Bayesian Networks: Performance Analysis of Loopy Belief Propagation

BY

XIANGQIONG SHI
B.S., Shanghai Jiao Tong University, Shanghai, China, 2003
M.S., Shanghai Jiao Tong University, Shanghai, China, 2006

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2011

Chicago, Illinois

Defense Committee:

Dan Schonfeld, Chair and Co-Advisor
Daniela Tuninetti, Co-Advisor
Natasha Devroye
Stephen Yau, Mathematics, Statistics, and Computer Science
Junhui Wang, Mathematics, Statistics, and Computer Science

ACKNOWLEDGMENTS

I would like to thank my advisors, Prof. Dan Schonfeld and Prof. Daniela Tuninetti, for supervising me to the accomplishment of this thesis. Their preciseness in theoretical analysis, enthusiasm toward innovational work, and persistence in good academic writing, influence me a lot. Their humors add flavor to project discussions during group meetings. I would also like to thank the defense committee members Prof. Stephen Yau, Prof. Natasha Devroye, Prof. Junhui Wang, and one of my prelim committee members Prof. Jing Wang, for their suggestions and support.

I would like to thank my labmates in MCL, faculty and staffs in ECE department. Special thanks go to Junlan Yang, Chong Chen, Xiang Ma, Pan Pan, Liuling Gong, Liming Wang, for their help in my life, study and research. I would also like to thank my other friends in Chicago for their help.

Finally but importantly, I would like to thank my parents and my husband, for their love, support, and encouragement. I feel so lucky and grateful to have them in my life.

XS

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF ABBREVIATIONS

LBP	Loopy Belief Propagation
BP	Belief Propagation
MRFs	Markov Random Fields
MAP	Maximum A Posteriori
SAW	Self-Avoiding Walk
UDB	Uniform Distance Bound
NUDB	Non-uniform Distance Bound

SUMMARY

Loopy belief propagation (LBP) is a probability inference algorithm on Graphical Models. It has been successfully applied in low-density parity-check codes, turbo codes, computer vision problems such as stereo matching, image segmentation and image impainting. Belief propagation was first proposed by Judea Pearl (1) in 1982 for tree-structured graphs, and was extended to polytrees (2) in 1983, and later applied to general graphs with empirical success. Since LBP ignores the loopy structure of graphs and blindly propagates messages, many researchers have focused in analyzing its performance of convergence and accuracy, and presented variations of message passing algorithm. In this thesis, we focus on the performance analysis of LBP with respect to error bound and convergence. We present novel results that show the relationship between the performance of LBP and the strength of potential functions as well as topology structure of graphical models. Specifically, we first present lower-bound and upper-bound on multiplicative message errors. Then we present uniform and non-uniform distance bound on beliefs, which can improve an existing accuracy bound between beliefs and true marginals. Thereafter, we present our non-uniform sufficient convergence condition for LBP, which is shown to be better than existing conditions. We also show our error bound is related with the rate of convergence of LBP. Furthermore, we analyze fixed points of completely uniform binary graph, and present tight distance bound between beliefs. Finally, we implement LBP algorithm for an image segmentation application, and show how the segmentation performance is affected by some parameters in potential functions.

CHAPTER 1

GRAPHICAL MODEL AND LOOPY BELIEF PROPAGATION

1.1 Introduction to Graphical Model and Loopy Belief Propagation

Graphical models, which are widely used in pattern recognition and machine learning, are a marriage between graph theory and probability theory for multivariate statistical modelling. They are used to intuitively provide conditional independency between random variables. Utilizing the structure of graphical models, computation complexity of inference and learning on random variables will be greatly reduced.

Recent years have seen a dramatic increase of publication about graphical models used in labelling problems for computer vision, because of the overwhelmingly large number of states of labels for those problems. Graphical model was presented in (3) to solve the labelling problems of super-resolution and color demosaicing. (4) presented graphical models unifying motion information, boundary information and spatial connectivity for spatiotemporal segmentation of video sequences. A general object detection framework using Hidden Markov Model and Discriminative Random Fields was presented in (5) for text detection. (6) modeled image denoising and inpainting problems using stochastic factor graphs and used variational expectation maximization algorithms to local distribution functions. Layered graphical models were proposed by (7) to track partially-occluded objects using an image plane representation of object motion.

(8) used conditional random fields and implicit deformable models to solve a joint Maximum a Posteriori problem for 3D image segmentation.

However, probabilistic inference for large-scale multivariate random variables, such as image labeling variables, is very expensive computationally. Belief propagation (BP) algorithms are designed to reduce the computational burden by exploiting the factorization of joint density functions captured by the topological structure of graphical models (9; 10; 11; 12). BP is known to converge to the exact inference on acyclic graphs (i.e. trees) or graphs that contain a single loop. In the case of graphs with multiple loops, BP results in an iterative method referred to as loopy belief propagation (LBP). The use of LBP generally provides remarkably good approximations in real-world applications; e.g., turbo decoding and stereo matching (13; 14).

Because LBP does not always converge, sufficient conditions for its convergence have been extensively investigated in the past using various approaches (15; 16; 17; 18). Necessary conditions for convergence of LBP, however, remain unknown. (15) related convergence of LBP to the uniqueness of a sequence of Gibbs measures defined on the associated computation tree. He subsequently developed a testable sufficient condition for convergence of LBP by applying Simon's condition (19). (16) presented sufficient conditions for uniqueness of fixed points in LBP by relying on the uniqueness of minima of the Bethe free energy. He related the strength of the potentials with the convergence of the LBP algorithm, which leads to milder sufficient conditions than those obtained by exclusively relying on the structure of the graph.

Recently, several papers have investigated the message updating functions of the LBP algorithm as contractive mappings. (17) analyzed the contractive effect of message-error prop-

agation in belief networks using the dynamic-range measure as a metric, and obtained error bounds and sufficient conditions for convergence of LBP message passing. (18) derived sufficient conditions for convergence of LBP based on quotient norms of contractive mappings, which are invariant to scaling and shown to be valid for potential functions containing zeros.

Although the beliefs may not be true marginal probabilities when the LBP algorithm converges, they have been shown to provide good approximations by (20). When the LBP algorithm does not converge, however, beliefs are not good approximations of true marginals because the Bethe free energy does not provide a good approximation of the Gibbs-Helmholtz free energy (21). Exactness and accuracy of the LBP algorithm has consequently gained interest in recent years. (22) derived bounds on exact marginals by relying on the girth of the graph (i.e. the number of edges in the shortest cycle in the graph) and the properties of Dobrushin's interdependence matrix (23). (24) used Dobrushin's theorem to present a distance bound on the marginal probabilities. (25) introduced a distance bound on the error between beliefs and marginals based on recent results for computing marginal probabilities for pairwise Markov random fields using Self-Avoiding Walk (SAW) trees (26). (27) propagate bounds on marginal probabilities over a subtree or the SAW tree of the factor graph, and demonstrate that their bounds perform well in terms of accuracy and computation time of LBP.

Several investigators have explored the consequence of scheduling on the convergence of BP. (28) discussed the impatient and lazy belief propagation algorithms and showed that the former is expected to converge faster than the latter. (29) proposed a residual belief propagation algorithm, which schedules messages in an informed manner thus significantly reducing the

running time needed for convergence of LBP. Inspired by (29)'s work, (30) further increased the rate of convergence by estimating the residual rather than computing it directly.

In this thesis, we derive uniform and non-uniform error bounds on LBP, which are tighter than existing ones in literature, and use these bounds to study the dynamic behavior of the sum-product algorithm. We subsequently use these bounds to derive sufficient conditions for the convergence of the sum-product algorithm, and analyze the relation between convergence of LBP and sparsity and walk-summability of graphical models. We finally use the bounds derived to investigate the accuracy of LBP, as well as the scheduling priority in asynchronous LBP. A preliminary version of some of the error bounds in this thesis has appeared in (31).

1.2 Basics of Graphical Model

In this section, we will introduce the basics of graphical model. Readers are kindly suggested to refer to (9)(10)(12)(32) for more details.

A graph is composed of nodes connected by edges. In a probabilistic graphical model, nodes correspond to random variables and edges represent probability relationship between them. The joint distribution over all the random variables can be decomposed into a product of factors according to the Markov property. When edges are directed, graphical models are generally referred to as belief networks or Bayesian networks; when undirected, graphical models are referred to as Markov random fields(MRFs). Directed graphs express causal relationships between random variables, while undirected graphs impose compliance constraints between random variables. In this thesis, we will mainly discuss MRFs.

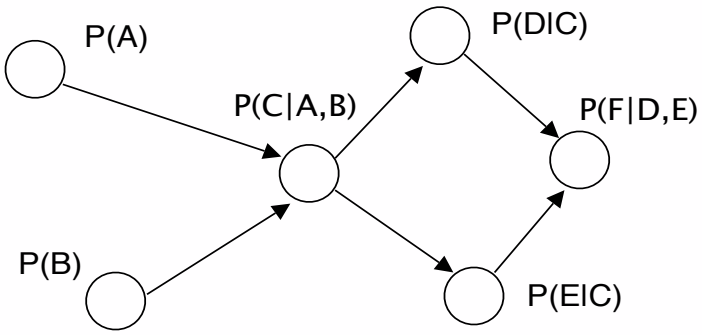


Figure 1. An Example of Bayesian Network

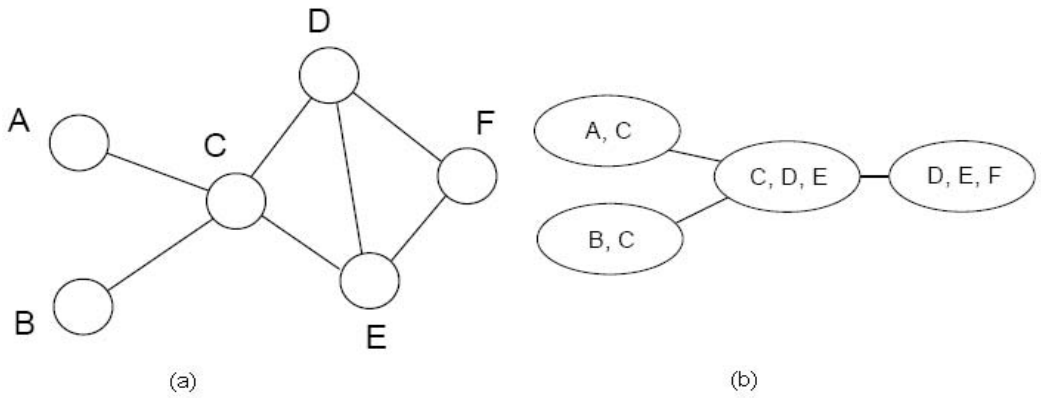


Figure 2. An Example of Markov Random Field

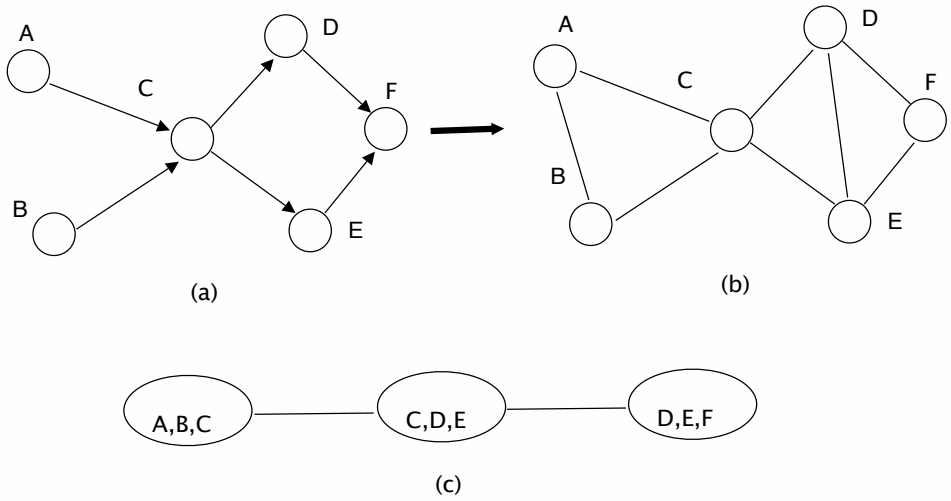


Figure 3. Directed Graph Transformed to Undirected Graph

1.2.1 Conditional Independency on Directed and Undirected Graph

Directed edges in Bayesian networks interpret parent-children relationship, whereas undirected edges in Markov random field show an interaction between neighboring nodes. For a Bayesian network, each variable is only dependent on its parents. Let $\{S\}$ represent the set of random variables corresponding to the N nodes of the graph. Let $pa\{S_i\}$ denote the set of parents of nodes S_i. Then $P(S) = \prod_i P(S_i|pa(S_i))$. For example, in Figure 1, the global probability P(A,B,C,D,E,F) can be factorized into product of the local probabilities

$$P(X_A, X_B, X_C, X_D, X_E, X_F)$$
$$= P(X_A)P(X_B)P(X_C|X_A, X_B)P(X_D|X_C)P(X_E|X_C)P(X_F|X_D, X_E).$$

Similarly, for a Markov random field, each node is only dependent on its neighbors. In Figure 2(a), A is independent of D given C. Each edge in the graph indicates a compliance relation between two nodes, which is expressed by a potential function over the corresponding random variables. Thus, joint distribution over a graph can be factorized into product of potential functions. For instance, $\phi_{A,C}(X_A, X_C)$, $\phi_{B,C}(X_B, X_C)$, $\phi_{C,D}(X_D, X_C)$, $\phi_{C,E}(X_C, X_E)$,$\phi_{D,E}(X_D, X_E)$, $\phi_{D,F}(X_D, X_F)$, $\phi_{E,F}(X_E, X_F)$ are the potential functions on edges AC, BC, CD, CE, DE, DF and EF. Then

$$P(X) = \frac{1}{Z}\phi_{A,C}(X_A, X_C)\phi_{B,C}(X_B, X_C)\phi_{C,D}(X_D, X_C)\phi_{C,E}(X_C, X_E) \times\phi_{D,E}(X_D, X_E)\phi_{D,F}(X_D, X_F)\phi_{E,F}(X_E, X_F),$$

where Z is the normalization constant.

Undirected graphs can also be expressed using clique graphs. Thus, global distribution can be factorized into product of potential functions on cliques. Let us denote a clique by C and the set of variables in that clique by X_C. Then the joint distribution can be written as a product of potential functions by $\psi_C(X_C)$ over the maximal cliques of the graph

$$P(X) = \frac{1}{Z}\prod_C \psi_C(X_C),$$

where Z is called partition function,which is a normalization constant given by $Z = \sum_x \prod_C \psi_C(X_C)$. For Figure 2 (a), the corresponding clique graph is shown in Figure 2 (b). Thus the joint distribution can be expressed as

$$P(X) = \frac{1}{Z}\psi_{A,C}(X_A, X_C)\psi_{B,C}(X_B, X_C)\psi_{C,D,E}(X_C, X_D, X_E)\psi_{D,E,F}(X_D, X_E, X_F).$$

Directed graphical model in Figure 1 can be transformed into undirected graphical model by removing the arrows on the edge and moralizing the parents of each node, which is shown in Figure 3 (a) and (b). The corresponding clique graph which contains the parents information is shown in figure Figure 3(c).

1.2.2 Factor Graphs

Both directed and undirected graphs can be represented by a factor graph. Factor graph allows the global function to be expressed by a product of factors over subsets of variables. For the graph in Figure 4, the global distribution over x_1, x_2, x_3 is

$$P(X) = \alpha\phi(x_1, x_2)\phi(x_1, x_3)\phi(x_2, x_3)\varphi(x_2, x_3).$$

Undirected graph in Figure 4(a) can be expressed by factor graph in Figure 4(b) with factor

$$f(x_1, x_2, x_3) = \phi(x_1, x_2)\phi(x_1, x_3)\phi(x_2, x_3)\varphi(x_2, x_3).$$

It can also be expressed by factor graph in Figure 4(c) with factor

$$f_a(x_1, x_2, x_3) = \phi(x_1, x_2)\phi(x_1, x_3)\phi(x_2, x_3),$$

$$f_b(x_2, x_3) = \varphi(x_2, x_3).$$

For the directed graph in Figure 5 (a), the global distribution is $P(X) = p(x_1)p(x_2)p(x_3|x_1, x_2)$. The factor graph representing the same distribution is shown in Figure 5(b) with factor

$$f(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2).$$

Otherwise, a different factor graph for the same distribution is shown in Figure 5 (c) with factor

$$f_a(x_1) = p(x_1), f_b(x_3) = p(x_3),$$

$$f_c(x_1, x_2, x_3) = p(x_3|x_1, x_2).$$

We can see that factor graph is not unique for directed or undirected graphs.

In the following, we will work on undirected graphs with pairwise potential functions. In other words, in the corresponding factor graph, each factor will control at most two random variables.

1.3 Efficient Inference and Belief Propagation

Probability inference for large scale multivariate random variables will be very computational complex. Efficient inference algorithm(33) can be obtained based on the structure of

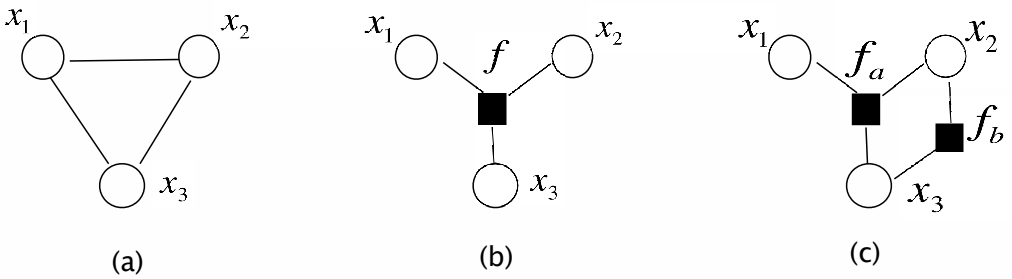


Figure 4. Factor Graph for Undirected Graph

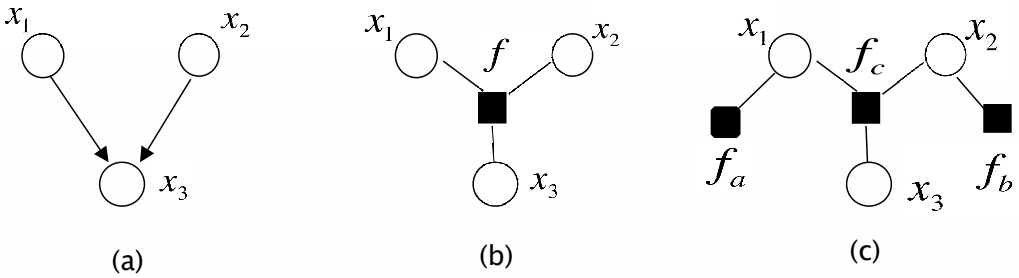


Figure 5. Factor Graph for Directed Graph

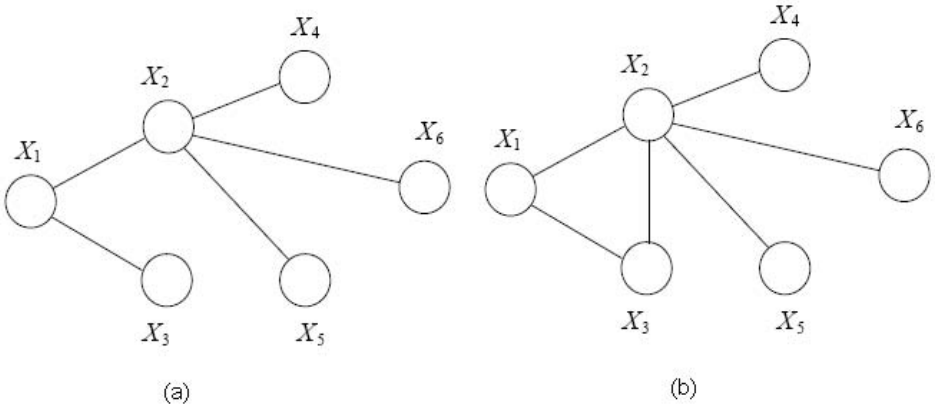


Figure 6. (a)A Tree Example (b)Graph With One Loop

the graphical model. In this section, we will introduce several probability inference algorithms: belief propagation, tree reweighted message passing, generalized belief propagation and norm-product belief propagation.

1.3.1 Belief Propagation

Belief propagation is essentially a dynamic programming algorithm, which utilizes the the intermediate values in order to reduce the number of calculations. Let us use the undirected tree in Figure 6 (a) to introduce this inference algorithm. The probability distribution for this tree can be factorized as

$$P(X) = \frac{1}{Z}\psi(x_1, x_2)\psi(x_1, x_3)\psi(x_2, x_4)\psi(x_2, x_5)\psi(x_2, x_6).$$

Assuming discrete random variables, we calculate a marginal probability by summing over the remaining variables:

$$P(x_1) = \sum_{x_2,x_3,x_4,x_5,x_6} \frac{1}{Z}\psi(x_1,x_2)\psi(x_1,x_3)\psi(x_2,x_4)\psi(x_2,x_5)\psi(x_2,x_6).$$

When each variable has r values, the computation cost will scale to r^6. If the number of random variables and the number of states are huge, the computation cost will explode exponentially. However, using the distribution law of multiplication and addition, we have the following:

$$\begin{aligned}
P(x_1) &= \frac{1}{Z}\sum_{x_2}\psi(x_1,x_2)\underbrace{\sum_{x_3}\psi(x_1,x_3)}_{m_3(x_1)}\underbrace{\sum_{x_4}\psi(x_2,x_4)}_{m_4(x_2)}\underbrace{\sum_{x_5}\psi(x_2,x_5)}_{m_5(x_2)}\underbrace{\sum_{x_6}\psi(x_2,x_6)}_{m_6(x_2)} \\
&= \frac{1}{Z}\sum_{x_2}\psi(x_1,x_2)m_3(x_1)m_4(x_2)m_5(x_2)m_6(x_2) \\
&= \frac{1}{Z}m_3(x_1)\underbrace{\sum_{x_2}\psi(x_1,x_2)m_4(x_2)m_5(x_2)m_6(x_2)}_{m_2(x_1)} \\
&= \frac{1}{Z}m_2(x_1)m_3(x_1), \\
Z &= \sum_{x_1} P(x_1).
\end{aligned}$$

The intermediate factors $m(x)$ in the equation are called messages passing through the edges. Some of those messages will be used when we calculate other marginal probabilities. For example, when we calculate $P(x_3)$, we will use $m_2(x_1)$ again. Thus, by calculating the messages each node sending to its neighbors and reusing those calculated messages for new

round of messages, we can obtain marginals efficiently. This message passing algorithm is called belief propagation.

