Inference in Sparse Graphs with Pairwise Measurements and Side Information

Dylan J. Foster
Cornell University

Daniel Reichman
University of California, Berkeley

Karthik Sridharan
Cornell University

Abstract

We consider the statistical problem of recovering a hidden "ground truth" binary labeling for the vertices of a graph up to low Hamming error from noisy edge and vertex measurements. We present new algorithms and a sharp finite-sample analysis for this problem on trees and sparse graphs with poor expansion properties such as hypergrids and ring lattices. Our method generalizes and improves over that of Globerson et al. (2015), who introduced the problem for two-dimensional grid lattices.

For trees we provide a simple, efficient, algorithm that infers the ground truth with optimal Hamming error has optimal sample complexity and implies recovery results for *all connected graphs*. Here, the presence of side information is critical to obtain a non-trivial recovery rate. We then show how to adapt this algorithm to tree decompositions of edge-subgraphs of certain graph families such as lattices, resulting in optimal recovery error rates that can be obtained efficiently

The thrust of our analysis is to 1) use the tree decomposition along with edge measurements to produce a small class of viable vertex labelings and 2) apply an analysis influenced by statistical learning theory to show that we can infer the ground truth from this class using vertex measurements. We show the power of our method in several examples including hypergrids, ring lattices, and the Newman-Watts model for small world graphs. For two-dimensional grids, our results improve over Globerson et al. (2015) by obtaining optimal recovery in the constant-height regime.

1 Introduction

Statistical inference over graphs and networks is a fundamental problem that has received extensive attention in recent years (Fortunato, 2010; Krzakala et al., 2013; Abbe et al., 2014; Hajek et al., 2014). Typical inference problems involve noisy observations of discrete labels assigned to edges of a given network, and the goal is to infer a "ground truth" labeling of the vertices (perhaps up to the right sign) that best explains these observations. Such problems occur in a wide range of disciplines including statistical physics, sociology, community detection, average case analysis, and graph partitioning. This inference problem is also related to machine learning tasks involving structured prediction that arise in computer vision, speech recognition and other applications such as natural language processing. Despite the intractability of maximum likelihood estimation, maximum a-posteriori estimation, and marginal inference for most network models in the worst case, it has been observed that approximate inference algorithms work surprisingly well in practice (Sontag et al., 2012), and recent work has focused on improving our theoretical understanding of this phenomenon (Globerson et al., 2015).

Globerson et al. (2015) introduced a new inference model with the key feature that, in addition to observing noisy edge labels, one also observes noisy vertex labels. The main focus of the present paper is to further examine the extent to which the addition of noisy vertex observations improves the statistical aspects of approximate recovery. Specifically, we analyze statistical recovery rates in Model 1.

As a concrete example, consider the problem of trying to recover opinions of individuals in social networks. Suppose that every individual in a social network can hold one of two opinions labeled by -1 or $+1$. We receive a measurement of whether neighbors in the network have the same opinion, but the value of each measurement is flipped with probability p. We further receive estimates of the opinion of each individual, perhaps using a classification model on their profile,

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

but these estimates are corrupted with probability q.

Model 1. We receive an undirected graph $G = (V, E)$ with $|V| = n$, whose vertices are labeled according to an unknown ground truth $Y \in \{\pm 1\}^V$. We receive noisy edge measurements $X \in \{\pm 1\}^E$, where $X_{uv} = Y_u Y_v$ with probability $1 - p$ and $X_{uv} = -Y_u Y_v$ otherwise. We receive "side information" vertex measurements $Z \in \{\pm 1\}^V$, where $Z_u = Y_u$ with probability $1 - q$ and $Z_u = -Y_u$ otherwise. We assume $p < q < 1/2$ Our goal is to produce a labeling $\widehat{Y} \in \{\pm 1\}^V$ such that with probability at least $1 - o_n(1)$ the Hamming error $\sum_{v \in V} \mathbb{1}\{\widehat{Y}_v \neq Y_v\}$ is bounded by $O(f(p)n)$ where $\lim_{p \to 0} f(p) = 0$.

The reader may imagine the pairwise measurements as fairly accurate and the side information vertex estimates as fairly noisy (since the flip probability q close to $1/2$). Model 1 then translates to the problem of producing an estimate of the opinions of users in the social network which predicts the opinion of few users incorrectly.

A first step in studying recovery problems on graphs with noisy vertex observations was taken by Globerson et al. (2014, 2015) who studied Model 1 on square grid lattices. They proved that the statistical complexity of the problem is essentially determined by the number of cuts with cutset of size k, where k ranges over nonnegative integers. This observation, together with a clever use of planar duality, enabled them to determine the optimal Hamming error for the square grid.

As in Globerson et al. (2014, 2015) we focus on finding a labeling of low Hamming error (as opposed to *exact recovery*, where one seeks to find the error probability that with which we recover all labels correctly). Chen et al. (2016) have recently considered exact recovery for edges in this setting for sparse graphs such as grid and rings. They consider the case where there are *multiple* i.i.d observations of edge labels. In contrast, we focus on the case where there is a single (noisy) observation for each edge, on side information, and on partial recovery[1].

The availability of vertex observations changes the statistical nature of the problem and — as we will show — enables nontrivial partial recovery rates in all sparsity regimes. For example, for the n-vertex path, it is not difficult (Globerson et al., 2014) to show that when there are only noisy edge observations any algorithm will fail to find the correct labeling (up to sign) of $\Omega(n)$ edges. In contrast, we show that when noisy vertex observations are available, one can obtain a labeling whose expected Hamming error is at most $\widetilde{O}(pn)$ for any p.

