RUMOR SOURCE DETECTION: A PROBABILISTIC PERSPECTIVE

Ting-Han Fan and I-Hsiang Wang

Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan
Email: {b02901062,ihwang}@ntu.edu.tw

ABSTRACT

In this paper we consider the problem of rumor source detection in a network. Our main contribution is an efficient Belief-Propagation-based (BP) algorithm to compute the joint likelihood function of the source location and the spreading time for the general continuous-time Susceptible-Infected epidemic model on trees. As a result, many probabilistic detection algorithms, including the joint maximum likelihood estimator, can be implemented with time complexity being nearly linear in the product of the size of the graph and the effective range of the spreading time. This is in sharp contrast to the widely employed discrete-time epidemic models where the complexity in computing the likelihood function of the source location is exponential. To extend the BP algorithm to general graphs, we propose a "Gamma Generated Tree" heuristic to convert the original graph to a tree with heterogeneous infection rates over edges. Compared to state-of-the-art methods, simulation results show that our algorithm provides better estimates of the source when the graph topology is similar to trees. As a byproduct, the spreading time can also be estimated, which is useful in some applications.

***Index Terms*—** Rumor Source Detection, Belief Propagation, Maximum Likelihood Estimation, Social Networks

1. INTRODUCTION

Rumor source detection is an important problem in tracking abnormal activities in networks, including rumors [1], diseases [2] and computer virus [3]. Shah and Zaman [4, 5] pioneered in the single source detection for the Susceptible-Infected (SI) model on trees. They proposed the notion of Rumor Center as the detected source and triggered plenty of extensions such as locating double sources [6], boosting with multiple observations [7] or temporal information [8], boosting by finding confidence sets of sources [9], and evaluating rumor centrality under sampling [10]. Besides, in [11,12], the notion of Jordan Center was proposed to detect the source, which was proved to be optimal for a sample-path-based approach. It also helped design algorithms for multiple sources detections [13, 14]. Finally, the Dynamic Message Passing algorithm proposed in [15,16] utilized cavity messages to approximate likelihood functions.

However, most previous works employ discrete-time epidemic models, in which computing the maximum likelihood estimator (MLE) of the source requires enumerating over all possible orders of the infection sequences, making the time complexity exponential in the size of the graph [4,11]. Hence, most previous works avoid deriving the exact MLE and employ other heuristics, as seen above. Meanwhile, to fully utilize the information about rumor spreading contained in the epidemic model, it is critical to overcome the computational barrier of the likelihood function.

The key observation made in this paper is that, for a continuous-time epidemic model, once we jointly consider the spreading time t and the source location s, the computation of the *joint* likelihood function can be done in polynomial time. This is in sharp contrast to the discrete-time models. To elaborate, let us consider the case where the underlying network is a tree, and assume that the infection graph G_I is broken into two infection sub-trees $\mathscr{T}_I^{(1)}$ and $\mathscr{T}_I^{(2)}$, both rooted at the source s. Due to the mutual independence of the spreading times across all edges, the joint likelihood can be factorized into two parts: $\mathbb{P}(G_I|s,t) = \mathbb{P}(\mathscr{T}_I^{(1)}|s,t)\mathbb{P}(\mathscr{T}_I^{(2)}|s,t)$. Thereby, recursively factorizing probabilities leads to a Belief Propagation [17] algorithm for computing the joint likelihood, which reduces time complexity.

Built upon the above key observation, our main contribution is an efficient Belief Propagation (BP) algorithm to compute the joint likelihood function of the source and spreading time for general continuous-time SI model on trees. As a result, many probabilistic detection algorithms can be implemented efficiently. In particular, the joint MLE is attained with time complexity $O(nL\log L)$, where n is the number of infected nodes and $[0, L]$ is the effective range of time considered in finding the joint MLE. For general graphs, we utilize the Gamma distribution to model every infection time, proposing the Gamma Generated Tree (GGT) heuristic to transform the graph into a spanning tree and then apply the BP algorithm. Simulation results on random trees and Erdös-Rényi (ER) graphs show that the BP algorithm outperforms Rumor Center, Jodan Center and Dynamic Message Passing.

