Soft Evidential Update via Markov Chain Monte Carlo Inference

Dominik Jain and Michael Beetz

Intelligent Autonomous Systems Group
Department of Informatics
Technische Universität München

Abstract. The key task in probabilistic reasoning is to appropriately update one's beliefs as one obtains new information in the form of evidence. In many application settings, however, the evidence we obtain as input to an inference problem may be uncertain (e.g. owing to unreliable mechanisms with which we obtain the evidence) or may correspond to (soft) degrees of belief rather than hard logical facts. So far, methods for updating beliefs in the light of soft evidence have been centred around the iterative proportional fitting procedure and variations thereof. In this work, we propose a Markov chain Monte Carlo method that allows to directly integrate soft evidence into the inference procedure without generating substantial computational overhead. Within the framework of Markov logic networks, we demonstrate the potential benefit of this method over standard approaches in a series of experiments on synthetic and real-world applications.

1 Introduction

A fundamental task in probabilistic reasoning is to compute the probability of a query q given evidence e, i.e. to compute the conditional probability $P(q \mid e)$; and one typically assumes that e is a plain fact that could be represented as a logical conjunction. On many occasions, however, the evidence that we can supply as input to an inference problem does not correspond to firm beliefs. Rather, evidence is often uncertain in the sense that it may represent a belief that either cannot be fully trusted (owing to unreliable mechanisms with which we obtain the evidence) or corresponds directly to a probability distribution that should be believed to hold.

For instance, we may regard the output of a probabilistic classifier as uncertain evidence. Surely it is more appropriate to make full use of the knowledge about the distribution over classes that such a classifier provides than to simply assume that the mode of the distribution is irrevocable fact. Moreover, it may be desirable to incorporate the (soft) beliefs that were computed by external sources of probabilistic information as input to an inference problem rather than to compute everything within a single model – given the fact that inference in all-encompassing probabilistic models is computationally infeasible. It may, therefore, be advisable to find ways of suitably factoring domains into loosely coupled probabilistic fragments and combine them as necessary. Whenever we do combine them, we will usually have to pass on probabilistic information rather than logical facts. The problem of updating beliefs in the light of soft evidence can thus be viewed as a subproblem of probabilistic information interchange. Probabilistic information interchange is particularly relevant within the context of multi-agent systems. Multiple agents that autonomously explore similar environments may want to share their probabilistic findings with other agents, and these agents will have to incorporate the received findings into their own probabilistic knowledge bases and update their beliefs accordingly [1].

In this work, we address precisely the issues pertaining to belief update that are fundamental for applications such as the ones we named above. To put our work in perspective, we first give an overview of semantically different ways in which evidential uncertainty can be interpreted, reviewing the notions of *virtual evidence* and *soft evidence* put forth in the literature (Section 2). Because the former is usually straightforward to handle, we concentrate, for the most part, on the harder problem of *soft evidential update*, i.e. the integration of firm beliefs on probability distributions (i.e. probability constraints). In Section 4, we present two solutions to this problem, one of which is essentially analogous to previous (state-of-the-art) approaches, while the other is fundamentally new in that it extends a Markov chain Monte Carlo (MCMC) method to directly incorporate probability constraints into the inference process, which can be significantly more efficient. We present our methods based on the widely used framework of Markov logic networks (MLNs), a highly general yet simple representation formalism for statistical knowledge [2] (Section 3). We thoroughly evaluate the performance of our newly proposed MCMC method in a series of experiments based on both synthetic and real-world applications (Section 5).

2 On the Semantics of Uncertain Evidence

Whenever evidence on some set of random variables X_E is uncertain, we associate with each assignment $X_E = x_E$ a degree of belief $p_{x_E} \in [0, 1]$. For every assignment $X_E = x_E$, there are (at least) two ways in which we might interpret the evidence and the corresponding degree of belief p_{x_E}:

(1) We have made an observation indicating $X_E = x_E$ and the degree to which this observation is reliable is p_{x_E}.
(2) The degree to which $X_E = x_E$ should be believed is p_{x_E}.

In either case, p_{x_E} is a well-defined degree of belief that should appropriately alter our posterior beliefs on unobserved variables, yet the semantics are fundamentally different. In the former case, we speak of *virtual* (or *intangible*) *evidence* (as the evidence impinges not immediately upon random variables in X but virtual observations thereof), while the latter is referred to as *soft evidence*. In the following, we will explain the differences between these two semantics in more detail.

