Relaxed Survey Propagation: A Sum-Product Algorithm for Max-SAT

Hai Leong Chieu
Singapore MIT Alliance
National University of Singapore
haileong@nus.edu.sg

Wee Sun Lee
Department of Computer Science
National University of Singapore
leews@comp.nus.edu.sg

Abstract

The survey propagation (SP) algorithm has been shown to work well on large instances of the random 3-SAT problem near its phase transition. It was shown that SP estimates marginals over *covers*, using *joker states* to represent clusters of configurations. The SP-y algorithm generalizes SP to work on the Max-SAT problem, but the *cover* interpretation of SP does not generalize to SP-y. Recently, a relaxed survey propagation (RSP) algorithm has been proposed for inference in Markov random fields (MRF). RSP for MRFs assigns zero probability to *joker states*, and hence the *cover* interpretation is also inapplicable. We adapt RSP to solve Max-SAT problems, and show that it has an interpretation of estimating marginals over $covers$ violating a minimum number of clauses. This naturally generalizes the *cover* interpretation of SP. Empirically, we show that RSP outperforms SP-y and other state-of-the-art solvers on random as well as benchmark instances of Max-SAT.

Introduction

The 3-SAT problem is the archetypical NP-complete problem, and the difficulty of solving random 3-SAT instances has been shown to be related to the clause to variable ratio, $\alpha = M/N$, where M is the number of clauses and N the number of variables. There exists a critical value $\alpha_c \approx 4.267$: random 3-SAT instances with $\alpha < \alpha_c$ are generally satisfiable, while instances with $\alpha > \alpha_c$ are not. In statistical mechanics language, one says that there exists a phase transition for the 3-SAT problem at α_c. Instances close to the phase transition are generally hard to solve using local search algorithms (Braunstein, Mezard, and Zecchina 2005).

The survey propagation (SP) algorithm (Braunstein, Mezard, and Zecchina 2005) was derived in statistical mechanics, and has been shown to work well on random 3-SAT problems near its phase transition. In SP, variables can take on joker states, thus allowing SP to reason over clusters of configurations. Maneva et al. (2004) showed that SP can be viewed as a sum-product belief propagation algorithm, and it estimates marginals of *covers* over satisfiable configurations. Kroc et al. (2007) recently showed the existence of covers in large satisfiable 3-SAT instances near its phase transition.

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In statistical mechanics, SP can be derived as a special case of the SP-y algorithm, with y taken to ∞. For finite values of y, the SP-y algorithm deals with the Max-SAT problem of minimizing the number of violated constraints (Battaglia, Kolar, and Zecchina 2004). However, the sum-product view of SP does not generalize to SP-y.

Recently, the relaxed survey propagation (RSP) (Chieu, Lee, and Teh 2007) algorithm generalizes the application of SP to general Markov random fields, and RSP was shown to work well on binary networks with mixed couplings. The RSP algorithm involves a transformation of the original MRF into a *relaxed* MRF, via a weighted Max-SAT conversion. In the transformation, a parameter vector $\mathbf{y}$ was introduced. It was shown that RSP estimates the marginals on the original MRF, regardless (to a large extent) of $\mathbf{y}$, allowing $\mathbf{y}$ to be optimized for better convergence of the RSP algorithm. This flexibility enables RSP to outperform other algorithms for energy minimization over binary networks with mixed couplings. However, the joker states of SP have zero probability in RSP (Lemma 2 in (Chieu, Lee, and Teh 2007)), and the cover interpretation of SP was lost on RSP for MRFs.

Although RSP was obtained via a weighted Max-SAT conversion of an MRF, the weighted Max-SAT objective is modified if $\mathbf{y} \neq \mathbf{0}$, and RSP as it was formulated in (Chieu, Lee, and Teh 2007) was not directly applicable to weighted Max-SAT problems. In this paper, we show that by taking $\mathbf{y} = y\mathbf{1}$, (y is a scalar), the RSP algorithm can be used to solve Max-SAT problems. With this formulation, joker states are present, and RSP and SP-y are similar in many ways. However, while the notion of covers for SP does not generalize to SP-y, we show in Theorem 2 that RSP returns marginals over covers of configurations that violate a minimum number of constraints. Empirically, we show that RSP outperforms other state-of-the-art solvers on random Max-3-SAT, as well as on benchmark Max-SAT instances.

Survey propagation

In this section, we give a brief review of the SP-y algorithm and its derivation from the cavity method, using factor graph notation (Kschischang, Frey, and Loeliger 2001). Our aim in developing SP-y in this section is to show the similarities between SP-y and RSP.

