Active-LATHE: An Active Learning Algorithm for Boosting the Error Exponent for Learning Homogeneous Ising Trees

Fengzhuo Zhang*, Anshoo Tandon† and Vincent Y. F. Tan‡
*Department of Electrical and Computer Engineering, National University of Singapore
†Department of Mathematics, National University of Singapore
Emails: fzzhang@u.nus.edu, anshoo.tandon@gmail.com, vtan@nus.edu.sg

Abstract—The Chow-Liu algorithm has been a mainstay for the learning of tree-structured graphical models from i.i.d. sampled data vectors. Its theoretical properties have been well-studied and are well-understood. In this paper, we focus on the class of trees that are arguably even more fundamental, namely homogeneous trees in which each pair of nodes that forms an edge has the same correlation ρ. We ask whether we are able to further reduce the error probability of learning the structure of the homogeneous tree model when active learning is allowed. Our figure of merit is the error exponent, which quantifies the exponential rate of decay of the error probability with an increasing number of data samples. We design and analyze an algorithm Active-Learning-Algorithm-for-Trees-with-Homogeneous-Edges (ACTIVE-LATHE), which surprisingly boosts (increases) the error exponent. Our analysis hinges on judiciously exploiting the minute but detectable statistical variation of the samples to allocate more data to parts of the graph in which we are less confident of being correct.

Index Terms—Active learning, Graphical models, Ising models, Error exponents

I. INTRODUCTION

Graphical models provide a succinct representation of the conditional independence relationships among random variables. Tree-structured graphical models are a special class of graphical models whose underlying graphs are trees, which are widely employed in domains such as computational biology [1], natural language processing [2], and signal processing [3].

The problem of learning the structure of graphical models is an important and fundamental task in model selection. With independent samples of random variables, the goal is to estimate the edges in the graph. The Chow–Liu (CL) algorithm proposed in [4] derive the maximum likelihood estimate of the underlying tree structure, which finds the Maximum Weight Spanning Tree (MWST) of the graph whose weights are empirical estimates of mutual information between random variables. Tan, Anandkumar, Tong, and Willsky [5] analyzed the exponential decay rate of the error probability, which is known as the error exponent. For the homogeneous Ising tree models, Tandon, Tan, and Zhu [6] showed that the exponent of the event that the CL algorithm estimates the wrong structure tends to 0 when the correlations between adjacent nodes approaches 1, which implies that the error probability decreases slowly with an increase in the number of independent samples. Fortunately, this performance degradation can be greatly mitigated by active learning.

Active learning is an umbrella term for algorithms that are allowed to actively select the data samples from which they learn the properties of the underlying distribution [7]. The freedom to choose the data samples to learn from allows the learner to home in on “most uncertain” parts of the model, which reduces the overall error probability or estimation error. Since all conventional structure learning algorithms, including the CL algorithm, rely on samples that are collected before the learning procedure commences, they are all passive learning algorithms. Although Vats, Nowak, and Baraniuk [8] and Dasarathy, Singh, Balcan, and Park [9] proposed active learning algorithms for general graphical models, their results cannot answer the question that whether the error exponent of the tree structure learning, and in particular homogeneous trees, can be improved compared to that of the CL algorithm. First, they focused on the sample complexities of the algorithms, and the exponents of the probabilities of errors cannot be directly derived from their results. Second, the effectiveness of their algorithms relies on the precise knowledge of bounds on the parameters of the graphical models to be estimated, but the CL algorithm does not require any information about the parameters of the distribution. Finally, the superiority of the active learning algorithm in [8] heavily relies on Assumption (A5) therein, which requires that one part of the graph must be much harder to learn than the other part. This assumption fails to hold in our homogeneous setting. Thus, whether active learning techniques can improve the error exponent of arguably the simplest class of graphical models, namely homogeneous trees, constitutes a fascinating and meaningful question to be resolved.

