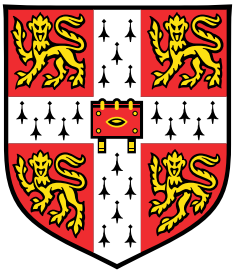
Data Summarizations for Scalable and Privacy-Aware Learning in High Dimensions



Dionysis Manousakas

Department of Computer Science and Technology
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Darwin College

August 2020

In order to write a single line, one must see a great many cities, people and things, have an understanding of animals, sense how it is to be a bird in flight, and know the manner in which the little flowers open every morning. In one's mind there must be regions unknown, meetings unexpected and long-anticipated partings, to which one can cast back one's thoughts—childhood days that still retain their mystery, [...] days in peacefully secluded rooms and mornings beside the sea, and the sea itself, seas, nights on journeys that swept by on high and flew past filled with stars ... And it is not yet enough to have memories ... Only when they become the very blood within us, our every look and gesture, nameless and no longer distinguishable from our inmost self, only then, in the rarest of hours, [...] can the first word arise in their midst and go out among them.

Rainer Maria Rilke

The Notebooks of Malte Laurids Brigge (translated by Michael Hulse)

To my parents

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the preface and specified in the text. It is not substantially the same as any work that has already been submitted before for any degree or other qualification except as declared in the preface and specified in the text. It does not exceed the prescribed word limit of 65,000 words for the Computer Science Degree Committee, including appendices, footnotes, tables and equations.

Dionysis Manousakas
August 2020

Acknowledgements

TBC

Abstract

...

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Chapter 2

Background

Chapter 3

Quantifying Privacy Loss of Human Mobility Graph Topology

Human mobility is often represented as a mobility network, or graph, with nodes representing places of significance which an individual visits, such as their home, work, places of social amenity, etc., and edge weights corresponding to probability estimates of movements between these places. Previous research has shown that individuals can be identified by a small number of geolocated nodes in their mobility network, rendering mobility trace anonymization a hard task. In this paper we build on prior work and demonstrate that even when all location and timestamp information is removed from nodes, the graph topology of an individual mobility network itself is often uniquely identifying. Further, we observe that a mobility network is often unique, even when only a small number of the most popular nodes and edges are considered. We evaluate our approach using a large dataset of cell-tower location traces from 1 500 smartphone handsets with a mean duration of 430 days. We process the data to derive the top$-N$ places visited by the device in the trace, and find that 93% of traces have a unique top-10 mobility network, and all traces are unique when considering top-15 mobility networks. Since mobility patterns, and therefore mobility networks for an individual, vary over time, we use graph kernel distance functions, to determine whether two mobility networks, taken at different points in time, represent the same individual. We then show that our distance metrics, while imperfect predictors, perform significantly better than a random strategy and therefore our approach represents a significant loss in privacy.

3.1 Introduction

Our mobile devices collect a significant amount of data about us and location data of individuals are particularly privacy sensitive. Furthermore, previous work has shown that removing direct identifiers from mobility traces does not provide anonymity: users can easily be reidentified by a small number of unique locations that they visit frequently [23, 129].

Consequently, some approaches have been proposed that protect location privacy by replacing location coordinates with encrypted identifiers, using different encryption keys for each location trace in the population. This preprocessing results in locations that are strictly user-specific and cannot be cross-referenced between users. Examples include the research track of the Nokia Mobile Data Challenge,[1] where visited places were represented by random integers [65]; and identifiable location information collected by the Device Analyzer dataset,[2] including WiFi access point MAC addresses and cell tower identifiers, are mapped to a set of pseudonyms defined separately for each handset [113]. Moreover, temporal resolution may also be deliberately decreased to improve anonymization [46] since previous work has demonstrated that sparsity in the temporal evolution of mobility can cause privacy breaches [23].

In this paper, *we examine the degree to which mobility traces without either semantically-meaningful location labels, or fine-grained temporal information, are identifying.* To do so, we represent location data for an individual as a mobility network, where nodes correspond to abstract locations and edges to their connectivity, i.e. the respective transitions made by an individual between locations. We then examine whether or not these graphs reflect user-specific behavioural attributes that could act as a fingerprint, perhaps allowing the re-identification of the individual they represent. In particular, we show how graph kernel distance functions [112] can be used to assist reidentification of anonymous mobility networks. This opens up new opportunities for both attack and defense. For example, patterns found in mobility networks could be used to support automated user verification where the mobility network acts as a behavioural signature of the legitimate user of the device. However the technique could also be used to link together different user profiles which represent the same individual.

