Probabilistic Logic for Intelligent Systems*

T.C. Henderson[1], R. Simmons[1], B. Serbinowski[1], X. Fan[2], A. Mitiche[3], and M. Cline[4]

[1] University of Utah, Salt Lake City, 84112, USA,
tch@cs.utah.edu
[2] University of Swansea, Swansea, UK
[3] INRS, Motreal, Canada
[4] Simon Fraser University, Vancouver, Canada

Abstract. Given a knowledge base in Conjunctive Normal Form for use by an intelligent agent, with probabilities assigned to the conjuncts, the probability of any new query sentence can be determined by solving the Probabilistic Satisfiability Problem (PSAT). This involves finding a consistent probability distribution over the atoms (if they are independent) or complete conjunction set of the atoms. We show how this problem can be expressed and solved as a set of nonlinear equations derived from the knowledge base sentences and standard probability of logical sentences. Evidence is given that numerical gradient descent algorithms can be used more effectively then other current methods to find PSAT solutions.

Keywords: PSAT, probabilistic knowledge base, nonlinear systems

1 Introduction

Given a logical knowledge base of m sentences over n atoms in Conjunctive Normal Form (CNF), and a probability associated with each sentence (conjunction), the probabilisitic satisfiability problem is to determine whether or not a probability distribution exists which assigns consistent probabilities to the complete conjunction basis set of the n atoms in the sentences. This means that for each basis element, $\omega_k, k = 0 \dots 2^n - 1$, $0 \leq \omega_k \leq 1$ and $\sum_k \omega_k = 1$, and, in addition, the sentence probabilities follow from the complete conjunction probabilities. Solutions for this problem were proposed by Boole in 1854 and again by Nilsson 1986; they set up a linear system relating the sentence probabilities to the models of the sentences. The major drawback is that this system is exponential in the number of sentences. Georgakopoulos 1988 showed this problem is NP-complete and exponential in n.

We have proposed that the individual conjuncts in the CNF be converted into a system of nonlinear equations, that is, since each conjunct is a disjunction, it can be re-written in terms of the probabilities of the atoms and conditional probabilities of a smaller number of terms Henderson 2017 and Henderson 2017a. [Note: although this expression itself is exponential in the number of literals in the disjunction, the CNF can always be converted to 3CNF form in polynomial time, and thus, limit the complexity to O(8).] Let the KB have m sentences, and let the k^{th} disjunction, $L_1 \vee R$,

* This research supported in part by Dynamic Data Driven Application Systems AFOSR grant FA9550-17-1-0077.

have probability, p, where L_1 is the first literal in the disjunction, and R represents the remaining literals in the sentence, then the corresponding equation is:

$$F(k) = -p + P(L_1) + P(R) - P(L_1 \mid R)P(R)$$

If R has more than one literal, then any $P(R)$ term will be recursively defined until there are only single literal probabilities, whereas the conditional probabilities will be left as is. Once these equations are developed, each unique probability expression will be replaced with an independent variable. At this point, there is a set of m equations with $n + c$ unknowns, where c is the number of unique conditional probabilities. A scalar error function can then be defined as the norm of the vector F defined above. It is now possible to apply nonlinear solvers to this problem and determine variable assignments that are constrained to be in $[0, 1]$, and which result in the assigned sentence probabilities.

We describe here detailed results of experiments comparing the use of (1) Newton's method, and (2) gradient descent using the Jacobian of the error function. We show that excellent results can be achieved using these methods, and provide a statistical framework for medium-sized KBs (this is to allow the computation of the complete conjunction set probabilities in order to know ground truth), as well some results on a large KB (e.g., 80 variables and over 400 sentences). Although the answer found by the method described here is not guaranteed to be a PSAT solution, it does provide an approximate solution, and in some cases can be determined not to be a solution.

2 Related Work

Stated more technically, consider a logical knowledge base expressed in CNF with m conjuncts over n logical variables: $CNF \equiv C_1 \wedge C_2 \wedge \ldots \wedge C_m$. Each conjunct is a disjunction of literals: $C_k = L_{k,1} \vee L_{k,2} \vee \ldots \vee L_{k,n_k}$ and has an associated probability, p_k. Let $\bar{p}$ be the m-vector of sentence probabilities. Define the complete conjunction set (all combinations of atom truth value assignments): $\omega_k = L_1 \wedge L_2 \wedge \ldots \wedge L_n$, $k = 0 : 2^n - 1$, where the literal assignments correspond to the bit assignments in the n-digit binary number representing k; Ω is the set of all ω_k. The *Probabilistic Satisfiability Problem* is to determine if there exists a probability distribution $\pi : \Omega \to [0, 1]$ such that $0 \leq \pi(\omega_k) \leq 1$ and $\sum_k P(\omega_k) = 1$, and

$$p_k = \sum_{\omega_j \vdash C_k} P(\omega_k)$$

We also let $\bar{\pi}$ represent the vector of probability assignments to the complete conjunctions.

