A conditional game for comparing approximations

Frederik Eaton
Computational and Biological Learning Laboratory
University of Cambridge, UK

Abstract

We present a "conditional game" to be played between two approximate inference algorithms. We prove that exact inference is an optimal strategy and demonstrate how the game can be used to estimate the relative accuracy of two different approximations in the absence of exact marginals.

1 INTRODUCTION

Bayesian machine learning is partly motivated by the recognition that an ability to express beliefs is fundamental to intelligence. These beliefs may be manifested either indirectly through decisions as in betting, or directly as probabilities [1]. The problem of Bayesian statistical inference - to compute such probabilities for a given probabilistic model - is powerful and general. Yet researchers should recognize that "real intelligence" goes beyond statistical inference in many ways. In particular, real intelligence is not just limited to expressing beliefs, but is also able to justify and possibly modify its beliefs through communication. The need for this arises not only in cases where two systems have different evidence, but also where they have reached different conclusions from the same evidence. That two systems might arrive at different beliefs about the same model follows from the reality that, because of constraints on resources, inference in most practical applications must get by with approximations. In such applications it is not feasible to establish which approximation is best by simply comparing with exact marginals, which will be unavailable. Any usable method for ranking two approximations would have to be based on some kind of direct comparison. This paper investigates such a method, based on a two-player game.

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

2 BACKGROUND

The problem of assessing the accuracy of approximations has been previously considered. In the domain of Monte-Carlo-based inference, techniques exist for determining whether a sequence of samples has converged [2]. For message-passing algorithms such as Belief Propagation, there are heuristics to bound and estimate the accuracy of the final approximation [3, 4]. And when samples from a true distribution are available, as when inference is combined with learning, then the approximation accuracy can be estimated from the log-likelihood of a test set.

These techniques have their uses. However, data points are expensive in some domains, so it is not always possible to validate inference using a test set. And heuristics may be unsuitable for making comparisons between two different types of approximation. In general, evaluating the accuracy of an approximation against itself by some internal metric is bound to be unreliable. Comparing two approximations by self-appraisal will fail when one of the approximations is overconfident due to its having overlooked some important structure in the model, for instance in the case of a sampling run which misses an important but isolated mode.

Situations where the computational effort of multiple humans has been expended in parallel analysis of the same model are common in real life, and humans are able to reclaim this seemingly duplicated effort by resolving their disagreements through argument and debate. For intelligent systems to accomplish the same kind of cooperation, they would seem first of all to require a way of directly comparing two approximations. To the best of our knowledge, this paper is the first to propose a formal method for doing so.

Finally, it is perhaps worth noting that approximate inference competitions, such as the UAI approximate inference competition, currently restrict themselves to medium-sized models for which exact inference is still tractable, because there has been no good way to compare the accuracy of approximations without reference to exact marginals. As acknowledged by Bilmes (2006) [5], in such a competition it would be helpful to be able to evaluate the relative performance of algorithms on large models. Our method provides a straightforward way of carrying out such an evaluation.

3 THE CONDITIONAL GAME

We define a game played on a factor graph, called the "conditional game" (CG). Factor graphs are a general representation for statistical models, defining a distribution over n variables $x := (x_1, \ldots, x_n)$ (here assumed discrete) as a normalized product of non-negative factors ψ_α (here assumed strictly positive)

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

where α indexes a collection of sets of variables [6].

Play alternates between two players, the "marginal player", MP, and the "conditional player", CP, over a total of n turns. At turn i the MP expresses marginals for variable x_i, say $q_i(x_i)$. The CP then chooses a value for x_i, say x_i^*. The variable x_i is then fixed to take value $x_i = x_i^*$ for the rest of the game. Play finishes when the variables are all fixed, giving a complete assignment $x = x^*$. A quantity which we will call the "value" of the game is then defined in terms of x^* and q:

$$V = \log \frac{\prod_{i=1}^n q_i(x_i^*)}{\prod_\alpha \psi_\alpha(x_\alpha^*)} \tag{2}$$

Note that if the approximations q are exact conditionals, i.e. if

$$q_i(x_i) = P(x_i | x_{1:i-1}^*) \tag{3}$$

then we have

$$V = \log \frac{\prod_{i=1}^n P(x_i^* | x_{1:i-1}^*)}{\prod_\alpha \psi_\alpha(x_\alpha^*)} = \log \frac{P(x^*)}{\prod_\alpha \psi_\alpha(x_\alpha^*)} \tag{4}$$

$$= -\log Z \tag{5}$$

Thus if MP is exact, the choices of CP have no effect on the value V of the game.

