Data Sampling for Graph Based Unsupervised Learning: Convex and Greedy Optimization

Saeed Vahidian
University of California, San Diego
San Diego, California, USA
saeed@ucsd.edu

Baharan Mirzasoleiman
Stanford University
Stanford, California, USA
baharanm@cs.stanford.edu

Alexander Cloninger
University of California, San Diego
San Diego, California, USA
acloninger@ucsd.edu

Abstract

In a number of situations, collecting a function value for every data point may be prohibitively expensive, and random sampling ignores any structure in the underlying data. We introduce a scalable optimization algorithm with no correction steps (in contrast to Frank–Wolfe and its variants), a variant of gradient ascent for coreset selection in graphs, that greedily selects a weighted subset of vertices that are deemed most important to sample. Our algorithm estimates the mean of the function by taking a weighted sum only at these vertices, and we provably bound the estimation error in terms of the location and weights of the selected vertices in the graph. In addition, we consider the case where nodes have different selection costs and provide bounds on the quality of the low-cost selected coresets. We demonstrate the benefits of our algorithm on point clouds and structured graphs, as well as sensor placement where the cost of placing sensors depends on the location of the placement. We also elucidate that the empirical convergence of our proposed method is faster than random selection and various clustering methods while still respecting sensor placement cost. The paper concludes with validation of the developed algorithm on both synthetic and real datasets, demonstrating that it performs very well compared to the current state of the art.

1 Introduction

Data summarization and dataset size reduction has recently received a lot of attention in the machine learning community. Since datasets and networks (graphs) continue to grow larger and larger over time, it is crucial for inference to be scalable while maintaining theoretical guarantees on the quality of inferential results. In this regard, an extensive set of inference algorithms in the literature falls short. A few examples include Standard MCMC algorithms, Variational methods [1], subsampling and streaming methods [2] and distributed “consensus" methods for MCMC [3] among several others. Standard MCMC algorithms typically are intractable for large-scale data. The shortcomings of such methods and results have been studied in some recent research papers. These methods have no guarantees on the quality of their inferential results.

In addition, they entail expensive iterative access to a constant fraction of the data. Motivated by the fact of redundancy of data in large datasets, an alternative approach has been modification of the dataset itself, such that its size is shrunken while preserving its original statistical properties. On this observation, [4] studied the reduction in size of a large dataset using random linear projection. On a

Preprint. Under review.

similar note, [5–7] constructed a weighted subset of a large dataset, called as a Bayesian coreset, for a wider class of Bayesian models. Even though the Bayesian coresets are easy for implementation and are not computationally complex, their construction may not reflect back the guarantees sought by a model-specific based task [8].

On the other hand, due to the proliferation of real-world network data along with natural representations of pairwise relationship of data, we are typically interested in working with data as represented by a graph structure. Hence, there is increasing interest in perusing models for such data and investigating their properties such as Bayesian generative models for graph based data (the stochastic block model) and its extensions [9]. A question that arises here is how well a million-node graph can be summarized (sampled) with a few points, in the sense that the points can be a good representative of the whole graph [10]. Some methods, such as graph clustering or community detection and betweenness centrality algorithms, can summarize a graph in terms of tightly connected clusters [11]. In contrast to the prior works in the literature which is full in comparing of clustering methods in terms of cluster quality, the authors in [12] defined "representative structure" in terms of some smooth function on the nodes of the graph, and say roughly that a subset of points is representative of the graph if the function sampled at those points has similar statistics to the function on the entire network. In particular, they give a concrete bound on the error in the mean of the function, with the bound depending on the points and weighting scheme selected.

In this work, we focus on the latter interpretation to address the data (graph) summarization. In particular, our approach builds on the work of Bayesian [7] coresets by seeking a greedy coreset of points to summarize the data, but ties the approach into the notion of active learning and function sampling. While the points chosen may be similar, we use these results to bound first moment estimates of a function sampled at those points. Similarly, we extend the previous coreset model to graphical geometries, rather than the log likelihood construction in [7]. Moreover, we consider an extended problem of having each node sampled come with a non-uniform cost, a problem that arises in applications such as sensor placement, marketing, and other knapsack type problems. We extend our algorithm to this setting, and characterize through a simple parameter the error in mean you're willing to pay in order to seek a low cost of placement set of points. In Section 2, we describe the mathematical framework of our problem and the costs we seek to optimize. In Section 3, we frame the greedy optimization algorithm for selecting points and weights, both in the setting of no cost of placement and when there is a placement cost associated. In Section 4, we prove bounds on the convergence of our algorithm and bound the mean error of a smooth function in terms of the algorithmically selected points. In Section 5, we demonstrate the success of our algorithm over random sampling and several benchmark unsupervised learning and clustering algorithms for a number of different applications.