2. PROBLEM FORMULATION

Suppose the underlying network $G = (\mathcal{V}, \mathcal{E})$ is a connected simple graph. At time $t = 0$, an infection, following the SI model, starts from source $s \in \mathcal{V}$.

In the SI model, there are two types of nodes: ***S**usceptible* and ***I**nfected*. An infected node can infect its susceptible neighbor. Once a node is infected, it cannot become susceptible again, and hence the transition of state is $S \to I$ one-way. For general SI model, the time τ_{ij} it takes an infected node i to infect its susceptible neighbor j follows some distribution P_{ij}, and τ_{ij}'s are mutually independent across all (i, j). A special case is the *uniform-rate* SI model, where $\tau_{ij} \sim \mathsf{Exp}(\lambda)$:

$$\Pr(\tau_{ij} < t) = 1 - e^{-\lambda t},\ \forall (i,j) \in \mathcal{E}.$$

Now suppose an infection in G starts from s at time 0. At an unknown time t, we observe a **realization**, denoted as $G_I = (\mathcal{V}_I, \mathcal{E}_I)$, of the infection *random* graph $\mathcal{G}_I^s(t)$, which is the random subgraph of G composed of all infected nodes and the edges between them. Given underlying network structure G and infection graph G_I, we aim to compute $L(v, t|G_I)$, the joint likelihood function of the source location and the spreading time (v, t) given observation G_I, where

$$L(v, t|G_I) \triangleq \Pr(\mathcal{G}_I^s(t) = G_I | s = v, t),$$

so that statistical estimation can be carried out. A particularly estimator we would like to use is the joint maximum likelihood estimator (JMLE) of source and spreading time:

$$(\hat{s}_{\mathsf{JMLE}}, \hat{t}_{\mathsf{JMLE}}) = \underset{(v,t)\in\mathcal{V}\times\mathbb{R}_+}{\arg\max}\ L(v, t|G_I). \tag{1}$$

3. RECAP OF PREVIOUS WORKS

Instead of deriving the true MLE, previous works focus on heuristics to implement efficient estimators, leaving gaps to optimality. Below, a short recap is presented.

Rumor Center $\hat{s}_R$ is defined as:

$$\hat{s}_R = \underset{v\in\mathcal{V}_I}{\arg\max}\, R(v, G_I), \quad R(v, G_I) \triangleq |\mathcal{V}_I|! \prod_{u\in\mathcal{V}_I} \frac{1}{|\mathscr{T}_u^v|},$$

where $|\mathscr{T}_u^v|$ is the number of nodes in the subtree rooted at u when v is the source and $R(v, G_I)$ counts the number of infection sequences rooted at v. Besides, Jordan Center $\hat{s}_J$ is defined as

$$\hat{s}_J = \underset{s\in\mathcal{V}_I}{\arg\min} \max_{u\in\mathcal{V}_I} d(s, u),$$

where $d(s, u)$ is the minimum number of hops from s to u. Finally, Dynamic Message Passing algorithm computes $\tilde{\mathbb{P}}_v(i, t)$, the estimate of the probability of a node i being infected at time t if the source is v, which is the exact probability if the underlying network is a tree and an approximation otherwise. Then, it take $\tilde{\mathbb{P}}(G_I|v, t)$ as the approximation of the true likelihood function:

$$\tilde{\mathbb{P}}(G_I|v, t) = \prod_{i\in\mathcal{V}_I} \tilde{\mathbb{P}}_v(i, t) \prod_{j\notin\mathcal{V}_I} (1 - \tilde{\mathbb{P}}_v(i, t))$$

From the definitions above, we see that Rumor Center and Jordan Center only consider the information from G_I, so information from the locations of susceptible nodes, or equivalently, side information from $\mathcal{V} \setminus \mathcal{V}_I$, is ignored. On the other hand, Dynamic Message Passing approximates the joint distribution by the product of marginal distributions, so it partially ignores the dependencies among marginal distributions, or equivalently, the structure of G_I. To sum up, there is a dilemma between exact computation and heuristic: the former is prohibited by high complexity and the latter is subject to lost of information.

4. MAIN RESULTS

We propose an efficient algorithm to compute the exact likelihood functions, which fixes the issues mentioned above.

