Relax, Compensate and then Recover

Arthur Choi and Adnan Darwiche

Computer Science Department,
University of California, Los Angeles, USA
{aychoi,darwiche}@cs.ucla.edu

Abstract. We present in this paper a framework of approximate probabilistic inference which is based on three simple concepts. First, our notion of an approximation is based on "relaxing" equality constraints, for the purposes of simplifying a problem so that it can be solved more readily. Second, is the concept of "compensation," which calls for imposing weaker notions of equality to compensate for the relaxed equality constraints. Third, is the notion of "recovery," where some of the relaxed equality constraints are incrementally recovered, based on an assessment of their impact on improving the quality of an approximation. We discuss how this framework subsumes one of the most influential algorithms in probabilistic inference: loopy belief propagation and some of its generalizations. We also introduce a new heuristic recovery method that was key to a system that successfully participated in a recent evaluation of approximate inference systems, held in UAI'10. We further discuss the relationship between this framework for approximate inference and an approach to exact inference based on symbolic reasoning.

1 Introduction

In this survey-style paper, we highlight two frameworks for probabilistic inference, one for exact probabilistic inference and the second for approximate probabilistic inference.

The framework for *approximate probabilistic inference* is based on performing exact inference on an approximate model, which is obtained by *relaxing* equivalence constraints in the original model. The framework allows one to improve the resulting approximations by *compensating* for the relaxed equivalences, which can be realized through the enforcement of weaker notions of equivalence. One can improve the approximations even further by *recovering* equivalence constraints in the relaxed and compensated model. Interestingly, the influential algorithm of loopy belief propagation [34, 37] can be characterized in terms of relaxing and compensating for equivalence constraints [7, 9]. The third step of recovery can be used here to find even better approximations, which correspond to some forms of generalized belief propagation.

The framework for *exact probabilistic inference* that we shall survey is based on reducing probabilistic inference into a process of enforcing certain properties on propositional knowledge bases. In particular, if we encode the Bayesian network into an appropriate logical form [17], the problem of exact probabilistic inference can be reduced to one of compiling the knowledge base into a tractable form that satisfies two properties, known as *decomposability* and *determinism* [14, 16, 20]. The resulting framework has an ability to perform exact probabilistic inference on models that are beyond the reach of more traditional algorithms for exact inference [18, 4, 3]. Moreover, this framework can significantly increase the effectiveness of the framework for approximate inference that we mentioned earlier, which requires an exact inference engine, although such integration has not been realized in practice yet.

These two frameworks for exact and probabilistic inference have been validated in practice: notably, they have formed the basis of systems successfully employed in (respectively) exact and approximate inference evaluations conducted by the Uncertainty in Artificial Intelligence (UAI) community in 2006, 2008 and 2010 [2, 19, 23]. We survey these two frameworks in this paper, highlighting some of the relevant results along the way. Moreover, we shall introduce a new recovery heuristic that was particularly key to the success of a solver employed in the UAI'10 approximate inference evaluation. We finally conclude with a discussion on the pending advances that are needed to allow one to effectively use the framework for exact inference as a basis for the one on approximate inference, which we call *relax, compensate and then recover.*

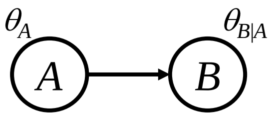
2 Exact probabilistic inference

Consider the Bayesian network in Figure 1(a). There are a variety of algorithms available for performing exact inference in such networks, including variable elimination [39, 21], the jointree algorithm [27, 30] and recursive conditioning [15], which have running time complexities that are generally exponential in a model's treewidth. In the first part of this paper, we will be more concerned with another approach to exact probabilistic inference that is based on *knowledge compilation*, which among other advantages, can be used to tackle networks having large treewidth.

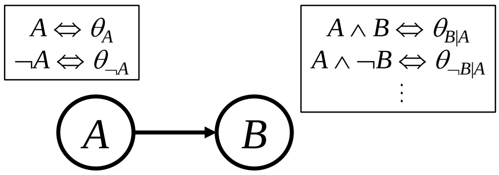
Consider Figure 1(b) where we "encode" the Bayesian network of Figure 1(a) as a propositional knowledge base, in conjunctive normal form (CNF), augmented with weights. Figure 1(c) further shows an equivalent, but more useful, representation in negation normal form (NNF), which is a DAG whose internal nodes are labeled with disjunctions and conjunctions, and whose leaf nodes are labeled with literals or the constants true and false.