3.1 From approximations to players

To make the conditional game a game, one player should be trying to maximize V and the other to minimize V. It doesn't matter who does which, as long as the two players are in competition.

Now suppose that MP is trying to maximize V, and CP to minimize it. Because probabilities sum to one, if CP has access to exact conditioned marginals then it is possible for him to guarantee through appropriate choice of x_i^* that

$$q_i(x_i^*) \leq P(x_i^* | x_1^*, \ldots, x_{i-1}^*) \tag{6}$$

If MP is not exact, then at least one of these inequalities can be made strict, in which case it follows that $V < -\log Z$. Thus exact marginals, yielding $V = -\log Z$, are the optimal (minimax) strategy for MP.

Given an approximation $Q(x; \psi)$ it is straightforward to derive an MP strategy: at turn i, modify the model ψ to condition on the appropriate variables (perhaps this may be implemented by introducing new factors $\prod_{k=1}^{i-1} \delta(x_k, x_k^*)$), and set q_i to the resulting approximate marginal of x_i under Q[1]

$$q_i(x_i) = Q(x_i | x_{1:i-1}^*) \tag{7}$$

$$\equiv Q(x_i; \prod_\alpha \psi_\alpha(x_\alpha) \prod_{k=1}^{i-1} \delta(x_k, x_k^*)) \tag{8}$$

Note that for most message passing algorithms, the cost of recomputing marginals after imposing a new condition can be mitigated by reusing the messages between runs. If two parts of the graph are uncorrelated or weakly correlated, then a variable in one part of the graph can be conditioned without affecting the messages in the other part.

Suppose that the conditional player CP trusts a different approximation to Q, call it R. A strategy for CP can be derived which employs R. In this case, CP has multiple options, but it seems sensible for him to choose at turn i:

$$x_i^* = \operatorname*{argmin}_{x_i} \frac{q_i(x_i)}{R(x_i | \ldots)} \tag{9}$$

which is guaranteed to satisfy (6) if R is exact, and to do so strictly if MP is not exact. Also, it is optimal at each turn, under the assumption that MP might play optimally for the rest of the game.

A general strategy for MP or CP could be arbitrarily complex, for instance attempting to look several moves ahead by simulating the opposing player. This would presumably be more expensive (or error-prone) than simply coming up with a more accurate approximation and using it in the "naive" strategies above. Thus we will assume below that an approximation will always be associated with one of the recommended strategies (including the amendment for CP of section 3.3). This allows us to drop the distinction between approximations and players, and to view V as a function of two

[1] In (8) we adopt the notation $Q(x; \psi)$ to indicate Q's approximation to a model which is specified by the factors ψ.

approximations. We will write $V^+(Q,R)$ for the game value when CP is trying to maximize V using approximation R against MP's Q; and similarly $V^-(Q,R)$ for when CP is minimizing V.

We now illustrate the CG with a simple example. The model is the fully-connected graph with four binary variables and six pairwise factors, each with entries $\begin{bmatrix} 0.1 & 1 \\ 1 & 1 \end{bmatrix}$. MP uses Belief Propagation [7] and CP uses Gibbs sampling with 10^3 passes. CP tries to minimize V.

i	x_1	x_2	x_3	x_4	MP		CP
1	?				0.743	<	0.798
2	1	?			0.705	<	0.738
3	1	1	?		0.645	>	0.628
4	1	1	0	?	0.909	>	0.908
	1	1	0	0			

Table 1: An example game

The game is depicted in Table 1. Shown are the probabilities that a variable takes the value 1, i.e. $q_i(x_i = 1)$. The final variable assignment is $x^* = (1,1,0,0)$. The final value of the game is $V = \log \frac{0.743 \times 0.705 \times (1-0.645) \times (1-0.909)}{0.1} = -1.778$. The true $\log Z$ is 1.723.

3.2 A bound

We have seen that if MP plays exact marginals, the game value will be optimal, with $V = -\log Z$. We can also derive a simple bound on V in the case that MP's marginals are not exact. We will assume that CP is trying to minimize V, but results for the opposite case are analogous. Let $p_i(x_i) = P(x_i|x^*_{1:i-1})$ denote the conditioned exact marginals.

