



IGNITED MINDS
Journals

*International Journal of
Information Technology
and Management*

*Vol. VI, Issue No. II,
May-2014, ISSN 2249-4510*

**A RESEARCH ABOUT LOW-COMPLEXITY
MESSAGE-PASSING ALGORITHMS FOR
DISPERSED COMPUTATION**

AN
INTERNATIONALLY
INDEXED PEER
REVIEWED &
REFEREED JOURNAL

A Research about Low-Complexity Message-Passing Algorithms for Dispersed Computation

Kanika Mehra

Research Scholar, Jodhpur National University, Rajasthan

Abstract – Central to many statistical inference problems is the computation of some quantities defined over variables that can be fruitfully modeled in terms of graphs. Examples of such quantities include marginal distributions over graphical models and empirical average of observations over sensor networks. For practical purposes, distributed message-passing algorithms are well suited to deal with such problems. In particular, the computation is broken down into pieces and distributed among different nodes. Following some local computations, the intermediate results are shared among neighboring nodes via the so called messages.

The process is repeated until the desired quantity is obtained. These distributed inference algorithms have two primary aspects: statistical properties, in which characterize how mathematically sound an algorithm is, and computational complexity that describes the efficiency of a particular algorithm. In this thesis, we propose low-complexity (efficient), message-passing algorithms as alternatives to some well-known inference problems while providing rigorous mathematical analysis of their performances. These problems include the computation of the marginal distribution via belief propagation for discrete as well as continuous random variables, and the computation of the average of distributed observations in a noisy sensor network via gossip-type algorithms.

INTRODUCTION

Many practical systems are affected by random variations such as observational error, communication noise, model uncertainty, and so on. Examples of such systems can be found in different fields including telecommunications, signal and image processing, computer vision, machine learning, finance, bioinformatics, among others. The purpose of statistical inference is to draw conclusions based on data arising from such systems. To characterize their behavior, the stochastic systems are first modeled, which means the relationship between random variables are formalized (typically via a set of parameters). Then, based on some realizations of the data, an estimate of the parameters that best describe the system are determined. Inference problems have two fundamental aspects: statistical properties that characterize the behavior of an algorithm (such as consistency, rate of convergence, etc.) as well as computational complexity that describes the efficiency of a particular algorithm.

In this dissertation, our primary focus will be on the latter. We will provide efficient, low-complexity solutions to some algorithms with wide range of applications, while analyzing their statistical behavior. Due to their nature, distributed computation is well suited for problems on graphs. The existence of the required infrastructure for the centralized computation is not necessarily guaranteed in the sensor network

application. Also, as will become clear in the following chapter, a form of the divide and conquer algorithm on graphs can significantly increases the efficiency of the marginalization algorithm. The general idea behind distributed computation is to break down the calculation and distribute the pieces among different nodes. Then following some local computations, every node shares its information with other nodes by passing the so called messages to its neighbors along the edges of the graph. The received messages constitute the intermediate results and could be of the form of a d-dimensional vector, a real-valued function, or a noise-corrupted signal. Every node, uses the received messages in order to update an estimate of the desired quantity. Of interested to us are the statistical properties of these estimates, as well as the efficiency of the local computations.

But first we need to formally introduce two popular mathematical models for distributed computations, graphical models and sensor networks.

GRAPHICAL MODELS

By bringing together graph theory and probability theory, graphical models provide a general and flexible framework for describing statistical interactions among random variables. A broad range of fields—among them statistical signal processing, computer vision, coding theory, bioinformatics, natural language processing—involve problems with

large number of random variables that can be fruitfully formulated in terms of graphs. A few of such examples include, positioning and tracking problems that can be modeled by chains, low-density parity-check codes that can be described by factor graphs, some image processing as well as vision problems that can be formulated by two dimensional grids, and text processing that can be modeled by Bayesian networks.

The statistical dependencies among random variables are encoded by the structure of the graph. This is accomplished by first mapping the random variables to the nodes of the graph and then factorizing the joint probability distribution over local functions defined on the graph. There are two common families of graphical models, directed models also known as Bayesian networks and undirected models also known as Markov random fields. See Figure for an illustration of these two models. Bayesian networks are defined on directed graphs that is every edge consists of an ordered pair of nodes. It can be shown that if the directed graphical model is acyclic (i.e. does not include any directed cycles), the joint probability density becomes equal to the product of a collection of local conditional probability densities. In contrast to Bayesian networks, Markov random fields are defined on undirected graphs in which each edge is an unordered pair of nodes. As we will see in the next chapter, probability densities over such graphs also have local factorization, in particular over positive functions defined on the fully connected sub-graphs. Even though these two models are closely related, they make different assertions of conditional independence between random variables. Therefore, depending on the application one might be a better option than the other. For instance, a Markov random field could a better choice for problems that involve random variables with no clear causal relationships.