Previous work has shown that if one has an ability to convert CNF to an NNF satisfying two key properties, *decomposability* and *determinism*, then one has the ability to perform probabilistic inference efficiently on a wide range of probabilistic representations [18, 4, 3]. Moreover, the latter can be performed in time *linear* in the NNF size, assuming the NNF satisfies decomposability and determinism, such as the NNF in Figure 1(c).[1] In this case, the NNF represents

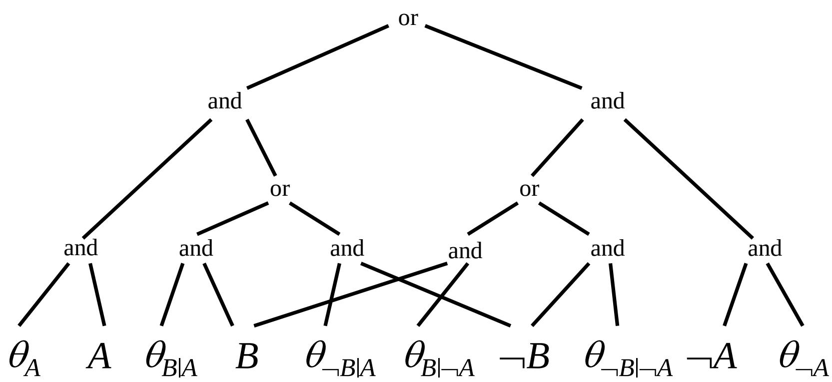
[1] An NNF is *decomposable* (called a DNNF) iff each of its conjunctions is decomposable, i.e., its conjuncts share no variables. A DNNF is *deterministic* (called a



(a) A simple Bayesian network.



(b) A Bayesian network as a weighted CNF (weights on literals θ not shown). We use $\alpha \Leftrightarrow \beta$ as shorthand for the two clauses $\neg\alpha \vee \beta$ and $\alpha \vee \neg\beta$.



(c) NNF that satisfies *decomposability* and *determinism*.

Fig. 1. Bayesian network inference as weighted model counting.

an *arithmetic circuit* for the Bayesian network: the leaf nodes input the network parameters and any given evidence, the or-nodes represent additions, and the and-nodes represent multiplications. A simple upward evaluation pass then computes the probability of evidence.

To illustrate this formulation more concretely, we note that the CNF Δ in Figure 1(b) "encodes" the Bayesian network of Figure 1(a) as follows. For any event α, the weighted model count of $\Delta \wedge \alpha$ equals the probability of event α according to the Bayesian network. Since the NNF in Figure 1(c) is equivalent to the CNF Δ, probabilistic inference on the Bayesian network can now be performed through weighted model counting on the NNF (a linear-time operation). Hence, to implement probabilistic reasoning, all one needs computationally is an ability to enforce decomposability and determinism, since Bayesian networks can be easily encoded as CNFs [4, 3].

---

d-DNNF) iff each of its disjunctions is deterministic, i.e., if each pair of its disjuncts is mutually exclusive. For more on decomposability and determinism, and knowledge compilation in general, see [14, 16, 20].

This approach has a number of advantages, as compared to more traditional approaches to exact probabilistic inference, and these advantages have been realized in practice. For example, systems based on this approach have successfully participated in the two evaluations of exact probabilistic inference conducted for UAI'06 and UAI'08 (the UAI'10 evaluation focused on approximate inference systems, which we consider again in the following section).

A system that utilizes the approach we described was submitted to UAI'06 by the Automated Reasoning Group at UCLA, and was the only system to solve all given benchmarks [2]. At UAI'08, it was again the only one to solve a large class of very difficult networks that are synthesized from relational models [19].

The key advantage of the described approach is that it *exploits local structure* (i.e., the structure of parameters that quantify the network) in addition to the *global structure* (i.e., network topology). The relational network suite mentioned earlier in connection to UAI'08 has an average clique size that is greater than 50. Given the current state of the art in exact probabilistic inference, any method that does not exploit local structure would have to be exponential in treewidth [4, 3] and is therefore not feasible on such networks.

