recently in Kschischang et al. (2001). Potential functions (for an MRF) are called ψ in Castillo et al. (1997). Wiegnerinck (2000) introduced the convention of indexing clusters of variables with Greek letters α, β, γ .

3 Theoretical results

3.1 Definition of reduction

We define what we mean when we say that marginal inference (MI) on members of one class of factor graphs is “reducible” to MI on members of another class. We will also use words like “conversion” or “representation” to denote the same property. Our notion of reducibility is modeled on the concept of “polynomial-time reducibility” from the theory of computational complexity. It is a rather natural definition, which is satisfied by existing reductions. It is based on a kind of Galois connection between event algebras (property 2 in the definition below). It may be possible to generalize this definition or

to say more about its formal structure, but the present form suffices for the arguments in this paper.

We would like to be able to motivate our definition more rigorously by showing that reductions of MI between two models must satisfy our definition provided they are: of polynomial time complexity; general, in the sense of not depending on a particular choice of factors; and furthermore “minimal”, meaning that unnecessary computation is avoided. But we cannot do so at present. Instead we make some high-level arguments to show why some simple variations on our definition are not suitable. We also give an example to illustrate the need for the “minimality” clause above. These arguments have been removed to Appendix 4, because of their still informal status. We give the end result, the definition of reduction, as follows.

Definition 1. *We say that marginal inference (MI) in a model $P(x)$ is “simply reducible”, or just “reducible”, to MI in a model $Q(y)$ if there exists a function F from PAs in P to PAs in Q such that $F(x_r) = y_s \implies$*

1. *Conservation of probability mass: $P(x_r) = Q(y_s)$*
2. *Preservation of containment: Given $r' \supset r$ and $x_{r'}$ such that $(x_{r'})_r = x_r$, we have $F(x_{r'}) = y_{s'}$ where $s' \supset s$ and $(y_{s'})_s = y_s$*

As a special case, the notion of representation of P by Q includes situations where the variables of P are “included” in Q , but when Q also has extra “latent” variables which need to be “marginalized out”. In this scenario we can see F as a kind of embedding. An example is the pairwise reduction of Theorem 2.

We describe some consequences of this definition. First, the condition that F preserve containment, motivated in the appendix with an appeal to time complexity, implies that we can write the value of F at a multi-variable PA in terms of its values at single variables: $F(x_r) = \bigcap_{i \in r} F(x_i)$.³ Thus, it is enough to define F on all (variable, value) pairs.

Note that the PA-map F cannot be multi-valued. Reductions are forbidden in which a probability $P(x_r)$ is calculated from the union of multiple PAs in Q , i.e. $P(x_r) = Q(\bigcup_i y_{s_i}) = \sum_i Q(y_{s_i})$. The reason for this is again to preserve polynomial time-complexity of the reduction. Due to the containment-preserving nature of F , if we apply F to the intersection of n

³In the appendix, the containment-preserving condition is motivated using an appeal to time-complexity.

PAs in P , each of which maps to a union of two PAs in Q , then the result will be a union of 2^n PAs in Q . So computing this mass will take exponential time.

Second, we note it may happen that F is not invertible: it is not the case, under our definition, that P is MI-reducible to $Q \Leftrightarrow Q$ is MI-reducible to P . For an example, suppose $P(x_i) = Q(F(x_i))$ where F encodes a four-valued x_1 in binary:

$$F(x_1 = 0) = \{y_{(1,2)} = (0, 0)\} \quad (11)$$

$$F(x_1 = 1) = \{y_{(1,2)} = (0, 1)\} \quad (12)$$

$$F(x_1 = 2) = \{y_{(1,2)} = (1, 0)\} \quad (13)$$

$$F(x_1 = 3) = \{y_{(1,2)} = (1, 1)\} \quad (14)$$

Then $Q(y_2 = 0) = P(x_1 = 0) + P(x_1 = 2)$, which is a relationship we cannot express through a simple reduction since our PA-maps must, as above, be single-valued.

It is easy to see that our notion of reduction is transitive: if P reduces to Q with PA-map F , and Q to R with PA-map G , then P reduces to R with PA-map $G \circ F$.

3.2 Pairwise factor graphs

A common restriction imposed on factor graphs is to require all factors to have size one or two. (Note that size one, or singleton, factors can be seen as degenerate factors of size 2.) Such graphs are called as “pairwise” factor graphs. It is easy to show that arbitrary (n -wise) factor graphs can be converted into pairwise form. A version of the following theorem was outlined in Yedidia et al. (2001).

Theorem 2. *Any factor graph can be converted to pairwise form*

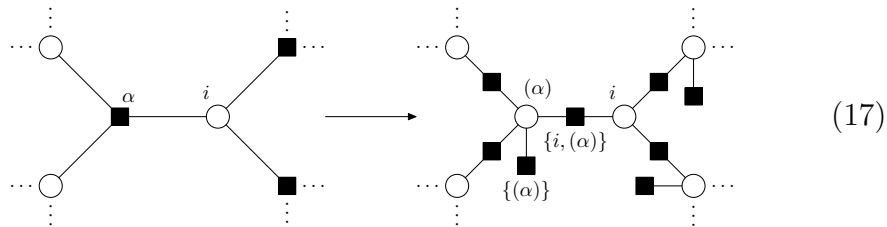
Proof. One way to effect this conversion is to create a variable (i or (α)) for each variable (i) and factor (α) in the old graph, introduce singleton factors $\{(\alpha)\}$ for each α and pairwise factors $\{i, (\alpha)\}$ for each $i \sim \alpha$, and assign to these factors the following potentials:

$$\hat{\psi}_{\{i, (\alpha)\}}(\hat{x}_i, \hat{x}_{(\alpha)}) = \begin{cases} 1 & \text{if } \hat{x}_i = [\hat{x}_{(\alpha)}]_i \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

$$\hat{\psi}_{\{(\alpha)\}}(\hat{x}_{(\alpha)}) = \psi_{\alpha}([\hat{x}_{(\alpha)}]) \quad (16)$$

where the domains are $\hat{\mathcal{X}}_i = \mathcal{X}_i$ and $\hat{\mathcal{X}}_{(\alpha)} = (\mathcal{X}_\alpha)$. Here $[\]$ is used as a kind of inverse of $(\)$, so $x_\alpha = [\hat{x}_{(\alpha)}]$ indicates the set of variable assignments x_α (in the old graph) corresponding to the single variable assignment $\hat{x}_{(\alpha)} = (x_\alpha)$ (in the new graph).

The new pairwise potentials are constructed to enforce consistency between the representatives of the old variables and copies of them appearing in representatives of the old factors by assigning zero weight to illegal states, and the new singleton potentials incorporate the values of the old factors, with the result that the legal states have the same weight as in the original graph. The transformation is illustrated in the following diagram:



□

It is straightforward to check that, for the above construction, Belief Propagation (BP) on the converted pairwise graph is equivalent to BP on the original graph. Interestingly Mean Field (MF) does not carry over.

