Approximate Inference for Stochastic Planning in Factored Spaces

Zhennan Wu ZWU1@IU.EDU

Roni Khardon RKHARDON@IU.EDU

Department of Computer Science
Luddy School of Informatics, Computing, and Engineering
Indiana University Bloomington
Bloomington, Indiana, USA

Abstract

Stochastic planning can be reduced to probabilistic inference in large discrete graphical models, but hardness of inference requires approximation schemes to be used. In this paper we argue that such applications can be disentangled along two dimensions. The first is the direction of information flow in the idealized exact optimization objective, i.e., forward vs. backward inference. The second is the type of approximation used to compute this objective, e.g., Belief Propagation (BP) vs. mean field variational inference (MFVI). This new categorization allows us to unify a large amount of isolated efforts in prior work explaining their connections and differences as well as potential improvements. An extensive experimental evaluation over large stochastic planning problems shows the advantage of forward BP over several algorithms based on MFVI. An analysis of practical limitations of MFVI motivates a novel algorithm, collapsed state variational inference (CSVI), which provides a tighter approximation and achieves comparable planning performance with forward BP.

1. Introduction

The connection between planning and probabilistic inference is well known and multiple reductions exist showing how inference algorithms can be used to solve stochastic planning problems. Such reductions are equivalent when one can perform exact inference but this is not typically the case for challenging planning problems that have many state variables, a.k.a. factored spaces, where approximate inference schemes are introduced. The planning and reinforcement learning literatures include multiple such efforts where different algorithmic frameworks are combined with different approximation schemes. For example, constructions exist through weighted model counting Domshlak and Hoffmann (2006), several forms of variational inference (e.g., (Toussaint and Storkey, 2006; Levine, 2018)), and several forms belief propagation (e.g., (Liu and Ihler, 2012; Cui et al., 2019)). However, it is not clear how different algorithmic approaches are related to one another and how the choice of approach interacts with the choice of approximation scheme.

The paper makes three contributions. First we provide a unified scheme that connects previous approaches along two dimensions, using either forward or backward reasoning, and choosing what approximation to use, where we address Belief Propagation (BP) and mean field variational inference (MFVI). This allows us to put prior work in a unified framework that explains choices made by corresponding algorithms. In particular, our analysis shows that Forward MFVI which is used in some papers can be understood to run multiple iterations of Backward MFVI and thus provides tighter approximations. Second, through extensive experiments over large planning problems, we show that forward reasoning with Belief Propagation provides the best performance among these algorithms, that MFVI provides poor performance in some domains, and that modifying MFVI using exponentiated rewards helps in some cases but not sufficiently. We also analyze the failures of MFVI experimentally pointing to sensitivity in updates. Third, based on the analysis, we propose a novel algorithm, Collapsed State Variational Inference (CSVI), that uses mean field with collapsed variational inference where state variables are integrated out. CSVI is motivated theoretically due to its tighter variational approximation and we show empirically that it matches the performance of Forward Belief Propagation. This shows that while naive application of mean field for planning fails, other variational approximations like CSVI can yield strong planning performance. Due to space constraints some technical details and experiment results are omitted from the main paper and are provided in (Wu and Khardon, 2022), which we refer to below as the full paper.

2. Problem Formulation

We consider finite horizon MDPs as specifications of planning problems. Such specifications are often compiled from a high level description language but this is orthogonal to the discussion in the paper. Specifically, consider Markov Decision Processes $\langle \mathcal{S}, p(s_0), \mathcal{A}, \mathcal{P}, \mathcal{R}, T, \gamma \rangle$, where $\mathcal{S}$ denotes the state space, $p(s_0)$ is a distribution over start states, $\mathcal{A}$ denotes the action space, $\mathcal{P}$ denotes the transition probability $p(s_{t+1}|s_t, a_t)$, $\mathcal{R}$ denotes the reward function $R(s_t, a_t)$, T denotes the horizon and γ is the discount factor. In this paper we set $\gamma = 1$, but this does not significantly affect any of the formulations. A solution is given by a policy, π, that specifies $p_{\theta_t}(a_t|s_t)$ with policy parameters $\theta = \{\theta_t\}$ allowing for non-stationary policies. The task in planning is to find a policy that maximizes the expected cumulative reward $\mathbb{E}[\sum_{t=0}^{T-1} R(s_t, a_t)]$ where the expectation is taken w.r.t. trajectories generated from the MDP with policy π, that is, $s_0 \sim p(s_0)$, $a_t \sim p_{\theta_t}(a_t|s_t)$, and $s_t \sim p(s_t|s_{t-1}, a_{t-1})$.