Preliminaries

The SAT problem consists of a set of boolean variables constrained by a boolean function in conjunctive normal form, which can be treated as a set of clauses. Each clause is a set of literals (a variable or its negation), and is satisfied if one of them evaluates to 1. It is represented by (V, C), where V is the set of boolean variables, and C the set of clauses. For a clause $a \in C$, we denote $C(a)$ as the set of variables in the clause a, and for a variable $i \in V$, we denote $C(i)$ as the set of clauses that contain i. The SAT problem consists of finding a configuration that satisfies all the clauses.

The Max-SAT problem consists of finding a configuration that minimizes the number of unsatisfied clauses. It can be represented as an energy minimization problem. Defining the Max-SAT energy as $E = \sum_a \prod_{j \in C(a)} \frac{1}{2}(1 + J_{a,j} s_j)$, a boolean variable $x_i \in \{0, 1\}$, is related to the spin $s_i \in \{-1, +1\}$ by $s_i = (-1)^{x_i}$. The coupling $J_{a,i}$ equals $+1$ (resp. -1) if the variable x_i appears non negated (resp. negated) in the clause a. In this form, the energy of a configuration equals the number of clauses violated by the configuration. Minimum energy configurations are equivalent to maximum-a-posteriori configurations in the Boltzmann-Gibbs distribution $P(\mathbf{x}) = \frac{1}{Z} \exp(-E(\mathbf{s}))$. In the factor graph representation (Kschischang, Frey, and Loeliger 2001), each factor is a clause in the Max-SAT problem.

The cavity method

The cavity method (at zero temperature) (Mezard and Parisi 2003) derives update equations by considering the consistencies between a system with N spins and a system with $N+1$ spins when the $(N+1)^{th}$ spin, s_0, is added. The main assumption behind the cavity method is that in the large N limit, the neighboring spins of s_0 are uncorrelated (locally "tree-like"). In the cavity approach, warning messages are sent from each factor (or clause) a to a neighboring variable i, denoted as $u_{a\to i}$, and from each variable i to a neighboring factor a, denoted as $h_{i\to a}$ (see Figure 1). The warning messages $u_{a\to i}$ takes values in $\{-1, 0, +1\}$. The case $u_{a\to i} = +1$ (resp. -1) correspond to a warning from a clause a to a variable s_i that the variable s_i should take the value -1 (resp. +1). $u_{a\to i} = 0$ means s_i is free to take +1 or -1. Under this perspective, the messages $h_{i\to a}$ simply tabulates incoming warnings from neighboring factors of s_i:

$$h_{i\to a} = \sum_{b \in C(i)-a} u_{b\to i}. \quad (1)$$

The message $u_{a\to i}$ takes the value $J_{a,i}$ only if all other neighboring variables of a are going to violate a. Otherwise, it takes the value 0:

$$u_{a\to i}(\{h_{j\to a}\}) = J_{a,i} \prod_{j \in C(a)\setminus i} \theta(J_{a,j} h_{j\to a}), \quad (2)$$

where $\theta(x) = 0$ if $x < 0$, and $\theta(x) = 1$ if $x \geq 0$.

In statistical mechanics, a *ground state* is a cluster of configurations of equal energy, related to each other by single spin flip moves, which are locally stable, in the sense that the energy cannot be decreased by any flip of a finite number of spins. (In statistical mechanics, physicists are interested in the case where $N \to \infty$). SP-y is derived under a framework of assumptions called "one-step replica symmetry breaking", where we assume that there are many local ground states (LGS). Under this framework, two important assumptions are made (Mezard and Parisi 2003)

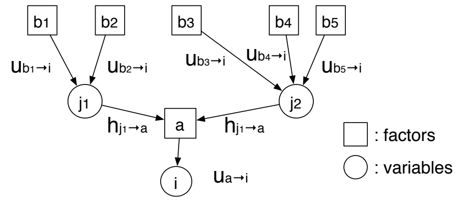

Figure 1: Warning messages between factor/variable nodes.

Assumption 1. *For a problem with N spins, denote $\epsilon = E/N$ as the energy density, and $\eta(\epsilon)$ the number of LGS at the energy density ϵ. Assume that these LGS are distributed according to an exponential distribution:*

$$\eta(\epsilon) \approx \exp(N\Sigma(\epsilon)) \quad (3)$$

where $\Sigma(\epsilon)$ is called the complexity.