Nilsson described a solution to this problem by creating a linear system based on the CNF sentences and a query sentence (Boole 1854 first proposed a very similar method). The semantic tree (see Kowalski 1983) over the sentence truth assignments is determined and the valid sentence truth assignments are used to form an $m \times 2^m - 1$ matrix, A. I.e., $A(i, j) = 1$ if ω_j satisfies C_i. Then he solves $\bar{p} = A\bar{\pi}$ for $\bar{\pi}$. For more detailed discussion of this approach, including the geometric approach, see Henderson 2017. The problem with this method is the exponential size of the matrix A. For broader discussions of this problem and approach (see Adams 1998, Hailperin 1996, and Hunter 2013).

Others have explored inconsistencies in probabilistic knowledge bases (see Thimm 2009 and Thimm 2012). Another important approach is Markov Logic Networks (see Biba 2009, Domingos

2009 and Gogate 2016). All these methods suffer from exponential cost (e.g., Markov Logic Networks in the number of maximal cliques), or resort to sampling methods to estimate the solution while not guaranteeing the result solves PSAT.

3 Method

The method proposed here, called *Nonlinear Probabilistic Logic (NLPL)* involves conversion of the probabilstic CNF to a set of equations and the use of numerical solvers. First, consider the case when the variables are independent, i.e., $P(A \wedge B) = P(A)P(B)$. For this, the basic formula is:

$$P(A \vee B) = P(A) + P(B) - P(A)P(B)$$

where this is applied recursively if B is not a single literal. Consider Nilsson's example problem (modus ponens):

$$[p_1 = 0.7]C_1 : P$$

$$[p_2 = 0.7]C_2 : \neg P \vee Q$$

with query: $P(Q)$?. This gives rise to the equations:

$$F(1) = -0.7 + P(P)$$

$$F(2) = -0.7 + P(\neg P) + P(Q) - P(\neg P)P(Q)$$

This requires no search (is not exponential) and can be solved:

$$P(P) = 0.7$$

$$P(Q) = \frac{0.7 - 0.3}{0.7} = 0.571$$

$P(Q)$ is in the solution range $[0.4, 0.7]$ and we provide an answer in this interval (note that MLNs produce a single answer as well: the maximal entropy solution).

A first question is: Does *NLPL* solve PSAT? Suppose that *NLPL* finds a probability assignment for the atoms that results in the sentence probabilities (using the nonlinear equations), where each atom probability is in the interval $[0, 1]$; i.e., the atom probability vector is in the unit hypercube in n-space. We need to show that the complete conjunction probabilities generated satisfy the properties described above.

Theorem: Given a set of atom probabilities, a_k, $k = 1 : n$, with $0 \leq a_k \leq 1$, then each complete conjunction probability is in the range $[0, 1]$ and the $\sum_k \omega_k = 1$.

Proof: First, note that since the variables are independent, the complete conjunction probabilities are the product of n numbers each between 0 and 1; thus their product is between 0 and 1. Next, we will show that the sum of the complete conjunction probabilities is 1. Each $\omega \in \Omega$ is the product of n distinct literals. Given variable independence, then:

$$P(\omega_k) = P(L_1 \wedge L_2 \wedge \ldots \wedge L_n)$$

$$= P(L_1)P(L_2)\ldots P(L_n)$$

where each $P(L_i)$ is either $P(a_i)$ or $1 - P(a_i)$. We show the theorem by induction on n.

Case $n = 1$: Then there are 2 complete conjunctions: A and $\neg A$; the sum of their probabilities is: $P(A) + (1 - P(A)) = 1$.

Case n: Suppose the theorem holds for $n - 1$. For n variables, there are 2^n summands which can be paired as follows:

$$P(L_1)P(L_2)\ldots \wedge P(L_k) \wedge \ldots \wedge P(L_n)$$

$$P(L_1)P(L_2)\ldots \wedge P(\neg L_k) \wedge \ldots \wedge P(L_n)$$

When these are summed, the result is:

$$(P(L_k) + (1 - P(L_k)))(P(L_1)\ldots \wedge P(L_{k-1})$$

$$\wedge P(L_{k+1}) \wedge \ldots \wedge P(L_n))$$

which reduces to an expression in $n - 1$ variables. Applying this to all appropriate pairs results in a sum of elements of length $n - 1$. **QED**

Call n-modus ponens the problem with conjuncts $A_1, \neg A_1 \vee A_2, \ldots, \neg A_{n-1} \vee A_n$. Then the standard approach needs 2^n models (as does MLNs), whereas we solve it in linear time.