We consider a full joint distribution over a set of random variables $X = \{X_1, \ldots, X_{|X|}\}$, which is partitioned into observed variables X_E (evidence variables) and unobserved variables X_U, i.e. $X = X_E \uplus X_U$. We denote by $\mathcal{X}$ the set of possible worlds, i.e. the set of possible assignments of values to each of the variables in X. For $x \in \mathcal{X}$, we simply write x as shorthand for $X = x$. Without loss of generality, we assume that all random variables under consideration are Boolean.

2.1 Virtual Evidence

Virtual evidence can be thought of as evidence concerning (undisclosed) observations that are outside of the set of variables X that we consider in our joint probability distribution but that have bearing on a subset X_E of these variables [3]. Technically, virtual evidence on X_E is realized by adding to the model auxiliary evidence variables V_E (corresponding to the observations that were made) and adding the likelihood of an assignment $X_E = x_E$ under these observations as an additional potential, i.e. we add $L(x_E) := P(V_E = v_E \mid X_E = x_E)$ for all $x_E \in \mathcal{X}_E$, where v_E is an arbitrarily chosen value assigned to the auxiliary evidence variables.

Adding virtual evidence on a variable X_i with beliefs $\langle p, q\rangle$ for *true* and *false* respectively can be viewed as adding an observation with a corresponding observation model in which the probability with which we observe $X_i =$ *true* (*false*) if X_i is indeed *true* (*false*) is p (q).[1]

By applying the principle of virtual evidence, we essentially weight each possible world with the likelihood of the evidence variables X_E:

$$\begin{aligned} &P(X \mid v_E) = P(X_U, X_E \mid v_E) = P(X_U \mid X_E) \cdot P(X_E \mid v_E) \\ &\propto P(X_U \mid X_E) \cdot P(v_E \mid X_E) \cdot P(X_E) = P(X_U, X_E) \cdot P(v_E \mid X_E) \\ &= P(X) \cdot L(X_E) \end{aligned} \tag{1}$$

2.2 Soft Evidence

The concept of soft evidence is both simpler and more difficult. It is simpler, because its interpretation is straightforward: Any piece of soft evidence is but a constraint on the probability distribution over assignments to (a subset of) X_E. Soft evidence on a single variable X_i with distribution $\langle p_i, 1-p_i\rangle$ simply requires the probability of $X_i =$ *true* to be p_i. However, as we will see, soft evidence is also the more difficult concept, because its treatment cannot simply be reduced to the addition of auxiliary variables for which potentials can straightforwardly be derived.

Let s_E represent the observations that have led us to believe in the soft evidence that we are given. In its entirety, soft evidence on the set of variables X_E can then be understood as defining the distribution $P(X_E \mid s_E)$. *Jeffrey's rule* [3] provides a way of updating our posterior beliefs on the unobserved variables X_U in the light of our observations:

$$P(X_U \mid s_E) = \sum_{x_E} P(X_U \mid x_E) \cdot P(x_E \mid s_E) \tag{2}$$

We obtain the posterior distribution over $\mathcal{X}$ as

$$\begin{aligned} P(X \mid s_E) &= P(X_U, X_E \mid s_E) = P(X_U \mid X_E) \cdot P(X_E \mid s_E) \\ &= \frac{P(X_U, X_E)}{P(X_E)} \cdot P(X_E \mid s_E) = P(X) \cdot \frac{P(X_E \mid s_E)}{P(X_E)} \end{aligned} \tag{3}$$

2.3 Discussion

Both virtual and soft evidence can be seen as generalizations of hard evidence. In the special cases where values of 1.0 or 0.0 are specified, both equivalently reduce to hard evidence. For values in between, however, the semantics differ greatly.

