A Universal Low Complexity Compression Algorithm for Sparse Marked Graphs

Payam Delgosha
EECS Department
University of California, Berkeley
Berkeley, CA 94720
Email: pdelgosha@eecs.berkeley.edu

Venkat Anantharam
EECS Department
University of California, Berkeley
Berkeley, CA 94720
Email: ananth@eecs.berkeley.edu

***Abstract*—Many modern applications involve accessing and processing graphical data, i.e. data that is naturally indexed by graphs. Examples come from internet graphs, social networks, genomics and proteomics, and other sources. The typically large size of such data motivates seeking efficient ways for its compression and decompression. The current compression methods are usually tailored to specific models, or do not provide theoretical guarantees. In this paper, we introduce a low–complexity lossless compression algorithm for sparse marked graphs, i.e. graphical data indexed by sparse graphs, which is capable of universally achieving the optimal compression rate in a precisely defined sense. In order to define universality, we employ the framework of local weak convergence, which allows one to make sense of a notion of stochastic processes for graphs. Moreover, we investigate the performance of our algorithm through some experimental results on both synthetic and real–world data.**

I. Introduction

Nowadays, a large amount of data arises in a form that is indexed by combinatorial structures, such as graphs, rather than classical time series. Examples include data arising from internet graphs, from social networks, and various kinds of biological data. Such data typically needs to be mined for practical reasons, for instance for inferring community membership in a social network, or in predicting whether two proteins interact in a biological network. Typically, the size of such graphical data is large, which argues for the need to find efficient and close to optimal ways of compressing and decompressing this data to store it for subsequent data mining.

The problem of graph compression has drawn a lot of attention in different fields. The existing algorithms for compressing internet graphs and social networks usually rely on some properties specific to such graphs [1], [2], [3]. Moreover, there has been some progress in finding best compression rates assuming that the graphical data is being generated from a statistical model [4], [5], [6], [7]. The key property distinguishing our approach from the existing ones is *universality*. More precisely, we introduce a scheme which is capable of compressing graphs which come from a certain "stochastic process" without any prior knowledge of this process, yet is able to achieve the optimal compression rate, in a precisely defined sense. Additionally, in contrast to several earlier works, we assume that the graphs are "marked" so that vertices and edges contain additional information on top of the connectivity structure of the graph. This is the key feature which makes our approach applicable in modeling real–world data indexed by graphs, which we call *graphical data*. We focus in this paper on sparse marked graphs, the motivation for this being that it is generally recognized that most real–world graphs are sparse.

To make sense of the notion of a "stochastic process" for sparse marked graphs, we employ the framework of local weak convergence [8], [9], [10]. Moreover, we employ a notion of entropy called the *marked BC entropy* [11], [12], which serves as a counterpart of the Shannon entropy rate for this framework, and governs the optimal compression rate for graphical data on sparse graphs. The authors have already introduced a universal compression scheme in [13] which shows that this notion of entropy is indeed the optimal information theoretic threshold of compression. The focus of this paper is to provide a version of such a scheme which is also computationally efficient. In [13], the encoder needs to find the index of the input graph among all graphs which have the same frequency of local structures as the input graph. However, in this paper, we exploit the properties of the marked BC entropy to partition the edges in the graph based on their types, and encode each group separately, so as to obtain a low–complexity compression scheme.

The structure of this paper is as follows. In Section II, we introduce our notation. In Sections III and IV we briefly introduce the local weak convergence framework and the marked BC entropy, respectively. With this behind us, we formally introduce the main properties of our compression algorithm in Section V. In Section VI, we highlight the steps of our algorithm. Finally, in Section VII, we describe some experimental results on both synthetic and real–world data to illustrate the value of our framework.

II. Notation

Table I provides a list of our basic notation. A simple marked graph is a simple graph such that every vertex carries a mark coming from a fixed finite vertex mark set Θ. Moreover, every edge carries two marks, one towards each of its endpoints, coming from a fixed and finite edge mark set Ξ. For an edge (v, w) in a simple marked graph G, we denote the marks of this edge towards v and w by

978-1-7281-6432-8/20/$31.00 ©2020 IEEE

ISIT 2020

Authorized licensed use limited to: Univ of Calif Berkeley. Downloaded on August 21,2021 at 17:07:24 UTC from IEEE Xplore. Restrictions apply.