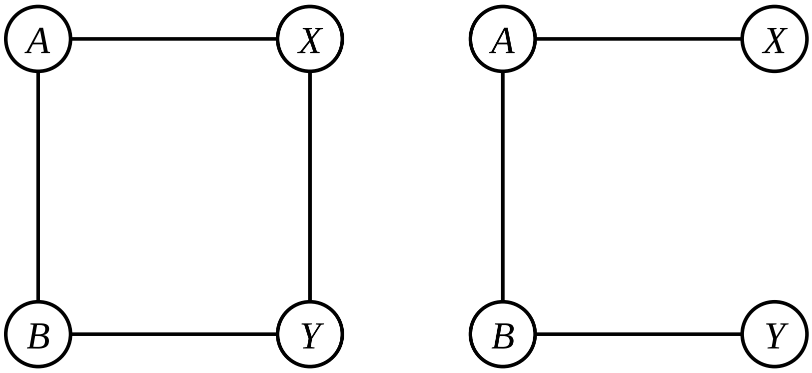
We remark that the above approach to exact inference applies equally to different probabilistic representations, such as Bayesian networks, Markov networks, and factor graphs. The only difference between all these representations is that each requires its own "encoding" into a propositional knowledge base. Hence, to apply this approach practically to a new probabilistic representation all one needs to do is provide an "encoder" of that representation into an appropriate propositional knowledge base. This approach has been successfully applied in other domains as well. For example, it has been recently used by database researchers to identify tractable classes of probabilistic databases, by showing that the knowledge bases they lead to have tractable compilations that satisfy decomposability and determinism [28].

Finally, the ACE system, which was successfully employed at the UAI'06 and UAI'08 inference evaluations as we just described, is available online.[2] It is based on two components: An encoder of Bayesian networks into CNFs, and a compiler called C2D which converts CNFs into NNFs that satisfy decomposability and determinism.[3] Indeed, these are two of the key dimensions of applying this approach to exact probabilistic inference in practice: (1) efficiently encoding the probabilistic representations to CNF, and (2) developing more efficient compilers for enforcing decomposability and determinism. More recently, a new open-source compiler, called DSHARP, was released by the University of Toronto,[4] which is claimed to be one-to-two orders of magnitude more efficient than C2D on some benchmarks.

[2] The ACE system is available at `http://reasoning.cs.ucla.edu/ace/`

[3] The C2D compiler is available at `http://reasoning.cs.ucla.edu/c2d/`

[4] The DSHARP system is available at `http://www.haz.ca/research/dsharp/`



(a) Pairwise MRF with equivalence constraint $X \equiv Y$.

(b) Relaxing the equivalence constraint of 2(a).

Fig. 2. Relaxing an equivalence constraint in a pairwise MRF.

3 Approximate Probabilistic Inference

Consider now the problem of *approximate* probabilistic inference. In particular, consider formulating approximate inference as *exact* inference in an *approximate* model.[5] In this section, we shall present a particular perspective on this approach that we call *relax, compensate and then recover*, which is quite special in a number of ways:

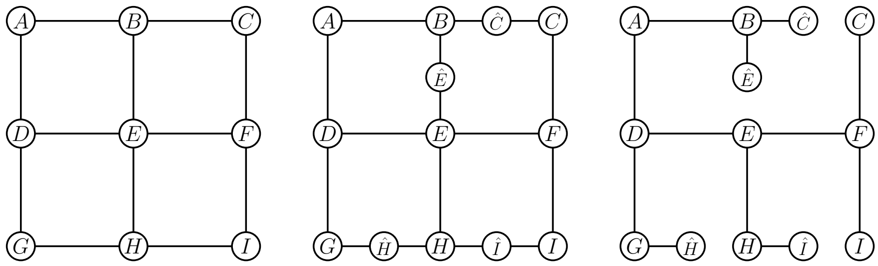
– It defines the approximate model specifically as the result of *relaxing* equivalence constraints in the original model and then *compensating* for the relaxed equivalences.
– It subsumes a number of well-known approximation schemes, including the influential algorithm of loopy belief propagation [34, 37, 7, 9].
– It was successfully employed in the UAI'10 evaluation of approximate probabilistic inference, where it was the leading system in two of the most time-constrained categories evaluated [23].

We will now provide a concrete description of each part of this relax, compensate and then recover framework. We further conclude this section by introducing a critical technique that was introduced for the UAI'10 evaluation.

3.1 Relax

Consider Figure 2(a), which depicts a pairwise MRF with an edge that represents an equivalence constraint between variables X and Y. Relaxing this equivalence

[5] This approach includes mean-field and other variational approximations [29, 26, 25], but as we shall highlight, it also includes approximations such as loopy belief propagation [34, 37] and mini-buckets [22] as well.



(a) A pairwise MRF. (b) A pairwise MRF augmented with equivalence constraints $X \equiv \hat{X}$. (c) Relaxing the equivalence constraints of 3(b).

Fig. 3. Introducing and then relaxing equivalence constraints.

constraint amounts to dropping this edge and its corresponding potential from the model. The resulting model in Figure 2(b) is sparser (a tree in particular) and, hence, is more amenable to exact inference algorithms.