3.1 When Variables are Not Independent

Suppose the independence of variables is not assumed. Then:

$$P(A \vee B) = P(A) + P(B) - P(A \mid B)P(B)$$

This equation is developed recursively for any non-singleton variable probability, but conditional probabilities are considered as unique unknowns. Solving the nonlinear system provides a set of atom and conditional probabilities. However, this may not be part of a PSAT solution; i.e., the space of solutions for this system contains the solutions (if they exist) for independent variables. In general, we propose to use the solution found by gradient descent.

Although we cannot guarantee that the result of the algorithm is a PSAT solution, there is a check to determine some cases when it is not. Given a conditional probability over two variables (i.e., one of $P(A \mid B)$, $P(\neg A \mid B)$, $P(A \mid \neg B)$, or $P(\neg A \mid \neg B)$) the other three can be determined from it. If any of these is not in the range [0,1], then the result is not a PSAT solution. This same check can be applied to conditionals over three variables as well.

3.2 Numerical Solutions

Given a set of nonlinear equations resulting from a CNF KB and the associated sentence probabilities, it is necessary to create the sentence error function, find an intitial guess at a solution, and then apply Newton's method or some other technique. We have applied two methods: (1) Newton's method, and (2) gradient descent using the Jacobian.

Newton's Method Given the vector function F defined above (a vector function of m elements), Newton's method iterates the following until within tolerance of a solution:

1. Produce next step vector
   $H_{F(\bar{x}^k)}\bar{s} = \nabla F$
2. Move toward solution
   $\bar{x}^{k+1} = \bar{s} + \bar{x}^k$

where H_F is the Hessian matrix for F, $\bar{s}$ is the step vectorm and $\bar{x}^k$ is the k_{th} solution estimate. The development of the Hessian is done symbollically then solved numerically in Matlab. This imposes constraints of the application of this method to larger problems. However, we give results below for moderate size KB's.

Gradient Descent using the Jacobian Gradient descent using the Jacobian should have q-quadratic convergence when starting not too far from a solution, but may hit a local (non-solution) minimum otherwise. The method iterates until within tolerance of a solution as follows:

1. Determine the Jacobian
   $J = \nabla F$
2. Move toward lower sentence error
   $\bar{x}^{k+1} = \alpha * J(\bar{x}^k) + \bar{x}^k$

where α is the step size.

4 Experiments and Results

4.1 Independent Variables

We have tested this approach on sets of randomly generated knowledge bases. This involves selecting a number of variables (n), specifying a maximum number of sentences to generate, as well as the maximal length of any one sentence. A set of sentences is generated which satisfies these constraints, and then a set of probabilities are produced for the compete conjunction set, and from these the sentence probablities are computed. This ensures that there is a solution, although it does not preclude the existence of other solutions (generally a non-zero measure subset of the unit hypercube).

A set of 100 KB's was generated this way, with $n = 5$, $m_{max} = 30$, and $len_{max} = 5$. Figure 1 shows the number of iterations required by Newton's method to solve PSAT; the blue trace shows when initial points are far from the known solution, and red when they are near (within 0.5 vector norm). The mean number of iterations is 4.26 when starting near, and 12.63 when starting far. The method fails on 6 of the 100 KB's. As for gradient descent, Figure 2 shows the number of required iterations. Although a few KB's require over 1000 iterations, the mean number of iterations required when starting near a solution is 21.19, and when starting far is 171.58. Note that the search is terminated when a sentence error of less than 0.01 is reached.