3.3 Binary pairwise factor graphs

The next restriction we consider is to require all variable domains to have size two. Factor graphs satisfying this restriction are called “binary”. This restriction is usually combined with the first, resulting in “binary pairwise” factor graphs (BPFs). Several algorithms and decompositions have been proposed which only apply to BPFs, so it is interesting to ask if it is possible to convert more general factor graphs to the binary pairwise form. Such a reduction might be imagined as first converting the input to binary form, by choosing an encoding of the input variables, and then adding latent variables to implement the correct distribution over the new graph. We show that for general input graphs - in particular, for those which may contain states having zero probability - such a reduction does not exist, at least according to our definition.

Our proof depends on a fact about k -SAT. Recall that k -SAT is the problem of finding satisfying assignments to Boolean formulae written in the format of equation 3, namely as a conjunction of disjunctive clauses. For such a formula to be satisfied, every clause must be true, which means that at least one of its positive variables must be true, or at least one of its negative variables must be false. For $k \geq 3$, k -SAT is NP-complete and we can create a k -SAT instance where a given set of assignments to some variables, and no other assignments, satisfies the formula (possibly by introducing extra auxiliary variables). This is not possible with 2-SAT, however, whose satisfying assignments form a structure called a “median graph” and can easily be shown to have the following property (Knuth, 2008, 64-74):

Lemma 3. (*Median property*) *Given a set of three satisfying assignments to a 2-SAT formula, if we construct a new assignment (the “median” of the three) in which each of the variables take the values they took in the majority of the other assignments, then the new assignment is also satisfying.*

Our theorem follows directly from the observation that the positive states of a binary pairwise factor graph correspond to solutions of 2-SAT.

Theorem 4. *There exist factor graphs which cannot be converted to binary pairwise form*

Proof. We will assume that the input graph is binary. Call this graph P and let it as usual be given by

$$P(x) = \frac{1}{Z} \prod_{\alpha} \psi_{\alpha}(x_{\alpha}) \quad (18)$$

We see that a state x^* has positive probability if and only if the following Boolean expression is true:

$$\bigwedge_{\alpha \in \mathcal{F}} \bigwedge_{\substack{x_{\alpha} \in \mathcal{X}_{\alpha} \\ \psi_{\alpha}(x_{\alpha})=0}} \bigvee_{i \in \alpha} x_i \neq x_i^* \quad (19)$$

Introduce a Boolean variable v_i which is true if $x_i^* = 1$ and false otherwise; the expression becomes:

$$\bigwedge_{\alpha \in \mathcal{F}} \bigwedge_{\substack{x_{\alpha} \in \mathcal{X}_{\alpha} \\ \psi_{\alpha}(x_{\alpha})=0}} \left(\left(\bigvee_{\substack{i \in \alpha \\ x_i=0}} v_i \right) \vee \left(\bigvee_{\substack{i \in \alpha \\ x_i=1}} \neg v_i \right) \right) \quad (20)$$

The positive states of P are thus exactly the solutions of a k -SAT instance, where k is the number of variables in the largest factor in P . Any set of states can be realized as a solution set of k -SAT when $k \geq 3$, but when $k = 2$, such sets must obey the median rule defined above. If we can show that our definition of reduction preserves lack of median structure, then we are done: an arbitrary model P (without median structure) cannot then be reduced to a binary pairwise model Q (with median structure).

Let F be the PA-map of a representation of $P(x)$ by $Q(y)$, where Q has median structure. Consider a triple of states $x^{(1)}, x^{(2)}, x^{(3)}$ in P (i.e. these are full, not partial, assignments), each with positive probability, and let x^* be their median. These map under F to a triple of PAs $y_{r_1}^{(1)}, y_{r_2}^{(2)}, y_{r_3}^{(3)}$ in Q . Since each PA $y_{r_i}^{(i)}$ has positive probability $Q(y_{r_i}^{(i)}) = P(x^{(i)})$, it can be extended to a full state $y^{(i)}$ with positive probability. The median of these three states let us call y^* . Since we assumed the median property for Q , we have $Q(y^*) > 0$. Now we would like to show that y^* is an extension of $F(x^*)$. This follows from the variable intersection rule for PA maps: $F(x) = \bigcap_i F(x_i)$. More specifically, let i be a variable in P . Since x_i^* is a median of $(x_i^{(1)}, x_i^{(2)}, x_i^{(3)})$, it must have the same value of two of these - say, without loss of generality, $x_i^{(1)}$ and $x_i^{(2)}$. But $y_{r_1}^{(1)}$ and $y_{r_2}^{(2)}$ will then both be consistent with $F(x_i^*) = F(x_i^{(1)}) = F(x_i^{(2)})$. As a consequence, y^* will share this consistency: any variable which is fixed in $F(x_i^*)$ will appear in both $y^{(1)}$ and $y^{(2)}$ and hence y^* . Since we have shown y^* is consistent with $F(x_i^*)$ for all i , it follows that y^* must be an extension of $F(x^*)$.

Now, $Q(y^*) > 0$ since we assumed Q to have median structure. But $y^* \in F(x^*)$ so $P(x^*) = Q(F(x^*)) \geq Q(y^*) > 0$. Thus x^* has positive probability in P . Hence, P has median structure.

We have proven that our reductions preserve lack of median structure, from which it follows that MI in a model whose positive states lack median structure cannot be reduced to MI in a binary pairwise factor graph. We have indicated that general factor graphs do not have median structure, but it may help to give a concrete counterexample. The following distribution, which we call the ‘‘XOR distribution’’, lacks the median structure and so is not representable by a binary pairwise graph:

$$P(s_1, s_2, s_3) = \begin{cases} \frac{1}{4} & \prod_i s_i = -1 \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

where $s_i \in \pm 1$. The median structure demands that “111” has a positive probability, since the following three positive configurations each have a majority value of 1 for each variable:

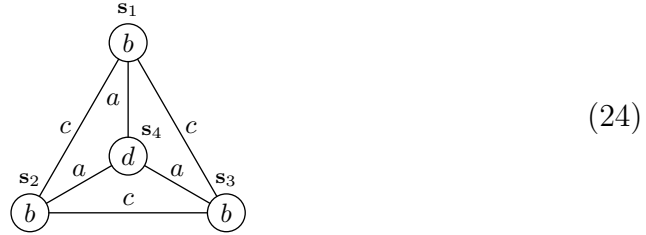
$$\begin{array}{ccc|c}
 & s_1 & s_2 & s_3 \\
 \hline
 & -1 & 1 & 1 \\
 & 1 & -1 & 1 \\
 & 1 & 1 & -1 \\
 \hline
 \text{median:} & 1 & 1 & 1
 \end{array} \tag{22}$$

