An Empirical Evaluation of Algorithms for Computing Equilibria in Games for Approximate Inference in Large Dimensional Probabilistic Graphical Models

by

Boshen Wang

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
(Computer and Information Science)
in the University of Michigan-Dearborn
2017

Master's Thesis Committee:

Assistant Professor Luis Ortiz, Chair
Professor William Grosky
Professor Bruce Maxim

TABLE OF CONTENTS

LIST OF FIGURES

Figure

ABSTRACT

Work in graphical models for game theory typically borrows from results in probabilistic graphical models. In this work, we instead consider the opposite direction. By using recent advances in equilibrium computation, we propose game-theoretic inspired, practical methods to perform probabilistic inference. We perform synthetic experiments using several different classes of Ising models, in order to evaluate our proposed approximation algorithms along with existing methods in the probabilistic graphical model literature. We also perform experiments using Ising models learned from the popular MNIST dataset. Our experiments show that the game-theoretic inspired methods are competitive with current state-of-the-art algorithms such as tree-reweighed message passing, and even consistently outperform said algorithms in certain cases.

Chapter I: Introduction

The connection between *graphical games* and *probabilistic graphical models (PGMs)* has been explored and exploited in a variety of works. These works typically use results from performing inference in PGMs to help facilitate equilibrium computation in graphical games. However, this approach has led to computational roadblocks, since exact inference in PGMs is tractable in graphs with bounded treewidth, but intractable in general (*Cooper*, 1990; *Shimony*, 1994; *Istrail*, 2000). Meanwhile, recent work in graphical games has produced efficient algorithms to compute correlated equilibria (*Papadimitriou*, 2005; *Jiang and Leyton-Brown*, 2015). Is there a way to apply these efficient algorithms to belief inference in PGMs, using the aforementioned connection between graphical games and PGMs? Trying to answer this question led us to take an approach contrary to most other works in this subject: We use results from equilibrium computation to help facilitate inference in PGMs.

It is well-known that *pure strategy Nash equilibrium (PSNE)* can be cast as a *maximum a posteriori (MAP)* assignment estimation problem in *Markov random fields (MRFs)*. We briefly explore how more general forms of equilibria relate to belief inference. We focus on a special type of game called *graphical potential games* (*Ortiz*, 2015), for which an equivalent MRF can be constructed whose "locally optimal" solutions correspond to arbitrary equilibria of the game. Thus, finding the equilibria in a graphical potential game would lead one to solutions in the equivalent MRF. We employ ideas from the literature on learning in games (*Fudenberg and Levine*, 1999), such as no-regret algorithms and fictitious play, to propose game-theoretic inspired, practical, and effective heuristics for belief inference in MRFs. We then experimentally evaluate our proposed algorithms, along with existing techniques from PGM literature.

Chapter II: Preliminaries

In this section we will briefly define various concepts and notation used in the remainder of this thesis.

2.1 Terminology and Notation

Denote $x \equiv (x_1, x_2, \ldots, x_n)$ as an n-dimensional vector, and $x_{-i} \equiv (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ as the same vector without component i. For every set $S \subset [n] \equiv \{1, \ldots, n\}$, denote by $x_S \equiv (x_i : i \in S)$ the sub-vector formed from x using only components in S. $S^c \equiv [n] - S$ denotes the complement of S. We can denote $x \equiv (x_S, x_{S^c}) \equiv (x_i, x_{-i})$ for every i. If $A_1, \ldots, A_n$ are sets, denote by $A \equiv \times_{i \in [n]} A_i$, $A_{-i} \equiv \times_{j \in [n]-\{i\}} A_j$ and $A_S \equiv \times_{j \in S} A_j$.

Let $G = (V, E)$ be an undirected graph, with a finite set of n *vertices* or *nodes* $V = \{1, \ldots, n\}$ and a set of undirected edges E. For each node i, let $\mathcal{N}(i) \equiv \{j \mid (i, j) \in E\}$ be the set of neighbors of i in G, *not including* i, and $N(i) \equiv \mathcal{N}(i) \cup \{i\}$ be the neighbors set *including* i. A *clique* C of is a set of nodes in G that are mutually connected: for all $i, j \in C, (i, j) \in E$. In addition, C is *maximal* if there is no other node k outside C that is also connected to each node in C: for all $k \in V - C, (k, i) \notin E$ for some $i \in C$.

Hypergraphs are generalizations of regular graphs. A *hypergraph graph* $\mathcal{G} = (V, \mathcal{E})$ is defined by a set of nodes V and a set of *hyperedges* $\mathcal{E} \subset 2^V$. The *primal graph* of the hypergraph is the graph induced by taking each hyperedge and forming cliques of nodes in a regular graph.

2.2 Probabilistic Graphical Models

Probabilistic Graphical Models (PGMs) are models which have shown themselves to be suitable for complex, structured, high-dimensional systems found in the real world. At the most basic level, PGMs consist of a graph in which each node i represents a random variable X_i, and the edges represent conditional independence assumptions about those random variables.

2.2.1 Markov Random Fields and Other Definitions

By definition, a joint probability distribution P is a *Markov random field (MRF)* with respect to (wrt) an undirected graph G, if for all x, for every node i, $P(X_i = x_i \mid X_{-i} = x_{-i}) = P(X_i = x_i \mid X_{\mathscr{N}(i)} = x_{\mathscr{N}(i)})$. In other words, the probability of X_i being some value is the same whether all other nodes are given or only the neighbors of X_i are given. The neighbors of X_i, $X_{\mathscr{N}(i)}$ are referred to as the *Markov blanket* of node/variable X_i.

Also by definition, a join distribution P is a *Gibbs distribution* wrt an indirected graph G if it can be expressed as $P(X = x) = \prod_{C \in \mathscr{C}} \Phi_C(x_C)$. $\mathscr{C}$ represents the set of all maximal cliques in G. Φ_C are some functions indexed by $C \in \mathscr{C}$ which map every possible value of x_c (the random variables associated with the nodes in C) can take to a non-negative number.

Finally, the Hammersly-Clifford Theorem (*Hammersley and Clifford*, 1971; *Besag*, 1974) states: Let P be a *positive* joint probability distribution. That is, $P(x) > 0$ for all x. Then, P is an MRF with respect to G if and only if P is a Gibbs distribution with respect to G.

In the context of the theorem, the functions Φ_C are in fact positive, which allows us to define MRFs in terms of *local potential functions* $\{\phi_C\}$ over each clique C in the graph. Define the function $\Psi(x) \equiv \sum_{C \in \mathscr{C}} \phi_C(x_C)$. Let us refer to any function of this form as a *Gibbs potential* with respect to G. Another way to express an MRF is $P(X = x) \propto \exp(\sum_{C \in \mathscr{C}} \phi_C(x_C)) = \exp(\Psi(x))$.

Ising models (IMs) are a special class of MRFs originating from statistical physics. An *Ising model (IM)* wrt an undirected graph $G = (V, E)$ is an MRF wrt G such that $P_\theta(x) \propto \exp(\sum_{i \in V} b_i x_i + \sum_{(i,j) \in E} w_{i,j} x_i x_j)$ where $\theta \equiv (\mathbf{b}, \mathbf{W})$ is the set of node biases b_i's and edge-weights w_{ij}'s, which are the parameters defining the joint distribution P_θ over $\{-1, +1\}^n$.

2.2.2 Inference-related Problems in MRFs

Several problems of interest exist in the context of MRFs. One problem is to compute a *most likely assignment* x^*, the most likely outcome for MRF P. More precisely, $x^* \in \arg\max_x P(X = x) = \arg\max_x \sum_{C \in \mathscr{C}} \phi_C(x_C))$. A related problem is to compute the *individual marginal probability* $P(X_i = x_i) = \sum_{x_{-i}} P(X_i = x_i, X_{-i} = x_{-i}) \propto \sum_{x_{-i}} \exp(\sum_{C \in \mathscr{C}} \phi_C(x_C)))$ for each variable X_i. Another related problem is to compute the normalizing constant $Z = \sum_x \exp(\sum_{C \in \mathscr{C}} \phi_C(x_C)))$ (also known as the *partition function* of the MRF).

There is also a set of problems which concern "belief updating." These problems involve computing information related to the *posterior probability distribution* P', after having observed the outcome of some variables (the *evidence*). For MRFs, this problem is computationally equivalent to that of computing prior marginal probabilities.

2.2.3 Brief Overview of Computational Results in PGMs

Exact versions of most inference-related problems in MRFs are in general intractable, though in certain cases polynomial-time algorithms do exist (for example, *Istrail* (2000), *Wang et al.* (2013)). Typically, running times for exact algorithms are polynomial only for graphs with bounded tree-width (*Russell and Norvig*, 2003).

Several heuristic approaches to approximate inference exist, although approximate inference is also intractable in general. One approximation approach of particular interest to us is *variational inference* (*Jordan et al.*, 1999; *Jaakkola*, 2000). The general idea is to approximate an intractable MRF P with a "closest" probability distribution Q^* within a "computationally tractable" class Q. More formally: $Q^* \in \arg\max_{Q \in \mathscr{Q}} \mathrm{KL}(Q \parallel P)$, where $\mathrm{KL}(Q \parallel P) \equiv \sum_x Q(x) \ln \frac{Q(x)}{P(x)}$ is the *Kullback-Leibler (KL) divergence* between probability distributions P and Q wrt Q. The simplest example is the *mean-field (MF) approximation*, in which $\mathscr{Q} = \{Q \mid Q(x) = \prod_i Q(x_i) \text{ for all } x \in \Omega\}$ consists of all possible *product* distributions.

2.3 Game Theory

Game theory (*von Neumann and Morgenstern*, 1947) mathematically models how rational agents interact with each other and make decisions in a system (a "game"). In this paper we focus on *non-cooperative* settings, where individuals act *independently* and *only* seek to maximize their own utility.

2.3.1 Game Representations

Basically speaking, games consist of three components: players of the game, actions for those players, and payoffs for those actions. Let $V = [n]$ denote a finite set of n players. For each player $i \in V$, let A_i denote the set of *actions* or *pure strategies* that i can choose to play. Let $A \equiv \times_{i \in V} A_i$ denote the set of *joint actions*, and let $x \equiv (x_1, ..., x_n) \in A$ denote one joint action. Denote x_i as the action of player i in x, and $x_{-i} \equiv (x_1, ...x_{i-1}, x_{i+1}, ...x_n)$ as the joint action of all players other than i. Finally, let $M_i : A_i \to \mathbb{R}$ denote the *payoff/utility function* of player i. If the A_i's are finite, then M_i is called the *payoff matrix* of player i. Games represented this way are called *normal-* or *strategic-form games*.

Games which include a large number of players and actions can have impractically large normal-form representations. Probabilistic graphical models offer options for more compact representations of games (*La Mura*, 2000; *Kearns et al.*, 2001; *Koller and Milch*, 2003; *Leyton-Brown and Tennenholtz*, 2003; *Jiang and Leyton-Brown*, 2008). In particular, this paper is presented in the context of a generalization of *graphical games* (*Kearns et al.*, 2001) called *graphical multi-hypermatrix games (GMhG)*.

A GMhG consists of a *directed* graph $G = (V, E)$ in which there is a node $i \in V$ in G for each of the n players in the game. The set of directed edges E defines a set of neighbors $\mathcal{N}(i) \equiv \{j \mid (j,i) \in E, i \neq j\}$ whose actions affect the payoff function of i. Each player $i \in V$ has a set of actions A_i, a hypergraph where the vertex set is the player's *inclusive neighborhood* $N(i) \equiv \mathcal{N}(i) \cup \{i\}$ and the hyperedge set is a set of *cliques* $\mathscr{C}_i \subset 2^{N(i)}$, and a set of *local-clique payoff hypermatrices* $\{M'_{i,C} : A_C \to \mathbb{R} \mid C \in \mathscr{C}_i\}$. Finally, *local* and *global payoff hypermatrices* $M'_i : A_{N(i)} \to \mathbb{R}$ and M_i :

$A_i \to \mathbb{R}$ of player i are defined as $M'_i(x_{N(i)}) \equiv \sum_{C \in \mathscr{C}_i} M'_{i,C}(x_C)$ and $M_i(x) \equiv M'_i(x_{N(i)})$, respectively.

GMhGs have a special class of instances referred to as *Graphical potential games*. A detailed characterization and discussion can be found in (*Ortiz*, 2015). Graphical potential games will be referenced later in Section 3.1.

2.3.2 Equilibria as Solution Concepts

Equilibria are generally considered to be the solutions of a game. Several different notions of equilibria exist, though the common theme among them (at least in the non-cooperative setting) is that there exists some situation where no players can improve their payoffs by deviating from their current course of action. Perhaps the most straightforward form of equilibrium is a *Pure Strategy Nash Equilibrium (PSNE)*. A PSNE is a joint action $x*$ such that for all players i, and for all actions for those players x_i, $M_i(x_i^*, x_{-i}^*) \geq M_i(x_i, x_{-i}^*)$. Essentially, if player i always takes action x_i^*, there is no other action that player could always take to improve their payoff, assuming no other player changes either. Only some games have a PSNE, however. The Prisoner's Dilemma is a famous example of a game with PSNE. On the other hand, even a simple game like "Rock, Paper, Scissors" does not have a PSNE.

For games like "Rock, Paper, Scissors," a more general form of equilibrium called *mixed strategy Nash equilibrium (MSNE)* is used. As the name implies, MSNEs involve players using *mixed strategies* instead of pure strategies. A mixed strategy of player i is a probability distribution Q_i over A_i such that $Q(x_i)$ is the probability that i chooses to play action x_i. A *joint mixed strategy* a joint probability distribution Q encompassing all players' mixed strategies. Because players are assumed to play independently, Q is a product distribution: $Q(x) = \prod_i Q_i(x_i)$. The joint mixed strategy without player i is: $Q_{-i}(x_{-i}) \equiv \prod_{j \neq i} Q_j(x_j)$. When employing a mixed strategy Q, a player i's payoff function becomes an *expected payoff*: $\sum_x Q(x) M_i(x)$, which we denote as simply $M_i(Q)$. The *conditional expected payoff* of player i given that they play action x_i is $\sum_{x_{-i}} Q_{-i}(x_{-i}) M_i(x_i, x_{-i})$, which we denote as $M_i(x_i, Q_{-i})$.