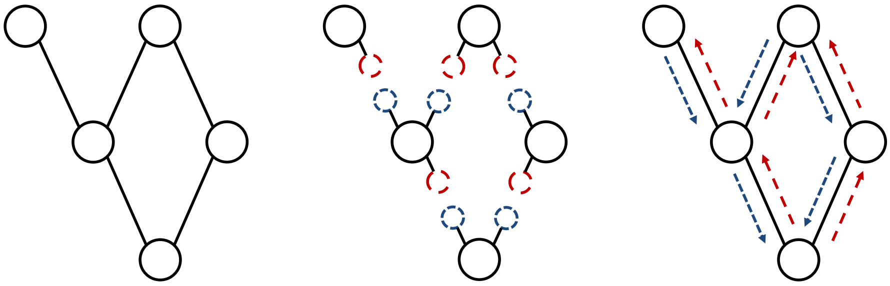
More generally, any pairwise MRF can be made as sparse as needed by relaxing enough equivalence constraints. Namely, if the original model does not contain equivalence constraints, one can always introduce them, as shown in Figure 3(b), to relax them later, as in Figure 3(c). For example, we can replace an edge X—Y with a chain X—$\hat{Y}$—Y where the original potential $\psi(X,Y)$ is now over variables X and $\hat{Y}$, and where potential $\psi(Y,\hat{Y})$ represents an equivalence constraint $Y \equiv \hat{Y}$ (where variable Y and $\hat{Y}$ have the same states).

In sum, one has fine control over the sparsity (treewidth) of a given model through only the relaxation of equivalence constraints, which in turn translates to fine control over the amount of work that exact inference algorithms will need to perform on the relaxed model. This is a general technique for controlling the difficulty of exact inference on all kinds of models, whether probabilistic or even symbolic as shown in [12, 13]. We shall highlight a few such examples later in this section.

3.2 Compensate

We have shown in [6] that performing exact inference on *relaxed* models leads to approximations that coincide with those computed by the mini-bucket elimination algorithm [22].

One can obtain better approximations, however, by *compensating* for a relaxed equivalence $X \equiv Y$. Broadly defined, compensation refers to imposing a weaker notion of equivalence in lieu of the relaxed equivalence (for example, ensuring that X and Y have the same probability in the approximate model). In pairwise MRFs, compensating for an equivalence constraint $X \equiv Y$ is done by adding additional potentials $\psi(X)$ and $\psi(Y)$, while choosing the values of these potentials carefully. Other models, such as Bayesian networks, suggest dif-



(a) A pairwise MRF. (b) Relaxing all equivalence constraints. (c) Messages of LBP.

Fig. 4. Compensating for relaxations, and loopy belief propagation.

ferent mechanics for compensation, but the goal is always the same: Imposing a weaker notion of equivalence between variables X and Y. We have proposed a number of compensation schemes in previous work and with respect to different representations [5, 7, 8, 13, 12].

One particular scheme, however, stands out. This scheme is derived while assuming that relaxing the equivalence constraint $X \equiv Y$ will split the model into two independent components, one containing variable X and another containing variable Y. Under this assumption, one can find unique values of potentials $\psi(X)$ and $\psi(Y)$ that guarantee the correctness of all marginal probabilities over single variables. In particular, we should assign the new auxiliary potentials $\psi(X)$ and $\psi(Y)$ so that the resulting model satisfies the following condition:

$$Pr'(X) = Pr'(Y) = \alpha\psi(X)\psi(Y) \tag{1}$$

where Pr' denotes the distribution induced by the simplified model. This condition states that after relaxing an equivalence constraint $X \equiv Y$, the marginals $Pr'(X)$ and $Pr'(Y)$ (and the marginals constructed from the new potentials) should at least be equivalent in the simplified model, a weaker notion of equivalence [7, 9].

In a tree, every equivalence satisfies the desired assumption (that it splits a model into two independence components) and, hence, this compensation scheme leads to exact node marginals in this case. But when applied to arbitrary models, one is no longer guaranteed correctness (which is to be expected). The surprise, however, is that if *all* equivalence constraints are relaxed from an arbitrary model (see Figure 4), the approximations returned under this compensation scheme correspond precisely to the ones computed by the influential algorithm of loopy belief propagation [7, 9].