But the distribution assigns it a zero probability. □

We saw that the XOR distribution of equation 21 cannot be represented by a binary pairwise factor graph. It is however possible to construct a sequence of binary pairwise graphs which approaches the XOR distribution with arbitrary precision. This is because it is possible to implement the following distribution as a binary-pairwise factor graph, for finite k :

$$P(s_1, s_2, s_3) = \exp \left(k \prod_{i=1}^3 s_i \right) \tag{23}$$

The following explicit construction is due to Martijn Leisink (Leisink, 2010). Introduce an auxiliary variable s_4 , and create a network:



with weights shown (corresponding to factors $\exp(as_1s_4)$, $\exp(bs_1)$, etc.), having values:

$$b = \frac{k}{4|k|} \operatorname{acosh}(e^{4|k|}) \tag{25}$$

$$c = -|b| \tag{26}$$

$$a = \frac{-k}{4|k|} \operatorname{acosh}(e^{8|b|}) \tag{27}$$

$$d = |a| \tag{28}$$

This set of weights is not unique, since although there are four unknown weights and four unique (up to permutation) values for the state $s_{1:3}$, the partition function of the new model is an extra degree of freedom which can be constrained by the simplifying choice, $d = |a|$, from which follows $c = -|b|$ and the other two equations.

It is straightforward to check that the network induces the distribution P of equation 23 on $s_{1:3}$ when marginalizing out s_4 . This allows us to prove the following theorem:

Theorem 5. *Every factor graph can be represented arbitrarily closely by a binary pairwise graph*

More precisely, the size of the output binary pairwise graph is a fixed function of the input graph, and only the parameters must change in order obtain a closer approximation to the input graph. This relates to a kind of “reduction in a limit”. We propose the terminology “almost universal”, as opposed to “universal”, to describe classes such as BCFGs (and planar BCFGs, as we shall see) which are able to reduce general problems in a limiting sense.

It has been known for a long time that inference in BCFGs is NP-hard (see for instance Barahona (1982)). It follows that these models are expressive enough to represent NP-complete problems such as SAT, even though working out the details of such a representation might be cumbersome. However, it is not necessarily clear from this result how to represent other NP-hard problems, such as inference in general factor graphs, using the binary pairwise form. As we saw in the previous theorem, it is not the case that there is a simple correspondence. We also remark that the problem of counting solutions to 2-SAT instances, called #2-SAT, is known to be #P-complete (Valiant, 1979), implying that it is NP-hard. This problem is analogous to that of computing the partition function of a BCFG, and we have already seen (equation 10 section 2.3) how computing the partition function can be used for MI. Thus, there is some existing theoretical support, though only of an indirect nature, for a kind of universality in BCFGs.

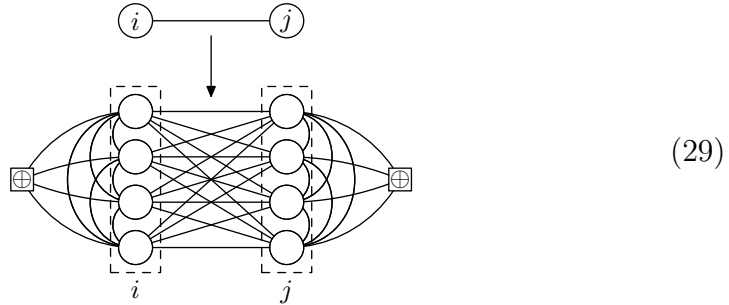
We proceed to prove Theorem 5.

Proof. Assume, without loss of generality, that the original graph is in pairwise form. Now create a new graph with a binary variable $k = (i, x_i)$ for each of the (variable, value) pairs in the old graph, which will be by construction $y_k = 1$ if the variable i takes value x_i in the old graph, and $y_k = 0$ otherwise.

Introduce an edge $(k, l) = ((i, x_i), (j, x_j))$ for each edge (i, j) in the old graph and each pair of values (x_i, x_j) , with factor potentials $\psi_{kl}(y_k, y_l)$ equal to 1 if either $y_k = 0$ or $y_l = 0$ and equal to $\psi_{ij}(x_i, x_j)$ otherwise. One can see that this graph has an unnormalized joint which coincides with that of the original graph for each “allowed” state. We still need to exclude states where a variable i takes “multiple values”, i.e. states y for which $y_{(i, x_i)} = y_{(i, x'_i)} = 1$ for some $x_i \neq x'_i$; and we need to ensure that at least one $y_{(i, x_i)}$ is 1 for each i .

We try to create a “1-of- n ” gadget as follows. For each variable i , introduce an edge $((i, x_i), (i, x'_i))$ for each pair of values $x_i \neq x'_i$ with factor potential equal to zero if both $y_{(i, x_i)}$ and $y_{(i, x'_i)}$ are 1, and equal to 1 otherwise. This ensures that no more than one $y_{(i, x_i)}$ is 1 for each i , but the remaining case where $y_{(i, x_i)} = 0$ for all x_i is not yet excluded by the new graph. In fact, it is impossible to exclude it using only binary pairwise factors when $n \geq 3$, since it is a median of the other valid states. We can however exclude it by introducing new a XOR factor of size $|\mathcal{X}_i|$ which ensures that an odd (and therefore non-zero) number of the $y_{(i, x_i)}$ are equal to 1.

The following diagram describes the transformation for the case $|\mathcal{X}_i| = |\mathcal{X}_j| = 4$ (the two XOR factors are marked \oplus):



It is easy to see that an XOR factor of size n can be constructed by combining $n - 2$ XOR factors of size 3 (and with a single edge when $n = 2$). Since XOR factors of size 3 can be achieved as a limit of binary pairwise graphs (by letting $k \rightarrow \pm\infty$ in equation 23) this completes the proof. \square

This also shows

Corollary 6. *Any discrete factor graph can be converted to binary 3-wise form*

Since the 3-wise to pairwise transformation of Leisink only breaks down in the presence of potential functions with zero entries, we ask whether it is possible to perform the binary pairwise conversion in a way that avoids resorting to a limit when graph potentials are strictly positive.

The answer is “yes”. We prove this result in two parts. First, we show how to convert a general factor graph with positive entries to positive-entry binary n -wise form. Then we show how to convert a binary n -wise graph with positive entries into binary-pairwise form.