2 General Framework

This project deals with sampling on graphs. In particular, the purpose is to choose some vertices and weights to be the best representative of the graph. More specifically, the problem is how to choose the vertices and the weight in order to have

$$\frac{1}{|V|}\sum_{v\in V} f(v) \sim \sum_{w\in W} a_w f(w). \tag{1}$$

We assume that the points V have some geometric structure encoded in a graph $G = (V, E, W)$. The graph can either be given a priori, or constructed on a point cloud $V \subset \mathbb{R}^d$ via a kernel $K : V \times V \to \mathbb{R}_+$. We must assume that the function $f : V \to \mathbb{R}$ must have some relationship to the graph, or else the optimal sampling would be a random search. In our context, this assumption takes the form that the function can be expressed in terms of a small number of eigenfunctions (with large eigenvalue) of a lazy walk graph transition matrix on the graph G.

Definition 2.1. The lazy random walk graph transition matrix P on a graph $G = (V, E, W)$ is constructed by

$$P = \frac{1}{d_{max}}(W - D) + I,$$

where D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$, $d_{max} = \max_i D_{ii}$, and I is the identity matrix.

Definition 2.2. A function $f : V \to \mathbb{R}$ is in P_λ for a lazy walk transition matrix P, with eigendecomposition $P = V\Lambda V^*$, if

$$f = \sum_{|\lambda_i| > \lambda} a_i V_i.$$

We can also consider the additional constraint that choosing nodes W may come with a cost $C : V \to \mathbb{R}_+$ of choosing the node. We seek an algorithm that will choose the subset of nodes W in a way that is

- Greedy in order to quickly choose and add additional points,
- Minimizes the error in estimation of the mean of f, and
- Incorporates the cost $C(v)$ by choosing low-cost points to sample.

The motivation for this framework and use of the lazy walk graph transition matrix comes out of the work in [12], in which it was noted that the mean error in f can be bounded in terms of the choice of points and weights, as in the following

$$\forall f \in P_\lambda, \quad \left| \frac{1}{n} \sum_{v \in V} f(v) - \frac{1}{n} \sum_{w \in W} a_w f(w) \right| \leq \|f\|_{P_\lambda} \min_{\ell \in N} \frac{1}{\lambda^\ell} \left(\left\| P^\ell \sum_{w \in W} a_w \delta_w \right\|_2^2 - \frac{1}{n} \right)^{\frac{1}{2}}. \quad (2)$$

This result implies that minimizing the right hand side in terms of a_w and W will yield a stronger bound on the moment estimation of f.

2.1 Problem Definition

Similar to the concept of duality in convex optimization, in order to achieve best results for the upper bound in (2) we shall minimize it. Therefore, the optimization problem can be formulated in the following from

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \|f\|_{P_\lambda} \frac{1}{\lambda^\ell} \left(\left\| P^\ell \sum_{w \in W} w \delta_w \right\|_2^2 - \frac{1}{n} \right)^{\frac{1}{2}} \\ \text{subject to} \quad & |W| \leq K, \; W \subset V, \\ & w > 0. \end{aligned} \quad (3)$$

Several discrete and continuous methods can be used to solve the preceding problem in (3). For example, due to the diminishing return nature of the objective function in (3), submodular minimization algorithms defined over discrete set functions, to the continuous domains, can be a candidate. Submodular functions [13], that have the intuitive notion of diminishing returns, have become highly important in a variety of machine learning applications. Examples include graph cuts in computer vision [14], data summarization [15]; active and semi-supervised learning [16].

On a similar note, another problem that is directly related to (3) is that of computing a cardinality constrained minimization by solving sparse principal component analysis (PCA) problem. In the well-known PCA analysis that has various applications in machine learning, given a sample covariance matrix, the problem of maximizing the variance explained by a linear combination of the input variables is examined while constraining the number of nonzero coefficients in this combination [17]. However, solving the sparse PCA optimization problem entails semidefinite relaxation to the problem and a greedy algorithm to calculate a full set of good solutions for all target numbers of non zero coefficients which is very expensive with total complexity of $O(n^3)$.