The previous belief propagation algorithm is also called sum-product algorithm, since summation and product are used in calculating marginals. If we want to compute maximum a posteriori (MAP) probabilities, the message passing algorithm will be called max-product algorithm. For the graph in Figure 6 (a), to obtain the MAP of $P(x_1)$, we compute:

$$
\begin{aligned}
g(x_1) \quad &\doteq \max_{x_2,x_3,x_4,x_5,x_6} P(x_1,x_2,x_3,x_4,x_5,x_6) \\
&= \frac{1}{Z}\max_{x_2}\psi(x_1,x_2)\underbrace{\max_{x_3}\psi(x_1,x_3)}_{m_3(x_1)}\underbrace{\max_{x_4}\psi(x_2,x_4)}_{m_4(x_2)}\underbrace{\max_{x_5}\psi(x_2,x_5)}_{m_5(x_2)}\underbrace{\max_{x_6}\psi(x_2,x_6)}_{m_6(x_2)} \\
&= \frac{1}{Z}\max_{x_2}\psi(x_1,x_2)m_3(x_1)m_4(x_2)m_5(x_2)m_6(x_2) \\
&= \frac{1}{Z}m_3(x_1)\underbrace{\max_{x_2}\psi(x_1,x_2)m_4(x_2)m_5(x_2)m_6(x_2)}_{m_2(x_1)} \\
&= \frac{1}{Z}m_2(x_1)m_3(x_1), \\
Z \quad &= \max_{x_1} g(x_1).
\end{aligned}
$$

1.3.1.1 Loopy Belief Propagation and Exact Inference

However, for graphs with loops, belief propagation will not give exact inference, since the message sent from a node will go back to itself. As in Figure 6 (b),the message sent from node 2 to node 1 will flow back into node 2 after it goes through the loop $2-1-3-2$. Nevertheless, belief propagation algorithm can still be applied on this graph and messages

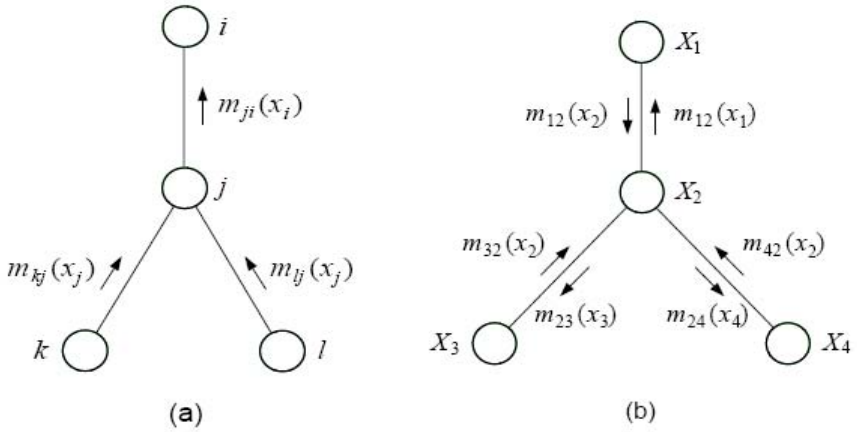


Figure 7. Illustration of Local Message Updating Rule

propagate "shortsightedly". In such cases, belief propagation is an approximation algorithm which is called loopy belief propagation. The general updating rules for messages are as follows:

$$m_{ij}(x_j) = \sum_{x_i} \psi_{ij}(x_i, x_j) \prod_{k \in \Gamma_i \backslash j} m_{ki}(x_i), \forall (i,j) \in \mathbb{E},$$

where $\mathbb{E}$ is the set of edges, Γ_i is the set of neighbors of i, $m_{ij}(x_j)$ is the message on edge (i,j), and $\psi_{ij}(x_i, x_j)$ is the potential function between x_i and x_j. The belief on each node, which is an approximation for marginal probability is computed as:

$$b_i(x_i) = \alpha \prod_{k \in \Gamma_i} m_{ki}(x_i),$$

where α is a normalization constant.

For instance, in Figure 7, the potential function between x_1, x_2 is $\psi_{12}(x_1, x_2)$. Thus, the message sent from node 2 to node 1 is $m_{21}(x_1) = \sum_{x_2} \psi_{12}(x_1, x_2) m_{32}(x_2) m_{42}(x_2)$ and the belief at node 2 is $b(x_2) = \alpha m_{12}(x_2) m_{32}(x_2) m_{42}(x_2)$.

1.3.1.2 Sufficient Convergence Condition for Sum Product Algorithm

Since in graphs with cycles, information can flow many times around the graph, loopy belief propagation will not always converge. Here, we will briefly discuss the convergence condition for loopy belief propagation.

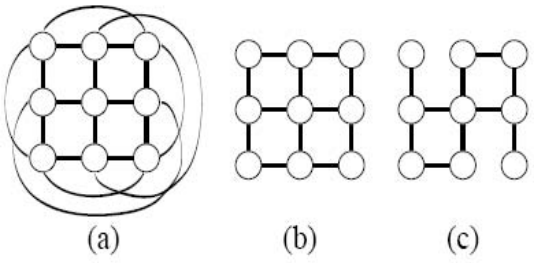
For convenience, we confine our discussion to graphical models with at most pairwise potential functions, so that the distribution factors as follows:

$$p(X) = \prod_{(s,t)\in E} \psi_{st}(x_s, x_t) \prod_s \psi_s(x_s).$$

For tree structured graphs, belief propagation will converge in a finite number of iterations (at most the length of the diameter of the graph) to the correct marginals. For an arbitrary graph, convergence is not guaranteed. The algorithm may converge to one fixed point, several fixed points or oscillate. Though convergence is not guaranteed for loopy belief propagation algorithm, when it converges, it usually gives good approximations to the exact marginals (34).

Many people presented sufficient conditions for loopy belief propagation algorithm based on different methodology. The sufficient condition given by (35)(17) is as follows:

$$\max_{(s,t)\in E} \sum_{u\in\Gamma_t\setminus s} \frac{d(\psi_{ut})^2 - 1}{d(\psi_{ut})^2 + 1} < 1,$$



Method	(a)	(b)	(c)
Simon's condition	.62	.62	.62
Heskes' condition	.55	.58	.65
This work	.67	.79	.88
Empirical	.67	.79	.88

η_{crit}

Figure 8. Comparison of Various Uniqueness Bounds in (17).

where ψ_{ut} are potential functions between node u and node t, and $d(\cdot)$ is dynamic range measure defined as follows: $d(\psi_{ut}) = \sup_{a,b,c,d} \sqrt{\psi(a,b)/\psi(c,d)}$. From the previous condition, we can easily see that sum product algorithm on simple loops will always converge, since we always have $\max_{(s,t)\in E} \frac{d(\psi_{ut})^2-1}{d(\psi_{ut})^2+1} < 1$.

(35)(17) proved their sufficient condition on binary belief networks shown in Figure 8(a-c),whose potentials are parameterized by a scaler $\eta > 0.5$ namely $\psi = [\eta \quad 1-\eta; 1-\eta \quad \eta]$, so that $d(\psi)^2 = \frac{\eta}{1-\eta}$. Their convergence condition almost gives the empirical values of parameters of potential functions for those graphical models, which is shown in the table of Figure 8 for η.

Sufficient condition for loopy belief propagation is our focus. We have worked on

$$\max_{(t,s)\in E} e_{ts}$$

as a different error measure from (35)(17)'s dynamic range error measure in order to obtain tighter sufficient condition.

(20) and (36) have derived an analytical relationship between beliefs and correct marginals for graphical models with single loops. They correspondingly proposed belief revision algorithm (37) in networks with a single loop, which achieves better exactness.

1.3.2 Tree-reweighted Message Passing

(38) presents a new message passing algorithm called tree-reweighted message passing to efficiently compute optimal values of upper bounds on log partition functions, as well as associated pseudo-marginals. The upper bounds can be used to derive a concave lower bound on log likelihood. The message passing rule is as follows:

$$M_{ts}(x_s) = \kappa \max_{x'_t \in \chi_t} \{\exp\,[\frac{\theta_{st}(x_s, x'_t)}{\rho_{st}} + \theta_t(x'_t)] \frac{\prod_{v \in \Gamma_t \setminus s} [M_{vt}(x'_t)]^{\rho_{vt}}}{[M_{st}(x'_t)]^{(1-\rho_{ts})}}\},$$

where $\theta_{st}(x_s, x_t)$ and $\theta_t(x_t)$ are log partition functions, $\rho_{st} \in [0, 1]$ re-weights messages and re-scales partition function on edge (s, t), . When $\rho_{st} = 1$ for all edges, it recovers standard message passing (belief propagation).

1.3.3 Bethe Free Energy and Generalized belief propagation

Let us first introduce the Bethe free energy here:

$$\begin{aligned} F_\beta(b_{ij}, b_i) = &\sum_{ij} \sum_{x_i, x_j} b_{ij}(x_i, x_j)[\ln b_{ij}(x_i, x_j) - \ln \psi_{ij}(x_i, x_j)] \\ &- \sum_i (q_i - 1) \sum_{x_i} b_i(x_i)[\ln b_i(x_i) - \ln \psi_i(x_i)], \end{aligned}$$

where b_{ij} and b_i are beliefs, ψ_{ij} and ψ_i are potential functions, q_i is the number of neighbors of node i. (39)and (40) have shown that stable fixed point of loopy belief propagation is the local

minima of Bethe free energy. However, for some graphical models, loopy belief propagation algorithm cannot reach a stable fixed point. Thus, more accurate free energy approximation algorithms have been proposed such as generalized belief propagation (40) based on Kikuchi free energy and unified propagation and scaling (41).

Kikuchi free energy gives better free energy approximation to Gibbs free energy. Kikuchi free energy is defined as follows:

$$F_K = \sum_{r \in R} c_r (\sum_{x_r} b_r(x_r)(-\ln \psi_r(x_r)) + \sum_{x_r} b_r(x_r) \ln b_r(x_r)),$$

where r is a region on the graph, ψ_r is the product of potential functions in region r, b_r is the belief on region r, and $c_r = 1 - \sum_{s \in super(r)} c_s$. $super(r)$ is the set of all super-regions of r.

Based on Kikuchi free energy, (40) presents general belief propagation algorithm:

$$\begin{aligned} m_{rs} &= \alpha [\sum_{x_{r \setminus s}} \psi_{r \setminus s}(x_{r \setminus s}) \prod_{m_{r''s''} \in M(r) \setminus M(s)} m_{r''s''}] / \prod_{m_{r's'} \in M(r,s)} m_{r's'}, \\ b_r &= \alpha \psi_r(x_r) \prod_{m_{r's'} \in M(r)} m_{r's'}, \end{aligned}$$

where $m_{rs}(x_s)$ are the messages between region r and its direct sub-region s, $M(r)$ is the set of messages in region r, $M(r,s)$ is the set of messages that start in sub-regions of r and also belong to $M(s)$.

1.3.4 Norm-Product Belief Propagation

Recently, (42) presents a unified message passing architecture which generalizes sum-product algorithm, max-product algorithm, and tree-reweighted sum-product and max-product algo-

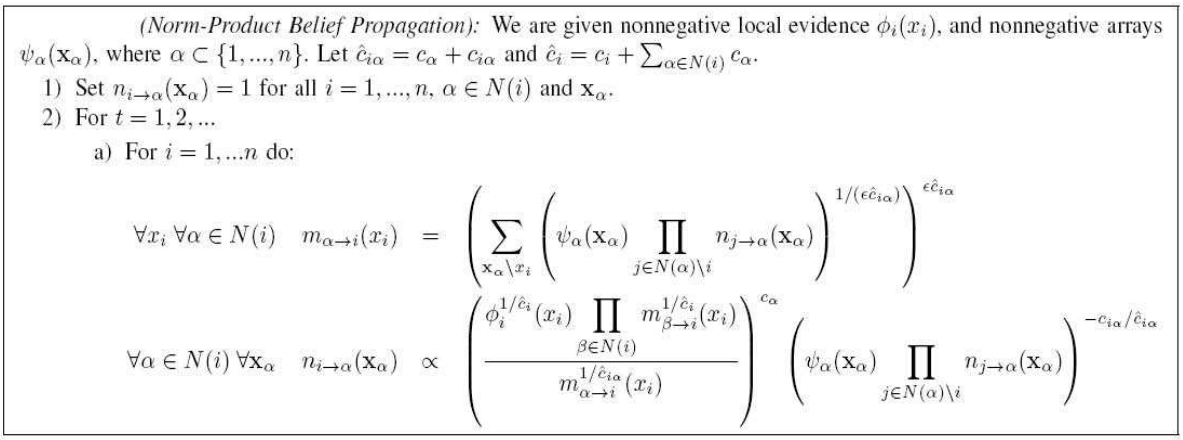


Figure 9. Norm-Product belief propagation in (24).

rithms. They present a set of convergent algorithms called norm-product belief propagation based on convex-free energy and linear programming. The norm product belief propagation is shown in Figure 9.

1.4 Conclusion

Graphical models are a useful tool to visualize the relationship between random variables. Utilizing the structure of graphs, efficient inference algorithms are proposed. For tree structured graph, all inference algorithms will give exact marginal probabilities, whereas for graphs with loops, some algorithms fail to obtain exact values. However, various approximation algorithms are proposed for loopy graphs and provide good results. Loopy belief propagation is such an approximation algorithm which empirically demonstrated good performance.

Since loopy belief propagation algorithm is an iterative method, convergence should be guaranteed for it to work properly. However, for some graphs,it is not the case. Some papers have presents sufficient conditions for the convergence of belief propagation algorithm. Though loopy belief propagation will obtain convergent beliefs, the exactness of the beliefs compared with the true marginal probabilities is not ensured. (40) has proved that stable fixed point of belief propagation corresponds to the local minimal value of Bethe free energy. In order to guarantee the algorithm to search toward minimal points of the free energy function, some gradient descent based approximation algorithms are proposed.

CHAPTER 2

MESSAGE ERROR ANALYSIS OF LOOPY BELIEF PROPAGATION FOR THE SUM-PRODUCT ALGORITHM

2.1 Introduction

Belief propagation is known to perform extremely well in many practical statistical inference and learning problems using graphical models, even in the presence of multiple loops. The use of the belief propagation algorithm on graphical models with loops is referred to as Loopy Belief Propagation (LBP). Various sufficient conditions for convergence of LBP have been presented; however, general necessary conditions for its convergence to a unique fixed point remain unknown. Because the approximation of beliefs to true marginal probabilities has been shown to relate to the convergence of LBP, various methods have been explored whose aim is to obtain distance bounds on beliefs when LBP fails to converge. People presented their performance analysis of LBP with respect to convergence and accuracy in literatures (40)(36)(22)(17)(25)(18)(27)(31).

In this chapter, we derive tight error bounds on LBP and use these bounds to study the dynamics—error, convergence, accuracy, and scheduling—of the sum-product algorithm. Specifically, in Section 2.2 and Section 2.3, we rely on the contractive mapping property of message errors to present novel uniform and non-uniform distance bounds between multiple fixed-point solutions. Several graphical networks are investigated and used to demonstrate that

the proposed distance bounds are tighter than existing bounds. We subsequently use these bounds to derive uniform and non-uniform sufficient conditions for convergence of the sum-product algorithm. Moreover, in Section 2.4, we analyze the relation between convergence and sparsity of graphs, and extend the convergence perspective of walk-summability from Gaussian graphical models to general graphical models. In Section 2.5, we present bounds on the distance between beliefs and true marginals by applying SAW trees and show that the proposed bounds can be used to improve existing bounds. Furthermore, we explore the use of the upper-bound on message errors as a criterion to rank the priority of message passing for scheduling in asynchronous LBP. We then present a case study of LBP by studying its dynamics on completely uniform graphs and analyzing its true fixed points and message-error functions in Section 2.6. Finally, we discuss the extension of the proposed message error analysis to the max-product algorithm in Section 2.7. We conclude the paper in Section 2.8.

2.2 Message-Error Propagation for the Sum-Product Algorithm

Belief propagation originated from exact inference on tree structured graphical models, though for graphs with loops it shows remarkable performance of approximate inference. BP is synonymously called sum-product algorithm for marginalization of global distribution or max-product algorithm to compute Maximum-A-Posteriori (MAP). In this paper, we will mainly talk about sum-product algorithm for graphs with loops.

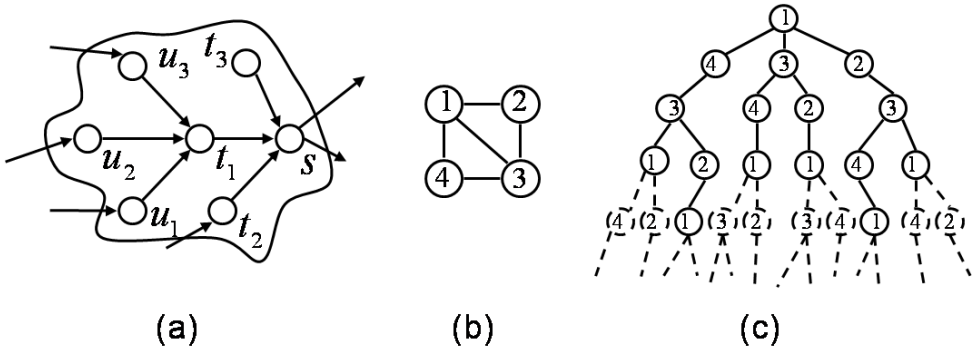


Figure 10. Graphical models: (a) message passing in a portion of a belief network; (b) a simple graph; and (c) Bethe tree (all nodes and edges) and Self-Avoiding Walk tree (black solid only) of (b).

2.2.1 Loopy Belief Propagation Updates

Let us consider a general graphical model $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ whose distribution factors as follows:

$$p(X) = \frac{1}{Z} \prod_{(s,t)\in\mathbb{E}} \psi_{st}(x_s, x_t) \prod_{s\in\mathbb{V}} \psi_s(x_s), \tag{2.1}$$

where Z is a normalization factor, $\psi_{st}(x_s, x_t)$ is the pairwise potential function between random variables x_s and x_t, and $\psi_s(x_s)$ is the single node potential function on x_s. (s, t) denotes an undirected edge, $\mathbb{V}$ is the set of nodes, and $\mathbb{E}$ is the set of edges. We assume that all the potential functions are positive.

Figure 10(a) illustrates the message passing mechanism used in BP. The updating rule of the sum-product algorithm for the message sent by node t to its neighbor node s at iteration i is:

$$m_{ts}^{i}(x_s) \propto \int \psi_{ts}(x_t, x_s)\psi_t(x_t) \prod_{u \in \Gamma_t \backslash s} m_{ut}^{i-1}(x_t) dx_t, \tag{2.2}$$

where Γ_t is the set of neighbors of node t. The belief, or pseudo-marginal probability of x_t, on node t at iteration i, is:

$$B_t^i(x_t) \propto \psi_t(x_t) \prod_{u \in \Gamma_t} m_{ut}^{i}(x_t). \tag{2.3}$$

A stable fixed point has been reached if $m_{ts}^{i}(x_s) = m_{ts}^{i+1}(x_s)$, $\forall s \in \mathbb{V}$. The pairwise belief of random variables x_s, x_t at iteration i is defined as:

$$B_{ts}^i(x_t, x_s) \propto \psi_{ts}(x_t, x_s)\psi_t(x_t)\psi_s(x_s) \prod_{u \in \Gamma_t \backslash s} m_{ut}^{i}(x_t) \prod_{p \in \Gamma_s \backslash t} m_{ps}^{i}(x_s). \tag{2.4}$$

The computation tree first introduced in (43) is always applied in the analysis of LBP. Bethe tree and SAW tree are two types of computation trees used in (25), which will also be used in the rest of the paper. Both Bethe tree and SAW tree are tree-structured unwrappings of a graph $\mathbb{G}$ from some node v. The Bethe tree, denoted as $T_B(\mathbb{G}, v, n)$, contains all paths of length n from v that do not backtrack, while the SAW tree, denoted as $T_{SAW}(\mathbb{G}, v, n)$, contains all paths of length $n \leq |\mathbb{V}| + 1$ that do not backtrack and have all nodes on the path unique. The belief on node v at iteration n in synchronous LBP is equivalent to the exact marginal of the root v in the n-level Bethe tree.

Figure 10(c) illustrates the Bethe tree and the SAW tree for the graphical model in Figure 10(b). For synchronous BP, each iteration of (Equation 2.2), (Equation 2.3) and (Equation 2.4) corresponds to a level in the Bethe tree.