Theorem 7. *Any discrete factor graph can be converted to binary form, in such a way that if the original graph had strictly positive potentials then the output graph also has strictly positive potentials*

Proof. This construction is simpler than it may appear. The idea is to choose a binary encoding for each of the values of each variable. These encodings may have to have different lengths. Extra scaling factors are introduced to compensate for states which appear multiple times in the new graph, as a consequence of the variable length of the encoding. The details follow:

Choose a minimal binary “prefix-free”⁴ encoding for the values in each variable’s domain \mathcal{X}_i . The encoded values may contain different numbers of bits, since $|\mathcal{X}_i|$ may not be a power of 2. The encoding will correspond to a binary tree with $|\mathcal{X}_i|$ leaves. In the new graph, for each variable i introduce k_i binary variables, where k_i is the maximum depth of the tree, and call this set β_i . For each factor α in the original graph, create a factor in the new graph containing variables $\bigcup_{i \in \alpha} \beta_i$, whose entry at a given (binary) assignment y_{β_i} corresponds to the entry of $\psi_\alpha(x_\alpha)$ in the original graph, where $(x_\alpha)_i$ is the unique decoding of the y_{β_i} for each i . Lastly, we need to compensate for the fact that a single variable assignment x_i in the original graph may correspond to multiple assignments y_{β_i} in the new graph, due to the presence of extra unused variables when a particular x_i is encoded with fewer than k_i bits. To this end, attach a factor to y_{β_i} with entries equal to $2^{l_i(y_{\beta_i}) - k_i}$, where $l_i(y_{\beta_i})$ is the length of the encoding of the value x_i corresponding to y_{β_i} . This ensures that summing over the unused variables in each encoding gives the correct probability of an assignment in the original graph. \square

⁴A prefix-free encoding is a variable-length encoding such that no codeword forms a prefix for a longer codeword.

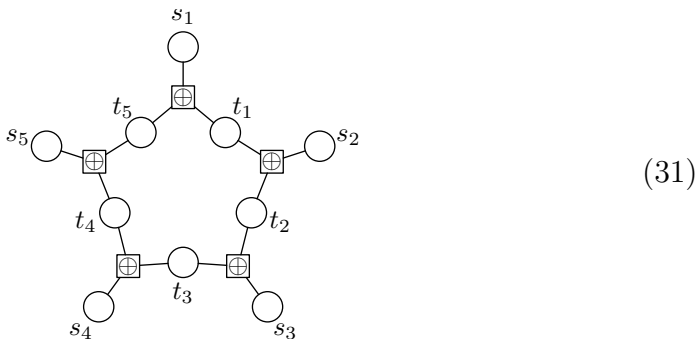
Proposition 8. *The n -wise soft-XOR factor*

$$P(s) \propto \exp\left(k \prod_{i=1}^n s_i\right) \quad (30)$$

(with k finite) can be represented in binary pairwise form.

Proof. A factor with $k \leq 0$ can be represented by a $k \geq 0$ factor by flipping the sign of one of the variables, so assume $k \geq 0$.

Consider connecting the n binary variables $s_{1:n}$ with 3-wise soft-XOR factors, each of strength k' , and n auxiliary variables $t_{1:n}$ in a loop as shown for $n = 5$:



We will prove that for any k we can always find a k' such that the above graph implements the distribution of equation 30. The probability of a configuration of the s variables is

$$P(s) \propto \sum_t \exp(k'(t_n s_1 t_1 + t_1 s_2 t_2 + \dots + t_{n-1} s_n t_n)) \quad (32)$$

This summation has 2^n terms. Observe that when two states s and s' have the same parity, then the terms in the summation over t for $P(s)$ are a permutation of those in the summation over t for $P(s')$. To prove this, consider inverting a neighboring pair of s variables, say s_i and s_{i+1} . The effect is the same as inverting t_i , which exchanges pairs of terms in the sum, leaving the total value invariant. But a sequence of such inversions can be used to go between any s and s' if they have the same parity. In particular, inverting all t_i for which $\prod_{j=1}^i s_j = -1$ rearranges the terms to correspond to

$s = (1, 1, \dots, 1)$ (if $\prod_{j=1}^n s_j = 1$) or to $s = (-1, 1, \dots, 1)$ (if $\prod_{j=1}^n s_j = -1$). This shows that $P(s)$ takes only one of two values:

$$P(s) = \begin{cases} p_1 & : \prod_{j=1}^n s_j = 1 \\ p_2 & : \prod_{j=1}^n s_j = -1 \end{cases} \quad (33)$$

for some p_1 and p_2 , which is the same as saying

$$P(s) \propto \exp\left(k \prod_i s_i\right) \quad (34)$$

where $k = \frac{1}{2} \log \frac{p_1}{p_2}$.

It remains to verify that the resulting relationship $k' \mapsto k$ can be inverted. At least for small n this relationship appears to be strictly monotonic, but it is not necessary to prove that fact in general. All that is needed is to observe that $k' = 0$ gives $p_1 = p_2 \implies k = 0$, and $k' \rightarrow \infty \implies k \rightarrow \infty$. The second implication follows from observing the values of the network when the 3-wise soft-XORs become “hard”-XORs. Finally, continuity and the intermediate value theorem imply that for any positive k , we can find a k' such that the above graph (31) is equivalent to a n -wise soft-XOR of strength k . \square

Corollary 9. *Any n -wise binary factor with strictly positive entries can be implemented in binary pairwise form*

Proof. The 2^n functions $s \mapsto s_1^{e_1} s_2^{e_2} \dots s_n^{e_n}$ parametrized by a vector $e \in \{0, 1\}^n$ form an independent basis for the space of real-valued functions of s , so we can write the factor’s potential function as $\exp(\sum_e a_e s_1^{e_1} s_2^{e_2} \dots s_n^{e_n})$ for some set of coefficients a_e . But such a potential can be implemented by superimposing 2^n soft-XOR factors of strength a_e , each covering subsets of the variables selected by the vector e . \square

Together with Theorem 7, this proves:

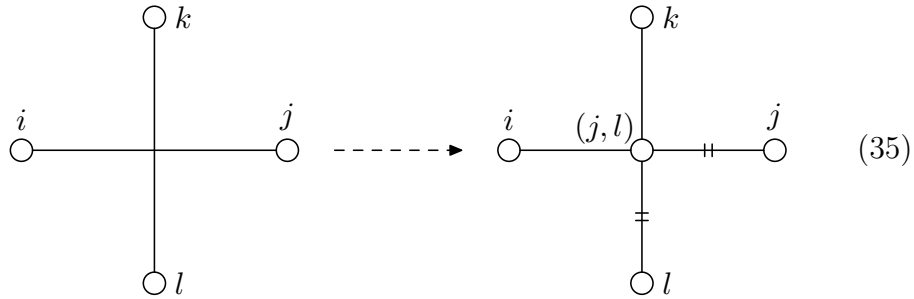
Theorem 10. *Any factor graph with strictly positive factors can be represented in binary pairwise form*

3.4 Planar binary pairwise graphs

Finally, we address the problem of converting an arbitrary factor graph to planar form. Planar graphs are defined as graphs which can be drawn in a

plane (\mathbb{R}^2) without any crossing edges. This condition is equivalent to forbidding K_5 and $K_{3,3}$ graph minors (see “Wagner’s theorem” (Wagner, 1937), related to “Kuratowski’s theorem” (Kuratowski, 1930)). Planar graphs have a number of special properties. For instance, a closed non-self-intersecting path splits a planar graph into two components, in analogy to the Jordan curve theorem. Also, a planar graph has a naturally-defined “dual” which is also planar. The “planar separator theorem” (Ungar, 1951) may be used to engineer efficient divide-and-conquer algorithms for planar graphs.