A better way to view this problem is in terms of an L_2-minimization problem on P. We define P_i to be the i^{th} column of P, and we also define $P(w) = \sum_i P_i w_i$. Also, we define our target function to be the vector $P^* = \frac{1}{n}\mathbb{1}$. To this end, (3) can be posed in an equivalent form as

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \|P(w) - P^*\|_2 \\ \textit{subject to} \quad & \sum_i \mathbb{1}[w_i > 0] \leq k \\ & w_i \geq 0 \end{aligned} \tag{4}$$

Problem (4) can be solved without relaxing the nonconvex cardinality constraint $\sum_i \mathbb{1}[w_i > 0] \leq k$ by either importance sampling (IS) or Frank–Wolfe (FW) algorithm. However, according to [7] there are some problems for which both FW and IS perform very poorly. In this paper, we mainly focus on the above-mentioned constrained optimization problem. In the following section, we provide a new radial optimization algorithm for problem (4) and demonstrate that it yields theoretical guarantees at a significantly reduced computational cost. More importantly, in contrast to FW and its extensions [18], the algorithm developed in this work has no correction steps and geometric error convergence.

3 Optimization Algorithm

The optimization problem in (4) is nonconvex in w. Inspired from [7], without any loss of generality, the weights, w could be scaled by an arbitrary constant $\beta \geq 0$ without affecting feasibility. This motivates rewriting (4) as

$$\begin{aligned} \underset{w,\beta}{\text{minimize}} \quad & \|\beta P(w) - P^*\|_2 \\ \textit{subject to} \quad & \sum_i \mathbb{1}[w_i > 0] \leq k \\ & w_i \geq 0, \beta \geq 0 \end{aligned} \tag{5}$$

Following [7] we now begin by solving the optimization problem in β. Afetr solving (5), we can define β^* as the solution to the problem given w which can be computed analytically as

$$\beta^* = \frac{P*}{P\,(w)} \max\left\{0, P_w^T P_*\right\} \tag{6}$$

where $P_w = \frac{P(w)}{\|P(w)\|}$ and $P_* = \frac{P*}{\|P*\|}$. We substitute β^* in the objective above, and solve instead:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \|P*\|^2 \left(1 - \max\left\{0, P_w^T P_*\right\}^2\right) \\ \textit{subject to} \quad & \sum_i \mathbb{1}[w_i > 0] \leq k \\ & w_i \geq 0 \end{aligned} \tag{7}$$

This result shows that the minimum of the problem in (7) occurs by alignment of the vectors P_w and P^*. With that in mind, we can reformulate (7) in an equivalent maximizing problem as in the following

$$\begin{aligned} \underset{w}{\text{maximize}} \quad & P_w^T P_* \\ \textit{subject to} \quad & \sum_i \mathbb{1}[w_i > 0] \leq k \\ & \|P\,(w)\| = 1 \\ & w_i \geq 0 \end{aligned} \tag{8}$$

Algorithm 1 Algorithm of SCGIGA

Initialization $w_0 \leftarrow 0, P_{w_0} \leftarrow 0$

for $k \in \{0, ..., K\}$ **do**

$a_k \leftarrow \frac{P_* - \left(P_*^T P_w\right) P_w}{\|P_* - (P_*^T P_w) P_w\|}$

$\forall v \in V, a_{kv} \leftarrow \frac{P_{*v} - \left(P_{*v}^T P_{w_k}\right) P_{w_k}}{\left\|P_{*v} - \left(P_{*v}^T P_{w_k}\right) P_{w_k}\right\|}$

$v^* \leftarrow \arg\max_{v \in V} {a_k}^T a_{kv}$ ▷ find the best vertex that maximizes the alignment.

$S \leftarrow \left\{v \in V | {a_k}^T a_{kv} \geq \kappa {a_k}^T a_{kv^*}\right\}$ ▷ find all the vertices that are within κ-percent of the maximum.

$S \leftarrow S \cup v^*$

$v_k \leftarrow \arg\min_{v \in S} C_v$ ▷ within set S find the vertex with minimum cost

$\delta_k \leftarrow \frac{P_*^T P_{*v} - \left(P_*^T P_{w_t}\right)\left(P_{*v}^T P_{w_t}\right)}{\left\|P_*^T P_{*v} - \left(P_*^T P_{w_t}\right)\left(P_{*v}^T P_{w_t}\right) + \left(P_*^T P_{w_t}\right) - \left(P_*^T P_{w_t}\right)\left(P_*^T P_{*v}\right)\right\|}$ ▷ choose the step size

$w_{k+1} \leftarrow \frac{(1-\delta_k) w_k + \delta_k P_{*v_k}}{\left\|(1-\delta_k) w_k + \delta_k P_{*v_k}\right\|}$ ▷ update the weight

$P_{w_{k+1}} \leftarrow \frac{(1-\delta_k) P_{w_t} + \delta_k P_{*v_k}}{\left\|(1-\delta_k) P_{w_t} + \delta_k P_{*v_k}\right\|}$

end

end for

Scale the weights by β

return w

According to the constraints in (8), we are optimizing the objective over a unit hypersphere rather than the simplex. Before solving the problem in (8), we extend it to a more general case and then will provide a new algorithm for solving it.