4.1. A Toy Example

We begin with a toy example of SI model (depicted in Figure 1) to illustrate how to derive the likelihood function $L(s, t|G_I)$ given $\tau_{ij} \sim P_{ij}$. Suppose A is the source, B got infected at time $\tau \sim P_{AB}$, and at time t, we observe G_I. Define message $m_{i\to j}$ as:

Definition 1 *$m_{i\to j}$ is the probability that at time t, if j is the source and infected, j causes the result of infection in the subtree rooted at i.*

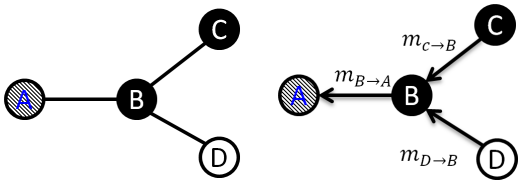


Fig. 1: A toy example for $L(s, t|G_I)$. A is the source, B got infected at time τ, C is infected, and D is not infected. The scenario in (a) can be formulated as (b).

Then, $L(s, t|G_I) = m_{B\to A}$ denotes the probability that if A is the source and infected, at time t, A causes B infected, C infected and D susceptible. In this way, we have the form of Belief Propagation as follows: ($*$ denotes convolution)

$$m_{C\to B} = F_{BC}(t) \ \ ; \ \ m_{D\to B} = 1 - F_{BD}(t)$$
$$m_{B\to A} = (m_{C\to B} m_{D\to B})(t) * f_{AB}(t),$$

where $F_{ij}(t)$ and $f_{ij}(t)$ are the CDF and PDF of P_{ij}.

4.2. Belief Propagation Algorithm

The toy example illustrates how the overall likelihood function can be computed using a Belief Propagation algorithm. Below, the general algorithm is described.

Algorithm 1 Belief Propagation For Single Source

Require: underlying network G and infection graph G_I

1: Find infected/susceptible boundary nodes.
2: For each boundary node i and its parent j, i passes to j $F_{ji}(t)$ if i is infected or $1 - F_{ji}(t)$ if i is susceptible.
3: **while** some nodes is unfinished **do**
4: **for all** i is unfinished **do**
5: **if** i gets all neighboring messages **then**
6: i is finished
7: **else if** i gets all but a neighbor j's message **then**
8: $m_{i\to j} = \Big(\prod_{k\in \mathsf{Nei}(i)\setminus\{j\}} m_{k\to i}(t)\Big) * f_{ji}(t)$
9: **end if**
10: **end for**
11: **end while**

Here $F_{ji}(t)$ and $f_{ji}(t)$ are the CDF and the PDF of the time it takes j to infect its susceptible neighbor i. For uniform-rate SI model, $F_{ji}(t) = 1 - e^{-\lambda t}$ and $f_{ji}(t) = \lambda e^{-\lambda t}$. $\mathsf{Nei}(i)$ denotes the neighbors of node i. Node i is a boundary node if either of the following is true:

- $i \in \mathcal{V}_I$ and its degree $deg(i) = 1$ on both G_I and G.
- $i \in \mathcal{V}\backslash\mathcal{V}_I$. i connects to a node $j \in \mathcal{V}_I$.

The algorithm starts from boundary nodes. A boundary node passes CDF to its parent if it is infected or complementary CDF if it is susceptible. Then, node i will pass a message to its neighbor j if i gets all but j's message. At line 8, all $m_{k\to i}$ are multiplied together because they are independent when i is the source. In addition, the resulting message after multipilcation is convolved with $f_{ji}(t)$ because passing the message is equivalent to the summation of random variables; that is, the time it take i to cause the result of infection on the subtree rooted at i pluses the time it takes j to infect i. Finally, the algorithm stops when all messages reach their destinations. Then by definition, $\prod_{j\in\mathsf{Nei}(i)} m_{j\to i}$ is the likelihood that when i is the source, at time t, i causes the result of the infection on each of its subtrees.

Suppose all messages are stored and computed numerically. Let L be the maximum number of points to store a message, n be the number of infected nodes, and D be the maximum degree. Note L is proportional to the range of time in which the estimator searches for the estimated spreading time. Then, multiplications of messages need $O(n(D-1)L)$ since there are $O(n(D-1))$ of them and each one needs $O(L)$. Convolutions with PDFs need $O(nL\log L)$ because there are $O(n)$ of them and each one takes $O(L\log L)$ if implemented by FFT (Fast Fourier Transformation). Thus, if $L \gg D$, then the time complexity of Algorithm 1 is $O(nL\log L)$.