In this paper we follow recent practice in stochastic planning and use the *online planning* framework, where in a state s, the algorithm computes for a limited time to pick an action a, uses a to control the MDP to get to the next state, and repeats this process. Online planning is often used with receding horizon control, where the planner uses a T step lookahead in its search and then extracts the first action a to be applied in s.

Solving a finite horizon MDP is equivalent to solving an inference problem in the corresponding Dynamic Bayesian Network (DBN), or more precisely in the dynamic decision network. We assume a factored form of states consisting of binary state variables, i.e. $s_t = (s_t^1, \cdots, s_t^M)$. We also assume a factored action representation $a_t = (a_t^1, \cdots, a_t^N)$.

The MDP formulation above requires real-value reward nodes in the DBN. To facilitate inference one can replace these nodes with constructions that use only binary variables, and various such constructions appear in the literature. In the following we develop one such construction and use that in our experiments. We introduce binary reward random variables r_t to capture the reward after taking action in the previous time step $t-1$, the distribution of which is defined as

$$p(r_t = 1|s_{t-1} = s, a_{t-1} = a) = \frac{R(s_{t-1} = s, a_{t-1} = a)}{\max_{s,a} R(s,a)}. \tag{1}$$

We can then define $\tilde{R}$ where $p(\tilde{R}=1)=\frac{\sum_1^T r_t}{T}$ to capture the cumulative reward. We call the resulting DBN the intermediate representation. However, $\tilde{R}$ has T parents which hinders efficient inference. To avoid the use of $\tilde{R}$, we introduce cumulative reward binary random variables c_t. To keep the consistency of the graphical structure, we create an auxiliary node $c_0 \equiv 1$, and for $t > 0$ the distribution of c_t is defined recursively depending on the previous cumulative reward c_{t-1} and current reward r_t:

$$p(c_t = 1 | c_{t-1}, r_t) = \frac{(t-1)c_{t-1} + r_t}{t}. \tag{2}$$

In many planning problems the reward is given as an additive function over a set of small factors. For such problems we introduce another chain of binary reward variables within a time step using a similar construction. This yields a DBN that only includes binary variables with a small number of parents. As the following proposition shows the three constructions, using cumulative reward, using $\tilde{R}$ and using c_T are equivalent. Further details and proofs are given in the full paper.

Proposition 1 *The construction satisfies* $\mathbb{E}[\sum_{t=0}^{T-1} R(s_t, a_t)] \propto \mathbb{E}(\tilde{R}) \propto \mathbb{E}(c_T)$ *where expectations are w.r.t. trajectories as above.*

3. Planning Through Inference

In the following we restrict our discussion to open loop policies, that is, $p_{\theta_t}(a_t|s_t) = p_{\theta_t}(a_t)$ where the policy is time dependent but does not depend on the state (other than s_0 if it is fixed). Thus for an action sequence $A = \{a_0, \ldots, a_{T-1}\}$, we have $p_\theta(A) = \prod p_{\theta_t}(a_t)$. This covers most of previous work on planning as inference in the literature. The extension to standard policies is straightforward but requires more complex algorithms for optimization.

3.1 Forward Backward Framework

We now present a simple framework that captures many algorithms in the literature. For the discussion below note that some algorithms optimize policy parameters θ and then choose the actions, whereas others optimize the action sequence A directly.

The Backward Framework: Observe that if θ is the uniform distribution, u, then

$$\arg\max_A p(c_T = 1|A) = \arg\max_A \frac{p_u(A|c_T = 1)p_u(c_T = 1)}{p_u(A)} = \arg\max_A p_u(A|c_T = 1) \tag{3}$$

where the second equality is true because $p_u(A)$ is a fixed constant for all A and $p_u(c_T = 1)$ does not depend on A. This suggests that we can optimize $p(c_T = 1|A)$ by optimizing $p_u(A|c_T = 1)$. Since calculating $p_u(A|c_T = 1)$ is hard, the backward framework optimizes an approximation of $p_u(A|c_T = 1)$. The choice of different approximations $q_\phi(A)$ will give us different concrete algorithms. This is captured in Algorithm 1