3.1 Optimizing by incorporating cost associated with each data

In many applications, selecting some reference points representing the whole data involve some factors such as cost of selection associated to each data. In financial or biological applications, each data correspond to a specific asset or gene. In problems such as these, it is natural to seek a trade-off between the two goals of minimizing the error (explaining most the difference of $P(w)$ and P^*) and the cost of choosing that reference points (data). To this end, we incorporate a new parameter C into the problem controlling the cost associated with each data. In what follows, we will focus on the reparameterized maximization problem, which can be written:

$$\begin{aligned} \underset{w}{\text{maximize}} \quad & P_w^T P_* - \lambda C(S) \\ \textit{subject to} \quad & |S| \leq k \text{ for } S = \{i : w_i > 0\} \\ & \|P(w)\| = 1 \\ & w_i \geq 0 \end{aligned} \tag{9}$$

Theorem 1. *Let C_{max}^k be the sum of the k largest elements of C. If we choose $\lambda \leq \frac{1-\kappa}{/} C_{max}^k \min \|P_i\| \sqrt{n}$, then the solution to* (9)*, P_{w^*}, satisfies*

$$P_{w^*}^T P_* \geq \kappa \max_w P_w^T P_*.$$

Proof can be found in the Appendix. This theorem implies we will never incur too much loss to the original objective by incorporating sensor cost placement. Similarly, this implies that we can make every greedy choice and step in whatever fashion is deemed best for cost, as long as the choice is within κ of the optimal step direction.

SCGIGA detailed in Algorithm 1 outlines how to solve the optimization problem in (9). It is noteworthy that SCGIGA is a general algorithm which is also valid for the case where there is no cost associated with each data. This corresponds to $\kappa = 1$ in our settings.

4 Theoretical Aspects

Here we examine the guarantees that Algorithm 1 yield for bounding the error in estimating the mean of $f \in P_\lambda$, where P_λ is the subspace spanned by the eigenfunctions of P associated with eigenvalue $\geq \lambda$. We will derive the general theorem for arbitrary κ, and as a special case we recover the results when placement cost is ignored ($\kappa = 1$). The result borrows from existing theorems in [12] and [19].

Theorem 2. *Let $f \in P_\lambda$ and assume we are given a cost of sensor placement $C(v) : V \to \mathbb{R}_+$ and a slack parameter κ. If we choose the set of points W and weights a_w using Algorithm 1 such that $|W| = K$, then*

$$\left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| \leq \frac{\|f\|_{P_\lambda}}{\lambda^\ell} \frac{\eta \, v_K}{\sqrt{n}},$$

where $v_K = O((1 - \kappa^2 \epsilon^2)^{K/2})$ for some ϵ and $\eta = \sqrt{1 - \kappa^2 \max_{i \in V} \left\langle \frac{P_i}{\|P_i\|}, \frac{1}{\sqrt{n}} \mathbb{1} \right\rangle^2}$.

The proof of the theorem can be found in the Appendix. As a particular case of this theorem, when we always choose the optimal sensor location independent of cost, we recover the following guarantee.

Corollary 3. *Let $f \in P_\lambda$, and choose the set of points W and weights a_w using Algorithm 1 such that $|W| = K$. Then*

$$\left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| \leq \frac{\|f\|_{P_\lambda}}{\lambda^\ell} \frac{\eta \, v_K}{\sqrt{n}},$$

where $v_K = O((1 - \epsilon^2)^{K/2})$ for some ϵ and $\eta = \sqrt{1 - \max_{i \in V} \left\langle \frac{P_i}{\|P_i\|}, \frac{1}{\sqrt{n}} \mathbb{1} \right\rangle^2}$.

The proof of this corollary is a special application of Theorem 2, which is proved in the Appendix.

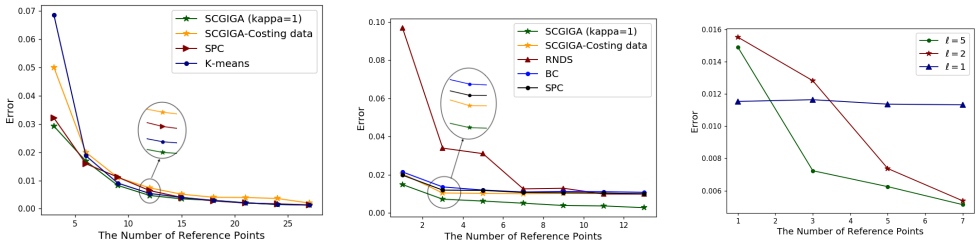
5 Empirical Evidence

In this section, we evaluate our model on some popular experiments. In order to compare both cases of our algorithm, i.e., when there is a cost on selecting each data ($\kappa \neq 1$) and when there is no cost associated with each data ($\kappa = 1$) we consider a fixed cost on each data randomly generated from a uniform distribution over $[0, 1)$. We set $\kappa = 0.8$, n=10000 and the total cost corresponding to the data is 5001.2.