BACKGROUND

In this chapter, we will review some of the mathematical concepts that will be employed throughout this thesis. These will include graphical models, the belief propagation algorithm, stochastic approximation, and gossip algorithms. We begin by reviewing the basic concepts of undirected graphical models as well as the belief propagation algorithm.

Undirected Graphical Models - An undirected graphical model, also known as a Markov random field, defines a family of joint probability distributions over the random vector X . These probability distributions are assumed to be absolutely continuous with respect to a given measure μ , typically the counting measure for the case of discrete random variables or the Lebesgue measure for continuous random variables. The structure of the graph describes the statistical dependencies among the different random variables—in particular via the notion of graph separation. For a set A , define the sub-vector $X_A := \{X_v | v \in A\}$, similarly defined for sets B and C . We say that:

“the random vector X_A is independent of X_C given X_B ”

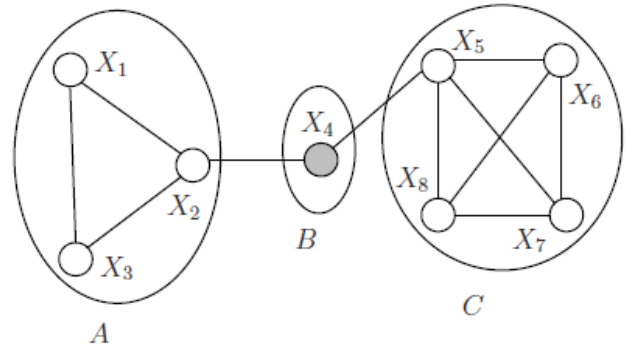


Figure: The notion of graph separation. Set B separates sets A and C. More precisely, every path from a node in set A to a node in set C goes through a node in set B. Accordingly, the set of random variables $\{X_1, X_2, X_3\}$ is independent of the set $\{X_5, X_6, X_7, X_8\}$ given X_4 .

if and only if set B separates sets A and C . More specifically, every path from a node in set A to a node in set C goes through the set B . See Figure for an illustration of the graph separation notion. Every graph G , induces a set of such conditional independence statements also known as Markov properties. Moreover, it should be noted that such Markov properties must hold for all members of the family of the probability distributions associated with G .

Therefore, the family of acceptable probability distributions is constrained by the set of all Markov properties and thus have a particular form of factorization. In order to make this statement precise, we need to define the notion of cliques.

Pairwise Markov Random Fields - Many applications involve pairwise interactions among nodes. In those instances, since cliques consist of the set of all vertices V together with the set of all edges E , the general factorization takes the special form

$$P(x_1, x_2, \dots, x_n) \propto \prod_{u \in V} \psi_u(x_u) \prod_{(u,v) \in E} \psi_{uv}(x_u, x_v),$$

BELIEF PROPAGATION ALGORITHM

Belief propagation is an iterative algorithm consisting of a set of local message-passing rounds, for computing either exact or approximate marginal distributions defined on a graph. As discussed in the previous section, if approached naively, computing marginal distributions is intractable. However, exploiting the particular form of the factorization induced by the graph, BP provides a fast and efficient method for dealing with this problem. In this section, we will provide an overview of the BP algorithm over the sum-product semi-ring.

In order to formally introduce the message-passing updates, we first need to define the notion of the factor graph. A factor graph is a graphical representation of factorizations.

In precise terms, it is a bipartite graph $\mathcal{G}' := (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}')$, consisting of variable nodes indexed by $\mathcal{V}_1 := \{1, 2, \dots, n\}$ and factor nodes $\mathcal{V}_2 := \{l \mid l \in C\}$. Moreover, an edge connects the variable node i to the factor l (i.e. $(i, l) \in \mathcal{E}'$) if and only if i belongs to the local function $\psi_l(\cdot)$.