An MSNE is a joint mixed strategy Q^* that is a product distribution formed by each player's

current mixed strategy Q_i^*, such that, for all players i, Q_i^* leads to a better expected payoff than any alternate mixed strategy Q_i': $M_i(Q_i^*, Q_{-i}^*) \geq M_i(Q_i', Q_{-i}^*)$. *Every* game in normal form has at least one such equilibrium (*Nash*, 1951). However, finding these equilibria exactly can be difficult if the game contains many players and actions. A useful relaxation of MSNE is treat cases where players can only gain a very small amount ε if they deviate as though they were at equilibrium. This relaxation allows for approximate equilibria solutions instead of exact ones, which eases computation. Given $\varepsilon \geq 0$, an *(approximate) ε-Nash equilibrium (MSNE)* is defined as above, except that the equilibrium condition becomes $M_i(Q_i^*, Q_{-i}^*) \geq M_i(Q_i', Q_{-i}^*) - \varepsilon$.

Another form of equilibrium which is more general than MSNE is called *correlated equilibrium (CE)* (*Aumann*, 1974). Unlike MSNE, a CE can be a full joint distribution rather than a product distribution. Formally, a CE is a joint probability distribution Q over A such that, for all players i, $x_i, x_i' \in A_i$, $x_i \neq x_i'$, and $Q(x_i) > 0$, $\sum_{x_{-i}} Q(x_{-i}|x_i) M_i(x_i, x_{-i}) \geq \sum_{x_{-i}} Q(x_{-i}|x_i) M_i(x_i', x_{-i})$, where $Q(x_i) \equiv \sum_{x_{-i}} Q(x_i, x_{-i})$ is the (marginal) probability that player i will play x_i according to Q and $Q(x_{-i}|x_i) \equiv Q(x_i, x_{-i}) / \sum_{x_i'} Q(x_i', x_{-i})$ is the conditional probability given x_i. The ε relaxation can also be applied to CEs, by adding the term "$-\varepsilon$" to the right-hand side of the condition above, given that $\varepsilon > 0$. This gives us the definition of an *approximate ε-CE*.

2.3.3 Brief Overview of Results in Computational Game Theory

In two-player *zero-sum* games, where the sum of the entries in the payoff matrices equals zero, an MSNE can be computed in polynomial time. In fact, the game is equivalent to linear programming (*von Neumann and Morgenstern*, 1947; *Szép and Forgoó*, 1985; *Karlin*, 1959). However, the complexity of computing MSNE in any normal form game was not settled until recently. For a more detailed discussion of recent results in this area, see (*Ortiz and Irfan*, 2017). The general theme of most results is that computing MSNE, and even PSNE in some cases, is intractable in the worst case.

Computing PSNE and MSNE in graphical games is similar to MRFs and constraint networks in terms of complexity: polynomial time for a bounded treewidth graph, but intractable in gen-

eral (*Kearns et al.*, 2001; *Gottlob et al.*, 2003; *Daskalakis and Papadimitriou*, 2006; *Ortiz*, 2014). Several heuristics for MSNE computation have been developed for use in general graphs (*Vickrey and Koller*, 2002; *Ortiz and Kearns*, 2003; *Daskalakis and Papadimitriou*, 2006). In contrast, computing CE can be done in polynomial time, both in normal form games, and other compactly representable games, including graphical games (*Papadimitriou*, 2005; *Jiang and Leyton-Brown*, 2015).

Chapter III: Theoretical Connections and Implications

Given the recent advancements in equilibria computation, we asked ourselves the following question: *Can we leverage advances in computational game theory for problems in the probabilistic graphical models community?* Establishing a strong bilateral connection between equilibria computation in games and inference in PGMs could allow us to apply algorithms from one area to the other. It is important to note that the idea of adopting methods from PGMs in order to compute equilibria in graphical games is not new. However, the focus of this paper is to go the opposite direction: using algorithms for computing equilibria in games to perform belief inference in PGMs.

3.1 PSNE and Approximate MAP Inference

It is possible to show that PSNE in a potential game can be considered as MAP assignments in an equivalent MRF. Consider an MRF P with respect to graph G and Gibbs potential Ψ defined by the set of potential functions $\{\phi_C\}$. For each node i, denote by $\mathscr{C}_i \subset \mathscr{C}$ the subset of cliques in G that include i. The inclusive neighborhood of player i is given by $N(i) = \cup_{C \in \mathscr{C}_i} C$. Define an *MRF-induced* GMhG with the same graph G, and for each player i, a hypergraph with hyperedges $\mathscr{C}_i$ and local clique payoff hypermatrices $M'_{i,C}(x_C) \equiv \phi_C(x_C)$ for all $C \in \mathscr{C}_i$. This MRF-induced game has two important properties:

Property 1. *The representation size of the MRF-induced game is the same as that of the MRF: not exponential in the largest neighborhood size, but the size of the largest clique in G.*

Property 2. *The MRF-induced game is a graphical potential game (Ortiz, 2015) with graph G and (Gibbs) potential function Ψ: i.e., for all i, x and x'_i, $M_i(x_i, x_{-i}) - M_i(x'_i, x_{-i}) = M'_i(x_i, x_{\mathscr{N}(i)}) -$*

$M_i'(x_i', x_{\mathcal{N}(i)})$

$$
\begin{aligned}
&= \sum_{C \in \mathscr{C}_i} \phi_C(x_i, x_{C-\{i\}}) - \sum_{C \in \mathscr{C}_i} \phi_C(x_i', x_{C-\{i\}}) \\
&= \sum_{C \in \mathscr{C}_i} \phi_C(x_i, x_{C-\{i\}}) + \sum_{C' \in \mathscr{C} - \mathscr{C}_i} \phi_{C'}(x_{C'}) - \\
&\quad \sum_{C \in \mathscr{C}_i} \phi_C(x_i', x_{C-\{i\}}) - \sum_{C' \in \mathscr{C} - \mathscr{C}_i} \phi_{C'}(x_{C'}) \\
&= \Psi(x_i, x_{-i}) - \Psi(x_i', x_{-i}).
\end{aligned}
$$

Remark 1. The MRF-induced game from the last property is guaranteed to be solvable with a PSNE by using *sequential best-response dynamics*, since it is a potential game. In fact, a joint action x^* is a PSNE of the game *if and only if* x^* is a local maxima or critical point of the original MRF P. Therefore, the MRF-induced game always has PSNE. Going the other direction, one can define a *game-induced MRF* for any potential game by using the potential function of the game. The set of local maxima and critical points of the game-induced MRF corresponds exactly to the set of PSNE of the potential game. This connection means that solving the local-MAP problem in MRFs is PLS-complete in general (*Johnson et al.*, 1988).

In short, we can use algorithms for PSNE computation as heuristics to compute *locally optimal* MAP assignments of P and *vice versa*.

3.2 CE and Belief Inference

Shifting our focus to more general types of equilibrium like CE allows us to make other connections between equilibria computation and probabilistic inference. Let S be a subset of the players, and denote by $Q_S(x_S) \equiv \sum_{x_{V-S}} Q(x)$ the marginal probability distribution of Q over possible joint actions of players in S. In the MRF-induced game, we can express the condition for CE as: for all $i, x_i, x_i' \neq x_i$, $\sum_{x_{\mathcal{N}(i)}} Q_{N(i)}(x_i, x_{\mathcal{N}(i)}) \sum_{C \in \mathscr{C}_i} \phi_C(x_i, x_{C-\{i\}}) \geq \sum_{x_{\mathcal{N}(i)}} Q_{N(i)}(x_i, x_{\mathcal{N}(i)}) \sum_{C \in \mathscr{C}_i} \phi_C(x_i', x_{C-\{i\}})$.

By commuting the sums and simplifying we get the following equivalent condition:

$$\begin{aligned}
&\sum_{C\in\mathscr{C}_i}\sum_{x_{C-\{i\}}} Q(x_i,x_{C-\{i\}})\phi_C(x_i,x_{C-\{i\}}) \\
\geq &\sum_{C\in\mathscr{C}_i}\sum_{x_{C-\{i\}}} Q(x_i,x_{C-\{i\}})\phi_C(x_i',x_{C-\{i\}})
\end{aligned} \tag{3.1}$$

This simplification is important because it shows that we only need distributions over the original cliques, *not* the induced neighborhoods, in order to represent CE in MRF-induced games. We are able to keep the CE representation size to be the same as the game's representation size, unlike in *Kakade et al.* (2003).

An alternative but equivalent condition can be found by using the fact that the MRF-induced game is also a potential game, along with some other definitions.

$$\begin{aligned}
&\sum_{x_{-i}} Q(x_i,x_{-i})M_i(x_i,x_{-i}) \geq \sum_{x_{-i}} Q(x_i,x_{-i})M_i(x_i',x_{-i}) \\
&\sum_{x_{-i}} Q(x_i,x_{-i})\left(M_i(x_i,x_{-i}) - M_i(x_i',x_{-i})\right) \geq 0 \\
&\sum_{x_{-i}} Q(x_i,x_{-i})\left(M_i(x_i,x_{-i}) - M_i(x_i',x_{-i})\right) \geq 0 \\
&\sum_{x_{-i}} Q(x_i,x_{-i})\left(\Psi(x_i,x_{-i}) - \Psi(x_i',x_{-i})\right) \geq 0 \\
&\sum_{x_{-i}} Q(x_i,x_{-i})\left(\ln P(x_i,x_{-i}) - \ln P(x_i',x_{-i})\right) \geq 0
\end{aligned}$$

The last expressions leads to the following equivalent condition:

$$\begin{aligned}
&\sum_{x_{-i}} Q(x_i,x_{-i})[-\ln P(x_i,x_{-i})] \\
\leq &\sum_{x_{-i}} Q(x_i,x_{-i})[-\ln P(x_i',x_{-i})]
\end{aligned} \tag{3.2}$$

Next, we will borrow some concepts from information theory, like (Shannon's) entropy, cross entropy, and relative entropy (or Kullback-Leibler divergence) to make some additional statements

about the previous condition's implications. For an introduction to these topics, see *Cover and Thomas* (2006).

Remark 2. For any distribution Q', let $H(Q',P) \equiv \sum_x Q'(x)[-\log_2 P(x)]$ be the *cross entropy* between probability distributions Q' and P, with respect to P. Let $Q_{-i}(x_{-i}) \equiv \sum_{x_i} Q(x_i,x_{-i})$ be the marginal distribution of play over the joint-actions of all players *except* player i. Finally, $Q'_i Q_{-i}$ is the joint distribution defined as $(Q'_i Q_{-i})(x) \equiv Q'_i(x_i)Q_{-i}(x_{-i})$ for all x. Then, condition 3.2 implies the following sequence of conditions, which hold for all i:

$$\begin{aligned}
\sum_x Q(x)[-\ln P(x)] &\leq \sum_{x_{-i}} Q_{-i}(x_{-i})[-\ln P(x'_i,x_{-i})] \text{ for all } x'_i \\
H(Q,P) &\leq \min_{x'_i} \sum_{x_{-i}} Q_{-i}(x_{-i})[-\log_2 P(x'_i,x_{-i})] \\
&= \min_{Q'_i} \sum_x Q'_i(x_i)Q_{-i}(x_{-i})[-\log_2 P(x_i,x_{-i})] \\
&= \min_{Q'_i} H(Q'_i Q_{-i},P)
\end{aligned}$$

Any CE of the MRF-induced game is an approximate local optimum (or critical point) of an approximation of the MRF based on a special type of cross entropy minimization. In actuality, the condition is that of a *coarse CE (CCE)* (*Hannan*, 1957; *Moulin and Vial*, 1978), which is a superset of CE. We will discuss the impact of this later in this section. The following property summarizes this remark.

Property 3. *For any MRF P, any correlated equilibria Q of the game induced by P satisfies* $H(Q,P) \leq \min_i \min_{Q'_i} H(Q'_i Q_{-i},P)$.

Remark 3. For any player i, for any marginal/individual distribution of play Q'_i, let $H(Q'_i) \equiv \sum_{x_i} Q'_i(x_i)[-\log_2 Q'_i(x_i)]$ be its marginal entropy. The *Kullback-Leibler (KL) divergence* between Q' and P, with respect to Q' for any distribution Q' and P is $\mathrm{KL}(Q' \parallel P) \equiv \sum_x Q'(x)\log_2(Q'(x)/P(x)) = H(Q',P) - H(Q')$. Denote by $H(Q_{i|-i}) \equiv$

$\sum_{x_i,x_{-i}} Q(x_i,x_{-i}) \log_2(Q(x_i,x_{-i})/Q_{-i}(x_i)) = H(Q_{-i}) - H(Q)$ the conditional entropy of the individual play of player i given the joint play of all the players except i, with respect to Q.

We can express condition 3.2 as the following equivalent conditions, which hold for all i.

$$\begin{aligned}
&\mathrm{KL}(Q \parallel P) + H(Q) \\
\leq &\min_{Q'_i} \mathrm{KL}(Q'_i Q_{-i} \parallel P) + H(Q'_i Q_{-i}) \\
&\mathrm{KL}(Q \parallel P) + H(Q_{i|-i}) \\
\leq &\min_{Q'_i} \mathrm{KL}(Q'_i Q_{-i} \parallel P) + H(Q'_i)
\end{aligned}$$

Any CE of a MRF-induced game is an approximate local optimum (or critical point) of a special kind of variational approximation of the MRF. This leads us to another property.

Property 4. *For any MRF P, any correlated equilibria Q of the game induced by P satisfies $KL(Q \parallel P) \leq \min_i \left[\min_{Q'_i} KL(Q'_i Q_{-i} \parallel P) + H(Q'_i)\right] - H(Q_{i|-i})$.*

3.2.1 MSNE and Mean-Field Approximations

MSNE are a special case of CE, where the joint mixed strategy $Q(x) = \prod_i Q_i(x_i)$ is actually a product distribution. Denote by $Q^{\times}_{-i}(x_{-i}) \equiv \prod_{j \neq i} Q_j(x_j) = \sum_{x_i} Q(x)$ the marginal joint action of play over all the players except i, and denote by $(Q'_i Q^{\times}_{-i})$ the probability distribution defined such that the probability of x is $(Q'_i Q^{\times}_{-i})(x) \equiv Q'_i(x_i) Q^{\times}_{-i}(x_{-i})$.