Notation	Meaning
$[n]$	$\{1,\dots,n\}$
$\log(.)$	logarithm in natural basis
$\{0,1\}^*$	the set of finite nonempty sequences of 0 and 1's
$\mathsf{nats}(x)$	length of $x \in \{0,1\}^*$ in nats: $\log 2 \times$length of x
Θ	fixed finite set of vertex marks
Ξ	fixed finite set of edge marks
$\xi_G(v,w)$	$\in \Xi$; mark of edge (v,w) in G towards w
$\deg_G(v)$	degree of node v in G
$v \sim_G w$	edge exists between nodes v and w in G

TABLE I: List of basic notation

Notation	Meaning
$[G,o]$	isomorphism class of G rooted at o
$[G,o]_h$	isomorphism class of G rooted at o up to depth h
$\bar{\mathcal{G}}_*$	set of isomorphism classes $[G,o]$
$\bar{\mathcal{G}}_*^h$	consisting of $[G,o] \in \bar{\mathcal{G}}_*$ with depth at most h
$\deg(\mu)$	average degree at the root for measure μ on $\bar{\mathcal{G}}_*$
$U(G)$	law of $[G,o]$, o chosen u.a.r. in G
$G[v,w]$	$\in \Xi \times \bar{\mathcal{G}}_*$; the pair $(\xi_G(v,w),[G',w])$
$\bar{\mathcal{T}}_*$	subset consisting of $[T,o] \in \bar{\mathcal{G}}_*$, where T is a tree
$\bar{\mathcal{T}}_*^h$	subset consisting of $[T,o] \in \bar{\mathcal{G}}_*^h$, where T is a tree
$\Sigma(\mu)$	marked BC entropy of probability measure μ on $\bar{\mathcal{G}}_*$

TABLE II: Summary of notation in Sections III and IV

and $\xi_G(v,w)$, respectively. The degree of a vertex v in G is denoted by $\deg_G(v)$. We denote the existence of an edge between two vertices v and w in a graph G by $v \sim_G w$. All graphs in this paper are simple, hence we may drop the qualifier "simple" when we discuss graphs.

III. The framework of Local Weak Convergence

We now briefly review the framework of local weak convergence, which makes sense of the notion of a stochastic processes for sparse marked graphs. The reader is referred to [8], [9], [10] for more details.

Fix vertex and edge mark sets Θ and Ξ and let G be a marked graph together with a vertex o in G. Let $[G,o]$ denote the isomorphism class of the connected component of o in G, rooted at o, where isomorphism is supposed to preserve connectivity and the root as well as the vertex and edge marks. Moreover, for $h \geq 0$, we let $[G,o]_h$ be the isomorphism class corresponding to the subgraph of G consisting of vertices with distance at most h from o, rooted at o. Let $\bar{\mathcal{G}}_*$ denote the space of isomorphism classes $[G,o]$ of connected marked graphs on a finite or countable vertex set, rooted at o, such that all degrees in G are finite. For $h \geq 0$, let $\bar{\mathcal{G}}_*^h$ be the subset of $\bar{\mathcal{G}}_*$ consisting of isomorphism classes of marked rooted graphs with depth no more than h. For a probability measure μ on $\bar{\mathcal{G}}_*$, let $\deg(\mu)$ denote the expected degree at the root in μ. For a finite marked graph G, let $U(G)$ denote the law of $[G,o]$, where o is chosen uniformly at random (u.a.r.) in G. We can think of $U(G)$ as the "empirical distribution" of G. Moreover, for two adjacent vertices v and w in G, let $G[v,w] \in \Xi \times \bar{\mathcal{G}}_*$ be the pair $(\xi_G(v,w),[G',w])$, where G' is the graph obtained from G by removing the edge (v,w).

We can equip $\bar{\mathcal{G}}_*$ with a metric so that the distance between two marked rooted graphs $[G,o]$ and $[G',o']$ is defined to be $1/(1+h_*)$, where h_* is the supremum over all h such that $[G,o]_h = [G',o']_h$. If no such h exists, we define the distance between $[G,o]$ and $[G',o']$ to be 1. One can show that $\bar{\mathcal{G}}_*$ equipped with this topology is a Polish space, i.e. it is a complete and separable metric space [10]. We say that a sequence of finite graphs G_n converges to a probability measure μ on $\bar{\mathcal{G}}_*$ if $U(G)$ converges weakly to μ. Not all probability measures on $\bar{\mathcal{G}}_*$ can show up as the limit of a sequence of finite graphs. For this, a necessary stationarity condition called "unimodularity" must hold for μ [10].

Let $\bar{\mathcal{T}}_*$ denote the subset of $[T,o] \in \bar{\mathcal{G}}_*$ where $[T,o]$ is the isomorphism class of a marked rooted tree. Likewise, we define $\bar{\mathcal{T}}_*^h$ to be the subset of $\bar{\mathcal{G}}_*^h$ consisting of isomorphism classes of marked rooted trees.