2.2.2 Message-Error Measures

Various approaches have been presented to derive convergence conditions for the sum-product algorithm, including analyzing contraction property of message errors on belief networks. Define *message error* as a multiplicative function $e^i_{ts}(x_s)$ that perturbs the fixed-point message $m_{ts}(x_s)$. The perturbed message at iteration i is hence

$$\hat{m}^i_{ts}(x_s) = m_{ts}(x_s)e^i_{ts}(x_s).$$

Dealing with normalized messages, we define *fixed-point incoming message products* as

$$M_{ts}(x_t) \propto \psi_t(x_t) \prod_{u \in \Gamma_t \backslash s} m_{ut}(x_t),$$

and *perturbed incoming message products* as

$$M^i_{ts}(x_t) \propto \psi_t(x_t) \prod_{u \in \Gamma_t \backslash s} m^i_{ut}(x_t),$$

and *incoming error products* as

$$E^i_{ts}(x_t) = \prod_{u \in \Gamma_t \backslash s} e^i_{ut}(x_t).$$

We have

$$M_{ts}^{i}(x_t) \propto M_{ts}(x_t)E_{ts}^{i}(x_t).$$

Thus, the *outgoing message error* from node t to node s at iteration $i+1$ is:

$$e_{ts}^{i+1}(x_s) = \frac{\hat{m}_{ts}^{i+1}(x_s)}{m_{ts}(x_s)} = \frac{\int \psi_{ts}(x_t, x_s)M_{ts}(x_t)E_{ts}^{i}(x_t)dx_t}{\int \psi_{ts}(x_t, x_s)M_{ts}(x_t)E_{ts}^{i}(x_t)dx_t dx_s} \times \frac{\int \psi_{ts}(x_t, x_s)M_{ts}(x_t)dx_t dx_s}{\int \psi_{ts}(x_t, x_s)M_{ts}(x_t)dx_t}.$$

In the following, we will introduce two measures on message errors.

2.2.2.1 Dynamic-Range Measure

The *dynamic-range measure* of error introduced by (17) is defined as:

$$d(e_{ts}^{i}) = \max_{a,b} \sqrt{\frac{e_{ts}^{i}(a)}{e_{ts}^{i}(b)}}. \tag{2.5}$$

We have $d(e_{ts}^{i}) \to 1$ when $e_{ts}^{i}(x) \to 1$. In (17) [Th.8] it was shown that when $d(\psi_{ts}) = \max_{a,b,c,d} \sqrt{\frac{\psi_{ts}(a,b)}{\psi_{ts}(c,d)}}$ is finite, the dynamic-range measure satisfies the following contraction:

$$d(e_{ts}^{i+1}) \leq \frac{d(\psi_{ts})^2 d(E_{ts}^{i}) + 1}{d(\psi_{ts})^2 + d(E_{ts}^{i})}, \tag{2.6}$$

in other words, based on the dynamic-range measure, the outgoing message error is bounded by a non-linear function of the potential function and the incoming error product.

2.2.2.2 Maximum-Error Measure

To study the dynamics of message error propagation, dealing directly with errors is more interesting than dealing with dynamic range. We thus introduce the following *maximum multiplicative error* function as an error measure:

$$\max_{x_s} e_{ts}^{i+1}(x_s) = \max_{x_s} \frac{\int \psi_{ts}(x_t, x_s) M_{ts}(x_t) E_{ts}^{i}(x_t) dx_t}{\int \psi_{t\star}(x_t) M_{ts}(x_t) E_{ts}^{i}(x_t) dx_t} \times \frac{\int \psi_{t\star}(x_t) M_{ts}(x_t) dx_t}{\int \psi_{ts}(x_t, x_s) M_{ts}(x_t) dx_t}, \tag{2.7}$$

where $\psi_{t\star}(x_t) = \int \psi_{ts}(x_t, x_s) dx_s$. It is immediate that the maximum-error measure approaches one when multiplicative errors vanish. We will show later that this error measure satisfies the following contraction:

$$\max_{x_s} e_{ts}^{i+1}(x_s) \leq \left(\frac{d(\psi_{ts}) d(\psi_{t\star}) d(E_{ts}^{i}) + 1}{d(\psi_{ts}) d(\psi_{t\star}) + d(E_{ts}^{i})} \right)^2. \tag{2.8}$$

Dynamic-range measure and maximum-error measure are equivalent when the maximum and minimum of an error function are reciprocal. By comparison, maximum-error measure gives an absolute error, while dynamic-range measure gives a relative error which is invariant to scaling. We will show in the following of the paper that maximum-error measure should be used, when we are interested in absolute errors. Furthermore, both defined in dynamic-range measure, $d(\psi_{ts})$ and $d(\psi_{t\star})$ correspond to two types of matrix norms on ψ_{ts}. $d(\psi_{t\star})$ in the RHS of Inequality in (Equation 2.8) characterizes the effect of normalization factor on $\max_{x_s} e_{ts}^{i+1}(x_s)$. We will discuss the influence of $d(\psi_{t\star})$ on error bounds in Section 2.2.4.

2.2.3 Strength of Potential Functions

(16), (17) and (18) have defined measures of strength of potential functions respectively, which help to obtain milder convergence conditions than those only related with topology of graphical models. In the following, we will show the relationship between beliefs and strength of pairwise potential functions.

2.2.3.1 Strength of Potential functions in (16)

(16) defined $\sigma_{t,s}$ as the strength of a pairwise potential function $\psi_{ts}(x_t, x_s)$ meeting the following equation:

$$\frac{1}{1-\sigma_{t,s}} = \max_{x_t, x_s, \hat{x}_t, \hat{x}_s} \frac{\psi_{ts}(x_t, x_s)\psi_{ts}(\hat{x}_t, \hat{x}_s)}{\psi_{ts}(x_t, \hat{x}_s)\psi_{ts}(\hat{x}_t, x_s)}.$$

This strength is related with the correlation of LBP marginals as follows:

$$\frac{B_{ts}(x_t, \hat{x}_s)}{B_t(x_t)B_s(\hat{x}_s)} \leq \frac{1}{1-\sigma_{t,s}},$$

which was then utilized to give a milder convergence condition than the one only depending on graph topology.

2.2.3.2 Strength of Potential functions in (17)

(17) proposed the dynamic-range measure $d(\psi_{ts})$ as the strength of potential functions $\psi_{ts}(x_t, x_s)$. Let us restate the definition of the strength of potential functions and its relationship with message errors in Section 2.2.1 as follows:

$$d(\psi_{ts}) = \max_{x_t, x_s, \hat{x}_t, \hat{x}_s} \sqrt{\frac{\psi_{ts}(x_t, x_s)}{\psi_{ts}(\hat{x}_t, \hat{x}_s)}},$$

$$d(e_{ts}) \leq \frac{d(\psi_{ts})^2 d(E_{ts}) + 1}{d(\psi_{ts})^2 + d(E_{ts})}.$$

2.2.3.3 Strength of Potential functions in (18)

(18) mentioned a measure of the strength of potential function $\psi_{ts}(x_t, x_s)$, which is defined as:

$$N(\psi_{ts}) = \max_{x_t \neq \hat{x}_t, x_s \neq \hat{x}_s} \frac{\sqrt{\frac{\psi_{ts}(x_t, x_s)\psi_{ts}(\hat{x}_t, \hat{x}_s)}{\psi_{ts}(\hat{x}_t, x_s)\psi_{ts}(x_t, \hat{x}_s)}} - 1}{\sqrt{\frac{\psi_{ts}(x_t, x_s)\psi_{ts}(\hat{x}_t, \hat{x}_s)}{\psi_{ts}(\hat{x}_t, x_s)\psi_{ts}(x_t, \hat{x}_s)}} + 1} = \frac{1 - \sqrt{1 - \sigma_{t,s}}}{1 + \sqrt{1 - \sigma_{t,s}}}. \tag{2.9}$$

They defined log dynamic range measure as metric of errors. Let λ_{ts} be the log message reparameterization of message m_{ts}. That is,

$$\lambda_{ts}(x_s) = \log m_{ts}(x_s).$$

Denote $\Delta\lambda$ as the difference of log messages. Thus, we have

$$\Delta\lambda_{ts}(x_s) = \log \hat{m}_{ts}(x_s) - \log m_{ts}(x_s) = \log e_{ts}(x_s).$$

By the quotient norm and Equation (41) in (18), we have the following metric of error

$$\|\overline{\Delta\lambda_{ts}}\| = \frac{1}{2}\sup_{x_s,x'_s}|\Delta\lambda_{ts}(x_s) - \Delta\lambda_{ts}(x'_s)| = \log d(e_{ts}). \tag{2.10}$$

Using the quotient mapping approach of parallel LBP update in (18), we will find the relationship between the strength of potential functions in (Equation 2.9) and the metric of message errors in (Equation 2.10) in the following.

Because $\|\overline{\Delta\lambda_{ts}}\| \leq \sum_{u\in\Gamma_t\setminus s}\|\overline{\frac{\partial\lambda_{ts}}{\partial\lambda_{ut}}}\|\|\overline{\Delta\lambda_{ut}}\|$ and $\|\overline{\frac{\partial\lambda_{ts}}{\partial\lambda_{ut}}}\| \leq N(\psi_{ts})$ by Equation (36-45) in (18), we have

$$\log d(e_{ts}) \leq N(\psi_{ts})\sum_{u\in\Gamma_t\setminus s}\log d(e_{ut}) \leq N(\psi_{ts})\log d(E_{ts}),$$

$$or, \quad d(e_{ts}) \leq d(E_{ts})^{N(\psi_{ts})}.$$

We can observe that the smaller $N(\psi_{ts})$ is, the smaller is $d(e_{ts})$; therefore, the faster is the contraction of errors. The previous inequality reveals another result on contractive property of message errors beside the one in (Equation 2.6).

In the following, we use the maximum-error measure in (Equation 2.7) to explore upper and lower bounds on message errors, and derive upper bounds on the distances between beliefs.

2.2.4 Upper- and Lower-Bounds on Message Errors

Let us first introduce a lemma that will be used to prove our following theorem.

Lemma 1. *For f_1, f_2, g_1, g_2 all positive,*

$$\frac{f_1+f_2}{g_1+g_2} \leq \max[\frac{f_1}{g_1}, \frac{f_2}{g_2}], \quad \frac{f_1+f_2}{g_1+g_2} \geq \min[\frac{f_1}{g_1}, \frac{f_2}{g_2}].$$

Proof. The left inequality is proved in (17). For the right inequality assume without loss of generality that $f_1/g_1 \leq f_2/g_2$ so that $f_1 g_2 \leq f_2 g_1 \Rightarrow f_1 g_2 + f_1 g_1 \leq f_2 g_1 + f_1 g_1 \Rightarrow \frac{f_1}{g_1} \leq \frac{f_1+f_2}{g_1+g_2}$. □

In the following, we shall omit reference to the iteration number of the messages and errors for simplicity and clarity of the presentation.

Theorem 2. *Multiplicative outgoing errors are bounded as:*

$$\left(\frac{d(\psi_{ts})d(\psi_{t\star})+d(E_{ts})}{d(\psi_{ts})d(\psi_{t\star})d(E_{ts})+1}\right)^2 \leq \min_{x_s} e_{ts}(x_s) \leq e_{ts}(x_s) \leq \max_{x_s} e_{ts}(x_s) \leq \left(\frac{d(\psi_{ts})d(\psi_{t\star})d(E_{ts})+1}{d(\psi_{ts})d(\psi_{t\star})+d(E_{ts})}\right)^2.$$

Proof. Similar to the analysis in (17, Lemma 26), for (Equation 2.7), $\max e_{ts}(x_s)$ reaches its maximum when $\psi_{ts}(x_t, x_s) = 1 + (d(\psi_{ts})^2 - 1)\chi_\psi(x_t)$, $\psi_{t\star}(x_t) = 1 + (d(\psi_{t\star})^2 - 1)\chi_\star(x_t)$ and $E_{ts}(x_t) = 1 + (d(E_{ts})^2 - 1)\chi_E(x_t)$, where χ_ψ, $\chi_\star$ and χ_E are indicator functions. Define the quantities:

$$M_A = \int M_{ts}(x_t)\chi_\psi(x_t)dx_t, \quad M_B = \int M_{ts}(x_t)\chi_\star(x_t)dx_t, \quad M_E = \int M_{ts}(x_t)\chi_E(x_t)dx_t,$$
$$M_{AE} = \int M_{ts}(x_t)\chi_\psi(x_t)\chi_E(x_t)dx_t, \quad M_{BE} = \int M_{ts}(x_t)\chi_\star(x_t)\chi_E(x_t)dx_t,$$
$$\alpha_1 = d(\psi_{ts})^2 - 1, \quad \alpha_2 = d(\psi_{t\star})^2 - 1, \quad \beta = d(E_{ts})^2 - 1.$$

The maximum multiplicative error is upper-bounded by $\max_{x_s} e_{ts}(x_s) \leq \Delta_1$ where

$$\Delta_1 = \max_M \frac{1+\alpha_1 M_A + \beta M_E + \alpha_1\beta M_{AE}}{1+\alpha_2 M_B + \beta M_E + \alpha_2\beta M_{BE}} \frac{1+\alpha_2 M_B}{1+\alpha_1 M_A}.$$

The maximum is obtained when $M_{AE} = M_A = M_E = 1 - M_B$ and $M_{BE} = 0$, which gives

$$\Delta_1 = \max_{M_E} \frac{1+(\alpha_1+\beta+\alpha_1\beta)M_E}{1+\alpha_2+(\beta-\alpha_2)M_E} \frac{1+\alpha_2-\alpha_2 M_E}{1+\alpha_1 M_E}.$$

Taking the derivative wrt M_E and setting it to zero, we obtain

$$\max_{x_s} e_{ts}(x_s) \leq \Delta_1 = \left(\frac{d(\psi_{ts})d(\psi_{t\star})d(E_{ts})+1}{d(\psi_{ts})d(\psi_{t\star})+d(E_{ts})}\right)^2. \tag{2.11}$$

Similarly to what we have done so far, by our Lemma 1, we can lower-bound $\min_{x_s} e_{ts}(x_s)$ with respect to $\psi_{ts}(x_t, x_s)$, $\psi_{t\star}(x_t)$ and $E_{ts}(x_t)$, to obtain

$$\min_{x_s} e_{ts}(x_s) \geq \left(\frac{d(\psi_{ts})d(\psi_{t\star})+d(E_{ts})}{d(\psi_{ts})d(\psi_{t\star})d(E_{ts})+1}\right)^2 = \frac{1}{\Delta_1}. \tag{2.12}$$

□

Theorem 3. *The upper bound on the multiplicative error provided in Theorem 2 is tighter than the following upper bound from (17, Th.2 and Th.8):*

$$\max_{x_s} e_{ts}(x_s) \leq d(e_{ts})^2 \leq \left(\frac{d(\psi_{ts})^2 d(E_{ts})+1}{d(\psi_{ts})^2+d(E_{ts})}\right)^2 = \Delta_2. \tag{2.13}$$

Proof. Because Δ_1 in (Equation 2.11) is increasing in $d(\psi_{t\star})$ we conclude that (Equation 2.11) implies (Equation 2.13), i.e., $\Delta_1 \leq \Delta_2$, because

$$d(\psi_{t\star}) = \max_{a,b} \sqrt{\frac{\psi_{t\star}(a)}{\psi_{t\star}(b)}} = \max_{a,b} \sqrt{\frac{\int \psi_{ts}(a, x_s) dx_s}{\int \psi_{ts}(b, x_s) dx_s}}$$
$$\leq \max_{a,b} \sqrt{\max_{c,d} \frac{\psi_{ts}(a, c)}{\psi_{ts}(b, d)}} = \max_{a,b,c,d} \sqrt{\frac{\psi_{ts}(a, c)}{\psi_{ts}(b, d)}} = d(\psi_{ts}).$$

□

We can see how $d(\psi_{t\star})$ tightens the upper-bound by analyzing the log-distance between Δ_1 and Δ_2. Let $d(\psi_{t\star}) = Kd(\psi_{ts})$, where $1/d(\psi_{ts}) \leq K \leq 1$. Therefore, the log-distance between Δ_1 and Δ_2 is denoted as

$$D(K) = \log \Delta_1 - \log \Delta_2 = 2 \times \log \{ \frac{Kd(\psi_{ts})^2 d(E_{ts}) + 1}{Kd(\psi_{ts})^2 + d(E_{ts})} \times \frac{d(\psi_{ts})^2 + d(E_{ts})}{d(\psi_{ts})^2 d(E_{ts}) + 1} \}.$$

We can easily find that the first gradient $D^{(1)}(K) > 0$ when $d(E_{ts}) > 1$. Thus, the maximum log-distance between Δ_1 and Δ_2 is obtained at $K = 1/d(\psi_{ts})$. In other words, when $d(\psi_{t\star}) = 1$, our upper-bound Δ_1 is tighter than Δ_2 at farthest.

2.3 Distance Bounds on Beliefs

In the study of convergence, we are interested to know how beliefs will vary at each iteration, when LBP fails to converge. We will show that beliefs are bounded given the strength of potential functions and the structure of the graph. In the following, we will present our *uniform distance bound* and *non-uniform distance bound* on beliefs. Based on those bounds, we further

present *uniform convergence condition* and *non-uniform convergence condition* for synchronous LBP.

2.3.1 Uniform Distance Bound

Corollary 4. ***(Uniform Distance Bound)***

The log-distance bound of fixed points on belief at node s is

$$\sum_{t\in\Gamma_s}\log(\frac{d(\psi_{ts})d(\psi_{t\star})\varepsilon+1}{d(\psi_{ts})d(\psi_{t\star})+\varepsilon})^2,$$

where ε should satisfy

$$\log\varepsilon=\max_{(s,p)\in\mathbb{E}}\sum_{t\in\Gamma_s\backslash p}\log(\frac{d(\psi_{ts})d(\psi_{t\star})\varepsilon+1}{d(\psi_{ts})d(\psi_{t\star})+\varepsilon})^2.$$

Proof. Let $\Delta_{ut}(x)=(\frac{d(\psi_{ut})d(\psi_{u\star})x+1}{d(\psi_{ut})d(\psi_{u\star})+x})^2, x\geq 1, ut\in\mathbb{E}$. Therefore,

$$d(E^i_{ts})\leq\prod_{u\in\Gamma_t\backslash s}d(e^i_{ut})=\prod_{u\in\Gamma_t\backslash s}\frac{\max\sqrt{e^i_{ut}(x_t)}}{\min\sqrt{e^i_{ut}(x_t)}}\leq\varepsilon^i_{ts}=\prod_{u\in\Gamma_t\backslash s}\Delta_{ut}(d(E^{i-1}_{ut})).$$

Thus, we have

$$\max_{x_s}E^{i+1}_{sp}(x_s)\leq\prod_{t\in\Gamma_s\backslash p}\max_{x_s}e^{i+1}_{ts}(x_s)\leq\varepsilon^{i+1}_{sp}=\prod_{t\in\Gamma_s\backslash p}\Delta_{ts}(d(E^i_{ts}))$$

$$\leq\prod_{t\in\Gamma_s\backslash p}\Delta_{ts}(\varepsilon^i_{ts})\leq\prod_{t\in\Gamma_s\backslash p}\Delta_{ts}(\max_{t\in\Gamma_s\backslash p}\varepsilon^i_{ts})=\Delta_3(\max_{t\in\Gamma_s\backslash p}\varepsilon^i_{ts}).$$

The term ε_{sp}^{i+1} is an upper-bound on the incoming error product $E_{sp}^{i+1}(x_s)$ at iteration $i+1$, while $\max_{t\in\Gamma_s\backslash p}\varepsilon_{ts}^{i}$ is the maximum of the upper-bounds on the incoming error products $\{E_{ts}^{i}(x_t), t\in\Gamma_s\backslash p\}$ at iteration i. We hope to achieve that $\varepsilon_{sp}^{i+1} < \max_{t\in\Gamma_s\backslash p}\varepsilon_{ts}^{i}$. Denoting $\varepsilon = \max_{t\in\Gamma_s\backslash p}\varepsilon_{ts}^{i}$, let us introduce an *error bound-variation function*:

$$G_{sp}(\log\varepsilon) = \log\Delta_3(\varepsilon) - \log\varepsilon \geq \log\varepsilon_{sp}^{i+1} - \log\max_{t\in\Gamma_s\backslash p}\varepsilon_{ts}^{i}, \varepsilon\geq 1, \tag{2.14}$$

which describes variation of error bound after each iteration. When $G_{sp}(\log\varepsilon) = 0$, the log-distance bound $\log\varepsilon$ will reach a fixed point, which is the maximal distance between message products at various iterations. Because $G_{sp}^{(2)}(\log\varepsilon) < 0$ for $\log\varepsilon > 0$ and $G_{sp}^{(1)}(\infty) = -1/2$, $G_{sp}^{(1)}(\log\varepsilon)$ will decrease until it is equal to $-1/2$. Therefore, it only has one crossing point besides $\log\varepsilon = 0$ (zero crossing point). This nonzero crossing point is a stable fixed point of function $G_{sp}(\log\varepsilon)$. In other words, once $\log\varepsilon$ leaves the zero crossing point, it will stay at this stable crossing point, $\log\varepsilon^*$, which corresponds to the upper bound on error products.