The Forward Framework: in contrast, the forward approach aims to directly optimize $p_\theta(c_T = 1)$ w.r.t the policy parameters (or alternatively, $p(c_T = 1|A)$ but we focus on the more general case). Approximating $p_\theta(c_T = 1)$ with a score function $sc(\theta)$ defined on policy parameters yields the forward framework. In the ideal case, maximizing $sc(\theta)$ will give us a delta function, directly selecting a concrete A sequence. If not, we can use $\arg\max$ or sample from the corresponding distribution. This is captured in Algorithm 2

Algorithm 1 Backward Inference

```
1. Calculate $q_\phi(A) \approx p_u(A|c_T = 1)$
2. Pick $A = \arg\max q_\phi(A)$
```

Algorithm 2 Forward Inference

```
1. Define a score function $sc(\theta) \triangleq sc(c_T = 1|\theta) \approx p_\theta(c_T = 1)$
2. Optimize $\theta$ to maximize the score function.
3. Pick $A$ using $p_\theta(A)$
```

3.2 Forward and Backward Loopy Belief Propagation

The forward and backward algorithms can be combined with any approximation scheme. We start by considering loopy BP (LBP) algorithms (Pearl, 1988; Kschischang et al., 2001). For this construction we translate the DBN into a factor graph using standard constructions. For backward LBP, we instantiate $c_T = 1$ as evidence, fix the factors corresponding to θ to be the uniform distribution, and run LBP to calculate the marginal probabilities on action variables. That is, $q_\phi(A)$ is given by the output of LBP. Note that this is algorithmically simple because we do not need a separate optimization step aside from Belief Propagation. However, LBP may need many iterations to converge or may not converge at all.

For the forward algorithm, we define $sc(\theta)$ to be the approximate marginal of $p_\theta(c_T)$ computed by LBP. However, LBP does not optimize θ. As discussed below, multiple techniques for optimizing θ for LBP exist in the literature. In the experiments we use the SOGBOFA system (Cui et al., 2019) that combines LBP with gradient based search.

3.3 Forward and Backward Mean Field Variational Inference

The idea in variational inference is to minimize the KL divergence between the approximate posterior and the true posterior over latent variables, i.e., in our case

$$d_{KL}(q_\phi(S, A, R, C_{\setminus T})||p_\theta(S, A, R, C_{\setminus T}|c_T = 1))$$

where the latent variables are S, A, R, C, that is, the sequences of state, action, reward, and cumulative reward variables, where $C_{\setminus T}$ excludes c_T. This is equivalent to maximizing the evidence lower bound (ELBO). In our case the ELBO is given in the next equation, where in the mean field approximation q_ϕ is a product of independent factors

$$\log p_\theta(c_T = 1) \geq \mathbb{E}_{q_\phi}[\log \frac{p_\theta(S, A, R, C_{\setminus T}, c_T = 1)}{q_\phi(S, A, R, C_{\setminus T})}] =: ELBO_{\theta,\phi}. \tag{4}$$

For backward MFVI, note that $p_u(A|c_T = 1)$ is the marginal distribution of the true posterior $p_u(S, A, R, C_{\setminus T}|c_T = 1)$. Therefore we first maximize $ELBO_{\phi,\theta=u}$ to obtain $q_\phi(S, A, R, C_{\setminus T})$ and then set $q_\phi(A)$ to be the corresponding marginal. Detailed update equations for MFVI are given in the full paper.

For forward MFVI, we can pick $sc(\theta) = ELBO_{\phi,\theta} \approx \log p_\theta(c_T = 1)$ where we need to optimize both ϕ and θ. For this, the standard approach is the Variational Expectation

Maximization algorithm which optimizes ϕ in the E step and θ in the M step. To elaborate the algorithm, note that the ELBO can be reformulated as follows:

$$ELBO_{\theta,\phi} = \mathbb{E}_{q_\phi}[\log \frac{p(S, R, C_{\setminus T}, c_T = 1|A)}{q_\phi(S, R, C_{\setminus T}|A)}] - d_{KL}(q_\phi(A)||p_\theta(A)) \tag{5}$$

where the first term does not depend on θ. Therefore:

- In the E step, we maximize $ELBO_{\theta,\phi}$ w.r.t. ϕ. *Note that this is exactly as in the Backward Algorithm but under a general* θ.
- In the M-step, we keep q_ϕ fixed and optimize the $ELBO_{\theta,\phi}$ w.r.t. θ. From Eq (5) we see that this is equivalent to minimizing $d_{KL}(q_\phi(A)||p_\theta(A))$. If $q_\phi(A)$ and $p_\theta(A)$ are from the same class of distributions, this step assigns $\theta \leftarrow \phi$.