Theorem 1.

$$V^- \geq -\log Z - \sum_i \max_{x_i} |\log q_i(x_i) - \log p_i(x_i)|$$

Proof. Let $d = \log \prod_{i=1}^n \frac{q_i(x_i^*)}{p_i(x_i^*)}$, then we can write $V = d - \log Z$. We have

$$d \geq \sum_i \min_{x_i} \log \frac{q_i(x_i)}{p_i(x_i)} = -\sum_i \max_{x_i} \log \frac{p_i(x_i)}{q_i(x_i)} \quad (10)$$

$$\geq -\sum_i \max_{x_i} |\log q_i(x_i) - \log p_i(x_i)| \quad \square \quad (11)$$

Thus, if we can guarantee that all of MP's marginals are within a certain distance (measured between logarithms) from the exact marginals, then we can lower-bound V. The accuracy constraint must hold for both unconditioned and conditioned marginals, but approximations usually become more accurate with conditioning, so this theorem gives some intuition as to the relationship between V and the error of the node marginals. We note, however, that the CG is less concerned about the L_1 error, and more concerned about absolute error in log-marginals, to which we refer as the $L_1^{\log}$ error. For example, estimating 10^{-3} when the true probability is 10^{-4} would give a greater $L_1^{\log}$ error than estimating 0.2 when the true probability is 0.3, even though the L_1 error is greater in the second case.

3.3 Variable order

There is nothing special about the order $i = 1, \ldots, n$ in which variables are conditioned at each turn, so it is possible to have CP specify a different order by choosing a variable as well as a value during his turn. (MP must also be modified so that at each turn he specifies marginals for all variables, and not just for the next variable, which he can no longer predict.) In the new flexible-order setting, simply extending the optimization of equation 9 to variables gives a similar optimality property. Thus at turn t, CP now chooses

$$(i_t, x^*_{i_t}) = \operatorname*{argmin}_{\substack{(j,x_j) \\ j \notin i_{1:t-1}}} \frac{Q(x_j|x^*_{i_{1:t-1}})}{R(x_j|x^*_{i_{1:t-1}})} \quad (12)$$

where Q is MP's estimate and R is CP's. The extra freedom for CP allows us to prove a complementary bound to the previous one. Assume, again, that CP wants to minimize V and has access to exact marginals P.

Theorem 2. *If CP is allowed to choose the variable ordering, then he can achieve* $V^- \leq -\log Z - \max_{(i,x_i)} \log \frac{P(x_i)}{Q(x_i)}$

Proof. $V = -\log Z + d$ where

$$d = \sum_{i=1}^n \log \frac{Q(x_i^*|x^*_{1:i-1})}{P(x_i^*|x^*_{1:i-1})} \quad (13)$$

An optimal CP will force each term of d to be negative (or zero). Taking only the first, we have $d \leq \log \frac{Q(x_1)}{P(x_1)}$. But the variable ordering is now decided by CP, who can choose the first variable to get the tightest bound. He also chooses the variable's value, so

$$d \leq \min_{(i,x_i)} \log \frac{Q(x_i)}{P(x_i)} = -\max_{(i,x_i)} \log \frac{P(x_i)}{Q(x_i)} \quad (14)$$

$\square$

3.4 The comparison of approximations

Having defined the conditional game, we now describe how this game can be used to compare two approximate inference methods.

The value V of a game is a number typically near $-\log Z$ (with equality in the case of an exact MP). We could declare a "winner" by comparing V to $-\log Z$, but the true value of $-\log Z$ is unknown and intractable. To identify the most accurate of two approximations, it is helpful to have a score which can be compared to zero. Call the two approximations Q and R and define the "difference score" by

$$S^-(Q,R) = V^-(Q,R) - V^-(R,Q) \tag{15}$$

i.e. the difference between two game values, played with approximations switching roles as CP and MP, and CP minimizing V. This will be ≥ 0 if Q is exact. We also define S^+ analogously using V^+, that is, where CP is maximising V.

We combine S^+ and S^- to get a "four-way score", based on the outcomes of four games[2]:

$$S_4(Q,R) = S^-(Q,R) - S^+(Q,R) \tag{16}$$

The advantage of S_4 can be expressed as follows. The difference score S^- selectively penalizes *under*-estimates of small probabilities by MP, while S^+ penalizes *over*-estimates. For example, if MP underestimates 0.01 for $P(x_i = 0)$ when the true probability is 0.1, and CP is trying to maximize V, then CP will be forced to choose the alternate value $x_i = 1$ (to which MP assigns probability 0.99) since he is only looking for over-estimates. The absolute contribution to the error (e.g. d, equation 13) will then be $\left|\log \frac{0.99}{0.9}\right| = 0.1$ rather than the much larger $\left|\log \frac{0.01}{0.1}\right| = 2.3$.