A. Main Contributions

In this paper, we design an active structure learning algorithm ACTIVE-LATHE1 for homogeneous Ising trees and
prove that its error exponent algorithm is no less than $c_\rho \geq 1$ times the error exponent of the CL algorithm on the homogeneous Ising trees, where $c_\rho$ is a non-decreasing function of the correlation parameter $\rho$ defined in Table II. Informally, denoting $T$, $T_{\text{active}}(n)$ and $T_{\text{passive}}(n)$ as the true tree, and the trees learned by ACTIVE-LATHE and the (minimum probability of error) CL algorithms (using $n$ vector-valued samples) respectively, then we have

$$\lim_{n \to \infty} -\frac{1}{n} \log \Pr\{T_{\text{active}}(n) \neq T\} \geq c_\rho \left( \lim_{n \to \infty} -\frac{1}{n} \log \Pr\{T_{\text{passive}}(n) \neq T\} \right).$$

Empirical simulations for a range of correlation parameters corroborate our theoretical findings. In this work, we have made no attempt to optimize (maximize) $c_\rho$ or to provide an upper bound for it (converse).

II. PRELIMINARIES AND PROBLEM STATEMENT

We use san-serif letters $x$, boldface letters $x$, uppercase letters $X$, and bold uppercase letters $X$ to denote scalars, vectors, random variables, and random vectors respectively. The notation $X^{(m)}$ represents the $m^{\text{th}}$ sample of $X$. The unique path connecting nodes $i$ and $j$ in the tree $T$ is denoted as $\text{Path}_T(i, j)$.

An undirected graphical model is a multivariate probability distribution that factorizes to an undirected graph [10]. More precisely, a $p$-dimensional random vector $X = (X_1, X_2, \ldots, X_p)$ is Markov on an undirected graph $G = (V, E)$, where $V = [p] \triangleq \{1, \ldots, p\}$ is the vertex set and $E \subseteq \binom{V}{2}$ is the edge set, if $\Pr(X_i | X_{V \setminus \{i\}}) = \Pr(X_i | X_{N(i)})$ for all $i \in [p]$, where $N(i) \triangleq \{j \in V \setminus \{i, j\} \in E\}$ is the neighborhood of node $i$. The degree of node $i$ is $d(i) \triangleq |N(i)|$, and the maximal degree of the graph is $d = \max_{i \in V} d(i)$. We focus on tree-structured graphical models, which are multivariate distributions that are Markov on acyclic and connected graphs.

A. System Model

We consider the binary vector $X = (X_1, X_2, \ldots, X_p) \in \{+1, -1\}^p$, which is also called an Ising model. The random vector $X$ is Markov on a tree $G = (V, E)$ and, in this paper, we focus on the models in which each random variable has zero mean and the interactions between the nodes that are connected by an edge are identical. This is also known as a homogeneous Ising model with zero external field [11], [12]. The probability mass function can be expressed as

$$P_\lambda(x) = \frac{1}{Z(\lambda)} \exp\left( \lambda \sum_{\{i, j\} \in E} x_i x_j \right),$$

where $\lambda \in \mathbb{R}$ is the parameter that describes the relationship of adjacent nodes and $Z(\lambda)$ is the partition function. For the model in (1), we have that for all $\{i, j\} \in E$,

$$\Pr(X_i = +1, X_j = +1) = \frac{e^\lambda}{2(e^\lambda + e^{-\lambda})}.$$
vector containing samples corresponding to a strict subset of the vertices in \( G \).

**Definition 1** (Exponent). The exponent of a sequence of events \( \mathcal{A} = \{\mathcal{A}(n)\} \) (where \( n \) denotes the number of vector samples) is defined as

\[
K(\mathcal{A}) \triangleq \lim_{n \to \infty} -\frac{1}{n} \log \Pr (\mathcal{A}(n)).
\]

The exponent \( K(\mathcal{A}) \) characterizes the asymptotic behavior of the exponential decay of the sequence of probabilities of the error events \( \{\mathcal{A}(n)\} \) for the structure estimate \( \hat{T}(n) \) of the homogeneous Ising tree \( T \) with \( n \) vector samples, our work focuses on the behavior of the exponent of the sequence of events \( \{\hat{T}(n) \neq T\} \). However, the exponent of \( \{\hat{T}(n) \neq T\} \) is intuitively difficult to improve with active learning techniques, because every edge has the same correlation \( \rho \), and so each part of the tree is equally good or bad.