For MSNE, the equilibrium conditions imply the following conditions, for all i, for all x_i such that $Q_i(x_i) > 0$:

$$\begin{aligned}
&\sum_{x_{-i}} Q_i(x_i) Q^{\times}_{-i}(x_{-i}) [-\ln P(x_i, x_{-i})] \\
= &\min_{x'_i} \sum_{x_{-i}} Q_i(x_i) Q^{\times}_{-i}(x_{-i}) [-\ln P(x'_i, x_{-i})]
\end{aligned}$$

If we denote by $\mathscr{X}_i^+ \equiv \{x_i \in A_i \mid Q_i(x_i) > 0$, the last condition implies that:

$$\sum_{x_i \in \mathscr{X}_i^+} \sum_{x_{-i}} Q_i(x_i) Q_{-i}^{\times}(x_{-i})[-\ln P(x_i, x_{-i})] =$$
$$\left(\sum_{x_i \in \mathscr{X}_i^+} Q_i(x_i) \right) \min_{x_i'} \sum_{x_{-i}} Q_{-i}^{\times}(x_{-i})[-\ln P(x_i', x_{-i})]$$

The previous condition is equivalent to:

$$\sum_{x_i} \sum_{x_{-i}} Q_i(x_i) Q_{-i}^{\times}(x_{-i})[-\ln P(x_i, x_{-i})]$$
$$= \min_{x_i'} \sum_{x_{-i}} Q_{-i}^{\times}(x_{-i})[-\ln P(x_i', x_{-i})] \ ,$$

which in turn is equivalent to the following two expressions:

$$H(Q, P) = \min_{Q_i'} H(Q' Q^{\times}, P)$$
$$\mathrm{KL}(Q \parallel P) + H(Q_i) = \min_{Q_i'} \mathrm{KL}(Q_i' Q_{-i}^{\times} \parallel P) + H(Q_i')$$

A NE Q of the MRF-induced game is almost *a locally optimal mean-field approximation, except for the extra entropic term.* For MSNE, we have the following condition which is tighter than for arbitrary CEs.

Property 5. *For any MRF P, any MSNE Q of the game induced by P satisfies* $KL(Q \parallel P) = \left[\min_{Q_i'} KL(Q_i' Q_{-i}^{\times} \parallel P) + H(Q_i')\right] - H(Q_i)$*, for all i.*

Remark 4. This discussion suggests that one could use modified versions of existing algorithms for computing MSNE, as heuristics to finding a mean-field approximation of the true marginals. Recent work in the other direction explores connections between *learning in games* and mean-field approximations in machine learning (*Fudenberg and Levine*, 1999).

3.2.2 Some Computational Implications

Algorithms developed by *Kakade et al.* (2003) can be used to compute a CE of an MRF-induced game in *polynomial time*. The resulting CE will be a *polynomially-sized mixture of product distributions*. However, these algorithms make a polynomial number of calls to an "ellipsoid algorithm," which is known to be slow in practice. Thus, these algorithms may not be very practical when used with large input sizes.

As an alternative, our discussions above (specifically that of condition 3.2) suggest that any learning algorithm that guarantees convergence to the set of *CCE* can be used as a heuristic for approximate inference. There exist "no-regret" learning algorithms which satisfy those conditions. These algorithms will be included in our experimental evaluation (Chapter IV).

3.3 Approximate Fictitious Play in a Two-player Potential Game for Belief Inference in Ising Models

This section presents a game-theoretic *fictitious play* approach to estimation of node marginal probabilities in MRFs. We focus on Ising models, a type of MRF which are simple and have uses in machine learning and AI applications. Generalizing this algorithm to arbitrary MRFs is possible, but not included here as it is not our focus.

The fictitious play algorithm constructs a two-player potential game, where both players have identical payoffs. This kind of game has the *fictitious play property* (*Monderer and Shapley*, 1996), which says that the empirical play of fictitious play is guaranteed to converge to an MSNE of the potential game. In fictitious play, each player uses the *empirical distribution of play* as an estimate of how the other players would behave in the future, then responding to that estimate. This is in contrast to *sequential best-response*, where players only look at other players' *last* action, to make their response. Sequential best-response converges to PSNE in potential games.

The algorithm goes as follows: denote by $\mathbb{T}_G$ the set of all spanning trees of connected (undirected) graph $G = (V,E)$ that are maximal with respect to E. If spanning tree $T \in \mathbb{T}_G$, denote by $E(T) \subset E$ the set of edges of T. Let $\widetilde{M}_{\mathscr{T}}(\mu,T) \equiv \sum_{(i,j)\in E} \mathbb{1}[(i,j) \in E(T)]\, w_{ij}\mu_{(i,j)}$ and

$\Psi_{X,\mathscr{T}}(x,T) \equiv \sum_{i\in V} b_i x_i + \sum_{(i,j)\in E} \mathbb{1}[(i,j)\in E(T)] w_{ij} x_i x_j$.

Initialize $x^{(1)} \leftarrow \text{Uniform}(\{-1,+1\}^n)$, and for each $(i,j)\in E$, $\widehat{\mu}^{(1)}_{(i,j)} \leftarrow x^{(1)}_i x^{(1)}_j$. At each iteration $l = 1,2,\ldots,m$,

1: $\mathbb{T}^{(l)} \leftarrow \arg\max_{T\in\mathbb{T}_G} \widetilde{M}_{\mathscr{T}}(\widehat{\mu}^{(l)}_{(i,j)}, T)$

2: $T^{(l)} \leftarrow \text{Uniform}\left(\arg\max_{T\in\mathbb{T}_G} \mathbb{T}^{(l)}\right)$

3: $s_l \leftarrow \text{Uniform}(\{1,\ldots,l\})$

4: $\mathscr{X}^{(l+1)} \leftarrow \arg\max_{x\in\{-1,+1\}^n} \Psi_{X,\mathscr{T}}(x, T^{(s_l)})$

5: $x^{(l+1)} \leftarrow \text{Uniform}\left(\mathscr{X}^{(l+1)}\right)$

6: **for all** $(i,j)\in E$ **do**

7: $v^{(l+1)}_{(i,j)} \leftarrow x^{(l+1)}_i x^{(l+1)}_j \times \begin{cases} 1, & \text{if MSNE,} \\ \mathbb{1}\left[(i,j)\in E(T^{(s_l)})\right], & \text{if CE} \end{cases}$

8: $\widehat{\mu}^{(l+1)}_{(i,j)} \leftarrow \frac{l\,\widehat{\mu}^{(l)}_{(i,j)} + v^{(l+1)}_{(i,j)}}{l+1}$

9: **end for**

Lastly, for each Ising-model's random-variable index $i = 1,\ldots,n$, set $p^{(m+1)}_i = \frac{1}{m+1}\sum_{l=1}^{m+1} \mathbb{1}\left[x^{(l)}_i = 1\right]$ as the estimate of the exact Ising-model's marginal probability $p_i \equiv \mathbf{P}(X_i = 1)$.

The running time of the algorithm is dominated by the computation of the maximum spanning tree in Step 1, which is $O(|E| + n\log n)$. All other steps take $O(|E|)$ or less.

The two-player potential game implicit in the algorithm consists of a *"joint-assignment" (JA)* player and a *"spanning-tree" (ST)* player. The potential function is $\Psi_{X,\mathscr{T}}(x,T)$. The payoff functions for both players equals the potential function. Determining the ST player's best response during each iteration involves computing a maximal spanning tree over the graph (Step 1), which can be done easily. However, for the JA player, finding the best response is as hard as computing a MAP assignment of another IM with the same graph, same node biases, and slightly different edge-weights. To circumvent this problem, we draw one tree *uniformly at random* from the empirical distribution (Step 4), rather than take the entire distribution into account. Then, the JA player simply uses the best-response to that randomly drawn tree.

While the ST player conducts standard fictitious play, the JA player actually behaves according to *stochastic fictitious play* (*Fudenberg and Levine*, 1999), since they are randomly picking from the empirical distribution. Stochastic fictitious play also converges to MSNE in potential games (*Hofbauer and Sandholm*, 2002). It is important to note that stochastic fictitious play actually involves *simultaneous* play, rather than *sequential* (as in this algorithm). Thus, we have a sort of "hybrid" sequential fictitious play algorithm.

Chapter IV: Experiments

In this section we present the results and methodology of our experimental evaluation of the performance of game-theoretic inspired heuristics compared to other popular approximation algorithms in PGM literature.

4.1 Synthetic Models

We performed experiments on synthetic models using the fictitious play heuristic proposed in this paper, as well as other popular algorithms and heuristics in PGM literature, as part of an empirical evaluation on these algorithms' relative performance in the context of belief inference.

4.1.1 Experimental Design

We used Ising models with $d \times d$ planar grid graphs. The number of nodes in these models is d^2, while the number of edges is $(d-1)*d*2$. The models consists of node biases b_i, which correspond to the a-priori probability of a node i taking the value +1 or -1, and edge weights w_{ij}, which correspond to the likelihood of two neighboring nodes i and j sharing the same value (if w_{ij} is positive), or taking opposite values (if w_{ij} is negative).

We generated Ising models of size $d \in \{8, 12\}$ by setting b_i to a value in the real-valued interval [-1, 1] uniformly at random and i.i.d. for each node i ($b_i \sim \text{Uniform}([-1,1])$, i.i.d.). To generate the w_{ij}'s , we considered three different scenarios. First, we considered a "mixed" case, where the w_{ij}'s could be positive or negative. For each edge (i, j) in the set of edges E, we set $w_{ij} \sim \text{Uniform}([-w,w])$, i.i.d., where w was the maximum weight magnitude. Next, we considered an "attractive" case, where w_{ij} could only be non-negative. That is, $w_{ij} \sim \text{Uniform}([0,w])$, i.i.d.

Finally, we considered a "constant" case, where the values of w_{ij} could only be w or $-w$, but the probability of any w_{ij} being positive was given by some probability q. In these models we independently set each w_{ij}'s value to w with probability q and $-w$ with probability $1-q$.

For all three cases, we considered models generated with different maximum weight magnitudes w. In the "mixed" and "attractive" cases, we generated 50 models for each $w \in \{2.0, 2.5, 3.0, 3.5, 4.0\}$. In total, this gave us 250 "mixed" models and 250 "attractive" models of size d. In the "constant" case, we generated 5 models for each $w \in \{2.0, 2.5, 3.0, 3.5\}$ and for each $q \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. 50 models were generated for $w = 4.0$ and for each value of q. In total this gives us 770 "constant" models of size d.

We evaluated several different approximation algorithms in the context of these synthetic Ising models. One set of algorithms we used is based on the *Multiplicative Weight Update (MWU)* algorithm (*Blum and Mansour*, 2007), which is a type of no-regret algorithm. In our implementation of the MWU algorithm, for each player i at each round $t \geq 1$, we set the probability of playing action x_i at round $t+1$, which we denote by $x_i^{(t+1)}$, to be $p_{t+1}(x_i) \propto p_t(x_i)\left(1 - \eta_t(1 - \bar{M}_i(x_i, x_{-i}^{(t)}))\right)$, where η_t is analogous to a learning rate in ML, and $\bar{M}_i$ is the normalized payoff function for player i. η is set to $\eta_t = \sqrt{\frac{\ln(2)}{t}}$. In this case, the players have no external regret, so we label the algorithm as "mw_er". We also run a variant of mw_er with a constant $\eta = 0.01$, labeled as "mw_er_cf," in which players have approximately no external regret. The minimization of external regret means mw_er has guaranteed convergence to the set of CCE, whereas mw_er_cf converges to the set of approximate CCE. MWU can also be adapted to minimize swap regret instead of external regret. In that case, it would converge to the set of CE or approximate CE, depending on if η is a constant or not. We include "mw_sr" and "mw_sr_cf" implementations, the former using $\eta = \sqrt{\frac{\ln(2)}{t}}$ like above and the latter using a constant $\eta = 0.01$.

We also include another approximate no swap-regret algorithm devised by *Hart and Mas-Colell* (2000), which we label as "nr". Since this algorithm minimizes swap regret, it has guaranteed convergence to the set of approximate CE. Our implementation deviates slightly from the original. Specifically, we evaluate a version in which we update the mixed-strategy each player uses to draw

an action at every iteration t as follows. For each player, (1) we set the probability of switching the player's last action being equal to the empirical regret, or 0 if the empirical regret is negative; and (2) we set the player's probability of playing action $+1$ by "damping" the currently suggested probability of playing $+1$, $p_t(1)$, using the update $0.99 \times p_t(1) + 0.01 \times (0.5)$. This was done as an adaptation to our belief-inference setting.

In addition to the mw-type algorithms and nr, we include standard mean-field approximation ("mf"), standard belief propagation ("bp"), TRW ("trw"), and the Gibbs sampler ("gs"). Finally, we also include our sequential, "semi-stochastic" fictitious play algorithm discussed in Section 3.3. As a simple baseline estimator ("bl"), we simply assign 0.5 as the exact marginal distribution of each variable. We compare the output of these approximation algorithms to that of exact inference (*Kakade et al.*, 2003). Table 4.1 shows the running time (big O) for each algorithm, the maximum number of iterations we chose, as well as other implementation notes.

Algorithm	Run Time O()	Max. Iterations	Notes
exact	$O(2^d)$	N/A	requires $O(2^d)$ memory
bl	$O(1)$	N/A	
mw-type	$O(\|E\|)$	10^5	
nr	$O(\|E\|)$	10^5	
fp (ce)	$O(\|E\| + n \log n)$	15	
mf	$O(\|E\|)$	10^6	sequential axis-parallel updates
bp	$O(\|E\|)$	10^5	simultaneous updates
trw	$O(\|E\|)$	10^5	$\rho = 0.55$
gs	$O(\|E\|)$	10^6	

Table 4.1: Algorithm Properties and Notes

As mentioned earlier, the maximum weight magnitudes we used were all greater than 2.0, and

the maximum bias magnitude was always 1.0. This is because when the weights are closer to 1.0 (or closer to the bias magnitude), the Gibbs sampler easily outperforms the other approximation algorithms. When the weight magnitudes are larger than the bias magnitudes (at least two or three times as large), gs doesnt perform nearly as well, and the other algorithms then become competitive. These kinds of problems appear to be harder, which is why we will focus on them for our experiments.

4.1.2 Experimental Results, Quantitative Evaluation

After running the various algorithms on the generated Ising models, we performed hypothesis testing on the individual differences between the algorithms' output versus the exact inference output. Specifically, we take the absolute difference between the estimated and exact marginal probability of each random variable corresponding to a node in the IM, average across those variables, then average again across all generated models for that case. The tests were done using paired z-tests with p-value 0.05. Here we present results for the 12×12 models which are statistically significant with respect to these hypothesis tests. The 8×8 results were nearly identical in terms of relative performance, so we omit a thorough discussion.

In each type of model ("mixed," "attractive," and "constant"), mw_er_cf consistently performed best among the mw-type algorithms, so we refer to mw_er_cf as simply "mw".