Our proposed method has now evolved from a simple two-player game with fixed roles into a more complex ritual incorporating four such games, during which players switch roles and objectives. The final product may seem ad-hoc and inelegant. It may help to draw a comparison to legal procedure, in which a simple building block - the questioning of a witness - is employed in four ways to achieve a "fair trial". The witness may be called by the defense or the prosecution, and may be examined and cross-examined.

[2] If CP uses the rule of equation 12 to choose (variable, value) pairs, then the four-way score incorporates four terms corresponding to the values of four games. However, there are only two state configurations x^*, since the configuration which a Q CP chooses when maximizing V against a R MP is the same as that chosen by a R CP when minimizing V against a Q MP. Thus there are two pairs of terms incorporating the same unnormalized probabilities $\prod_\alpha \psi_\alpha(x^*_\alpha)$. However, the terms in each pair do not cancel out, because they occur with the same sign.

Finally, we combine the ideas of Theorems 1 and 2 to prove a simple bound on S_4.

Theorem 3. *Suppose that we are given two approximations Q and R to a true distribution P, with*

$$\sum_t \left|\log \frac{R(x^*_{i_t}|x^*_{i_{1:t-1}})}{P(x^*_{i_t}|x^*_{i_{1:t-1}})}\right| \leq \delta \tag{17}$$

for all x^* and all sequences $i_{1:t}$, while

$$\max_{(i,x_i)} \left|\log \frac{Q(x_i)}{P(x_i)}\right| \geq \epsilon \tag{18}$$

Then $S_4(R,Q) \geq \epsilon - 5\delta$.

Proof. Write $S_4(R,Q) = V^-(R,Q) - V^-(Q,R) - V^+(R,Q) + V^+(Q,R)$. We bound each of the terms:

(a) By Theorem 1, $V^-(R,Q) \geq -\log Z - \delta$ and $V^+(R,Q) \leq -\log Z + \delta$.

(b) We bound $V^-(Q,R)$:

$$V^-(Q,R) + \log Z \tag{19}$$

$$= \sum_t \log \frac{Q(x^*_{i_t}|x^*_{i_{1:t-1}})}{P(x^*_{i_t}|x^*_{i_{1:t-1}})} \tag{20}$$

$$= \sum_t \left(\log \frac{Q(\dots)}{R(\dots)} + \log \frac{R(\dots)}{P(\dots)}\right) \tag{21}$$

$$\leq \sum_t \log \frac{R(\dots)}{P(\dots)} \tag{22}$$

$$\leq \delta \tag{23}$$

Equation 22 follows from the fact that CP will choose $\log \frac{R}{Q}$ to be negative.

(c) For the last term $V^+(Q,R)$, suppose the first condition of the game is $(i_1, x_{i_1}) = (k, x^*_k)$. Let (j, x^*_j) be the maximizing assignment in equation 18, so that either $\log \frac{Q(x^*_j)}{P(x^*_j)} \geq \epsilon$ or $\leq -\epsilon$. Assume the first case; the proof for the second follows by similarly modifying part (b) above. Now,

$$\log \frac{Q(x^*_k)}{R(x^*_k)} \geq \log \frac{Q(x^*_j)}{R(x^*_j)} \tag{24}$$

$$= \log \frac{Q(x^*_j)}{P(x^*_j)} - \log \frac{R(x^*_j)}{P(x^*_j)} \tag{25}$$

$$\geq \epsilon - \delta \tag{26}$$

Then as in (b),

$$V + \log Z \tag{27}$$

$$= \sum_t \left(\log \frac{Q(\dots)}{R(\dots)} + \log \frac{R(\dots)}{P(\dots)}\right) \tag{28}$$

$$\geq \epsilon - \delta + \sum_t \log \frac{R(\dots)}{P(\dots)} \tag{29}$$

$$\geq \epsilon - 2\delta \tag{30}$$

where equation 29 follows from using 26 and $i_1 = k$ for the first term of the summation.

Combining (a), (b), and (c) gives $S_4 \geq \epsilon - 5\delta$.