It seems useful to consider the possibility of reducing MI on general graphs to MI on planar graphs, partly because of the existence of a handful of results which apply to MI on planar graphs, and also because one can anticipate that more of these results may be derived in the future. If we allow planar graphs to have variables with arbitrarily large domain, then the reduction task is straightforward: we simply draw the graph in two dimensions, and introduce a new variable wherever two edges cross. The new variable encodes the values at an endpoint of each of the two original edges.



Here, the factor between the new (j, l) variable and j enforces consistency between x_j and $x_{(j,l)}$; similarly for the factor between (j, l) and l (in both cases this is indicated with a double tic). The factors between i and (j, l) and between k and (j, l) copy the entries of ψ_{ij} and ψ_{kl} , respectively, in a sense which is similar to that of the pairwise conversion theorem (Theorem 2).

Finding a conversion for the *binary* pairwise planar case is more difficult since only two values can be used to propagate data across an intersection. Inference in binary pairwise planar graphs was shown to be NP-hard by Barahona (1982) (although in the special case of Ising “spin glasses”, with soft-XOR pairwise factors and no unary factors, it is tractable (ibid.)) which

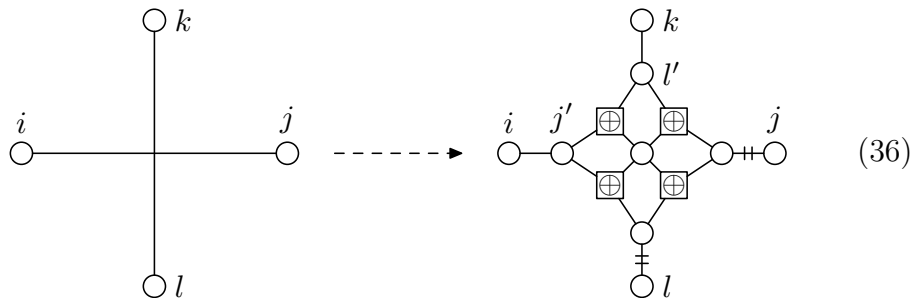
suggests that there could be a way to convert from ordinary factor graphs to binary pairwise planar factor graphs. Such a conversion would be of interest because of the existence of a number of results which apply only to the planar binary pairwise case, in approximate inference (Globerson and Jaakkola, 2007; Chertkov et al., 2009) and statistical physics (Fisher, 1966; Kasteleyn, 1963). However, note that these results all specify the additional constraint of pure interactions, demanding that a model assign equal probability to a state and its complement. The class of models with this property is quite restrictive, and is not even closed under variable conditioning.

We are not aware of a way to turn arbitrary factor graphs into binary pairwise planar graphs exactly, but it is not difficult to effect such a reduction in a limit.

Theorem 11. *Any discrete factor graph can be represented arbitrarily closely by a planar binary pairwise factor graph.*

In other words, planar BPFs are “almost universal”, in the terminology proposed after the statement of Theorem 5. As in that theorem, the structure of the output graph is fixed and only the parameters must vary to achieve an arbitrarily accurate representation.

Proof. Convert the graph to binary pairwise form as described above, and replace each pair of overlapping edges with the following subgraph, using soft-XOR 3-wise nodes of strength m .



As previously, edges with a double tic enforce the constraint that their endpoint variables match (i.e. in this case they have potentials $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$). The factor connecting i and j' in the new graph should be the same as ψ_{ij} in the old graph, and similarly for k and l' .

In the limit as $m \rightarrow \infty$, the soft-XOR factors become XOR factors; then, note that $x_{i'}$ is forced to take the value of x_i , and $x_{j'}$ to take the value of x_j . The auxiliary variable in the center simply reflects whether or not $x_i = x_j$. \square

3.5 Summary

We have demonstrated a number of formal conversions between different types of factor graphs, which prove that inference in one class of graphs can be implemented using inference in a more restrictive class. To the best of our knowledge, of the theorems appearing in this subsection only Theorem 2 has been published before.

We summarize the results. General discrete factor graphs can be converted to pairwise form and to binary form and in particular to binary 3-wise form. They can be converted to binary pairwise form if they have positive entries (and some discrete factor graphs with zeroes, such as the binary 3-wise XOR, cannot be implemented in binary pairwise form). If they have zero entries, they can still be represented arbitrarily closely by a binary pairwise graph. Additionally, general discrete factor graphs can be represented arbitrarily closely by a planar binary pairwise graph. In short, all of the three classes (pairwise, binary, planar) by themselves are universal; binary pairwise factor graphs (BPFs) and planar BPFs are what we have called “almost universal”. BPFs are universal for models with strictly positive potential functions, but not for general models. The question of whether planar BPFs are also universal for positive models is an open problem.

It is interesting to ask whether we can quantify the extent to which transformations such as the binary pairwise transformation may make inference more difficult. We are not yet able to provide any theoretical insight into this question, but we present the results of some empirical investigations in the next section.

4 Experiments

Since the purpose of our factor graph reductions is to assist with marginal inference, we are curious to know to what extent inference is made more difficult as a result of the extra complexity introduced by these reductions. We have already remarked that BP commutes with the pairwise transformation.

This is certainly not true for other transformations, which may distribute information across multiple variables, or introduce “frustrated” factors (which is to say, factors whose effects tend to almost cancel each other out, as in the case of Leisink’s 3-wise-to-pairwise transformation) or factors with very large or very small entries.

As in the case of k -SAT vs. 2-SAT, it may be that inference is qualitatively more tractable in binary pairwise graphs than in more general graphs. This would partly explain the number of algorithms that only apply to binary pairwise graphs. Quantifying the extent to which the binary pairwise transformation amplifies “frustration” or some other (not yet formalized) measure of inference difficulty would help set bounds on the extent to which binary-pairwise-specific algorithms can be more powerful than general algorithms. We seem a long way from verifying theoretically whether marginal inference in BPFs is in any sense more tractable than that in general factor graphs, but we can try to use empirical techniques to understand the situation at a more practical level: if inference in BPFs is more tractable than that in general factor graphs, we would expect all possible BPF reductions to result in graphs for which standard inference algorithms (which don’t take advantage of the binary-pairwise nature of the graph) should find inference relatively difficult. If on the other hand transforming general factor graphs to BPF form did not introduce significant added complexity, the implication would be that inference should be equally difficult in both classes. Of course, it may well be that the performance of presently available inference algorithms falls so far short of the best achievable performance that one is unable to draw conclusions from such experiments in either case.