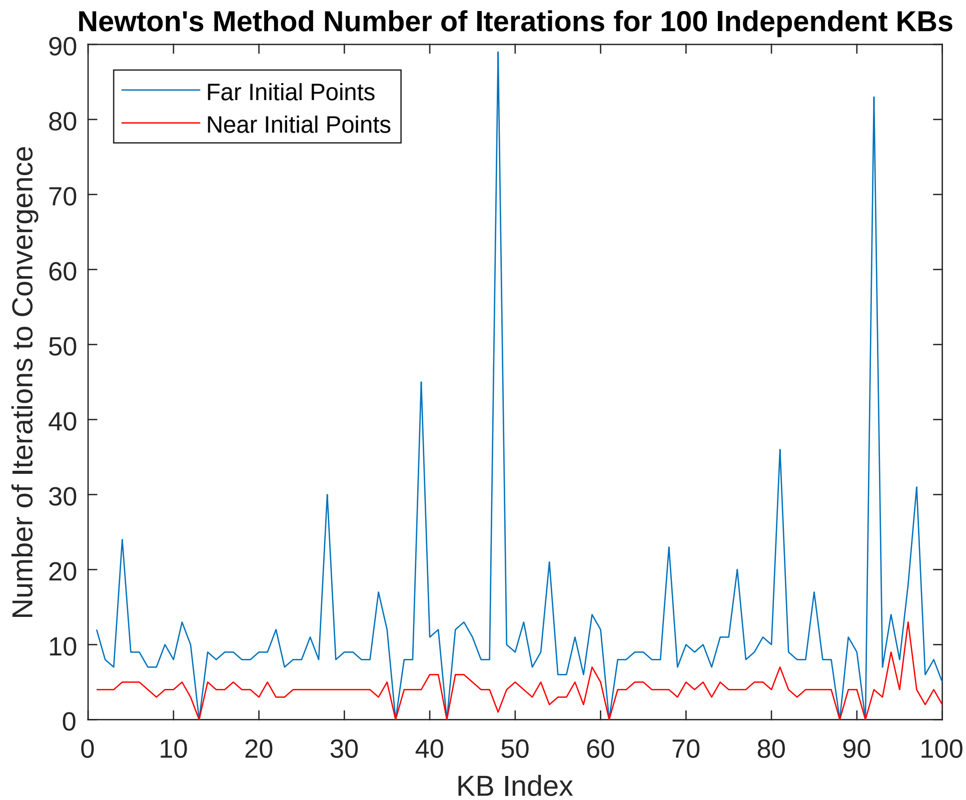

Fig. 1. Newton's Method Results for 100 KB's.

4.2 Non-Independent Variables

The equations must include variables for whatever conditional probabilities arise from the sentences, and are thus a bit more complicated. Figure 3 shows the number of iterations required for Newton's method on 100 random general KB's with the same parameters as above. In this case, solutions were found for 75 of the 100 KB's, and the mean number of iterations was 3.91 when starting near the known solution, and 10.27 when starting far from it. Figure 4 shows the results for gradient descent (which found solutions for all the KB's) and had mean number of iterations 662.17 for far starting points and mean number of iterations 638.61 for near points.

What these results indicate is that Newton's method should be tried first given the low iteration cost, and then gradient descent used if Newton's Method fails. Also, note that even though in the case of failure (i.e., local minimum found), the methods were allowed to re-start at new random initial locations. Gradient descent only re-started this way twice and then only tried 2 alternate points. When Newton's Method finds a solution is does so with the initial guess; when it failed, it did so for both near and far initial starting points.

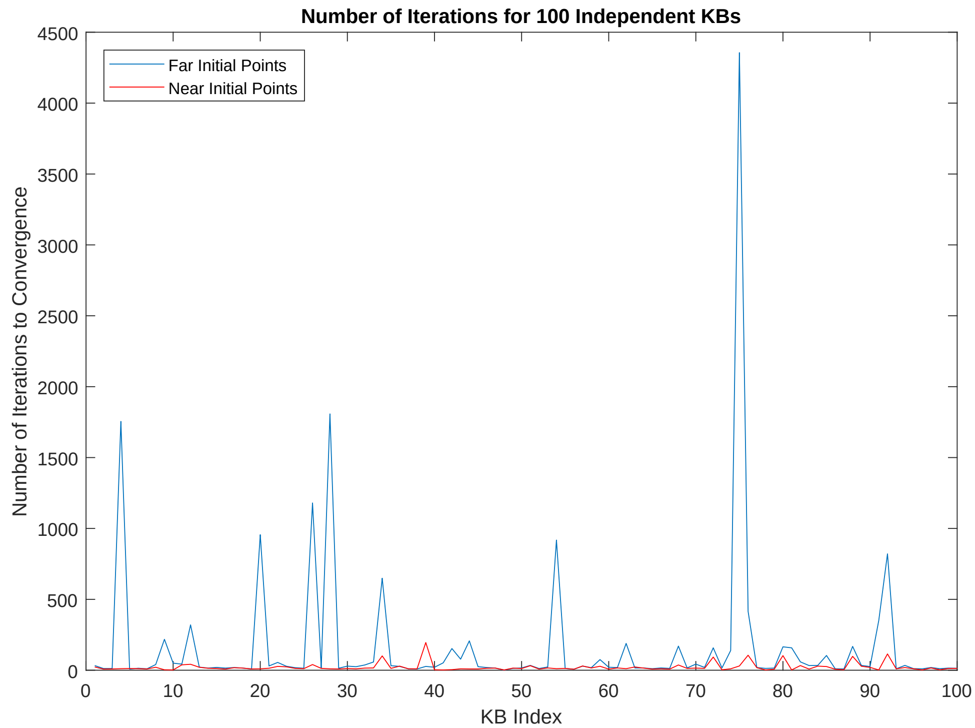

Fig. 2. Gradient Descent Results for 100 KB's.