From a computational perspective, virtual evidence does not pose any challenges, as it imposes only strictly local dependencies by introducing additional likelihoods, which

[1] Note that the values appearing in a virtual evidence vector (here p and q) need not sum to 1 (only the ratio between any two values matters), but we could equivalently represent them as a distribution (and Section 2.3 further on assumes that we do).

can be handled by adding appropriate potentials to a probabilistic model [3,4]. Its use is appropriate only if the uncertain information originates from a source that indeed computes mere likelihoods. The effect that a piece of virtual evidence has on the posterior of the respective variable(s) is highly dependent on the probability that is indicated by the model without the respective piece of evidence (and given any further evidence).[2] Soft evidence is concerned precisely with defining that effect, and its treatment therefore needs to take into consideration the fact that several pieces of uncertain evidence cannot usually be treated independently but influence each other.

3 Markov Logic Networks

Markov logic networks (MLNs) [2] combine first-order logic with the semantics of probabilistic graphical models. An MLN L is given by a set of pairs $\langle F_i, w_i \rangle$, where F_i is a formula in first-order logic and w_i is a real-valued weight. For each finite domain of discourse D (set of constants), an MLN L defines a *ground Markov random field* $M_{L,D} = \langle X, G \rangle$ as follows:

1. X is a set of boolean variables. For each possible grounding of each predicate appearing in L, we add to X a boolean variable (ground atom). We denote by $\mathcal{X} := \mathbb{B}^{|X|}$ the set of possible worlds, i.e. the set of possible assignments of truth values to the variables in X.
2. G is a set of weighted ground formulas, i.e. a set of pairs $\langle \widehat{F}_j, \widehat{w}_j \rangle$, where $\widehat{F}_j$ is a ground formula and $\widehat{w}_j$ a real-valued weight. For each possible grounding $\widehat{F}_j$ of each formula F_i in L, we add to G the pair $\langle \widehat{F}_j, \widehat{w}_j = w_i \rangle$. With each such pair, we associate a feature $\widehat{f}_j : \mathcal{X} \to \{0, 1\}$, whose value for $x \in \mathcal{X}$ is 1 if $\widehat{F}_j$ is satisfied in x and 0 otherwise, and whose weight is $\widehat{w}_j$.

$M_{L,D}$ specifies a probability distribution over $\mathcal{X}$ as follows,

$$P(X = x) = \tfrac{1}{Z} \exp\left(\textstyle\sum_j \widehat{w}_j \widehat{f}_j(x)\right) \tag{4}$$

where Z is a normalization constant. The probability of a possible world $x \in \mathcal{X}$ is thus proportional to the exponentiated sum of weights of formulas that are satisfied in x, i.e. $P(x) \propto \exp(\sum_j \widehat{w}_j \widehat{f}_j(x)) =: \omega(x)$. With $s(F) := \sum_{x \in \mathcal{X},\, x \models F} \omega(x)$, we can calculate the probability of any ground formula F_1 given any other formula F_2 as

$$P(F_1 \mid F_2) = \frac{P(F_1, F_2)}{P(F_2)} = \frac{s(F_1 \wedge F_2)}{s(F_2)} \tag{5}$$

4 Soft Evidential Update

In the following, we address the question of how beliefs can efficiently be updated in the light of soft evidence. We will present two methods for the problem of soft evidential update that are considerably more practical than the naive approach involving $2^{|X_E|}$ inference runs that Equation 2 gives rise to, discussing related work along the way.

[2] For example, if the probability of $X_i = \mathit{true}$ given the other evidence is 0.1, adding virtual evidence on X_i with values $\langle 0.8, 0.2 \rangle$ will cause the probability of $X_i = \mathit{true}$ to become $0.1 \cdot 0.8/(0.1 \cdot 0.8 + 0.9 \cdot 0.2) \approx 0.31$, not 0.8.

4.1 Probability Constraints and Iterative Fitting

The problem of soft evidential update is closely related to the concept of imposing probability constraints on a distribution. To make this explicit, we briefly review the *iterative proportional fitting procedure* (IPFP) (cf. [5]). This procedure allows to modify a joint distribution such that it satisfies a set $\mathcal{R} = \{R_k\}$ of probability constraints, a *probability constraint* on a distribution $P(X)$ being a probability distribution $R_k(Y_k)$ over $Y_k \subseteq X$. IPFP guarantees that the distribution $P(X)$ is modified in such a way that the resulting distribution satisfies all of the given probability constraints and is maximally similar to $P(X)$ (with respect to Kullback-Leibler divergence). It achieves this by iteratively adjusting the distribution to fit one of the probability constraints in a round robin fashion. The initial distribution is $Q_0(X) = P(X)$. Then, in the i-th step, IPFP adjusts the (support of the) distribution to satisfy the k-th constraint $R_k(Y_k)$ (where $k = ((i-1) \bmod m) + 1$ if m is the number of probability constraints) using

$$Q_i(X) = Q_{i-1}(X) \cdot \frac{R_k(Y_k)}{Q_{i-1}(Y_k)} \tag{6}$$

Iterative fitting continues until convergence, which can always be achieved provided that the probability constraints are consistent.