"Mixed" 12×12 **results, (Fig. 4.1).** The figure shows average marginal error across different weight levels w. As shown in the plot, gs clearly performs the best for all w. For the other algorithms, we observed the following:

1. Fp (ce) is worse than bp for $w < 3.5$, and indistinguishable from bp for $w \geq 3.5$.

2. Fp (ce) is consistently better than trw.

3. Trw is consistently worse than bp.

4. Mw is worse than fp (ce) for $w < 3.0$ and indistinguishable from fp (ce) for $w \geq 3.0$.

5. Mf and nr are consistently worse than bl, while all other methods are consistently better than bl.

6. Mf is better than nr for $w \geq 3.0$, but indistinguishable from nr for $w < 3.0$.

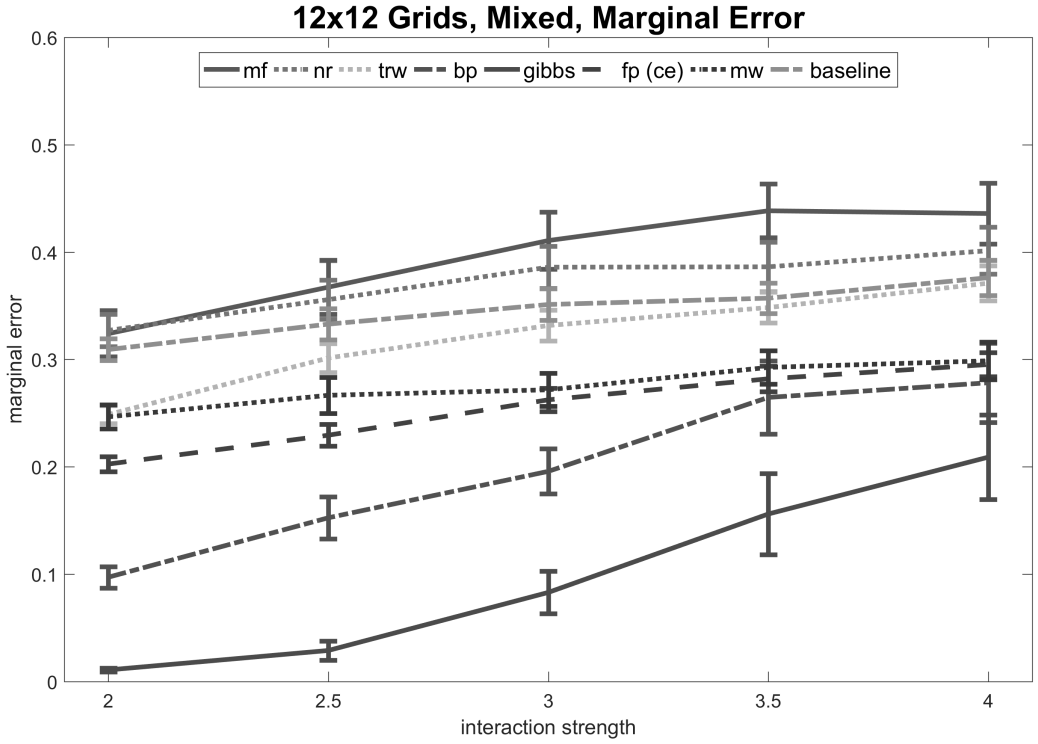


Figure 4.1: **Evaluation on IMs with** 12×12 **Grids, "Mixed" Case**

"Attractive" 12×12 **results, (Fig. 4.2).** In this case there is no clear overall best. We also observe the following:

1. Trw is best among all methods for $w \geq 3.0$, indistinguishable from gs for $w = 2.5$, and worse than gs for $w = 2.0$.

2. Fp (ce) is worse than gs for $w = 2.0$, but better than gs for $w = 4.0$, and indistinguishable from gs otherwise.

3. Mw and fp (ce) are consistently indistinguishable.

4. Mf, nr, and bp are consistently indistinguishable from each other, except for $w = 2.0$ where bp is better than nr.

5. Bp and bl are consistently indistinguishable, except for $w = 4.0$, where bp is better.

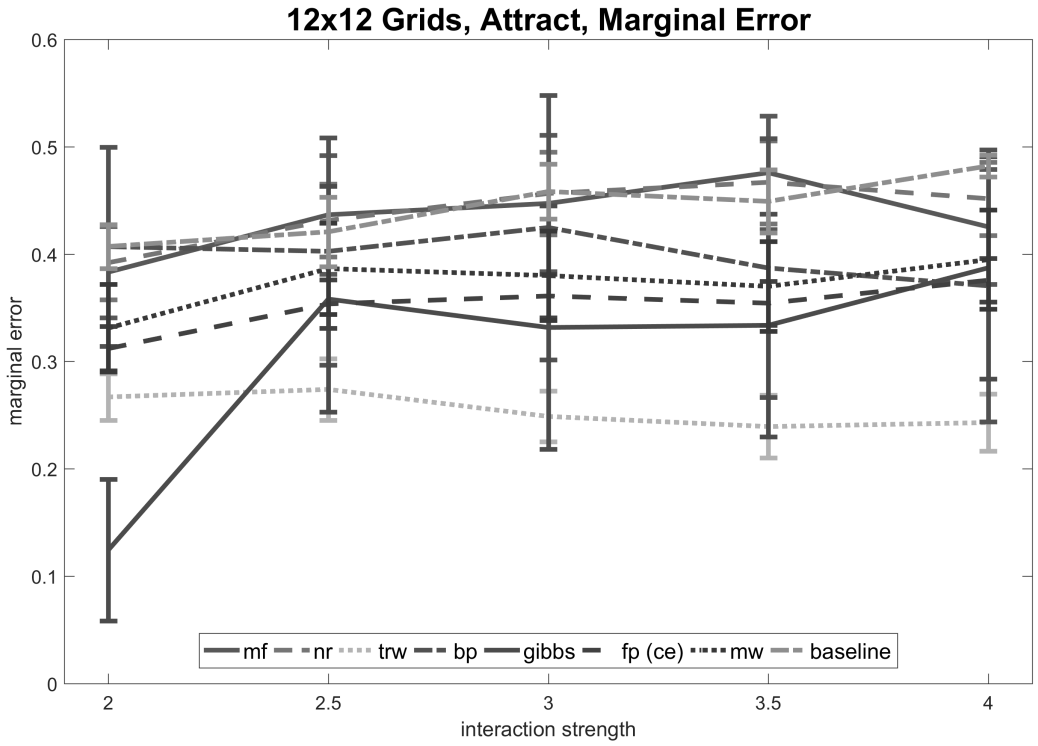


Figure 4.2: **Evaluation on IMs with** 12×12 **Grids, "Attract" Case**

Next, we consider the "constant" Ising model case. This class of models appear to lead to "harder" instances. Since these models are generated in a slightly different way than the others (i.e., we tried different probabilities of attractive interactions q, in addition to different weight magnitudes), we present the results in a slightly different way. First we present a set of "aggregate" results, in which we average the marginal errors across all q's and all models. Then, we present "detailed" results for $w = 4$, in which we show the marginal errors for each q, averaged across all

models at each q.

Earlier we mentioned that for $w = 4$, we generated 50 models as samples for each q, while for $w \leq 3.5$ we generated only 5 models as samples for each q. For $w = 4$, hypothesis testing was done in the same way as in the "mixed" and "attractive" cases. For $w \leq 3.5$, we used 100 bootstrap samples taken from the 5 original samples, then performed the pairwise individual z-tests with p-value 0.05 as in the other cases. The following statements are statistically significant with respect to their corresponding hypothesis tests.

"Constant" 12×12 **aggregate results, (Fig. 4.3).:**

1. Fp (ce) is best among all methods except for when $w = 2.0$, where gs is better.
2. Trw is second best among all methods, except for when $w = 2.0$, where it is third best (behind fp (ce) and gs).
3. Bp is consistently better than mf and nr except when $w = 3.5$, where it is indistinguishable from nr (but still better than mf).
4. Mf is consistently worse than bl, except when $w = 4.0$, where they are indistinguishable. Nr is also consistently worse than bl, except when $w = 2.5$, where they are indistinguishable.
5. Gs is consistently better than mf, nr, and bl, except when $w = 4.0$, where gs and bl are indistinguishable.
6. Mw is better than bl when $w < 3.5$, but indistinguishable from bl when $w >= 3.5$.

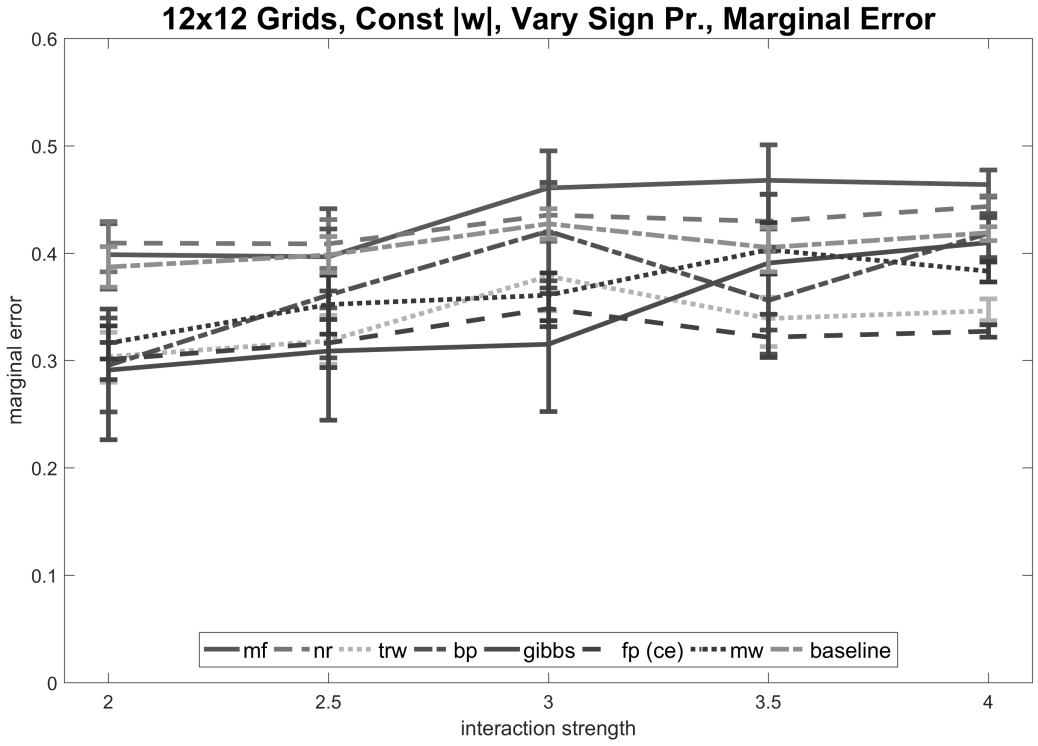


Figure 4.3: **Evaluation on IMs with** 12×12 **Grids, "Constant" Case**

"Constant" 12×12 **detailed results for** $w = 4$**, (Fig. 4.4).:** Most algorithms were unable to outperform bl in this case, remaining statistically indistinguishable from it across all q. Other observations include:

1. Trw, mw, and fp (ce) beat bl at extreme q, i.e $q \in \{0.0, 1.0\}$.

2. Fp (ce) is consistently better than bl.

3. Trw is better than fp (ce) for $q \in \{0.0, 1.0\}$, and worse or indistinguishable otherwise.

4. Mw and fp (ce) are indistinguishable, except for $q \in \{0.0, 1.0\}$, where fp (ce) is better.

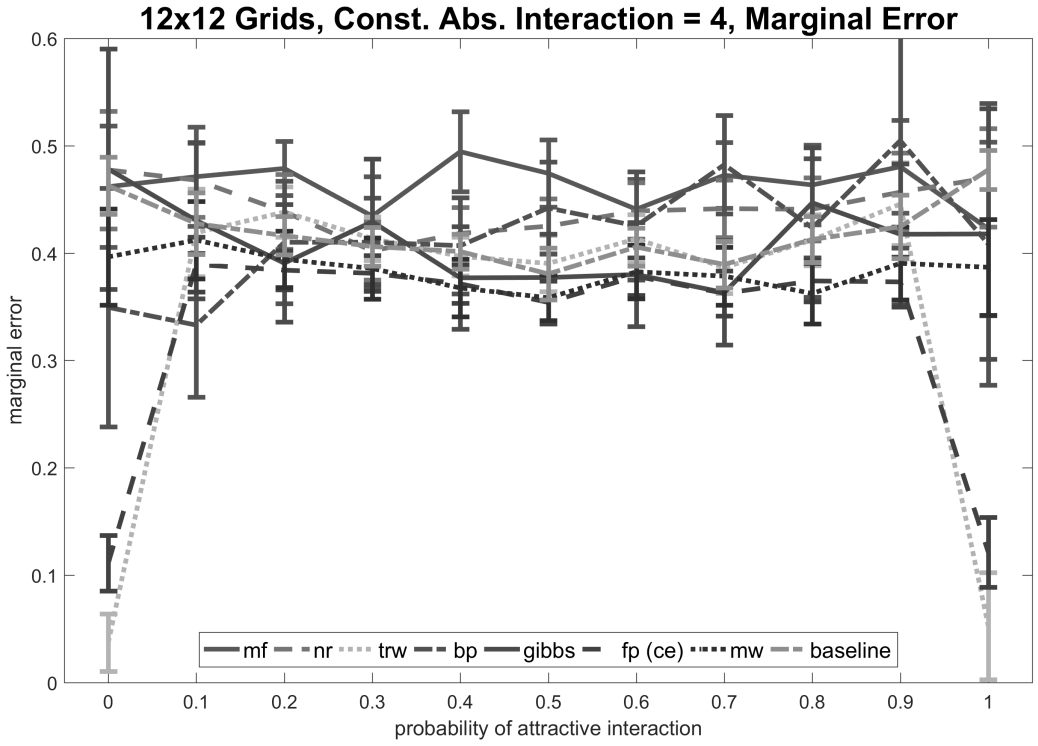


Figure 4.4: **Evaluation on IMs with** 12×12 **Grids, "Constant" Case, Edge Weight Magnitude** $w = 4.0$**, Varied Probability of Attractive Interactions**

Results for MWU algorithm variants. Fig 4.5 compares the performance of the four types of MWU algorithms described earlier. It is evident from the plots (and reinforced by the same hypothesis testing done for the other methods) that mw_er_cf consistently does the best, or at least no worse than the other variants. This is somewhat unexpected, since the external regret versions of MWU converge to the set of CCE, whereas the swap regret versions converge to CE, which is a smaller and less relaxed set of equilibria. One possible explanation is that the swap regret variants are simply better at avoiding "bad" local minima, but are much slower to converge overall. The qualitative visual results appear to support this idea, as the swap regret variants' results roughly match the results of exact inference, albeit with much greater uncertainty.