IV. The Marked BC Entropy

In this section, we introduce our notion of entropy, which is a generalization of the one introduced by Bordenave and Caputo in [11] to the marked regime discussed above. We call this notion the "marked BC entropy". This generalization is due to us, and the reader is referred to [12] for more details.

Let μ be a probability measure on $\bar{\mathcal{G}}_*$. For integer n and vectors $\vec{m}^{(n)} = (m^{(n)}(x,x') : x,x' \in \Xi)$ and $\vec{u}^{(n)} = (u^{(n)}(\theta) : \theta \in \Theta)$, let $\mathcal{G}^{(n)}_{\vec{m}^{(n)},\vec{u}^{(n)}}$ be the set of marked graphs on the vertex set $[n]$, with $u^{(n)}(\theta)$ many vertices with mark θ, and $m^{(n)}(x,x') = m^{(n)}(x',x)$ many edges with mark pair (x,x'). Moreover, for $\epsilon > 0$, let $\mathcal{G}^{(n)}_{\vec{m}^{(n)},\vec{u}^{(n)}}(\mu,\epsilon)$ be the subset of "ϵ–typical graphs", i.e. the set of graphs $G \in \mathcal{G}^{(n)}_{\vec{m}^{(n)},\vec{u}^{(n)}}$ such that $d_{\mathrm{LP}}(U(G),\mu) < \epsilon$, where d_{LP} denotes the Lévy–Prokhorov distance [14] on probability measures on $\bar{\mathcal{G}}_*$. Roughly speaking, it can be shown that, if for $x,x' \in \Xi$ $2m^{(n)}(x,x')/n$ is close to the expected number of the edges connected to the root in μ with mark pair (x,x') or (x',x) and, for $\theta \in \Theta$, $u^{(n)}(\theta)/n$ is close to the probability of the root in μ having mark θ, then we have

$$\lim_{\epsilon \downarrow 0} \lim_{n \to \infty} \frac{1}{n} \Big(\log |\mathcal{G}^{(n)}_{\vec{m}^{(n)},\vec{u}^{(n)}}(\mu,\epsilon)| - m_n \log n \Big) = \Sigma(\mu),$$

where $m_n = \sum_{x \in \Xi} m^{(n)}(x,x) + \frac{1}{2} \sum_{x \neq x' \in \Xi} m^{(n)}(x,x')$ is the total number of edges, and $\Sigma(\mu)$ is a constant depending only on μ, possibly $-\infty$, called the marked BC entropy of μ.

It can be shown [12] that the marked BC entropy $\Sigma(\mu)$ does not depend on the choice of vectors $\vec{m}^{(n)}$ or $\vec{u}^{(n)}$. Moreover, $\Sigma(\mu) = -\infty$ if μ is not unimodular, or the support of μ is not contained in $\bar{\mathcal{T}}_*$. Motivated by this, we restrict our analysis to unimodular probability measures μ on $\bar{\mathcal{T}}_*$. Table II summarizes the important notation in Sections III and IV.

V. Problem Statement and Main Results

Our compression algorithm has two positive integer parameters h and δ. The encoding function, $f^{(n)}_{h,\delta}$, maps marked graphs on the vertex set $[n]$ to $\{0,1\}^*$, and the decoding function, $g^{(n)}_{h,\delta}$, is such that $g^{(n)}_{h,\delta} \circ f^{(n)}_{h,\delta}$ is the identity map,

Authorized licensed use limited to: Univ of Calif Berkeley. Downloaded on August 21,2021 at 17:07:24 UTC from IEEE Xplore. Restrictions apply.

i.e. the compression scheme is lossless. We introduce such a compression/decompression scheme which has two properties: (1) it is universally optimal and (2) it is computationally efficient. The following theorem summarizes these properties. In Section VI, we highlight the steps of this algorithm.

Theorem 1: There exists a compression/decompression algorithm as above, which has the following properties:

1) **(Optimality)** Assume a unimodular probability measure μ on $\bar{\mathcal{T}}_*$ with $\deg(\mu) \in (0,\infty)$ is given such that $\mathbb{E}_\mu[\deg_T(o)\log\deg_T(o)] < \infty$ and $\Sigma(\mu) > -\infty$. Assume that a sequence $G^{(n)}$ of marked graphs is given such that $U(G^{(n)}) \Rightarrow \mu$ and, with $m^{(n)}$ being the total number of edges in $G^{(n)}$, we have $m^{(n)}/n \to \deg(\mu)/2$. Then, for $h \geq 1$ and $\delta \geq 1$, with $l(n,h,\delta) := (\mathsf{nats}(f^{(n)}_{h,\delta}(G^{(n)})) - m^{(n)}\log n)/n$, we have
$$\limsup_{h\to\infty}\limsup_{\delta\to\infty}\limsup_{n\to\infty} l(n,h,\delta) \leq \Sigma(\mu). \tag{1}$$
2) **(Computational Complexity)** Assuming that $m^{(n)} = O(n)$ and h, δ, $|\Xi|$, and $|\Theta|$ are all constants, the time and memory complexities of the compression and decompression algorithms are $O(n\mathrm{polylog}(n))$.