Because the distance between fixed points of $B_s(x_s)$ is

$$\log E_s(x_s) = \log\prod_{t\in\Gamma_s} e_{ts}(x_s) \leq \log\prod_{t\in\Gamma_s}\Delta_{ts}(\varepsilon^*),$$

we can obtain the log-distance bound on $B_s(x_s)$ by taking the maximum ε^*. □

Adopting the upper-bound Δ_2 in (Equation 2.13), the error bound-variation function is:

$$G^I_{sp}(\log \varepsilon) = \log \prod_{t \in \Gamma_s \backslash p} (\frac{d(\psi_{ts})^2 \varepsilon + 1}{d(\psi_{ts})^2 + \varepsilon})^2 - \log \varepsilon. \tag{2.15}$$

Let us denote our error bound-variation function in (Equation 2.14) as $G^O_{sp}(\log \varepsilon)$. We can see that $G^O_{sp}(\log \varepsilon) < G^I_{sp}(\log \varepsilon)$. In other words, the uniform distance bound using our upper-bound Δ_1 is tighter than that using (17)'s upper-bound Δ_2, which is illustrated in Figure 11.

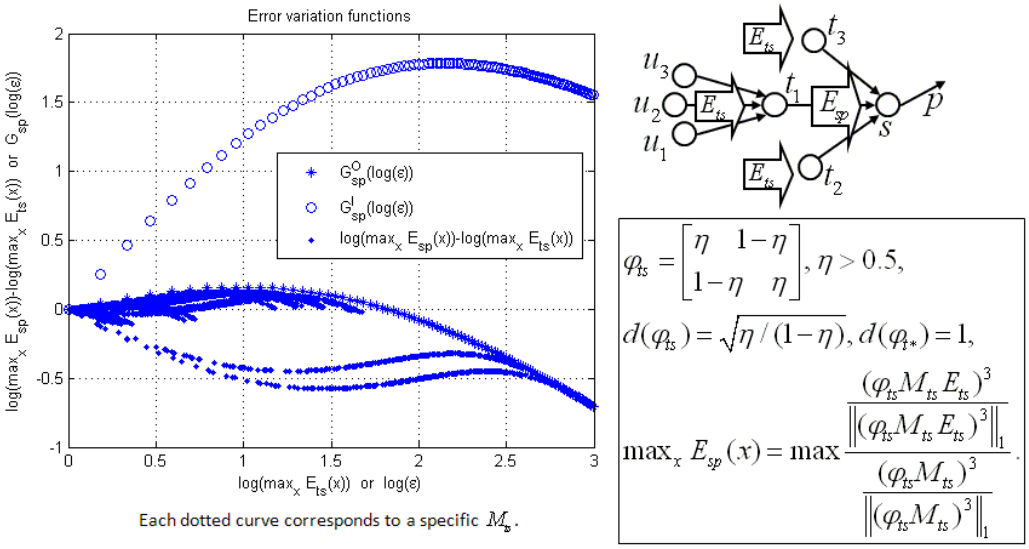


Figure 11. Error bound-variation functions versus true error-variation function for the local graph of node s. Potential functions on edges $(t_1, s), (t_2, s), (t_3, s)$ are the same. We also impose the same incoming error product E_{ts} on nodes t_1, t_2, t_3. The dotted curves depict the true error variation functions, $\{\log \max_x E_{sp}(x) - \log \max_x E_{ts}(x), t \in \Gamma_s \backslash p\}$, which are enveloped by our error bound-variation function $G^O_{sp}(\log \varepsilon)$.

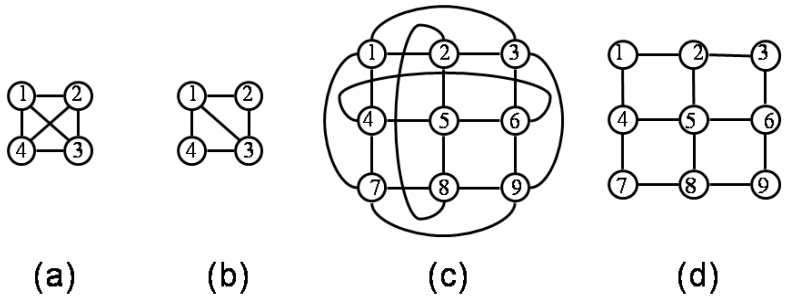


Figure 12. Four simple graphical models: (a) a four-node fully connected graph; (b) a partial graph that has one less edge than (a); (c) a nine-node fully connected graph ; and (d) a 3×3 grid that is a partial graph of (c).

When the error bound-variation function is always less than zero, the maximum of error bounds decreases after each iteration of LBP. In other words, LBP will converge. Therefore, our uniform distance bound in Corollary 4 will lead to a sufficient condition for convergence of LBP.

Theorem 5. ***(Uniform Convergence Condition)***

Based on maximum-error measure, the sufficient condition for the convergence of sum-product algorithm is

$$\max_{(s,p)\in\mathbb{E}} \sum_{t\in\Gamma_s\setminus p} \frac{d(\psi_{ts})d(\psi_{t\star})-1}{d(\psi_{ts})d(\psi_{t\star})+1} < \frac{1}{2}.$$

Proof. Let us revisit the *error bound-variation function* in (Equation 2.14):

$$G_{sp}(\log\varepsilon) = \log \prod_{t\in\Gamma_s\backslash p} (\frac{d(\psi_{ts})d(\psi_{t\star})\varepsilon + 1}{d(\psi_{ts})d(\psi_{t\star}) + \varepsilon})^2 - \log\varepsilon,$$

which describes the variation of the error bound after each iteration. To guarantee that LBP converges, it is sufficient to require $G_{sp}(\log\varepsilon) < 0, \forall \log\varepsilon > 0$. Let $z = \log\varepsilon$. The second derivative of $G_{sp}(z)$ is

$$G^{(2)}_{sp}(z) = 2 \times \sum_{t\in\Gamma_s\backslash p} \frac{d(\psi_{ts})d(\psi_{t\star})e^z((d(\psi_{ts})d(\psi_{t\star}))^2 - 1)(1 - e^{2z})}{(d(\psi_{ts})d(\psi_{t\star})e^z + 1)^2(d(\psi_{ts})d(\psi_{t\star}) + e^z)^2} \leq 0,$$

when $d(\psi_{ts})d(\psi_{t\star}) > 1$ and $z \geq 0$. When $z > 0$, $G_{sp}(z)$ is strictly concave.

The first derivation of $G_{sp}(z)$ is

$$G^{(1)}_{sp}(z) = 2 \times \sum_{t\in\Gamma_s\backslash p} \frac{e^z((d(\psi_{ts})d(\psi_{t\star}))^2 - 1)}{(d(\psi_{ts})d(\psi_{t\star})e^z + 1)(d(\psi_{ts})d(\psi_{t\star}) + e^z)} - 1.$$

Because $G_{sp}(z = 0) = 0$, if the first derivative $G^{(1)}_{sp}(z = 0) < 0$, we will have $G_{sp}(z > 0) < 0$. Therefore,

$$G^{(1)}_{sp}(0) = 2 \times \sum_{t\in\Gamma_s\backslash p} \frac{((d(\psi_{ts})d(\psi_{t\star}))^2 - 1)}{(d(\psi_{ts})d(\psi_{t\star}) + 1)(d(\psi_{ts})d(\psi_{t\star}) + 1)} - 1 < 0$$

$$\Rightarrow \sum_{t\in\Gamma_s\backslash p} \frac{d(\psi_{ts})d(\psi_{t\star}) - 1}{d(\psi_{ts})d(\psi_{t\star}) + 1} < \frac{1}{2}.$$

□

Lemma 6. *Our sufficient condition* $\sum_{t\in\Gamma_s\setminus p}\frac{d(\psi_{ts})d(\psi_{t\star})-1}{d(\psi_{ts})d(\psi_{t\star})+1} < \frac{1}{2}$ *is worse than the sufficient condition in (17), which is* $\sum_{t\in\Gamma_s\setminus p}\frac{d(\psi_{ts})^2-1}{d(\psi_{ts})^2+1} < 1$.

Proof. $2(\frac{d(\psi_{ts})d(\psi_{t\star})-1}{d(\psi_{ts})d(\psi_{t\star})+1}) > \frac{d(\psi_{ts})^2-1}{d(\psi_{ts})^2+1}$. □

This shows that *dynamic-range measure* is better than *maximum-error measure* with respect to the sensitivity of the measure to convergence. Nevertheless, as for the upper bound on a multiplicative message error $e_{ts}(x)$, *maximum-error measure* gives a tighter result, which is shown in Theorem 3.

Inspired by the sensitivity of dynamic-range measure to convergence, we present the following *improved uniform distance bound*, which first calculates the fixed-point values of error bounds in dynamic-range measure, and then computes the error bounds among beliefs in maximum-error measure.

Corollary 7. ***(Improved Uniform Distance Bound)***

The log-distance bound of fixed points on belief at node s is

$$\sum_{t\in\Gamma_s}\log(\frac{d(\psi_{ts})d(\psi_{t\star})\varepsilon+1}{d(\psi_{ts})d(\psi_{t\star})+\varepsilon})^2,$$

where ε *should satisfy*

$$\log\varepsilon = \max_{(s,p)\in\mathbb{E}}\sum_{t\in\Gamma_s\setminus p}\log\frac{d(\psi_{ts})^2\varepsilon+1}{d(\psi_{ts})^2+\varepsilon}.$$

Proof. Using the approach in (17, Theorem 12) to obtain distance bounds on incoming error products in dynamic-range measure and applying our Theorem 2, we obtain our corollary. □

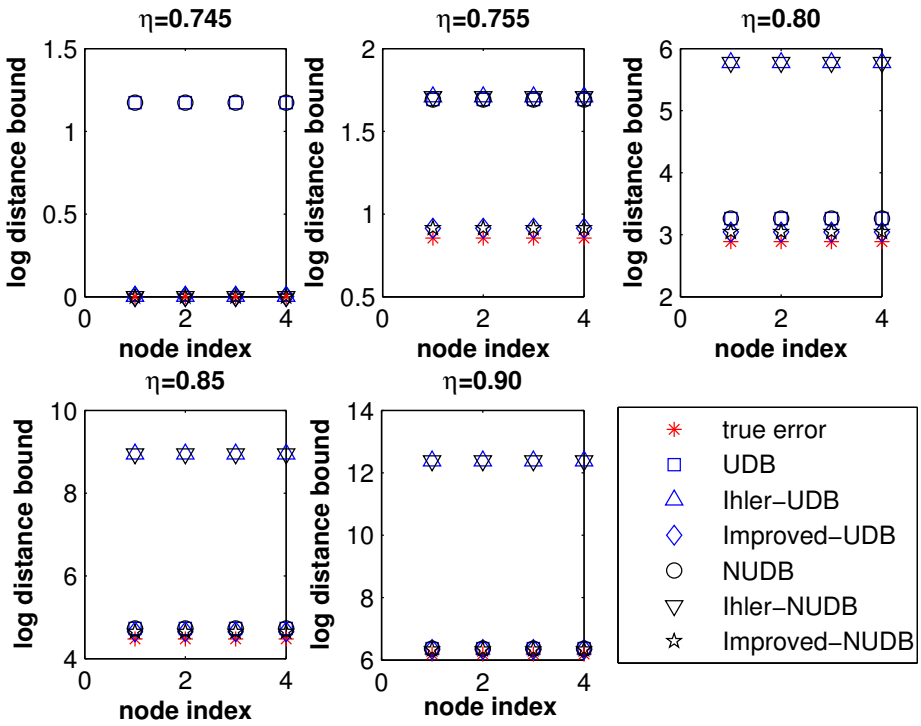


Figure 13. True distance, uniform distance bounds and non-uniform distance bounds for the graph in Figure 12(a) with various η's. The empirical critical value of η for LBP to converge is $\eta < 0.75$.

Let see how our *uniform distance bound* and *improved uniform distance bound* perform for graphical models in Figure 12 by comparison to the *Fixed-point distance bound* in (17). Let

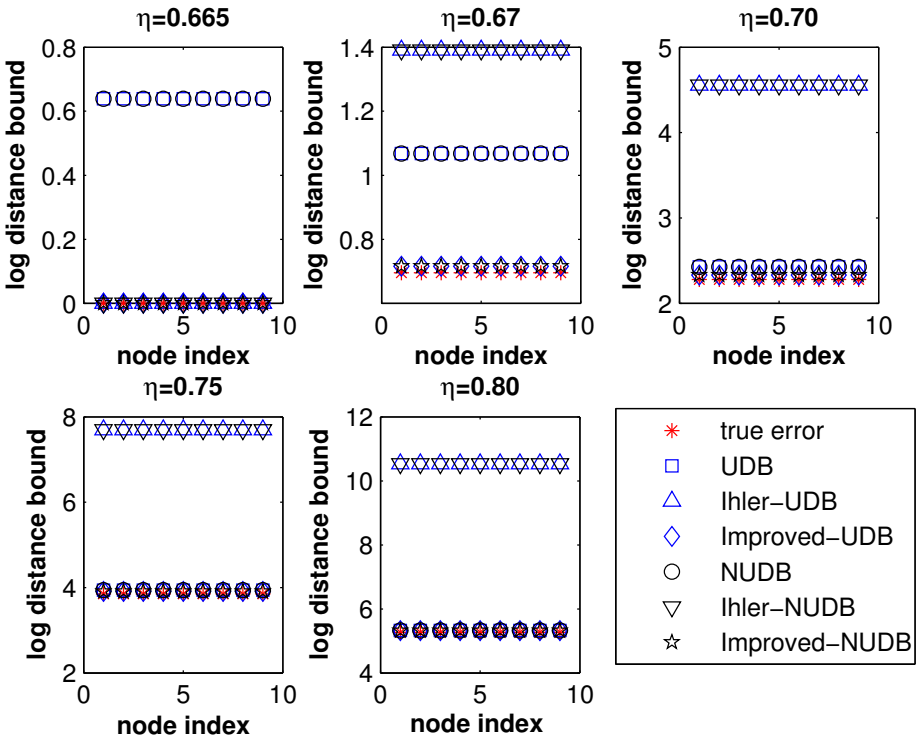


Figure 14. True distance, uniform distance bounds and non-uniform distance bounds for the graph in Figure 12(c) with various η's. The empirical critical value of η for LBP to converge is $\eta < 0.67$.

all the pairwise potential functions be $\begin{pmatrix} \eta & 1-\eta \\ 1-\eta & \eta \end{pmatrix}$ where $\eta > 0.5$ and all the single node potentials be $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Therefore, $d(\psi_{ts}) = \sqrt{\eta/(1-\eta)}$ and $d(\psi_{t\star}) = 1$ for $\forall\ (t,s) \in \mathbb{E}$.

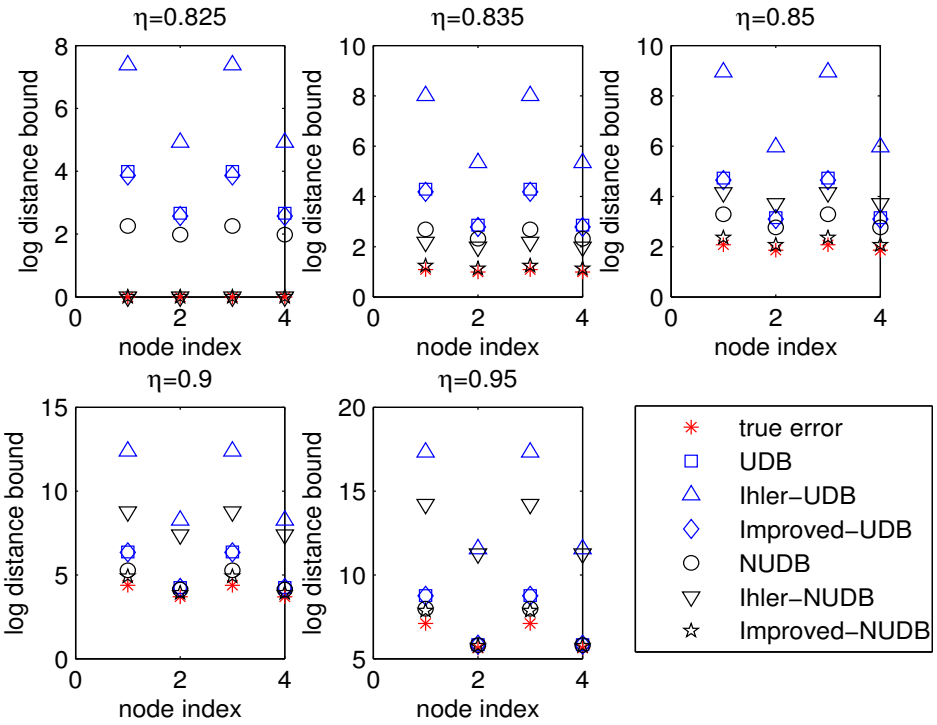


Figure 15. True distance, uniform distance bounds and non-uniform distance bounds for the graph in Figure 12(b) with various η's. The empirical critical value of η for LBP to converge is $\eta < 0.83$.

We compare the following bounds in our simulations: UDB, our uniform distance bound in Corollary 4; Improved-UDB, our improved uniform distance bound in Corollary 7; Ihler-UDB, Fixed-point distance bound in (17, Theorem 13). Figure 13 - Figure 16 illustrate the performances of those bounds for graphs in Figure 12(a), (c), (b) and (d), respectively.

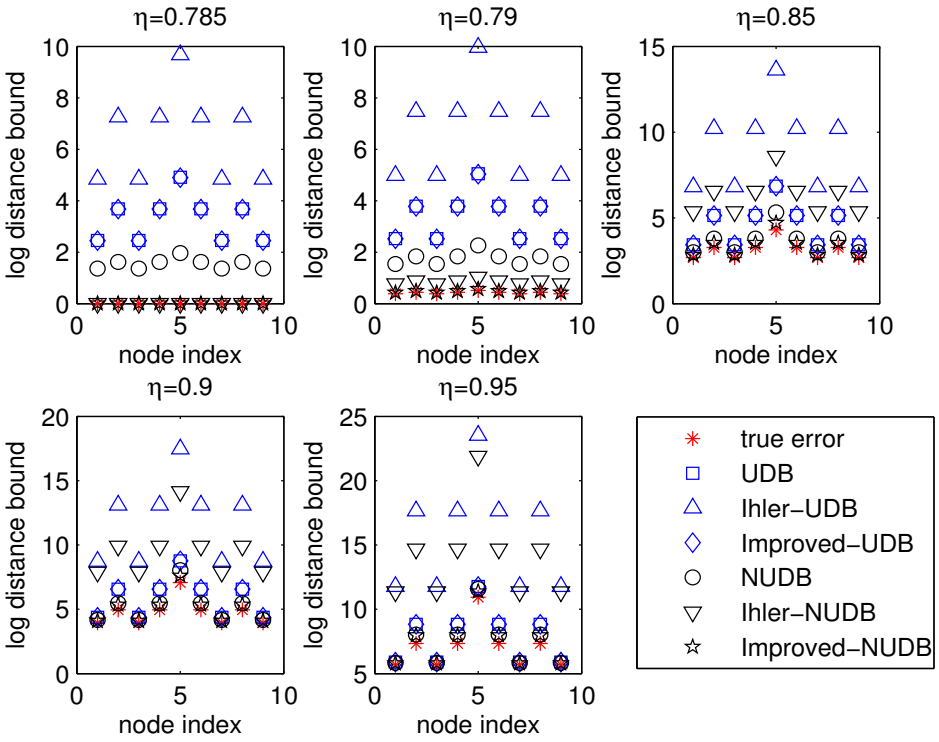


Figure 16. True distance, uniform distance bounds and non-uniform distance bounds for the graph in Figure 12(d) with various η's. The empirical critical value of η for LBP to converge is $\eta < 0.79$.

Graphs in Figure 12(a) and (c) are uniform (uniform degrees, uniform potential functions). Given a specific η, all nodes have the same distance bound. For those two graphs, the empirical critical values of η with respect to the convergence of LBP are 0.75 and 0.67 respectively. We can see that, for various η's, our Improved-UDBs are very close to the true errors between beliefs. Our UDBs become tighter when η increases, while Ihler-UDBs become looser. From Figure 13

and Figure 14, we can see that, compared to Ihler-UDB, our UDB requires stricter critical values of η to ensure error bounds to be zeros. Specifically, for Figure 13, when $\eta = 0.745$, our UDBs are non-zeros and Ihler-UDBs are zeros; hence, our UDB requires $\eta < 0.745$ for the convergence of LBP, while Ihler-UDB only requires $\eta < 0.75$. Nevertheless, the critical values by our UDB are 0.735 for Figure 12(a) and 0.66 for Figure 12(c), which are close to the empirical critical values. Based on our UDB and Ihler-UDB, our Improved-UDBs will approximate zeros when η approaches 0.75 and give tightest distance bounds for any η.

2.3.2 Non-Uniform Distance Bound

Figure 12(b) and Figure 12 (d) are non-uniform graphs. Because uniform distance bounds are computed locally, beliefs on the nodes with different topologies will have different error bounds, which can be observed from Figure 15 and Figure 16. We can also find that when the true errors are zeros, uniform bounds are not all zeros. In other words, η must be smaller than the empirical critical value to ensure the largest uniform distance bounds to be zero. Furthermore, in such cases, uniform convergence conditions derived from uniform distance bounds will not perform well as for uniform graphs. Therefore, when every loop contains potentials with various strengths and each node has different topology, we present the following *non-uniform distance bound* and *improved non-uniform distance bound*.