Finally, with the likelihood functions at hand, the next step of JMLE is to find the pair (v,t) that maximizes $L(v,t|G_I)$. Note that searching for all pairs of (v,t) has a natural upper bound $O(nL)$. Hence the overall complexity of the Belief Propagation algorithm is $O(nL\log L)$.

4.3. Risk Minimization

In some scenarios, the loss functions other than the 0-1 loss may be of practical interested. For example, the distance error of the source $d(s,\hat{s})$ and the absolute error of the time $|t-\hat{t}|$. Hence it is reasonable to consider a framework to achieve the minimization for general risk functions.

First, assume that we are interested in a joint risk. Let $\ell(s,t,\hat{s},\hat{t})$ be the loss function with respect to the true parameters (s,t) and their estimates $(\hat{s},\hat{t})$. Suppose the prior distributions of s and t are uniform in $\mathcal{V}_I$ and $[0,T]$. Then the expected loss, or the risk, is proportional to

$$\tilde{l}(\hat{s},\hat{t}) = \int_0^T \sum_{v\in\mathcal{V}_I} \ell(v,\tau,\hat{s},\hat{t}) L(v,\tau|G_I) d\tau$$

The estimates $(\hat{s}_\ell,\hat{t}_\ell)$ that minimize the joint risk is obtained by minimizing $\tilde{l}(\hat{s},\hat{t})$.

On the other hand, suppose we are interested in minimizing risks separately. Let $\ell(s,\hat{s})$ and $\mathscr{L}(t,\hat{t})$ be the losses of the source and spreading time. Then the Iterative Risk Minimization allow us to minimizing them iteratively until convergence. Note that we call Algorithm 2 Iterative Minimum Distance Estimator (IMDE) when $\ell(s,\hat{s})$ is the distance error in hops and $\mathscr{L}(t,\hat{t})$ is the absolute error.

Algorithm 2 Iterative Risk Minimization

Require: infection graph $G_I = (\mathcal{V}_I, \mathcal{E}_I)$, likelihood function $L(\cdot,\cdot|G_I)$, $\hat{s}$, $\hat{t}$

1: **repeat**
2: $\hat{s} = \arg\min_{v\in\mathcal{V}_I} \sum_{i\in\mathcal{V}_I} \ell(i,v) L(i,\hat{t}|G_I)$
3: $\hat{t} = \arg\min_{t\in[0,T]} \int_0^T \mathscr{L}(\tau,t) L(\hat{s},\tau|G_I) d\tau$
4: **until** convergence

4.4. GGT Heuristic For General Graph

As suggested in [4], the estimation on general graphs can be done by constructing a Breadth-first Search (BFS) spanning tree and then applying the source detection algorithm. However, the BFS heuristic is not enough for our BP algorithm because the transmission on the spanning tree is heterogeneous. Thereby, the infections on the original graph and the spanning tree are both modeled by the *non-uniform-rate* SI model:

$\forall (j,v) \in \mathcal{E},\ \tau_{jv} \sim \mathsf{Exp}(\lambda_{jv})$. We propose the Gamma Generated Tree (GGT) heuristic to construct a weighted spanning tree, with weights on edges representing infection rates.

The spirit of GGT is to use the Gamma distribution to model every node's infection time (the time when the node got infected). First, we want to compute the model of a the infection time given candidate parents. This process is called Gamma Aggregation. Then apply Gamma Aggregation when constructing spanning tree, as shown in Algorithm 3.