Our approach differs from previous studies in location data deanonymization [42, 24, 81, 45], in that *we aim to quantify the breach risk in preprocessed location data that do not disclose explicit geographic information*, and where instead locations are replaced with a set of user-specific pseudonyms. Moreover, we also do not assume specific timing information for the visits to abstract locations, *merely ordering*.

We evaluate the power of our approach over a large dataset of traces from 1 500 smartphones, where cell tower identifiers (*cid*s) are used for localization. Our results show that the data contains structural information which may uniquely identify users. This fact then supports the development of techniques to efficiently reidentify individual mobility profiles. Conversely, our analysis may also support the development of techniques to cluster into larger groups with similar mobility; such an approach may then be able to offer better anonymity guarantees.

A summary of our contributions is as follows:

[1] `http://www.idiap.ch/project/mdc`

[2] `https://deviceanalyzer.cl.cam.ac.uk`

- We show that network representations of individual longitudinal mobility display distinct topology, even for a small number of nodes corresponding to the most frequently visited locations.
- We evaluate the sizes of identifiability sets formed in a large population of mobile users for increasing network size. Our empirical results demonstrate that all networks become quickly uniquely identifiable with less than 20 locations.
- We propose kernel-based distance metrics to quantify mobility network similarity in the absence of semantically meaningful spatial labels or fine-grained temporal information.
- Based on these distance metrics, we devise a probabilistic retrieval mechanism to reidentify pseudonymized mobility traces.
- We evaluate our methods over a large dataset of smartphone mobility traces. We consider an attack scenario where an adversary has access to historical mobility networks of the population she tries to deanonymize. We show that by informing her retrieval mechanism with structural similarity information computed via a deep shortest-path graph kernel, the adversary can achieve a median deanonymization probability 3.52 times higher than a randomised mechanism using no structural information contained in the mobility networks.

3.2 Related Work

3.2.1 Mobility Deanonymization

Protecting the anonymity of personal mobility is notoriously difficult due to sparsity [3] and hence mobility data are often vulnerable to deanonymization attacks [82]. Numerous studies into location privacy have shown that even when an individual's data are anonymized, they continue to possess unique patterns that can be exploited by a malicious adversary with access to auxiliary information. Zang et al. analysed nationwide call-data records (CDRs) and showed that the N most frequently visited places, so called $top{-}N$ data, correlated with publicly released side information and resulted in privacy risks, even for small values of Ns [129]. This finding underlines the need for reductions in spatial or temporal data fidelity before publication. De Montjoye et al. quantified the unicity of human mobility on a mobile phone dataset of approximately $1.5M$ users with intrinsic temporal resolution of one hour and a 15-month measurement period [23]. They found that four random spatio-temporal points suffice to uniquely identify 95% of the traces. They also observe that the uniqueness of traces decreases as a power law of spatio-temporal granularity, stressing the hardness of achieving privacy via obfuscation of time and space information.

Several inference attacks on longitudinal mobility are based on probabilistic models trained on individual traces and rely on the regularity of human mobility. Mulder et al.

developed a re-identification technique by building a Markov model for each individual in the training set, and then using this to re-identify individuals in the test set by likelihood maximisation [24]. Similarly, Gambs et al. used Markov chains to model mobility traces in support of re-identification [42].

Naini et al. explored the privacy impact of releasing statistics of individuals mobility traces in the form of histograms, instead of their actual location information [81]. They demonstrated that even this statistical information suffices to successfully recover the identity of individuals in datasets of few hundred people, via jointly matching labeled and unlabeled histograms of a population. Other researchers have investigated the privacy threats from information sharing on location-based social networks, including the impact of location semantics on the difficulty of re-identification [95] and location inference [4].

All the above previous work assumes locations are expressed using a universal symbol or global identifier, either corresponding to (potentially obfuscated) geographic coordinates, or pseudonymous stay points. Hence, cross-referencing between individuals in the population is possible. This is inapplicable when location information is anonymised separately for each individual. Lin et al. presented a user verification method in this setting [69]. It is based on statistical profiles of individual indoor and outdoor mobility, including cell tower ID and WiFi access point information. In contrast, we employ network representations based solely on cell tower ID sequences without explicit time information.