Averaging a Function on Clustered Data A standard unsupervised learning task is to learn clusters from data, either on a graph or a point cloud, and use those clusters to select points and weights for averaging a function. Standard clustering algorithms include K-means clustering and spectral clustering [1].

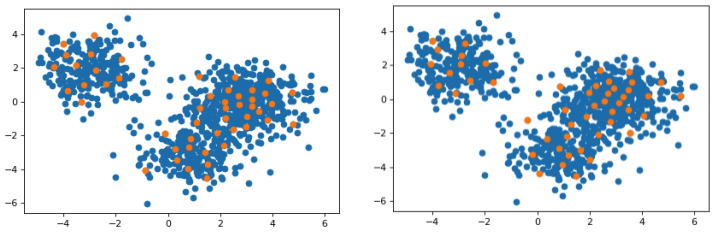
The first experiment we run is on Gaussian model with three components. The components have the mean vectors of $[1 \ -3]$, $[-3 \ 2]$, $[3 \ 0]$, and all have the covariance matrix of the identity matrix, I. The small component contains 20% of the data and the other two components contain 30% and 50% of the whole data. The function we choose to model is a simple smooth function that is an indicator function of the small cluster (1 on the small cluster, 0 on other clusters). The results in Fig. 1a show the error comparison of three algorithms including K-means and spectral clustering (SPC) versus the number of clusters (centroids) or reference points. The performance of the algorithms is quantified by an error metric which is defined as the $\mathrm{Err} = \left| \sum_{|W|=k} a_k f(k) - E(f) \right|^2$ where, a_k in K-means and SPC is defined as the ratio of the data in each cluster to the whole data, while $a_k = a_w$ in our algorithm. Here, f is the the indicator function defined on the data and $\mathrm{E}(f)$ is the mean of the function. As is evident from the figure, our algorithm outperforms K-means and spectral clustering for the whole range of the number of reference points. As can bee seen in this figure the maximum

[1] In multivariate statistics spectral clustering techniques gets rid of some of the eigenvalues of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions.



(a) Gaussian model (b) Stochastic Block Model (c) Stochastic Block Model

Figure 1: The error comparison of estimating the mean of the function defined on the clusters (the two left ones), and the impact of the shaping parameter ℓ on the performance (Right one).



(a) Cost associated. (b) No cost associated.

Figure 2: 40 data out of 1000 selected from 3 clusters for two cases, i.e., costing data (left) with $\kappa = 0.2$ and non costing data (right)

number of the reference data is 14 out of 10000 and our results reveal that the cost of the optimal solution (C_{COS}) is 5.920, while the cost we got with our cost aware algorithm, i.e., the cost of sub-optimal solution (C_{CSO}) is 0.106. The more surprising observation in these figures is that even in the case where a fixed cost associated with each data in which results in our algorithm to yield a sub-optimal solution (the solution which is in the κ percent of the optimal) our algorithm continues to do very well and work better than standard algorithms.

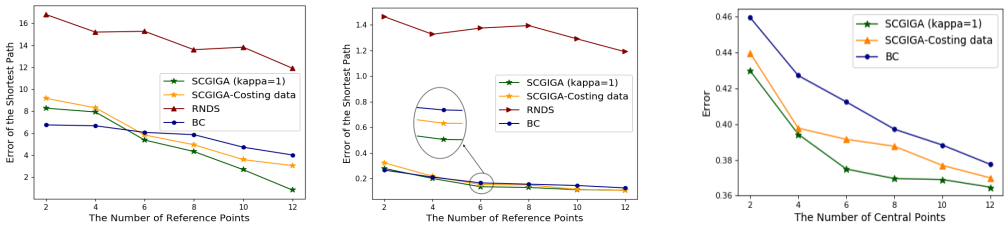
We consider another experiment on clustered graphical data. In Fig. 1b a stochastic block model with three clusters is designed where the first cluster contained 10% of the data and the other two clusters contain 50% and 40% of the data. Then we define an indicator function that is 1 on the small cluster, and 0 on the other two large clusters. Then we look for the average function value on these three clusters. As can be seen from these figures, our algorithm estimates the mean very well while random sampling (RNDS) needs more reference points to catch up the mean. More importantly, in this experiment by selecting 28 reference data, $C_{\text{COS}} = 15.159$ while $C_{\text{CSO}} = 0.075$.