□

In other words, if we can bound the total $L_1^{\log}$ error of one approximation above by δ, and if we know that another approximation does worse than ϵ for the maximum error of one of its variable marginals, and if $\epsilon - 5\delta > 0$, then the first approximation will win against the second one by the four-way score.

This bound is strict and so assumes the worst case scenario for every game. A probabilistic analysis estimating average-case performance given a random distribution of marginal errors might provide a more realistic picture of the CG's effectiveness, but we do not undertake such an analysis here.

4 EXPERIMENTS

4.1 Alarm graph

We first present the results of playing the conditional game between five different pairs of approximate inference algorithms, using the implementation in libDAI [8], running on the "alarm graph" found in libDAI, with 37 variables. The algorithms we consider are:

- **Gibbs** - Gibbs sampling, with 10^5 passes.
- **BP** - Belief Propagation, sequential updates [7].
- **CBP** - Conditioned Belief Propagation, with 4 levels [9].
- **TreeEP** - Tree Expectation Propagation [10].
- **LCBP** - Loop Corrected Belief Propagation [11].

The L_1 and $L_1^{\log}$ errors of the algorithms are shown in Table 2. The S_4 scores are shown in Table 3. We see that the S_4 scores agree with the *average* $L_1^{\log}$ errors on all pairs except BP vs Gibbs, where Gibbs wins even though it has a larger average $L_1^{\log}$ error. But note that the *maximum* $L_1^{\log}$ error of Gibbs is smaller than that of BP, so there is at least one sensible error measure which is consistent with the result of the game in every case.

4.2 Generalized Belief Propagation

We might also be interested in measuring the relationship between score and error for multiple models and a larger space of approximations. To this end we used approximations consisting of Generalized Belief Propagation (GBP, [12])[3] on a fully connected binary pairwise factor graph with triangular regions (regions of size 3). Each approximation was defined by a random set of triangular regions, chosen to be non-singular [14]. As models we used factor graphs of 7 nodes with edge potentials drawn as $\exp(2W)$, with W a standard normal deviate. Figure 1 plots the results of playing the CG between 16 random pairs of GBP approximations on each of 120 random models; shown is the four-way score S_4 and the difference in $L_1^{\log}$ error.[4] Figure 2 plots the same results but shows difference in L_1 error instead, illustrating that the S_4 score is better at capturing relative $L_1^{\log}$ error than L_1 error.

We will term the "agreement rate" of the CG against a certain error metric as the rate at which the CG correctly identifies the approximation with smallest error. This depends on the particular set of approximations which are being compared (in our case, GBP with different sets of triangular regions). It can be estimated from the fraction of points in the first and third quadrant in Figure 1 and Figure 2. For $L_1^{\log}$ error, the agreement rate was 0.754. For L_1 error, it was 0.639.

Method	avg L_1	avg $L_1^{\log}$	max $L_1^{\log}$
LCBP	8.981e-05	5.586e-4	0.01684
TreeEP	0.008652	0.04424	0.5475
CBP	0.01110	0.05355	1.256
BP	0.01627	0.0712988	1.6424
Gibbs	0.02251	0.2111	0.8298

Table 2: Errors between approximate and exact variable marginals for different approximations.

S_4 vs:	TreeEP	CBP	BP	Gibbs
LCBP	5.2507	13.795	22.75	12.998
TreeEP		8.383	13.453	3.996
CBP			27.575	3.734
BP				-4.032

Table 3: Scores of games between approximations

4.3 Comparison to code-length game

We next compare the effectiveness of the conditional game against another simple game, a modification of the "code length game" [15]. The outcome of the code-

[3]For reliable convergence, our implementation used the algorithm of Heskes, Albers, and Kappen [13], which has the same fixed-points as GBP. We ran it with a tolerance of 10^{-7}.

[4]Each point is also reflected about the origin.