Often, studies in human mobility aim to model properties of a population, thus location data are published as aggregate statistics computed over the locations of individuals. This has traditionally been considered a secure way to obfuscate the sensitive information contained in individual location data, especially when released aggregates conform to $k-$anonymity [107] principles. However, recent results have questioned this assumption. Xu et al. recovered movement trajectories of individuals with accuracy levels of between 73% and 91% from aggregate location information computed from cellular location information involving 100 000 users [124]. Similarly, Pyrgelis et al. performed a set of inference attacks on aggregate location time-series data and detected serious privacy loss, even when individual data are perturbed by differential privacy mechanisms before aggregation [90].

3.2.2 Anonymity of Graph Data

Most of the aforementioned data can be represented as *microdata* with rows of fixed dimensionality in a table. Microdata can thus be embedded into a vector space. In other applications, datapoints are *relational* and can be naturally represented as *graphs*. Measuring the similarity of such data is significantly more challenging, since there is no definitive method. Deanonymization attacks on graphs have mostly been studied in the context of social networks and aimed to either align nodes between an auxiliary and an unknown targeted graph [83, 99], or quantify the leakage of private information of a graph node via its neighbors [132].

In the problem studied here, *each individual's information is an entire graph*, rather than a node in a graph or a node attribute, and thus deanonymization is reduced to a graph matching or classification problem. To the best of our knowledge, this is the first attempt to deanonymize an individual's structured data by applying graph similarity metrics. Since we are looking at relational data, not microdata, standard theoretical results on microdata anonymization, such as differential privacy [32], are not directly applicable. However, metrics related to structural similiarity, including $k-$anonymity, can be generalized in this framework.

3.3 Proposed Methodology

In this section, we first adapt the privacy framework of $k-$anonymity to the case of graph data (Section 3.3.1). Next we introduce our methodology: We assume that all mobility data are initially represented as a sequence of pseudonymous locations. We also assume the pseudonymisation process is distinct per user, and therefore locations cannot be compared between individuals. In other words, it is not possible to determine whether pseudonymous location l_u for user u is the same as (or different from) location l_v for user v. We convert a location sequence for each user into a mobility network (Section 3.3.2). We then extract feature representations of these networks and embed them into a vector space. Finally, in the vector space, we can define pairwise distances between the network embeddings (Section 3.3.3) and use them in a deanonymization scenario (Section 3.3.4).

Our methodology is, in principle, applicable to many other categories of recurrent behavioural trajectories that can be abstracted as graphs, such web browsing sessions [85, 128] or smartphone application usage sequences [120]. We leave such analysis as future work.

3.3.1 $k-$anonymity on Graphs

Anonymity among networks refers to topological (or structural) equivalence. In our analysis we adopt the privacy framework of $k-$anonymity [107] which we summarize as follows:

Definition 1 (($k-$**anonymity**))**.** A microdata release of statistics containing separate entries for a number of individuals in the population satisfies the $k-$anonymity property if the information for each individual contained in the release is indistinguishable from at least $k-1$ other individuals whose information also appears in the release.

Therefore we interpret $k-$anonymity in this paper to mean that the mobility network of an individual in a population should be identical to the mobility network of at least $k-1$ other individuals. Recent work casts doubt on the protection guarantees offered by $k-$anonymity in location privacy [102], motivating the definition of $l-$diversity [72] and $t-$closeness [68]. Although $k-$anonymity may be insufficient to ensure privacy in the presence of adversarial knowledge, $k-$anonymity is a good metric to use to measure the uniqueness of an individual

in the data. Moreover, this framework is straightforwardly generalizable to the case of graph data.

Structural equivalence in the space of graphs corresponds to isomorphism and, based on this, we can define $k-$*anonymity on unweighted graphs* as follows:

Definition 2 (**Graph Isomorphism**)**.** Two graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* (or *belong to the same isomorphism class*) if there exists a bijective mapping $g : V \to V'$ such that $(v_i, v_j) \in E$ iff $(g(v_i), g(v_j)) \in E'$.

Definition 3 (**Graph** $k-$**anonymity**)**.** *Graph* $k-$*anonymity* is the minimum cardinality of isomorphism classes within a population of graphs.

After clustering our population of graphs into isomorphism classes, we can also define the *identifiability set* and *anonymity size* [88] as follows:

Definition 4 (**Identifiability Set**)**.** *Identifiability set* is the percentage of the population which is uniquely identified given their top$-N$ network.