The weaker notion of equivalence given by Equation 2 happens to be equivalent to the following local property on the parameters $\psi(X)$ and $\psi(Y)$ introduced for the compensation:

$$\psi(X) = \alpha \frac{\partial Z'}{\partial \psi(Y)} \qquad \text{and} \qquad \psi(Y) = \alpha \frac{\partial Z'}{\partial \psi(X)} \tag{2}$$

where Z' denotes the partition function (normalizing constant) of the compensated model, and α is a normalizing constant for the parameters $\psi(X)$ and $\psi(Y)$. Equation 2 can also be viewed as update equations, suggesting an iterative method that searches for parameters $\psi(X)$ and $\psi(Y)$ [7]. Starting with an initial approximation at iteration $t = 0$ (say, with uniform parameters), we can compute parameters $\psi_t(X)$ and $\psi_t(Y)$ for an iteration $t > 0$ by performing exact inference in the approximate network of the previous iteration $t - 1$, computing the right-hand sides of Equation 2. We repeat this process until all of our parameter updates converge (if ever), in which case we have a model satisfying Equation 2, and equivalently Equation 1.

In Bayesian Networks. As we described for pairwise Markov random fields, we can introduce and then relax equivalence constraints to simplify a Bayesian network. In particular, we can replace any edge $U \rightarrow X$ with a chain $U \rightarrow \hat{U} \rightarrow X$, where $U \rightarrow \hat{U}$ denotes an equivalence constraint $U \equiv \hat{U}$.

To *compensate* for the relaxation of an equivalence constraint $U \equiv \hat{U}$ (i.e., the deletion of the edge $U \rightarrow \hat{U}$), we first add an observed variable $\hat{S}$ as a child of U. We then need to specify the CPTs $\Theta_{\hat{U}}$ and $\Theta_{\hat{S}|U}$ for variables $\hat{S}$ and $\hat{U}$, so that the resulting network satisfies a weaker notion of equivalence. A condition analogous to the one given in Equation 1 will also produce an approximation corresponding to loopy belief propagation, but for Bayesian networks [7].

In Factor Graphs. We can introduce and then relax equivalence constraints in factor graphs as well. For a factor graph with a factor $\psi(\mathbf{X})$, we can replace a variable X in the set $\mathbf{X}$ with a clone $\hat{X}$, and then introduce a factor $\psi(X, \hat{X})$ representing an equivalence constraint $X \equiv \hat{X}$. When we now relax this equivalence constraint, we are effectively disassociating the factor $\psi(\mathbf{X})$ from the variable X, simplifying the topology of the factor graph. To compensate for this relaxation, we can introduce auxiliary factors $\psi(X)$ and $\psi(\hat{X})$, which we use to enforce a weaker notion of equivalence like the one given in Equation 1. In this case, we again identify an approximation corresponding to loopy belief propagation, now for factor graphs.

In Weighted CNF Models. The same approach to approximate probabilistic inference can be applied to tasks in logical reasoning as well. Consider for example the following weighted CNF over three variables A, B and C:

$$\{(a \vee b, w_1), (\bar{b} \vee c, w_2), (\bar{c} \vee d, w_3)\}.$$

where each clause is assigned a non-negative weight w_i. It is also straightforward to augment such models to an equivalent one where equivalence constraints can be relaxed. For example, we can replace the variable C appearing as a literal in the third clause with a clone variable C', and add an equivalence constraint $C \equiv C'$, giving us:

$$\{(a \vee b, w_1), (\bar{b} \vee c, w_2), (\bar{c}' \vee d, w_3), (C \equiv C', \infty)\}.$$

Here, the equivalence constraint $C \equiv C'$ is a hard constraint (having "infinite" weight) that ensures C and C' take the same value. When we remove, in this example, the equivalence constraint $C \equiv C'$, we have a relaxed formula composed of two independent models: $\{(a \vee b, w_1), (\bar{b} \vee c, w_2)\}$ and $\{(\bar{c}' \vee d, w_3)\}$.

We can introduce four auxiliary clauses into the relaxation:

$$\{(c, w_4), (\bar{c}, w_5), (c', w_6), (\bar{c}', w_7)\}$$

with four auxiliary weights that we can use to compensate for the relaxation of the equivalence constraint $C \equiv C'$. How we compensate for the relaxation of course depends on the query that we are interested in. In the case of weighted Max-SAT, it is possible to specify a weaker notion of equivalence so that the weighted Max-SAT solution of the simpler, compensated model is an upper-bound on the weighted Max-SAT solution of the original [13].

3.3 Recover

Now that we have discussed the *relaxation* and *compensation* components of this framework — what about *recovery?*

As described thus far, our framework leaves out the question of which equivalence constraints to relax. Answering this question intelligently typically requires inference, which is not possible — otherwise, we would not be thinking about relaxing equivalence constraints to start with. Instead of thinking about relaxing constraints, however, our framework calls for thinking about recovering them. That is, we can start by relaxing too many equivalence constraints, to reach, say, a tree model. We can then perform inference on the resulting compensated model to try to identify those equivalence constraints whose relaxation has been most damaging, and then *recover* these constraints. The recovery process can be incremental. That is, as we recover more and more equivalence constraints, we expect our approximate model to improve, leading to better decisions about which equivalence constraints to recover. This incremental process stops when we have recovered enough constraints to make the model inaccessible to exact inference algorithm.