Related community detection models such as the well known Stochastic Block Model (SBM) and Censored Block Model (CBM) consider the case where one wishes to detect two communities based on noisy edge observations. Namely, in these models only noisy edges observations are provided and one wishes to recover the correct labeling of vertices up to sign. Block model literature has focused on graphs which have good expansion properties such as complete graphs, random graphs, and spectral expanders. By including side information, our model allows for nontrivial recovery rates and efficient algorithms for graphs with "small" separators such as trees, thin grids, and ring lattices. Studying recovery problems in such "non-expanding" graphs is of interest as many graphs arising in applications such as social networks (Flaxman, 2007) have poor expansion.

Challenges and Results The key challenge in designing algorithms for Model 1 is understanding statistical performance: Even for graphs such as trees in which the optimal estimator (the marginalized estimator) can be computed efficiently, it is unclear what Hamming error rate this estimator obtains. Our approach is to tackle this statistical challenge directly; we obtain efficient algorithms as a corollary.

Our first observation is that the optimal Hamming error for trees is $\widetilde{\Theta}(pn)$ provided q is bounded away from $1/2$[2]. This is obtained by an efficient message passing algorithm. We then (efficiently) extend our algorithm for trees to more general graphs using a tree decompositions of (edge)-subgraphs. Our main observation is that if we are given an algorithm that obtains a non-trivial error rate for inference in each constant-sized component of a tree decomposition, we can lift this algorithm to obtain a non-trivial error rate for the entire graph by leveraging side information.

This approach has the advantage that it applies to non-planar graphs such as high dimensional grids; it is not clear how to apply the machinery of Globerson et al. (2015) to such graphs because planar duality no longer applies. Our decomposition-based approach also enables us to obtain optimal error bounds for ring lattices and thin grids which do not have the so-called weak expansion property that is necessary for

[1] We refer the reader to Appendix A for further discussion of related models.

[2] The assumption on q is necessary, as when q approaches $1/2$ it is proven in Globerson et al. (2015) that an error of $\Omega(n)$ is unavoidable for certain trees.

the analysis in Globerson et al. (2015). See Section 4 for an extensive discussion of concrete graph families we consider and the error rates we achieve.

1.1 Preliminaries

We work with an undirected graph $G = (V, E)$, with $|V| = n$ and $|E| = m$. For $W \subseteq V$, we let $G(W)$ be the induced subgraph and $E(W)$ be the edge set of the induced subgraph. Let $N(v)$ be the neighborhood of a vertex v. When it is not clear from context we will use $N_G(v)$ to denote neighborhood with respect to a specific graph G. Likewise, for $S \subseteq V$ we use $\delta_G(S)$ to denote its cut-set (edges with one endpoint in S) with respect to G. For a directed graph, we let $\delta_+(v)$ denote the outgoing neighbors and $\delta_-(v)$ denote the incoming neighbors of v. For a subset $W \subseteq V$ we let $N_G(W) = \bigcup_{v \in W} N_G(v)$. We let $\mathsf{deg}(G)$ denote the maximum degree and Δ_{avg} the average degree.

Parameter range We treat $q = 1/2 - \epsilon$ as constant unless otherwise specified. Furthermore, we shall assume throughout that $p \geq w(1/n)$, so the expected number of edge errors is super-constant. We use $\widetilde{O}$ to suppress $\log(n)$, $\log(1/p)$, and $1/\epsilon$ factors. We use the phrase "with high probability" to refer to events that occur with probability at most $1 - o_n(1)$.

In the appendix (Theorem 6) we show that if the minimum degree of the graph is $\Omega(\log n)$ there is a trivial strategy that achieves arbitrarily small Hamming error. We therefore restrict to $\mathsf{deg}(G)$ constant, as this is representative of the most interesting parameter regime.

2 Inference for Trees

In this section we show how to efficiently and optimally perform inference in Model 1 when the graph G is a tree. As a starting point, note that the expected number of edges (u, v) of the tree with X_{uv} flipped is $p(n-1)$. In fact, using a simple Chernoff bound, one can see that with high probability at most $2pn + \widetilde{O}(1)$ edges are flipped. This implies that for the ground truth Y, $\sum_{(u,v)\in E} \mathbb{1}\{Y_u \neq X_{u,v} Y_v\} \leq 2pn + \widetilde{O}(1)$ with high probability over sampling of the edge labels. Hence to estimate ground truth, it is sufficient to search over labelings $\widehat{Y}$ that satisfy the inequality

$$\sum_{(u,v)\in E} \mathbb{1}\{\widehat{Y}_u \neq X_{u,v} \widehat{Y}_v\} \leq 2pn + \widetilde{O}(1). \tag{1}$$

We choose the estimator that is most correlated with the vertex observations Z subject to the aforementioned inequality. That is, we take $\widehat{Y}$ to be the solution to[3]

$$\begin{aligned} &\text{minimize} && \sum_{v \in V} \mathbb{1}\{\widehat{Y}_v \neq Z_v\} \\ &\text{subject to} && \sum_{(u,v)\in E} \mathbb{1}\{\widehat{Y}_u \neq X_{u,v}\widehat{Y}_v\} \leq 2pn + \widetilde{O}(1). \end{aligned} \tag{2}$$

This optimization problem can be solved efficiently — $O(\lceil pn \rceil^2 n \mathsf{deg}(G))$ time for general trees and $O(\lceil pn \rceil n)$ time for stars and path graphs — with message passing. The full algorithm is stated in Appendix D.