Remark 1: One can also incorporate $m^{(n)}$, h, δ, $|\Xi|$, and $|\Theta|$ in the complexity bounds. However, to simplify the discussion, we have only presented complexity bounds in terms of n.

Remark 2: As discussed in [13], a matching converse argument suggests that $\Sigma(\mu)$ is the best compression rate. Also, as the leading term in the marked BC entropy is $m^{(n)}\log n$, which scales as $n\log n$, there is a computational lower bound of $\Omega(n\log n)$. Hence, our algorithm is computationally optimal up to logarithmic factors.

VI. Steps of the Universal Compression Algorithm

We now highlight the steps of our universal compression algorithm. To simplify the discussion, we only present the complexities in terms of n, the number of vertices, assuming that $m^{(n)} = O(n)$, and δ, h, $|\Xi|$, and $|\Theta|$ are constants. For $v \in [n]$, let $\theta_v^{(n)}$ denote the mark of v in $G^{(n)}$, and let $d_v^{(n)} := \deg_{G^{(n)}}(v)$. Also, let $\gamma^{(n)}_{v,1} < \cdots < \gamma^{(n)}_{v,d_v^{(n)}}$ denote the list of neighbors of v. Algorithm 1 highlights the steps of the compression algorithm discussed below. Also, Table III summarizes the main notation in this section.

Notation	Meaning
$d_v^{(n)}$ and $\theta_v^{(n)}$	degree of v and mark of v, resp.
$\gamma^{(n)}_{v,i}$	ith neighbor of v
$\mathcal{F}^{(\delta,h)}$	elements in $\Xi \times \bar{\mathcal{T}}_*^{h-1}$ with bounded degrees
$\star_x$ for $x \in \Xi$	auxiliary element not present in $\mathcal{F}^{(\delta,h)}$
$\bar{\mathcal{F}}^{(\delta,h)}$	$\mathcal{F}^{(\delta,h)} \cup \{\star_x : x \in \Xi\}$
$\mathsf{UC}_v(G)$	universal cover of G rooted at v
$t_h^{(n)}(v,w)$	$(\mathsf{UC}_v(G^{(n)}))[w,v]_{h-1}$
$\tilde{t}^{(n)}_{h,\delta}(v,w)$	$t_h^{(n)}(v,w)$ if (2) holds, $\star_{\xi_{G^{(n)}}(w,v)}$ o.t.w.
$\psi^{(n)}_{h,\delta}(v,w)$	$(\tilde{t}^{(n)}_{h,\delta}(v,w), \tilde{t}^{(n)}_{h,\delta}(w,v))$
$c_{v,i}$	integer representing $\tilde{t}^{(n)}_{h,\delta}(v,\gamma^{(n)}_{v,i})$
$\mathcal{V}_\star^{(n)}$	set of star vertices
$D^{(n)}_{t,t'}(v)$	$\lvert\{w \sim_{G^{(n)}} v : \psi^{(n)}_{h,\delta}(v,w) = (t,t')\}\rvert$

TABLE III: Summary of notation in Sections VI

Algorithm 1 Compression algorithm

Find $(c_{v,i} : v \in [n], i \in [d_v^{(n)}])$, TMark and TIsStar
for $1 \leq v \leq n$ **do**
  $y_v \leftarrow 0$
  for $1 \leq i \leq d_v^{(n)}$ **do**
    $(a,b) \leftarrow c_{v,i}$
    if $\mathsf{TIsStar}(a) = 1$ or $\mathsf{TIsStar}(b) = 1$ **then**
      $y_v \leftarrow 1$
    end if
  end for
end for
Encode $\vec{y} = (y_v : v \in [n])$ ▷ encode $\mathcal{V}_\star^{(n)}$
for $(x,x') \in \Xi \times \Xi$ **do** ▷ encode star edges
  for $v \in \mathcal{V}_\star^{(n)}$ **do**
    for $1 \leq i \leq d_v^{(n)}$ **do**
      $(a,b) \leftarrow c_{v,i}$
      if $\gamma^{(n)}_{v,i} > v$ **then**
        if $\mathsf{TIsStar}(a) = 1$ and $\mathsf{TIsStar}(b) = 1$ **then**
          $y \leftarrow \xi_{G^{(n)}}(\gamma^{(n)}_{v,i}, v)$, $y' \leftarrow \xi_{G^{(n)}}(v, \gamma^{(n)}_{v,i})$
          if $y = x$ and $y' = x'$ **then**
            Write a single bit with value 1
            Write $\gamma^{(n)}_{v,i}$ to the output
          end if
        end if
      end if
    end for
    Write 0 to the output ▷ we are done with v
  end for
end for
Encode $(\theta_v^{(n)}, D^{(n)}(v))$ for $1 \leq v \leq n$ ▷ vertex types
Find partition graphs $(G^{(n)}_{t,t'} : (t,t') \in \mathcal{E}^{(n)}_{\leq})$
for $(t,t') \in \mathcal{E}^{(n)}_{\leq}$ **do**
  Find integer representation of $G^{(n)}_{t,t'}$
  Write this representation to the output
end for