Further, Fig. 1c sketches the impact of the shaping parameter, ℓ, in our formulation on the error behavior of the same stochastic block model.

Finally, in Fig. 2a, 2b three Gaussian clusters with the same mean vectors and covariance matrices as in the first experiment is considered. We construct a graph based on nearest neighbor distances (KNN=10) weighted with a Gaussian kernel. These two figures demonstrate the way that the proposed algorithm selects points when there is a fixed cost associated with each data (2a) and the special case of the latter where there is no cost for choosing each data point (2b). On a similar note, in this experiment by selecting 40 reference data, $C_{\text{COS}} = 20.300$ while $C_{\text{CSO}} = 1.811$.

The error performance of our algorithm improves as more reference points are added and that is due to the fact that it tries to selects the most useful points incrementally, creating intuitive core data (vertices) that outperforms the other methods. Interestingly, the results disclose that the slack variable, κ, can play an important role in the developed algorithm and is able to reduce the overall cost of sampling on data by several orders of magnitude.

Shortest Path on Graph Given an adjacency matrix graph representing paths between the nodes in a given graph. The goal is to find the average distance from a point to the rest of the graph, where



(a) A Powerlaw tree graph (b) A random generated graph (c) Facebook ego network

Figure 3: Shortest path comparison of different algorithms on different graphs.

the distance between two points is computed using Dijkstra's algorithm [2]. Herein this experiment rather than computing the average shortest path between each vertex of the graph and all the other vertices which is really expensive (use of Dijkstra's algorithm n^2 times), we calculate the average shortest path between all the vertices and the reference points W. The latter can be computed by using Dijkstra's algorithm only kn times, where k is the number of reference vertices.

Fig. 1a shows the results of the comparison of our algorithm, Betweenness Centrality (BC) and random sampling (RS) on a Powerlaw Tree graph [20]. In this experiment the error is defined as difference between the weighted average distance of each vertex of the graph from the reference vertices and the average distance of each vertex from all the vertices of the graph. Fig. 1b demonstrates the same results as presented in Fig. 1a for a randomly generated graph. In these two figures, both cases i.e., when a fixed cost associated with each data (vertex) and when there is no cost on the data is included.

Ego Networks Various types of real-world problems capture dependencies between records in the data. For instance, in bioinformatics, learning the way that proteins regulate the actions of other proteins is important. The dependencies of these types of problems can often be studied in the context of graphs. Herein this experiment we consider a special type of network called an Ego Network. In an Ego Network, there is a "central" vertex (ego vertex) which the network highly depends on or revolves around. We consider a real-world dataset: Facebook Ego Networks dataset. The dataset contains the aggregated network of some users' Facebook friends. In this dataset, the vertices represent individuals on Facebook, and an edge between two users means they are Facebook friends. The Ego Network connects a Facebook user to all of his Facebook friends and are then aggregated by identifying individuals who appear in multiple Ego Network. In order to measure the importance (centrality) of a user in the Facebook, some algorithms such as betweenness centrality (BC) was proposed. These algorithm select a user as a central one by looking at how many shortest paths pass through that user (vertex). The more shortest paths that pass through the user, the more central the user is in the Facebook network. We run our algorithm on the Facebook dataset to choose the central nodes. We then compare our algorithm with the BC in terms of the error performance metrics described in the first experiment. The results in Fig. 3c show that the developed algorithm in this work perform better in selecting the central points in the Facebook graph.

6 Discussion and Conclusions

In this paper, we introduced a new scalable, and theoretically-sound algorithm for a constrained optimization problem which can be applied to social network graphs, coreset construction, and on many other applications. The algorithm proposed in this work is simple to implement, reliably provide data summarization (choosing reference points) at a fraction of the cost of running algorithms such as shortest path on graphs on the full dataset (the whole graph). We provided theoretical guarantees on the convergence of the proposed algorithm and validated its efficiency empirically on some real and synthetic datasets. Our work also opens up the notion of optimizing over cost-of-placement data that can be a useful in settings where all data that contribute towards the performance also contribute towards expenses. This is the case in speech recognition, recommender systems, online advertising or dataset size reduction.

[2] Dijkstra's algorithm is a greedy nature algorithm that looks for the minimum weighted vertex on every iteration.

References

[1] Matthew D. Hoffman, David M. Blei, Chong Wang, and John W. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[2] Rémi Bardenet, Arnaud Doucet, and Christopher C. Holmes. On markov chain monte carlo methods for tall data. *Journal of Machine Learning Research*, 18:47:1–47:43, 2017.

[3] Maxim Rabinovich, Elaine Angelino, and Michael I. Jordan. Variational consensus monte carlo. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1207–1215, 2015.