We proceed to describe a series of experiments to measure the difficulty of inference on typical outputs of our BPF reductions. We choose for our “transformed” model the 1-of- n construction of Theorem 5, using the pairwise soft-XOR factors and three different numbers of vertices: four, five, and seven. Additionally, by inverting the sign of one of the soft-XOR components we created a “zero-of- n ” graph, in which the soft-XOR nodes should enforce the constraint that no vertex takes the value 1. The results for the six combinations of model are shown in Figure 1. The horizontal axis plots the strength (k) of the soft-XOR structures and the vertical axis shows the average L_1 error in marginals of the primary nodes, compared to their ideal values in the target model. The soft-XOR components approach true XORs as $k \rightarrow \infty$. For $k = 0$ the soft-XOR components are uniform, while for $k = 10$ the marginals of the resulting structure are within at most 2×10^{-8}

of the true marginals.

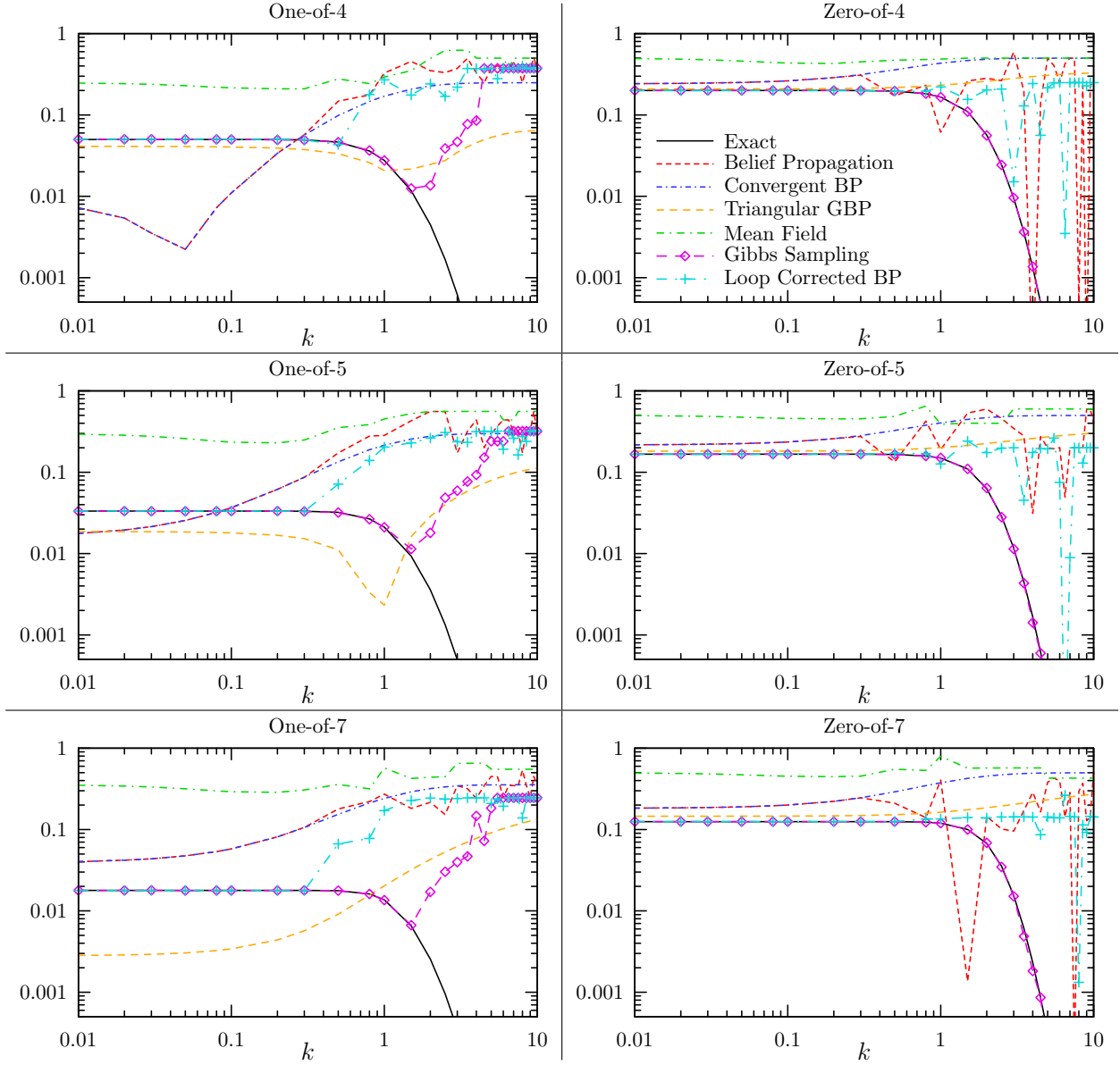


Figure 1: Results of experiments on one-of- n and zero-of- n models implemented using pairwise soft-XOR factors of strength k . Vertical axis is average L_1 error, measured with respect to the marginals of the target model.

We plotted the errors as a function of k for six inference algorithms. We used the libDAI implementation for each one (Mooij et al., 2010). Message-passing algorithms were run to convergence with a tolerance of 10^{-9} . The six algorithms are:

- **Exact** - True marginals in the “transformed” model are calculated using Junction Tree (Jensen et al., 1990)
- **Belief Propagation** - Belief Propagation (BP) with random order message updates
- **Convergent BP** - BP using the convergent double-loop algorithm of Heskes, Albers, and Kappen (HAK) (Heskes et al., 2003)
- **Mean Field** - the Mean Field approximation
- **Gibbs Sampling** - with 10^5 passes
- **Triangular GBP** - Generalized Belief Propagation, with triangular regions, using HAK updates
- **Loop Corrected BP** - Algorithm of Mooij et al. (2007), full cavity updates

One can see that all of the inference algorithms perform relatively poorly on the pairwise soft-XOR models. When the soft-XOR interactions are weak (small k), then Gibbs and LCBP have an accuracy which is close to exact; but in this regime the target model is not being accurately represented. When the interactions are strong, all of the algorithms but “exact” fail to capture the true marginals, except for Gibbs on the zero-of- n graph which is able to correctly report that all of the mass is in the all-zero state (presumably sampling gets “stuck” in this state). It is interesting that as k increases, LCBP deteriorates slightly earlier than Gibbs.

For small k , BP converges to a single stable fixpoint. For larger k it starts to deviate from the marginals found by the convergent HAK algorithm, and the error fluctuates randomly; for these runs we confirmed that it fails to converge to a stable fixpoint. It is interesting to note that sometimes the unstable marginals identified by the BP algorithm have much better accuracy than the HAK marginals, but this effect seems to be limited to the zero-of- n models, where the target distribution has all of its mass concentrated

on the all-zero state. LCBP also shows fluctuations, presumably because it harnesses the original (single-loop) BP algorithm to initialize its cavity distributions. It is reassuring that triangular GBP outperforms convergent BP, even if only by a small factor.