### III. Structure Learning Algorithms and Main Result

In this section, we describe both the passive (CL-based) and active tree structure learning algorithms (ACTIVE-LATHE) and state the main result. Due to the space limitations, we omit all the proofs, which can be found in [13].

#### A. Passive Structure Learning Algorithm

The CL algorithm is a classical tree structure learning algorithm that finds the maximum likelihood estimate of the tree structure given independent samples [4]. It finds the MWST of the graph whose weights are the empirical estimates of the mutual information between vertices, i.e.,

\[
\hat{T}_{\text{CL}}(n) = \arg \max_{T \in \mathcal{T}^n} \sum_{(i,j) \in \mathcal{E}(T)} \hat{I}(X_i; X_j),
\]

where \( \mathcal{T}^n \) is the set of all the trees with \( n \) vertices, and \( \hat{I}(X_i; X_j) \) is the empirical estimate of the mutual information between \( X_i \) and \( X_j \) computed using \( n \) independent samples. The CL algorithm takes \( n \) vector samples (i.e., \( n \) scalar samples) as the inputs and acquires the samples of all the vertices at once. Using the terminology in this paper, the CL algorithm is a passive learning algorithm. We note that the mutual information between any two vertices \( I(X_i; X_j) \) is an increasing function of the correlation \( 0 < \rho_{ij} < 1 \) for Ising trees with zero external field. Thus, instead of taking the mutual information as the weights of edges, the Simplified Chow–Liu (SCL) algorithm takes the empirical correlation estimates \( \hat{\rho}_{ij}(m) \) as the inputs, where \( m \) is the number of samples used, as the weights of the edges, i.e.,

\[
\hat{T}_{\text{SCL}}(n) \triangleq \hat{T}_{\text{SCL}}(n) = \arg \max_{T \in \mathcal{T}^n} \sum_{(i,j) \in \mathcal{E}(T)} \hat{\rho}_{ij}(m),
\]

where the subscript in \( \hat{T}_{\text{SCL}} \) stands for the SCL algorithm. The relationship between \( \hat{T}_{\text{SCL}}(n) \) and \( \hat{T}(n) \) was stated in [14, Lemma A.2]. We restate it here for ease of reference.

**Proposition 1.** For the homogeneous Ising tree-structured graphical model with zero external field parameterized by \( 0 < \rho < 1 \), \( \hat{T}_{\text{SCL}}(n) = \hat{T}_{\text{CL}}(n) \) for all \( n \in \mathbb{N} \).

Since the homogeneous Ising tree \( T \) is parameterized by a single parameter \( \rho \), the exponent of the error events \( \{\hat{T}_{\text{SCL}}(n) \neq T\} \) is, i.e., the passive learning algorithm in Eqn. (2) learns the wrong structure, can be expressed as

\[
K_{\text{passive}}(T, \rho) \triangleq \lim_{n \to \infty} -\frac{1}{n} \log \Pr (\hat{T}_{\text{passive}}(n) \neq T).
\]

For homogeneous Ising trees, the passive error exponent \( K_{\text{passive}}(T, \rho) \) can be expressed as an explicit function of \( \rho \).

**Proposition 2** (Tandon, Tan, and Zhu [6]). For the homogeneous Ising tree with zero external field \( T \) parameterized by \( 0 < \rho < 1 \), the error exponent of the SCL algorithm can be expressed as

\[
K_{\text{passive}}(T, \rho) = -\log \left( 1 - \frac{1}{2} \frac{\rho}{(1 - 1 - \rho^2)} \right).
\]

**Remark 1.** The error exponent for the passive learning algorithm on homogeneous Ising models does not depend on the tree structure \( T \), but only depends on the edge correlation \( \rho \). In the following, we will denote \( K_{\text{passive}}(T, \rho) \) as \( K_{\text{passive}}(\rho) \) to suppress its dependence on the underlying structure \( T \).

When \( \rho \) tends to 1, the exponent \( K_{\text{passive}}(\rho) \) sharply decreases to 0, which implies that the error probability of the passive learning algorithm decreases very slowly with an increasing number of i.i.d. vector samples. In the following, we present an active learning algorithm ACTIVE-LATHE that mitigates this problem. ACTIVE-LATHE significantly improves the error exponent for large values of \( \rho \). In particular, for \( \rho \in [0.8, 1) \), the error exponent of ACTIVE-LATHE will be shown to be at least 1.4 times that of \( K_{\text{passive}}(\rho) \).