We have proposed a number of recovery heuristics in previous work [7, 11, 10, 9]. In the following, we highlight a few of these heuristics. The heuristics that we do highlight have been designed for the particular form of compensation based on the condition of Equation 1, which characterizes loopy belief propagation when we have relaxed enough equivalence constraints to simplify the model to a tree [7]. Thus, when we start with a tree model, we start with loopy belief propagation as a base approximation, hopefully identifying improved approximations as we recover equivalence constraints.[6] We shall also introduce a new heuristic that was developed for and successfully employed at the latest evaluation of approximate inference systems conducted by UAI'10 [23].

Mutual Information. Remember the case where we relax an equivalence constraint $X \equiv Y$ that splits a model into two independent components: one containing variable X and the other containing variable Y. In this case, we can effectively compensate for the relaxation (by enforcing the weaker notion of equivalence in Equation 1), and guarantee the correctness of the marginal probabilities for each variable. Thus, we would see no benefit in recovering an equivalence constraint that had split a model into two independent components. On the other hand, if variables X and Y remain highly dependent on each other, after relaxing the constraint $X \equiv Y$, then our compensation scheme may in fact be over-compensating, since it expected that they would be independent.

Thus, we proposed in [7] to use the mutual information between variables X and Y as a way to *rank* equivalence constraints to recover. More specifically, we compute the mutual information between X and Y in the compensated model and recover first those equivalence constraints with highest mutual information. After recovering a few equivalence constraints, we can continue by re-scoring and re-ranking the other equivalence constraints, recovering them in an adaptive fashion. In [7], we found empirically that it was possible to identify a *small* set of equivalence constraints that can effectively improve the quality of an approximation without impacting much the complexity of inference.

Focused Recovery. The mutual information heuristic we just described is based on a property (that relaxing an equivalence constraint splits the model into two) that guarantees exact marginals for every variable in the model. This approach can provide good approximations (even exact) for many variables, but may still provide only poor approximations for others. One must then ask if this is the ideal approach when one is interested only in a particular query variable. Indeed, from query-to-query, ones focus may change from one sub-model to another, while varying observations may render different parts of the model irrelevant. Ideally, one would like to target the approximation so as to maximize the accuracy of the probabilities one is truly interested in, giving less weight to those parts of the model that are only weakly relevant to the query at hand.

In [10], we proposed a more refined mutual information heuristic that focuses recovery on targeted query variables. It is based on simple independence conditions that can guarantee the exactness of a specific variable's marginal in

[6] Each time we recover constraints, we update the compensation so that the resulting network again satisfies Equation 1. In this case, the resulting approximations correspond to iterative joingraph propagation approximations [37, 1, 32, 7]. From this perspective, our *relax, compensate and then recover* framework can also be seen as a way to find good joingraphs, and thus as a concrete way to design iterative joingraph and generalized belief propagation approximations.

the case where a single equivalence constraint is relaxed. This in turn leads to a more refined mutual information heuristic based on the query variable Q and the variables X and Y of the equivalence constraint $X \equiv Y$ relaxed. When used in an adaptive recovery process, we found empirically that a focused approach to approximation can indeed be more effective than an unfocused approach.

Residual Recovery. Finally, we introduce now a new heuristic that was critical to the success of a system based on this relax, compensate and then recover framework, at the UAI'10 approximate inference evaluation [23]. Our solver won first place in the most demanding categories of 20-second response time, for both probability of evidence and marginal probabilities. It also won second place in four other categories, for 20-minute and 1-hour response time (by a very thin margin). The system used was quite simple, as we have described thus far: relax enough equivalence constraints to reach a tree, and then incrementally recover equivalence constraints until the time allotted is up.

The recovery heuristic, which we refer to as "residual recovery", was inspired in part by the residual belief propagation algorithm [24]. Consider that the primary failure mode of loopy belief propagation is its failure to converge in certain cases. In these cases, even if loopy belief propagation is made to converge (using convergent alternatives), it has been observed that the quality of the approximation is often not good anyways; for more on loopy belief propagation and convergence, see e.g., [38, 33, 24]. On the other hand, when loopy belief propagation is able to converge naturally, then the quality of the approximation tends to be good, at least in practice.