Notice that the update rule above is analogous to Equation 3. Hence soft evidence can be handled by applying IPFP with a single constraint $R_1(X_E) := P(X_E \mid s_E)$.

In practice, however, we will rarely be given a joint distribution $P(X_E \mid s_E)$ as evidence, for such distributions are exponential in the size of X_E and therefore impractical to obtain and represent explicitly. Rather, from now on, consider the (perhaps most relevant) case where we are given a number of soft evidences, each concerning a single variable. Let $\{i_1, \dots, i_{|X_E|}\}$ be the set of indices of variables in X_E. The set of probability constraints $\mathcal{R}$ then contains, for each variable $X_{i_k} \in X_E$, a distribution $R_k(X_{i_k}) = \langle p_k, 1-p_k \rangle$. Note that this means that the distribution $P(X_E \mid s_E)$ is no longer given – and Jeffrey's rule is inapplicable unless we make additional assumptions. We should, of course, not simply assume independence of the individual pieces of soft evidence, since we do have information about dependencies in the form of the prior $P(X)$. An application of IPFP with the set of constraints $\mathcal{R} = \{R_1, \dots, R_{|X_E|}\}$ will ensure that this information is appropriately considered.

Unfortunately, a direct application of IPFP is generally infeasible, because it requires us to represent the full-joint distribution over $\mathcal{X}$ explicitly. Therefore, it seems more appropriate to apply iterative fitting not to the full-joint distribution itself but rather to a model that is capable of representing it compactly. Consider a ground Markov random field $M_{L,D} = \langle X, G \rangle$ and the distribution $P(X)$ it represents. For each constraint $R_k(X_{i_k})$, we add to G the ground formula $\widehat{F}_{R_k} := X_{i_k}$ with weight $\widehat{w}_{R_k}$; Initially, we set $\widehat{w}_{R_k} \leftarrow 0$ for all $R_k \in \mathcal{R}$ and thus obtain as the initial distribution $Q_0(X) = P(X)$. We will now show that we can achieve the transition described in Equation 6 by modifying only the respective weight $\widehat{w}_{R_k}$. The probability of X_{i_k} indicated by the current model is $q := s(X_{i_k})/(s(X_{i_k}) + s(\neg X_{i_k}))$ (i), yet R_k requires this probability to become p_k. By adding $\log(\lambda)$ to $\widehat{w}_{R_k}$, we scale the sum of exponentiated sums of weights of possible worlds satisfying X_{i_k}, i.e. $s(X_{i_k})$, by a factor of λ. Therefore, we need to find λ such that $p_k = s(X_{i_k})\lambda/(s(X_{i_k})\lambda + s(\neg X_{i_k}))$ (ii). Combining (i) and (ii) yields $\lambda = p_k \cdot (1-q)/((1-p_k) \cdot q)$ (see also [6]). We can thus achieve the transition from Q_{i-1} to Q_i by applying the update

$$\widehat{w}_{R_k} \leftarrow \widehat{w}_{R_k} + \log\left(\frac{p_k \cdot (1-q)}{(1-p_k) \cdot q}\right) \tag{7}$$

and perform iterative fitting at the level of model parameters. In each step, we need to run an inference algorithm in order to compute q, the current probability of the random variable (ground atom) whose probability we want to fit. Once the procedure has converged, we have the desired model parameters and can update our beliefs based on inference over the model. Henceforth, we refer to the algorithm described above as *IPFP-M* (IPFP at the model level).