STOCHASTIC BELIEF PROPAGATION

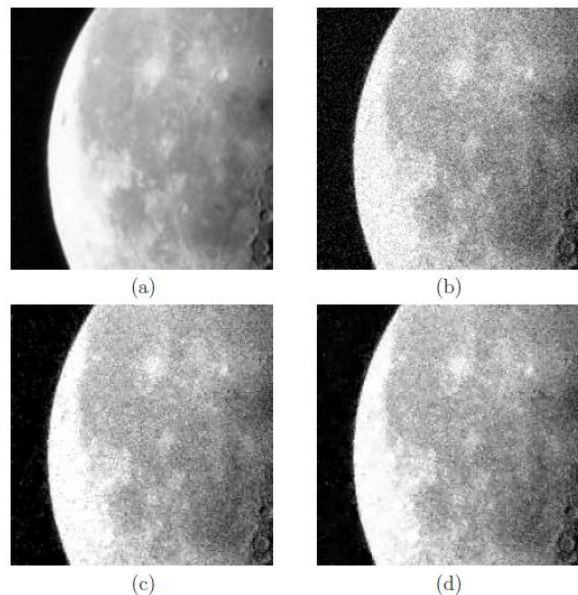
In this chapter, we focus on the problem of implementing the belief propagation message-passing for high-dimensional discrete random variables. In many applications of BP, the messages themselves are high-dimensional in nature, either due to discrete random variables with a very large number of possible realizations d (which will be referred to as the number of states), due to factor nodes with high degree, or due to continuous random variables that are discretized. Examples of such problems include disparity estimation in computer vision, tracking problems in sensor networks, and error-control decoding. For such problems, it may be expensive to compute and/or store the messages, and as a consequence, BP may run slowly, and be limited to small-scale instances. Motivated by this challenge, researchers have studied a variety of techniques to reduce the complexity of BP in different applications. At the core of the BP message-passing is a matrix-vector multiplication, with complexity scaling quadratically in the number of states d . Certain graphical models have special structures that can be exploited so as to reduce this complexity. For instance, in applications to the decoding of low-density parity-check codes in channel coding, the complexity of message-passing, if performed naively, would scale exponentially in the factor degrees. However, a clever use of the fast Fourier transform over $GF(2^r)$ reduces this complexity to linear in the factor degrees. Other problems arising in computer vision involve pairwise factors with a circulant structure for which the fast Fourier transform can also reduce complexity. Similarly, computation can be accelerated by exploiting symmetry in factors, or additional factorization properties of the distribution.

APPLICATIONS IN IMAGE PROCESSING AND COMPUTER VISION

In our next set of experiments, we study the SBP on some larger scale graphs and more challenging problem instances, with applications to image processing and computer vision. Message-passing algorithms can be used for image denoising, in particular, on a two dimensional square grid where every node corresponds to a pixel. Running the BP

algorithm on the graph, one can obtain (approximations to) the most likely value of every pixel based on the noisy observations. In this experiment, we consider a 200×200 image with $d = 256$ gray-scale levels, as shown in Figure (a). We then contaminate every pixel with an independent Gaussian random variable with standard deviation $\sigma = 0.1$, as shown in Figure (b). Enforcing the potts model with smoothness parameter $\beta = 0.05$ as the edge potential, we run BP and SBP for the total of $t = 5$ and $t = 100$ iterations, respectively, to obtain the refined images (see panels (c) and (d), respectively, in Figure). Figure illustrates the mean-squared error versus the running time for both BP and SBP denoising.

As one can observe, despite smaller jumps in the error reduction, the per-iteration running time of SBP is substantially lower than BP. Overall, SBP has done a marginally better job than BP in a substantially shorter amount of time in this instance. Note that the purpose of this experiment is not to analyze the potential of SBP (or for that matter BP) in image denoising, but to rather observe their relative performances and computational complexities.



DESCRIPTION OF THE SOSMP ALGORITHM

In this section, we turn to the description of the SOSMP algorithm. The SOSMP algorithm involves a combination of the orthogonal series expansion techniques and stochastic methods previously described. Any particular version of the algorithm is

specified by the choice of basic functions $\{\phi_j\}_{j=1}^{\infty}$ and two positive integers: the number of coefficients r that are maintained, and the number of samples k used in the stochastic update. Prior to running the algorithm, for each directed edge $(u \rightarrow v)$, we pre-compute the inner products

$$\gamma_{uv;j}(x_u) := \int_{\mathcal{X}} \underbrace{\Gamma_{uv}(x_v, x_u) \phi_j(x_v)}_{(\Gamma_{uv}(\cdot, x_u), \phi_j(\cdot))} dx_v, \quad \text{for } j = 1, \dots, r.$$

When ψ_{vu} is a symmetric and positive semi definite kernel function, these inner products have an explicit and simple representation in terms of its Mercer Eigen decomposition. In the general setting, these r inner products can be computed via standard numerical integration techniques. Note that this is a fixed (one-time) cost prior to running the algorithm.

Moving beyond simulated problems, we conclude by showing the SOSMP algorithm in application to a larger scale problem that arises in computer vision—namely, that of optical flow estimation. In this problem, the input data are two successive frames of a video sequence. We model each frame as a collection of pixels arrayed over $\sqrt{n} \times \sqrt{n}$ grid, and measured intensity values at each pixel location of the form $\{I(i, j), I'(i, j)\}_{i,j=1}^{\sqrt{n}}$. Our goal is to estimate a 2-dimensional motion vector $x_u = (x_{u,1}, x_{u,2})$ that captures the local motion at each pixel $u = (i, j), i, j = 1, 2, \dots, \sqrt{n}$ of the image sequence.

