Scaling Graph-based Semi Supervised Learning to Large Number of Labels Using Count-Min Sketch

Partha Pratim Talukdar
Machine Learning Department
Carnegie Mellon University, USA
ppt@cs.cmu.edu

William Cohen
Machine Learning Department
Carnegie Mellon University, USA
wcohen@cs.cmu.edu

Abstract

Graph-based Semi-supervised learning (SSL) algorithms have been successfully used in a large number of applications. These methods classify initially unlabeled nodes by propagating label information over the structure of graph starting from seed nodes. Graph-based SSL algorithms usually scale linearly with the number of distinct labels (m), and require $O(m)$ space on each node. Unfortunately, there exist many applications of practical significance with very large m over large graphs, demanding better space and time complexity. In this paper, we propose MAD-SKETCH, a novel graph-based SSL algorithm which compactly stores label distribution on each node using Count-min Sketch, a randomized data structure. We present theoretical analysis showing that under mild conditions, MAD-SKETCH can reduce space complexity at each node from $O(m)$ to $O(\log m)$, and achieve similar savings in time complexity as well. We support our analysis through experiments on multiple real world datasets. We observe that MAD-SKETCH achieves similar performance as existing state-of-the-art graph-based SSL algorithms, while requiring smaller memory footprint and at the same time achieving up to 10x speedup. We find that MAD-SKETCH is able to scale to datasets with one million labels, which is beyond the scope of existing graph-based SSL algorithms.

Appearing in Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33. Copyright 2014 by the authors.

1 Introduction

Graph-based semi-supervised learning (SSL) methods based on label propagation work by associating classes with each labeled "seed" node, and then propagating these labels over the graph in some principled, iterative manner [23, 17]. The end result of SSL is thus a matrix Y, where $Y_{u\ell}$ indicates whether node u is positive for class ℓ. SSL techniques based on label propagation through graphs are widely used, and they are often quite effective for very large datasets. Since in many cases these method converge quite quickly, the time and space requirements scale linearly with both the number of edges $|E|$ in the graph, and the number of classes m.

Unfortunately, there exist a number of applications where both m and $|E|$ are very large: for instance, Carlson et al. [4] describe an SSL system with hundreds of overlapping classes, and Shi et al [14] describe a text classification task with over 7000 classes. Similarly, the ImageNet dataset [10] poses a classification task involving 100,000 classes. In this paper, we seek to extend graph-based SSL methods to cases where there are a large number of potentially overlapping labels. To do this, we represent the class scores for u with a *count-min sketch* [6], a randomized data structure.

Graph-based SSL using a count-min sketch has a number of properties that are desirable, and somewhat surprising. First, the sketches can be asymptotically smaller than conventional data structures. Normally, the vectors of label scores that are propagated through a graph are dense, i.e. of size $O(m)$. However, we show that if the seed labels are sparse, or if the label scores at each node have a "power-law" structure, then the storage at each node can be reduced from $O(m)$ to $O(\log m)$. Experimentally, we show that power-law label scores occur in natural datasets. Analytically, we also show that a similar space reduction can be obtained for graphs that have a certain type of community structure. A second useful property of the count-min sketch is that for label propagation algorithms using a certain family of updates—including Modified Adsorption (MAD) [17], the harmonic method [23] and several other methods [21]—the linearity of the count-min sketch implies that a similarly large reduction in *time complexity* can be obtained. Experimentally, little or no loss in accuracy is observed on moderate-sized datasets, and MAD-Sketch, the new method scales to a SSL problem with millions of edges and nodes, and one million labels.

2 Related Work

Although tasks with many labels are common, there has been surprisingly little work on scaling learning algorithms to many labels. Weinberger et al. [20] describe a feature-hashing method called a "hash kernel", which is closely related to a count-min sketch, and show that it can help supervised learning methods scale to many classes. Other approaches to scaling classifiers to many classes include reducing the dimension of label space, for instance by error-correcting output codes [9] or label embedding trees [3]. Certain supervised learning methods, notably nearest-neighbor methods, also naturally scale well to many classes [22]. The main contribution of this work is provide similarly scalable methods for SSL methods based on label propagation. A graph-based SSL method aimed at retaining only top ranking labels per-node out of a large label set is presented in [1]. Similar ideas to induce sparsity on the label scores were also explored in [18, 8]. In contrast to these methods, the method presented in paper doesn't attempt to enforce sparsity, and instead focuses on compactly storing the *entire* label distribution on each node using Count-min Sketch.