From the procedure, we have the following observation.

Remark 2 *For the mean field approximation, the forward algorithm is an iterative process that alternates the backward algorithm with policy updates.*

This connection was not observed in prior work where the forward and backward algorithms are not clearly distinguished. Finally, as pointed by Toussaint and Storkey (2006) the E step is analogous to policy evaluation (except that we calculate marginals for many variables besides the reward) and the M step is analogous to policy improvement, so forward MFVI can be seen as an approximate version of Policy Iteration.

4. Related Work

The idea of using inference for stochastic planning has a long history and has attracted many different approaches. For example, Cooper (1988) showed how inference can be used for decision making in influence diagrams, Domshlak and Hoffmann (2006) use an approach based on weighted model counting, Nitti et al. (2015) use a probabilistic programming formulation, and Lee et al. (2021) use anytime marginal MAP solvers for planning problems.

Several groups have developed approaches that follow the forward variational framework, going back to Dayan and Hinton (1997). This idea is often developed by defining a reward weighted path distribution which is similar to conditioning on $c_T = 1$ in our framework, and developing algorithms from this formulation (Furmston and Barber, 2010, 2011; Toussaint and Storkey, 2006; Kumar et al., 2015). We note, however, that these works did not explicitly address factoring over state and action variables.

On the other hand, some papers in robotics and reinforcement learning (RL) (Toussaint, 2009; Kappen et al., 2012; Levine, 2018) follow the backward variational framework. In contrast with the discussion above they use a formulation where the reward over trajectories is exponentiated. As shown by Levine (2018) this modifies the original optimization objective by adding a term with the expected entropy of the policy, and hence solves a slightly different problem, but the entropy term may be beneficial for exploration in RL. In addition, the work of Neumann (2011) uses the forward variational algorithm, but with an exponentiated reward, and additional sampling-based approximations. We can see that the forward and backward variational approaches have been widely used but have not been differentiated before. Our analysis above clarifies the relationship between these approaches.

For the case of BP approximation, Murphy and Weiss (2001) proposed the Factored Frontier Algorithm which is a forward BP method for marginal inference, and Boyen and Koller (1998) developed approximation bounds for forward inference. The work of Liu and Ihler (2012); Kiselev and Poupart (2014) follows the forward BP framework, but develops a generalized belief propagation algorithm that solves both optimization and expectation steps using message passing. The work of Cui et al. (2019) also follows the forward BP framework but decouples the expectation which is done through BP from the optimization that uses an approximation based on gradient search.

Several works have made additional assumptions on the structure of the DBN in their discussion of graph-based MDPs. Cheng et al. (2013) extend the algorithm of Liu and Ihler (2012) to this case. Peyrard and Sabbadin (2006) and Sabbadin et al. (2012) use the Mean Field approximation method but only use it to approximate the distribution over state variables. They then use the approximate distribution to approximate steps of the Policy Iteration algorithm. Hence their algorithm is different from MFVI in that reward variables are not included in the variational approximation. Finally, our work can be seen to extend the comparison of Mean Field and Loopy BP for general inference tasks (Weiss, 2001). As in this early work, our experiments show that optimization of variational objectives can lead to local optima and that BP can provide some advantage.

5. Experiments and Analysis of MFVI & Loopy BP Algorithms

This section presents an experimental evaluation of the algorithms. The code for regenerating all the results is available on Github[1]. Our goal in this paper is to understand the *quality of decisions* provided by different approximate inference schemes, ignoring implementation details. Therefore, during the experiments we do not limit run time but instead allow the algorithms to converge, within bounds given below, before proposing a decision. We chose 6 problem domains from the ICAPS 2011 International Probabilistic Planning Competition to conduct our experiments. Each domain has 10 instances with factorized structure, horizon of 40 and discount factor of 1, and instances differ by the number of state and action variables. For our experiments we use the SPUDD (Hoey et al., 1999) translation of the original RDDL (Sanner et al., 2010) specification, which compiles away action factoring. This simplifies the implementation because it removes the need to reconcile action constraints with factoring. To control our overall experimental time we use online planning with receding horizon control, where we set the search horizon to be the minimum value between 9 and the remaining time steps.