Corollary 8. ***(Non-uniform Distance Bound)***

The non-uniform log-distance bound of fixed points on belief at node s after $n \geq 1$ iterations is

$$\sum_{t \in \Gamma_s} \log\left(\frac{d(\psi_{ts})d(\psi_{t\star})\varepsilon_{ts}^{n} + 1}{d(\psi_{ts})d(\psi_{t\star}) + \varepsilon_{ts}^{n}}\right)^2,$$

where ε_{ts}^{i} *is updated by*

$$\log \varepsilon_{ts}^{i} = \sum_{u \in \Gamma_t \backslash s} \log(\frac{d(\psi_{ut})d(\psi_{u\star})\varepsilon_{ut}^{i-1} + 1}{d(\psi_{ut})d(\psi_{u\star}) + \varepsilon_{ut}^{i-1}})^2$$

with initial condition

$$\log \varepsilon_{ut}^{1} = \sum_{v \in \Gamma_u \backslash t} \log(d(\psi_{vu})d(\psi_{v\star}))^2.$$

Proof. The result can be easily proved from Corollary 4, by defining the *error bound-variation function* in (Equation 2.14) as follows:

$$G_{ts}(\log \varepsilon_{ts}^{i}) = \log \prod_{u \in \Gamma_t \backslash s} \Delta_{ut}(\varepsilon_{ut}^{i-1}) - \log \varepsilon_{ts}^{i} = \sum_{u \in \Gamma_t \backslash s} \log(\frac{d(\psi_{ut})d(\psi_{u\star})\varepsilon_{ut}^{i-1} + 1}{d(\psi_{ut})d(\psi_{u\star}) + \varepsilon_{ut}^{i-1}})^2 - \log \varepsilon_{ts}^{i}.$$

□

Similarly, based on the fact that the dynamic-range measure gives better convergence condition than the maximum-error measure, we improve the previous non-uniform distance bound in the following.

Corollary 9. ***(Improved Non-uniform Distance Bound)***

The improved non-uniform log-distance bound of fixed points on belief at node s *after* $n \geq 1$ *iterations is*

$$\sum_{t \in \Gamma_s} \log(\frac{d(\psi_{ts})d(\psi_{t\star})\varepsilon_{ts}^{n} + 1}{d(\psi_{ts})d(\psi_{t\star}) + \varepsilon_{ts}^{n}})^2,$$

where ε_{ts}^{i} *is updated by*

$$\log \varepsilon_{ts}^{i} = \sum_{u \in \Gamma_t \backslash s} \log \frac{d(\psi_{ut})^2 \varepsilon_{ut}^{i-1} + 1}{d(\psi_{ut})^2 + \varepsilon_{ut}^{i-1}}$$

with initial condition $\log \varepsilon_{ut}^{1} = \sum_{v \in \Gamma_u \backslash t} \log d(\psi_{vu})^2$.

Proof. The proof is similar to that for Corollary 7. □

Let see the performaces of our *non-uniform distance bound* and *improved non-uniform distance bound* for the graphs in Figure 12 compared with the non-uniform distance bound in (17, Thm. 14). We denote the bounds in our simulation as follows: NUDB, our non-uniform distance bound in Corollary 8; Improved-NUDB, our improved non-uniform distance bound in Corollary 9; Ihler-NUDB, non-uniform distance bound in (17, Theorem 13).

For uniform graphs in Figure 12(a) and (c), NUDB performs exactly the same as UDB. However, for non-uniform graphs in Figure 12(b) and (d), because NUDB propagates error bounds throughout the whole graph rather than on a local neighborhood, NUDBs are tighter than UDBs, which can be observed from Figure 15 and Figure 16. For various η's, our Improved-NUDBs always approach the true errors. Therefore, when our Improved-NUDB is zero, η almost equals the empirical critical value to ensure convergence of LBP. Though worse than Improved-NUDB, our NUDB performs better than Ihler-NUDB when η is far way from the area of convergence.

2.3.2.1 Non-Uniform Convergence

Based on our Improved-NUDB or Ihler-NUDB, a sufficient convergence condition of LBP can be derived, which is based on the dynamic-range measure of propagating errors.

For each cycle-involved vertex v, $T_B(\mathbb{G}, v, n)$ is the corresponding Bethe tree. Let $\mathbb{V}_B$ be the set of vertices in the Bethe tree. For $w_i \in \mathbb{V}_B, i = 0, ..., |\mathbb{V}_B| - 1$, $l(w_i)$ is the labelling function which maps w_i to the original vertex in $\mathbb{G}$. Let $l(w_0) = v$.

Theorem 10. ***(Non-Uniform Convergence Condition)***

For a graphical model $\mathbb{G}(\mathbb{V}, \mathbb{E})$, $\{T_B(\mathbb{G}, v, n), v \in \mathbb{V}\}$ is the set of n-th level Bethe trees. The non-uniform sufficient condition for the convergence of sum-product algorithm is:

$$\max_{l(w_0)=v\in\mathbb{V}} \sum_{w_i\in\Gamma_{w_0}} \frac{d(\psi_{l(w_i)l(w_0)})^2 - 1}{d(\psi_{l(w_i)l(w_0)})^2 + 1} \sum_{w_j\in\Gamma_{w_i}\setminus w_0} \frac{d(\psi_{l(w_j)l(w_i)})^2 - 1}{d(\psi_{l(w_j)l(w_i)})^2 + 1} ... \sum_{w_r\in\Gamma_{wq}\setminus w_p} \frac{d(\psi_{l(w_r)l(w_q)})^2 - 1}{d(\psi_{l(w_r)l(w_q)})^2 + 1} < 1, \tag{2.16}$$

where Γ_{w_i} is the set of neighbors of w_i.

Proof. Recall that in the proof of *uniform convergence condition*, we use an *error bound-variation function* $G_{sp}(\log \varepsilon)$, which is originally to describe $(\log \varepsilon_{sp}^{i+1} - \log \varepsilon_{ts}^{i})$, for $\forall (s, p) \in \mathbb{E}$. For each $T_B(\mathbb{G}, v, n)$, let us introduce the following error bound-variation function:

$$G_v(\{\log \varepsilon_{w_i w_0}\}, \log \varepsilon) = \textstyle\sum_{w_i\in\Gamma_{w_0}} \log \frac{d(\psi_{l(w_i)l(w_0)})^2 \varepsilon_{w_i w_0} + 1}{d(\psi_{l(w_i)l(w_0)})^2 + \varepsilon_{w_i w_0}} - \log \varepsilon,$$

$$\log \varepsilon_{w_i w_0} = \textstyle\sum_{w_j\in\Gamma_{w_i}\setminus w_0} \log \frac{d(\psi_{l(w_j)l(w_i)})^2 \varepsilon_{w_j w_i} + 1}{d(\psi_{l(w_j)l(w_i)})^2 + \varepsilon_{w_j w_i}},$$

$$...$$

$$\log \varepsilon_{w_q w_p} = \textstyle\sum_{w_r\in\Gamma_{wq}\setminus w_p} \log \frac{d(\psi_{l(wr)l(wq)})^2 \varepsilon + 1}{d(\psi_{l(wr)l(wq)})^2 + \varepsilon},$$

where $\{w_r\}$ is the set of leaf nodes of $T_B(\mathbb{G}, v, n)$.

To guarantee LBP to converge, it is sufficient to have $G_v(\log \varepsilon) < 0$ for $\forall \log \varepsilon > 0$. Because $G_v(\log \varepsilon = 0) = 0$, when $G'_v(\log \varepsilon = 0) < 0$, we will definitely have $G_v(0 < \log \varepsilon < \delta) < 0$, where δ is a small positive value. When $G_v(\log \varepsilon)$ is concave, δ can be infinity so that the convergence of LBP is true for $\forall \log \varepsilon > 0$. However, because $G_v(\log \varepsilon)$ is not guaranteed to be concave, we will only obtain local convergence for an infinitesimal δ.

Define $f_{w_j w_i}(\varepsilon_{w_j w_i}) = \log \frac{d(\psi_{l(w_j)l(w_i)})^2 \varepsilon_{w_j w_i} + 1}{d(\psi_{l(w_j)l(w_i)})^2 + \varepsilon_{w_j w_i}}$. Thus, we have the first derivative of $G_v(\{\log \varepsilon_{w_i w_0}\}, \log \varepsilon)$ as follows:

$$\frac{\partial G_v(\{\log \varepsilon_{w_i w_0}\}, \log \varepsilon)}{\partial \log \varepsilon} = \sum_{w_i \in \Gamma_{w_0}} f'_{w_i w_0} \sum_{w_j \in \Gamma_{w_i} \setminus w_0} f'_{w_j w_i} .... \sum_{w_r \in \Gamma_{wq} \setminus w_p} f'_{w_r w_q} - 1,$$

where $f' = \frac{\partial f(\log \varepsilon)}{\partial \log \varepsilon} = \frac{(d(\psi)^4 - 1)\varepsilon}{(d(\psi)^2 \varepsilon + 1)(d(\psi)^2 + \varepsilon)}$. Plugging $\log \varepsilon = 0$ into the previous equation, we obtain our non-uniform convergence condition. □

When we derive our *non-uniform convergence condition* based on the SAW tree, we will have the following corollary. For each cycle-involved vertex v, $T_{SAW}(\mathbb{G}, v, n)$ is the corresponding SAW tree. Let $\mathbb{V}_{SAW}$ be the set of vertexes in the SAW tree. For $w_i \in \mathbb{V}_{SAW}, i = 0, ..., |\mathbb{V}_{SAW}| - 1$, $l(w_i)$ is the labelling function which maps w_i to the original vertex in $\mathbb{G}$. Let $l(w_0) = v$.

Corollary 11. ***Non-uniform convergence condition (SAW tree)***

For a graphical model $\mathbb{G}(\mathbb{V}, \mathbb{E})$, $\{T_{SAW}(\mathbb{G}, v, n), v \in \mathbb{V}\}$ is the set of SAW trees. The non-uniform sufficient condition for the convergence of sum-product algorithm is:

$$\max_{l(w_0)=v \in \mathbb{V}} \sum_{w_i \in \Gamma_{w_0}} \frac{d(\psi_{l(w_i)l(w_0)})^2 - 1}{d(\psi_{l(w_i)l(w_0)})^2 + 1} \sum_{w_j \in \Gamma_{w_i} \setminus w_0} \frac{d(\psi_{l(w_j)l(w_i)})^2 - 1}{d(\psi_{l(w_j)l(w_i)})^2 + 1} ... \sum_{w_r \in \Gamma_{wq} \setminus w_p} \frac{d(\psi_{l(w_r)l(w_q)})^2 - 1}{d(\psi_{l(w_r)l(w_q)})^2 + 1} < 1,$$

where Γ_{w_i} *is the set of neighbors of* w_i.

When a graph has uniform potential functions with strength $d(\psi)$, to ensure convergence, it is sufficient to have

$$\max_{l(w_0)=v\in\mathbb{V}} \sum_{w_i\in\Gamma_{w_0}} \frac{d(\psi)^2-1}{d(\psi)^2+1} \sum_{w_j\in\Gamma_{w_i}\setminus w_0} \frac{d(\psi)^2-1}{d(\psi)^2+1} \cdots \sum_{w_r\in\Gamma_{wq}\setminus w_p} \frac{d(\psi)^2-1}{d(\psi)^2+1} < 1. \tag{2.17}$$

Let us apply our *non-uniform convergence condition* to the graphs in Figure 12(b) and (d) with uniform potential functions as in the previous simulations. For the graph in Figure 12(b), we obtain the critical value $\eta < 0.78$ for convergence of LBP, which is closer to the empirical value $\eta < 0.83$, compared to $\eta < 0.75$ obtained by *uniform convergence condition*. For the graph in Figure 12(d), we obtain the critical value $\eta < 0.77$, while the empirical value is $\eta < 0.79$ and the critical value obtained by *uniform convergence condition* is $\eta < 0.67$.

2.4 Convergence of Loopy Belief Propagation

2.4.1 Sparsity and Convergence

To compute our *non-uniform convergence condition* is not easy, when the graph is not sparse or not symmetric. Nevertheless, our Theorem 10 can be used to deduce convergence properties of sparse graphs.

It lacks theoretical verification that the more sparse a graph is, the less stricter is its convergence condition. However, the definition of sparse graphs is vague; therefore, to be confined, we would relate sparsity with partial graphs. Let us define partial graphs and introduce the convergence property of such graphs in the following.

Definition 1. ***(Reduction)***

A path composed of two edges (v_1, v_2) *and* (v_2, v_3) *can be reduced to a path composed of one edge* (v_1, v_3)*, where* $\psi_{v_1v_3}(x_{v_1}, x_{v_3}) = \int_{x_{v_2}} \psi_{v_1v_2}(x_{v_1}, x_{v_2})\psi_{v_2v_3}(x_{v_2}, x_{v_3})dx_{v_2}$*, when there is no branch on the path.*

Definition 2. ***(Extension)***

A path composed of one edge (v_1, v_3) *can be extended to a path composed of two edges* (v_1, v_2) *and* (v_2, v_3)*, where* $\int_{x_{v_2}} \psi_{v_1v_2}(x_{v_1}, x_{v_2})\psi_{v_2v_3}(x_{v_2}, x_{v_3})dx_{v_2} = \psi_{v_1v_3}(x_{v_1}, x_{v_3})$.

Definition 3. ***(Partial Graphs)***

For two graphical models $\mathbb{G}_1(\mathbb{V}_1, \mathbb{E}_1)$ *and* $\mathbb{G}_2(\mathbb{V}_2, \mathbb{E}_2)$ *after reduction and extension, there exists an isomorphism between graphs* $\mathbb{G}_1(\mathbb{V}_1, \mathbb{E}_1)$ *and* $\mathbb{G}_2(\mathbb{V}_2^*, \mathbb{E}_2^*)$*, when* $\mathbb{V}_2^* \subseteq \mathbb{V}_2$ *and* $\mathbb{E}_2^* \subset \mathbb{E}_2$*. When* $\mathbb{E}_2 - \mathbb{E}_2^*$ *is cycle-involved, we call* $\mathbb{G}_1$ *a partial graph of* $\mathbb{G}_2$ *and denote it as* $\mathbb{G}_1 \subset \mathbb{G}_2$.

Theorem 12. ***(Strictness of Convergence Condition for Two Partial Graphs)***

Given $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *as defined in Definition 3, assume that* $\mathbb{G}_1 \subset \mathbb{G}_2$*. Assume the dynamic-range measures of potential functions for edges in* $\mathbb{E}_1$ *are not greater than those of potential functions for corresponding edges in* $\mathbb{E}_2^*$*. Then, when LBP for* $\mathbb{G}_2(\mathbb{V}_2, \mathbb{E}_2)$ *converges, LBP for* $\mathbb{G}_1(\mathbb{V}_1, \mathbb{E}_1)$ *must converge; however, the reverse implication is not true in general.*

Proof. Because $\mathbb{G}_1 \subset \mathbb{G}_2$ and $\mathbb{E}_2 - \mathbb{E}_2^*$ are cycle-involved, $T_B(\mathbb{G}_1, v, n) \subset T_B(\mathbb{G}_2, v, n)$. Therefore, LHS of Inequality (Equation 2.16) for $\mathbb{G}_2$ has more summands than the corresponding quantity for $\mathbb{G}_1$. When $\mathbb{G}_2$ satisfies the convergence condition, $\mathbb{G}_1$ must satisfy it. However, when $\mathbb{G}_1$ satisfies the convergence condition, $\mathbb{G}_2$ may not satisfy it. □

When the potential functions of a graph are uniform, we have the following corollary.

Corollary 13. ***(Critical Values of Convergence for Two Partial Graphs)*** *Given $\mathbb{G}_1 \subset \mathbb{G}_2$, $\mathbb{G}_1$ and $\mathbb{G}_2$ have uniform potential functions $\psi_i = \begin{pmatrix} \eta_i & 1-\eta_i \\ 1-\eta_i & \eta_i \end{pmatrix}, i = 1, 2$ on all edges. Then, the critical values for convergence of LBP satisfy $\eta_2 < \eta_1$.*

Proof. Because LHS of (Equation 2.17) for $\mathbb{G}_2$ has more summands than the corresponding quantity for $\mathbb{G}_1$, we easily have $d(\psi_2) < d(\psi_1)$ to satisfy the inequality. Because $d(\psi_i) = \sqrt{\eta_i/(1-\eta_i)}$, we get $\eta_2 < \eta_1$. □

Our Theorem 12 and Corollary 13 can be easily extended to strictness of convergence condition of LBP for a set of partial graphs, and for those with uniform potential functions.

Corollary 14. ***(Strictness of Convergence Condition for Set of Partial Graphs)*** *Given $\mathbb{G}_1 \subset \mathbb{G}_2 ... \subset \mathbb{G}_k$, assuming the dynamic-range measures of potential functions on isomorphous edges of those graphs are correspondingly non-decreasing in the previous partial order, LBP convergence for $\mathbb{G}_m$ implies LBP convergence for $\mathbb{G}_n$, where $m > n$ and $m, n = 1, ..., k$. However, the reverse implication is not true in general.*

Corollary 15. ***(Critical Value of Convergence for Set of Partial Graphs)*** *Given $\mathbb{G}_1 \subset \mathbb{G}_2 ... \subset \mathbb{G}_k$, $\mathbb{G}_1$,..., $\mathbb{G}_k$ have uniform potential functions $\begin{pmatrix} \eta_i & 1-\eta_i \\ 1-\eta_i & \eta_i \end{pmatrix}, 1 \leq i \leq k$ on all edges. Then, the critical values for convergence of LBP satisfy $\eta_k < \eta_{k-1} ... < \eta_1$.*

By our Corollary 14 on partially ordered graphs, we can conclude that graphs with less cycle-induced edges are more sparse and thus have weaker convergence condition. It is intuitively

true that the strength of potential functions for Figure 12(a) or Figure 12(c) should be weaker than that for Figure 12(b) or Figure 12(d) to ensure convergence of LBP. However, it can be soundly verified by our previous corollaries.

2.4.2 Walk-Summability and Convergence

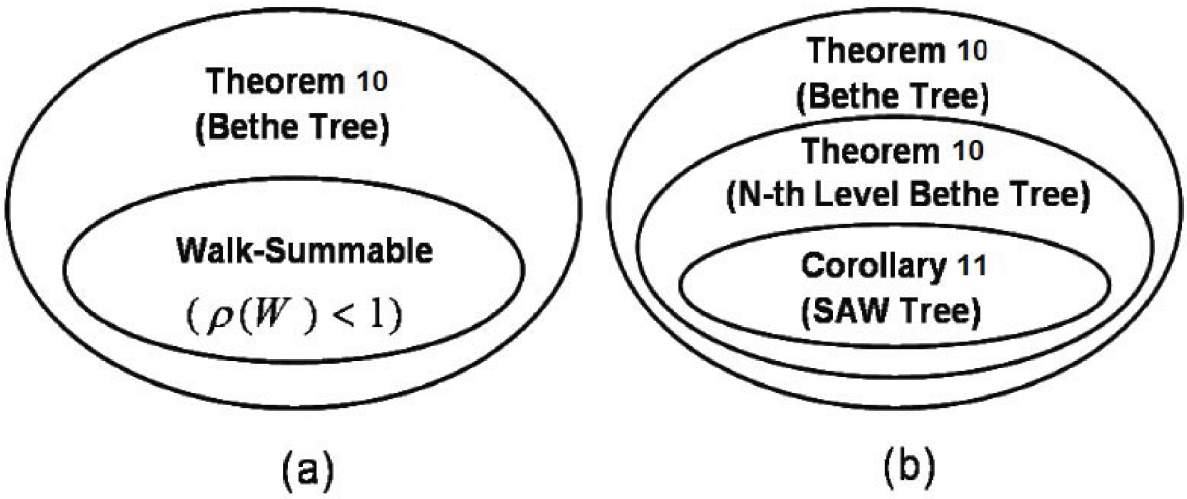


Figure 17. Diagram summarizing mildness of convergence conditions. The SAW tree is a partial tree of the N-level Bethe tree, therefore, convergence condition based on the SAW tree is stronger.

When a model is related to a Gaussian Distribution, the analysis of its properties becomes comparatively simple. Proposition 21 in (44) states that "If a model on a (Gaussian) graph G is walk-summable, then LBP is well-posed, the means converge to the true means and the

LBP variances converge to walk-sums over the backtracking self-return walks at each node". Enlightened by the analysis for Gaussian graphical model, we will extend its walk-summability perspective to general graphical models and relate the convergence property of LBP with walk-sums in the following.

Let R be the correlation matrix of a Gaussian graphical model. When $\sum_l \bar{R}^l$ converges, where $\bar{R}_{ij} = |R_{ij}|$ and l is the length of walk, the Gaussian graphical model is walk-summable. Let $\rho(\bar{R})$ be the spectral radius of $\bar{R}$. The previous definition of walk-summability is equivalent to the spectral radius satisfying $\rho(\bar{R}) < 1$. For a Gaussian graphical model, the interaction between two random variables is the correlation coefficient. However, for a general graphical model, we have multi-dimensional potential functions. We hope to find a scaler quantity to represent the interaction between two random variables as well. In Theorem 10, we add up all the n-th step walks from a root node, where a walk on edge (t, s) is the quantity $\frac{d(\psi_{ts})^2-1}{d(\psi_{ts})^2+1}$. Similarly to the correlation coefficient, we can treat this quantity as the strength of the interaction. Let W be the strength matrix with entry $w_{ts} = \frac{d(\psi_{ts})^2-1}{d(\psi_{ts})^2+1}$. When a general graphical model satisfies $\rho(W) < 1$, we say it as walk-summable, which has also been shown to be related with the convergence of LBP. (18) present a convergence condition for general graphical model: spectral radius $\rho(\hat{W}) < 1$, by using a different strength matrix $\hat{w}_{ts} = \frac{\hat{d}(\psi_{ts})^2-1}{\hat{d}(\psi_{ts})^2+1}$, where $\hat{d}(\psi_{ts})^2 = \max_{x_t,x_s,\hat{x}_t,\hat{x}_s} \frac{\psi_{ts}(x_t x_s)\psi_{ts}(\hat{x}_t\hat{x}_s)}{\psi_{ts}(\hat{x}_t x_s)\psi_{ts}(x_t\hat{x}_s)}$. This convergence condition is equivalent to the walk-summability of the graphical model with the strength matrix $\hat{W}$.