The count-min sketch structure has been used for a number of tasks, such as for the storage distributional similarity data for words in NLP [11]. Perhaps the most related of these tasks is the task of computing and storing personalized PageRank vectors for each node of a graph [13, 15]. Prior analyses, however, do not apply immediately to SSL, although they can be viewed as bounds for a special case of SSL in which each node is associated with a 1-sparse label vector and there is one label for every node. We also include new analyses that provide tighter bounds for skewed labels, and graphs with community structure.

3 Preliminaries

3.1 Count-Min Sketches (CMS)

The count-min sketch is a widely-used probabilistic scheme [6]. At a high level, it stores an approximation of a mapping between integers i and associated real values y_i. Specifically, the count-min sketch consists of a $w \times d$ matrix $\mathbb{S}$, together with d hash functions $h_1, \ldots, h_d$, which are chosen randomly from a pairwise-independent family. A sketch is always initialized to an all-zero matrix.

Let $\mathbf{y}$ be a sparse vector, in which the i-th component has value y_i. To store a new value y_i to be associated with component i, one simply updates as follows:

$$\mathbb{S}_{j,h_j(i)} \leftarrow \mathbb{S}_{j,h_j(i)} + y_i, \ \forall 1 \leq j \leq d$$

To retrieve an approximate value $\hat{y_i}$ for i, one computes the minimum value over all $j : 1 \leq j \leq d$ of $\mathbb{S}_{j,h_j(i)}$ as follows:

$$\hat{y_i} = \min_{1 \leq j \leq d} \mathbb{S}_{j,h_j(i)} \tag{1}$$

We note that $\hat{y_i}$ will never underestimate y_i, but may overestimate it if there are hash collisions. As it turns out, the parameters w and d can be chosen so that with high probability, the overestimation error is small (as we discuss below, in Section 5).

Importantly, count-min sketches are linear, in the following sense: if $\mathbb{S}_1$ is the matrix for the count-min sketch of $\mathbf{y}_1$ and $\mathbb{S}_2$ is the matrix for the count-min sketch of $\mathbf{y}_2$, then $a\mathbb{S}_1 + b\mathbb{S}_2$ is the matrix for the count-min sketch of the vector $a\mathbf{y}_1 + b\mathbf{y}_2$.

3.2 Graph-based Semi-Supervised Learning

We now briefly review graph-based SSL algorithms.

Notation

All the algorithms compute a soft assignment of labels to the nodes of a graph $G = (V, E, W)$, where V is the set of nodes with $|V| = n$, E is the set of edges, and W is an edge weight matrix. Out of the $n = n_l + n_u$ nodes in G, n_l nodes are labeled, while the remaining n_u nodes are unlabeled. If edge $(u, v) \notin E$, $W_{uv} = 0$. The (unnormalized) Laplacian, L, of G is given by $L = D - W$, where D is an $n \times n$ diagonal degree matrix with $D_{uu} = \sum_v W_{uv}$. Let S be an $n \times n$ diagonal matrix with $S_{uu} = 1$ iff node $u \in V$ is labeled. That is, S identifies the labeled nodes in the graph. C is the set of labels, with $|C| = m$ representing the total number of labels. Q is the $n \times m$ matrix storing training label information, if any. Y is an $n \times m$ matrix of soft label assignments, with Y_{vl} representing the score of label l on node v. A graph-based SSL computes Y from $\{G, S, Q\}$.

Modified Adsorption (MAD)

Modified Adsorption (MAD) [17] is a modification of an SSL method called Adsorption [2]. MAD can be expressed as an unconstrained optimization problem, whose objective is shown below.

$$\min_{Y} \sum_{l \in C} \Big[\mu_1 \left(Q_l - Y_l\right)^\top S \left(Q_l - Y_l\right) + \mu_2 \, Y_l^\top \, L^{'} \, Y_l \; + \; \mu_3 \left\| Y_l - R_l \right\|_2^2 \Big] \quad (2)$$

where μ_1, μ_2, and μ_3 are hyperparameters; $L^{'} = D^{'} - W^{'}$ is the Laplacian of an undirected graph derived from G, but with revised edge weights; and R is an $n \times m$ matrix of per-node label prior, if any, with R_l representing the l^{th} column of R. As in Adsorption, MAD allows labels on seed nodes to change. In case of MAD, the three random-walk probabilities, p_v^{inj}, p_v^{cont}, and p_v^{abnd}, defined by Adsorption on each node are folded inside the matrices $S, L^{'}$, and R, respectively. The optimization problem in (2) can be solved with an efficient iterative update which is shown below and repeated until convergence, described in detail by [17].

$$\begin{aligned} Y_v^{(t+1)} &\leftarrow \tfrac{1}{M_{vv}}(\mu_1 \times S_{vv} \times Q_v + \mu_2 \times D_v^{(t)} + \mu_3 \times R_v), \ \forall v \in V \\ \text{where } D_v^{(t)} &= \textstyle\sum_{u \in V}(W^{'}_{uv} + W^{'}_{vu}) \times Y_u^{(t)}, \\ \text{and } M_{vv} &= \mu_1 S_{vv} + \mu_2 \textstyle\sum_{u \neq v}(W^{'}_{uv} + W^{'}_{vu}) + \mu_3 \end{aligned}$$

Typically $Y^{(t)}$ is sparse for very small t, and becomes dense in later iterations. MAD is easily parallelizable in a MapReduce framework [19, 12], which makes it suitable for SSL on large datasets.