Algorithmic parameters for MFVI: we perform at most 100 Variational updates and stop early if the infinity norm of the difference between consecutive approximation distributions is less than 0.1. We perform 3 outer iterations, i.e., policy updates for the forward version.

Algorithmic parameters for BP variants: We use SOGBOFA (Cui et al., 2019)[2] as forward Loopy BP, fixing search depth to 9, and limiting the number of gradient updates to 500. We note that SOGBOFA has outperformed other planners, including search based

[1] https://github.com/Zhennan-Wu/AISPFS

[2] https://github.com/hcui01/SOGBOFA

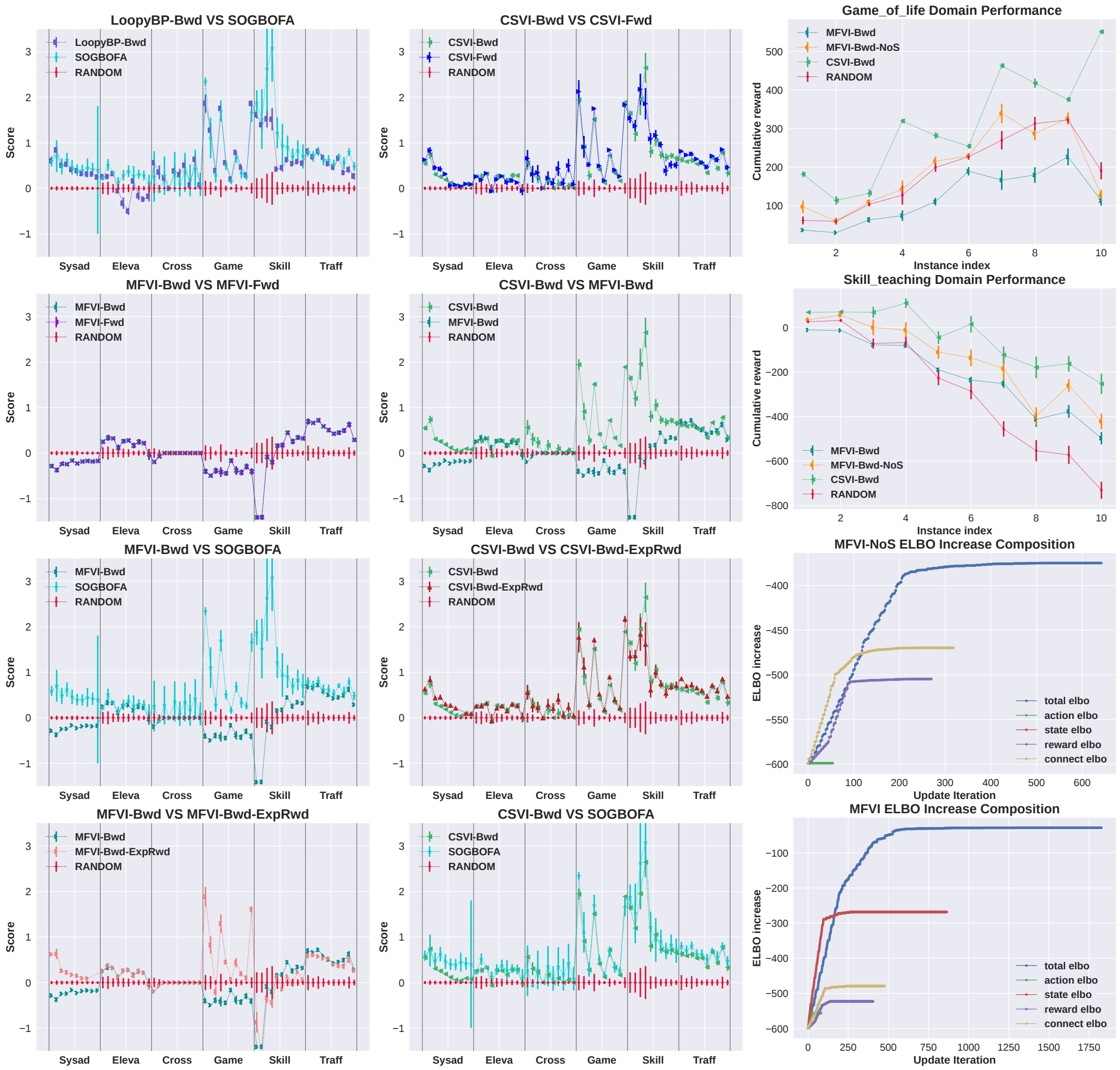

Figure 1: First and second columns: algorithm comparisons on 60 problem instances, averaged over 12 simulations on each instance. Third column: comparing MFVI variants with and without state updates and the contributions of variable groups to the increase in the ELBO (in Skill teaching, instance 1, step 2 of execution).

planners, in IPPC 2018 problems and is a state of the art baseline for the evaluation. For the backward algorithm, our implementation is based on Zhou et al. (2022) with parallel message update and a bound of 100 iterations with no damping ($\beta = 0$).