Pan et al. have previously proposed a similar method for Bayesian networks, which extends the model with virtual evidence nodes whose conditional distributions they iteratively adjust to fit the probability constraints (Alg. 1 in [7]). Indeed, virtually all related work on the subject of soft evidential update has been centred around variations of IPFP [1,5,7,8]. It has even been suggested to directly apply IPFP but reduce the set of variables for which the joint distribution needs to be explicitly updated from X to X_E (Alg. 2 in [7]) – by first precomputing the prior $P(X_E)$, applying IPFP to $P(X_E)$ and then running an inference algorithm to update the beliefs on unobserved variables using a model extended with the results of IPFP. Thus, the complexity of every IPFP step is exponential in the size of X_E, which renders the algorithm inapplicable as X_E grows larger (it is, however, advantageous if X_E is very small). The same is true for the Big Clique algorithm [1], which takes a similar approach, albeit within a more confined setting (it is formulated strictly as an extension to the junction tree algorithm). IPFP-M, however, scales to larger sets of evidence variables; its performance is largely dependent on the efficiency of the underlying inference method. Most importantly, we can use methods that do not require an explicit representation of a cumbersome full-joint distribution and, in particular, approximate inference methods.

4.2 A Markov Chain Monte Carlo Method

We now introduce a Markov chain Monte Carlo (MCMC) method, which allows to directly integrate soft evidence into the inference procedure – without generating substantial overhead. Specifically, we present an extension of the algorithm MC-SAT [9], which, unlike other MCMC inference algorithms, can soundly handle deterministic and near-deterministic dependencies.

Consider a ground Markov random field $M_{L,D} = \langle X, G\rangle$ in which all formula weights are non-negative. Please note that this is not a restriction, as any pair $\langle \widehat{F}_j, \widehat{w}_j\rangle \in G$ can be equivalently replaced by $\langle \neg\widehat{F}_j, -\widehat{w}_j\rangle$. MC-SAT is a slice sampler that uses one auxiliary variable U_j per ground formula $\langle \widehat{F}_j, \widehat{w}_j\rangle \in G$. The joint distribution over the random variables X and the auxiliary variables U is given as $P(X = x, U = u) = (1/Z)\prod_j I_{[0,\exp(\widehat{f}_j(x)\widehat{w}_j)]}(u_j)$, where I is an indicator function which returns 1 if the argument is within the given interval and 0 otherwise. For a given $x \in \mathcal{X}$, $P(U = u \mid X = x)$ is uniform in the assignments that are within the intervals imposed by x, and the product of the sizes of these intervals is $\omega(x)$ and thus proportional to $P(x)$. $P(X = x \mid U = u)$ is uniform in the slice (subset) of $\mathcal{X}$ where $u_j < \exp(\widehat{f}_j(x)\widehat{w}_j)$ for all j, which implies that if $u_j > 1$, then $\widehat{f}_j(x) = 1$ and $\widehat{F}_j$ must be satisfied.

The Markov chain is formed by alternatingly sampling the auxiliary variables and the actual state as follows. The initial state $x^{(0)} \in \mathcal{X}$ must be one that satisfies all hard constraints in G. Then, in step i, the auxiliary variables are first resampled given the previous state $x^{(i-1)}$. If $\widehat{F}_j$ was satisfied in $x^{(i-1)}$, then with probability $(\exp(\widehat{w}_j) - 1)/\exp(\widehat{w}_j)$, we sample $u_j > 1$ and therefore $\widehat{F}_j$ must also be satisfied in the current step. We thus obtain a set $M = \{\widehat{F}_j \mid u_j > 1\}$ of formulas that must be satisfied and use a SAT sampler to uniformly sample a state $x^{(i)} \in \mathcal{X}$ that satisfies the formulas in M. For this purpose, MC-SAT uses the SampleSAT algorithm [10], which extends the WalkSAT algorithm by injecting randomness via simulated annealing-type moves, allowing near-uniform samples to be obtained. Because SampleSAT requires formulas to be in conjunctive normal form, we apply a conversion beforehand.