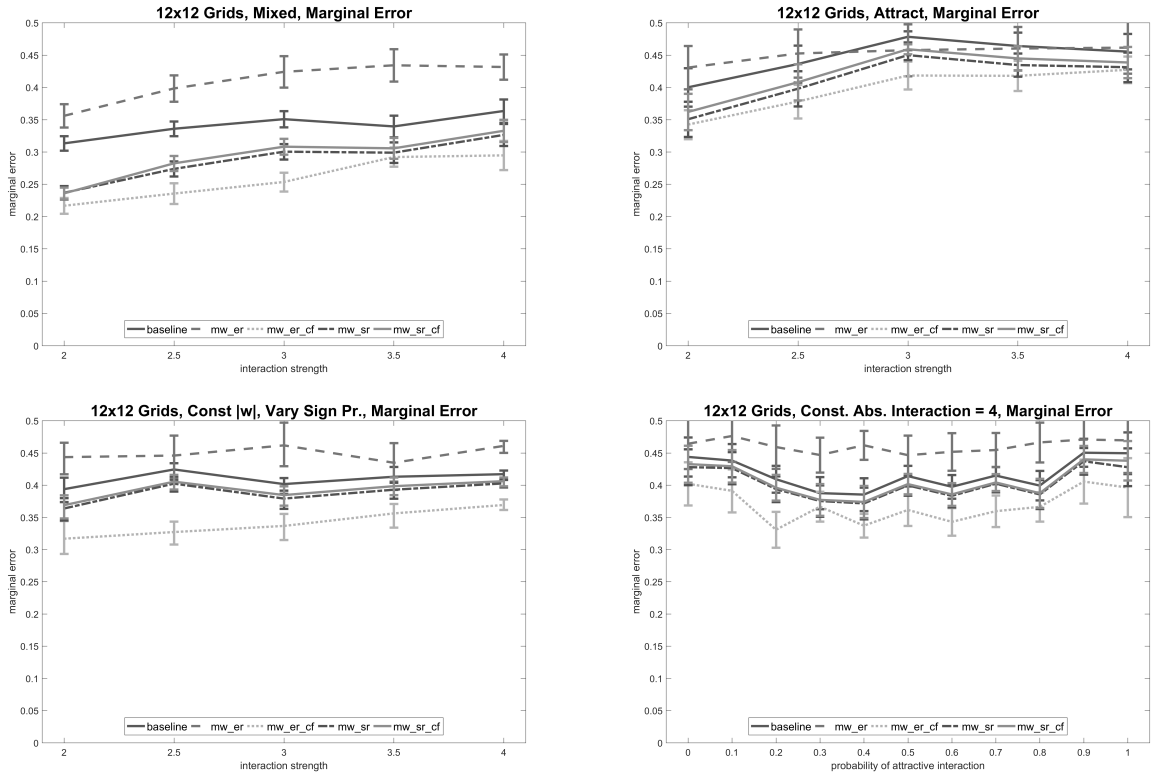


Figure 4.5: **Evaluation on IMs with** 12×12 **Grids, MWU Variants**

Results of varying the number of maximum iterations. We tried running our proposed fp (ce) algorithm with different numbers of iterations. We found that increasing the number of iterations from 15 to 50 (and greater) only leads to minimal improvement in average marginal error, as shown in Fig 4.6. The average marginal error is shown as a line, and is obtained from 20 randomly generated Ising models and corresponding estimates. The marginal error of each individual model is represented by a circle on the graph. Each marginal error is the result of averaging over all values of the probability of attractive interaction $q \in \{0.0, 0.1, \ldots, 0.9, 1.0\}$, like what was done for the "constant" aggregate results.

The plot also shows that each run of fp (ce) on different models results in a consistent level of error. That is, the variance of the error is low. Compare this with a similar experiment on trw and gs, which exhibit greater error variance, even when the number of iterations is much higher. This

suggests that fp (ce) converges to an estimate in a lower number of iterations than gs and trw, and does so with better consistency.

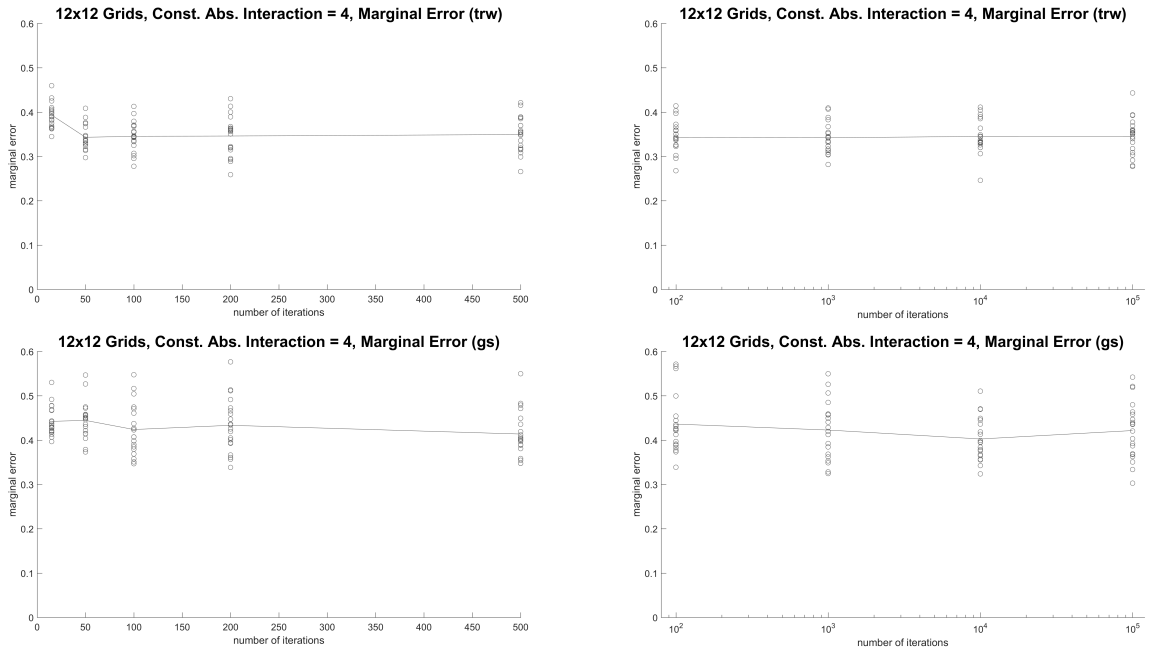


Figure 4.6: **Evaluation on IMs with** 12×12 **Grids, "Constant" Case, Uniform Interaction Magnitude** ($w = 4.0$)**: Marginal Error by Number of Iterations**

4.1.3 Experimental Results, Qualitative Evaluation

In addition to the results discussed above, we also produced visualizations of each algorithm's output for visual comparison. This helped us see if certain algorithms which performed poorly in the quantitative evaluation were at least producing estimates which seem reasonable or not. The visual representations also highlight how the structure of the "mixed," "attractive," and "constant" models differ from each other in certain situations.

To build the visualization, we simply used the final estimated marginal probabilities produced by each algorithm and created a corresponding gray-scale $d \times d$ image. Each pixel in the image represents one node variable in the original Ising model. Whether a pixel is white, black, or gray is determined by the estimated marginal probability of the corresponding node's assignment. White

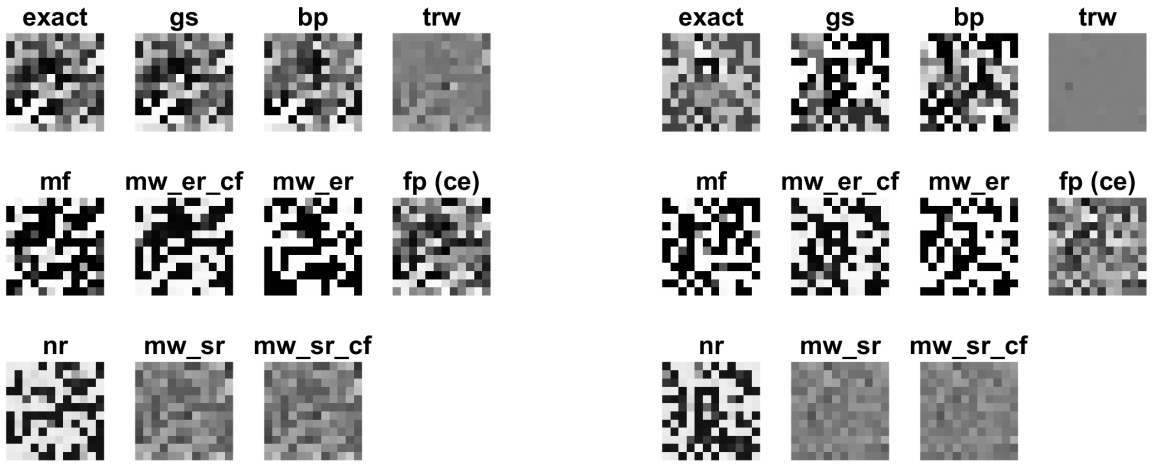
pixels represent nodes whose estimated probability of being assigned "+1" is 1.0, while black pixels represent nodes with estimated probability 0.0. Gray-scale pixels represent non-extreme node probabilities of varying magnitude, depending on how light or dark the pixel is.

Note that the visualizations only show the result of *one* of the models generated for each case. We did not perform any sort of aggregation (such as averaging) to try to represent every model's results, like what was done in the quantitative evaluation. Therefore, we cannot say that the visualizations shown in each case are truly representative of the behavior of each algorithm overall.

"Mixed" 12×12 results (Fig. 4.7). Each algorithm used in the quantitative evaluation is included in the visualization, with the exception of baseline. This is because bl would simply show up as a completely gray image, since every node's estimated probability is 0.5. We can see from the exact inference result that these "mixed" models lead to chaotic assignments without any sort of discernible pattern. This makes intuitive sense, because "mixed" models have edge weights which are equally likely to be positive or negative. Positive edge weights cause neighboring nodes to be more likely to be the same, whereas negative edge weights make them more likely to be different. There do not appear to be many general differences between the $w = 2.0$ and $w = 4.0$ models, at least based on the exact inference output.

We can see that the output of gs almost exactly mirrors that of exact inference in the $w = 2.0$ case. This is not too surprising, given that gs performed very well in the "mixed" case at lower values of w. On the other hand, in the "harder" $w = 4.0$ case, the gs estimation was not quite as accurate, though from a qualitative standpoint it still appears to be fairly good. Compare that to the results of trw, which in the $w = 2.0$ case gives a reasonably accurate estimation, but in the $w = 4.0$ case does not produce a useful estimate at all. The resulting estimation puts nearly all nodes at about 0.5 probability of positive assignment, putting this estimation more or less on par with our simple baseline estimator. Consider the results of mf, which in the quantitative evaluation we found to be "consistently worse than bl". However, even this method appears to be better in a qualitative sense than trw, especially in the $w = 4.0$ case, where mf at least estimates some portions of the

distribution correctly. We can also see that certain methods, specifically mf, mw_er, mw_er_cf, and nr, seem to converge to mostly pure strategies. That is, the resulting node probabilities are almost always very close to zero or one. This is a trend that we observe in other cases as well.



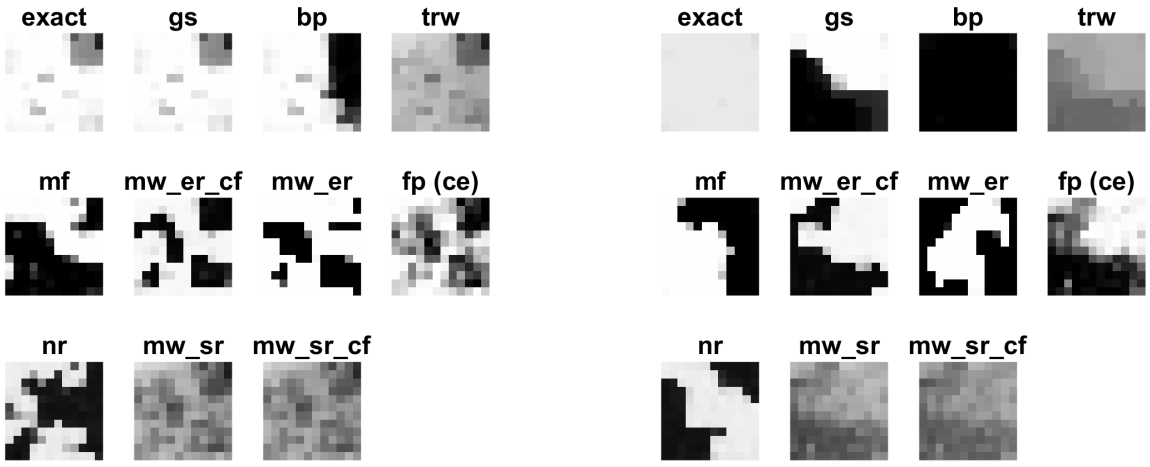
$w = 2.0$ $w = 4.0$

Figure 4.7: **Visual Representation of Two Example IMs with** 12×12 **Grids, "Mixed" Case, Edge Weight Magnitude** $w \in \{2.0, 4.0\}$

"Attractive" 12×12 **results (Fig. 4.8).** Exact inference results in a fairly homogeneous assignment, especially in the higher weight case. This makes sense because every edge weight is positive, meaning nodes will generally take the same value as their neighbors. This leads to a more "structured" appearance than the "mixed" models.

For $w = 2.0$, gs once again performs very well, finding a distribution which appears identical to exact inference. The other algorithms do not perform quite as well, but most of them are able to get some elements of the distribution correct. For example, bp's estimate of the left half of the model appears correct, while trw, mw_er_cf, and fp (ce) found the dark box in the upper right corner. Like the high weight "mixed" models, the high weight "attractive" models appeared to lead to harder problems, as evidenced by the fact that *none* of the evaluated algorithms produced a good

estimation. It is interesting how different algorithms seem to produce similar (but not necessarily good) results. For example, in the $w = 4.0$ case, gs, trw, mw_er_cf, fp (ce), and even mw_sr and mw_sr_cf all have roughly similar output, with the bottom left being dark and the upper right being light. These methods also produce roughly similar estimates in the $w = 2.0$ case, though only gs, and trw to a lesser extent, closely match exact inference.