Finally, Figure 5 shows the maximal individual atom probability error comparing the atom probabilities from the actual 100 general KBs to the atom probabilities found by the numerical solver. The mean of the max atom probability error for near starting points is 0.09, while for far starting points is 0.10. This is very promising in that the discovered solutions are near the actual underlying solution for most KBs.

4.3 Trajectory Visualization and Finding Good Initial Guesses

As pointed out above, if the initial guess is too far from a solution, these methods may not converge. Thus, it would help to be able to identify good starting points. In order to get insight into the convergence sequence, we have developed a visualization method which maps n-D points to 2-D points. Given a point, $\bar{a}$, in n-D, define the corresponding 2-D coordinates as follows:

$$x = \sum_{i=1}^{n} (a_i cos(\frac{(i-1)\pi}{n}) \tag{1}$$

$$y = \sum_{i=1}^{n} (a_i sin(\frac{(i-1)\pi}{n}) \tag{2}$$

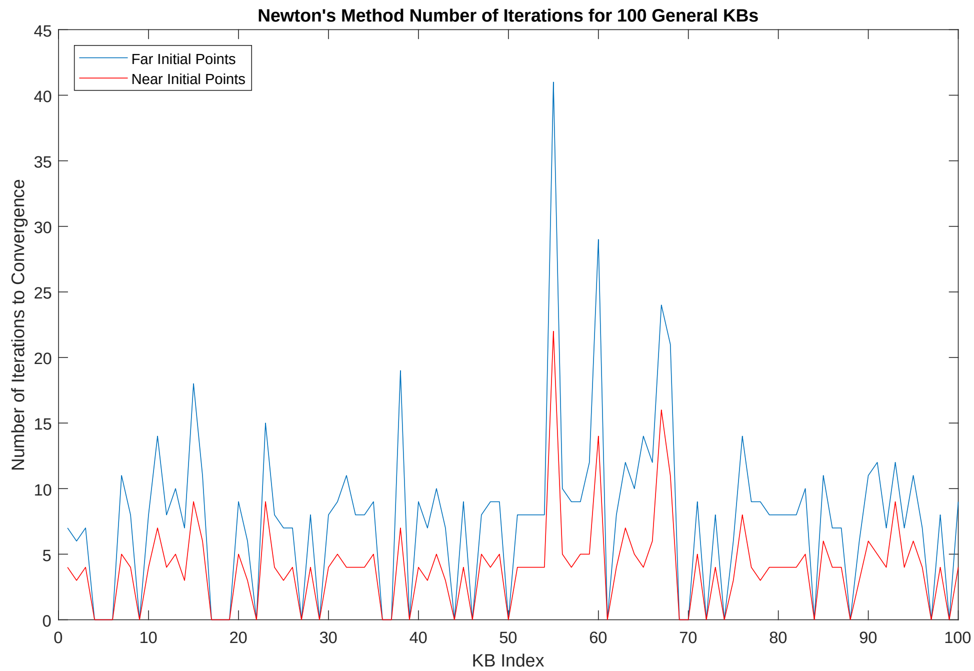

Fig. 3. Newton's Method Results for 100 KB's.

Figure 6 shows the convergence trajectories for four different initial points. The q-convergence of the method can be estimated by determining the c_k's in the following equation:

$$| \bar{x}^{k+1} - \bar{x}^* | \leq c_k | \bar{x}^k - \bar{x}^* | \tag{3}$$

Figure 7 shows these values for the 100 tracks for gradient descent on the general KB's. The plots indicate that the method is q-superlinear/quadratic.

Another interesting aspect of this visualization method is its use to find good starting points. Given fixed x and y in the plane, we have have developed a method to obtain a unique point in the pre-image of Eqns (1) and (2). Each equation defines a hyperplane in n-space; taken together they represent a hyperplane of dimension $n-2$. One way to understand the map defined by Eqns (1) and (2) is as an n-joint prismatic manipulator, where joint k translates in the direction $\theta = \frac{(k-1)\pi}{n}$. The manipulator's workspace is a $2n$-gon (as shown in Figure 6). By uniformly sampling this workspace, and then finding pre-image points in n-D, the sentence error can be found, and then the lowest such value used to pick the initial point. Of course, since there is an infinite number of potential pre-image points for each x and y location, other methods can be used to sample that subspace to find better starting points.