Normalized mean ± one standard deviation of the cumulative reward over 12 simulations are shown in all the plots. Denote the mean value and standard deviation of the cumulative reward of algorithm a on instance i to be $\bar{r}_i(a)$, $\sigma_i(a)$, respectively. To facilitate comparisons across domains we report scores normalized relative to the random policy. Specifically, for algorithm a on instance i, score-mean$_i(a) = \frac{|\bar{r}_i(a) - \bar{r}_i(RANDOM)|}{|\bar{r}_i(RANDOM)|}$ and score-std$_i(a) =$

$\frac{\sigma_i(a)}{|\bar{r}_i(RANDOM)|}$ where the random algorithm has score 0 and higher scores indicates better performance. For reference, the raw results are given in the full paper.

Comparison of Algorithms Results are shown in the left column of Figure 1. The top plot shows that search direction is important for BP: the forward algorithm (SOGBOFA) outperforms the backward algorithm.[3] In contrast, the second plot shows that for MFVI, there is no significant difference between the forward and backward variants. This is an interesting result because, as shown above, the forward algorithms mimic Policy Iteration and they provide a tighter approximation. The third plot compares MFVI to BP showing that MFVI has poor performance in some problems and forward BP dominates in all problems. Finally, we can show (see full paper) that the exponentiated variant of MFVI can be captured in our framework by conditioning on all reward and cumulative reward variables. The bottom plot compares this variant to standard MFVI. We see that the performance improves in two domains but the exponentiated variant is still dominated by forward BP.

Exploring the performance of MFVI We believe that the main reason for the failure of MFVI is due to interaction between the flexibility that the mean field approximation allows with many state variables, and the sensitivity to ordering of updates due to local optima. To explore this we performed several additional experiments. In the first we introduce a new variant algorithm, MFVI-NoS, which does not update the marginal distribution over state variables, i.e. keeps them at the initialized value of 0.5. Results for two domains are shown in the top half of the third column of Figure 1. We see that while the NoS variant restricts the algorithm it improves the performance in these domains (this does not happen in all domains). Another view of this phenomenon is given by the relative contribution of each group of variables to the increase in the ELBO during updates of variational parameters. The bottom half of the third column of Figure 1 visualizes this for the MFVI and MFVI-NoS variants in one problem. We see that for MFVI the largest increase in ELBO is contributed by adjusting state variables and the NoS variant increases the share of other variables. We further explore this in the full paper using an artificial problem, showing that in this case limiting the flexibility of MFVI can lead to better posterior, that MFVI is sensitive to a choice of which subset of state variables is updated, and in addition to the order of updates.

6. Collapsed State Variational Inference

Motivated by the analysis above, we propose a new algorithm for variational inference in planning. Instead of treating all the latent nodes in the DBN in the same manner and computing approximate distributions over all these variables, the algorithm focuses on the action variables and effectively marginalize out other terms to achieve a tighter ELBO. This type of approach is known as collapsed variational inference, which has been shown to be effective in models where the marginalization can be done analytically (e.g., Teh et al. (2006)) but for planning one has to resolve additional computational challenges as we show

[3] While our focus is on the quality of approximation it is worth noting that Cui et al. (2018) have shown that with a directed model (equivalent to the Forward Framework with no downstream evidence as in our case), LBP converges in one iteration. Thus the forward algorithm is also faster.

below. Specifically we propose to use the following provably tighter ELBO

$$\log p_\theta(c_T = 1) = \log \mathbb{E}_{p_{\theta(A)}}[p(c_T = 1, A)] \geq \mathbb{E}_{q_\phi}[\log \frac{p_\theta(c_T = 1, A)}{q_\phi(A)}]. \tag{6}$$

Here we have the same factorized transitions and policy distribution. However, we do not compute approximation distributions over state, reward, and cumulative reward variables. With mean field, the standard solution (Bishop, 2006) yields the update equation

$$\log q_\phi(a_t^l) \propto \mathbb{E}_{q_\phi \setminus a_t^l(A)} \log p_\theta(c_T = 1, A) = \mathbb{E}_{q_\phi \setminus a_t^l(A)} \log g_\theta(A) \tag{7}$$

where

$$g_\theta(A) = \mathbb{E}_{S,R,C_{\setminus T}}[p_\theta(A, S, R, C, c_T = 1)]. \tag{8}$$

The tighter approximation appears to yield an infeasible update, because A is entangled in $g()$ and we must perform an explicit marginalization in $g()$ for each update.