On the statistical side we use results from statistical learning theory to show that the Hamming error of $\widehat{Y}$ obtained above is with high probability bounded by $\widetilde{O}(pn)$. To move to the statistical learning setting (see Appendix C for an overview) we first define a "hypothesis class" $\mathcal{F} \triangleq \{Y' \in \{\pm 1\}^V \mid \sum_{(u,v)\in E} \mathbb{1}\{Y'_u \neq X_{u,v} Y'_v\} \leq 2pn + \widetilde{O}(1)\}$; note that this is precisely the set of Y' satisfying (1). The critical observation here is that for any $\widehat{Y}$ the Hamming error (with respect to the ground truth) is proportional to the *excess risk* in the statistical learning setting over Z with class $\mathcal{F}$:

$$\begin{aligned} &\sum_{v \in V} \mathbb{1}\{\widehat{Y}_v \neq Y_v\} \\ &= \frac{1}{1-2q}\left[\sum_{v \in V} \mathbb{P}_Z\{\widehat{Y}_v \neq Z_v\} - \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{P}_Z\{Y'_v \neq Z_v\}\right]. \end{aligned} \tag{3}$$

Combining (3) with a so-called *fast rate* from statistical learning theory (Corollary 2) implies that if we take $\widehat{Y}$ to be the *empirical risk minimizer* over $\mathcal{F}$ given Z, which is in fact the solution to (2), then we have $\sum_{v \in V} \mathbb{1}\{\widehat{Y}_v \neq Y_v\} \leq O(\log(|\mathcal{F}|/\delta)/\epsilon^2)$ with probability at least $1 - \delta$. Connectivity of G then implies $|\mathcal{F}| \approx (\frac{e}{p})^{2pn + \widetilde{O}(1)}$, giving the final $\widetilde{O}(pn)$ rate. Theorem 1 makes this result precise:

Theorem 1 (Inference in Trees)**.** *Let $\widehat{Y}$ be the solution to (2). Then with probability at least $1 - \delta$,*

$$\begin{aligned} \sum_{v \in V} \mathbb{1}\left\{\widehat{Y}_v \neq Y_v\right\} &\leq \frac{1}{\epsilon^2}(2pn + 2\log(2/\delta) + 1)\log(2e/p\delta) \\ &= \widetilde{O}(pn). \end{aligned} \tag{4}$$

We emphasize that side information is critical in this result. For trees — in particular, the path graph — no estimator can achieve below $\Omega(n)$ hamming error unless $p = O(1/n)$ (Globerson et al., 2014).

3 Inference for General Graphs

3.1 Upper Bound: Inference with Tree Decompositions

Our main algorithm, TreeDecompositionDecoder (Algorithm 1) produces estimators for Model 1 for

[3] See appendix for constants.

graphs G that admit a *tree decomposition* in the sense of Robertson and Seymour (Robertson and Seymour (1986)). Recall that a tree decomposition for a graph $G = (V, E)$ is new graph $T = (\mathcal{W}, F)$ in which each node in $\mathcal{W}$ corresponds to a subset of nodes in the original graph G. The edge set F forms a tree over $\mathcal{W}$ and must satisfy a property known as *coherence*, which guarantees that the connectivity structure of T captures that of G. The approach of TreeDecompositionDecoder is to use the edge observations X to produce a *local* estimator for each component of the tree decomposition T, then use the vertex observations Z to combine the many local estimators into a single *global* estimator.

Tree decompositions have found extensive use in algorithm design and machine learning primarily for computational reasons: These objects allow one to lift algorithmic techniques that are only feasible computationally on constant-sized graphs, such as brute force enumeration, into algorithms that run efficiently on graphs of all sizes. It is interesting to note that our algorithm obeys this principle, but for *statistical performance* in addition to computational performance: We are able to lift an analysis technique that is only tight for constant-sized graphs, the union bound, into an analysis that is tight for arbitrarily large graphs from families such as grids. However, as our analysis for trees shows, this approach is only made possible by the side information Z.

The *width* $\mathsf{wid}(T)$ of a tree decomposition T is the size of the largest component in T, minus one (by convention). To place a guarantee on the performance of TreeDecompositionDecoder, both statistically and computationally, it is critical that the width be at most logarithmic in n. At first glance this condition may seem restrictive there are graphs of interests such as grids for which the *treewidth* $\mathsf{tw}(G)$ — the smallest treewidth of *any* tree decomposition — is of order $\sqrt{n}$. For such graphs, our approach is to choose a subset $E' \subseteq E$ of edges to probe so that the graph $G' = (V, E')$ has small treewidth. For all of the graphs we consider this approach obtains optimal sample complexity in spite of discarding information.

Having found a decomposition of small treewidth for G' we apply the following algorithm. For each component of this decomposition, we compute the maximum likelihood estimator for the labels in this component given the edge measurements X. This is done by brute-force enumeration over vertex labels, which can be done efficiently because we require small treewidth. For a given component, there will be two estimators that match the edges in that component equally well due to sign ambiguity. The remaining problem is to select a set of signs — one for each component — so that the local estimators agree globally. For this task we leverage the side information Z. Our approach will mirror that of Section 2: To produce a global prediction $\widehat{Y}$ we solve a global optimization problem over the tree decomposition using dynamic programming, then analyze the statistical performance of $\widehat{Y}$ using statistical learning theory.

Informally, if there is some Δ such that we can show a p^{Δ} failure probability for estimating up to sign the vertex labels within each component of the tree decomposition, the prediction produces by Algorithm 1 will attain a high probability $p^{\Delta}n$ Hamming error bound for the entire graph. For example, in Section 4 we show a p^2 failure probability for estimating vertex labels in a grid of size 3×2, which through Algorithm 1 translates to a $O(p^2 n)$ rate with high probability on both $\sqrt{n} \times \sqrt{n}$ and $3 \times n/3$ grids.