Our "residual recovery" heuristic thus seeks to recover first those equivalence constraints that cause the most difficulty in the iterative process of compensation. Namely, we measure how close each equivalence constraint $X \equiv Y$ is to satisfying Equation 1, during the iterative process of compensation.[7] We then simply recover first the most problematic equivalence constraints, the ones that are furthest from convergence.

There are a number of advantages of the residual recovery heuristic:

- It encourages faster convergence during the iterative process of compensation (analogous to message passing in loopy belief propagation and iterative join-graph propagation). This translates directly into improved efficiency (fewer iterations required).
- Assuming that improved convergence behavior indicates improved approximation quality, we can expect our heuristic to have a positive impact on approximation quality as well.
- The heuristic is extremely efficient, and introduces little overhead to the overall relax, compensate and then recover process. In fact, the computations required to rank equivalence constraints are already required by the iterative compensation process to determine convergence.

[7] In the UAI'10 evaluation, we used a 3-way symmetric KL–divergence to measure how close an equivalence constraint was to converging.

The original mutual information heuristic we discussed, while still tending to identify better equivalence constraints to recover (in terms of the accuracy of the resulting approximation), is still relatively expensive [11]. A good, but more efficient heuristic, is more critical in time-constrained situations such as the strict 20-second response time categories that our system won in the UAI'10 evaluation [23].

4 Discussion

The performance of our system in the UAI'10 approximate inference evaluation was clearly a strong demonstration of the practical effectiveness of the *relax, compensate and then recover* framework. The latest public release of the SamIam system is available at `http://reasoning.cs.ucla.edu/samiam/`, which includes an implementation of this framework based on the mutual-information recovery heuristic described earlier; the "residual recovery" heuristic used in UAI'10 will be included in an upcoming release.

The system we employed in the UAI'10 approximate inference evaluation used only standard jointree algorithms for exact inference in the simplified model. Since the *relax, compensate and then recover* framework requires only a black-box for exact reasoning in the simplified model, one can in principle employ other inference engines as well. The ace system is particularly attractive for this purpose, given its strong performance in the UAI'06 and UAI'08 exact inference evaluations. However, to fully exploit this system, or other systems based on the same principle, in this framework for approximate inference, one requires some more advances that we are currently pursuing.

Consider for example our more recent efforts on knowledge compilation, based on a refinement of the decomposability property called *structured* decomposability [35, 36]. The property allows for an efficient conjoin operation between knowledge bases that satisfy structured decomposability. The *relax, compensate and then recover* framework stands to benefit immensely from this conjoin operation, as recovery corresponds to a process where we conjoin equivalence constraints with an existing knowledge base. Thus, it allows one to smoothly recover equivalence constraints without having to re-initiate exact inference each time an equivalence constraint is recovered.

The suggested integration between the two inference approaches that we discussed in this paper could also admit more sophisticated and accurate approximations. Using a black-box inference engine that is exponential in treewidth, such as the jointree algorithm, limits one's ability to recover equivalence constraints as one is limited to approximate models that have low enough treewidth. On the other hand, one can direct the *relax, compensate and then recover* process towards approximate models that are efficiently *compilable* by systems such as ace even if they have a large treewidth. This is the subject of current work.

References

1. Aji, S.M., McEliece, R.J.: The generalized distributive law and free energy minimization. In: Proceedings of the 39th Allerton Conference on Communication, Control and Computing. pp. 672–681 (2001)
2. Bilmes, J.: Results from the evaluation of probabilistic inference systems at UAI-06 (2006), `http://ssli.ee.washington.edu/~bilmes/uai06InferenceEvaluation/results`
3. Chavira, M., Darwiche, A.: On probabilistic inference by weighted model counting. Artificial Intelligence 172(6–7), 772–799 (April 2008)
4. Chavira, M., Darwiche, A., Jaeger, M.: Compiling relational Bayesian networks for exact inference. International Journal of Approximate Reasoning 42(1–2), 4–20 (May 2006)
5. Choi, A., Chan, H., Darwiche, A.: On Bayesian network approximation by edge deletion. In: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI). pp. 128–135. Arlington, Virginia (2005)
6. Choi, A., Chavira, M., Darwiche, A.: Node splitting: A scheme for generating upper bounds in bayesian networks. In: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI). pp. 57–66 (2007)
7. Choi, A., Darwiche, A.: An edge deletion semantics for belief propagation and its practical impact on approximation quality. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI). pp. 1107–1114 (2006)
8. Choi, A., Darwiche, A.: A variational approach for approximating Bayesian networks by edge deletion. In: Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI). pp. 80–89 (2006)
9. Choi, A., Darwiche, A.: Approximating the partition function by deleting and then correcting for model edges. In: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI). pp. 79–87 (2008)
10. Choi, A., Darwiche, A.: Focusing generalizations of belief propagation on targeted queries. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI). pp. 1024–1030 (2008)
11. Choi, A., Darwiche, A.: Many-pairs mutual information for adding structure to belief propagation approximations. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI). pp. 1031–1036 (2008)
12. Choi, A., Darwiche, A.: Relax then compensate: On max-product belief propagation and more. In: Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS). pp. 351–359 (2009)
13. Choi, A., Standley, T., Darwiche, A.: Approximating weighted Max-SAT problems by compensating for relaxations. In: Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming (CP). pp. 211–225 (2009)
14. Darwiche, A.: Decomposable negation normal form. Journal of the ACM 48(4), 608–647 (2001)
15. Darwiche, A.: Recursive conditioning. Artificial Intelligence 126(1-2), 5–41 (2001)
16. Darwiche, A.: A compiler for deterministic, decomposable negation normal form. In: Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI). pp. 627–634. AAAI Press, Menlo Park, California (2002)
17. Darwiche, A.: A logical approach to factoring belief networks. In: Proceedings of KR. pp. 409–420 (2002)