Defining $g(y) = y\Phi(y)$, the Legendre transform for $\Sigma(\epsilon)$,

$$\Sigma(\epsilon) = \min_y \epsilon y - y\Phi(y), \quad (4)$$

$$y\Phi(y) = \min_\epsilon \epsilon y - \Sigma(y), \quad (5)$$

and the minimization condition yields

$$\epsilon = \Phi(y) + y\frac{d\Phi(y)}{dy}, \quad (6)$$

$$y = \frac{d\Sigma(\epsilon)}{d\epsilon}. \quad (7)$$

Assumption 2. *(section 4.4 in (Mezard and Parisi 2003)) There is a unique global ground state (GGS), U, where $\Sigma(\epsilon = U/N) = 0$, and*

$$\epsilon = \Phi(y) \quad (8)$$

$$\frac{d\Phi(y)}{dy} = 0. \quad (9)$$

Equations 8 and 9 result from 4 and 6 respectively. When using the SP-y algorithm, y should be selected to satisfy Equation 9. It has been established that in the satisfiable region, we should select $y \to \infty$. In the unsatisfiable region, y takes on a finite value (Battaglia, Kolar, and Zecchina 2004).

The update equations for SP-y are

$$P_{j\to a}(h) = \sum_{\{u_{b\to i}\}\to h} \prod_{b \in C(j)\setminus a} Q_{b\to j}(u_{b\to j}) \exp(-y\delta E) \quad (10)$$

$$Q_{a\to i}(u) = \sum_{\{h_{j\to a}\}\to u} \prod_{j \in C(a)\setminus i} P_{j\to a}(h_{j\to a}), \quad (11)$$

where the notation $\{u_{b\to i}\} \to h$ denotes the set of $\{u_{b\to i}\}$ that gives rise to the message h by Equation 1. Similarly for the notation $\{h_{j\to a}\} \to u$. The term δE in Equation 10 measures the change in energy when a new spin is added. In the Max-SAT problem, δE corresponds to the number of violated constraints. Hence, in the notations of (Battaglia, Kolar, and Zecchina 2004), a penalty term of $\exp(-2y)$ is multiplied into the distribution for each violated constraint. Battaglia et al. (2004) formulated an efficient algorithm for performing the SP-y updates for solving Max-SAT.

Relaxed Survey Propagation

In the following, given a Max-SAT problem, we set up an alternative MRF, on which we run the sum-product algorithm. In Theorem 1, we show that this formulation generalizes the sum-product interpretation of SP given in (Maneva, Mossel, and Wainwright 2004), and in the main theorem (Theorem 2), we show that running the sum-product belief propagation on this MRF estimates marginals over covers of configurations violating a minimum number of constraints, (i.e. *min-covers* defined in Definition 7).

In survey propagation, in addition to the values $\{0, 1\}$, variables can take a third value, * (joker state), signifying that the variable is free to take either 0 or 1, without violating any clause. This corresponds to a "no-warning" message (e.g. $u_{a\to i} = 0$) in notations of the previous section. In this section, we assume that variables x_i take values in $\{0, 1, *\}$.

Definition 1. *(Maneva, Mossel, and Wainwright 2004) A variable x_i is constrained by the clause $a \in C$ if it is the unique satisfying variable for a (all other variables violate a). Define $\mathrm{CON}_{i,a}(\mathbf{x}_{\mathbf{C(a)}}) = \delta(x_i \text{ is constrained by } a)$, where $\delta(P)$ equals 1 if P is true, and 0 otherwise.*

We introduce the scalar y in the following definition:

Definition 2. *An assignment $\mathbf{x}$ is invalid for clause a if and only if all variables are unsatisfying except for exactly one for which $x_i = *$. (In this case, x_i cannot take * as it is constrained). Define*

$$\mathrm{VAL}_a(\mathbf{x}_{\mathbf{C(a)}}) = \begin{cases} 1 & \text{if } \mathbf{x}_{\mathbf{C(a)}} \text{ satisfies } a \\ \exp(-y) & \text{if } \mathbf{x}_{\mathbf{C(a)}} \text{ violates } a \\ 0 & \text{if } \mathbf{x}_{\mathbf{C(a)}} \text{ is invalid} \end{cases} \quad (12)$$

The term $\exp(-y)$ is the penalty for violating a clause.