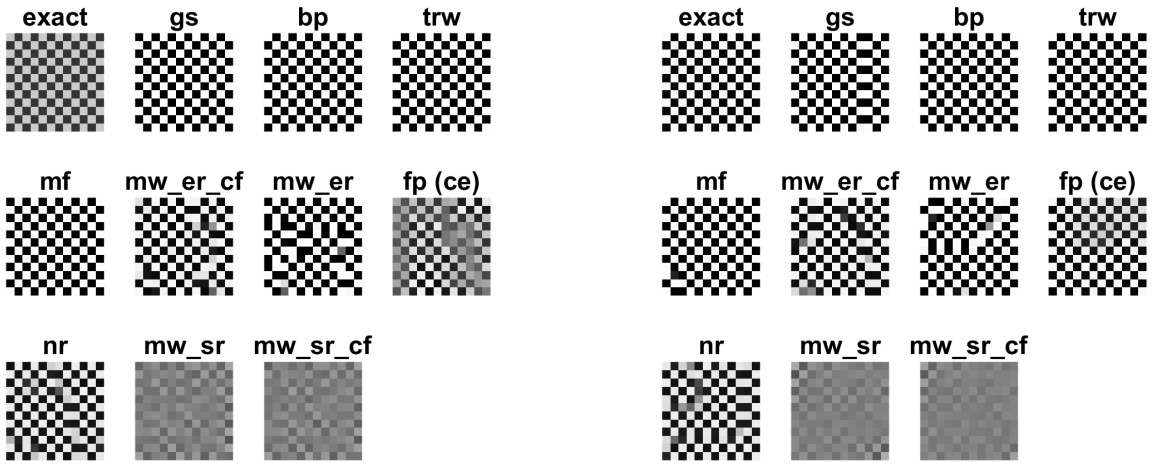
$w = 2.0$ $w = 4.0$

Figure 4.8: **Visual Representation of Two Example IMs with** 12×12 **Grids, "Attract" Case, Edge Weight Magnitude** $w \in \{2.0, 4.0\}$

"Constant" 12×12 **results (Fig. 4.9, Fig. 4.10, Fig. 4.11).** Since the "constant" models were built with different probabilities of attractive interaction q, we include one visualization each for $q = 0.0$, $q = 0.5$, and $q = 1.0$. First we will focus on the $q = 0.0$ models (Fig. 4.9). We can see that these models exhibit a very distinctive "checkerboard" pattern in the inference results. This is caused by the fact that when $q = 0.0$, every edge weight is negative. Thus, every node is likely to have a different assignment than its neighbors. In addition, because all the edge weights are the same constant $-w$, any two neighboring nodes are just as likely to be different than any other two neighbors. Note that because every edge weight is identical, the node biases are the deciding

factor between whether the checkerboard is "black-white-black" or "white-black-white," starting from the upper left corner. Note also that the $w = 2.0$ checkerboard is "fuzzier" than the $w = 4.0$ checkerboard, that is, the node probabilities are not as close to zero or one. This is simply because the lower weight model has a weaker negative interaction between neighbors, so the probability that the neighbors are actually different is not as high as in the higher weight model.

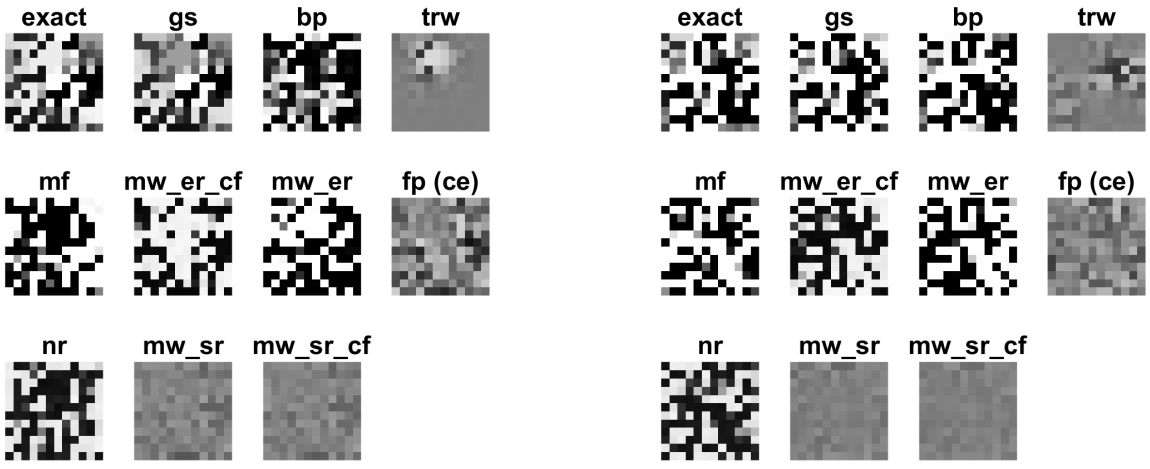
All of the approximation algorithms were able to find at least a "checkerboard-like" strategy. However, simply having a checkerboard strategy does not mean the estimation is accurate. For instance, in the $w = 4.0$ case, trw matches exact inference perfectly, but bp has every node assigned incorrectly. Recall that in the quantitative evaluation, trw and fp (ce) performed very well in the extreme q models. This implies that these two algorithms are able to use information from the node biases effectively in cases where the edge weights are identical, while the other algorithms cannot. On a related note, the mw_sr and mw_sr_cf methods produce what appear to be "fuzzy checkerboards". Although a checkerboard pattern exists, it is "fuzzified" across all nodes seemingly at random. This can be explained by the fact that node biases are still generated uniformly at random, even though the edge weights are not in this case. This tells us that the mw_sr and mw_sr_cf algorithms are strongly affected by node biases even when the edge weight magnitudes are much greater than the node bias magnitudes. We will observe this phenomenon again in Section 4.2.3, when we look at similar experiments done on MNIST-derived Ising models.



$w = 2.0$ $w = 4.0$

Figure 4.9: **Visual Representation of Two Example IMs with** 12×12 **Grids, "Constant" Case with** $q = 0.0$**, Edge Weight Magnitude** $w \in \{2.0, 4.0\}$

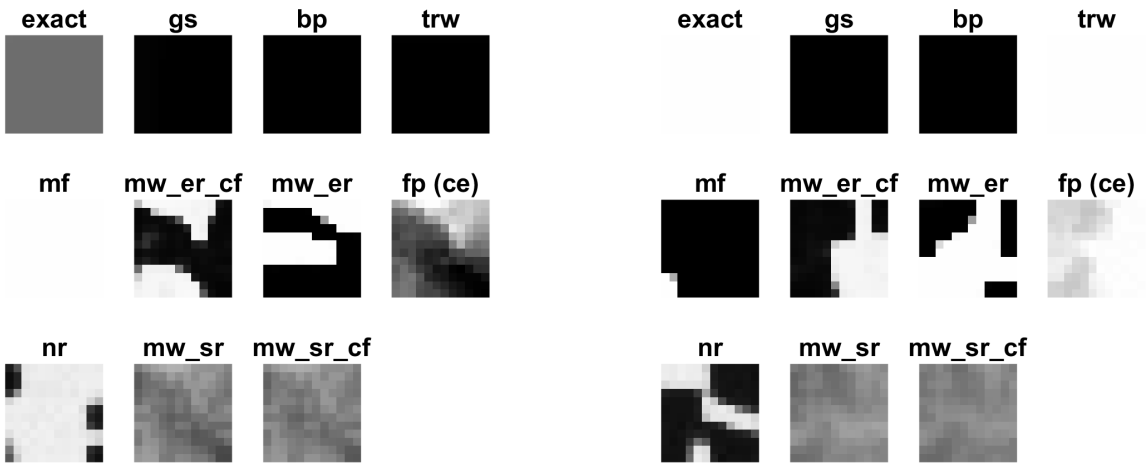
Next is the $q = 0.5$ case (Fig. 4.10). Based on the exact inference results, these models appear similar to the "mixed" case models discussed earlier. In the low weight case, every algorithm other than trw was able to find a solution which at least bear some semblance to exact inference. The output of trw is especially interesting. We can see that it is able to perform the inference well in a small region, while the rest of the model remains quite fuzzy, with node probabilities near 0.5. Perhaps given many more iterations, it would eventually converge to a good estimate. However, it turns out that about half the time in non-extreme "constant" cases, trw simply finds a local minimum where *every* node probability is approximately 0.5, and stops early. That is, about half the time trw performs approximately as well as a simple baseline estimator. This highlights how these "constant" models can be tricky to solve even for state-of-the-art algorithms.



$w = 2.0$ $w = 4.0$

Figure 4.10: **Visual Representation of Two Example IMs with** 12×12 **Grids, "Constant" Case with** $q = 0.5$**, Edge Weight Magnitude** $w \in \{2.0, 4.0\}$

Finally, we have the $q = 1.0$ case (Fig. 4.11). In these models, every edge weight is positive with value w. Thus it makes sense that, in contrast to the $q = 0.0$ "checkerboard" where every neighbor is different, here a "sheet" is formed where all neighbors (and thus all nodes) have the same assignment. Like when $q = 0.0$, the $w = 2.0$ case has a fuzzier exact inference result, leading to either a light gray or dark gray assignment rather than white or black. From the quantitative evaluation, we know that trw and fp (ce) should perform very well in this extreme q setting. While trw outputs all-black or all-white, fp (ce) produces mostly-black or mostly-white due to influence from the node biases. It is worth noting that mw_sr and mw_sr_cf also visibly contain the same node bias influence that fp (ce) does. Despite being able to find checkerboard assignments when $q = 0.0$, mw_er, mw_er_cf, and nr do not seem to be able to find a simple all-white or all-black assignment when $q = 1.0$.



$w = 2.0$ $w = 4.0$

Figure 4.11: **Visual Representation of Two Example IMs with** 12×12 **Grids, "Constant" Case with** $q = 1.0$**, Edge Weight Magnitude** $w \in \{2.0, 4.0\}$

4.2 MNIST-based Models

After conducting our empirical evaluation on synthetically generated Ising models, we investigated using Ising models created in a more "realistic" context. To that end, we used the popular MNIST image dataset (*LeCun et al.*, 1998) to build Ising models for the purpose of soft de-noising. It is important to note that our primary interest in these experiments is not classifying the images, or performing a MAP estimation. Instead, we are interested only in belief inference; that is, using the individual marginal probabilities as a confidence measure of the individual pixel values of the de-noised image.

4.2.1 Experimental Design

The MNIST images are 28×28 grayscale pixel images of handwritten digits. Therefore, we construct 28×28 simple planar Ising models, again consisting of node biases b_i and edge weights w_{ij}, just like in the synthetic experiments. Instead of randomly generating the bias and edge values,

we use the following process. First, we convert the grayscale pixels to black (-1) and white (+1) using a threshold of 0.5. Next, for each pixel, we find the average value of that pixel across all training images, which is used as the node bias. Finally, we compute the average product between neighboring pixels, taken across all training images, and use that average product as the edge weight between those neighboring pixels. Said more precisely: Denote I_l as the matrix representation of the I-th image in the MNIST training dataset, and denote by m the number of images in that training dataset. The node bias $b_{(i,j)}$ for node or pixel (i, j) is given by:

$$b_{(i,j)} \propto \frac{1}{m} \sum_{l=1}^{m} I_l(i, j)$$

The edge weights between node (i, j) and neighboring node $(i, j+1)$, for example, is given by:

$$w_{(i,j),(i,j+1)} \propto \frac{1}{m} \sum_{l=1}^{m} I_l(i, j+1)\, I_l(i, j)$$

We apply a normalization factor to both the weights and biases such that $\max_{(i,j)} |b_{(i,j)}| = 1$, in order to more closely replicate the environment of the synthetic experiments, where the maximum node bias magnitude was also 1.0. It is important to note that these values are calculated from all training images, regardless of label. Thus we are building models of "handwritten digits" in general rather than "handwritten 1," "handwritten 2," and so forth.

After building these node biases and edge weights, we prepare test images for evaluation. We select uniformly at random 100 images from the MNIST test dataset, and convert them to black/white images using the same thresholding method as above. Then, we add noise to the black/white image by independently flipping each pixel value with some probability p (the "noise level"). Then, we modify our existing node biases for each noisy image by taking into account our "observation" of the noisy image. If I denotes the matrix representation of the noisy black/white test image, the Gibbs potential of that image's Ising model is

$$\Psi_I(x) \equiv \sum_{((i,j),(r,s))\in E} w_{(i,j),(r,s)} x_{(i,j)} x_{(r,s)} + \sum (i,j) \widetilde{b}_{(i,j)}(I(i,j))\, x_{(i,j)}$$

where

$$\widetilde{b}_{(i,j)}(I(i,j)) \propto b_{(i,j)} + \frac{1}{2} I(i,j) \ln \frac{1-p}{p}$$

We re-normalize such that $\max_{(i,j)} |\widetilde{b}_{(i,j)}| = 1$. Note that each model has different node biases based on the observation of their associated noisy image, but have identical edge weight values.

We ran the exact same algorithms on the MNIST-based Ising models as in the synthetic experiments, with one major exception. It was not possible to run exact inference on the MNIST image models, since the associated Ising models were 28×28 instead of 12×12 like in the synthetic experiments and thus had exponentially greater memory requirements. We also used a slightly different number of maximum iterations for most of the other algorithms: 10^4 for bp, nr, and mw (all variants), 10^5 for trw and gs, and 10^6 for mf. fp (ce) remained at 15 iterations.

4.2.2 Experimental Results, Quantitative Evaluation

Fig. 4.12 shows the results of our experimental quantitative evaluation on MNIST-based Ising models, using a noise level $p = 0.05$. Like in the synthetic experiments, we performed hypothesis testing on the results using paired z-tests on the individual differences, each with p-value 0.05. Since we do not have the results of exact inference to compare differences to, we instead use gs as a stand-in for exact inference. The MNIST-based Ising models have low maximum edge weight magnitude compared to the node bias magnitudes, and as we mentioned earlier, the Gibbs sampler easily outperforms all other approximation algorithms in this type of situation.

The computed edge weights for our MNIST-based Ising models were almost all positive, meaning the soft de-noising problem closely corresponded to the "attractive" case in the synthetic experiments. However, because the edge weight magnitudes were quite low as we mentioned earlier, these instances were much "easier" to solve than the synthetic instances. This is evidenced by the plot, since it is clear that every approximation algorithm performs much better than baseline. Compare this result to the attractive synthetic experiments, where many algorithms could not consistently beat the simple baseline estimator. One result which stands out in particular is mf, as it was often indistinguishable from bl in the synthetic experiments, but easily outperforms bl in this

experiment. In order, the statistically best performing algorithms were: 1) bp, 2) trw, 3) mf, 4) mw_er, 5) mw_er_cf, 6) fp, 7) nr, 8) mw_sr, 9) mw_sr_cf, and 10) bl.