[4] Rémi Bardenet and Odalric-Ambrym Maillard. A note on replacing uniform subsampling by random projections in mcmc for linear regression of tall datasets. In *2015. ffhal-01248841f*, 2015.

[5] Jonathan H. Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable bayesian logistic regression. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4080–4088, 2016.

[6] Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John W. Paisley, and David M. Blei. Variational inference via \chi upper bound minimization. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2729–2738, 2017.

[7] Trevor Campbell and Tamara Broderick. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, 2018.

[8] Michael Langberg and Leonard J. Schulman. Universal epsilon-approximators for integrals. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 598–607, 2010.

[9] Zhao Xu, Volker Tresp, Shipeng Yu, Kai Yu, and Hans-Peter Kriegel. Fast inference in infinite hidden relational models. In *Mining and Learning with Graphs, MLG 2007, Firence, Italy, August 1-3, 2007, Proceedings*, 2007.

[10] Aamir Anis, Akshay Gadde, and Antonio Ortega. Efficient sampling set selection for bandlimited graph signals using graph spectral proxies. *IEEE Trans. Signal Processing*, 64(14):3775–3789, 2016.

[11] Santo Fortunato. Community detection in graphs. *CoRR*, abs/0906.0612, 2009. URL `http://arxiv.org/abs/0906.0612`.

[12] George C Linderman and Stefan Steinerberger. Numerical integration on graphs: where to sample and how to weigh. In *arXiv preprint arXiv:1803.06989*, 2018.

[13] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978.

[14] Stefanie Jegelka and Jeff A. Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 1897–1904, 2011.

[15] Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Deletion-robust submodular maximization: Data summarization with "the right to be forgotten". In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 2449–2458, 2017.

[16] Andrew Guillory and Jeff A. Bilmes. Interactive submodular set cover. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 415–422, 2010.

[17] Alexandre d'Aspremont, Francis R. Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

[18] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 496–504, 2015.

[19] Alexander Cloninger. Bounding the error from reference set kernel maximum mean discrepancy. In *arXiv preprint arXiv:1812.04594*, 2018.

[20] William Aiello, Fan Chung Graham, and Linyuan Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10(1):53–66, 2001.

A Proof of Theorem 1

Because P is a bistochastic matrix, and we know $P_* = \frac{1}{\sqrt{n}}\mathbb{1}$, we can lower bound

$$P_w^T P* \leq 1 = \sum_i w_i \frac{P_i^T}{\|P_i\|} \frac{1}{\sqrt{n}}\mathbb{1} = \frac{1}{\sqrt{n}} \sum_i \frac{w_i}{\|P_i\|} \geq \frac{1}{\min \|P_i\| \sqrt{n}}$$

Similarly, $C(S) \leq C_{max}^k$. Now given $\lambda \leq \frac{1-\kappa}{C_{max}^k \min \|P_i\| \sqrt{n}}$, we compute

$$\begin{aligned}\max_w P_w^T P_* - \lambda C(S) &\geq \max_w P_w^T P_* - \frac{1-\kappa}{\min \|P_i\| \sqrt{n}} \geq \max_w P_w^T P_* \left(1 - \frac{1-\kappa}{\min \|P_i\| \sqrt{n}} \frac{1}{P_w^T P_*}\right) \\ &\geq max_w P_w^T P_* \left(1 - \frac{1-\kappa}{\min \|P_i\| \sqrt{n}} \min \|P_i\| \sqrt{n}\right) \geq \kappa \max_w P_w^T P_*.\end{aligned}$$

B Auxiliary Lemmas

First, we make a few statements related to initialization of the process. Lemma 3.4 from [7] directly applies to this problem, and thus $\delta_k \in [0, 1]\ \forall k$.

Lemma 4. $\langle P(w_1), P^* \rangle \geq \frac{\kappa}{\sqrt{n} \sum_i \|P_i\|}$

Proof follows equivalently to Lemma 3.1 from [7], with added caveat that our choice of weights is within κ of maximum value.