We now extend this algorithm to explicitly consider the probability constraints $\mathcal{R}$ corresponding to a set of soft evidences. Intuitively, we would like to achieve the effect of applying MC-SAT to the model we would have obtained by applying IPFP-M to $P(X)$ and $\mathcal{R}$, yet without having to actually compute the weights w_{R_k} that IPFP-M indicates upon convergence. Consider a constraint $R_k(X_{i_k}) = \langle p_k, 1-p_k \rangle$ and the pair $\langle X_{i_k}, w_{R_k} \rangle$ that IPFP-M would have added to G. Observe that without $\langle X_{i_k}, w_{R_k} \rangle$, there is some relative frequency with which states in which X_{i_k} is *true* would have appeared in MC-SAT's Markov chain. The weight w_{R_k} computed by IPFP-M would achieve that this relative frequency becomes approximately p_k (given that the number of samples is large enough) – simply by specifying the probability with which the Markov chain should remain within the subspace of the support of $P(X)$ where X_{i_k} (or, if w_{R_k} is negative, $\neg X_{i_k}$) is satisfied once it reaches a state where X_{i_k} ($\neg X_{i_k}$) is satisfied. Technically, if X_{i_k} ($\neg X_{i_k}$) was satisfied in the previous step, it is *reselected* for addition to the set M of formulas to satisfy in the current step with probability $1 - e^{-w_{R_k}}$ (respectively $1 - e^{w_{R_k}}$). We can essentially simulate this (without knowing the actual probability with which reselection should occur) by keeping track of the relative frequency with which $X_{i_k} =$ *true* indeed appeared in the Markov chain and, if X_{i_k} was satisfied in the previous step, adding X_{i_k} to M whenever the relative frequency is, thus far, lower than p_k. Inversely, if X_{i_k} was not satisfied in the previous step, we add $\neg X_{i_k}$ to M if the relative frequency of X_{i_k} is greater than p_k. We therefore simulate the effect of w_{R_k} by splitting its impact across the formulas X_{i_k} and $\neg X_{i_k}$.[3] Because the resulting algorithm effectively integrates posterior probability constraints into MC-SAT, we refer to it as *MC-SAT-PC* (see Algorithm 1). We invoke the algorithm with G as the first argument and $\{\langle X_{i_k}, p_k \rangle \mid R_k \in \mathcal{R}\}$ as the second. Please note that the algorithm is not restricted to constraints on atomic formulas; arbitrary factorizations of the soft evidence could be considered.

5 Evaluation

In the following, we evaluate the performance of MC-SAT-PC by comparing it to IPFP-M, noting that IPFP-M belongs to a class of methods that could presently be considered as the methods of choice whenever the set X_E is not extremely small.

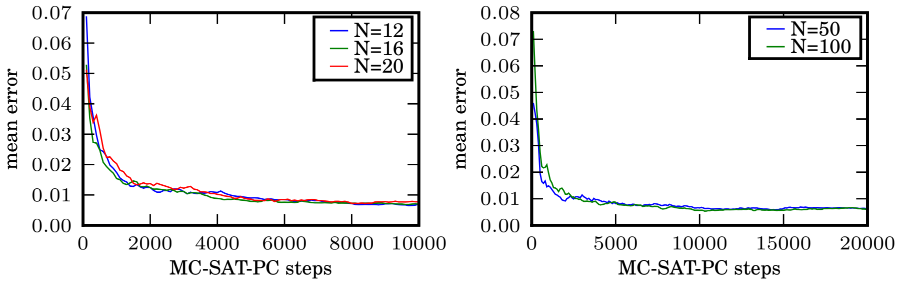
[3] In general, F with weight $\log(p/(1-p))$ is equivalent to the combination of F with weight $\log(p)$ and $\neg F$ with weight $\log(1-p)$.

Algorithm 1 MC-SAT-PC(*groundFormulas, probConstraints, numSamples*)

```
$n_k \leftarrow 0$ for all $\langle F_k, p_k \rangle$ in probConstraints
for $i \leftarrow 0$ to numSamples $- 1$ do
    if $i = 0$ then
        $M \leftarrow \{F_k \mid \langle F_k, w_k \rangle \in$ groundFormulas $\wedge\, w_k$ is hard$\}$
    else
        $M \leftarrow \emptyset$
        for all $\langle F_k, w_k \rangle \in$ groundFormulas do
            if $x^{(i-1)} \models F_k$ then with probability $1 - e^{-w_k}$ add $F_k$ to $M$
        for all $\langle F_k, p_k \rangle \in$ probConstraints do
            if $x^{(i-1)} \models F_k$ then
                if $n_k/i < p_k$ then add $F_k$ to $M$
            else
                if $1 - n_k/i < 1 - p_k$ then add $\neg F_k$ to $M$
    sample $x^{(i)} \sim$ Uniform$(\mathrm{SAT}(M))$
    for all $\langle F_k, p_k \rangle \in$ probConstraints do
        if $x^{(i)} \models F_k$ then $n_k \leftarrow n_k + 1$
```