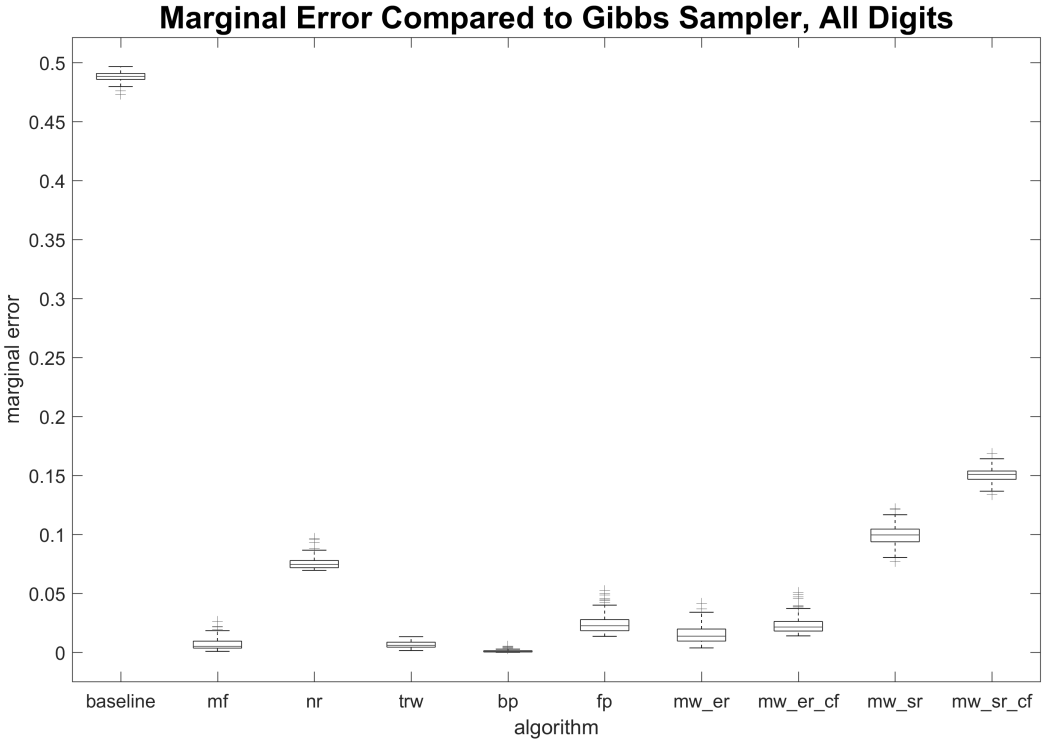


Figure 4.12: **Evaluation on IMs Derived from MNIST Images,** 28×28 **Grids**

Our models were originally built using examples of all possible types of handwritten digits. However, we also tried building models using only a specific digit. We computed edge weights and biases using only images in the MNIST training dataset with a label of "1," and observed only image samples from the test dataset with a label of "1". The results of the handwritten "1" experiment are shown in Fig. 4.13. Although the algorithms retain their relative order to each other when run on "only 1's" versus "all" digits, the range of average marginal errors they achieved tightened.

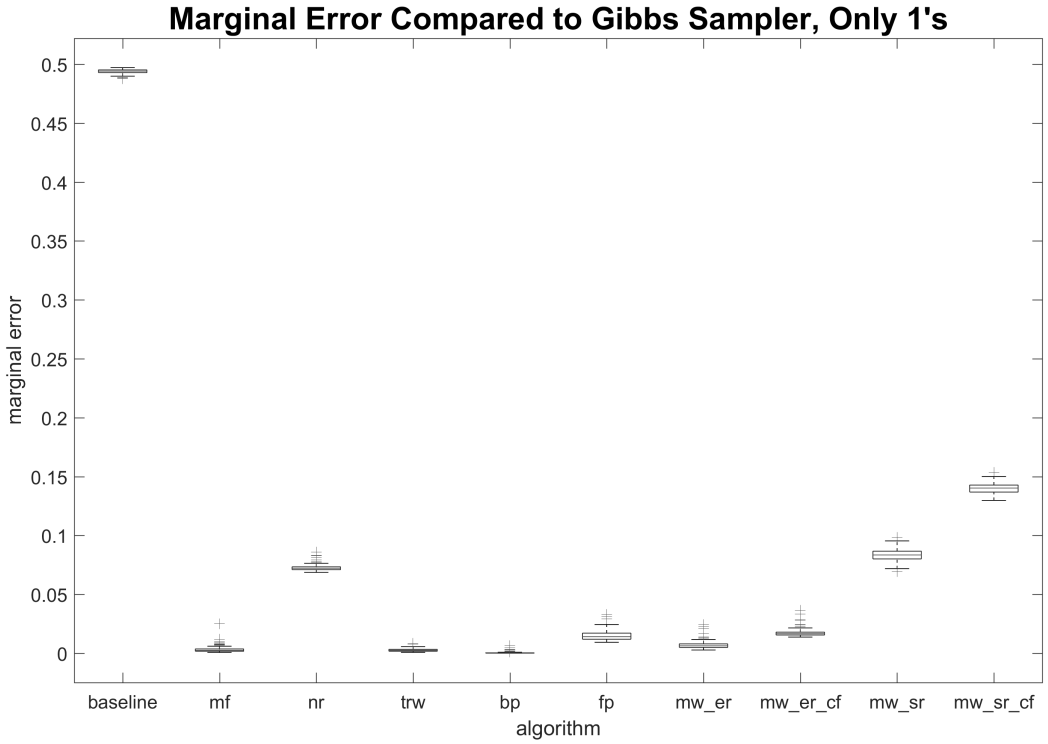


Figure 4.13: **Evaluation on IMs Derived from MNIST Images,** 28×28 **Grids**

4.2.3 Experimental Results, Qualitative Evaluation

We performed additional preliminary experiments with the intent of making the MNIST-based Ising models "harder," by increasing the magnitude of the edge weights w_{ij}, while keeping the node biases b_i low. The most straightforward way to accomplish this was to simply increase the noise level p. Increasing p will lead to $\widetilde{b}_{(i,j)}$ decreasing, which in turn decreases the normalization factor, which makes w_{ij} larger. As we know from the synthetic experiments, once the w_{ij} magnitudes increase to values above 2.0 or so, the Gibbs sampler no longer performs exceptionally well. Therefore we cannot necessarily use Gibbs sampler as a proxy for exact inference in the higher noise settings, which means we do not have an appropriate way to quantitatively measure the quality of a result.

This limitation was our main motivator for creating visual representations of the results for

inspection. The algorithms' output at noise levels $p \in \{0.05, 0.10, 0.15, 0.25\}$ are shown in Figure 4.14. The grayscale images show the corresponding marginal probability estimate for each individual pixel/node in the Ising model. Note that while we do include the original handwritten digit, it is not fair to compare the results directly to that original image. We are interested in probabilistic inference, while the original image is akin to a MAP assignment.

The noise level $p = 0.05$ results are comparable to the quantitative evaluation in the previous section. We can see that all methods output estimates which are close to gs, our "ground truth" algorithm in this case. We can say that the inference done by gs does appear to be good quality, since if we were to build a MAP assignment based on the inference we would recover the original digit with very good accuracy. However, as the noise level increases, the inference done by gs becomes less and less good. The quality of bp, mf, nr, mw_er, and mw_er_cf deteriorate as well. At noise level $p = 0.25$, the image becomes obscured to a point where even a human may not be able to confidently recover the original digit. Only trw, fp (ce), mw_sr, and mw_sr_cf maintain a reasonable quality of estimation at this noise level.

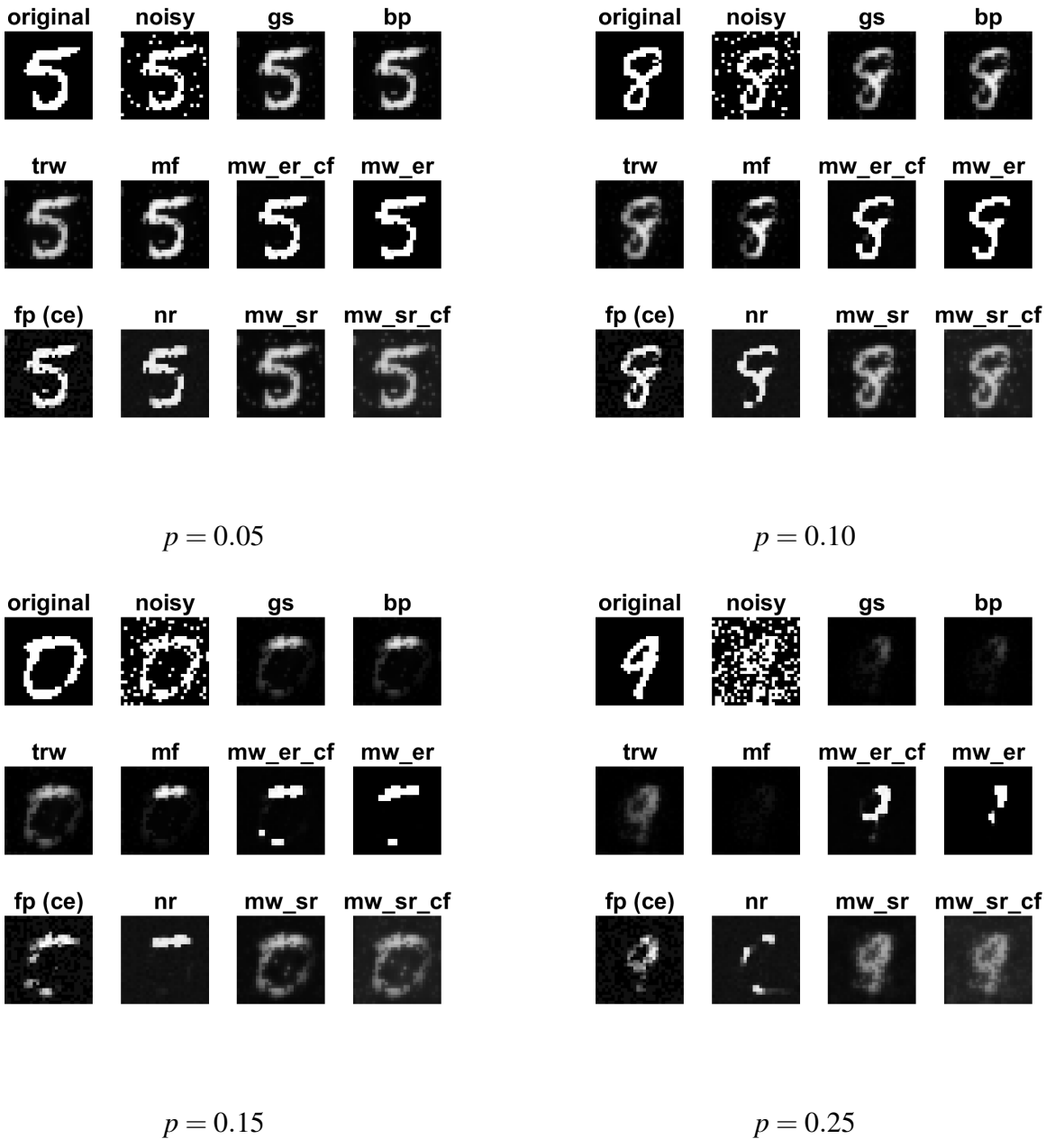


Figure 4.14: **Visual Representation of MNIST-based IMs with Increasing Noise Levels**

Upon inspection of the output files, we found that the weight magnitudes w_{ij} ranged between 0.45 and 0.70, depending on the noise level. To highlight how these edge weight magnitudes can affect the quality of estimation, we simply ran the experiment again, but with all edge weights multiplied by a constant factor 2.0. The results of this experiment are shown in Figure 4.15.

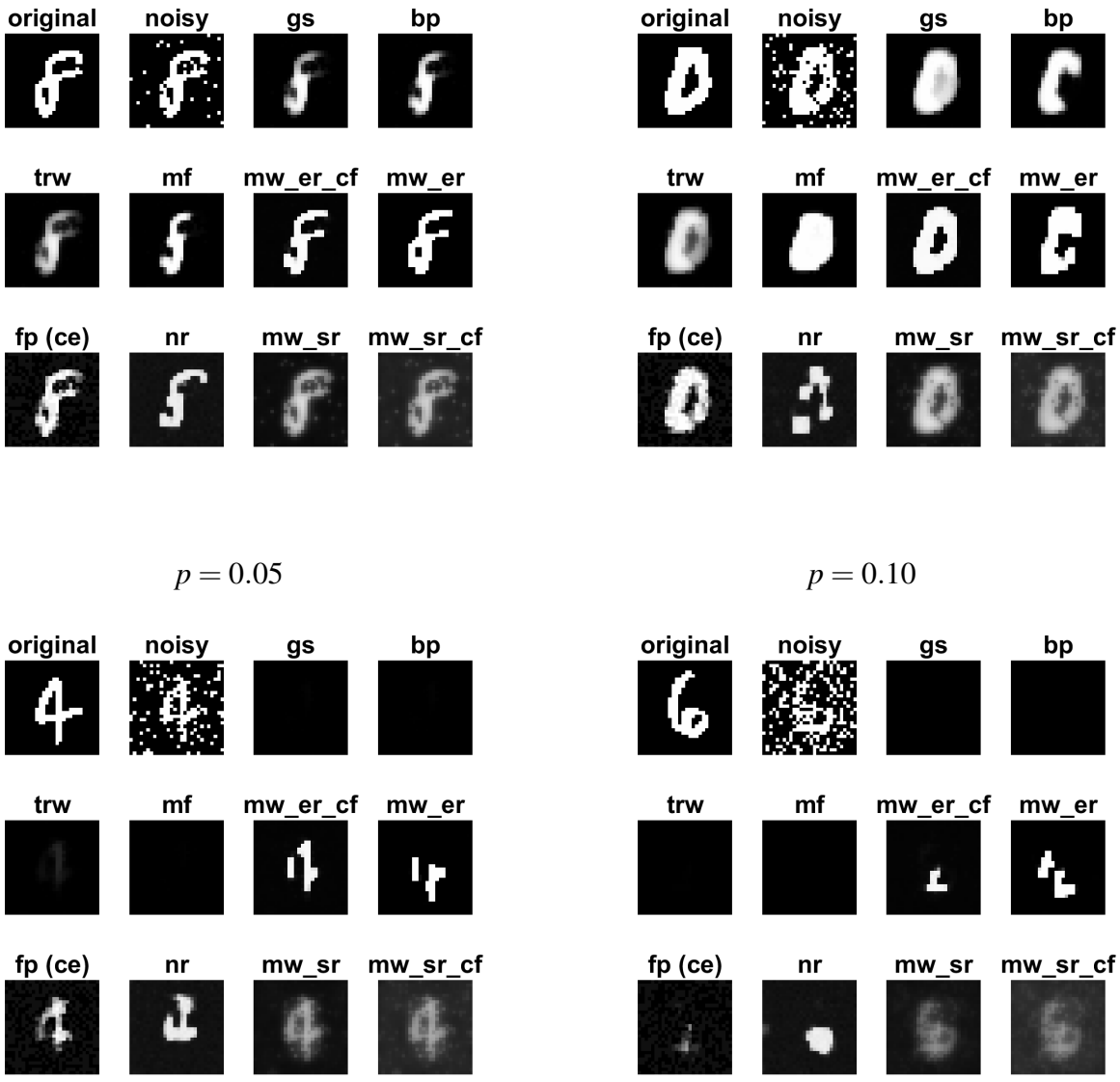


Figure 4.15: **Visual Representation of MNIST-based IMs with Increasing Noise Levels, Doubled Edge Weights**

Now with the doubled edge weights, we observe the following. Like before, the estimates appear reasonably good at lower noise levels, but get worse as noise level increases. At noise level $p = 0.15$, however, we notice some discrepancies with the previous experiment. Here, gs,

bp, and mf were unable to output anything resembling the original digit, unlike before. In the $p = 0.25$ case, the *only* algorithms able to produce a reasonable qualitative estimation were mw_sr and mw_sr_cf. The other methods either produced nothing (all black), or shapes which do not appear to strongly correlate with the original digit.