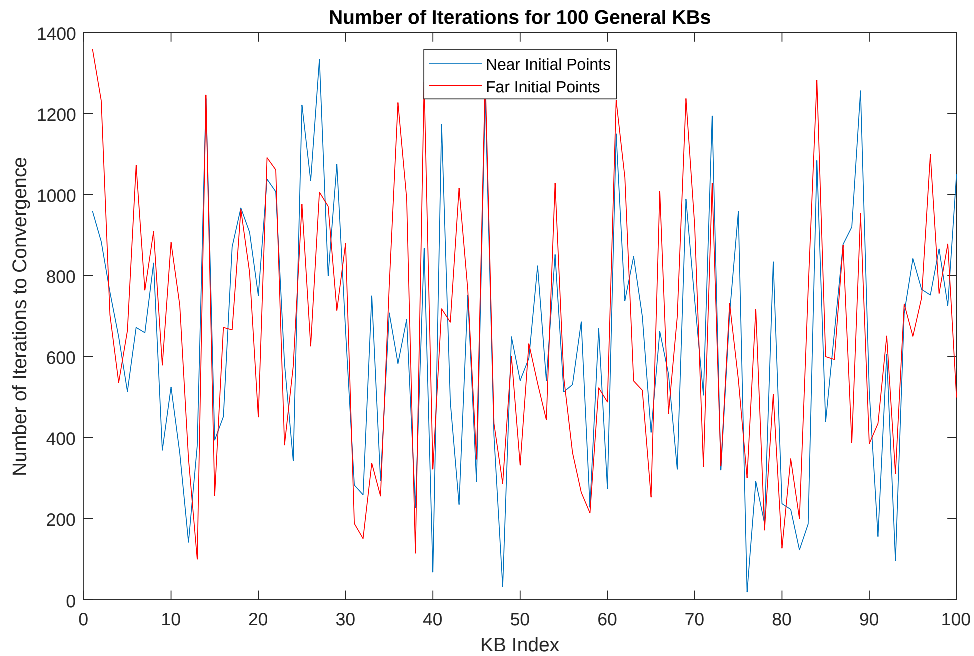

Fig. 4. Gradient Descent Results for 100 KB's.

5 Conclusions and Future Work

We propose a novel approach to approximately solve PSAT (or at least as much of it as is useful) which avoids the computational complexity of previous methods as well as the error introduced using MC-SAT methods. Instead we solve a system of nolinear equations derived directly from the meaning of the probability of the logical sentences. The experiments reported here show that solving these systems is possible and not overly complex (evidence shows q-superlinear/quadratic convergence). The number of variables and sentences used in these experiments are modest as is required so that the ground truth can be ascertained (i.e., that there is a solution!); however, in previous work we have demonstrated that the method can solve problems with 80 variables and over 400 sentences (Henderson 2017).

It is also possible to convert SAT instances into the form of PSAT by assigning every clause the probability of 1, and then apply numerical solvers to this problem. In particular, if we limit our input to 3-SAT instances, we can recursively expand the equation for a CNF equation to:

$$P(A \vee B \vee C) = P(A) + P(B) + P(C) - P(A \wedge B)$$

$$-P(A \wedge C) - P(B \wedge C) + P(A \wedge B \wedge C)$$

where we can treat each distinct probability as a separate variable. Note that we restrict input to 3-SAT instances because this recursive expansion is exponential in the number of variables in the

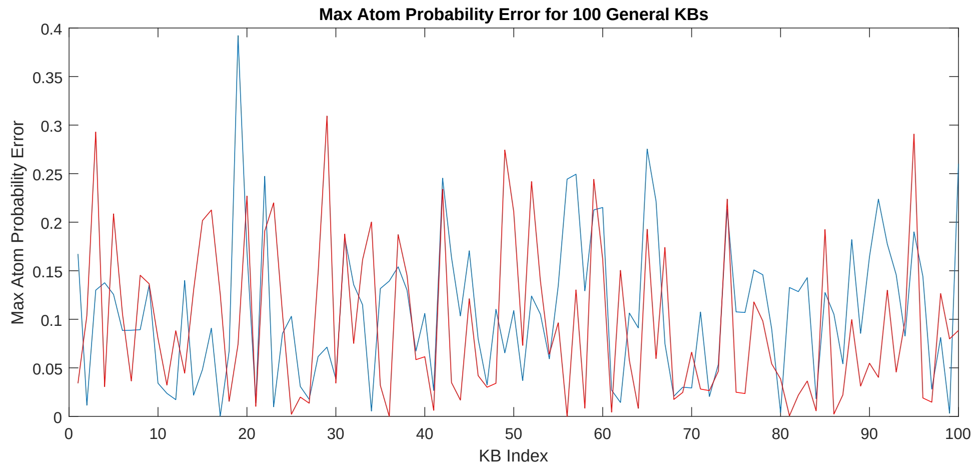

Fig. 5. Maximum Atom Probablity Error for 100 General KBs (Near Starting Points in red and Far Starting Points in blue).