#### B. Active Structure Learning Algorithm

ACTIVE-LATHE has two phases—the global learning and local refinement phases. In the following subsections, we describe these phases.
With the CL structure estimate, we then estimate the correlation $\hat{\alpha}$.

Algorithm 2: Active Learning Algorithm for Trees with Homogeneous Edges (ACTIVE-LATHE)

**Input:** The number of nodes $p$, the number of vector samples $n$

**Output:** Estimated tree structure $\hat{T}_{active}(n)$

**Procedure:**
1. # Global Learning Phase
2. Initialize $i = 1$, $\hat{\alpha}_0 = 0$, $\hat{\alpha}_1 = 0.8$.
3. while $\hat{\alpha}_i > \hat{\alpha}_{i-1}$ do
4. Acquire $[\hat{\alpha}_i n] - [\hat{\alpha}_{i-1} n]$ vector samples of all the nodes.
5. Learn the CL tree $\hat{T}_{SCL}([\hat{\alpha}_i n])$ as in Eqn. (3).
6. Calculate $\hat{\rho}([\hat{\alpha}_i n])$ as in Eqn. (4).
7. Set $\hat{\alpha}_{i+1}$ according to $\hat{\rho}([\hat{\alpha}_i n])$ and Table I.
8. $i = i + 1$.
9. end while
10. Set $\hat{\alpha} = \hat{\alpha}_{i-1}$.
11. # Local Refinement Phase
12. Set $c_v([\hat{\alpha}_i n])^\mathcal{E} = \emptyset$ and $c_e([\hat{\alpha}_i n])^\mathcal{E} = \emptyset$.
13. for all node triples $(i, j, k)$ where $(i, j, k) \in \mathcal{E}(\hat{T}_{SCL}([\hat{\alpha}_i n]))$ do
14. if $c_{ijk}([\hat{\alpha}_i n])$ (defined in Eqn. (5)) does not hold then
15. add $(i, j, k)$ to $c_e([\hat{\alpha}_i n])^\mathcal{E}$ and add nodes $i, j, k$ to $c_v([\hat{\alpha}_i n])^\mathcal{E}$.
16. end if
17. end for
18. Acquire $[(1 - \hat{\alpha}) np/|c_v([\hat{\alpha}_i n])^\mathcal{E}|]$ sub-vector samples corresponding only to the nodes in $c_v([\hat{\alpha}_i n])^\mathcal{E}$.
19. Implement the SCL algorithm to relearn the structure of connected components in the subgraph formed by $c_v([\hat{\alpha}_i n])^\mathcal{E}$ and $c_e([\hat{\alpha}_i n])^\mathcal{E}$ using all the sub-vector samples corresponding to each connected component.
20. Replace every connected component with the relevant structure.
21. Return the derived $\hat{T}_{active}(n)$.

1) Global learning phase: The global learning phase is designed to be iterative up to a stopping time. As indicated in Algorithm 2, in the $i$th iteration of the global learning phase, we first implement the SCL algorithm with $\hat{\alpha}_i n$ vector samples, where we set $\hat{\alpha}_1 = 0.8$ for the first iteration; this is the minimum of $\alpha$ shown in Table I, i.e.,

$$\hat{T}_{SCL}(\hat{\alpha}_i n) = \arg \max_{T \in \mathcal{T}} \sum_{e \in \mathcal{E}(T)} \hat{\rho}_e(\hat{\alpha}_i n).$$

With the CL structure estimate, we then estimate the correlation $\rho$ of adjacent vertices as

$$\hat{\rho}(\hat{\alpha}_i n) = \frac{1}{p-1} \sum_{e \in \mathcal{E}(\hat{T}_{SCL}(\hat{\alpha}_i n))} \hat{\rho}_e(\hat{\alpha}_i n).$$

The estimate $\hat{\rho}(\hat{\alpha}_i n)$ is used to find the corresponding $\hat{\alpha}_{i+1} \leq \hat{\alpha}(\hat{\rho}(\hat{\alpha}_i n))$ according to Table I. If $\hat{\alpha}_{i+1} \leq \hat{\alpha}_i$, we will stop and move on to the local refinement phase. Otherwise, we go to the next iteration. Note that the $\hat{\alpha}_i$s are random variables that are functions of all the acquired samples. We denote the $\hat{\alpha}_{i-1}$ in the last iteration as $\hat{\alpha}$.