Definition 1 (Cowell et al. (2006))**.** *A tree $T = (\mathcal{W}, F)$ is a tree decomposition for $G = (V, E)$ if it satisfies*

1. ***Vertex Inclusion:*** *Each node in $v \in V$ belongs to at least one component $W \in \mathcal{W}$.*
2. ***Edge Inclusion:*** *For each edge $(u, v) \in E$, there is some $W \in \mathcal{W}$ containing both u and v.*
3. ***Coherence:*** *Let $W_1, W_2, W_3 \in \mathcal{W}$ with W_2 on the path between W_1 and W_3 in T. Then if $v \in V$ belongs to W_1 and W_3, it also belongs to W_2.*

We assume without loss of generality that T is not redundant, i.e. there is no $(W, W') \in F$ with $W' \subseteq W$.

The next definition concerns the subsets of the graph G used in the local inference procedure within Algorithm 1. We allow the local maximum likelihood estimator for a component W to consider a superset of nodes, Extend(W), whose definition will be specialized to different classes of graphs.

Definition 2 (Component Extension Function)**.** *For a given $W \in \mathcal{W}$, the* extended component *$W^{\star} \supseteq W$ denotes the result of* Extend(W).

Choices we will use for the extension function include the identity $\textsc{Extend}(W) = W$ and the neighborhood of W with respect to the probed graph:

$$\textsc{Extend}(W) = \left(\bigcup_{v \in W} N_{G'}(v) \right) \cup W. \tag{5}$$

Concrete instantiations of Extend are given in Section 4.

We define quantitative properties of the tree decomposition in Table 1. For a given property, the corresponding ($\star$) version will denote the analogue the arises in analyzing performance when using extended components.

For simplicity, the reader may wish to imagine each $(\star)$ property as the corresponding non-$(\star)$ property on their first read-through.

Definition 3 (Admissible Tree Decomposition)**.** *Call a tree decomposition $T = (\mathcal{W}, F)$* admissible *if it satisfies the following properties:*

- $\mathsf{deg}(T)$, $\mathsf{deg}^\star_E(T)$, $\max_{W\in\mathcal{W}}|E(W^\star)|$, *and* $\mathsf{wid}^\star(T)$ *are constant.*
- $G'(W^\star)$ *is connected for all* $W \in \mathcal{W}$[4].

In the rest of this section, the $\widetilde{O}$ notation will hide all of the constant quantities from Definition 3.

Theorem 2 (Main Theorem)**.** *Let $\widehat{Y}$ be the labeling produced using Algorithm 1 with an admissible tree decomposition. Then, with high probability over the draw of X and Z,*

$$\sum_{v\in V}\mathbb{1}\left\{\widehat{Y}_v \neq Y_v\right\} \le \widetilde{O}\left(\sum_{W\in\mathcal{W}} p^{\lceil \mathsf{mincut}^\star(W)/2\rceil}\right). \qquad (6)$$

In particular, let Δ be such that $\Delta \le \mathsf{mincut}^\star(W)$ for all $W \in \mathcal{W}$. Then, with high probability,

$$\sum_{v\in V}\mathbb{1}\left\{\widehat{Y}_v \neq Y_v\right\} \le \widetilde{O}\left(p^{\lceil\Delta/2\rceil}n\right). \qquad (7)$$

Algorithm 1 runs in time $\widetilde{O}(\lceil p^{\Delta/2}n\rceil^2 n)$ for general tree decompositions and time $\widetilde{O}(\lceil p^{\Delta/2}n\rceil n)$ when T is a path graph.

3.2 Main theorem: Proof sketch

Let us sketch the analysis of Theorem 2 in the simplest case, where $\textsc{Extend}(W) = W$ for all $W \in \mathcal{W}$ and consequently all $(\star)$ properties are replaced with their non-$(\star)$ counterparts. We give a bound begin by bounding that probability that a single component-wise estimator $\widehat{Y}^W$ computed on line 5 of Algorithm 1 fails to exactly recover the ground truth within its component.

Definition 4 (Component Estimator)**.** *The (edge) maximum likelihood estimator for W is given by*

$$\widehat{Y}^W \triangleq \mathop{\arg\min}_{\widehat{Y}\in\{\pm1\}^W} \sum_{uv\in E'(W)} \mathbb{1}\{\widehat{Y}_u\widehat{Y}_v \neq X_{uv}\}. \qquad (8)$$

$\widehat{Y}^W$ can be computed by enumeration over all labelings in time $2^{|W|}$. There are always two solutions to (8) due to sign ambiguity; we take one arbitrarily.