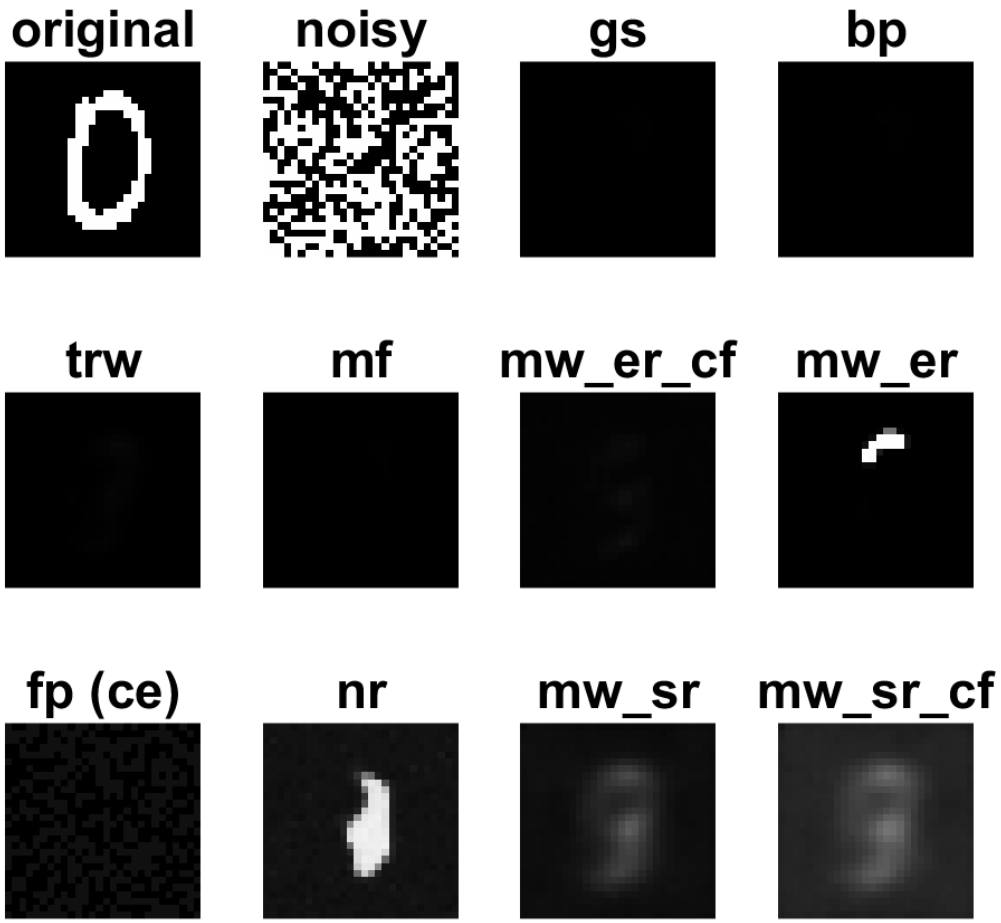


Figure 4.16: **Visual Representation of MNIST-based IMs at Noise Level** $p = 0.50$

In Section 4.1.3 we observed that mw_sr and mw_sr_cf may be more strongly influenced by node biases than other methods. This property could explain why only those two methods were

able to consistently output good estimations, since in this MNIST setting, the *only* thing which differentiates two image examples are the node biases. We can highlight the influence of the node bias by running an experiment where the noise level is $p = 0.50$. Under these conditions, the entire image becomes entirely noise. This makes the observed node bias $\widetilde{b}_{(i,j)}(I(i,j))$ equivalent to $b_{(i,j)}$, since the difference $\frac{1}{2}I(i,j)\ln\frac{1-p}{p}$ becomes zero when $p = 0.50$. In Figure 4.16 we can see that mw_sr and mw_sr_cf still output a very fuzzy shape, despite the input being just noise. In fact, those two methods always output the same fuzzy shape in every model at $p = 0.50$, which makes sense since $b_{(i,j)}$ is the same across all images. One can think of the fuzzy shape as an average MNIST handwritten digit.

The effect of the edge weight magnitudes was much more apparent in the MNIST setting than in the synthetic setting. Once the edge weights became large enough, almost all methods were unable to recover anything about the original digit. There is a simple explanation for this phenomenon, however. We mentioned that the MNIST-inspired models have nearly all positive, or attractive edge weights. As the edge weights become larger, each pixel is more strongly encouraged to be the same as its neighbors. Handwritten digit images typically contain more black pixels than white pixels, especially along the outer edges. In other words, most nodes/pixels are already biased to be black or at least partially black. This bias towards black, coupled with the strong attractiveness of the edge weights, means that *all* the nodes/pixels are estimated as very likely to be black, thus resulting in all-black output. Remarkably, only mw_sr and mw_sr_cf did not appear to be strongly affected by the edge weight magnitudes. In fact, despite performing worse than other methods according to the quantitative evaluation, mw_sr and mw_sr_cf consistently performed the best in "harder" MNIST-based instances, qualitatively speaking.

Chapter V: Conclusions

In this thesis, we provide general formulations of the problem of belief inference in MRFs as equilibrium computation in graphical potential games, as well as some immediate algorithmic, computational, and theoretical implications derived from this connection. We propose several game-theoretic inspired methods which can be applied to the problem of belief inference in PGMs. We evaluate the effectiveness of these proposed algorithms in the context of Ising models with grid graphs, in both synthetic randomly generated problems as well as problems derived from a real-world data set. We show that many methods, even state-of-the-art algorithms from PGM literature, are often not better than a simple baseline estimation (that all marginal probabilities are equal to 0.5). We show that certain classes of synthetic Ising models lead to "harder" instances where our proposed game-theoretic methods compete with and even beat state-of-the-art algorithms like TRW. The subsequent MNIST-based experiments highlight how these game-theoretic methods can excel even in a "real-world" image de-noising application. We encountered "harder" instances of MNIST-derived Ising models where a game-theoretic method based on the Multiplicative Weight Update algorithm clearly outperformed all other evaluated algorithms. Although we cannot claim that game-theoretic inspired algorithms excel in every situation, this promising result gives us confidence that there are other practical applications for these methods waiting to be discovered.

In closing, our goal is that the work presented in this paper can establish sufficient precedent for research in formulating probabilistic inference problems as problems of equilibrium computation. We believe that the synergy between equilibrium computation and belief inference can potentially lead to many new and interesting discoveries in both computational game theory and probabilistic graphical models.

BIBLIOGRAPHY

Aumann, R. (1974), Subjectivity and correlation in randomized strategies, *Journal of Mathematical Economics*, *1*.

Besag, J. (1974), Spatial interaction and the statistical analysis of lattice systems, *Journal of the Royal Statistical Society. Series B (Methodological)*, *36*(2), 192–236.

Blum, A., and Y. Mansour (2007), Learning, regret minimization, and equilibria, in *Algorithmic Game Theory*, edited by N. Nisan, T. Roughgarden, Éva Tardos, and V. V. Vazirani, chap. 4, pp. 79–102, Cambridge University Press.

Cooper, G. F. (1990), The computational complexity of probabilistic inference using Bayesian belief networks (research note), *Artif. Intell.*, *42*(2-3), 393–405.

Cover, T. M., and J. A. Thomas (2006), *Elements of Information Theory*, second ed., Wiley & Sons, New York.

Daskalakis, C., and C. H. Papadimitriou (2006), Computing pure Nash equilibria in graphical games via Markov random fields, in *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pp. 91–99, ACM, New York, NY, USA, doi: http://doi.acm.org/10.1145/1134707.1134718.

Fudenberg, D., and D. Levine (1999), *The Theory of Learning in Games*, MIT Press.

Gottlob, G., G. Greco, and F. Scarcello (2003), Pure Nash equilibria: Hard and easy games, in *TARK '03: Proceedings of the 9th conference on Theoretical aspects of rationality and knowledge*, pp. 215–230, ACM, New York, NY, USA, doi:http://doi.acm.org/10.1145/846241.846269.

Hammersley, J., and P. Clifford (1971), Markov fields on finite graphs and lattices, unpublished.

Hannan, J. (1957), Approximation to Bayes risk in repeated play, in *Contributions to the Theory of Games*, vol. III, edited by M. Dresher, A. W. Tucker, and P. Wolfe, pp. 97–140, Princeton University Press.

Hart, S., and A. Mas-Colell (2000), A simple adaptive procedure leading to correlated equilibrium, *Econometrica*, *68*(5), 1127 – 1150.

Hofbauer, J., and W. H. Sandholm (2002), On the global convergence of stochastic fictitious play, *Econometrica*, *70*(6), 2265–2294.

Istrail, S. (2000), Statistical mechanics, three-dimensionality and NP-completeness: I. universality of intracatability for the partition function of the Ising model across non-planar surfaces (extended abstract), in *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pp. 87–96, ACM, New York, NY, USA, doi: http://doi.acm.org/10.1145/335305.335316.

Jaakkola, T. S. (2000), Tutorial on variational approximation methods, in *Advanced Mean Field Methods: Theory and Practice*, edited by M. Opper and D. Saad, pp. 129–159, MIT Press, Cambridge, MA.

Jiang, A. X., and K. Leyton-Brown (2008), Action-graph games, *Tech. Rep. TR-2008-13*, University of British Columbia.

Jiang, A. X., and K. Leyton-Brown (2015), Polynomial-time computation of exact correlated equilibrium in compact games, *Games and Economic Behavior*, *91*, 347 – 359, doi: http://dx.doi.org/10.1016/j.geb.2013.02.002.

Johnson, D. S., C. H. Papadimitriou, and M. Yannakakis (1988), How easy is local search?, *Journal of Computer and System Sciences*, *37*(1), 79 – 100.

Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, and L. K. Saul (1999), An introduction to variational methods for graphical models, *Mach. Learn.*, *37*(2), 183–233.

Kakade, S., M. Kearns, J. Langford, and L. Ortiz (2003), Correlated equilibria in graphical games, in *EC '03: Proceedings of the 4th ACM Conference on Electronic Commerce*, pp. 42–47, ACM, New York, NY, USA.

Karlin, S. (1959), *Mathematical Methods and Theory in Games, Programming, and Economics*, Addison Wesley Publishing Company.

Kearns, M., M. Littman, and S. Singh (2001), Graphical models for game theory, in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 253–260.

Koller, D., and B. Milch (2003), Multi-agent influence diagrams for representing and solving games, *Games and Economic Behavior*, *45*(1), 181–221.

La Mura, P. (2000), Game networks, in *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998), Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, *86*(11), 2278–2324.

Leyton-Brown, K., and M. Tennenholtz (2003), Local-effect games, in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 772–777.

Monderer, D., and L. S. Shapley (1996), Fictitious play property for games with identical interests, *Journal of Economic Theory*, *68*(1), 258 – 265, doi:http://dx.doi.org/10.1006/jeth.1996.0014.

Moulin, H., and J. P. Vial (1978), Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon, *International Journal of Game Theory*, *7*(3), 201–221, doi:10.1007/BF01769190.

Nash, J. (1951), Non-cooperative games, *Annals of Mathematics*, *54*, 286–295.

Ortiz, L. E. (2014), On sparse discretization for graphical games, *CoRR*, *abs/1411.3320*.

Ortiz, L. E. (2015), Graphical potential games, *CoRR*, *abs/1505.01539*.

Ortiz, L. E., and M. T. Irfan (2017), Tractable algorithms for approximate Nash equilibria in generalized graphical games with tree structure, in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pp. 635–641, AAAI Press.

Ortiz, L. E., and M. Kearns (2003), Nash propagation for loopy graphical games, in *Advances in Neural Information Processing Systems 15*, edited by S. B. Becker, S. T. Thrun, and K. Obermayer, pp. 817–824.

Papadimitriou, C. H. (2005), Computing correlated equilibria in multi-player games, in *STOC '05: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of cComputing*, pp. 49–56.

Russell, S. J., and P. Norvig (2003), *Artificial Intelligence: A Modern Approach*, Pearson Education.

Shimony, S. E. (1994), Finding MAPs for belief networks is NP-hard, *Artificial Intelligence*, *68*(2), 399 – 410, doi:DOI: 10.1016/0004-3702(94)90072-8.

Szép, J., and F. Forgoó (1985), *Introduction to the Theory of Games*, D. Reidel Publishing Company.

Vickrey, D., and D. Koller (2002), Multi-agent algorithms for solving graphical games, in *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, pp. 345–351.

von Neumann, J., and O. Morgenstern (1947), *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, second Edition.

Wang, C., N. Komodakis, and N. Paragios (2013), Markov random field modeling, inference & learning in computer vision & image understanding: A survey, *Computer Vision and Image Understanding*, *117*(11), 1610 – 1627, doi:http://dx.doi.org/10.1016/j.cviu.2013.07.004.