Definition 3. *(Maneva, Mossel, and Wainwright 2004) Define the parent set P_i of a variable x_i to be the set of clauses for which x_i is the unique satisfying variable, (i.e. the set of clauses constraining x_i).*

We now construct another MRF $G_s = (V_s, F_s)$ as follows: variables $\lambda_i \in V_s$ are of the form $\lambda_i = (x_i, P_i)$, where x_i are variables in the Max-SAT problem. We define variable and clause compatibilities as in (Maneva, Mossel, and Wainwright 2004). The single variable compatibilities (Ψ_i) are:

$$\Psi_i(\lambda_i = \{x_i, P_i\}) = \begin{cases} \omega_0 & \text{if } P_i = \emptyset, x_i \neq * \\ \omega_* & \text{if } P_i = \emptyset, x_i = * \\ 1 & \text{for any other valid } (x_i, P_i) \end{cases}, \quad (13)$$

where $\omega_0 + \omega_* = 1$. The clause compatibilities (Ψ_a) are:

$$\Psi_a(\lambda_{\mathbf{a}} = \{x_k, P_k\}_{k\in C(a)}) = \mathrm{VAL}_a(\mathbf{x}_{\mathbf{C(a)}}) \times \textstyle\prod_{k\in C(a)} \delta((a \in P_k) = \mathrm{CON}_{a,k}(\mathbf{x}_{\mathbf{C(a)}})), \quad (14)$$

where δ is defined in Definition 1. The single-variable compatibilities $\Psi_k(\lambda_k)$ are defined so that when x_k is unconstrained (i.e. $P_k = \emptyset$), $\Psi_k(\lambda_k)$ takes the values ω_* or ω_0 depending on whether x_k equals *. The clause compatibilities introduce the penalties into the joint distribution. Since the values of $\{x_k\}_k$ determines uniquely the values of $\{P_k\}_k$,

$$P(\mathbf{x}) = P(\{x_k, P_k\}_k) \propto \omega_0^{n_0}\omega_*^{n_*} \prod_{a\in \mathrm{UNSAT}(\mathbf{x})} \exp(-y), \quad (15)$$

where n_0 is the number of unconstrained variables in $\mathbf{x}$, and n_* the number of variables taking $*$.

Maneva et al. (2004) formulated the SP-ρ algorithm, which, for $\rho = 1$, is equivalent to the SP algorithm in (Braunstein, Mezard, and Zecchina 2005). Comparing RSP and SP-ρ, we have (Chieu, Lee, and Teh 2007):

Theorem 1. *By taking $y \to \infty$, RSP is equivalent to SP-ρ (Maneva, Mossel, and Wainwright 2004), with $\rho = \omega_*$.*

Proof. RSP and SP-ρ differ only in Definition 2, and with $y \to \infty$ in RSP, their definitions become identical. □

Taking y to infinity correspond to disallowing violated constraints, and SP-ρ was formulated for satisfiable SAT problems, where all clauses must be satisfied.

The update equations of RSP is given in Figure 2. In the worst case in a densely connected factor graph, each iteration of updates can be performed in $O(MN)$ time, where N is the number of variables, and M the number of clauses.

Covers in Max-SAT

In this section, we discuss the concept of covers. First, we define a partial order on the set of all valid assignments (defined in Definition 2) as follows (Maneva, Mossel, and Wainwright 2004)

Definition 4. *Let $\mathbf{x}$ and $\mathbf{y}$ be two valid assignments. We write $\mathbf{x} \leq \mathbf{y}$ if $\forall i$, (1) $x_i = y_i$ or (2) $x_i = *$ and $y_i \neq *$.*

This partial order defines a lattice, and Maneva et al. (2004) showed that SP is a "peeling" procedure that peels a satisfying assignment to its minimal element in the lattice. A cover is a minimal element in the lattice. In the SAT region, Kroc et al. (2007) defined covers as follows:

Definition 5. *A cover is an assignment $\mathbf{x} \in \{0, 1, *\}^N$ such that*

1. *every clause has at least one satisfying literal, or at least two literals with value * under $\mathbf{x}$, and*
2. *$\mathbf{x}$ has no unconstrained variables assigned 0 or 1.*

The SP algorithm was shown to return marginals over covers (Maneva, Mossel, and Wainwright 2004). In principle, there are two kinds of covers: *true* covers which correspond to satisfying configurations, and *false* covers which do not. Kroc et al. (2007) showed empirically that the number of false covers is negligible for SAT instances. For RSP, we introduce the notion of v-cover:

Definition 6. *A v-cover is an assignment $\mathbf{x} \in \{0, 1, *\}^N$ such that*