Algorithm 1 $\textsc{TreeDecompositionDecoder}$

Parameters: Graph $G = (V, E)$. Probed edges $E' \subseteq E$. Extension function $\textsc{Extend}$.
Tree decomposition $T = (\mathcal{W}, F)$ for (V, E'). Failure probability $\delta > 0$.
Input: Edge measurements $X \in \{\pm1\}^E$. Vertex measurements $Z \in \{\pm1\}^V$.

```
 1: procedure TreeDecompositionDecoder
    Stage 1
    /* Compute estimator for each tree
    decomposition component. */
 2:   for $W \in \mathcal{W}$ do
 3:     $W^\star \leftarrow \textsc{Extend}(W)$.
                   // See Definition 2.
 4:     $\widetilde{Y}^{W^\star} \leftarrow \arg\min_{\widetilde{Y}\in\{\pm1\}^{W^\star}} \sum_{uv\in E'(W^\star)} \mathbb{1}\{\widetilde{Y}_u\widetilde{Y}_v \neq X_{uv}\}$.
 5:     Let $\widehat{Y}^{W^\star}$ be the restriction of $\widetilde{Y}^{W^\star}$ to $W$.
 6:   end for
    Stage 2
    /* Use component estimators to assign
    edge costs to tree decomposition. */
 7:   for $W \in \mathcal{W}$ do
 8:     $\mathrm{Cost}_W[+1] \leftarrow \sum_{v\in W} \mathbb{1}\{\widehat{Y}^{W^\star}_v \neq Z_v\}$
 9:     $\mathrm{Cost}_W[-1] \leftarrow \sum_{v\in W} \mathbb{1}\{-\widehat{Y}^{W^\star}_v \neq Z_v\}$.
10:   end for
11:   for $(W_1, W_2) \in F$ do
12:     Let $v \in W_1 \cap W_2$.
13:     $S(W_1, W_2) \leftarrow \widehat{Y}^{W_1^\star}_v \cdot \widehat{Y}^{W_2^\star}_v$.
14:   end for
    /* Run tree inference algorithm from
    Section 2 over tree decomposition. */
15:   $\hat{s} \leftarrow \textsc{TreeDecoder}(T, \mathrm{Cost}, S, L_n)$.
                   // See eq. (18) for constant $L_n$.
16:   for $v \in V$ do
17:     Choose arbitrary $W$ s.t. $v \in W$
         and set $\widehat{Y}_v \leftarrow \hat{s}_W\widehat{Y}^{W^\star}_v$.
18:   end for
19: return $\widehat{Y}$.
20: end procedure
```

Proposition 1 (Error Probability for Component Estimator)**.**

$$\mathbb{P}\left(\min_{s\{\pm1\}} \mathbb{1}\{s\widehat{Y}^W \neq Y^W\} > 0\right) \le \widetilde{O}(p^{\lceil \mathsf{mincut}(W)/2\rceil})$$

Proof. Assume that both $\widehat{Y}^W$ and $-\widehat{Y}^W$ disagree with

[4] Together with our other assumptions, this implies the *connected treewidth* of G' (Diestel and Müller, 2016) is constant.

Table 1: Tree decomposition properties.

$\mathsf{deg}(T) = \max_{W\in\mathcal{W}}\lvert\{(W,W')\in F\}\rvert$	
$\mathsf{wid}(T) = \max_{W\in\mathcal{W}}\lvert W\rvert - 1$	$\mathsf{wid}^\star(T) = \max_{W\in\mathcal{W}}\lvert W^\star\rvert - 1$
$\mathcal{W}(e) = \{W\in\mathcal{W} \mid e\in E(W)\}$	$\mathcal{W}^\star(e) = \{W\in\mathcal{W} \mid e\in E(W^\star)\}$
$\mathsf{deg}_E(T) = \max_{e\in E}\lvert\mathcal{W}(e)\rvert$	$\mathsf{deg}_E^\star(T) = \max_{e\in E}\lvert\mathcal{W}^\star(e)\rvert$
$\mathsf{mincut}(W) = \min_{S\subset W, S\neq\emptyset}\lvert\delta_{G(W)}(S)\rvert$	$\mathsf{mincut}^\star(W) = \min_{S\subset W^\star, S\cap W\neq\emptyset, \bar{S}\cap W\neq\emptyset}\lvert\delta_{G(W)}(S)\rvert$

the ground truth or else we are done. Let S be a maximal connected component of the set of vertices v for which $\widehat{Y}_v^W \neq Y_v$. It must be the case that at least $\lceil|\delta(S)|/2\rceil$ edges (u,v) in $\delta(S)$ have X_{uv} flipped from the ground truth, or else we could flip all the vertices in S to get a new estimator that agrees with X better than $\widehat{Y}$; this would be a contradiction since $\widehat{Y}$ minimizes $\sum_{uv\in E'(W)} \mathbb{1}\{\widehat{Y}_u\widehat{Y}_v \neq X_{uv}\}$. Applying a union bound, the failure probability is bounded by

$$\sum_{S\subseteq W: S\neq\emptyset, S\neq W} p^{\lceil|\delta(S)/2|\rceil} \leq 2^{|W|}p^{\lceil \mathsf{mincut}(W)/2\rceil}.$$

□

Proposition 1 bounds the probability of failure for *individual components*, but does not immediately imply a bound on the total number of components that may fail for a given realization of X. If the components $\mathcal{W}$ did not overlap one could apply a Chernoff bound to establish such a result, as their predictions would be independent. Since components can in fact overlap their predictions are dependent, but using a sharper concentration inequality (from the *entropy method* (Boucheron et al., 2003)) we can show that — so long as no edge appears in too many components — an analogous concentration result holds and total number of components failures is close to the expected number with high probability.