Definition 5 (**Anonymity Size**)**.** The *anonymity size* of a network within a population is the cardinality of the isomorphism class to which the network belongs.

3.3.2 Mobility Information Networks

To study the topological patterns of mobility, we represent user movements by a mobility network. A preliminary step is to check whether a first-order network is a reasonable representation of movement data, or whether a higher-order network is required.

First-order network representations of mobility traces are built on the assumption of a *first-order temporal correlation* among their states. In the case of mobility data, this means that the transition by an individual to the next location in the mobility network can be accurately modelled by considering only their current location. For example, the probability that an individual visits the shops or work next depends only on where they are located now, and a more detailed past history of places recently visited does not offer significant improvements to the model. The alternative is that a sequences of the states are better modelled by higher-order Markov chains, namely that transitions depend on the current state and one or more previously visited states. For example, the probability that an individual visits the shops or work next depends not only on where they are now, but where they were earlier in the day or week. If higher-order Markov chains are required, we should assume a larger state-space and use these states as the nodes of our individual mobility networks. Recently proposed methods on optimal order selection of sequential data [125, 97] can be directly applied at this step.

Let us assume a mobility dataset from a population of users $u \in U$. We introduce two network representations of user's mobility.

Definition 6 (**State Connectivity Network**)**.** A **state connectivity network** for u is an unweighted directed graph $C^u = (V^u, E^u)$. Nodes $v_i \in V^u$ correspond to states visited by the user throughout the observation period. An edge $e_{ij} = (v_i^u, v_j^u) \in E^u$ represents the information that u had at least one recorded transition from v_i^u to v_j^u.

Definition 7 (**Mobility Network**)**.** A **mobility network** for u is a weighted and directed graph $G^u = (V^u, E^u, W^u) \in \mathcal{G}$, with the same topology as the state connectivity network and additionally an edge weight function $W^u : E^u \to \mathbb{R}^+$. The weight function assigns a frequency w_{ij}^u to each edge e_{ij}^u, which corresponds to the number of transitions from v_i^u to v_j^u recorded throughout the observation period.

To facilitate comparisons of frequencies across networks of different sizes in our experiments, we normalize edge weights on each mobility network to sum to 1.

In first-order networks, nodes correspond to distinct places the user visits. Given a high-frequency, timestamped, sequence of location events for a user, distinct places can be extracted as small geographic regions where a user stays longer than a defined time interval, using existing clustering algorithms [59]. Nodes in the mobility network have no geographic or timing information associated with them. Nodes may have *attributes* attached to them reflecting additional side information. For example, in this paper we consider whether attaching the frequency of visits a user makes to a specific node aids an attacker attempting to deanonymize the user.

In some of our experiments, we prune the mobility networks of users by reducing the size of the mobility network to the N most frequent places and rearranging the edges in the network accordingly. We refer to these networks as **top**$-N$ **mobility networks**.

3.3.3 Graph Similarity Metrics

It is not possible to apply a graph isomorphism test to two mobility networks to determine if they represent the same underlying user because a user's mobility network is likely to vary over time. Therefore we need distance functions that can measure the degree of similarity between two graphs. Distance functions decompose the graph into feature vectors (smaller substructures and pattern counts), or histograms of graph statistics, and express similarity as the distance between those feature representations. In the following, we introduce the notion of graph kernels and describe the graph similarity metrics used later in our experiments.

We wish to compute the similarity between two graphs $G, G' \in \mathcal{G}$. Kernel functions [112], or *kernels*, are symmetric positive semidefinite functions, where $K(G, G') : \mathcal{G} \times \mathcal{G} \to \mathcal{R}^+$, meaning that for all $n > 1$, $G_1, ..., G_n \in \mathcal{G}$, and $c_1, ..., c_n \in \mathcal{R}$, we have $\sum_{i,j=1}^{n} c_i c_j K(G_i, G_j) \geq 0$. Each kernel function corresponds to some feature map $\phi(G)$, where the kernel function can be expressed as the inner product between feature maps, i.e., $K(G, G') = \langle \phi(G), \phi(G') \rangle$.

In order to ensure the result from the kernel lies in the range from -1 to 1 inclusive, we apply *cosine normalization* as follows:

$$K(G, G') = \left\langle \frac{\phi(G)}{||\phi(G)||}, \frac{\phi(G')}{||\phi(G')||} \right\rangle. \tag{3.1}$$

One interpretation of this function is as the *cosine similarity of the graphs in the feature space* defined by the map of the kernel.