4 MAD-Sketch: Using Count-Min Sketch for Graph SSL

In this section, we propose MAD-Sketch, a count-min sketch-based extension of MAD[1] [17]. Instead of storing labels and their scores exactly as in MAD, MAD-Sketch uses count-min sketches to compactly store labels and their scores. MAD-Sketch uses the following equation to update label sketches at each node, and this process is repeated until convergence.

$$\begin{aligned} \mathbb{S}_{Y,v}^{(t+1)} \leftarrow \; & \frac{1}{M_{vv}} \Big(\mu_1 \times S_{vv} \times \mathbb{S}_{Q,v} + \\ & \mu_2 \times \sum_{u \in V} (W^{'}_{uv} + W^{'}_{vu}) \times \mathbb{S}_{Y,u}^{(t)} + \\ & \mu_3 \times \mathbb{S}_{R,v}\Big), \ \forall v \in V \end{aligned} \quad (3)$$

where $\mathbb{S}_{Y,v}^{(t+1)}$ is the count-min sketch corresponding to label score estimates on node v at time $t+1$, $\mathbb{S}_{Q,v}$ is the sketch corresponding to any seed label information, and finally $\mathbb{S}_{R,v}$ is the sketch corresponding to label regularization targets in node v. Please note that due to linearity of CMS, we don't need to unpack the labels for each update operation. This results in significant runtime improvements compared to non-sketch-based MAD as we will see in Section 6.3.

[1]We use MAD as a representative graph-based SSL algorithm, which also generalizes prior work of [23].

After convergence, MAD-Sketch returns $\mathbb{S}_{Y,v}$ as the count-min sketch containing estimated label scores on node v. The final label score of a label can be obtained by querying this sketch using Equation 1. We denote the result of this query by $\hat{Y}$.

5 Analysis

5.1 Sparse initial labelings

We now turn to the question of how well MAD-Sketch approximates the exact version—i.e., how well $\hat{Y}$ approximates Y. We begin with a basic result on the accuracy of count-min sketches [6].

Theorem 1 (Cormode and Muthukrishnan [6]) *Let $\boldsymbol{y}$ be an vector, and let $\tilde{y}_i$ be the estimate given by a count-min sketch of width w and depth d for y_i. If $w \geq \frac{e}{\eta}$ and $d \geq \ln(\frac{1}{\delta})$, then $\tilde{y}_i \leq y_i + \eta \|\boldsymbol{y}\|_1$ with probability at least 1-δ.*

To apply this result, we consider several assumptions that might be placed on the learning task. One natural assumption is that the initial seed labeling is *k-sparse*, which we define to be a labeling so that for all v and ℓ, $\|Y_{v\cdot}\|_1 \leq k$ for some small k (where $Y_{v\cdot}$ is the v-th row of Y). If $\hat{Y}$ is a count-min approximation of Y, then we define the *approximation error of $\hat{Y}$ relative to Y* as $\max_{v,\ell}(\hat{Y}_{v,\ell} - Y_{v,\ell})$. Note that approximation error cannot be negative, since a count-min sketch can only overestimate a quantity.

Theorem 2 *If the parameters of* MAD-Sketch *μ_1, μ_2, μ_3 are such that $\mu_1 + \mu_2 + \mu_3 \leq 1$, Y is k-sparse and binary, and sketches are of size $w \geq \frac{ek}{\epsilon}$ and $d \geq \ln \frac{m}{\delta}$ then the approximation error* MAD-Sketch *is less than ϵ with probability 1-δ.*

Proof. The result holds immediately if $\forall v$, $\|Y_{v\cdot}\|_1 \leq k$, by application of the union bound and Theorem 1 with $\eta = \epsilon/k$. However, if $\mu_1 + \mu_2 + \mu_3 \leq 1$, then at each iteration of MAD's update, Equation 3, the new count-min sketch at a node is bounded by weighted average of previously-computed count-min sketches, so by induction the bound on $\|Y_{v\cdot}\|_1$ will hold. □

Although we state this result for MAD-Sketch, it also holds for other label propagation algorithms that update scores by a weighted average of their neighbor's scores, such as the harmonic method [23].