Lemma 1 (Informal)**.** With high probability over the draw of X,

$$\min_{s\in\{\pm1\}^{\mathcal{W}}} \sum_{W\in\mathcal{W}} \mathbb{1}\{s_W\widehat{Y}^W \neq Y^W\} \leq \widetilde{O}\left(\sum_{W\in\mathcal{W}} p^{\lceil \mathsf{mincut}(W)/2\rceil}\right). \tag{9}$$

In light of (9), consider the signing of the component-wise predictions $(\widehat{Y}^W)$ that best matches the ground truth.

$$s^\star = \operatorname*{arg\,min}_{s\in\{\pm1\}^{\mathcal{W}}} \sum_{W\in\mathcal{W}} \mathbb{1}\{s_W\widehat{Y}^W \neq Y^W\}.$$

If we knew the value of $s^\star$ we could use it to produce a vertex prediction with a Hamming error bound matching (6). Computing $s^\star$ is not possible because we do not have access to Y. We get the stated result by proceeding in a manner similar to the algorithm (2) for the tree. We first define a class $\mathcal{F}\subseteq\{\pm1\}^{\mathcal{W}}$ which has the property that 1) $s^\star\in\mathcal{F}$ with high probability and 2) $|\mathcal{F}| \lesssim 2^{\widetilde{O}(\sum_{W\in\mathcal{W}} 2^{|W|}p^{\lceil \mathsf{mincut}(W)/2\rceil})}$. Then we take the component labeling $\hat{s}$ is simply the element of $\mathcal{F}$ that is most correlated with the vertex observations Z: $\hat{s} = \operatorname{arg\,min}_{s\in\mathcal{F}} \sum_{W\in\mathcal{W}}\sum_{v\in W} \mathbb{1}\left\{s_W\widehat{Y}_v^W \neq Z_v\right\}$ (this is line 15 of Algorithm 1). Finally, to produce the final prediction $\widehat{Y}_v$ for a given vertex v, we find $W\in\mathcal{W}$ with $v\in W$ and take $\widehat{Y}_v = \hat{s}_W\cdot\widehat{Y}_v^W$. A generalization bound from statistical learning theory then implies that this predictor enjoys error at most $\widetilde{O}(\log|\mathcal{F}|) = \widetilde{O}\big(\sum_{W\in\mathcal{W}} 2^{|W|}p^{\lceil \mathsf{mincut}(W)/2\rceil}\big)$, which establishes the main theorem.

Efficient implementation Both the tree algorithm and Algorithm 1 rely on solving a constrained optimization problem of the form (2). In Appendix D we show how to perform this procedure efficiently using a message passing scheme.

3.3 Lower Bounds: General Tools

In this section we state simple lower bound techniques for Model 1. Recall that we consider q as a constant, and thus we are satisfied with lower bounds that coincide with our upper bounds up to polynomial dependence on q.

Theorem 3. *Assume $p < q$. Then any algorithm for Model 1 incurs expected hamming error $\Omega(\sum_{v\in V} p^{\lceil \mathsf{deg}(v)/2\rceil})$.*

Corollary 1. Any algorithm for Model 1 incurs expected hamming error $\Omega(p^{\Delta_{\text{avg}}/2+1}n)$.

Theorem 4. *Let $\mathcal{W}$ be a collection of disjoint constant-sized subsets of V. Then for all p below some constant, any algorithm for Model 1 incurs expected Hamming error $\Omega(\sum_{W\in\mathcal{W}} p^{\lceil|\delta_G(W)|/2\rceil})$.*

4 Concrete Results for Specific Graphs

We now specialize the tools developed in the previous section to provide tight upper and lower bounds on recovery for concrete classes of graphs.

4.1 Connected Graphs

Example 1 (Arbitrary graphs)**.** *For any connected graph G, the following procedure attains an error rate of $\widetilde{O}(pn)$ with high probability:* **1.** *Find a spanning tree T for G.* **2.** *Run the algorithm from Section 2 on T.*

This rate is sharp, in the sense that there are connected graphs — in particular, all trees — for which $\Omega(pn)$ Hamming error is optimal. Furthermore, for all graphs one can attain an estimator whose Hamming error is bounded as $\widetilde{O}(pn + \#\text{connected components})$ by taking a spanning tree for each component. This bound is also sharp.

The next example shows that there are connected graphs beyond trees for which $\Omega(pn)$ Hamming error is unavoidable. More generally, $\Omega(pn)$ Hamming error is unavoidable for any graph with a linear number of degree-2 vertices.

Looking at Theorem 3, one might be tempted to guess that the correct rate for inference is determined entirely by the degree profile of a graph. This would imply, for instance, that for any d-regular graph the correct rate is $\Theta(p^{\lceil d/2 \rceil}n)$. The next example — via Theorem 4 — shows that this is not the case.

Example 2. *For any constant d, there exists a family of d-regular graphs on n vertices for which no algorithm in Model 1 attains lower than $\Omega(pn)$ Hamming error.*

This construction for $d = 3$ is illustrated in Figure 1. We note that this lower bound hides a term of order $q^{\Omega(d)}$, but for constant q and d it is indeed order $\Omega(pn)$.

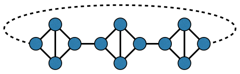
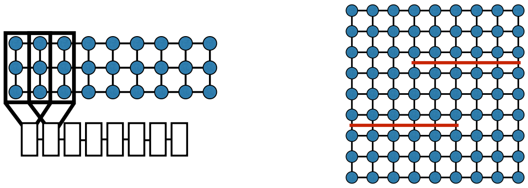


Figure 1: 3-regular graph for which $O(pn)$ error rate is optimal.

4.2 Grid Lattices



(a) Tree decomposition for $3 \times n/3$ grid.

(b) E' for $\sqrt{n} \times \sqrt{n}$ grid.

Figure 2

In this section we illustrate how to use the tree-decomposition based algorithm, Algorithm 1, to obtain optimal rates for grid lattices.