In our first experiment, we compare the results of IPFP-M with exact inference as the underlying inference algorithm (IPFP-M[exact]) to MC-SAT-PC. We randomly generated Markov random fields (MRFs) of three different sizes ($N \in \{12, 16, 20\}$), ten of each size. For an MRF with N boolean variables, we created N random features by generating clauses with a random number of literals (uniformly sampled from $\{1, \ldots, 10\}$), drawing weights from $\mathcal{N}(0, (\log(50)/2)^2)$. For each MRF, we randomly chose $N/2$ variables as the set X_E of (soft) evidence variables, sampling beliefs uniformly from $[0, 1]$. We ran IPFP-M[exact] on the inference problems thus obtained, terminating the procedure as soon as the maximum absolute error across all soft evidence variables dropped below $\varepsilon = 0.001$. Typically, IPFP-M[exact] terminated after at most 5 rounds over all constraints (i.e. $5N/2$ iterations and therefore $5N/2$ inference runs). In each iteration, we inferred not only the probability of the constraint to be fitted but the probabilities of all constraints in order to determine convergence, as well as the probabilities of all unobserved variables, $P(X_i \mid s_E)$ for all $X_i \in X_U$, as queries. We ran MC-SAT-PC with the same queries, comparing the results to the ones obtained by IPFP-M[exact] every 100 steps. As shown in Figure 1a, MC-SAT-PC quickly produces results that are, on average, within 0.01 of the true results. After 10000 steps, the maximum error in any query variable across all 30 experiments was 0.035. (The error margins are equivalent to what one typically obtains with MC-SAT.) These results indicate that, MC-SAT-PC is, in general, a suitable method for the approximation of posterior beliefs in the presence of soft evidence.

Exact inference is infeasible in all but the smallest domains (IPFP-M[exact] took over 80 minutes to converge for $N = 20$), and many approximate inference algorithms fail in the presence of deterministic constraints; MC-SAT is one of very few choices available. In our second experiment, we therefore compare MC-SAT-PC to IPFP-M with MC-SAT as the underlying inference algorithm (IPFP-M[MC-SAT]) in order to evaluate potential time savings. We generated random MRFs of sizes $N = 100$ and $N = 50$. For each experiment, we randomly selected 20 soft evidence variables, and, as before, used the remaining variables as query variables. Owing to the stochasticity



(a) Experiment 1

(b) Experiment 2

Fig. 1: Mean error in posterior marginals of variables in X_U computed by MC-SAT-PC with respect to results computed by (a) IPFP-M[exact], averaged across all 10 experiments, and (b) IPFP-M[MC-SAT]. N is the number of random variables considered in the respective (series of) experiments.

of MC-SAT, we had to relax IPFP-M's termination criterion: We stopped the procedure as soon as the mean absolute error across all soft evidence variables dropped below $\varepsilon_1 = 0.01$ and the the maximum error dropped below $\varepsilon_2 = 0.05$. In each iteration of IPFP-M[MC-SAT], we ran MC-SAT for 10000 steps. Figure 1b summarizes the main result: MC-SAT-PC produces roughly the same results; yet it does so in a fraction of the time. While IPFP-M[MC-SAT] ran for 9.5 and 76.6 minutes (3 and 7 rounds) for $N = 50$ and $N = 100$ respectively, MC-SAT-PC drew 20000 samples in less than 35 seconds in each case. The maximum deviation from the results computed by IPFP-M was 0.024 and therefore within the expected range.

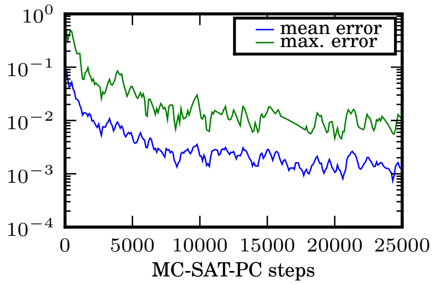


Fig. 2: Experiment 3: Mean and maximum errors across all soft evidence variables for an instance of an MLN for object identity resolution (logarithmic scale); final errors are 0.0014 and 0.0103 respectively.