1. *there are exactly v clauses violated by the configuration,*
2. *violated clauses do not contain variables assigned *,*
3. *satisfied clauses have at least one satisfying literal, or at least two literals with value * under $\mathbf{x}$, and*
4. *$\mathbf{x}$ has no unconstrained variables assigned 0 or 1.*

Starting from any configuration violating v clauses, we can "peel" it down to its v-cover by considering all variables in violated clauses as "frozen", and peeling only variables that do not appear in violated clauses.

$$M^s_{a\to i} = \prod_{j\in C(a)\setminus\{i\}} R^u_{j\to a} \tag{16}$$

$$M^u_{a\to i} = \left[\prod_{j\in C(a)\setminus\{i\}} (R^u_{j\to a} + R^*_{j\to a}) + \sum_{k\in C(a)\setminus\{i\}} (R^s_{k\to a} - R^*_{k\to a}) \prod_{j\in C(a)\setminus\{i,k\}} R^u_{j\to a}\right] + (e^{-y} - 1) \prod_{j\in C(a)\setminus\{i\}} R^u_{j\to a} \tag{17}$$

$$M^*_{a\to i} = \prod_{j\in C(a)\setminus\{i\}} (R^u_{j\to a} + R^*_{j\to a}) - \prod_{j\in C(a)\setminus\{i\}} R^u_{j\to a} \tag{18}$$

$$R^s_{i\to a} = \prod_{\beta\in C^u_a(i)} M^u_{\beta\to i} \left[\prod_{\beta\in C^s_a(i)} (M^s_{\beta\to i} + M^*_{\beta\to i})\right] \tag{19}$$

$$R^u_{i\to a} = \prod_{\beta\in C^s_a(i)} M^u_{\beta\to i} \left[\prod_{\beta\in C^u_a(i)} (M^s_{\beta\to i} + M^*_{\beta\to i}) - (1-\omega_0) \prod_{\beta\in C^u_a(i)} M^*_{\beta\to i}\right] \tag{20}$$

$$R^*_{i\to a} = \prod_{\beta\in C^u_a(i)} M^u_{\beta\to i} \left[\prod_{\beta\in C^s_a(i)} (M^s_{\beta\to i} + M^*_{\beta\to i}) - (1-\omega_0) \prod_{\beta\in C^s_a(i)} M^*_{\beta\to i}\right] + \omega_* \prod_{\beta\in C^s_a(i)\cup C^u_a(i)} M^*_{\beta\to i} \tag{21}$$

$$B_i(0) \propto \prod_{\beta\in C^+(i)} M^u_{\beta\to i} \left[\prod_{\beta\in C^-(i)} (M^s_{\beta\to i} + M^*_{\beta\to i}) - \omega_* \prod_{\beta\in C^-(i)} M^*_{\beta\to i}\right] \tag{22}$$

$$B_i(1) \propto \prod_{\beta\in C^-(i)} M^u_{\beta\to i} \left[\prod_{\beta\in C^+(i)} (M^s_{\beta\to i} + M^*_{\beta\to i}) - \omega_* \prod_{\beta\in C^+(i)} M^*_{\beta\to i}\right] \tag{23}$$

$$B_i(*) \propto \prod_{\beta\in C(i)} M^*_{\beta\to i} \tag{24}$$

Figure 2: The update equations for RSP. These equations are sum-product belief propagation equations for the MRF defined in the text. The notations used here are identical to those in (Maneva, Mossel, and Wainwright 2004). The only difference is in Equation 17, where the penalty $\exp(-y)$ is introduced for violated clauses. Each iteration of the belief propagation updates can be done in $O(MN)$ time, where N is the number of variables, and M the number of clauses.

Definition 7. *We define a min-cover for a Max-SAT problem as the m-cover, where m is the minimum number of violated constraints for the problem.*

Theorem 2. *For $\omega_0 = 0$ and $\omega_* = 1$, and for sufficiently large y, RSP estimates marginals over min-covers.*

Proof. Having $\omega_0 = 0$ corresponds to forcing $n_0 = 0$, hence disallowing unconstrained variables. In this case, only covers have non-zero probabilities, and a v-cover has probability proportional to $\exp(-vy)$ in Equation 15. The ratio of the probability of a v-cover and that of a $(v+1)$-cover equals $\exp(y)$. For large y, the probability of min-covers dominates the distribution in Equation 15. Hence RSP, as the sum-product algorithm over the factor graph representing Equation 15, estimates marginals over min-covers. □

In Figure 3, we show the v-covers of a small example. In this example, there are four 1-covers, which are also the min-covers. Assumption 2 for SP-y states that $\Sigma(U) = 0$, where U is the global ground state energy (i.e. minimum number of violated constraints). This corresponds to the assumption that there is only one unique min-cover for a Max-SAT instance. While this is false for the example in Figure 3, it is assumed to be true for large random Max-3-SAT problems.