EFFICIENT DISTRIBUTED AVERAGING

Given the communication randomness, any algorithm is necessarily stochastic, and the corresponding sequence of random variables can be analyzed in various ways. The simplest question to ask is whether the algorithm is consistent—that is, does it compute an approximate average or achieve consensus in an asymptotic sense for a given fixed graph? A more refined analysis seeks to provide information about this convergence rate. In this chapter, we do so by posing the following question: for a given algorithm, how does number of iterations required to compute the average to within ϵ -accuracy scale as a function of the graph topology and number of nodes n ? For obvious reasons, we refer to this as the network scaling of an algorithm, and we are interested in finding an algorithm that has optimal scaling law.

The issue of network scaling has been studied by a number of authors in the noiseless setting, in which the communication between nodes is perfect. Of particular relevance here is the work of Benezit et al., who in the case of perfect communication provided a scheme that has essentially optimal message scaling law for random geometric graphs.

In order to demonstrate the effectiveness of the proposed algorithm, we conducted a set of simulations. More specifically, we apply the proposed algorithm to four nearest-neighbor square grids of different sizes. We initially generate the data $\theta_i^0, i = 1, 2, \dots, n$ as random $N(1, 1)$ variables

and fix them throughout the simulation. So for each run of the algorithm the initial data is fixed. In implementing the algorithm, we adopt $\sigma^2 = 1$ as the channel noise variance, and we set the tolerance parameter $\delta = 0.1$, leading to the step size $\epsilon^\tau = 1/(10 + \tau)$.

We estimated the mean-squared error, defined in equation, by taking the average over 50 sample paths. As discussed in every outer phase update requires $M = \mathcal{O}(\sqrt{n})$ time steps.

CONCLUSION

In this paper, we have developed and analyzed a new and low-complexity alternative to the BP message-passing. The SBP algorithm has per iteration computational complexity that scales linearly in the state dimension d , as opposed to the quadratic dependence of BP, and a communication cost of $\log_2 d$ bits per edge and iteration, as opposed to $d-1$ real numbers for standard BP message updates. Stochastic belief propagation is also easy to implement, requiring only random number generation and the usual distributed updates of a message-passing algorithm. Our main contribution was to prove a number of theoretical guarantees for the SBP message updates, including convergence for any tree-structured problem, as well as for general graphs for which the ordinary BP message update satisfies a suitable contraction condition. In addition, we provided non-asymptotic upper bounds on the SBP error, both in expectation and in high probability.

Our work leaves open a number of interesting questions. First, although we have focused exclusively on models with pairwise interactions, it should be possible to develop forms of SOSMP for higher-order factor graphs. Second, the bulk of our analysis was performed under a type of contractivity condition, as has been used in past work on convergence of the standard BP updates. However, we suspect that this condition might be somewhat relaxed, and doing so would demonstrate applicability of the SOSMP algorithm to a larger class of graphical models.

There are various issues left open in this work. First, while the AWGN model is more realistic than noiseless communication, many channels in wireless networks may be more complicated, for instance involving fading, interference and other types of memory. In principle, our algorithm could be applied to such channels and networks, but its behavior and associated convergence rates remain to be analyzed. In a separate direction, it is also worth noting that gossip-type algorithms can be used to solve other problems, such as distributed optimization problems and kernel density estimation. Complexity reduction and studying the issue of

optimal network scaling for such problems is also of interest.

REFERENCE

- Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, May 2012.
- Benveniste, M. Metivier, and P. Priouret. *Stochastic Approximations and Adaptive Algorithms*. Springer-Verlag New York, Inc., New York, 1990.
- G. Dimakis, A. Sarwate, and M. J. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Trans. Signal Processing*, 53:1205–1216, March 2008.
- D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *18th Annual Conference on Learning Theory (COLT)*, July 2005.
- F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli. Order-optimal consensus through randomized path averaging. *IEEE Transaction on Information Theory*, 56(10):5150–5167, 2010.
- G. Boccignone, A. Marcelli, P. Napoletano, and M. Ferraro. Motion estimation via belief propagation. In *Proceedings of the International Conference on Image Analysis and Processing*, 2007.
- J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Stat. Soc. Series B*, 36:192–236, 1974.
- J. Coughlan and H. Shen. Dynamic quantization for belief propagation in sparse spaces. *Computer Vision and Image Understanding*, 106(1):47–58, 2007.
- J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23:462–466, 1952.
- K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In *Proceedings Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, 2009.
- M. H. deGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, March 1974.
- P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *Ann. Applied Probability*, 1:36–61, 1991.
- R. Diestel. *Graph Theory*. Springer-Verlag, New York, 2000.
- R. P. Agarwal, M. Meehan, and D. O'Regan. *Fixed Point Theory and Applications*. Cambridge University Press, 2004.