Let us compare the walk-summability of a general graph with our non-uniform convergence condition in Theorem 10. Because $\rho(W) < 1$ is equivalent to $\|W^N\|_1 < 1, N \to \infty$ (18), and

the walk-sum in the LHS of (Equation 2.16) for a N-level Bethe tree is smaller than $\|W^N\|_1$, we can derive that our non-uniform convergence condition in Theorem 10 is milder than $\rho(W) < 1$, which is illustrated in Figure 17(a).

When the convergence condition based on N-level Bethe tree is satisfied, the convergence condition based on infinite Bethe tree must be satisfied, because the error bounds are guaranteed to decrease after N iterations of error propagation. Similarly, convergence condition based on N-level Bethe tree is milder than that based on SAW tree. Mildness of convergence conditions is shown in Figure 17(b).

2.5 Accuracy and Convergence Rate of Loopy Belief Propagation

In the following, we will analyze the performance of LBP with respect to accuracy and convergence rate.

2.5.1 Accuracy Bounds for Loopy Belief Propagation

Recently, (25) presented an accuracy bound for LBP which relates the belief of a random variable to its true marginal. He showed that there exists a configuration on some nodes of the SAW tree rooted at certain node s of the original graph, such that the true maginal at node s is equal to the belief at root s of the SAW tree. Therefore, given certain external force functions on a subset of nodes, he adopted the non-uniform distance bound in (17, Thm. 14) to obtain an accuracy bound between beliefs and true marginals.

Given $d(p(x)/b(x)) \leq \delta$, his accuracy bound is as follows:

$$\frac{b(x)}{\delta^2 + (1-\delta^2)b(x)} \leq p(x) \leq \frac{\delta^2 b(x)}{1-(1-\delta^2)b(x)}, \tag{2.18}$$

where δ is an error bound in dynamic-range measure, $p(x)$ is the normalized true marginal and $b(x)$ is the normalized belief. Note that δ in (25, Lemma 5) should be δ^2.

Because our *improved non-uniform distance bound* has been shown tighter than his non-uniform bound, we can improve his accuracy bound between the belief and the true marginal. Let $\max_x |\log p(x)/b(x)| \leq \log \varepsilon$, where ε is an error bound in maximum-error measure applying our Corollary 8, under certain external force functions on a subset of nodes of the SAW tree. Therefore, we have the accuracy bound as $b(x)/\varepsilon \leq p(x) \leq \varepsilon b(x)$, where $\varepsilon < \delta^2$. Combining our accuracy bound with the bound in (Equation 2.18), we have the improved bound

$$\max\{b(x)/\varepsilon, \frac{b(x)}{\delta^2 + (1-\delta^2)b(x)}\} \leq p(x) \leq \min\{\varepsilon b(x), \frac{\delta^2 b(x)}{1-(1-\delta^2)b(x)}\}.$$

2.5.2 Rate of Convergence and Residual Scheduling

For an iterative algorithm such as LBP, the rate of convergence is an important criteria of performance. We will analyze the convergence rate of LBP by looking into the decreasing gradient of error bounds on messages. The error bound-variation function $G_{sp}(\log \varepsilon)$ in (Equation 2.14) is a measure of the variation of error bounds between successive iterations; on the other hand, it reflects how fast LBP converges, because the smaller $G_{sp}(\log \varepsilon)$ is, the faster error bounds decrease. Because dynamic-range measure is better than maximum-error measure in terms of convergence of LBP, we will use the error bound-variation function as follows:

$$G_{sp}(\log \varepsilon) = \log \prod_{t \in \Gamma_s \setminus p} \frac{d(\psi_{ts})^2 \varepsilon + 1}{d(\psi_{ts})^2 + \varepsilon} - \log \varepsilon,$$

where ε is an error bound in dynamic-range measure on incoming error product. We will use the first derivative of the function as a metric on the rate of convergence:

$$G_{sp}^{(1)}(\log \varepsilon) = \sum_{t \in \Gamma_s \setminus p} \frac{\varepsilon((d(\psi_{ts})^4 - 1)}{(d(\psi_{ts})^2 \varepsilon + 1)(d(\psi_{ts})^2 + \varepsilon)} - 1.$$

Recall that $G_{sp}^{(1)}(\log \varepsilon)$ should be less than zero to ensure convergence. When we have infinitesimal error disturbance, $|G_{sp}^{(1)}(0)|$ will be used as the local rate of convergence. Because our rate of convergence varies with each direction of message passing, messages on the direction with the greatest rate will be updated prior to others in dynamic scheduling.

Some works have been done to utilize message residuals as a way of priority in dynamic scheduling by (29) and (30). Rather than calculating future message residuals, (30) utilized their upper-bounds as estimates of message residuals in their scheduling algorithm *RBP0L*. They adopted maximum-error measure as a metric of message residuals, which was defined by them as $r(m_{ts}) = \max_{x_s} |\log e_{ts}(x_s)|$. They showed that by the contraction property of maximum-error measure it can be upper-bounded as $r(m_{ts}) \leq \sum_{u \in \Gamma_t \setminus s} r(m_{ut})$. However, their upper-bound is not theoretically sound, because they ignored the normalization factor in their proof. Therefore, we can modify their *RBP0L* by utilizing our upper-bound in (Equation 2.8). However, from some simulation results, the rate of convergence of their original heuristic algorithm is not substantially improved.

2.6 Message Errors for Completely Uniform Binary Graphs

In Section 2.3, we discussed uniform and non-uniform distance bounds on beliefs. An error bound-variation function was introduced to study the variation between error bounds at one level and those at its upper-level in the Bethe tree. However, to study the mechanism behind message passing, we are more interested to know the variation of true errors. Because it is usually hard to identify the true error-variation function except for binary graphs, in this section, we will explore fixed points and true error variation functions for binary graphs.

Let us first introduce a well-studied binary graph – Ising model. The probability measure of Ising model can be expressed as:

$$P(x) = \frac{1}{Z} \exp\Big(\sum_{(s,t)\in\mathbb{E}} J_{st} x_s x_t + \sum_{s\in\mathbb{V}} \theta_s x_s \Big), \tag{2.19}$$

corresponding to $\psi_{st}(x_s, x_t) = \exp(J_{st} x_s x_t)$ and $\psi_s(x_s) = \exp(\theta_s x_s)$ in (Equation 2.1). Because $\{x_s\}$ are ± 1-valued, potential functions can also be expressed as $\begin{pmatrix} \exp(J_{st}) & \exp(-J_{st}) \\ \exp(-J_{st}) & \exp(J_{st}) \end{pmatrix}$ and $\begin{pmatrix} \exp(\theta_s) \\ \exp(-\theta_s) \end{pmatrix}$. However, rather than working on the Ising model, we will study a completely uniform model (uniform connectivity, uniform potential functions) in the following, which has the pairwise potential functions $\begin{pmatrix} a & b \\ b & a \end{pmatrix}$ and single-node potential functions $\begin{pmatrix} c \\ d \end{pmatrix}$, where a, b, c, d are positive. Similar to (Equation 2.3), we will multiply single-node potentials in beliefs and only discuss the influence of pairwise potential functions on message errors.

We can easily obtain that a completely uniform graph has uniform messages.

Property 1. *For a completely uniform graphical model, when synchronous LBP reaches a steady state, all messages are the same.*

Proof. Completely uniform graphs are topologically invariant for each node. In other words, each message has the same LBP update equation. If some messages are different, for the symmetric network, LBP will not reach a steady state. □

Because messages have the same LBP update equation, we can calculate the fixed-point messages exactly and discuss the distances between them.

2.6.1 Fixed Points and Quasi-Fixed Points

Let us first discuss fixed-point messages for completely uniform graphs. Assume the degree of each node is k. Let $m_{out} = \begin{pmatrix} y \\ 1-y \end{pmatrix}$ denote the outgoing message and $m_{in} = \begin{pmatrix} x \\ 1-x \end{pmatrix}$ denote each incoming message. Therefore, we have the following LBP updating function:

$$y = F(x) = \frac{ax^k + b(1-x)^k}{(a+b)(x^k + (1-x)^k)}. \tag{2.20}$$

We can easily find that (Equation 2.20) is symmetric with respect to the point $(x = 0.5, y = 0.5)$. Synchronous LBP update corresponds to the fixed-point iteration function $x_{n+1} = F(x_n)$, where n is the iteration number. When $x_{n+1} = x_n$, LBP message reaches a *fixed point*. However, we sometimes have $x_{n+k} = x_n$ or $F^k(x) = x$, where $F^k(x)$ is the composition function of $F(x)$ with itself k times, which shows kth-order periodicity. We define the solutions to $F^k(x) = x, k > 1$

as *quasi-fixed points*, among which a belief network will oscillate. In the following, we will show that LBP for completely uniform binary graphs will have at most second order periodicity.

Property 2. *LBP updating function in* (Equation 2.20) *has at most three real fixed points.*

Proof. The second derivative of $F(x)$ is as follows: when $a > b$

$$F^{(2)}(x) = ((2x-k-1)x^k+(2x+k-1)(1-x)^k)\times\frac{k(a-b)x^{k-2}(1-x)^{k-2}}{(a+b)(x^k+(1-x)^k)^3} = \begin{cases} > 0, x \in (0, 0.5) \\ < 0, x \in (0.5, 1) \\ = 0, x = 0, 1, 0.5 \end{cases} .$$

We can see that $F(x)$ is strictly convex when $0 < x < 0.5$ and strictly concave when $0.5 < x < 1$. Similarly, for $a < b$, $F(x)$ is strictly concave when $0 < x < 0.5$ and strictly convex when $0.5 < x < 1$. When this function intersects with an arbitrary line, there must be at most three crossing points. As shown in Figure 18(a), it must have at most three crossings with $y = x$; similarly with $y = 1 - x$ in Figure 18(b). □

We will show the symmetry of fixed-point messages for uniform binary graphs as follows.

Property 3. *For a completely uniform binary graph, synchronous LBP will either converge to the unique fixed point* $\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$, *or converge to one of* $\begin{pmatrix} x^* \\ 1-x^* \end{pmatrix}$ *and* $\begin{pmatrix} 1-x^* \\ x^* \end{pmatrix}$ *when* $a > b$ *(ferromagnetic), or oscillate between* $\begin{pmatrix} x^* \\ 1-x^* \end{pmatrix}$ *and* $\begin{pmatrix} 1-x^* \\ x^* \end{pmatrix}$ *when* $a < b$ *(anti-ferromagnetic). When* $a > b$, x^* *is the solution to* $x^* = F(x^*)$; *otherwise,* x^* *is the solution to* $1 - x^* = F(x^*)$.

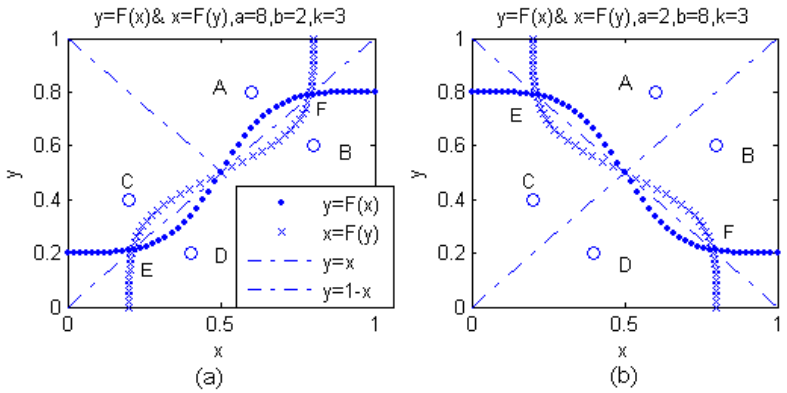


Figure 18. LBP updating function in (Equation 2.20) for $a > b$ and $a < b$.

Proof. Let us analyze the fixed points by solving the set of equations

$$y = F(x) \tag{2.21a}$$

$$x = F(y) \tag{2.21b}$$

which corresponds to second order periodicity $x = F^2(x)$. The set of equations is depicted in Figure 18 for $a > b$ and $a < b$ respectively. We can easily find that $F(x)$ and $F(y)$ are symmetric with respect to $y = x$. Moreover, because $F(x)$ is symmetric about the point $(0.5, 0.5)$, we have $F(1-x) = 1 - F(x)$. Therefore, it is easy to see that $F(x)$ and $F(y)$ are also symmetric with respect to $y = 1 - x$. Let us check whether the two functions are symmetric with respect to other lines such as $y = \beta + \alpha x$. Substitute $y = \beta + \alpha x$ and $x = \frac{1}{\alpha}(y - \beta)$ in (Equation 2.21a). We have $\beta + \alpha x = \frac{a(y-\beta)^k + b(\alpha-(y-\beta))^k}{(a+b)((y-\beta)^k + (\alpha-(y-\beta))^k)}$. For this equation to be always

equivalent to (Equation 2.21b), we have $(\alpha = 1, \beta = 0)$ or $(\alpha = -1, \beta = 1)$. Thus, the set of equations is only symmetric with respect to $y = x$ and $y = 1 - x$.

When $y = F(x)$ and $x = F(y)$ intersect, they must have crossing points on $y = x$ or $y = 1-x$. In the following, we will show that they do not cross elsewhere. When $a > b$, let us assume these two functions have one crossing point A not on $y = x$ and $y = 1 - x$, which is illustrated in Figure 18 (a). Due to the symmetry between $F(x)$ and $F(y)$, they must have the other three crossing points B, C and D shown in Figure 18 (a) respectively. Both functions must go through those points. The first derivative of $F(x)$ is $F^{(1)}(x) = \frac{k(a-b)x^{k-1}(1-x)^{k-1}}{(a+b)((1-x)^k+x^k)^2} = \begin{cases} > 0, a > b \\ \\ < 0, a < b \end{cases}$, which shows that function $F(x)$ is either monotonic increasing or monotonic decreasing. Because $y_B < y_A$, when $x_B > x_A$, we arrive at a contradiction with the monotonic increasing property under the condition $a > b$. Similar result is for $a < b$. According to Property 2, $y = F(x)$ and $x = F(y)$ have at most three real crossings points with an arbitrary line. Therefore, we can see that the set of equations will have at most three crossing points with either $y = x$ or $y = 1 - x$.

The set of equations in (Equation 2.21a) and (Equation 2.21b) has a naive fixed point $(0.5, 0.5)$. However, it is only stable when the set of equations in (Equation 2.21a) and (Equation 2.21b) crosses nowhere else on $y = x$ and $y = 1 - x$. When $a > b$ and $F^{(1)}(\frac{1}{2}) = \frac{k(a-b)}{(a+b)} > 1$, we can see that the belief network will either converge at fixed point E or at fixed point F on $y = x$. In this case, the fixed point at $x = 0.5$ is an unstable point. When $a < b$ and $F^{(1)}(\frac{1}{2}) < -1$, the belief network will eventually oscillate between E and F on $y = 1 - x$, which is shown in Figure 18 (b). The fixed point at $x = 0.5$ is again an unstable fixed point. Because $F(x)$ is

symmetric with respect to $(x = 0.5, y = 0.5)$, points E and F are symmetric with respect to $(x = 0.5, y = 0.5)$. □

From the previous property, we can conclude that completely uniform binary graphs will have at most second order periodicity. In other words, $F^{2n}(x) = x \Leftrightarrow F^2(x) = x$ and $F^{2n-1}(x) = x \Leftrightarrow F(x) = x$.

Let us calculate the fixed points and quasi-fixed points for the uniform graph in Figure 12(c) with $a = \eta$ and $b = 1 - \eta$. Solving $x = \frac{\eta x^3 + (1-\eta)(1-x)^3}{x^3 + (1-x)^3}$ and $1 - x = \frac{\eta x^3 + (1-\eta)(1-x)^3}{x^3 + (1-x)^3}$ yields the fixed points and quasi-fixed points respectively, for the graph in Figure 12(c). Specifically, we can obtain four solutions of fixed points $\{\frac{1}{2}, \frac{1}{2}, \frac{-2+\eta-\sqrt{-4+8\eta-3\eta^2}}{2(-2+\eta)}, \frac{-2+\eta+\sqrt{-4+8\eta-3\eta^2}}{2(-2+\eta)}\}$ and four solutions of quasi-fixed points $\{\frac{1}{2}, \frac{1}{2}, \frac{1+\eta-\sqrt{1-2\eta-3\eta^2}}{2(1+\eta)}, \frac{1+\eta+\sqrt{1-2\eta-3\eta^2}}{2(1+\eta)}\}$. When $\eta > 2/3$, the graph has two real fixed points except 0.5; when $\eta < 1/3$, the graph has two real quasi-fixed points except 0.5; when $1/3 < \eta < 2/3$, the graph has one real fixed point 0.5. For instance, when $\eta = 0.7$, we have two stable fixed points $(0.9071, 0.0929)$ and $(0.0929, 0.9071)$; when $\eta = 0.3$, we have two quasi-fixed points $(0.9071, 0.0929)$ and $(0.0929, 0.9071)$. We observe that both cases have the same strength of potential function $d(\psi)^2 = 0.7/0.3$, though their dynamic characteristics are different.

Based on Property 3, we find that for completely uniform graphs, the maximum multiplicative error and the minimum multiplicative error between two fixed-point messages are reciprocal. In other words, $d(e(x)) = \max e(x)$. Therefore, compared to our uniform distance bound in Corollary 4, we have a tighter distance bound as follows.

Corollary 16. ***(Uniform Distance Bound for Completely Uniform Binary Graph)*** *$\mathbb{G}(\mathbb{V}, \mathbb{E})$ is a completely uniform binary graphical model. The log-distance bound on beliefs at node s is*

$$\sum_{t \in \Gamma_s} \log \frac{d(\psi_{ts})^2 \varepsilon + 1}{d(\psi_{ts})^2 + \varepsilon},$$

where ε should satisfy

$$\log \varepsilon = \max_{(s,p) \in \mathbb{E}} \sum_{t \in \Gamma_s \setminus p} \log \frac{d(\psi_{ts})^2 \varepsilon + 1}{d(\psi_{ts})^2 + \varepsilon}.$$

Proof. $\log \max E_s = \log d(E_s) \leq \sum_{t \in \Gamma_s} \log d(e_{ts}) \leq \sum_{t \in \Gamma_s} \log \frac{d(\psi_{ts})^2 \varepsilon + 1}{d(\psi_{ts})^2 + \varepsilon}$. □

For the uniform graph in Figure 12(c), when $\eta = 0.7$, we have the true log-distance equal to 2.2785, while our previous log-distance bound in Corollary 16 obtains 2.2785, which is exactly equal to the true value, and our *Improved-UDB* in Corollary 7 obtains 2.3318.

2.6.2 True Error-Variation Function

In this section, we characterize the true error-variation function for a completely uniform binary graph. We have the following message updating equation:

$$\begin{pmatrix} m e_1^{out} \\ \\ (1-m) e_2^{out} \end{pmatrix} = \frac{1}{a+b} \begin{pmatrix} a & b \\ \\ b & a \end{pmatrix} \begin{pmatrix} M E_1^{in} \\ \\ (1-M) E_2^{in} \end{pmatrix},$$

where M is the product of fixed-point incoming messages, m is the fixed-point outgoing message, E^{in} represents the product of incoming errors and e^{out} represents the outgoing error. Assuming E^{in} is the same for each node at a level on the Bethe tree, we have the following error equation:

$$\begin{pmatrix} E_1^{out} \\ E_2^{out} \end{pmatrix} = \frac{(aM + b(1-M))^k + (bM + a(1-M))^k}{(aME_1^{in} + b(1-M)E_2^{in})^k + (bME_1^{in} + a(1-M)E_2^{in})^k} \begin{pmatrix} \frac{(aME_1^{in}+b(1-M)E_2^{in})^k}{(aM+b(1-M))^k} \\ \frac{(bME_1^{in}+a(1-M)E_2^{in})^k}{(bM+a(1-M))^k} \end{pmatrix},$$

where E^{out} is the product of outgoing errors flowing into a node at the upper level. When

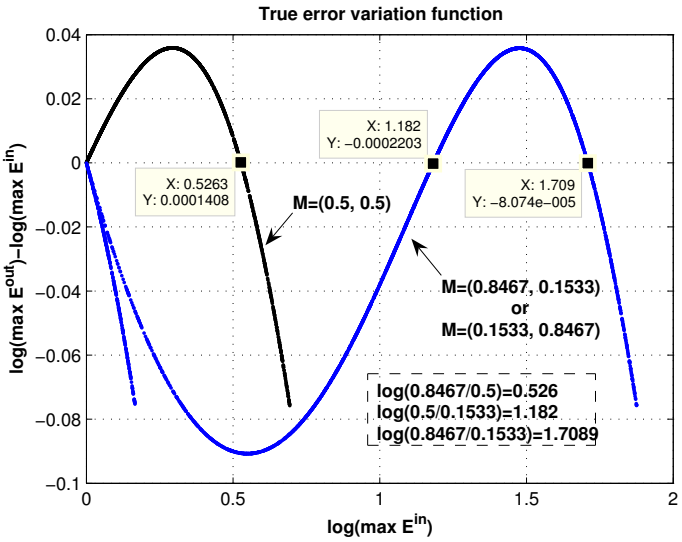


Figure 19. True error variation function when Ms are fixed-point messages for the completely uniform graph in Figure 12(c). $a = 0.7$, $b = 0.3$. The fixed-point messages are: $M = (0.8467, 0.1533)$, $M = (0.1533, 0.8467)$ and $M = (0.5, 0.5)$.