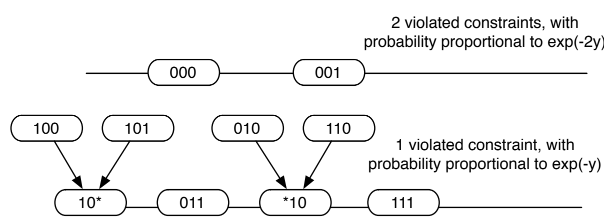

Figure 3: Energy landscape for the problem $(\bar{x_1} \vee x_2) \wedge (\bar{x_2} \vee x_3) \wedge (\bar{x_3} \vee x_1) \wedge (\bar{x_1} \vee \bar{x_2} \vee \bar{x_3}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2)$

In the case where $\omega_0 \neq 0$, the smoothing interpretation of SP-ρ also applies to RSP: the probability on a v-cover is spread over its lattice. See Theorem 6 in (Maneva, Mossel, and Wainwright 2004) for more details.

Experimental Results

We run experiments on random Max-3-SAT, as well as on a few instances used in the paper (Lardeux, Saubion, and Hao 2005). All experiments are run with $\omega_0 = 0$, and $\omega_* = 1$.

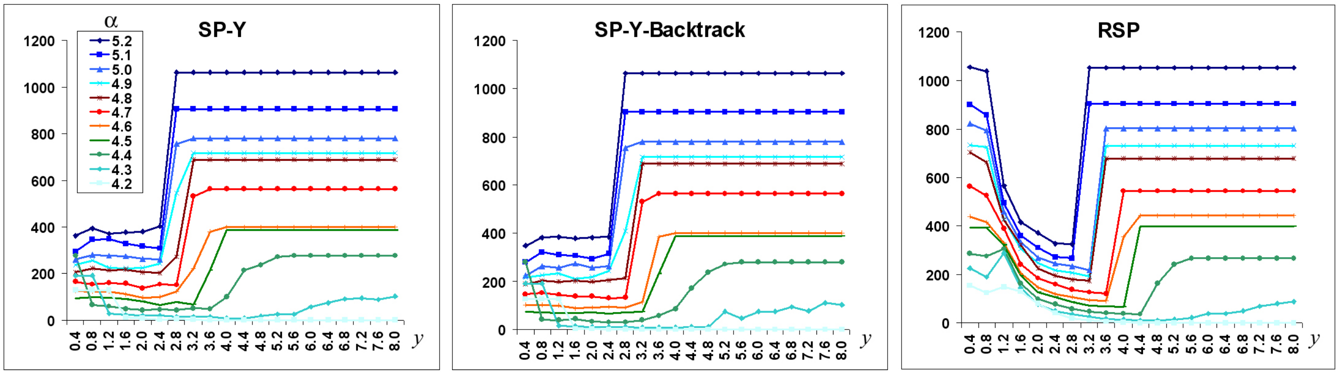

Figure 4: Performance of SP-y and RSP on Max-3-SAT over varying values of y, for $N = 10^4$ and α from 4.2 to 5.2. The x-axis shows the value of y used, and the y-axis the number of violated constraints returned by each method. These figures show that the performance of RSP over varying y is consistent with Theorem 2: as long as RSP converges, its performance improves as y increases. This property allows for a systematic search for a good value of y to be used.

Random Max-3-SAT

We run experiments on randomly generated Max-3-SAT instances of 10^4 variables, with different clause-to-variable ratios. The random instances are generated by the SP-y code available online (Battaglia, Kolar, and Zecchina 2004). In Figure 4, we compare SP-y and RSP on random Max-3-SAT with different clause-to-variable ratio, α. We vary α from 4.2 to 5.2 to show the performance of SP-y and RSP in the UNSAT region of 3-SAT, beyond its phase transition at $\alpha_c \approx 4.267$. For each value of α, the number of violated constraints (y-axis) is plotted against the value of y used.