Example 3 (2-dimensional grid)**.** *Let G be a 2-dimensional grid lattice of size $c \times n/c$ where $c \leq \sqrt{n}$. For grid of height $c = 3$ (or above) using Algorithm 1, we obtain an estimator $\widehat{Y}$ such that with high probability, the Hamming error is bounded as $O(p^2 n)$. This estimator runs in time $O(\lceil p^2 n \rceil n)$, By the degree profile argument (also given in Globerson et al. (2015)), there is a matching lower bound of $\Omega(p^2 n)$. For a grid of height $c = 1$ there is an obvious lower bound of $\Omega(pn)$ since this graph is a tree.*

The estimator of Globerson et al. (2015) can be shown to have expected Hamming error of $O(p^2 n)$ for the 2-dimensional grid with $c = \Omega(\log n)$. Our method works for *constant height grids* ($c = O(1)$) and with high probability.

Algorithm 1 of course requires a tree decomposition as input. The tree decomposition used to obtain Example 3 for constant-height grids is illustrated in Figure 2a for $c = 3$: The grid is covered in overlapping 3×2 components, and these are connected as a path graph to form the tree decomposition.

The reader will observe that this tree decomposition has $\mathsf{mincut}(W) = 2$, and so only implies a $O(pn)$ Hamming error bound through Theorem 2. This rate falls short of the $O(p^2 n)$ rate promised in the example; it is no better than the rate if G were a tree. The problem is that within each 3×2 block, there are four "corner" nodes each with degree 2. Indeed if either edge connected to a corner is flipped from the ground truth, which happens with probability p, this corner is effectively disconnected from the rest of W in terms of information. To sidestep this issue, we define $\textsc{Extend}(W) = \bigcup_{v \in W} N(v)$. With this extension, we have $\mathsf{mincut}^{\star}(W) = 3$ for all components except the endpoints, which implies the $O(p^2 n)$ rate.

Probing Edges We now illustrate how to extend the tree decomposition construction for constant-height grids to a construction for grids of arbitrary height. Recall that Algorithm 1 takes as input a subset $E' \subseteq E$ and a tree decomposition T for $G' = (V, E')$. To see where using only a subset of edges can be helpful consider Figure 2a and Figure 2b. The $3 \times n/3$ grid is ideal for our decoding approach because it can be covered in 3×2 blocks as in Figure 2a and thus has treewidth at most 5. The $\sqrt{n} \times \sqrt{n}$ grid is more troublesome because it has treewidth $\sqrt{n}$, but we can arrive at G' with constant treewidth by removing $\Theta(n)$ edges through the "zig-zagging" cut shown in Figure 2b. Observe that once the marked edges in Figure 2b are removed we can "unroll" the graph and apply a decomposition similar to Figure 2a.

The tree decomposition construction we have outlined for two-dimensional grids readily lifts to higher dimension. This gives rise to the next example.

Example 4 (Hypergrids and Hypertubes)**.** *Consider a three-dimensional grid lattice of of length n/c^2, height c, and width c. If $c = n^{1/3}$ — that is, we have a cube — then Algorithm 1 obtains Hamming error $\widetilde{O}(p^3 n)$ with high probability, which is optimal by Theorem 3.*

When c is constant, however, the optimal rate is $\Omega(p^2 n)$; this is also obtained by Algorithm 1. This contrasts the two-dimensional grid, where the optimal rate is the same for all $3 \leq c \leq \sqrt{n}$.

Algorithm 1 can be applied to any d-dimensional hypergrid of the form $c \times c \times \ldots n/c^{d-1}$ to achieve $O(p^d n)$ Hamming error when $c \approx n^{1/d}$. For constant c, the optimal rate is $\Theta(p^{\lceil \frac{d+1}{2} \rceil} n)$. More generally, the optimal rate interpolates between these extremes.

The next two examples briefly sketch how to apply tree decompositions to more lattices. Recall that the triangular lattice and hexagonal lattice are graphs whose drawings can be embedded in $\mathbb{R}^2$ to form regular triangular and hexagonal tilings, respectively.

Example 5 (Triangular Lattice)**.** *Consider a triangular lattice of height and width $\sqrt{n}$. Let each component to be a vertex and its 6 neighbors (except for the edges of the mesh), and choose these components such that the graph is covered completely. For a given component, let $W^\star$ to be the neighborhood of W. For this decomposition $\mathsf{mincut}^\star(W)$ is 6 and consequently Algorithm 1 achieves Hamming error $\widetilde{O}(p^3 n)$. This rate is optimal because all vertices in the graph have degree 6 besides those at the boundary, but the number of vertices on the boundary is sub-linear.*

The triangular lattice example in particular shows that there exist graphs of average degree 3 for which an error rate of $O(p^2 n)$ is achievable.

Example 6 (Hexagonal Lattice)**.** *Consider a $\sqrt{n} \times \sqrt{n}$ hexagonal lattice. Take each component W to be a node v and its neighbors, and choose the nodes v so that the graph is covered. Choose $W^\star$ to be the neighborhood of the component W. The value of $\mathsf{mincut}^\star(W)$ for each component is 3, leading to a Hamming error rate of $\widetilde{O}(p^2 n)$. This rate is optimal because all vertices on the lattice except those at the boundary have degree 3.*

4.3 Newman-Watts Model

To define the Newman-Watts small world model (Newman and Watts, 1999), we first define the regular ring lattice, which serves as the base graph for this model. The *regular ring lattice* $R_{n,k}$ is a $2k$-regular graph on n vertices defined as follows: 1) $V = \{1, \ldots, n\}$. 2) $E = \{(i,j) \mid j \in \{i+1, \ldots, i+k \pmod{n}\}\}$. Theorem 3 immediately implies that the best rate possible in this model is $\Omega(p^k n)$. Using Algorithm 1 with an appropriate decomposition it is indeed possible to achieve this rate.