As Table I suggests, this iterative design of the global learning phase aims to use more vector samples to learn the global structure of the model when $\rho$ is small.

2) Local refinement phase: In the local refinement phase, we first detect the confident part of $\hat{T}_{SCL}(\hat{\alpha}_i n)$ using the correlation decay property [15]. For a three-node subtree $i - j - k$ in a homogeneous tree with $0 < \rho < 1$, we have that $\rho_{ik} = \rho_{ij} \rho_{jk}$, which follows from the correlation decay property. Since $\rho_{ij} = \rho_{jk} = \rho$ and $0 < \rho < 1$, we have that

$$\rho_{ik} \leq \rho_{ij} \cdot \frac{11 + 9\rho}{20}$$

and $\rho_{ik} \leq \rho_{jk} \cdot \frac{11 + 9\rho}{20}$.

The term $(11 + 9\rho)/20$ is chosen to balance the exponents corresponding to errors in the confident and unconfident edges. If this term is increased, the exponent corresponding to an error in the confident edge set may decrease. On the other hand, if this term is decreased, then the condition for declaring an edge as confident becomes stricter. The samples allocated for the difficult part will be distracted, and the error exponent for the unconfident part will decrease. In general, for any three nodes $i$, $j$, and $k$ of the graph, we define the $(i, j, k)$-confident event based on correlation estimates with $m$ vector samples as

$$C_{ijk}(m) \triangleq \left\{ \hat{\rho}_{ik}(m) \leq \hat{\rho}_{ij}(m) \cdot \frac{11 + 9\hat{\rho}(m)}{20} \right\} \cap \left\{ \hat{\rho}_{ik}(m) \leq \hat{\rho}_{jk}(m) \cdot \frac{11 + 9\hat{\rho}(m)}{20} \right\}.$$ (5)

When the number of samples $m$ is clear from the context, we abbreviate $\hat{\rho}_{ij}(m)$ and $C_{ijk}(m)$ as $\hat{\rho}_{ij}$ and $C_{ijk}$ respectively.

Remark 2. The event $C_{ijk}$ implies that $(i, k) \notin \mathcal{E}(\hat{T}_{SCL})$. In fact, if $C_{ijk}$ holds, we are “overconfident” about the fact that the subgraph $i - j - k$ among nodes $i$, $j$, and $k$ is correct, since the SCL algorithm only requires $\hat{\rho}_{ik} < \hat{\rho}_{ij}$ and $\hat{\rho}_{ik} < \hat{\rho}_{jk}$ to recover the correct underlying subtree $i - j - k$.

Intuitively, if the confident event $C_{ijk}$ holds, $\hat{\rho}_{ik}$ is sufficiently smaller than $\hat{\rho}_{ij}$ and $\hat{\rho}_{jk}$. This condition suffices for the SCL algorithm to find the correct subgraph among these three nodes. To describe the subroutines of the local refinement steps clearly, we define the confident edges, the unconfident edges and the unconfident nodes as follows.

Definition 2 (Confident edges, Unconfident edges and Unconfident nodes). For the CL tree estimated with $m$ vector samples $\hat{T}_{SCL}(m)$, the confident and unconfident edge sets are defined as

$$c_v(m) = \left\{ i, j \in \mathcal{E}(\hat{T}_{SCL}(m)) \right\} \cap \left\{ \forall (i, k), (i, l) \in \mathcal{E}(\hat{T}_{SCL}(m)) \right\}$$

and $c_e(m)^\mathcal{E} = \mathcal{E}(\hat{T}_{SCL}(m)) \setminus c_v(m)$, respectively. The edges in $c_v(m)$ (resp. $c_e(m)^\mathcal{E}$) are called confident (resp. unconfident) edges. The set of unconfident nodes is defined as

$$c_v(m)^\mathcal{V} = \left\{ i \in \mathcal{V} \mid \exists j \in \mathcal{V} \text{ s.t. } (i, j) \in c_e(m)^\mathcal{E} \right\}.$$
Remark 3. The set of unconfident nodes can be equivalently defined as the set of all nodes that are incident to an unconfident edge.