We perform a decimation procedure for RSP. We select the variables to decimate by ranking them according to their bias, defined as $|P(x_i = 0) - P(x_i = 1)|$. For each value of y, we run RSP until convergence (for a maximum of 500 iterations, allowing 3 tries with random initializations), decimates the first 100 variables with bias larger than 0.5, and run the algorithm on the remaining problem. We stop the decimation process when the algorithm fails to converge, or when all variables have bias smaller than 0.5. At this point, we run 1000 attempts of walksat on the remaining problem. For SP-y, we run the SP-y code available on line, with the option of decimating 100 variables at each iteration, and with two different settings: with and without backtracking (Battaglia, Kolar, and Zecchina 2004). Backtracking is a procedure used in SP-y to improve performance, by unfixing previously fixed variables at a rate $r = 0.2$, so that errors made by the decimation process can be corrected. For RSP, we do not run backtracking. Note that the y in our formulation equals to $2y$ in the formulation in (Battaglia, Kolar, and Zecchina 2004).

Both SP-y and RSP fail to converge when y becomes large enough. When this happens, the output of the algorithm is the result returned by walksat on the original problem. In Figure 4, we see this happening when a curve reaches a horizontal line, signifying that the algorithm is returning the same configuration regardless of y (we "seed" the randomized walksat so that results are identical when problems are identical). From Figure 4, we see RSP performs more consistently than SP-y: as y increases, the performance of RSP improves, until a point where RSP fails to converge. Hence, the best value of y for RSP is obtainable without going through the decimation process: we can commence decimation at the largest value of y for which RSP converges. In Table 1, we show that RSP (without backtracking) outperforms SP-y, with or without backtracking, for $\alpha \geq 4.7$. We also compare RSP and SP-y with the local search solvers implemented in UBCSAT (Tompkins and Hoos 2004). We run 1000 iterations of each of the 20 Max-SAT solvers in UBCSAT, and take the best result among the 20 solvers. The results are shown in Table 1. We see that the local solvers in UBCSAT does worse than both RSP and SP-y. We have also tried running complete solvers such as toolbar (de Givry et al. 2005) and maxsatz (Li, Manyà, and Planes 2006). They are unable to deal with instances of size 10^4.

Table 1: Number of violated constraints attained by each method. For SP-y, "SP-y (BT)" (SP-y with backtracking), and RSP, the best result is selected over all y. For each α, we show the best performance in bold face. The column "Fix" shows the number of variables fixed by RSP at the optimal y, and "Time" the time taken by RSP (in minutes) to fix those variables, on an AMD Opteron 2.2GHz machine.

α	UBCSAT	SP-y	SP-y(BT)	RSP	Fix	Time
4.2	47	**0**	**0**	**0**	7900	24
4.3	68	9	**7**	10	7200	43
4.4	95	42	**31**	36	8938	82
4.5	128	67	67	**65**	9024	76
4.6	140	98	**89**	90	9055	45
4.7	185	137	130	**122**	9287	76
4.8	232	204	189	**172**	9245	52
4.9	251	223	211	**193**	9208	62
5.0	278	260	224	**218**	9307	66
5.1	311	294	280	**267**	9294	42
5.2	358	362	349	**325**	9361	48

Benchmark Max-SAT instances

We compare RSP with UBCSAT on instances used in (Lardeux, Saubion, and Hao 2005), which were instances used in the SAT 2003 competition. Among the 27 instances,

Table 2: Benchmark Max-SAT instances. Columns: "instance" shows the instance name in (Lardeux, Saubion, and Hao 2005), "nvar" the number of variables, "ubcsat" and "rsp-x" (x is the number of decimations at each iteration) the number of violated constraints returned by each algorithm, and "fx-x" the number of spins fixed by RSP. Best results are indicated in bold face.

instance	nvar	ubcsat	rsp-100	fx-100	rsp-10	fx-10
family: purdom-10142772393204023						
fw	9366	**83**	357	0	357	0
nc	8372	74	**33**	8339	35	8316
nw	8589	73	**24**	8562	28	8552
family: pyhala-braun-unsat						
35-4-03	7383	58	68	7295	**44**	7299
35-4-04	7383	62	53	7302	**41**	7304
40-4-02	9638	86	**57**	9547	65	9521
40-4-03	9638	76	77	9521	**41**	9568

we use the seven largest instances with more than 7000 variables. (On the smaller instances, RSP performs comparably with UBCSAT. These instances might be small enough for local search solvers to perform well).