Lemma 5. *The cost aware geodesic alignment $\langle a_t, a_{t,v_k} \rangle$ satisfies*

$$\langle a_k, a_{k,v_k} \rangle \geq \kappa\tau\sqrt{J_t} \vee f(t),$$

for

$$f(x) = \kappa \frac{\sqrt{1-x}\sqrt{1-\beta^2}\epsilon + \sqrt{x}\beta}{\sqrt{1 - \left(\sqrt{x}\sqrt{1-\beta^2}\epsilon - \sqrt{1-x}\beta\right)^2}}$$

and

$$\beta = 0 \wedge \min\langle \ell_n, \frac{1}{\sqrt{n}}\mathbb{1}\rangle \text{ s.t. } \langle \ell_n, \frac{1}{\sqrt{n}}\mathbb{1}\rangle > -1.$$

Proof. The lemma is equivalent to proving Lemma 3.6 in [7] with one caveat. Here our choice of node is v_k, which comes from choosing the cheapest cost node location from the set $S = \{v \in V | \langle a_k, a_{kv} \rangle \geq \kappa \langle a_k, a_{kv_k} \rangle\}$ Because of this, we can recover all results from $\langle a_k, a_{kv_k} \rangle$ with only a constant κ in front, as our choice satisfies $\langle a_k, a_{kv_k} \rangle \geq \kappa \langle a_k, a_{kv_k} \rangle$. □

We apply Lemma 5 to prove the following Theorem that is needed, and mirrors the results from [7].

Theorem 6. *Assume a cost of sensor placement $C(v) : V \to \mathbb{R}_+$ and a slack parameter κ. If we choose the set of points W and weights a_w using Algorithm 1 such that $|W| = K$, then*

$$\|P(w) - \frac{1}{n}\mathbb{1}\| \leq \frac{\eta\, v_K}{\sqrt{n}},$$

where $v_K = O((1-\kappa^2\epsilon^2)^{K/2})$ for some ϵ and $\eta = \sqrt{1 - \kappa^2 \max_{i \in V} \left\langle \frac{P_i}{\|P_i\|}, \frac{1}{\sqrt{n}}\mathbb{1} \right\rangle^2}$.

Proof. We mimic the results from [7], incorporating the additional cost parameter. We denote $J_k := 1 - \langle \frac{P(w_k)}{\|P(w_k)\|}, \frac{1}{\sqrt{n}}\mathbb{1} \rangle$. If we substitute this into the formula for δ_t, we get

$$J_{k+1} = J_k(1 - \langle a_t, a_{kv_k} \rangle^2).$$

Applying our bound from Lemma 5, we get

$$J_{k+1} \leq J_k \left(1 - \kappa^2\tau^2 J_k\right).$$

By applying the standard induction argument used in [7], we get

$$J_k \leq B(k) := \frac{J_1}{1 + \kappa^2\tau^2(k-1)}.$$

Because $B(k)$ still goes to 0, and $f(B(k)) \to \kappa\epsilon$, there exists a k^* such that $f(B(k)) \geq \kappa\tau\sqrt{B(k)}$, and since f is monotonic decreasing, $f(J_t) > f(B(k))$. Using Lemma 5, we finish with

$$J_k \leq B(k \wedge k^*) \prod_{s=k^*+1}^{k} (1 - f^2(B(s)))$$

We note that $\frac{1}{n} J_k = \|\beta^* P(w) - P^*\|^2$, so this means

$$\|\beta^* P(w) - P^*\| \leq \frac{\eta\, C_K}{\sqrt{n}},$$

for constant C_K combining the denominator in $B(k)$ and the product of $\prod_{s=k^*+1} k1 - f^2(B(s))$, and $\sqrt{J_1} = \eta$. Notice that $f(B(k)) \to \kappa\epsilon$ shows a rate of decay of $v = \sqrt{1 - \kappa^2\epsilon^2}$. □

C Proof of Theorem 2

We note that [12] proves multiple bounds on $\left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right|$. The main bound in the paper comes from using the fact that they assume $\sum_w a_w = 1$, which allows them to break up the inner product $\|P \sum_w a_w \delta_w - \frac{1}{n}\mathbb{1}\|$ into its subsequent terms $\left(\|P \sum_w a_w \delta_w\|^2 - \frac{1}{n}\right)^{1/2}$. We step away from this assumption and will instead work directly with the norm $\left\| P \sum_w a_w \delta_w - \frac{1}{n}\mathbb{1} \right\| = \|P(w) - P^*\|$.

By the same logic as in [12], we know

$$\left| \frac{1}{n} \sum_{v \in V} f(v) - \sum_{w \in W} a_w f(w) \right| \leq \frac{\|f\|_{P_\lambda}}{\lambda^\ell} \min_{\beta, w} \|\beta P^\ell(w) - P^*\|.$$

We can simply replace $\widetilde{P} = P^\ell$ and inherit on $\widetilde{P}$ in Theorem 6, in particular that we still have $\sum_i \frac{1}{n} \widetilde{P}_i = \frac{1}{n}\mathbb{1}$. Thus, we can apply the guarantees of Algorithm 1 and Theorem 6 to bound $\|\beta P^\ell(w) - P^*\| \leq \frac{\eta\, v_K}{\sqrt{n}}$ and attain the desired result.