5.2 Skewed labels

Perhaps more interestingly, the label weights $Y_{v\cdot}$ associated with real datasets exhibit even more structure than sparsity alone would suggest. Define a Zipf-like distribution to be one where the frequency of the i-th most frequent value $f_i \propto i^{-z}$. For graphs from two real-world datasets, one with 192 distinct labels and another with 10,000 labels, Figure 1 shows the label scores at each rank, aggregated across all nodes in these two graphs. Note that the score distributions are Zipf-like[2] with $z \approx 1$, indicating a strong skew toward a few large values.

This structure is relevant because it is known that count-min sketches can store the largest values in a skewed data stream even more compactly. For instance, the following lemma can be extracted from prior analysis [7].

Lemma 1 [Cormode and Muthukrishnan [7], Eqn 5.1] *Let $\boldsymbol{y}$ be an vector, and let $\tilde{y}_i$ be the estimate given by a count-min sketch of width w and depth d for y_i. Let the k largest components of $\boldsymbol{y}$ be $y_{\sigma_1}, \ldots, y_{\sigma_k}$, and let $t_k = \sum_{k'>k} y_{\sigma_{k'}}$ be the weight of the "tail" of $\boldsymbol{y}$. If $w \geq \frac{1}{3k}$, $w > \frac{e}{\eta}$ and $d \geq \ln \frac{3}{2} \ln \frac{1}{\delta}$, then $\tilde{y}_i \leq y_i + \eta t_k$ with probability at least 1-δ.*

The proof for this statement comes from breaking down the count-min error analysis into two independent parts: (1) errors due to collisions among the k largest values, which are unlikely since there are only k values to be hashed into the $w > 3k$ bins; and (2) errors due to collisions with the tail items, which have small impact since t_k is small. As a application of this analysis, Cormode and Muthukrishnan also showed the following.

Theorem 3 [Cormode and Muthukrishnan [7], Theorem 5.1] *Let $\boldsymbol{y}$ represent a Zipf-like distribution with parameter z. Then with probability at least 1-δ, $\boldsymbol{y}$ can be approximated to within error η by a count-min sketch of width $O(\eta^{-\min(1,1/z)})$ and depth $O(\ln \frac{1}{\delta})$.*

Note that this result holds even the absence of label sparsity; also, it gives a strictly more space-efficient sketch when $z > 1$ (for instance, when $z = 1.5$, then we can reduce the width of each sketch to $\lceil \frac{e}{\sqrt{\epsilon}} \rceil$), as observed below.

Corollary 1 *If the vector label scores $Y_{v\cdot}$ for every node v is bounded by a Zipf-like distribution parameter $z > 1$, and sketches are of size $w \geq \frac{e}{\epsilon^{z-1}}$ and $d \geq \ln \frac{m}{\delta}$ then the approximation error of* MAD-Sketch *is less than ϵ with probability 1-δ.*

[2]Although in the case of the 192-label dataset, the Zipf-like distribution is clearly truncated somewhere below $i =$ 192.

Again, although we state this result for MAD-Sketch, it would also hold for sketch versions of other label propagation algorithms: the only requirement is that label scores are (empirically) always skewed during propagation.

5.3 Graphs with community structure

Lemma 1 can also be used to derive a tighter bound for a certain type of nearly block-diagonal graph. Let $G = (V, E, W)$ be a graph, let $S \subset V$ be a set of vertices. For a vertex $u \in S$ we define $\partial(u, S)$ to be the total weight of all edges from u to a vertex outside of S, i.e. $\partial(u, S) \equiv \sum_{v \notin S} W_{uv}$. For convenience, we define $vol(u) \equiv D_{uu}$, and $vol(S) \equiv \sum_{u \in S} vol(u)$. The *max-conductance of* S, denoted $\psi(S)$, is defined to be

$$\psi(S) \equiv \max_{u \in S} \frac{\partial(u, S)}{vol(u)}$$

Notice that this is different from the conductance of S, which is often defined as

$$\phi(S) \equiv \frac{\sum_{u \in S} \partial(u, S)}{\min(vol(S), vol(V) - vol(S))}$$

When $vol(S)$ is small and edge weights are uniform, conductance $\phi(S)$ can be thought of as the probability that an edge will leave S when it is chosen uniformly from the set of all edges exiting any node $u \in S$. In contrast, the quantity $\frac{\partial(u,S)}{vol(u)}$ is the probability that an edge will leave S when it is chosen uniformly from the set of all edges exiting a *particular* node u, and max-conductance is the maximum of this quantity over $u \in S$. It is not hard to see that $\psi(S) \geq \phi(S)$ for any S, so having small max-conductance is a stronger condition than having small conductance. We have the following result.