$E_1^{in} > E_2^{in}$ and $a > b$, we have $E_1^{out} > E_2^{out}$. Therefore, letting E denote E_1^{in}, we obtain the true error variation function:

$$G(log(E)) = \log \max E^{out} - \log \max E^{in}$$

$$= \log \left(\frac{(aME + b(1-ME))^k}{(aM + b(1-M))^k} \cdot \frac{(aM + b(1-M))^k + (bM + a(1-M))^k}{(aME + b(1-ME))^k + (bME + a(1-ME))^k}\right) - \log E, \tag{2.22}$$

when $1 < E < 1/M$ and $a > b$.

An example of the true error variation function is illustrated in Figure 19. The curve of $G(\log E)$ varies with the choice of M. When $G(\log E)$ does not cross the horizontal axis except $\log E = 0$, $\log E$ will eventually decrease to zero. In such a case, LBP converges to a unique fixed point. However, when $G(\log E)$ crosses the horizontal axis besides $\log E = 0$, $\log E$ will decrease to stable points of solutions of $G(\log E) = 0$, which implies that the product of the incoming errors at one level equals the product of the incoming errors at its upper level. In other words, errors will not decrease after one LBP update. From the example in Figure 19, we can observe that the zero-crossing points of $\log E$ correspond to the exact log distances between two fixed-point messages. Therefore, our true error function in (Equation 2.22) characterizes the true distance between fixed points, when LBP does not converge. Furthermore, from the curve corresponding to the unstable message $M = (0.5, 0.5)$ in Figure 19, we can find that a small perturbation on the message will easily change it to a stable one. On the other hand, as observed

from curves corresponding to $M = (0.8467, 0.1533)$ or $M = (0.1533, 0.8467)$ in Figure 19, the stable message can still be perturbed into other stable message by a big perturbation.

2.7 Error Bound for the Max-Product Algorithm

Our goal is to apply message error analysis on the max-product algorithm, which is the LBP algorithm for MAP estimation. We present here an upper-bound on max-message errors.

Max-product algorithm works well for MAP estimation, which aims to obtain a configuration that maximizes (Equation 2.1). We can formulate the MAP problem as follows: $\hat{x}_{MAP} = \arg\max_x P(x)$.

For a tree-structured graph, when the MAP configuration is unique, max-product algorithm in (45) can provide max-marginals which correspond to the MAP configuration. However, for graphs with cycles, max-product algorithm can still provide remarkable good approximations. The max-product update rules are:

$$m_{ts}^i(x_s) \propto \max_{x_t} \psi_{ts}(x_t, x_s)\psi_t(x_t) \prod_{u \in \Gamma_t \setminus s} m_{ut}^{i-1}(x_t), \tag{2.23}$$

$$B_t^i(x_t) \propto \psi_t(x_t) \prod_{u \in \Gamma_t} m_{ut}^i(x_t). \tag{2.24}$$

$$B_{ts}^i(x_t, x_s) \propto \psi_{ts}(x_t, x_s)\psi_t(x_t)\psi_s(x_s) \prod_{u \in \Gamma_t \setminus s} m_{ut}^i(x_t) \prod_{p \in \Gamma_s \setminus t} m_{ps}^i(x_s). \tag{2.25}$$

Similarly to sum-product algorithm, we can define message errors and analyze the error propagation of max-product algorithm. By way of p-norm-product belief propagation in (42), we present the following contraction property of max-product message errors.

Corollary 17. $\max_{x_s} e_{ts}(x_s) \leq \min((d(\psi_{ts})d(\psi_{t\star}))^2, d(E_{ts})^2)$, *where* $\psi_{t\star}(x_t) = \max_{x_s} \psi_{ts}(x_t, x_s)$.

Proof. Max norm is equivalent to the p-norm when p tends to ∞, which is defined in the following for an arbitrary function $f(x)$:

$$\max_x f(x) = \|f(x)\|_p = \left(\int f(x)^p dx \right)^{1/p}, p \to \infty.$$

Therefore, the message error for p-norm-product message passing is

$$e_{ts}(x_s) = \frac{(\int (\psi_{ts}(x_t, x_s) M_{ts}(x_t) E_{ts}(x_t))^p dx_t)^{1/p}}{(\int (\psi_{ts}(x_t, x_s) M_{ts}(x_t) E_{ts}(x_t))^p dx_t dx_s)^{1/p}} \times \frac{(\int (\psi_{ts}(x_t, x_s) M_{ts}(x_t))^p dx_t dx_s)^{1/p}}{(\int (\psi_{ts}(x_t, x_s) M_{ts}(x_t))^p dx_t)^{1/p}}.$$

Let us denote $\psi_{t\star}(x_t) = (\int (\psi_{ts}(x_t, x_s))^p dx_s)^{1/p}$. The maximum-error measure is thus as follows:

$$\max_{x_s} e_{ts}(x_s) = \max_{x_s} \left(\frac{\int (\psi_{ts}(x_t, x_s) M_{ts}(x_t) E_{ts}(x_t))^p dx_t}{\int (\psi_{t\star}(x_t) M_{ts}(x_t) E_{ts}(x_t))^p dx_t} \times \frac{\int (\psi_{t\star}(x_t) M_{ts}(x_t))^p dx_t}{\int (\psi_{ts}(x_t, x_s) M_{ts}(x_t))^p dx_t} \right)^{1/p}.$$

By similar approach to the proof for Theorem 2, we have

$$\max_{x_s} e_{ts}(x_s) \leq \left(\frac{d(\psi_{ts})^p d(\psi_{t\star})^p d(E_{ts})^p + 1}{d(\psi_{ts})^p d(\psi_{t\star})^p + d(E_{ts})^p} \right)^{2/p}.$$

When $p \to \infty$, we have $\max_{x_s} e_{ts}(x_s) \leq \min(d(\psi_{ts})^2 d(\psi_{t\star})^2, d(E_{ts})^2)$, where $\psi_{t\star}(x_t) = \max_{x_s} \psi_{ts}(x_t, x_s)$.

□

However, we have not yet obtained distance bounds on beliefs (pseudo-max-marginals) for the max-product algorithm, which remains an open problem and the focus of our future work.

2.8 Conclusion

In this paper, we presented tighter error bounds on Loopy Belief Propagation (LBP) and used these bounds to study the dynamics—error, convergence, accuracy, and scheduling—of the sum-product algorithm. Specifically, we derived tight upper- and lower-bounds on error propagation in synchronous belief networks. We subsequently relied on these bounds to provide uniform and non-uniform distance bounds for the sum-product algorithm. We then used the distance bounds to obtain uniform and non-uniform sufficient conditions for convergence of the sum-product algorithm. We investigated the relation between convergence of LBP with sparsity and walk-summability of graphical models. We also showed that upper-bounds on message errors can be utilized to determine a priority for scheduling in sequential belief propagation. Moreover, we studied the accuracy of the bounds on the sum-product algorithm based on our error bounds. We also presented a case study of LBP by characterizing the dynamics of the sum-product algorithm for completely uniform graphs and analyzed its fixed and quasi-fixed (oscillatory) points. Finally, we provided an upper-bound on the message error in the max-product algorithm, and presented the extension of our approach to deriving distance bounds on beliefs for the max-product algorithm as an open problem.

CHAPTER 3

BELIEF PROPAGATION AND COMPUTER VISION PROBLEMS

In this chapter, we will introduce some applications of belief propagation algorithm on computer vision problems, such as stereo matching and image segmentation. Those problems are usually called labelling problem and modelled by Markov Random Fields (MRFs).

3.1 Introduction to Labelling Problem

Let us first introduce the general MRFs for labelling problems. One MRF model is shown in Figure 20, where white nodes correspond to observed variables, and shadowed nodes correspond to latent variables for labelling. We want to obtain the Maximum A Posteriori (MAP) of the states of latent variables given the states of observations. The MRF model is usually defined as follows:

$$P(X|Y) \propto \prod_{s} \psi_s(x_s, y_s) \prod_{(s,t)\in\mathbb{E}} \psi_{st}(x_s, x_t),$$

where X is the set of labelling variables, Y is the set of observed variables, and $\mathbb{E}$ is the set of edges.

The MRF model can also be expressed using energy function, when the joint probability is defined as $P(X|Y) \propto \exp\{-E(X,Y)\}$, where:

$$E(X|Y) \propto \sum_{s} E_s(x_s, y_s) + \sum_{(s,t)\in\mathbb{E}} E_{st}(x_s, x_t).$$

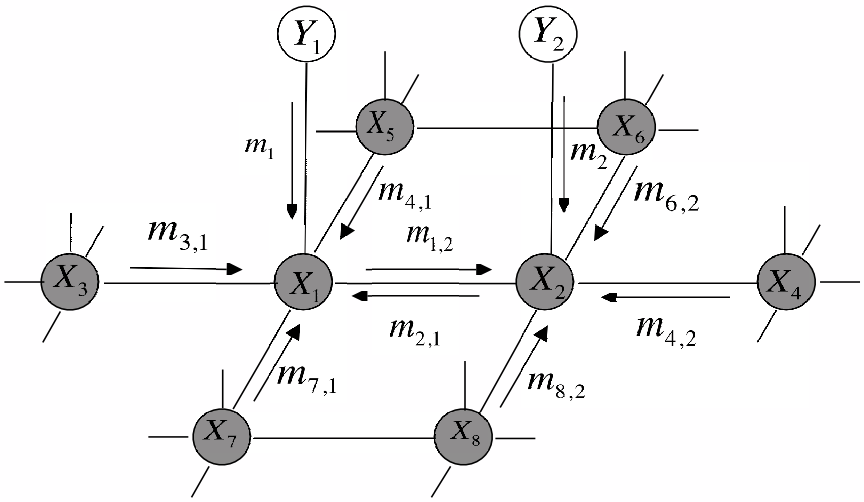


Figure 20. MRF model for image labelling problem.

Specifically, the general framework for labelling problems is summarized as follows:

Objective:$f = \arg\min_{f \in L} E(f)$,

$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in \mathbb{E}} W(f_p, f_q)$.

P: the set of pixels,

$\mathbb{E}$: the set of edges,

L: the set of labels, for instance, disparity, intensity, segment,

f: the labelling function,

$E(f)$: energy function of labelling problem,

$D_p(f_p)$: the cost of assigning certain label to pixel p, called data cost,

$W(f_p, f_q)$: the cost of assigning label to two neighboring pixels, called discontinuity cost.

To solve the energy minimization problem, we will use the min-sum algorithm of belief propagation, which is given as follows:

Message Updates: $m_{st}^{i+1}(x_t) = \kappa + \min_{x_s}(E_{st}(x_s, x_t) + E_s(x_s, y_s) + \sum_{k \in \Gamma_s \setminus t} m_{ks}^i(x_s))$,

Beliefs: $b_s(x_s) = \kappa + E_s(x_s, y_s) + \sum_{k \in \Gamma_s} m_{ks}^T(x_s)$,

where κ is a constant, and T is the final iteration. After the algorithm converges, we have $x_s = \arg\min_{\hat{x}_s} b_s(\hat{x}_s)$,.

3.2 Stereo Matching

We will introduce the application of belief propagation to stereo matching problem using the MRF model in (14). For stereo matching problem, given the left image and right image of one scene captured by a stereo camera, we want to calculate the depth map of the objects in the scene. In other words, given the number of depth levels, we want to label each pixel with a certain level. To accomplish this labelling problem, we usually model the energy function based on several penalty functions. The MRF model presented in (14) is as follows:

$$\text{Objective: } \min_{d_s \in D} \sum_s \rho_d(d_s) + \min_{d_s, d_t} \sum_{(s,t)} \rho_p(d_s, d_t),$$

$$\rho_d(d_s) = \min_{o_s \in O} F(s, d_s, I)(1 - o_s) + \eta_c(o_s) o_s,$$

$$\rho_p(d_s, d_t) = \min_{l_{s,t} \in L} \phi(d_s, d_t)(1 - l_{s,t}) + \gamma(l_{s,t}),$$

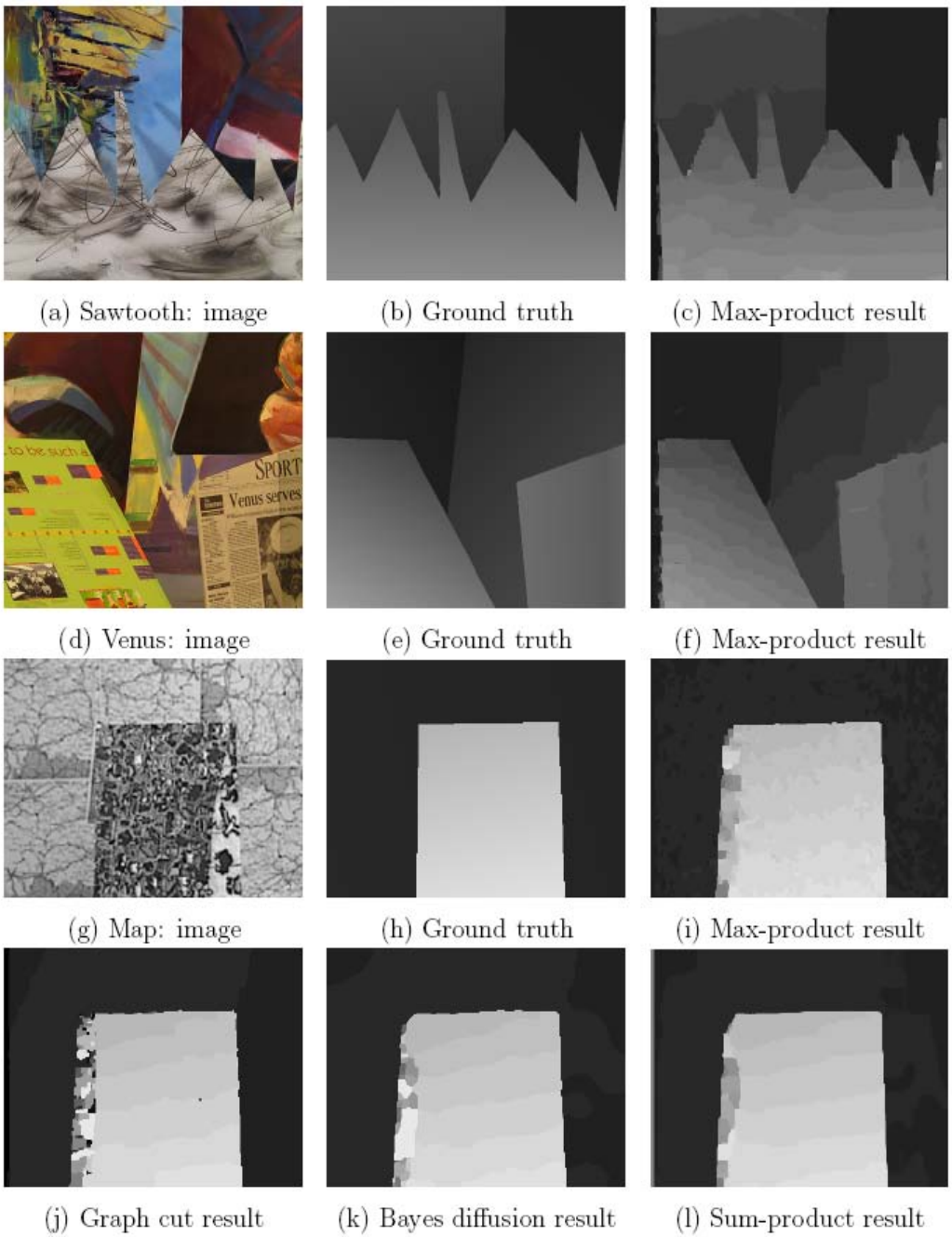


Figure 21. Stereo matching result from (14).

where

D: smooth disparity field of a view,

L: line process to represent the presence of depth discontinuity,

O: binary process to indicate occlusion region,

$F(s, d_s, I)$: matching cost function,

$\eta_c(o_s)$: penalty function for occlusion,

$\phi(d_s, d_t)$: penalty function for non-smoothness when no discontinuity,

$\gamma(l_{s,t})$:penalty function for discontinuity.

For detailed definition of penalty functions, readers are kindly referred to (14). We show some results in Figure 21 using the previous MRF model from (14). We find that both sum-product algorithm and max-product algorithm were implemented, and the former performs better for the Map image in (g). More cues, such as region similarity, motion estimation, help to make the stereo matching model better. (46) added motion cues from stereo video to augment the disparity estimation model.

3.3 Image Segmentation

Image segmentation is a hot topic, on which people presented a great number of algorithms using various approaches. Among image segmentation problems, the simplest case is to separate one object from the background, which is called binary image segmentation. Graph cut works well for this problem. However, to segment several objects, some algorithms are either computationally expensive or fail to give good results. Belief propagation is one of the suc-

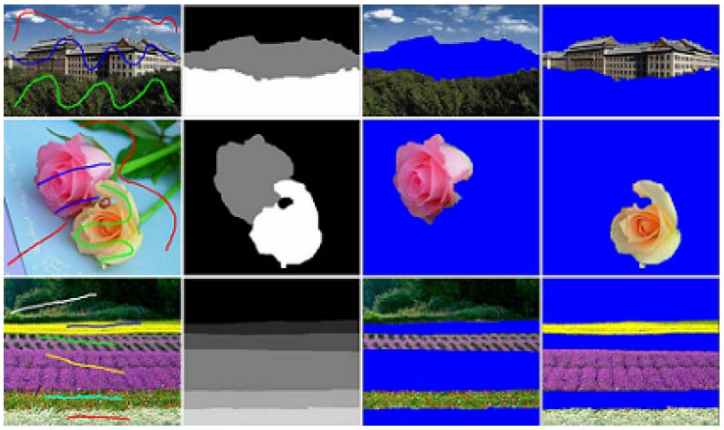


Figure 22. Results of three examples with multi-labels from (47).

cessful algorithms which provide good performance of segmentation. Recently, some interactive segmentation methods have been presented. In this section, we will use (47) to illustrate image segmentation as a labelling problem.

The MRF model used in (47) is as follows:

$$
\begin{aligned}
E(l) &= \sum_{(p,q)\in\mathbb{E}} V(l_p, l_q) + \sum_{p\in P} D_p(l_p), \\
D_p(l_p) &= \begin{cases} \min_j \|C(p) - K^O_{l_p}(j)\|, l_p = 1, ..., n \\ \min_j \|C(p) - K^B(j)\|, l_p = n+1 \end{cases} \\
V(l_p, l_q) &= \begin{cases} 0, \text{if} \quad l_p = l_q \\ d, \text{otherwise} \end{cases}
\end{aligned}
$$

O: point set of foreground object,

B: point set of background object,

$D_p(l_p)$: minimum color distance,

$C(p)$: color of p,

$K_{l_p}^{O}(j)$: mean color of the l_p cluster by K-means,

$K^{B}(j)$: mean color of the background by K-means,

$V(l_p, l_q)$: penalty function for label jumping,

d: standard deviation of all the color distance.

Min-sum algorithm was used to solve this energy minimization problem. We show the results from (47) in Figure 22. We also implement an interactive image segmentation application and apply sum-product belief propagation to solve the labelling problem. We will discuss the affect of $V(l_p, l_q)$ on the convergence of belief propagation in 3.5.1.

3.4 Speed-up Methods for Belief Propagation

Though belief propagation has reduced the computation cost of labelling problem, for large image size, it is still very time consuming. The computation cost is $O(nk^2T)$, where n is the number of pixels, k is the number of labels, and T is the unit processing cost. Some works have been done to speed up belief propagation. (48) presented efficient belief propagation based on some special models. We will introduce some techniques used in (48) here.

Let us recall the energy function.

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in \mathbb{E}} V(f_p, f_q).$$

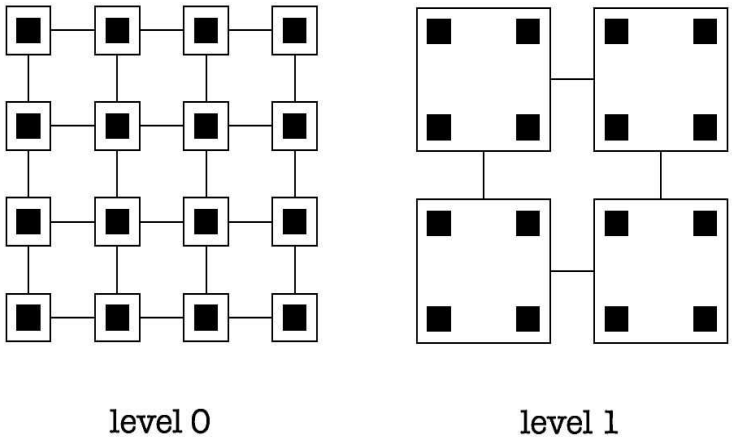


Figure 23. Hierarchical level for multi-grid belief propagation in (48).

Usually, we have $V(f_p, f_q) = V(f_p - f_q)$. In such a case, the message update is as follows:

$$\begin{aligned} m_{pq}^t(f_q) &= \min_{f_p}(V(f_p - f_q) + h(f_p)), \\ h(f_p) &= D_p(f_p) + \sum m_{sp}^{t-1}(f_p). \end{aligned}$$

When we assume Potts model for $V(f_p - f_q)$ as follows:

$$V(x) = \begin{cases} 0, \text{if} \quad x = 0 \\ \\ d, \text{otherwise} \end{cases},$$

message passing can be simplified as follows:

$$m_{pq}^t(f_q) = \min(h(f_q), \min_{f_p} h(f_p) + d).$$

Since $\{h(f_p)\}$ only need to be calculated once, the computation cost reduces to $O(nkT)$. Linear model and quadratic model are also presented in (48).