A. Definition of Edge Types

Let $\mathcal{F}^{(\delta,h)} \subset \Xi \times \bar{\mathcal{T}}_*^{h-1}$ be the set of all $(x,[T,o]) \in \Xi \times \bar{\mathcal{T}}_*^{h-1}$ such that $\deg_T(o) < \delta$ and $\deg_T(v) \leq \delta$ for $v \neq o$. Moreover, for $x \in \Xi$, let $\star_x$ be fictitious distinct elements not present in $\mathcal{F}^{(\delta,h)}$, and define $\bar{\mathcal{F}}^{(\delta,h)} := \mathcal{F}^{(\delta,h)} \cup \{\star_x : x \in \Xi\}$. Note that $\star_x$ for $x \in \Xi$ are auxiliary objects, and are not of the form of a pair of a mark and a rooted tree. For a marked graph G, we denote the universal cover of G rooted at v by $\mathsf{UC}_v(G)$. The mark component of $(x,[T,o]) \in \mathcal{F}^{(\delta,h)}$ is defined to be x, and the mark component of $\star_x$ is defined to be x. For adjacent vertices v and w in $G^{(n)}$, we

Authorized licensed use limited to: Univ of Calif Berkeley. Downloaded on August 21,2021 at 17:07:24 UTC from IEEE Xplore. Restrictions apply.

define $t_h^{(n)}(v,w) := (\mathsf{UC}_v(G^{(n)}))[w,v]_{h-1}$. Moreover, define $\tilde{t}_{h,\delta}^{(n)}(v,w)$ to be $t_h^{(n)}(v,w)$ if the following conditions hold:

$$\begin{aligned} t_h^{(n)}(v,w) \in \mathcal{F}^{(\delta,h)}, \qquad & t_h^{(n)}(w,v) \in \mathcal{F}^{(\delta,h)}, \\ \deg_{G^{(n)}}(v) \le \delta, \qquad & \deg_{G^{(n)}}(w) \le \delta. \end{aligned} \tag{2}$$

Otherwise, let $\tilde{t}_{h,\delta}^{(n)}(v,w) := \star_{\xi_{G^{(n)}}(w,v)}$. Note that the last two conditions in (2) automatically follow from the first two conditions when $h > 1$. However, this is not true when $h = 1$. Note that, by definition, $\tilde{t}_{h,\delta}^{(n)}(v,w) \in \bar{\mathcal{F}}^{(\delta,h)}$. We define the "type" of an edge (v,w) as $\psi_{h,\delta}^{(n)}(v,w) := (\tilde{t}_{h,\delta}^{(n)}(v,w), \tilde{t}_{h,\delta}^{(n)}(w,v))$.

B. Finding Edge Types

Next, we find $\psi_{h,\delta}^{(n)}(v,w)$ for adjacent vertices v and w in $G^{(n)}$. It can be shown that this can be done using a message passing algorithm, which returns an array $\vec{c} = (c_{v,i} : v \in [n], i \in [d_v^{(n)}])$ where $c_{v,i} = (a,b)$ is a pair of integers, such that a represents $\tilde{t}_{h,\delta}^{(n)}(v, \gamma_{v,i}^{(n)})$, and b represents $\tilde{t}_{h,\delta}^{(n)}(\gamma_{v,i}^{(n)}, v)$. In addition to this, the algorithm returns two arrays TMark and $\mathsf{TIsStar}$ such that, for an integer a, with $t \in \bar{\mathcal{F}}^{(\delta,h)}$ being the element in $\bar{\mathcal{F}}^{(\delta,h)}$ corresponding to a, $\mathsf{TMark}(a)$ is the mark component of t, and $\mathsf{TIsStar}(a)$ is 1 if t is of the form $\star_x$ for some $x \in \Xi$, and 0 otherwise. It can be shown that the time and memory complexities of this algorithm are $O(n\mathrm{polylog}(n))$.