Theorem 4 *Let u be a vertex and S be a vertex set such that $u \in S$, $|S| = k$, and the min-conductance of S is ψ, as measured using the modified weight matrix W' used by MAD. Assume also that initial labels are binary, and that $w \geq \frac{1}{3k}$, $w > \frac{e}{\eta}$ and $d \geq \ln \frac{3}{2} \ln \frac{1}{\delta}$. Then for all ℓ, the approximation error of* MAD-Sketch *is bounded by $\eta\psi$.*

Proof. Let $\Pr(u \overset{W'}{\to} v)$ be the probability of a random walk from u to v in the transition matrix defined by W', and let $\Pr(u \overset{S,W'}{\to} v)$ be the (improper) probability[3] of a random walk restricted to vertices in S.

[3]An improper distributions may sum to less than one.

We observe that min-conductance of S can be used to bound $\sum_{v \notin S} \Pr(u \overset{W'}{\rightarrow} v)$. Let B_S be the set of vertexes not in S, but connected to some node $z \in S$. Since any path to $v \notin S$ must pass through some $z \in B_S$, we have that

$$\begin{aligned}
\sum_{v \notin S} \Pr(u \overset{W'}{\rightarrow} v) &\leq \sum_{b \in B_S} \Pr(u \overset{W'}{\rightarrow} b) \\
&= \sum_{a \in S, b \in B_S} \Pr(u \overset{S,W'}{\rightarrow} a) W'_{ab} \\
&= \sum_{a \in S} \Pr(u \overset{S,W'}{\rightarrow} a) \sum_{b \in B_S} \frac{W_{ab}}{vol(a)} \\
&\leq \sum_{a \in S} \Pr(u \overset{S,W'}{\rightarrow} a) \psi \\
&\leq \psi
\end{aligned}$$

It can be shown that $Y_{u\ell} \leq \sum_{v: Y_{v\ell}=1} \Pr(u \overset{W'}{\rightarrow} v)$—i.e., that MAD is based on a partially absorbing random walk [21]; hence the theorem holds by application of Lemma 1 with the tail t_k defined as the weight of $\sum_{v \notin S} \Pr(u \overset{W'}{\rightarrow} v)$. $\square$

An immediate consequence of this is that if W' is composed of size-k subcommunities with min-conductance ψ, then the sketch size can again be reduced by a factor of ψ:

Corollary 2 *If Y is binary and W' has the property that all nodes u are inside some subset S such that $|S| < k$ and $\psi(S) \leq \psi$, and if sketches are of size $w \geq \frac{e\psi}{\epsilon}$ and $d \geq \ln \frac{m}{\delta}$ then the approximation error of* MAD-SKETCH *is less than ϵ with probability 1-δ.*

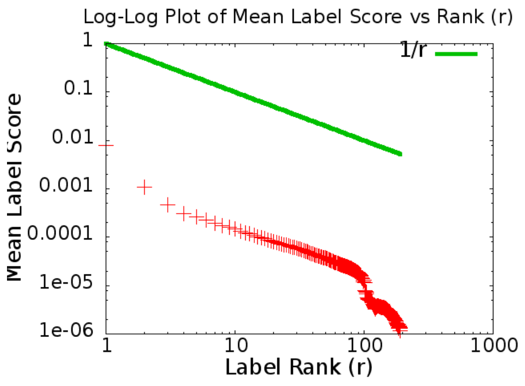
Again, although we state this result for MAD, it would also hold for sketch versions of other label propagation algorithms that can be modeled as partially absorbing random walks, which include the harmonic functions method and several other well-known approaches to SSL [21].