Example 7. *The optimal Hamming rate for $R_{n,k}$ in Model 1 is $\widetilde{\Theta}(p^k n)$. Moreover, this rate is achieved by an efficiently by Algorithm 1 in time $O(\lceil p^k n \rceil n)$.*

Note that for constant k, $R_{n,k}$ does not have the weak expansion property, and so the algorithm of Globerson et al. (2015) does not apply. We can now move on to the Newman-Watts model:

Definition 5 (Newman-Watts Model)**.** *To produce a sample from the Newman-Watts model $H_{n,k,\alpha}$, begin with $R_{n,k}$, then independently replace every non-edge with an edge with probability α/n.*

For any constant $\alpha < 1$, a constant fraction of the vertices in $R_{n,k}$ will be untouched in $H_{n,k,\alpha}$. Thus, the inference lower bound for Example 7 still applies, meaning that the optimal rate is $O(p^k n)$. Algorithmically, this result can be obtained by discarding the new edges and using the same decomposition as in Example 7.

Example 8. *For any $\alpha < 1$, the optimal Hamming rate for $H_{n,k,\alpha}$ in Model 1 is $\widetilde{\Theta}(p^k n)$. Moreover, this rate is achieved in time $O(\lceil p^k n \rceil n)$ by Algorithm 1 .*

5 Discussion

We considered Model 1, introduced in Globerson et al. (2015), for approximately inferring the ground truth labels for nodes of a graph based on noisy edge and vertex labels. We provide a general method to deal with arbitrary graphs that admit small width tree decompositions of (edge)-subgraphs. As a result, we recover the results in Globerson et al. (2015) for grids, and are able to provide rates for graphs that do not satisfy the weak expansion property which is needed for the proof techniques in Globerson et al. (2015). Furthermore, in contrast to most existing work, we demonstrate that recovery tasks can be solved even on sparse "nonexpanding" graphs such as trees and rings.

There are several future directions suggested by this work. Currently, it is a nontrivial task to characterize the optimal error rate achievable for a given graph, and it is unclear how to extend our methods to families beyond lattices and graphs of small treewidth. Exploring connections between further graph parameters and achievable error rates is a compelling direction, as our understanding of optimal sample complexity remains quite limited. The challenge here entails both finding what can be done information-theoretically, as well as understanding what recovery rates can be obtained efficiently.

References

Emmanuel Abbe, Afonso S Bandeira, Annina Bracher, and Amit Singer. Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery. *Network Science and Engineering, IEEE Transactions on*, 1(1):10–22, 2014.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. Concentration inequalities using the entropy method. *Annals of Probability*, pages 1583–1614, 2003.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced lectures on machine learning*, pages 169–207. Springer, 2004.

Yuri Boykov and Olga Veksler. Graph cuts in vision and graphics: Theories and applications. In *Handbook of mathematical models in computer vision*, pages 79–96. Springer, 2006.

Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of inference in graphical models. *arXiv preprint arXiv:1206.3240*, 2012.

Yuxin Chen and Andrea J Goldsmith. Information recovery from pairwise measurements. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 2012–2016. IEEE, 2014.

Yuxin Chen, Govinda Kamath, Changho Suh, and David Tse. Community recovery in graphs with locality. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 689–698, 2016.

Robert G Cowell, Philip Dawid, Steffen L Lauritzen, and David J Spiegelhalter. *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer Science & Business Media, 2006.

Reinhard Diestel and Malte Müller. Connected treewidth. 2016.

Abraham D Flaxman. Expansion and lack thereof in randomly perturbed graphs. *Internet Mathematics*, 4(2-3):131–147, 2007.

Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

Amir Globerson, Tim Roughgarden, David Sontag, and Cafer Yildirim. Tight error bounds for structured prediction. *arXiv preprint arXiv:1409.5834*, 2014.

Amir Globerson, Tim Roughgarden, David Sontag, and Cafer Yildirim. How hard is inference for structured prediction? *Proceedings of The 32nd International Conference on Machine Learning*, pages 2181–2190, 2015.

Bruce Hajek, Yihong Wu, and Jiaming Xu. Computational lower bounds for community detection on random graphs. *arXiv preprint arXiv:1406.6625*, 2014.

Thorsten Joachims and John E Hopcroft. Error bounds for correlation clustering. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*, pages 385–392, 2005.

Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.

Nikos Komodakis and Georgios Tziritas. Approximate labeling via graph cuts based on linear programming. *IEEE transactions on pattern analysis and machine intelligence*, 29(8):1436–1453, 2007.

Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Correlation clustering with noisy partial information. In *Proceedings of The 28th Conference on Learning Theory*, pages 1321–1342, 2015.

Mark EJ Newman and Duncan J Watts. Renormalization group analysis of the small-world network model. *Physics Letters A*, 263(4):341–346, 1999.

Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.

Alaa Saade, Florent Krzakala, Marc Lelarge, and Lenka Zdeborova. Spectral detection in the censored block model. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 1184–1188. IEEE, 2015.

Nicol N Schraudolph and Dmitry Kamenetsky. Efficient exact inference in planar ising models. In *Advances in Neural Information Processing Systems*, pages 1417–1424, 2009.

David Sontag, Talya Meltzer, Amir Globerson, Tommi S Jaakkola, and Yair Weiss. Tightening lp relaxations for map using message passing. *arXiv preprint arXiv:1206.3288*, 2012.

Olga Veksler. Efficient graph-based energy minimization methods in computer vision. 1999.