clause. We can also force consistency constraints given, for any literals L_i, L_j, L_k:

$$P(L_i) = P(L_i \wedge L_j) + P(L_i \wedge \neg L_j)$$
$$P(L_i \wedge L_j) = P(L_i \wedge L_j \wedge L_k) + P(L_i \wedge L_j \wedge \neg L_k)$$

There are O(n^3) such constraints. The system of equations that this creates is far too large to be a practical SAT solver, but it is nevertheless a polynomial number of linear equations, which can be solved in polynomial time. If some π can be found that satisfies the new sentences, every w_k such that $\pi(w_k) \geq 0$ is a solution to the original SAT equation. Every sentence has probability 1, and since the sum of all $\pi(w_k)$ must be equal to 1, only atom truth assignments that satisfy every clause may hold probability. As such, if it can be shown that nonindependent solutions can be used to construct valid probabilities for the complete conjunction set, then $P = NP$.

Other future work includes the investigation of:

1. The problem encountered with Newton's Method. It is possible that the Hessian as computed does not remain positive definite which can cause failure. It may be possible to address this with SVD methods.
2. The discovery of good initial starting points. For this, the trajectory visualization method will be studied; i.e., the inverse kinematics of the planar n-joint prismatic manipulator.
3. The exploitation of the method to support a knowledge base providing probabilistic logic and in the future, argumentation. such a capability will provide decision makers and analysts a robust estimate of the confidence of a statement or the consequences of an action. The current application domain for this is geospatial knowledge bases (see Henderson2017b).

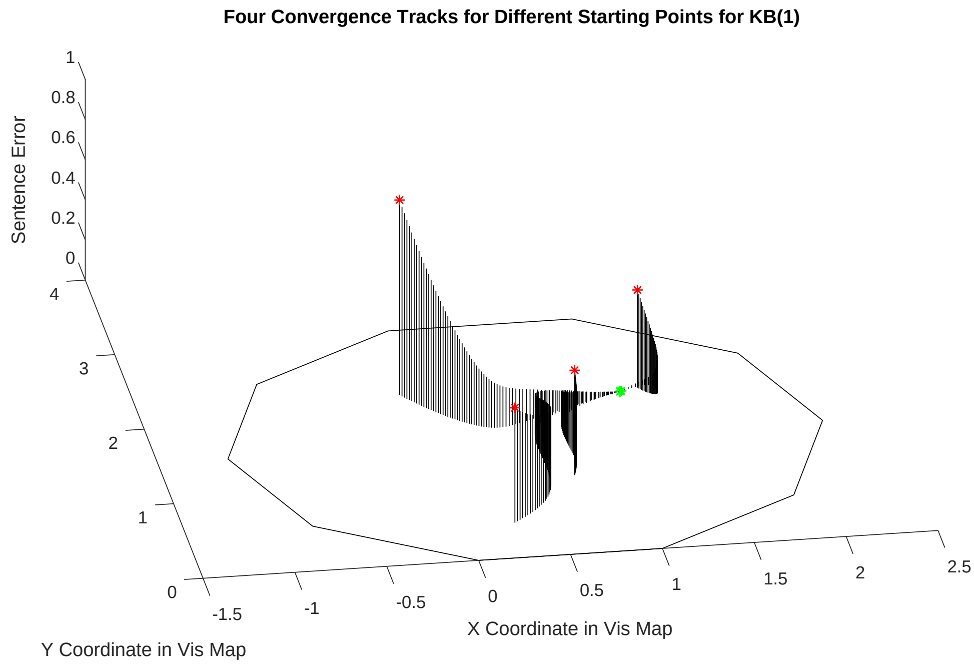

Fig. 6. Convergence Tracks for 4 Random Starting Points for 5-D Problem; x and y values are projections of 5-D points to 2-D, and z value is sentence error value.

6 References

Adams, E.W.: A Primer of Probability Logic. CLSI Publications, Stanford, CA, 1998.

Biba, M.: Integrating Logic and Probability: Algortihmic Improvements in Markov Logic Networks. PhD Thesis, University of Bari, Bari, Italy, 2009.

Boole, G.: An Investigation of the Laws of Thought. Walton and Maberly, London, UK, 1854.

Domingos, P. and Dowd, D.: Markov Logic: An Interface Layer for Artificial Intelligence. Morgan and Claypool, San Rafael, CA, 2009.