C. Encoding Star Vertices and Star Edges

We call an edge (v,w) with type $(\star_x, \star_{x'})$ for some $x, x' \in \Xi$ a *"star edge"*. We call a vertex $v \in [n]$ a *"star vertex"* if it has at least one star edge connected to it. Let $\mathcal{V}_\star^{(n)}$ denote the set of star vertices in $G^{(n)}$. At this point, we encode $\mathcal{V}_\star^{(n)}$ and the star edges in $G^{(n)}$. Note that by definition, both endpoints of a star edge are in $\mathcal{V}_\star^{(n)}$. See Algorithm 1 for the details.

D. Encoding Vertex Types

For $1 \le v \le n$, define $D^{(n)}(v) := (D_{t,t'}^{(n)}(v) : t, t' \in \mathcal{F}^{(\delta,h)})$ so that for $t, t' \in \mathcal{F}^{(\delta,h)}$, $D_{t,t'}^{(n)} := |\{w \sim_{G^{(n)}} v : \psi_{h,\delta}^{(n)}(v,w) = (t,t')\}|$. We define the "type" of a vertex $v \in [n]$ to be the pair $(\theta_v^{(n)}, D^{(n)}(v))$. Next, we encode vertex types, i.e. the sequence $((\theta_v^{(n)}, D^{(n)}(v)) : v \in [n])$.

E. Encoding Partition Graphs

Our next step is to encode those edges which are not star edges. In order to do so, we partition such edges based on their types. This will result in a number of unmarked graphs which will be encoded separately. More precisely, let $\mathcal{E}^{(n)}$ denote the set of all edge types $(t,t') \in \mathcal{F}^{(\delta,h)} \times \mathcal{F}^{(\delta,h)}$ such that $\psi_{h,\delta}^{(n)}(v,w) = (t,t')$ for some edge (v,w). For each $(t,t') \in \mathcal{E}^{(n)}$, we form the *partition graph* $G_{t,t'}^{(n)}$ which is an unmarked graph defined as follows. First, assume that $t \neq t'$. In this case, $G_{t,t'}^{(n)}$ is a bipartite graph, where those vertices v in $G^{(n)}$ with at least one edge with type (t,t') appear as left nodes in $G_{t,t'}^{(n)}$. Likewise, each vertex in $G^{(n)}$ which has at least one edge with type (t',t) appears as a right node in $G_{t,t'}^{(n)}$. A node can appear both as a left node and as a right node. Then, every edge (v,w) in $G^{(n)}$ with $\psi_{h,\delta}^{(n)}(v,w) = (t,t')$ appears as an edge in $G_{t,t'}^{(n)}$ connecting the left node corresponding to v to the right node corresponding to w. Now, consider the case $t = t'$. In this case, $G_{t,t}^{(n)}$ is a simple graph, where each node in $G^{(n)}$ with at least one edge with type (t,t) appears as a node in $G_{t,t}^{(n)}$ and each edge (v,w) in $G^{(n)}$ such that $\psi_{h,\delta}^{(n)}(v,w) = (t,t)$ appears as an edge in $G_{t,t}^{(n)}$ connecting the node corresponding to v to the node corresponding to w.

At this point, we encode each partition graph $G_{t,t'}^{(n)}$ for $(t,t' \in \mathcal{E}^{(n)})$. Note that for $(t,t') \in \mathcal{E}^{(n)}$, $t \neq t'$, $G_{t',t}^{(n)}$ can be obtained from $G_{t',t}^{(n)}$ by switching left and right nodes. Hence, in order to avoid redundancy, we define $\mathcal{E}_{\le}^{(n)}$ in a certain way so that only one of the pairs (t,t') and (t',t) appears in it. Note that each edge in $G^{(n)}$ which is not a star edge appears in exactly one of the partition graphs $(G_{t,t'}^{(n)} : (t,t') \in \mathcal{E}_{\le}^{(n)})$. Hence, the decoder can reconstruct the original graph $G^{(n)}$ by decoding partition graphs and putting them together.

Recall from Section VI-D that the decoder has access to vertex types. Therefore, it can determine the number of edges of each type connected to each vertex. Hence, the decoder knows the degrees of the nodes in each partition graph. To encode each partition graph, roughly speaking, we find its index among all graphs with the same degree sequence, when such graphs are sorted with respect to the lexicographic order of their adjacency matrix. It can be shown that this index has a closed form, and there is an efficient algorithm for computing it, such that the time and memory complexities of encoding and decoding each partition graph are $O(n\mathrm{polylog}(n))$.

VII. Experimental Results

In this section we discuss the performance of our algorithm.