For comparison, we calculate the errors of the same algorithms on the one-of-7 and zero-of-7 models, when the XORs are hard-XORs implemented using three-wise factors:⁵

Algorithm	One-of-7	Zero-of-7
Belief Propagation	0.039	0.18
Mean Field	0.36	0.5
GBP (Triangular)	0.0028	0.146
Gibbs	0.24	0

We see that error of BP, MF, and GBP in the three-wise graphs is equivalent to the $k = 0$ error with soft-XORs in Figure 1, which suggests that these algorithms are not even able to capture the XOR behavior when it is encoded in this easy three-wise form, and that their additional performance deterioration as the strength of the soft-XOR widget increases is entirely due to “frustration” introduced by the difficult Leisink construction. Gibbs, on the other hand, gets “stuck” in the first state and is unable to make any transitions, so its error here is the same as its $k = 10$ soft-XOR error, in which the same behavior is seen. For the first three algorithms, which are the “message-passing”-based ones, the auxiliary variables had uniform marginals in both the $k = 0$ soft-XOR models and the hard-XOR models.

In summary, we have performed experiments with some simple structures representative of the output of our transformations. Only the exact Junction Tree algorithm was able to produce reasonable answers. These results suggest to us that inference in more general models under our BCFG transformation will also be difficult for modern inference algorithms. Although it is possible that an alternative transformation avoiding the pairwise soft-XOR technique explored here would result in more tractable models, we are skeptical that this would be the case. Our findings for the present experiments weakly support our conjecture that the MI problem is somehow more tractable in BCFGs. Additionally, we have uncovered an interesting class of models on which modern approximate inference algorithms are observed to perform very poorly. These models might make useful targets for evaluating future

⁵LCBP is not included because it had normalization problems in these models. Convergent BP has the same errors as BP.

research in approximate marginal inference.

Conclusion

By extending existing reducibility concepts to the problem of marginal inference, we were able to shed light on the generality and usefulness of a number of familiar classes of factor graph. Although the study of reducibility in constraint-satisfaction problems like SAT and optimization problems like MAP bears a number of similarities to marginal inference, in the process of our investigations we uncovered some new phenomena which seem to be unique to the marginal inference problem. It was first of all necessary to distinguish between strictly positive input graphs and those with zeroes. We also introduced the idea of “reducibility in a limit” to describe the expressive power of some of our classes. We hope that these developments, and especially the negative result for the full universality of binary pairwise factor graphs, have succeeded in demonstrating that reduction in marginal inference is a non-trivial subject which merits independent study.

We have left many questions to be addressed by future work. A rigorous justification of our definition of reducibility remains to be fleshed out. We are also curious to know whether planar binary pairwise factor graphs can reduce ordinary positive factor graphs without a limit. It may be interesting to look for alternative reductions which produce models on which inference is more tractable. Our empirical results demonstrated that existing approximate inference algorithms are not useful for analyzing the outputs of some of our reductions. Perhaps the algorithms themselves can be improved. Finally, there is a difficult open problem: given that binary pairwise factor graphs are only “almost universal”, can we identify and characterize a way in which marginal inference is more tractable on this class?

The results of these investigations and the questions raised by them should be of interest to anyone who is working on marginal inference, whether exact or approximate.

Acknowledgments

We are indebted to Martijn Leisink for his 3-wise soft-XOR construction, and to Joris Mooij for pointing us to it. We would also like to thank Tom Minka, Yee Whye Teh, Jonathan Yedidia, Tom Heskes, and Wim Wiegand.

for replying to queries.

Appendix: A motivation for the definition of reduction

We start with the definition of “polynomial-time reducibility”. A decision problem B is said to be *polynomial-time reducible* to a decision problem A if, given an “oracle” (idealized subroutine) which can solve instances of A in constant time, we can program a Turing machine to solve instances of B in polynomial time.

The marginal inference (MI) problem is not a decision problem, but we can easily generalize the concept of reducibility to this domain. As mentioned above (section 2.3), we consider MI to be the problem of estimating the probability of a given partial assignment (PA) in a given model. Reducing MI in one model to MI in another model thus equates to finding a way to estimate the probability of a PA in one model given the probability of one or more PAs in the second model.

We would like the cost of the reduction, excluding the cost of using MI to weigh the PAs in the target model, to be bounded by a polynomial function of the size of the input. Here the input size should be measured by the number of model parameters and the number of variables in the query PA. The numerical precision of the input model might also play a role, but we ignore such considerations here.

Our image of a MI reduction is at this point

$$P(x_r) = g(Q(y_{s_1}), Q(y_{s_2}), \dots, Q(y_{s_k})) \quad (37)$$

In other words, to calculate the mass in the original model P of a PA x_r we first calculate the masses in some transformed model Q of PAs y_{s_1}, \dots, y_{s_k} , where k is polynomially-bounded by the size of P , and then we apply some function $g : \mathbb{R}_+^k \rightarrow \mathbb{R}_+$ which can be computed with polynomial time-complexity. What forms is g allowed to take? Given that we expect the form of our reductions to be independent of the actual values of the factors, and given that $P(x_r)$ is just a sum of unnormalized joint values matching x_r , each of which is a product of factors, we might conjecture that g should be restricted to some combination of sums and products. In fact, since the product of probabilities usually only has meaning in cases where some kind

of independence condition is known to hold, which depends on factor values lying in some restricted subspace, we could further conjecture that g must consist only of summations; it must output a sum of its inputs. But a simple counterexample is the case of model duplication: let Q consist of two copies of P , each inducing an independent distribution on corresponding disjoint sets of variables x and x' , so that

$$Q(x, x') = \frac{1}{Z^2} \prod_{\alpha} \psi_{\alpha}(x_{\alpha}) \prod_{\alpha'} \psi_{\alpha'}(x'_{\alpha'}) = P(x)P(x') \quad (38)$$

Then we can see that $P(x_r^*) = \sqrt{Q(x_r = x_r^*, x'_r = x_r^*)}$, so in this case $g(x) = \sqrt{x}$ gives a valid “reduction,” in which we map the PA in P , x_r^* , to the single target PA in Q , $\{x_r = x_r^*, x'_r = x_r^*\}$. We attempted to exclude such arguably pathological cases with the “minimality” clause in the discussion at the start of section 3.1, but we have no rigorous proof that G must take the form of a summation even when such a condition is imposed. Instead, let us suppose as a matter of intuition that this is the case.