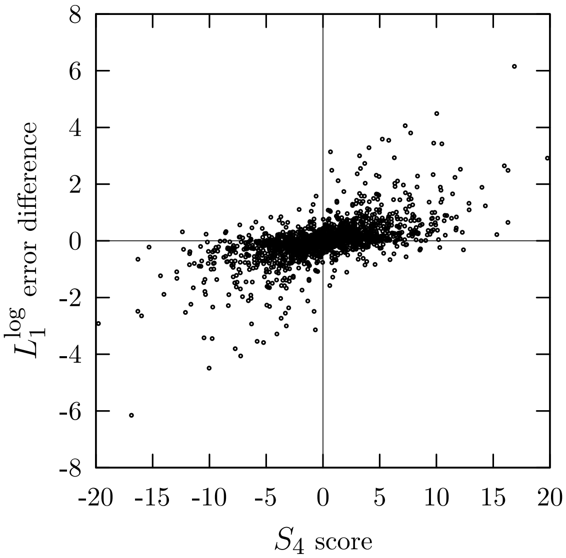

Figure 1: Four-way score vs difference in $L_1^{\log}$ error for GBP

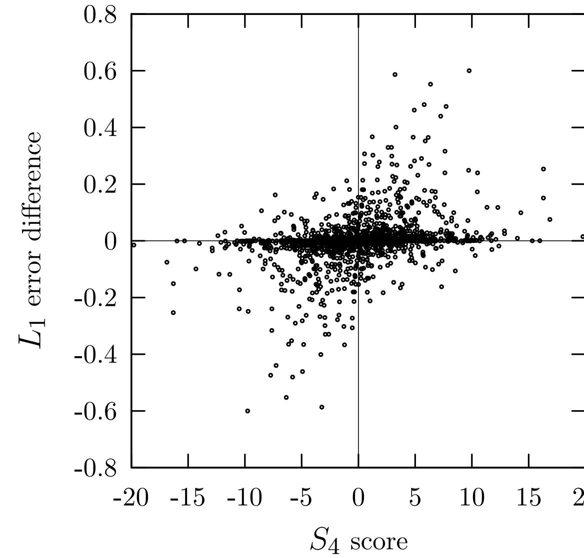

Figure 2: Four-way score vs difference in L_1 error for GBP

length game (CLG) is defined as follows:

$$\max_{p:\sum_x p(x)=1} \min_{\kappa:\sum_x e^{-\kappa(x)}\leq 1} \mathbb{E}_p \left[\kappa + \sum_\alpha \log \psi_\alpha\right] \quad (31)$$

We have added the term $\sum_\alpha \log \psi_\alpha$ to achieve the correct equilibrium for the model. The standard interpretation of this saddle point is that one player chooses a set of code lengths satisfying the Kraft inequality, while another chooses a normalized distribution (P) over symbols. The first player wants to minimize the (modified) expected code length, and the second to maximize it. The equilibrium is at $p = e^\kappa = P$. Note that a sample from this expectation (sign inverted) can be implemented by changing the behavior of CP in the CG so that he chooses a value randomly from his own distribution $R(x_i|\ldots)$ at each turn. This is not a good strategy for CP in the CG, since in particular it ignores the marginals proposed by MP, but the CLG is a simultaneous game, where each player is unaware of the other's actions, and so in that setting CP (i.e., the distribution player) should act randomly. If the distribution player wants to do well in the CLG in expectation, then his best strategy is to sample as described above. The expected value of the game is

$$\mathbb{E}[V] = \mathbb{E}_R\left[\log \frac{Q}{\prod_\alpha \psi_\alpha}\right] \quad (32)$$

$$= \mathbb{E}_R\left[\log \frac{Q}{P}\right] - \log Z \quad (33)$$

where Q is MP's approximation. This is equal to $-\log Z$ if Q is exact, and less than or equal to $-\log Z$ if R is exact. The equivalent of the S^- difference score yields a value which can be compared to zero. The drawback of the CLG is that its outcome is stochastic, and so one must average over many trials to get a score of low variance. As a consequence, one might object that a comparison between the CG and CLG is unfair. However, the CLG is the only other game of this type, of which we are aware.

We want to show that the CG is better on average than the CLG at discriminating the error of many similar approximations. For this we used the same GBP approximations as in section 4.2 (parameterized by triangular region configurations) on the same distribution over models. We played these approximations against each other in a "single-elimination tournament". Players are initialized at the leaves of a binary tree of uniform depth (here 8), and each node represents the winner of a game played between its two children. The "round" of a node is its distance from the leaves. See Figure 3.

The tournament was repeated with different methods of comparing approximations:

- conditional game (S_4, S^+, S^-);
- code-length game (averaging over 1, 3, and 8 runs);
- exact comparison (comparing actual error of approximations).

The "exact" method is shown only as a reference, as we are ultimately interested in problems for which exact marginals are intractable.

The results are shown in Figure 4. The error shown in this plot is average $L_1^{\log}$ over variable marginals, averaged over all approximations in the same round, geometrically averaged over 120 random factor graphs generated as above. In both cases one can see that S_4 outperforms S^+ and S^- by a small amount, while the code length game performs poorly. In both cases, the slope of the S_4 curve was close to half of the slope of the exact reference curve.