A. Synthetic data

We generate a random graph $G^{(n)}$ on n vertices as follows. At each vertex $v \in [n]$ we generate a Poisson random variable d_v with mean 3 and connect v to d_v many vertices chosen u.a.r. from $[n] \setminus \{v\}$. If v connects to w and w also connects to v, we treat this as a single edge between v and w. We also add independent vertex marks for each vertex with $\Theta = \{1,2\}$, and two independent edge marks in each direction for each edge with $\Xi = \{1,2\}$. It can be seen that the local weak limit of this model is a Poisson Galton–Watson tree with mean degree 6 and independent vertex and edge marks. Since the limit distribution is completely characterized by the depth 1 neighborhood distribution at the root, we choose $h = 1$, and run the algorithm with different values of δ. See Figure 1 for the behavior of $l_n := (\mathsf{nats}(f_{h,\delta}^{(n)}(G^{(n)})) - m^{(n)} \log n)/n$, where $m^{(n)}$ is the number of edges in $G^{(n)}$. As we see, for large values of δ, l_n converges to the marked BC entropy of the limit as n gets large, which is consistent with Theorem 1.

B. Locally tree–like data

Recall from Theorem 1 that our theoretical guarantee holds when the limit μ is supported on $\bar{\mathcal{T}}_*$. Motivated by this, we test our algorithm on the following two real–world locally tree–like datasets collected from [15]. We also compare the compression

Authorized licensed use limited to: Univ of Calif Berkeley. Downloaded on August 21,2021 at 17:07:24 UTC from IEEE Xplore. Restrictions apply.

Dataset	[3] (BPL)	best BPL [relative%] (h, δ)	encode/decode time (sec)	best 20% BPL [relative%] (h, δ)	encode/decode time (sec)
roadnet-CA	10.58	5.93 [+44%] (4,2)	15.75/58.2	10.25 [+3%] (4,20)	20.24/94.4
roadnet-PA	10.07	5.94 [+41%] (3,2)	7.4/28.9	9.80 [+2.7%] (2,10)	9.8/46.5

TABLE IV: Comparing the compression ratios of our algorithm with those in [3] for road networks. In the third column, our best ratio together with the relative improvement over [3] are given. In the fourth column, the corresponding encoding/decoding times in seconds are given. As we can see, for both datasets, δ in the best cases is small. Motivated by Figure 1, one explanation can be that we are not yet in the asymptotic regime. In order to address this, in the fifth and the sixth columns, we report the best ratio of our algorithm as well as encoding/decoding times assuming that δ is chosen so that at most 20% of the edges are allowed to be star edges. As we can see, even in this case, our compression ratios are better compared to those in [3].

Dataset	[2] (BPL)	best BPL [relative%] (h, δ)	encode/decode time (sec)	best 40% BPL [relative%] (h, δ)	encode/decode time (sec)
dblp-2010	6.78	5.23 [23%] (4,2)	2.25/7.62	7.13 [-5.16%] (1,10)	2.53/12.1
amazon-2008	9.12	8.31 [9%] (4,2)	6.42/20.57	11.1 [-21.7%] (1,15)	13.1/76.2
hollywood-2009	5.14	4.67 [9%] (4,2)	33.67/41.88	5.22 [-1.6%] (2,200)	71.1/157.57
ljournal-2008	10.90	8.73 [20%] (4,2)	74.76/210.54	10.77 [1.2%] (1,100)	611.7/515

TABLE V: Comparison for social networks. The structure of this table is similar to that of Table IV.

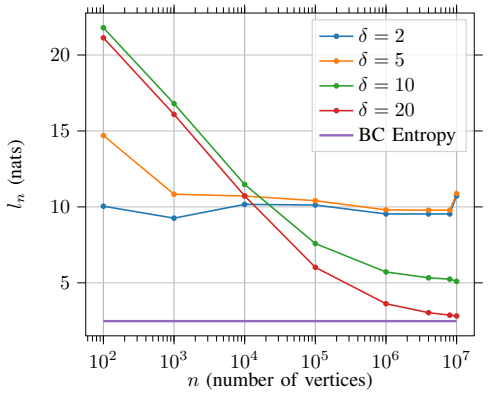

Fig. 1: Synthetic data results. Note that for large δ the asymptotic performance converges to the actual BC entropy.

results with the ones reported in [3], which, to the best of our knowledge, are the best results for these datasets.

- *roadnet-CA*: the graph of the road network of California, consisting of 1,965,206 vertices and 5,533,214 edges.
- *roadnet-PA*: the graph of the road network of Pennsylvania, consisting of 1,088,092 vertices and 3,083,796 edges.