Now our definition of MI reduction amounts to estimating the mass of a PA in model P by weighing one or more PAs in a new model Q and summing these weights. Thus $P(x_i) = \sum_k Q(y_{s_k})$ for some set of PAs y_{s_k} , this set being a function of x_i . Given that the k conditions on the RHS are not intersecting⁶, we can say

$$P(x_r) = Q(y_{s_1} \cup \dots \cup y_{s_k}) \quad (39)$$

We first ask how this correspondence should treat two events, one of which is a subset of the other. Consider a variable-value x_j distinct from x_i , with $P(x_j) = \sum_k Q(y'_{r_k})$. What form should be taken by $P(x_i, x_j)$, which is another way of writing the probability of the event $x_i \cap x_j$, which is a subset of the events x_i and x_j ? It would be most natural for such a quantity to be calculated in our reduction by the intersection of the above events in Q , viz

$$P(x_i \cap x_j) = \sum_{k,l} Q(y_{s_l} \cap y'_{r_k}) = Q((y_{s_1} \cup y_{s_2} \cup \dots) \cap (y'_{r_1} \cup y'_{r_2} \cup \dots)) \quad (40)$$

This relationship follows from the laws of probability, and should be the only relation involving intersections of the image PAs for x_i and x_j that is

⁶If they were intersecting, it would imply some kind of strange overcounting, and ought to either lack generality or be rewritable in the non-intersecting form we desire.

preserved under changes in the model parameters. An alternative possibility is that $P(x_i, x_j)$ is represented by a summation containing an entirely new set of PAs, involving variables distinct from the first. But this would, by extension, require us to introduce new variables in Q for every subset of the variables in P . As the number of such subsets is exponential in the size of the model P , this would violate our constraint that the reduction be polynomial-time. Thus it makes sense to propose that our reductions must be containment-preserving, in the sense of equation 40.

Finally, we consider the number of terms allowed in the summation. Suppose that $P(x_{i_1}), P(x_{i_2}), \dots, P(x_{i_n})$, are all calculated under the reduction using a sum of more than one PA of Q . Then from the containment-preserving hypothesis above, and the distributive expansion of equation 40, the probability $P(x_{i_1}, \dots, x_{i_n})$ should correspond to a summation of at least 2^n PAs in Q , again violating the polynomial-time constraint. We conclude that the number of variables mapping to multiple PAs must be bounded. But since a reduction should apply to general models, the requirement that there be no “special” variables of this kind doesn’t restrict our allowed reductions in any significant way; thus, we demand that *each* PA in P must map to a single PA in Q . This gives us our definition of reduction.

References

- Baker, B. (1994). Approximation algorithms for np-complete problems on planar graphs. *Journal of the ACM (JACM)*, 41(1):153–180.
- Barahona, F. (1982). On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15:3241.
- Boros, E. and Hammer, P. (2002). Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225.
- Castillo, E., Gutiérrez, J., and Hadi, A. (1997). *Expert systems and probabilistic network models*. Springer Verlag.
- Chertkov, M. and Chernyak, V. (2006). Loop series for discrete statistical models on graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006.

- Chertkov, M., Gomez, V., and Kappen, H. (2009). Approximate inference on planar graphs using loop calculus and belief Propagation. Technical report, Los Alamos National Laboratory (LANL).
- Cooper, G. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial intelligence*, 42(2-3):393–405.
- Eaton, F. (2011). A conditional game for comparing approximations. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15.
- Fisher, M. (1966). On the dimer solution of planar Ising models. *Journal of Mathematical Physics*, 7:1776.
- Gallager, R. (1963). Low Density Parity Check Codes. Number 21 in Research monograph series.
- Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman & Co. New York, NY, USA.
- Globerson, A. and Jaakkola, T. (2007). Approximate inference using planar graph decomposition. *Advances in Neural Information Processing Systems 19*, 19:473.
- Heskes, T., Albers, K., and Kappen, B. (2003). Approximate inference and constrained optimization. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*, volume 13, pages 313–320.
- Jensen, F., Olesen, K., and Andersen, S. (1990). An algebra of Bayesian belief universes for knowledge-based systems. *Networks*, 20(5).
- Jung, K. and Shah, D. (2006). Inference in binary pair-wise markov random field through self-avoiding walk.
- Kasteleyn, P. (1963). Dimer statistics and phase transitions. *Journal of Mathematical Physics*, 4:287.
- Knuth, D. (2008). *The Art of Computer Programming, IV, Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions*. Addison-Wesley.

- Kschischang, F., Frey, B., and Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519.
- Kuratowski, K. (1930). Sur le problème des courbes gauches en topologie. *Fund. Math.*, 15:271–283.
- Leisink, M. (2010). Personal communication.
- Lichtenstein, D. (1982). Planar formulae and their uses. *SIAM journal on computing*, 11:329–343.
- Montanari, A. and Rizzo, T. (2005). How to compute loop corrections to the Bethe approximation. *Journal of Statistical Mechanics: Theory and Experiment*, 10:P10011.
- Mooij, J. et al. (2010). libDAI 0.2.5: A free/open source C++ library for Discrete Approximate Inference. <http://www.libdai.org/>.
- Mooij, J., Wemmenhove, B., Kappen, H., and Rizzo, T. (2007). Loop corrected belief propagation. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*.
- Pearl, J. (1982). Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the AAAI National Conference on AI*, pages 133–136.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- Sanghavi, S., Shah, D., and Willsky, A. (2009). Message passing for maximum weight independent set. *Information Theory, IEEE Transactions on*, 55(11):4822–4834.
- Sudderth, E., Wainwright, M., and Willsky, A. (2008). Loop series and bethe variational bounds in attractive graphical models. In *Advances in Neural Information Processing Systems 20*, pages 1425–1432. MIT Press, Cambridge, MA.
- Ungar, P. (1951). A theorem on planar graphs. *Journal of the London Mathematical Society*, 1(4):256.

- Valiant, L. (1979). The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8:410.
- Wagner, K. (1937). Über eine eigenschaft der ebenen komplexe. *Math. Ann.*, 144:570–590.
- Wainwright, M., Jaakkola, T., and Willsky, A. (2002). A new class of upper bounds on the log partition function. In *Proceedings of the Proceedings of the Eighteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 536–54.
- Watanabe, Y. and Fukumizu, K. (2011). New graph polynomials from the bethe approximation of the ising partition function. *Combinatorics, Probability and Computing*, 20(02):299–320.
- Weitz, D. (2006). Counting independent sets up to the tree threshold. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 140–149. ACM.
- Welling, M. and Teh, Y. (2001). Belief optimization for binary networks: A stable alternative to loopy belief propagation. *Uncertainty in Artificial Intelligence*.
- Wiegerinck, W. (2000). Variational approximations between mean field theory and the junction tree algorithm. In *Uncertainty in Artificial Intelligence*, volume 16, pages 626–633.
- Yedidia, J., Freeman, W., and Weiss, Y. (2001). Understanding belief propagation and its generalizations. *International Joint Conference on Artificial Intelligence*.