6 Experiments

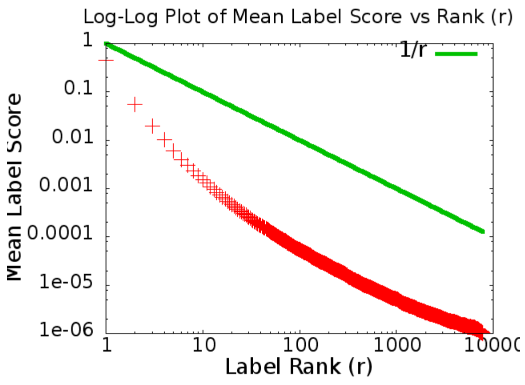
6.1 Experimental Setup

The three real-world datasets used for the experiments in this section are described in Table 1. The Freebase dataset was constructed out of 18 domains (e.g., *music, people* etc.) from the full dataset, and was previously used in [18] for determining semantic types of entities. This dataset consists of 192 labels, with 10 seeds per labels. This results in the seeding of 1917 nodes, with three nodes acquiring 2 labels each. This results in a $k-$sparsity with $k = 2$. Both Flickr-10k and Flickr-1m datasets consist of edges connecting images to various metadata, e.g., tags, geo-location, etc. For these two datasets, each seed node was injected with its own unique label, and hence the number of labels are equal to the number of seed nodes. Given seed node, the goal here is to identify other similar nodes (e.g., other images, tags, etc.). This self-injection-based seeding scheme results in a $k-$sparsity with $k = 1$. In the last two columns of Table 1, we calculate the width and depth of sketches as prescribed by Theorem 2 for $\epsilon = 0.05$ and $\delta = 0.1$.

We shall compare two graph-based SSL algorithms: **MAD-Exact**, the default MAD algorithm [17], where labels and their scores are stored exactly on each node without any approximation, and **MAD-Sketch**, the sketch-based version of MAD, where the labels and their scores on each node are stored inside a count-min



(a) Freebase



(b) Flickr-10k

Figure 1: Plots in log-log space demonstrating skewness of label scores estimated by MAD-EXACT over two datasets: (a) Freebase and (b) Flickr-10k. For reference, the plot for $\frac{1}{r}$ is also shown, which is Zipfian law with $z = 1$. Please note that the label scores used here are exact, and not approximated using any sketch. See Section 6.2 for details.

Name	Nodes (n)	Edges	Labels (m)	Seed Nodes	$k-$Sparsity	$\lceil \frac{ek}{\epsilon} \rceil$	$\lceil \ln \frac{m}{\delta} \rceil$
Freebase	301,638	1,155,001	192	1917	2	109	8
Flickr-10k	41,036	73,191	10,000	10,000	1	55	12
Flickr-1m	1,281,887	7,545,451	1,000,000	1,000,000	1	55	17

Table 1: Description of various graph-based datasets used in Section 6 using $\epsilon = 0.05, \delta = 0.1$. Please note that Flickr-10k is a strict subset of the Flickr-1m dataset. The last two columns compute the width and depth of a Count-min Sketch as prescribed by Theorem 2. See Section 6.1 for further details.

	Average Memory Usage (GB)	Total Runtime (s) [Speedup w.r.t. MAD-Exact]	MRR
MAD-Exact	3.54	516.63 [1.0]	0.28
MAD-Sketch ($w = 109, d = 8$)	2.68	110.42 [4.7]	0.28
MAD-Sketch ($w = 109, d = 3$)	1.37	54.45 [9.5]	0.29
MAD-Sketch ($w = 20, d = 8$)	1.06	47.72 [10.8]	0.28
MAD-Sketch ($w = 20, d = 3$)	1.12	48.03 [10.8]	0.23

Table 2: Table comparing average per-iteration memory usage (in GB), runtime (in seconds), and MRR of MAD-Exact and MAD-Sketch (for various sketch sizes) when applied over the Freebase dataset. Using sketch size of $w = 109$ and $d = 8$ as prescribed by Theorem 2 for this dataset, we find that MAD-Sketch is able to obtain the same MRR (performance) as MAD-Exact, while using a reduced memory footprint and achieving about 4.7 speedup. This is our main result in the paper. Additionally, we find that even more aggressive sketches (e.g., $w = 20, d = 8$ in this case) may be used in practice, with the possibility of achieving further gains in memory usage and runtime. See Section 6.3 for details.

sketch.[4]. For the experiments in this paper, we found about 10 iterations to be sufficient for convergence of all algorithms.

Once we have label assignments Y on the evaluation nodes on the graph, we shall use Mean Reciprocal Rank (MRR) as the evaluation metric, where higher is better.[5]

6.2 Checking for Label Skew

Our benchmark tasks are k-sparse for small k, so a compact sketch is possible. However, analysis suggests other scenarios will also be amenable to small sketches. While community structure is expensive to find in large graphs, it is relatively straightforward to check for skewed label scores. To this end, we applied MAD-Exact on two datasets, Freebase and Flickr-10k. Plots showing mean label scores vs. label rank in the log-log space are shown in Figure 1, along with a plot of $\frac{1}{rank}$ for each dataset. In Zipfian distribution with $z = 1$, these plots should be asymptotically parallel. We indeed see highly skewed distributions, with most of the weight concentrated on the largest scores. These observations demonstrate existence of skewed label scores (during exact label propagation) in real-world datasets, thereby providing empirical justification for the analysis in Section 5.