We also tested MC-SAT-PC on a complex real-world problem (which in fact provided the initial motivation for the design of MC-SAT-PC): temporal object identity resolution, where the goal is to manage associations between observations and (not necessarily uniquely identifiable) objects over time – based on the similarity in appearance between observed entities and models of the objects and based on information on previous observations. We created a Markov logic network for this problem, which is applied iteratively as new observations come in. Previously computed beliefs are provided as soft evidence for a subsequent application of the model (as are beliefs indicating degrees of similarity). An exemplary ground instance of this problem contained 279 random variables and 2688 features; 42 variables were given as soft evidence, 36 as hard evidence. An application of IPFP-M[MC-SAT] to a problem of this size is essentially infeasible, as far too many iterations are required, each of which involves only insignificantly less effort than a full application of MC-SAT-PC. Therefore, because we do not have any results on the unobserved variables to compare against, we plot in Figure 2 the number of steps taken by MC-SAT-PC against the mean and maximum errors we obtain for the soft evidence variables X_E, which (if $|X_E|$ is large enough) are fairly good indicators of convergence. The errors drop fairly quickly; and the results we obtained for X_U were consistent with our common sense expectations.

As with any MCMC method, there can be cases where MC-SAT-PC may take a long time to converge. If, for example, the probability indicated by the model for some X_{i_k} is very low (given the remaining evidence), e.g. 10^{-4}, but R_k requires it to be substantially higher, then a large number of steps may be required for the Markov chain to enter and appropriately explore the subspace where $X_{i_k} = \mathit{true}$. Of course, in such cases, IPFP-M will also have to use a rather large number of steps in the underlying inference algorithm, as we are required to compute an approximately correct non-zero probability for X_{i_k} if Equation 6 is to be applicable and have the desired effect.

6 Conclusion

In this work, we presented MC-SAT-PC, an MCMC algorithm for soft evidential update, which effectively integrates (posterior) probability constraints into the inference procedure.[4] We described its relation to IPFP-based methods and demonstrated that, particularly for hard problems involving a large number of evidence variables, it can improve performance by orders of magnitude without sacrificing accuracy. We presented our methods within the framework of Markov logic networks, yet our results carry over to other formalisms that are subsumed by MLNs (including Bayesian networks and statistical relational models based upon them).

Acknowledgements. This work was partly supported within the DFG cluster of excellence CoTeSys (Cognition for Technical Systems).

References

1. Kim, Y.G., Valtorta, M.: Soft Evidential Update for Communication in Multiagent Systems and the Big Clique Algorithm (2000)
2. Richardson, M., Domingos, P.: Markov Logic Networks. Mach. Learn. **62** (2006) 107–136
3. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
4. Li, X.: On the Use of Virtual Evidence in Conditional Random Fields. In: EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Morristown, NJ, USA, Association for Computational Linguistics (2009) 1289–1297
5. Peng, Y., Ding, Z.: Modifying Bayesian Networks by Probability Constraints. In: Proceedings of the 24th Conference on Uncertainty in AI (UAI). (2005) 26–29
6. Jain, D., Kirchlechner, B., Beetz, M.: Extending Markov Logic to Model Probability Distributions in Relational Domains. In: Proceedings of the 30th German Conference on Artificial Intelligence (KI-2007). (2007) 129–143
7. Pan, R., Peng, Y., Ding, Z.: Belief Update in Bayesian Networks Using Uncertain Evidence. In: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2006). (2006) 441–444
8. Langevin, S., Valtorta, M.: Performance Evaluation of Algorithms for Soft Evidential Update in Bayesian Networks: First Results. In Greco, S., Lukasiewicz, T., eds.: SUM. Volume 5291 of Lecture Notes in Computer Science., Springer (2008) 284–297
9. Poon, H., Domingos, P.: Sound and Efficient Inference with Probabilistic and Deterministic Dependencies. In: AAAI, AAAI Press (2006)
10. Wei, W., Erenrich, J., Selman, B.: Towards Efficient Sampling: Exploiting Random Walk Strategies. In: AAAI. (2004) 670–676

[4] Our implementations of both MC-SAT-PC and IPFP-M are freely available as part of the ProbCog toolbox: http://ias.cs.tum.edu/research/probcog