Assume node i has candidate parents p_j, $1 \le j \le n$, whose infection times are modeled as $\tilde{T}_{p_j}$. Then the infection from p_j achieves i at time $\tilde{T}_{p_j} + \mathsf{Exp}(\lambda_{p_j i})$, which is further modeled by a Gamma r.v. $\tilde{T}'_{p_j}$ with mean $\mathbb{E}[\tilde{T}_{p_j}] + 1/\lambda_{p_j i}$ and variance $Var[\tilde{T}_{p_j}] + 1/\lambda^2_{p_j i}$. The Gamma Aggregation posits that for node i, its infection time is modeled as $\tilde{T}_i$:

$$\begin{aligned} &\tilde{T}_i \sim \Gamma(k, r),\ \frac{k}{r} = \mathbb{E}[\tilde{T}],\ \frac{k}{r^2} = \mathrm{Var}[\tilde{T}] \\ &\tilde{T} = \min\{\tilde{T}'_{p_j} :\ 1 \le j \le n\} \end{aligned} \tag{2}$$

Finally, the GGT heuristic is shown below. Note that at line 6, $\mathbb{E}[\tilde{T}_i - \tilde{T}_{\mathrm{par}_i} | \tilde{T}_i > \tilde{T}_{\mathrm{par}_i}]$ has a closed form. If $\tilde{T}_i \sim \Gamma(k_1, r_1)$, $\tilde{T}_{\mathrm{par}_i} \sim \Gamma(k_2, r_2)$, then it equals to

$$\frac{k_1 + k_2}{r_1} - \frac{k_2(r_1 + r_2)}{r_1 r_2} \frac{I_{r_2/(r_1+r_2)}(k_2 + 1, k_1)}{I_{r_2/(r_1+r_2)}(k_2, k_1)}, \tag{3}$$

where $I_z(a, b)$ is the regularized incomplete beta function.

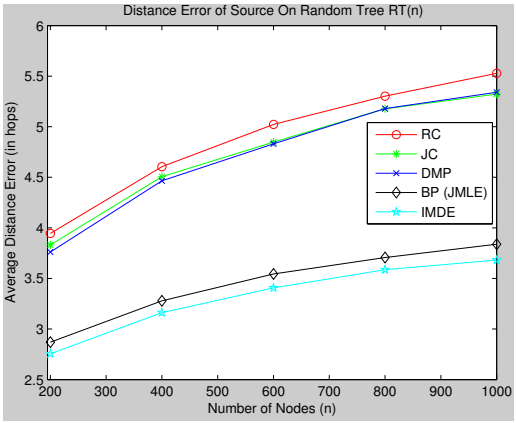
Algorithm 3 Gamma Generated Tree (GGT)

Require: Initial point s.

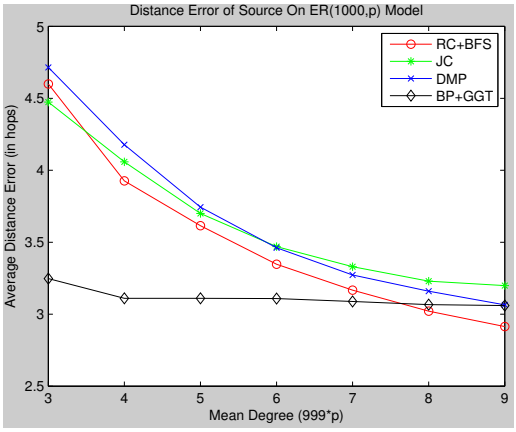
1: $\mathbf{I} = \{s\}$
2: **while** any infected or boundary node is not searched **do**
3: Pick $i = \arg\max_{v \in \mathcal{V} \setminus \mathbf{I}} \sum_{j \in \mathbf{I} \cap \mathsf{Nei}(v)} \lambda_{jv}$
4: Compute $\tilde{T}_i$ by Gamma Aggregation.
5: Pick $\mathrm{par}_i = \arg\min_{v \in \mathbf{I} \cap \mathsf{Nei}(i)} \mathbb{E}[\tilde{T}_v]$
6: Tree.insert$(i,\ \mathrm{par}_i,\ 1/\mathbb{E}[\tilde{T}_i - \tilde{T}_{\mathrm{par}_i} | \tilde{T}_i > \tilde{T}_{\mathrm{par}_i}])$
7: $\mathbf{I} = \mathbf{I} \cup \{i\}$
8: **end while**
9: **return** Tree

5. SIMULATION RESULTS

We present simulation results on Random Tree $RT(n)$ and Erdös-Rényi model $ER(n, p)$, where n is the number of nodes and p is the connection probability in ER model. The Random Tree we used is constructed by: (1) Initialize with a single node. (2) Each new node connects to one of the existing nodes with equal probability. The simulation is run in uniform-rate(1) SI model until half of the nodes are infected.