We next show how the update equation can be approximated via sampling. The key is to first extract $p_\theta(A)$ from the expectation. We therefore have:

$$\log g_\theta(A) = \log p_\theta(A) + \log \mathbb{E}_{S,R,C_{\setminus T}}[p_\theta(S, R, C, c_T = 1|A)]. \tag{9}$$

Recall that $p_\theta(A)$ is a product of independent terms. This implies that the first part can be substituted with $\log p_\theta(a_h^l)$ since all other terms are constants w.r.t the variable of interest in (7) and they will vanish in the normalized update of $q_\phi(a_t^l)$. The second part is conditioned on A and does not include $p(A)$ terms. Its expectation can be estimated through sampling. In particular, sampling can be intuitively done as follows: keeping a_t^l fixed, sample the action sequence from approximate distribution $q_{\phi \setminus a_t^l}(A)$. Then complement this by sampling values for s_t, r_t, c_t nodes, including c_T. The resulting values for c_T are generated from the correct distribution and the average over c_T gives an estimate of the expectation. Since we are using sampling and averaging inside the logarithm this yields biased estimates for updates, but this type of biased estimates has been shown to work in other cases in machine learning (e.g., (Wei et al., 2021)) and it can be mitigated by taking sufficient samples. It is interesting to note from the above update that the policy distribution serves as a weight bias in the action update procedure. Algorithm 3 summarizes the update procedure.

Performance of CSVI For CSVI our implementation uses the same parameters as in MFVI except that we make at most 10 variational updates. The sample sizes are set to $M_1 = 20$ and $M_2 = 50$. Results are shown in the middle column of Figure 1. Considering the plots from top to bottom we observe that there is no significant difference between forward and backward variants of CSVI and that CSVI is significantly better than MFVI. The third plot shows that the exponentiated reward variant does not improve the performance of CSVI. This suggests that the improvement over exponential variant for MFVI is due to stabilizing the optimization rather than presenting a better objective. The fourth plots shows that the performance of CSVI is competitive with forward BP and therefore CSVI provides state of the art performance in stochastic planning.

Algorithm 3 Collapsed State Variational Inference

```
for $t = 1, 2, \ldots, T$ do
    for $l = 1, 2, \ldots, N$ do
        for value of action variable $l$ at time $t$ fixed to be $0, 1$ do
            for action sequence sample index $i = 1, \ldots, M_1$ do
                Sample action sequence $A = a_1, \ldots, a_T$ from $q_\phi$
                for trajectory sample $= 1, \ldots, M_2$ do
                    Sample and record cumulative reward variable $c_T$ from $g_\theta(A)$
                end for
                Estimate $\hat{p}_i = \#(c_T = 1)/M_2$
            end for
            Calculate $\log q_\phi(a_t^l) \propto \log p_\theta(a_t^l) + \sum_i (\log \hat{p}_i)/M_1$
        end for
        Update $q_\phi(a_t^l)$ by calculating the normalizing factor
    end for
end for
```

7. Conclusion

In this paper we provide a unified scheme that categorizes many previous approaches along two dimensions, using either forward or backward reasoning and choosing an approximation scheme. Specifically, we focus on belief propagation and mean field variational inference as the approximation choices. In this context, we illustrate the advantage of Forward Loopy BP as providing the best performance. Algorithms based on MFVI perform poorly in some domains. They are improved by exponential reward weighting but not sufficiently so. An experimental analysis points to sensitivity of the optimization as a source for this failure. Motivated by this analysis we propose a novel algorithm, Collapsed State Variational Inference, which provides a tighter variational approximation, and while being computationally demanding it performs competitively with Forward Loopy BP. The results highlight that while BP has been less in focus in recent years, it provides a strong baseline for stochastic planning. It also shows the importance of focusing variational approximations on variables of interest as done in CSVI and the potential for developing strong variational algorithms for planning. These observations suggest interesting directions for future work including developing efficient variants of CSVI, using amortized variational inference in planning to improve CSVI, alternative schemes to capture the posterior distributions in VI, and developing tighter approximations and optimization algorithms through BP methods.