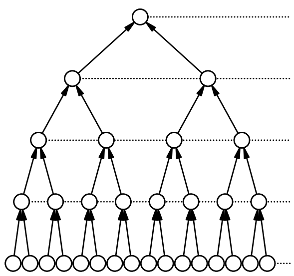

Figure 3: Schematic of the single-elimination tournament on a binary tree

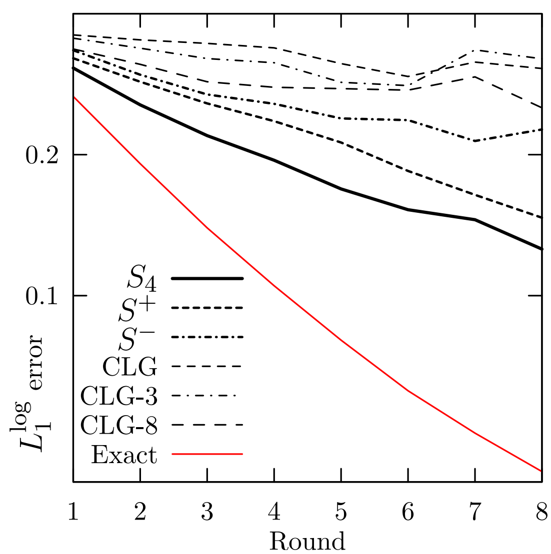

Figure 4: Plot of $L_1^{\log}$ error as a function of round, for tournament experiment (round 0 omitted).

It is interesting to see how the agreement rate, defined in section 4.2, changes as a function of tournament round. For $L_1^{\log}$ error and S_4 score, the agreement rates (averaged over all the approximations and all the graphs) for tournament rounds one through eight are shown in table 4. There is a downward trend for both games, which means that they are having a more difficult time discriminating errors with each new round. This is consistent with the usual state of affairs when a tournament is being played - it is easier to predict the outcome of earlier matches than later ones, since the earlier matches are more likely to involve an uneven pairing of players.

Round	1	2	3	4
S_4	0.71	0.68	0.67	0.65
CLG	0.55	0.53	0.53	0.51
Round	5	6	7	8
S_4	0.65	0.63	0.55	0.53
CLG	0.49	0.56	0.33	0.50

Table 4: Agreement rates vs $L_1^{\log}$ error for S_4 and CLG

5 DISCUSSION AND FUTURE WORK

We have described a technique for comparing two different approximations to a statistical model. The only interface requirement for the approximation algorithms is that they support variable conditioning, i.e. can give estimates of marginals in a conditioned model where a variable is conditioned to take a given value. Some algorithms which satisfy this requirement particularly well are Belief Propagation [7] and instances of Expectation Propagation [16], and GBP [12].

The original motivation of this research was to explore ways of moving beyond the dominant approximate inference framework, in which algorithms are only able to express beliefs. It seemed that if one were to be more adventurous, as a natural progression one might seek frameworks in which algorithms are able to defend or modify their beliefs through dialog. We decided that an appropriate prototype for such communication should be a two-player game. This was supported partly by the observation that there is no easy fitness function with which to measure the error of an approximation, but that relative comparisons (such as the code-length game) are possible. We also noticed that two-player games already appear in many places in machine learning, in the form of saddle points $\min_x \max_y f(x, y)$: for example in the Convex-Concave Procedure [17], Tree-Reweighted Belief Propagation [18], Boosting [19], and the EM algorithm [20].

There is also a well-known (to formal semanticists) two-player game which can be used to define the truth value of a formula in first-order logic. The state of the game is a node in the syntax tree of the formula. Play starts at the root. A "falsifier" chooses branches of conjunctions (AND clauses) which he thinks are false, while a "verifier" chooses branches of conjunctions (OR clauses) which he thinks are true. Upon encountering a negation, they switch roles. (The falsifier and verifier can also instantiate the arguments of ∀ and ∃ quantifiers, respectively.) The formula is true if and only if the verifier can win. [21] We find this game particularly interesting, although it is not clear what kind of analogy best relates it to the conditional game.

Finally, we note that there is a body of literature which applies iterative, message-passing-like algorithms to look for solutions of games which have a graphical structure, called "graphical games" [22]. We have not found a way to make use of it here.