We denote the number of unconfident nodes as \( \hat{p}(m) \triangleq |\mathcal{C}_V(m)| \). Note that \( \hat{p}(m) \) is a function of the number of the vector samples used to learn the tree \( m \). As shown in Algorithm 2, the local refinement phase first checks the occurrences of each of the events \( C_{ijk} \) for all the triples \( (i, j, k) \) such that \( \{i, j\}, \{j, k\} \in \mathcal{E}(T_{SCL}(\hat{a}, n)) \). Since the confident edges are correct with higher probability compared to the unconfident edges, we then only assign samples to the unconfident nodes \( \mathcal{C}_V(\hat{a}, n) \) and relearn the subgraph of these unconfident nodes. The number of scalar samples assigned to each unconfident node is about \( \hat{\alpha}(1-\hat{\alpha})p/\hat{p}(\hat{a}, n)n \geq n \), so the probability of the event that the subgraph of the unconfident nodes is wrong is reduced compared to the passive learning algorithm. Then we can show (see Lemma 4 in [13]) that

\[
\left[\hat{\alpha} + (1-\hat{\alpha})\frac{p}{\hat{p}(\hat{a}, n)}\right]n \geq c_\rho n, \tag{6}
\]

where \( c_\rho \) is the piecewise constant function defined in Table II.

Eqn. (6) shows that the number of scalar samples assigned to the unconfident nodes is strictly larger than that in the passive learning scenario.

C. The Main Result

With the homogeneous Ising tree \( T \) parameterized by \( \rho \), the exponent of the (sequence of) error events \( \{T_{active}(n) \neq T\}_{n \in \mathbb{N}} \) (this event corresponds to \( \text{ACTIVE-LATHE} \) learning the wrong structure), can be expressed as

\[
K_{active}(T, \rho) \triangleq \lim_{n \to \infty} -\frac{1}{n} \log \Pr \left( T_{active}(n) \neq T \right).
\]

Our main theorem is stated as follows.

Theorem 1. For the homogeneous Ising tree with zero external field \( T \) parameterized by \( 0 < \rho < 1 \) that satisfies Assumption 1, the error exponent of the \( \text{ACTIVE-LATHE} \) algorithm is at least \( c_\rho \), times the error exponent of the passive learning algorithm \( K_{passive}(\rho) \), i.e.,

\[
K_{active}(T, \rho) \geq c_\rho K_{passive}(T, \rho) \quad \forall T \in \mathcal{F},
\]

where \( c_\rho \) is the piecewise constant function defined in Table II.

Theorem 1 shows that the error exponent of the \( \text{ACTIVE-LATHE} \) algorithm \( K_{active}(T, \rho) \) is no less than the error exponent of the passive learning algorithm \( K_{passive}(\rho) \). In fact, it is strictly larger than \( K_{active}(T, \rho) \) by a multiplicative factor of \( c_\rho \). In particular, when \( \rho \in [0.8, 1) \), in which \( K_{passive}(\rho) \) suffers from a rapid decline to 0 as \( \rho \to 1^- \), the \( \text{ACTIVE-LATHE} \) algorithm improves the error exponent by at least 40% relative to its passive counterpart, which implies that the probability of error decays significantly faster if active learning is allowed.

IV. SIMULATION RESULTS

We present simulation results to demonstrate the efficacy of \( \text{ACTIVE-LATHE} \) compared to its passive counterpart. Samples are generated from a binary tree with \( p = 255 \) (seven levels). The correlations between adjacent nodes \( \rho \) are set to 0.9, 0.7, and 0.5. Theorem 3 in [6] provides an exact asymptotic expression for the best possible error probability, and it serves as a baseline of the simulation; this is indicated as “Passive SCL: Theory” in Fig. 1. Note that no other passive algorithm can perform better than that given by Theorem 3 in [6] (i.e., “Passive SCL: Theory”) since this is based on the maximum likelihood or minimum error probability principle. Hence, “Passive SCL: Theory” serves as a \textit{bona fide} impossibility (converse) result for tree structure learning using passive strategies.