In our experiments we apply a number of scalable kernel functions on our mobility datasets and assess their suitability for deanonymization applications on mobility networks. We note in advance that as the degree distribution and all substructure counts of a graph remain unchanged under structure-preserving bijection of the vertex set, all examined graph kernels are invariant under isomorphism. We briefly introduce these kernels in the remainder of the section.

Kernels on degree distribution

The degree distribution of nodes in the graph can be used to quantify the similarity between two graphs. For example, we can use a histogram of weighted or unweighted node degree as a feature vector. We can then compute the pairwise distance of two graphs by taking either the inner product of the feature vectors or passing them through a Gaussian radial basis function (RBF) kernel:

$$K(G, G') = \exp\left(- \frac{||\phi(G) - \phi(G')||^2}{2\sigma^2} \right). \tag{3.2}$$

Here, the parameters of the kernel are the variance σ (in case RBF is used) and the number of bins in the histogram.

Kernel on graph atomic substructures

Kernels can use counts on substructures, such as subtree patterns, shortest paths, walks or limited-size subgraphs. This family of kernels are called $R-$*convolution graph kernels* [48]. In this way, graphs are represented as vectors with elements corresponding to the frequency of each such substructure over the graph. Hence, if $s_1, s_2, ... \in \mathcal{S}$ are the substructures of interest and $\#(s_i \in G)$ the counts of s_i in graph G, we get as feature map vectors

$$\phi(G) = [\#(s_1 \in G), \#(s_2 \in G), ...]^T \tag{3.3}$$

with dimension $|\mathcal{S}|$ and kernel

$$K(G, G') = \sum_{s \in \mathcal{S}} \#(s \in G)\#(s \in G'). \tag{3.4}$$

In the following, we briefly present some kernels in this category and explain how they are adapted in our experiments.

Shortest-Path Kernel

The Shortest-Path (*SP*) graph kernel [14] expresses the similarity between two graphs by counting the co-occuring shortest paths in the graphs. It can be written in the form of equation (3) where each element $s_i \in \mathcal{S}$ is a triplet $(a^i_{\text{start}}, a^i_{\text{end}}, n)$, where n is the length of the path and $a^i_{\text{start}}, a^i_{\text{end}}$ the attributes of the starting and ending nodes. The shortest path set is computable in polynomial time using, for example, the Floyd-Warshall algorithm, with complexity $O(|V|^4)$, where $|V|$ is number of nodes in the network.

Weisfeiler-Lehman Subtree Kernel

Shervashidze et al. proposed an efficient method [101] to construct a graph kernel utilizing the Weisfeiler-Lehman (*WL*) test of isomorphism [119]. The idea of the *WL* kernel is to measure co-occurences of subtree patterns across node attributed graphs.

Computation progresses over iterations as follows:

1. each node attribute is augmented with a multiset of attributes from adjacent nodes;
2. each node attribute is then compressed into a single attribute label for the next iteration; and
3. the above steps are repeated until a specified threshold h is reached.

An example is shown in Figure 3.1.

If G and G' are the two graphs, the *WL* subtree kernel is defined as follows:

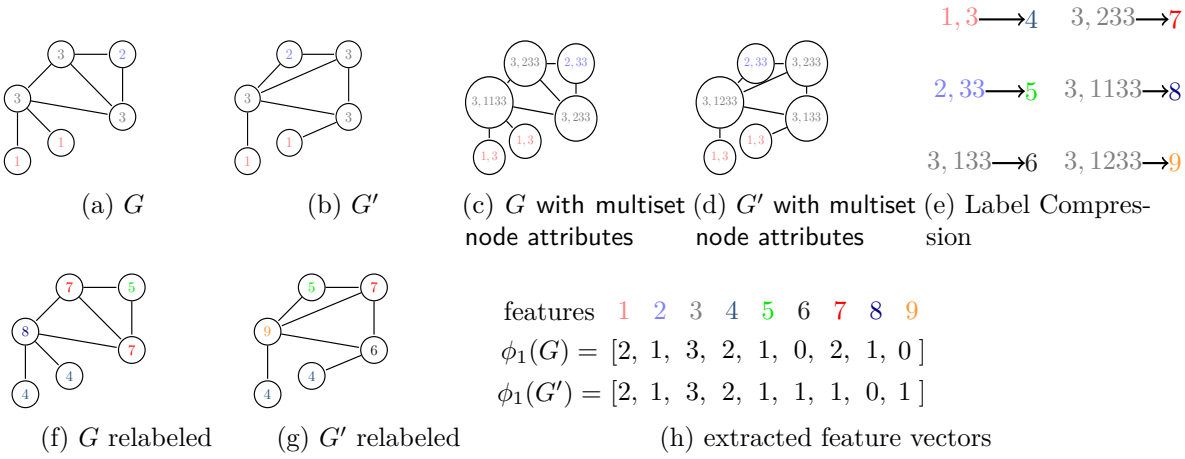
$$K^h_{WL}(G, G') = \langle \phi_h(G), \phi_h(G') \rangle, \tag{3.5}$$