We run RSP in two settings: decimating either 10 or 100 variables at a time. We run RSP for increasing values of y: for each y, RSP fixes a number of spins, and we stop increasing y when the number of spins fixed decreases over the previous value of y. For UBCSAT, we run 1000 iterations for each of the 20 solvers. Results are shown in Table 2. Out of the seven instances, RSP fails to fix any spins on the first one, but outperforms UBCSAT on the rest. Lardeux et al. (2005) did not show best performances in their paper, but their average results were an order of magnitude higher than results in Table 2.

Figure 4 shows that finding a good y for SP-y is hard. On the benchmark instances, we run SP-y with the "-Y" option (Battaglia, Kolar, and Zecchina 2004) that uses dichotomic search for y: SP-y failed to fix any spins on all 7 instances.

Related work

While recent work on Max-SAT tends to focus more on complete solvers, these solvers are unable to handle large problems. In the Max-SAT competition 2007 (Argelich et al. 2007), the largest Max-3-SAT problems used have only 70 variables. For large instances, complete solvers are still not practical, and local search procedures have been the only feasible alternative. SP-y, generalizing SP, has been shown to be able to solve large Max-3-SAT problems at its phase transition, but lacks the theoretical explanations that recent work on SP has generated.

This paper adapts the RSP algorithm in (Chieu, Lee, and Teh 2007) to the problem of Max-SAT. While Chieu et al. (2007) formulate RSP for general MRFs, they develop the algorithm under settings which result in zero probability for all configurations containing joker states. However, the success of SP algorithms have largely been attributed to the presence of joker states, which allows SP to reason on covers (or clusters) of configurations. In this paper, we adapt the RSP algorithm for Max-SAT, and show that it naturally generalizes the sum-product interpretation of SP.

For 3-SAT, there is an easy-hard-easy transition as the clause-to-variable ratio increases. For Max-3-SAT, however, it has been shown empirically that beyond the phase transition of satisfiability, all instances are hard to solve (Zhang 2001). In this paper, we show that RSP outperforms SP-y as well as other local search algorithms on Max-3-SAT problems, well beyond the phase transition region.

Conclusion

While SP-y does well on Max-SAT problems near the phase transition, the intuition behind SP-y is still unclear from a mathematical point of view. In this paper, we show an alternative algorithm RSP, that not only outperforms SP-y, but also has a clear sum-product interpretation. The mechanisms behind SP-y and RSP are similar: both algorithms impose a penalty term for each violated constraint, and both reduce to SP when $y \to \infty$. SP-y uses a population dynamics algorithm, which can also be seen as a warning propagation algorithm. RSP, on the other hand, uses the well-studied sum-product algorithm. This enables us to understand RSP as estimating marginals over min-covers, which gives a clearer picture on its empirical success.

References

Argelich, J.; Li, C. M.; Manya, F.; and Planes, J. 2007. Second evaluation of max-sat solvers. In *SAT 2007*.

Battaglia, D.; Kolar, M.; and Zecchina, R. 2004. Minimizing energy below the glass thresholds. *Physical Review E* 70.

Braunstein, A.; Mezard, M.; and Zecchina, R. 2005. Survey propagation: An algorithm for satisfiability. *Random Struct. Algorithms* 27(2).

Chieu, H. L.; Lee, W. S.; and Teh, Y. W. 2007. Cooled and relaxed survey propagation for MRFs. In *NIPS 2007*.

de Givry, S.; Heras, F.; Zytnicki, M.; and Larrosa, J. 2005. Existential arc consistency: Getting closer to full arc consistency in weighted csps. In *IJCAI 2005*.

Kroc, L.; Sabharwal, A.; and Selman, B. 2007. Survey propagation revisited. In *UAI 2007*.

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

Lardeux, F.; Saubion, F.; and Hao, J.-K. 2005. Three truth values for the sat and max-sat problems. In *IJCAI 2005*.

Li, C. M.; Manyà, F.; and Planes, J. 2006. Detecting disjoint inconsistent subformulas for computing lower bounds for max-sat. In *AAAI 2006*.

Maneva, E.; Mossel, E.; and Wainwright, M. 2004. A new look at survey propagation and its generalizations. arXiv:cs/0409012v3.

Mezard, M., and Parisi, G. 2003. The cavity method at zero temperature. *Journal of Statistical Physics* 111:1.

Tompkins, D., and Hoos, H. 2004. UBCSAT: An implementation and experimentation environment for SLS algorithms for SAT and MAX-SAT. In *SAT 2004*.

Zhang, W. 2001. Phase transitions and backbones of 3-sat and maximum 3-sat. In *Proceedings of the 7th International Conference on Principles and Practice of Constraint Programming*.