6.3 Is Sketch-based Graph SSL Effective?

In this section, we evaluate how MAD-Sketch compares with MAD-Exact in terms of memory usage, runtime, and performance measured using MRR. We set $\mu_1 = 0.98, \mu_2 = 0.01, \mu_3 = 0.01$, and run both algorithms for 10 iterations, which we found to be sufficient for convergence. Experimental results comparing these two algorithms when applied over the Freebase dataset are shown in Table 2.

First, we compare MAD-Exact and MAD-Sketch ($w = 109, d = 8$), where the width and depth of the sketch is as suggested by Theorem 2 (see Table 1). From Table 2, we observe that MAD-Sketch ($w = 109, d = 8$) achieves the same MRR as MAD-Exact, while using a lower per-iteration memory footprint. Moreover, MAD-Sketch also achieves a significant speedup of $4.7x$ compared to MAD-Exact. This is our main result, which validates the central thesis of the paper: *Sketch-based data structures when used*

[4]We use the implementation of MAD in the Junto toolkit (https://github.com/parthatalukdar/junto) as MAD-Exact, and extend it to handle sketches resulting in the implementation of MAD-Sketch. We model our count-min sketches after the ones in the springlib library (https://github.com/clearspring/stream-lib), and use linear hash functions [5]. MAD-Sketch source code is available as part of the Junto toolkit or by contacting the authors.

[5] MRR is defined as MRR $= \frac{1}{|F|} \sum_{v \in F} \frac{1}{r_v}$ where $F \subset V$ is the set of evaluation nodes, and r_v is the highest rank of the gold label among all labels assigned to node v.

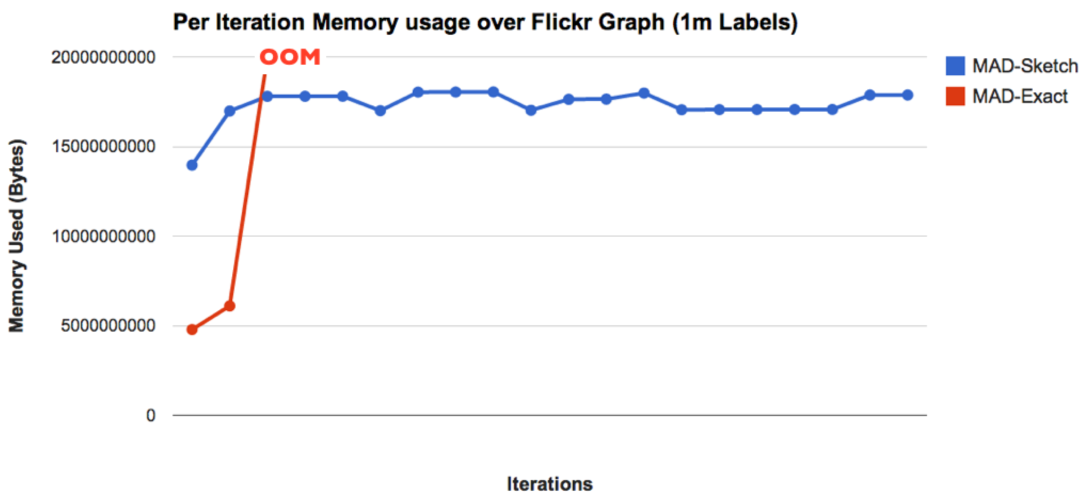


Figure 2: Per-iteration memory usage by MAD when labels and their scores on each node are stored exactly (MAD-Exact) vs using Count-Min Sketch (MAD-Sketch) in the Flickr-1m dataset, with total unique labels $m = 1000000$, and sketch parameters $w = 55$, $d = 17$. We observe that even though MAD-Exact starts out with a lower memory footprint, it runs out of memory (OOM) by third iteration, while MAD-Sketch is able to compactly store all the labels and their scores with almost constant memory usage over the 20 iterations. See Section 6.4 for details.

within graph-based SSL algorithms can result in significant memory and runtime improvements, without degrading performance. This also provides empirical validation of the analysis presented earlier in the paper.

In Table 2, we also explore effect of sketches of various sizes on the performance of MAD-SKETCH. We observe that even though very small sketches do degrade accuracy, sketches smaller than predicted by Theorem 2 do not, suggesting that the existence of additional useful structure in these problems—perhaps a combination of label skew and/or community structure.