We have made preliminary attempts to harness the conditional game in an approximate inference method, by using it to guide a kind of natural selection between competing approximations. This has proved difficult because the game is approximate and so does not prevent regression in fitness. Such regression is impossible in the traditional Genetic Algorithm setting where the fitness of single individuals can be evaluated using an absolute (rather than relative) fitness function, but has been recognized in that field in the context of "population-dependent" fitness functions and coevolution [23].

A fundamental drawback of the conditional game is that it requires a complete traversal of all variables in the model, where the algorithm must be re-run once for each variable. This is still faster than the presumably exponential cost of exact inference, but would seem unsuitable for large real-world models. One remedy would be to use approximate inference algorithms that "compile" a model into a form through which conditional and marginal queries can be executed quickly. An example of such an algorithm is described in recent work applying Arithmetic Circuits [24] to approximate inference [25].

Ideally, it would be possible to devise a game which can be played locally on the nodes of a graphical model, so that inference in different weakly-coupled areas of the model can proceed asynchronously, together with co-evolution towards locally superior approximations. It is not yet clear how this could be done.

In conclusion, we have presented a novel game which can be used for comparing approximate inference algorithms in the absence of exact marginals. We have shown that it has exact inference as an optimal strategy, and we have proven theoretical bounds on its performance in the case where neither player is exact. We have presented experimental results which demonstrate its effectiveness in distinguishing inference algorithms on a graph of moderate difficulty, the alarm graph. We have experimentally demonstrated its superiority to another simple game, the code-length game, for the purpose of comparing approximations based on GBP. We hope that this research will help generate interest in applications, techniques, and formalisms for approximate inference which extend beyond the current paradigm of simply expressing beliefs.

Acknowledgements

The author would like to thank Iain Murray for useful discussions.

References

[1] B. de Finetti. Probabilism: A critical essay on the theory of probability and on the value of science. *Erkenntnis*, 31(2):169–223, 1989.

[2] J.G. Propp and D.B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random structures and Algorithms*, 9(1-2):223–252, 1996.

[3] J.M. Mooij and H.J. Kappen. Bounds on marginal probability distributions. In *Advances in Neural Information Processing Systems 21*, pages 1105–1112. 2009.

[4] S. Shekhar. Fixing and Extending the Multiplicative Approximation Scheme. Master's thesis, University of California, Irvine, 2009.

[5] J. Bilmes. UAI06 inference evaluation results. 2006.

[6] F.R. Kschischang, B.J. Frey, and H.A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

[7] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3):241–288, 1986.

[8] J.M. Mooij et al. libDAI 0.2.5: A free/open source C++ library for Discrete Approximate Inference. http://www.libdai.org/, 2010.

[9] F. Eaton and Z. Ghahramani. Choosing a variable to clamp: approximate inference using conditioned belief propagation. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 145–152, 2009.

[10] T. Minka and Y. Qi. Tree-structured approximations by expectation propagation. In *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, page 193, 2004.

[11] J.M. Mooij, B. Wemmenhove, HJ Kappen, and T. Rizzo. Loop corrected belief propagation. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

[12] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Generalized belief propagation. *Advances in neural information processing systems*, pages 689–695, 2001.

[13] T. Heskes, K. Albers, and B. Kappen. Approximate inference and constrained optimization. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*, volume 13, pages 313–320, 2003.

[14] M. Welling, T. Minka, and Y.W. Teh. Structured region graphs: Morphing EP into GBP. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence*, pages 609–616, 2005.

[15] F. Topsøe. Information theoretical optimization techniques. *Kybernetika*, 15(1):8–27, 1979.

[16] T.P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*, volume 17, pages 362–369, 2001.

[17] A.L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[18] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudomoment matching. In *Workshop on Artificial Intelligence and Statistics*, volume 21, 2003.

[19] Y. Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

[20] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[21] P. Lorenzen and K. Lorenz. *Dialogische logik*. Wissenschaftliche Buchgesellschaft Darmstadt, Germany, 1978.

[22] M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 253–260, 2001.

[23] L.M. Schmitt. Theory of genetic algorithms II: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling. *Theoretical Computer Science*, 310(1-3):181–231, 2004.

[24] A. Darwiche. A Differential Approach to Inference in Bayesian Networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 123–132, 2000.

[25] D. Lowd and P. Domingos. Approximate inference by compilation to arithmetic circuits. In *Advances in Neural Information Processing Systems 23*, pages 1477–1485. 2010.