Another technique is to use multi-grid (hierarchical) belief propagation. (48) used the hierarchy to initialize messages at successively finer levels to reduce message passing iterations. For instance, after messages propagate on Level 1 in Figure 23, they are passed to level 0 and propagate on that level. Reader can refer to (48) for more detail.

3.5 Performance Analysis of Belief Propagation for Image Segmentation

In this chapter, we will illustrate our theoretical results on convergence of sum-product algorithm for binary graph, using an image segmentation application.

3.5.1 An Image Segmentation Application

Our segmentation application is shown in Figure 24. We implemented binary segmentation, which separates one object from the background. We used a simple Markov Random Field model for label random variables and used sum-product algorithm to obtain marginal probabilities.

Specifically, we want to calculate $P(X|Y)$, where X represents label variable and Y represents color feature. We have the following model:

$$
\begin{aligned}
\text{Objective:} \quad & \max P(X|Y) \\
P(X|Y) \quad & \propto P(Y|X)P(X) \\
P(Y|X) \quad & = \prod_{y \in Y} (\sum_{i} \lambda_i (2\pi)^{\frac{-3}{2}} |\Sigma_{i,x}|^{\frac{-1}{2}} \exp\{-\frac{1}{2}(y-\mu_{i,x})' \Sigma_{i,x}^{-1} (y-\mu_{i,x})\}), \\
P(X) \quad & = \prod_{x_i, x_j \in X} \exp\{f(x_i, x_j)\}, \\
f(x_i, x_j) \quad & = \begin{cases} \alpha, & \text{if} \quad x_i = x_j \\ \beta d(y_i, y_j), & \text{otherwise} \end{cases}
\end{aligned}
$$

where X is the set of label variables, $x = 0$ for foreground, $x = 1$ for background, Y is the set of color feature variables, α is the parameter to emphasize uniformity, β is the parameter to emphasize color difference, $d(y_i, y_j)$ is the color difference between pixel i and pixel j, $\mu_{i,x}$ and $\Sigma_{i,x}$ are the mean color and variance for the foreground when $x = 0$ or the background when $x = 1$.

We use Gaussian Mixture Model for data cost function $P(Y|X)$ to model color feature given certain label. In the smoothness function $P(X)$ of our MRF model, parameter α is to make neighbor labelling locally homogeneous, and parameter β is to flip neighbor labelling when the neighboring pixels have big color difference.

After sum-product belief propagation algorithm converges and gives pseudo-marginal probabilities, we assign the label with bigger posterior to each pixel. For example, if $P(x_i = 0|y_i) > P(x_i = 1|y_i)$, we label pixel i as foreground. We also implemented max-product belief propagation to solve this MRF model, and found that it gave quite similar results as sum-product algorithm. The authors in (14) implemented both sum-product algorithm and max-product algorithm for stereo matching application. They observed that in most cases sum-product algorithm gave overly smooth results. But they showed one result for which sum-product gave better results, which is shown in Figure 21 in the thesis. Here, we show segmentation results by sum-product algorithm and verify the theoretical results we have derived in 2.6.

Recall the uniform binary graph in 2.6 of Chapter 2, which has potential function $\psi_{x_i,x_j}(x_i, x_j) = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$. We have proved in Property 3 that for a completely uniform binary graph, when $a > b$, LBP will converge to certain fixed point; when $a < b$, LBP will oscillate between two quasi-fixed points which have reverse states. Our smoothness functions $P_{x_i,x_j}(x_i, x_j) = \begin{pmatrix} \exp\{\alpha\} & \exp\{\beta * d(y_i, y_j)\} \\ \exp\{\beta * d(y_i, y_j)\} & \exp\{\alpha\} \end{pmatrix}$, are not uniform over the graph because of the color difference. However, they are usually distinctly different only for the pixels on edges. Therefore, we will still have segmentation results complying with Property 3, when we choose various α and β

Using our interactive image segmentation application (49), we first need to draw an area that includes the interested object, and enter α and β to set the MRF model, and then we run sum-product algorithm or max-product algorithm to obtain segmentation. From our simulation

results, we observe that for our MRF model when $\alpha > \beta * d(y_i, y_j)$, the segmentation result will converge; otherwise, the segmentation result will oscillate. In other words, when labelling homogeneous is dominant, segmentation will converge to a certain result, see the results in Figure 25 (c), and Figure 26 (c); when color difference is emphasized, the segmentation will oscillate between two reverse labelling results, see Figure 25 (d) and (e), and Figure 26 (d) and (e).

3.6 Conclusion

Belief propagation has been widely used in solving image labelling problems such as image segmentation and stereo matching. In literature, people usually use max-product belief propagation to obtain maximum a posteriori (MAP) estimation based on a Markov Random Fields (MRF) model. We have both implemented sum-product belief propagation algorithm and max-product belief propagation algorithm to solve MAP of labelling variables for image segmentation. We found both algorithms give similar results for our specific MRF model. Furthermore, through our interactive image segmentation application, we verify our theoretical results on pseudo-marginal probabilities for binary graphs.

Figure 24. Image segmentation application.

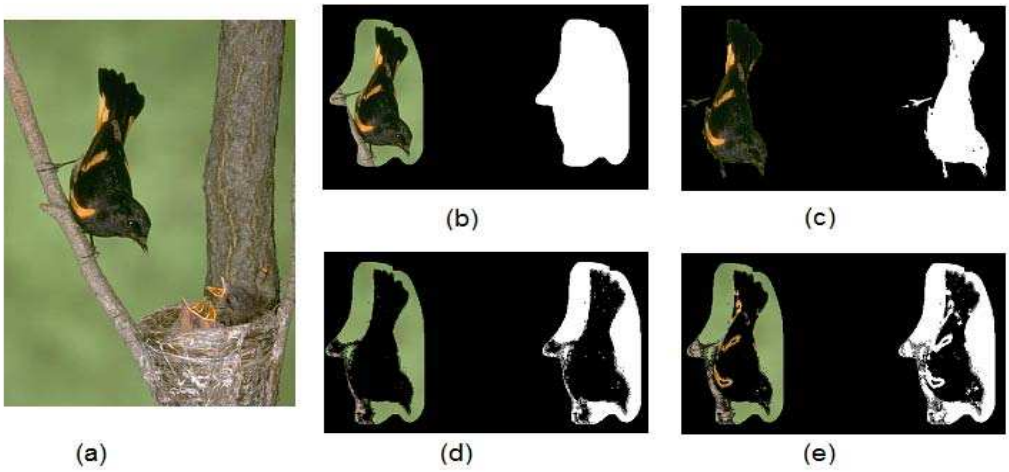


Figure 25. Image segmentation example: bird.(a) Original Image; (b) Selected Area; (c) Segmentation Result for $\alpha = 4$ and $\beta = 1$; (d) Segmentation Result for $\alpha = 0$ and $\beta = 1$ at some iteration i; (e) Segmentation Result for $\alpha = 0$ and $\beta = 1$ at some iteration $i + 1$.

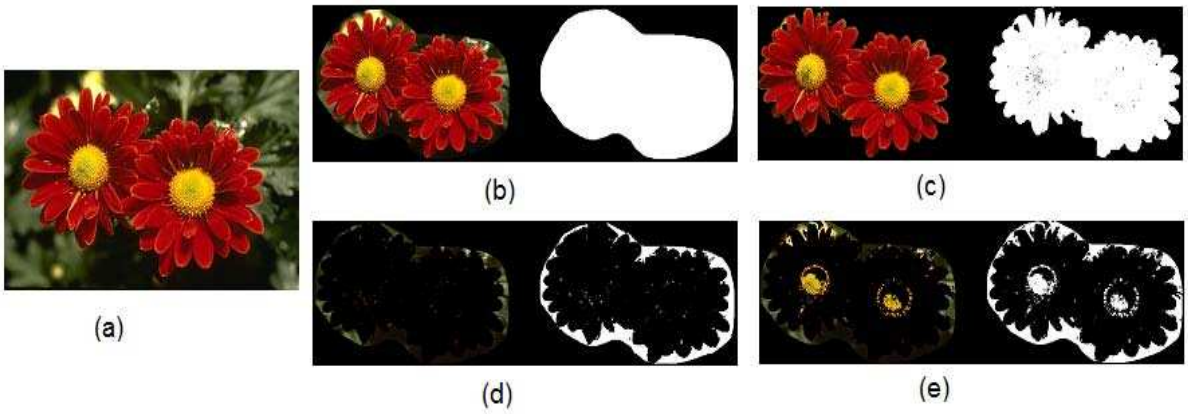


Figure 26. Image segmentation example: flower.(a) Original Image; (b) Selected Area; (c) Segmentation Result for $\alpha = 4$ and $\beta = 1$; (d) Segmentation Result for $\alpha = 0$ and $\beta = 2$ at some iteration i; (e) Segmentation Result for $\alpha = 0$ and $\beta = 2$ at some iteration $i + 1$.

CITED LITERATURE

1. Pearl, J.: Reverend bayes on inference engines: A distributed hierarchical approach. pages 133–136, 1982.

2. Kim, J. H. and Pearl, J.: A computational model for combined causal and diagnostic reasoning in inference systems. pages 190–193, 1983.

3. Tappen, M., Russell, B., and Freeman, W.: Efficient graphical models for processing images. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004.

4. Wang, Y., Loe, K.-F.and Tan, T., and Wu, J.-K.: Spatiotemporal video segmentation based on graphical models. In IEEE Transactions on Image Processing, 2005.

5. Liu, D. and Chen, T.: Object detection in video with graphical models. In 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006.

6. Li, X.: Variational bayesian image processing on stochastic factor graphs. In ICIP 2008. 15th IEEE International Conference on Image Processing, 2008.

7. Ablavsky, V., Thangali, A., and Sclaroff, S.: Layered graphical models for tracking partially-occluded objects. In IEEE Conference on Computer Vision and Pattern Recognition, 2008.

8. Tsechpenakis, G., Wang, J., Mayer, B., and Metaxas, D.: Coupling crfs and deformable models for 3d medical image segmentation. In IEEE 11th International Conference on Computer Vision, 2007.

9. Bishop, C. M.: Pattern Recognition and Machine Learning (Information Science and Statistics). 2006 Springer Science+Business Media, LLC, 2006.

10. Jordan, M.: Learning in Graphical Models. MIT Press, 1999.

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

CITED LITERATURE (Continued)

12. Wainwright, M. J. and Jordan, M. I.: Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning, 1:1–305, 2008.

13. Mceliece, R. J., Mackay, D. J. C., and Cheng, J.-f.: Turbo decoding as an instance of Pearl's belief propagation algorithm. IEEE Journal on Selected Areas in Communications, pages 140–152, 1998.

14. Sun, J., yeung Shum, H., and ning Zheng, N.: Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25:787–800, 2003.

15. Tatikonda, S. and Jordan, M. I.: Loopy belief propogation and gibbs measures. In UAI, pages 493–500, 2002.

16. Heskes, T.: On the uniqueness of loopy belief propagation fixed points. Neural Computation, 16:2379–2413, 2004.

17. Ihler, A. T., III, J. W. F., and Willsky, A. S.: Loopy belief propagation: Convergence and effects of message errors. Journal of Machine Learning Research, 6:905–936, May 2005.

18. Mooij, J. M. and Kappen, H. J.: Sufficient conditions for convergence of the sum-product algorithm. IEEE Transactions on Information Theory, 53(12):4422–4437, December 2007.

19. Georgii, H.-O.: Gibbs Measures and Phase Transitions. Walter de Gruyter and Co., 1988.

20. Weiss, Y.: Correctness of local probability propagation in graphical models with loops. Neural Computation, 12(1):1–41, 2000.

21. Yedidia, J. S., Freeman, W. T., and Weiss, Y.: Constructing free energy approximations and generalized belief propagation algorithms. IEEE Transactions on Information Theory, 51:2282–2312, 2004.

22. Tatikonda, S.: Convergence of the sum-product algorithm. In Information Theory Workshop, 2003. Proceedings. 2003 IEEE, pages 222–225, March-4 April 2003.

23. Salas, J. and Sokal, A.: Absence of phase transition for antiferromagnetic potts models via the dobrushin uniqueness theorem. Journal of Statistical Physics, 86:551–579, 1997.

CITED LITERATURE (Continued)

24. Taga, N. and Mase, S.: Error bounds between marginal probabilities and beliefs of loopy belief propagation algorithm. In MICAI, pages 186–196, 2006.

25. Ihler, A.: Accuracy bounds for belief propagation. In Proceedings of UAI 2007, July 2007.

26. Weitz, D.: Counting independent sets up to the tree threshold. In In STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, pages 140–149. ACM Press, 2006.

27. Mooij, J. M. and Kappen, H. J.: Bounds on marginal probability distributions. In Advances in Neural Information Processing Systems 21 (NIPS*2008).

28. Taga, N. and Mase, S.: On the convergence of loopy belief propagation algorithm for different update rules. IEICE Transactions, 89-A(2):575–582, 2006.

29. Elidan, G., Mcgraw, I., and Koller, D.: Residual belief propagation: informed scheduling for asynchronous message passing. In Uncertainty in Artificial Intellignece, 2006.

30. Sutton, C. and Mccallum, A.: Improved dynamic schedules for belief propagation. In Conference on Uncertainty in Artificial Intelligence (UAI), 2007.

31. Shi, X., Schonfeld, D., and Tuninetti, D.: Message error propagation for belief propagation. In 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2010.

32. Jordan, M. I.: Graphical models. In Statistical Science (Special Issue on Bayesian Statistics), 2004.

33. Frey, B. and Jojic, N.: A comparison of algorithms for inference and learning in probabilistic graphical models. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005.

34. Murphy, K., Weiss, Y., and Jordan, M.: Loopy belief propagation for approximate inference: An empirical study. In Proceedings of Uncertainty in AI, 1999.

35. Ihler, Fisher, and Willsky: Message errors in belief propagation. In Neural Infor-mation Processing Systems, 2004.

36. Weiss, Y.: Correctness of belief propagation in gaussian graphical models of arbitrary topology. In Neural Computation, 2001.

CITED LITERATURE (Continued)

37. Weiss, Y.: Belief propagation and revision in networks with loops. Technical report.

38. Wainwright, M. J., Jaakkola, T. S., and Willsky, A. S.: Tree-reweighted belief propagation algorithms and approximate ml estimation by pseudo-moment matching. In Workshop on Artificial Intelligence and Statistics 2003, 2003.

39. Yedidia, J., Freeman, W., and Weiss, Y.: Generalized belief propagation. In Advances in neural information processing systems, 2001.

40. Yedidia, J., Freeman, W., and Weiss, Y.: Bethe free energy, kikuchi approximations, and belief propagation algorithms. In Advances in neural information processing systems, 2001.

41. Teh, Y. and Welling, M.: The unified propagation and scaling algorithm. In Advances in neural information processing systems, 2002.

42. Hazan, T. and Shashua, A.: Norm-product belief propagation: Primal-dual message-passing for approximate inference. CoRR, abs/0903.3127, 2009.

43. Wiberg, N.: Codes and Decoding on General Graphs. Ph.D. dessertation, Linkoping Univ., 1996.

44. Malioutov, D. M., Johnson, J. K., and Willsky, A. S.: Walk-sums and belief propagation in gaussian graphical models. J. Mach. Learn. Res., 7:2031–2064, 2006.

45. Wainwright, M., Jaakkola, T., and Willsky, A.: Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. Statistics and Computing, 14(2):143–166, 2004.

46. Isard, M. and Maccormick, J.: Dense motion and disparity estimation via loopy belief propagation. Technical report, 2005.

47. Xiang, S., Nie, F., Zhang, C., and Zhang, C.: Interactive visual object extraction based on belief propagation. In Advances in Multimedia Modeling, volume 4351 of Lecture Notes in Computer Science, pages 24–33. 2006.

48. Felzenszwalb, P. and Huttenlocher, D.: Efficient belief propagation for early vision. In International Journal of Computer Vision, 2006.

CITED LITERATURE (Continued)

49. Shi, X.: Image segmentation application based on markov random fields and belief propagation. http://sites.google.com/site/xiangqiongshi.

50. Sun, J., Zheng, N.-N., and Shum, H.-Y.: Stereo matching using belief propagation. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003.

51. Freeman, W. and Weiss, Y.: On the fixed points of the max-product algorithm. In IEEE Trans. on Info. Theory, 2000.

52. Mooij, J. M. and Kappen, H. J.: On the properties of the Bethe approximation and loopy belief propagation on binary networks. Journal of Statistical Mechanics: Theory and Experiment, 2005(11):P11012, 2005.

53. Wainwright, M. J., Jaakkola, T. S., and Willsky, A. S.: Tree-based reparameterization framework for analysis of sum-product and related algorithms. IEEE Transactions on Information Theory, 49(5):1120–1146, 2003.

VITA

NAME Xiangqiong Shi

EDUCATION Ph.D., Electrical and Computer Engineering,
University of Illinois at Chicago, Chicago, Illinois, 60607

M.S., B.S., Electrical and Computer Engineering,
Shanghai Jiao Tong University, Shanghai, China, 200240

ACADEMIC **University of Illinois at Chicago**
Multimedia Communications Laboratory, Dept. of ECE

Performance Analysis of Message Passing Algorithms for Computer Vision Problems 08/2010 - 06/2011
- Analyzing the performance of message passing algorithms for image segmentation.

Performance Analysis of Loopy Belief Propagation Algorithm 08/2008 - 05/2010
- Presented novel distance bounds between multiple fixed-point solutions.
- Proposed uniform and non-uniform sufficient convergence conditions.
- Improved existing distance bounds between beliefs and true marginal probabilities.

Motion Trajectory Indexing and Retrieval Based on Multi-linear Analysis 08/2009 - 12/2009
- Provided flexibility of query dimension to efficiently index and retrieve multi-camera based trajectories.

Discrete Wavelet Transform Approach for Blind Adaptive Filtering (DWT-BAF) of Image from Unknown Noise Spring 2007
- Enhanced non-stationarity and SNR of images with unknown noises by utilizing DWT-BAF.

Video Classification Based on Cross-Correlation Analysis 08/2006 - 12/2007
- Presented a novel statistical cross-correlation analysis for video classification by

VITA (Continued)

modeling time-variant cross correlation as random process.
- Reduced dramatically the search time for the maximum cross correlation by randomly sampling the correlation function and monitoring the likelihood of a higher value using sequential hypothesis testing.

Shanghai Jiao Tong University

Deadlock Free Scheduling for Flexible Manufacturing Systems (FMSs) 09/2003 - 02/2006
- Solved the NP-hard deadlock-free scheduling problems by a filtered beam search method and used siphon technology to prevent deadlock.

Improvements on Simulation Software for FMSs 09/2002 - 06/2003
- Improved the graphical user interface (GUI) and database design of FMSs simulation software.

PUBLICATIONS

Journal Papers

- **Xiangqiong Shi**, Dan Schonfeld and Daniela Tuninetti, "Performance Analysis of Loopy Belief Propagation", *submitted to Journal of Machine Learning Research*, http://arxiv.org/abs/1009.2305.

- **Xiangqiong Shi** and Dan Schonfeld, "Cross-Correlation Analysis for Video Classification and Mining Based on Statistical Methods", *submitted to IEEE Transactions on Image Processing.*

Conference Papers

- Qun Li, **Xiangqiong Shi** and Dan Schonfeld, "A GENERAL FRAMEWORK FOR ROBUST HOSVD-BASED INDEXING AND RETRIEVAL WITH HIGH-ORDER TENSOR DATA", *to appear in ICASSP 2011.*

- Qun Li, **Xiangqiong Shi** and Dan Schonfeld, "Robust HOSVD-based multi-camera motion trajectory indexing and retrieval", *to appear in IS&T SPIE Electronic Imaging 2011.*

- **Xiangqiong Shi**, Dan Schonfeld and Daniela Tuninetti, "Message Error Analysis of Loopy Belief Propagation", *IEEE International Conference on Acoustics Speech and Signal Processing*, 2010, Page(s): 2078 - 2081.

VITA (Continued)

- **Xiangqiong Shi** and Dan Schonfeld, "Video Classification and Mining Based on Statistical Methods for Cross-Correlation Analysis", *IEEE/SP Workshop on Statistical Signal Processing*, 2007, Page(s):586 - 590.

- **Xiangqiong Shi** and Zhiming Wu, "Deadlock-Free Scheduling Method for FMSs Using Beam Search", *2005 IEEE International Conference on Systems, Man and Cybernetics*,Volume: 2, Page(s): 1188- 1193.

EXPERIENCE

Visual Computing and Ubiquitous Imaging Team, Research Intern 10/2010 - 05/2011
- Improved performance of 2D image stitching for mobile Panorama application and Robust Instant Cloning application on Nokia smartphone.

Technicolor, 3D Video Compression Group, Summer Intern 05/2010 - 08/2010
- Improved the subjective quality of 3D video with high motion, blur, or coding artifacts by adding comfort noise to decoder.

Motorola Inc., Image and Video Communication Group, Summer Intern 05/2008 - 08/2008
- Implemented real-time view synthesis based on Bumble Bee camera (Tool: OpenCV and OpenGL).

Gelber Group, LLC, Research Assistant 09/2006-08/2007
- Realized real-time financial forecasting based on statistical signal processing.

SKILL

Programming Languages: Matlab, C/C++, VB, MASM, SQL, Qt.
Operating System: Windows, Mac OS, Linux.

AWARDS

Student Travel Award, UIC, 2010
University Fellowship, (offered up to 15% of total fall enrollment), University of Illinois at Chicago, 2006-2007, 2009-2010 Academic Years
National Excellence Scholarship, Shanghai Committee of Education, 2004-2005 Academic Year
Shanghai Excellent Graduate, Shanghai Committee of Education, June 2003