18. Darwiche, A.: A differential approach to inference in bayesian networks. Journal of the ACM 50(3), 280–305 (2003)
19. Darwiche, A., Dechter, R., Choi, A., Gogate, V., Otten, L.: Results from the probablistic inference evaluation of UAI-08 (2008), `http://graphmod.ics.uci.edu/uai08/Evaluation/Report`
20. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research 17, 229–264 (2002)
21. Dechter, R.: Bucket elimination: A unifying framework for probabilistic inference. In: Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI). pp. 211–219 (1996)
22. Dechter, R., Rish, I.: Mini-buckets: A general scheme for bounded inference. J. ACM 50(2), 107–153 (2003)
23. Elidan, G., Globerson, A.: Summary of the 2010 UAI approximate inference challenge (2010), `http://www.cs.huji.ac.il/project/UAI10/summary.php`
24. Elidan, G., McGraw, I., Koller, D.: Residual belief propagation: Informed scheduling for asynchronous message passing. In: Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence (2006)
25. Geiger, D., Meek, C., Wexler, Y.: A variational inference procedure allowing internal structure for overlapping clusters and deterministic constraints. J. Artif. Intell. Res. (JAIR) 27, 1–23 (2006)
26. Jaakkola, T.: Tutorial on variational approximation methods. In: Saad, D., Opper, M. (eds.) Advanced Mean Field Methods, chap. 10, pp. 129–160. MIT Press (2001)
27. Jensen, F.V., Lauritzen, S., Olesen, K.: Bayesian updating in recursive graphical models by local computation. Computational Statistics Quarterly 4, 269–282 (1990)
28. Jha, A., Suciu, D.: Knowledge compilation meets database theory: Compiling queries to decision diagrams. In: Proceedings of the 14th International Conference on Database Theory (2011), to appear.
29. Jordan, M.I., Ghahramani, Z., Jaakkola, T., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning 37(2), 183–233 (1999)
30. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of Royal Statistics Society, Series B 50(2), 157–224 (1988)
31. Lowd, D., Domingos, P.: Approximate inference by compilation to arithmetic circuits. In: Lafferty, J., Williams, C.K.I., Shawe-Taylor, J., Zemel, R., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23. pp. 1477–1485 (2010)
32. Mateescu, R., Kask, K., Gogate, V., Dechter, R.: Join-graph propagation algorithms. J. Artif. Intell. Res. (JAIR) 37, 279–328 (2010)
33. Mooij, J.M., Kappen, H.J.: Sufficient conditions for convergence of the sum-product algorithm. IEEE Transactions on Information Theory 53(12), 4422–4437 (2007)
34. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
35. Pipatsrisawat, K., Darwiche, A.: New compilation languages based on structured decomposability. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI). pp. 517–522 (2008)
36. Pipatsrisawat, K., Darwiche, A.: Top-down algorithms for constructing structured DNNF: Theoretical and practical implications. In: Proceedings of the 19th European Conference on Artificial Intelligence. pp. 3–8 (2010)

37. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. In: Lakemeyer, G., Nebel, B. (eds.) Exploring Artificial Intelligence in the New Millennium, chap. 8, pp. 239–269. Morgan Kaufmann (2003)
38. Yuille, A.L.: Cccp algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. Neural Computation 14(7), 1691–1722 (2002)
39. Zhang, N.L., Poole, D.: Exploiting causal independence in bayesian network inference. Journal of Artificial Intelligence Research 5, 301–328 (1996)