6.4 Graph SSL with Large Number of Labels

In previous section, we have observed that MAD-SKETCH, a sketch-based graph SSL algorithm, can result in significant memory savings and runtime speedups compared to MAD-EXACT. In this section, we evaluate sketches in graph SSL when a large number of labels are involved. For this, we run MAD-EXACT and MAD-SKETCH over the Flickr-1m, a dataset with 1 million labels. Both algorithms were run for 20 iterations, and plots comparing their per-iteration memory usage are shown in Figure 2. From this figure, we observe that even though MAD-EXACT starts out with a lower memory footprint (due to its sparse representation of label scores, which are very sparse initially), it runs out of memory by the third iteration, even when 100GB RAM was available. In contrast, MAD-SKETCH when run with a sketch of $w = 55, d = 17$ as prescribed by Theorem 2 (see Table 1) is able to compactly store all labels and their scores, and does not result in an explosion of space requirements in later iterations. This demonstrates scalability of MAD-SKETCH, and sketch-based Graph SSL in general, when applied to datasets with large number labels—the main motivation of this paper.

7 Conclusion

Graph-based Semi-supervised learning (SSL) algorithms have been successfully used in a large number of applications. Such algorithms usually require $O(m)$ space on each node. Unfortunately, for many applications of practical significance with very large m over large graphs, this is not sufficient. In this paper, we propose MAD-SKETCH, a novel graph-based SSL algorithm which compactly stores label distribution on each node using Count-min Sketch, a randomized data structure. We present theoretical analysis showing that under mild conditions, MAD-SKETCH can reduce space complexity at each node from $O(m)$ to $O(\log m)$, and achieve similar savings in time complexity as well. We support our analysis through experiments on multiple real world datasets. We find that MAD-SKETCH is able to achieve same performance as MAD-EXACT, a state-of-the-art graph-based algorithm with exact estimates, while requiring smaller memory footprint and at the same time achieving up to 10x speedup. Also, we find that MAD-SKETCH is able to scale up to datasets containing millions of nodes and edges, and more importantly one million labels, which is beyond the scope of existing graph-based SSL algorithms such as MAD-Exact. As part of future work, we hope to explore how such sketching ideas may be effectively used in other graph-based SSL techniques (e.g., [16]) which use loss functions other than the squared-loss.

Acknowledgments

This work was supported in part by DARPA (award number FA87501320005), IARPA (award number FA865013C7360), and Google. Any opinions, findings, conclusions and recommendations expressed in this papers are the authors' and do not necessarily reflect those of the sponsors.

References

[1] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[2] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of WWW*, 2008.

[3] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. *NIPS*, 23(163-171):3, 2010.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010.

[5] J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of computer and system sciences*, 18(2):143–154, 1979.

[6] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[7] G. Cormode and S. Muthukrishnan. Summarizing and mining skewed data streams. In *Proc. of ICDM*, pages 44–55, 2005.

[8] D. Das and N. A. Smith. Graph-based lexicon expansion with sparsity-inducing penalties. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 677–687. Association for Computational Linguistics, 2012.

[9] O. Dekel and Y. Singer. Multiclass learning by probabilistic embeddings. In *In NIPS*, pages 945–952, 2002.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. IEEE, 2009.

[11] A. Goyal, J. Jagarlamudi, H. Daumé III, and S. Venkatasubramanian. Sketch techniques for scaling distributional similarity to the web. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*. Association for Computational Linguistics, 2010.

[12] D. Rao and D. Yarowsky. Ranking and Semi-supervised Classification on Large Scale Graphs Using Map-Reduce. *TextGraphs*, 2009.

[13] T. Sarlós, A. A. Benczúr, K. Csalogány, D. Fogaras, and B. Rácz. To randomize or not to randomize: space optimal summaries for hyperlink analysis. In *Proceedings of WWW*, 2006.

[14] Q. Shi, J. Petterson, G. Dror, J. Langford, A. Smola, and S. Vishwanathan. Hash kernels for structured data. *The Journal of Machine Learning Research*, 10:2615–2637, 2009.

[15] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu. Scalable proximity estimation and link prediction in online social networks. In *Proceedings of ACM SIGCOMM*, 2009.

[16] A. Subramanya and J. Bilmes. Semi-supervised learning with measure propagation. *The Journal of Machine Learning Research*, 12:3311–3370, 2011.

[17] P. P. Talukdar and K. Crammer. New regularized algorithms for transductive learning. In *ECML-PKDD*, 2009.

[18] P. P. Talukdar and F. Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of ACL*, 2010.

[19] P. P. Talukdar, J. Reisinger, M. Pasca, D. Ravichandran, R. Bhagat, and F. Pereira. Weakly-Supervised Acquisition of Labeled Class Instances using Graph Random Walks. In *Proceedings of EMNLP*, 2008.

[20] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of ICML*, 2009.

[21] X.-M. Wu, Z. Li, A. M.-C. So, J. Wright, and S.-F. Chang. Learning with partially absorbing random walks. In *NIPS*, 2012.

[22] Y. Yang and C. Chute. An example-based mapping method for text classification and retrieval. *ACM Transactions on Information Systems*, 12(3), 1994.

[23] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.