Fig. 1 clearly indicates the efficacy of \( \text{ACTIVE-LATHE} \) whose error probabilities are much smaller than those of the passive Chow–Liu learning algorithm, especially when \( n \) is sufficiently large. The \textit{slopes} of the error curves correspond to the error exponents. The improvements to the slopes of the \( \text{ACTIVE-LATHE} \) algorithm (the slopes are more negative for \( \text{ACTIVE-LATHE} \) compared to the passive algorithm) are especially pronounced when \( \rho \) is large. The large reductions of the error probability (the error exponent increases by at least 40%) shown in Fig. 1(a) greatly mitigate the inherent deficiencies of the passive learning for large \( \rho \) as discussed in Section III-A. The code to reproduce the figures is available at Github https://github.com/Zhang-Fengzhuo/Active-LATHE.
REFERENCES

[1] N. Saitou and M. Nei, “The neighbor-joining method: A new method for reconstructing phylogenetic trees,” Mol. Bio. Evol., vol. 4, no. 4, pp. 406–425, 1987.
[2] A. Parikh, S. B. Cohen, and E. Xing, “Spectral unsupervised parsing with additive tree metrics,” in Proceedings of ACL, 2014, pp. 1062–1072.
[3] A. Montanari, “Graphical models concepts in compressed sensing,” in Compressed Sensing: Theory and Applications, Y. C. Eldar and G. Kutyniok, Eds. Cambridge University Press, 2012, pp. 394–438.
[4] C. Chow and C. Liu, “Approximating discrete probability distributions with dependence trees,” IEEE Trans. Inform. Theory, vol. 14, no. 3, pp. 462–467, 1968.
[5] V. Y. F. Tan, A. Anandkumar, L. Tong, and A. S. Willsky, “A large-deviation analysis of the maximum-likelihood learning of markov tree structures,” IEEE Trans. Inf. Theory, vol. 57, no. 3, pp. 1714–1735, 2011.
[6] A. Tandon, V. Y. F. Tan, and S. Zhu, “Exact asymptotics for learning tree-structured graphical models: Noiseless and noisy samples,” IEEE Journal on Selected Areas of Information Theory, vol. 1, no. 3, pp. 760–776, 2020.
[7] B. Settles, Active Learning Literature Survey. University of Wisconsin-Madison Department of Computer Sciences, 2009.
[8] D. Vats, R. Nowak, and R. Baraniuk, “Active learning for undirected graphical model selection,” in Artificial Intelligence and Statistics. PMLR, 2014, pp. 958–967.
[9] G. Dasarathy, A. Singh, M. Balcan, and J. H. Park, “Active learning algorithms for graphical model selection,” in Artificial Intelligence and Statistics. PMLR, 2016, pp. 1356–1364.
[10] S. L. Lauritzen, Graphical models. Clarendon Press, 1996, vol. 17.
[11] C. Daskalakis, N. Dikkala, and G. Kamath, “Testing Ising models,” IEEE Trans. Inf. Theory, vol. 65, no. 11, pp. 6829–6852, 2019.
[12] A. Tandon, A. H. J. Yuan, and V. Y. F. Tan, “SGA: A robust algorithm for partial recovery of tree-structured graphical models with noisy samples,” in International Conference on Machine Learning. PMLR, 2021.
[13] F. Zhang, A. Tandon, and V. Y. F. Tan, “Active-lathe: An active learning algorithm for boosting the error exponent for learning homogeneous Ising trees,” arXiv preprint arXiv:2110.14341, 2021.
[14] G. Bresler and M. Karzand, “Learning a tree-structured Ising model in order to make predictions,” The Annals of Statistics, vol. 48, no. 2, pp. 713–737, 2020.
[15] K. E. Nikolakakis, D. S. Kalogerias, and A. D. Sarwate, “Predictive learning on hidden tree-structured Ising models,” Journal of Machine Learning Research, vol. 22, no. 59, pp. 1–82, 2021.