Georgakopoulos, G, Kavvadiass, D. and Papadimitriou, C.H.: Probabilistic Satisfiability. Journal of Complexity, 4:1–11, 1988.

Gogate, V. and Domingo, P.: Probabilistic Theorem Proving. Communications of the ACM, 59(7):107–115, 2016.

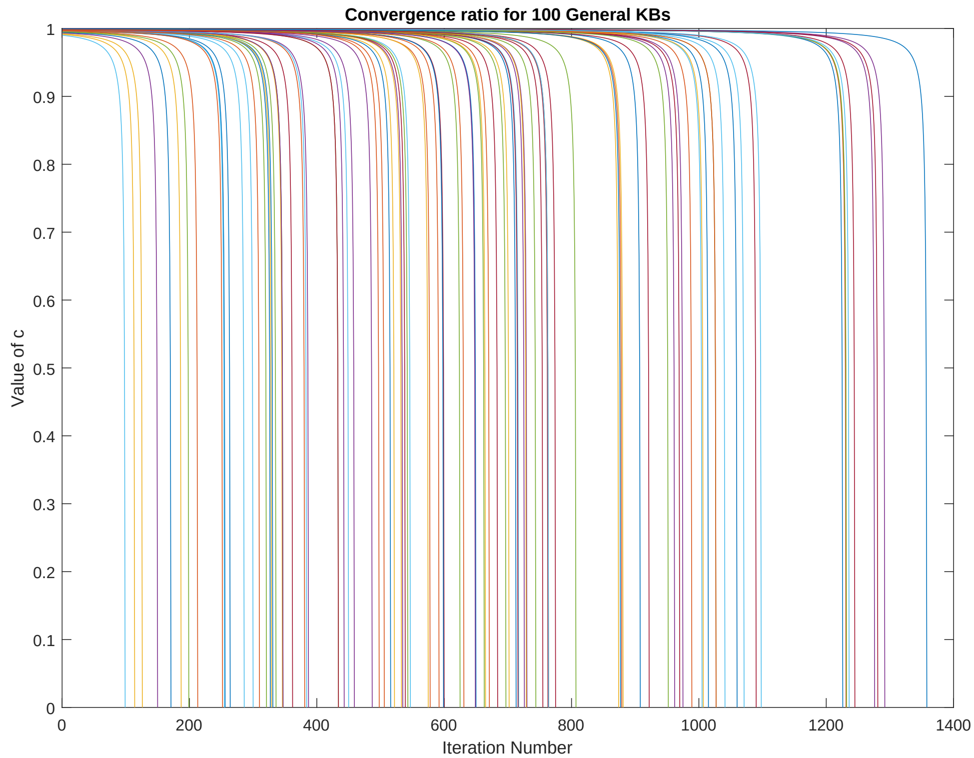

Fig. 7. Convergence step ratio for 100 general KB's using Gradient Descent.

Hailperin, T.: Sentential Probability Logic. Lehigh University Press. Cranbury, NJ, 1996.

Henderson, T.C., Mitiche, A., Simmons, R. and Fan, X.: A Preliinary Study of Probabilistic Argumentation. Technical Report UUCS-17-001, University of Utah, February, 2017.

Henderson, T.C., Simmons, R., Mitiche, A., Fan, X. and Sacharny, D.: A Probabilistic Logic for Multi-source Heterogeneous Information Fusion. In Proceedings of the IEEE Conference on Multi-sensor Fusion and Integration for Intelligent Systems, Daegu, S. Korea, November, 2017b.

Hunter, A.: A Probabilistic Approach to Modelling Uncertain Logical Arguments. Internatoinal Journal of Approximate Reasoning, 54:47–81, 2013.

Kowalski, R. and Hayes, P.J.: Semantic Trees in Automatic Theorem Proving. In Automation of Reasoning, pp. 217–232, Berlin, 1983.

Nilsson, N.: Probabilistic Logic. Artificial Intelligence Journal, 28:71–87, 1986.

Sacharny, D., Henderson, T.C., Simmons, R., Mitiche, A., Welker, T. and Fan, X.: BRECCIA: A Novel Multi-Source Fusion Framework for Dynamic Geospatial Data Analysis. In Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, Daegu, S. Korea, November, 2017.

Thimm, M.: Measuring Inconsistency in Probablilistic Knowledge Bases. In Proceeding of the 25th Conference on Uncertainty in Artificial Intelligence, pp. 530–537, Montreal , Quebec, Canada, June, 2009.

Thimm, M.: A Probablilistic Semantics for Abstract Argumentation. In Proceeding of the 20th European Conference on Artificial Intelligence, Montpellier, France, August, 2012.