(a) Random Tree $RT(n)$



(b) ER model $ER(1000, p)$

Fig. 2: Performance on Random Tree and ER model.

From Figure 2, the BP algorithm outperforms others on $RT(n)$ and $ER(1000, p)$ when mean degree ≤ 7. Also, the IMDE algorithm works slightly better than the JMLE on $RT(n)$ since IMDE's objective is to minimize the distance error. Because $ER(1000, p)$ is similar to a tree when p is small, we conclude that the more similar the graph is to a tree, the more powerful the BP algoithm will be.

6. CONCLUSION

This paper addresses the rumor source detection in a general setting. To circumvent the issues in previous works, the problem is formulated as a joint maximum likelihood estimation (JMLE) over source and spreading time. For SI model on trees, we propose Belief Propagation algorithm (BP) to achieve the JMLE in time $O(nL \log L)$. For general graphs, we design the Gamma Generated Tree heuristic. Finally, the simulation concludes that the more similar the graph is to a tree, the more powerful the BP algoithm will be. The future directions could be to discuss the time error, practical risk functions and other infection models.

7. REFERENCES

[1] P. Li and Q. Zhao, "Rumor spreading in local-world evolving network," in *Applied Informatics and Communication: International Conference, ICAIC 2011, Xi'an, China, August 20-21, 2011, Proceedings, Part IV*, J. Zhang, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 693–699.

[2] M. Boots and A. Sasaki, "'small worlds' and the evolution of virulence: infection occurs locally and at a distance." *Proc Biol Sci*, vol. 266, no. 1432, pp. 1933–1938, Oct 1999, 10584335[pmid].

[3] M. S. S. Khan, "A computer virus propagation model using delay differential equations with probabilistic contagion and immunity," *International Journal of Computer Networks and Communications (IJCNC)*, vol. 6, no. 5, 2014.

[4] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5163–5181, Aug 2011.

[5] ——, "Finding rumor sources on random trees," *Operations Research*, vol. 64, no. 3, pp. 736–755, 2016.

[6] W. Luo, W. P. Tay, and M. Leng, "Identifying infection sources and regions in large networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2850–2865, June 2013.

[7] Z. Wang, W. Dong, W. Zhang, and C. W. Tan, "Rooting our rumor sources in online social networks: The value of diversity from multiple observations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 663–677, June 2015.

[8] A. Kumar, V. S. Borkar, and N. Karamchandani, "Temporally agnostic rumor-source detection," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 2, pp. 316–329, June 2017.

[9] J. Khim and P. L. Loh, "Confidence sets for the source of a diffusion in regular trees," *IEEE Transactions on Network Science and Engineering*, vol. 4, no. 1, pp. 27–40, Jan 2017.

[10] N. Karamchandani and M. Franceschetti, "Rumor source detection under probabilistic sampling," in *2013 IEEE International Symposium on Information Theory*, July 2013, pp. 2184–2188.

[11] K. Zhu and L. Ying, "Information source detection in the sir model: A sample-path-based approach," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 408–421, Feb 2016.

[12] W. Luo, W. P. Tay, M. Leng, and M. K. Guevara, "On the universality of the jordan center for estimating the rumor source in a social network," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, July 2015, pp. 760–764.

[13] Z. C. K. Zhu and L. Ying, "Catchem all: Locating multiple diffusion sources in networks with partial observations," in *AAAI Conference on Artificial Intelligence*, 2017.

[14] F. Ji and W. P. Tay, "An algorithmic framework for estimating rumor sources with different start times," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2517–2530, May 2017.

[15] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, "Inferring the origin of an epidemic with a dynamic message-passing algorithm," *Phys. Rev. E*, vol. 90, p. 012801, Jul 2014.

[16] A. Y. Lokhov, M. Mézard, and L. Zdeborová, "Dynamic message-passing equations for models with unidirectional dynamics," *Phys. Rev. E*, vol. 91, p. 012811, Jan 2015.

[17] J. S. Yedidia, W. T. Freeman, and Y. Weiss, *Exploring Artificial Intelligence in the New Millennium*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, ch. Understanding Belief Propagation and Its Generalizations, pp. 239–269.