Following the convention in the literature, we report the compression ratios in *bits per link* (BPL). Table IV compares the best compression ratios of our algorithm, which, as we can see, are more than 40% better than the ones in [3].

C. Social networks

We consider the following social network datasets available on the Laboratory of Web Algorithms (http://law.di.unimi.it).

- *dblp-2010*: an undirected simple graph consisting of 326,186 vertices and 1,615,400 edges, where each vertex represents a scientist, and two vertices are connected if they have a joint article.
- *hollywood-2009*: a simple undirected graph, consisting of 1,139,905 vertices and 113,891,327 edges, where each vertex represents an actor, and two vertices are connected if the corresponding actors have appeared in a movie together.
- *amazon-2008*: a simple undirected graph, consisting of 735,323 vertices and 5,158,388 edges, describing similarity among books as reported by the Amazon store.
- *ljournal-2008*: collected by [16], based on the social website LiveJournal started in 1999, this dataset consists of 5,363,260 vertices and 79,023,142 edges.

We compare our compression ratios to those reported in [2], which are the best in the literature to the best of our knowledge. Note that (a) the above datasets are not locally tree–like and our theoretical optimality guarantee of Theorem 1 does not hold for them, and (b) the compression method in [2] is tailored for social networks, whereas our method is universal. Nonetheless, as suggested by Table V, our compression ratio is comparable in most cases, and is even better in some cases.

VIII. Conclusion

We introduced a computationally efficient lossless compression algorithm for sparse marked graphs which is universally optimal. We investigated the performance of our algorithm through synthetic and real–world data.

Acknowledgments

The authors acknowledge support from the NSF grants ECCS–1343398, CNS–1527846, CCF–1618145, CCF-1901004, the NSF Science & Technology Center grant CCF–0939370 (Science of Information), and the William and Flora Hewlett Foundation supported Center for Long Term Cybersecurity at Berkeley.

Authorized licensed use limited to: Univ of Calif Berkeley. Downloaded on August 21,2021 at 17:07:24 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] P. Boldi and S. Vigna, "The webgraph framework i: compression techniques," in *Proceedings of the 13th international conference on World Wide Web*. ACM, 2004, pp. 595–602.

[2] P. Boldi, M. Rosa, M. Santini, and S. Vigna, "Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 587–596.

[3] P. Liakos, K. Papakonstantinopoulou, and M. Sioutis, "Pushing the envelope in graph compression," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 1549–1558.

[4] Y. Choi and W. Szpankowski, "Compression of graphical structures: Fundamental limits, algorithms, and experiments," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 620–638, 2012.

[5] D. J. Aldous and N. Ross, "Entropy of some models of sparse random graphs with vertex-names," *Probability in the Engineering and Informational Sciences*, vol. 28, no. 02, pp. 145–168, 2014.

[6] E. Abbe, "Graph compression: The effect of clusters," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 1–8.

[7] T. Luczak, A. Magner, and W. Szpankowski, "Structural information and compression of scale-free graphs," *Urbana*, vol. 51, p. 618015, 2017.

[8] I. Benjamini and O. Schramm, "Recurrence of distributional limits of finite planar graphs," *Electron. J. Probab.*, vol. 6, pp. no. 23, 13 pp. (electronic), 2001. [Online]. Available: http://dx.doi.org/10.1214/EJP.v6-96

[9] D. Aldous and J. M. Steele, "The objective method: probabilistic combinatorial optimization and local weak convergence," in *Probability on discrete structures*. Springer, 2004, pp. 1–72.

[10] D. Aldous and R. Lyons, "Processes on unimodular random networks," *Electron. J. Probab*, vol. 12, no. 54, pp. 1454–1508, 2007.

[11] C. Bordenave and P. Caputo, "Large deviations of empirical neighborhood distribution in sparse random graphs," *Probability Theory and Related Fields*, vol. 163, no. 1-2, pp. 149–222, 2015.

[12] P. Delgosha and V. Anantharam, "A notion of entropy for stochastic processes on marked rooted graphs," *arXiv preprint arXiv:1908.00964*, 2019.

[13] ——, "Universal lossless compression of graphical data," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 1578–1582.

[14] P. Billingsley, *Convergence of probability measures*. John Wiley & Sons, 2013.

[15] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[16] F. Chierichetti, R. Kumar, S. Lattanzi, M. Mitzenmacher, A. Panconesi, and P. Raghavan, "On compressing social networks in: Proc. of 15th conference on knowledge discovery and data mining (kdd09)," 2009.

Authorized licensed use limited to: Univ of Calif Berkeley. Downloaded on August 21,2021 at 17:07:24 UTC from IEEE Xplore. Restrictions apply.