Acknowledgements

This work was supported by NSF under grant IIS-1906694 and grant IIS-2002393. Some of the experiments in this paper were run on the Big Red 3 computing system at Indiana University, supported in part by Lilly Endowment,Inc., through its support for the Indiana University Pervasive Technology Institute.

References

C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 33–42, 1998.

Q. Cheng, Q. Liu, F. Chen, and A. T. Ihler. Variational planning for graph-based MDPs. *Advances in Neural Information Processing Systems*, 26, 2013.

G. F. Cooper. A method for using belief networks as influence diagrams. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 1988.

H. Cui, R. Marinescu, and R. Khardon. From stochastic planning to marginal MAP. *Advances in Neural Information Processing Systems*, 31, 2018.

H. Cui, T. Keller, and R. Khardon. Stochastic planning with lifted symbolic trajectory optimization. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 119–127, 2019.

P. Dayan and G. E. Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

C. Domshlak and J. Hoffmann. Fast probabilistic planning through weighted model counting. In *International Conference on Automated Planning and Scheduling*, pages 243–252, 2006.

T. Furmston and D. Barber. Variational methods for reinforcement learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 241–248, 2010.

T. Furmston and D. Barber. Efficient inference in markov control problems. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 221–229, 2011.

J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. SPUDD: stochastic planning using decision diagrams. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 279–288, 1999.

H. J. Kappen, V. Gómez, and M. Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182, 2012.

I. Kiselev and P. Poupart. POMDP planning by marginal-MAP probabilistic inference in generative models. In *Proceedings of the 2014 AAMAS Workshop on Adaptive Learning Agents*, 2014.

F. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

A. Kumar, S. Zilberstein, and M. Toussaint. Probabilistic inference techniques for scalable multiagent decision making. *Journal of Artificial Intelligence Research*, 53:223–270, 2015.

J. Lee, R. Marinescu, and R. Dechter. Submodel decomposition bounds for influence diagrams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

S. Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Q. Liu and A. Ihler. Belief propagation for structured decision making. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 523–532, 2012.

K. Murphy and Y. Weiss. The factored frontier algorithm for approximate inference in DBNs. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 378–385, 2001.

G. Neumann. Variational inference for policy search in changing situations. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 817–824, 2011.

D. Nitti, V. Belle, and L. D. Raedt. Planning in discrete and continuous markov decision processes by probabilistic programming. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 327–342. Springer, 2015.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann, 1988.

N. Peyrard and R. Sabbadin. Mean field approximation of the policy iteration algorithm for graph-based markov decision processes. In *17th European Conference on Artificial Intelligence August 29–September 1, 2006, Riva del Garda, Italy*, pages 595–599, 2006.

R. Sabbadin, N. Peyrard, and N. Forsell. A framework and a mean-field algorithm for the local control of spatial processes. *International Journal of Approximate Reasoning*, 53(1):66–86, 2012.

S. Sanner et al. Relational dynamic influence diagram language (rddl): Language description. *Unpublished ms. Australian National University*, 32:27, 2010.

Y. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. *Advances in neural information processing systems*, 19, 2006.

M. Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th annual international conference on machine learning*, pages 1049–1056, 2009.

M. Toussaint and A. J. Storkey. Probabilistic inference for solving discrete and continuous state markov decision processes. In *Machine Learning, Proceedings of the Twenty-Third International Conference*, volume 148, pages 945–952, 2006.

Y. Wei, R. Sheth, and R. Khardon. Direct loss minimization for bayesian predictors. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS*, 2021.

Y. Weiss. Comparing the mean field method and belief propagation for approximate inference in MRFs. In *Advanced Mean Field Methods: Theory and Practice*, pages 229–239. MIT Press, 2001.

Z. Wu and R. Khardon. Approximate inference for stochastic planning in factored spaces. *arXiv*, 2203.12139, 2022.

G. Zhou, N. Kumar, A. Dedieu, M. Lázaro-Gredilla, S. Kushagra, and D. George. PGMax: Factor Graphs for Discrete Probabilistic Graphical Models and Loopy Belief Propagation in JAX. *arXiv preprint arXiv:2202.04110*, 2022.