where $\phi_h(G)$ and $\phi_h(G')$ are the vectors of labels extracted after running h steps of the computation (Figure 3.1h). They consist of h blocks, where the $i-$th component of the $j-$th block corresponds to the frequency of label i at the $j-$th iteration of the computation. The computational complexity of the kernel scales *linearly* with the number of edges $|E|$ and the length h of the WL graph sequence.

Deep Graph Kernels

Deep graph kernels (*DK*s) are a unified framework that takes into account similarity relations at the level of atomic substructures in the kernel computation [127]. Hence, these kernels can quantify *similar substructure* co-occurence, offering more robust feature representations. DKs are based on computing the following inner product:

$$K(G, G') = \phi(G)^T M \phi(G'), \tag{3.6}$$



(a) G (b) G' (c) G with multiset node attributes (d) G' with multiset node attributes (e) Label Compression

(f) G relabeled (g) G' relabeled (h) extracted feature vectors

Figure 3.1: Computation of the Weisfeiler-Lehman subtree kernel of height $h = 1$ for two attributed graphs.

where ϕ is the feature mapping of a classical R-convolution graph kernel.

In the above, $M : |\mathcal{V}| \times |\mathcal{V}|$ is a positive-definitive matrix encoding the relationships between the atomic substructures and $\mathcal{V}$ is the vocabulary of the observed substructures in the dataset. Here, M can be defined using the edit distance of the substructures, i.e. the number of elementary operations to transform one substructure to another; or M can be learnt from the data, applying relevant neural language modeling methods [77].

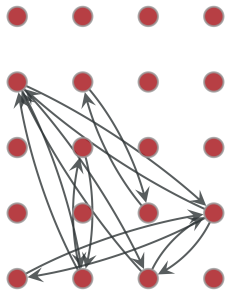
3.3.4 Deanonymization of User Mobility Networks and Privacy Leakage Evaluation

Hypothesis

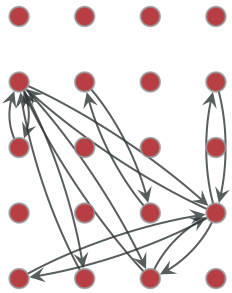
The basic premise of our deanonymization approach can be postulated as follows:

The mobility of a person across different time periods is stochastic, but largely recurrent and stationary, and its expression at the level of the individual mobility network is discriminative enough to reduce a person's privacy within a population.

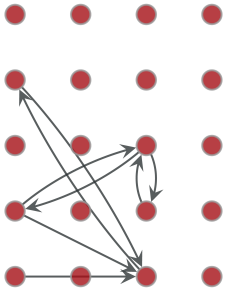
For example, the daily commute to work corresponds to a relatively stable sequence of cell towers. This can be expressed in the mobility network of the user as a persistent subgraph and forms a characteristic behavioural pattern that can be exploited for deanonymization of mobility traces. Empirical evidence for our hypothesis is shown in Figure 3.2. For ease of presentation, in the figure, nodes between the disparate observation periods of the users can be cross-referenced. We assume that cross-referencing is not possible in our attack scenario, as locations are independently pseudonymized.



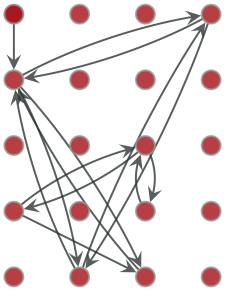
(a) **user 1**: first half of the observation period



(b) **user 1**: second half of the observation period



(c) **user 2**: first half of the observation period



(d) **user 2**: second half of the observation period

Figure 3.2: Top−20 networks for two random users from the Device Analyzer dataset. Depicted edges correspond to the 10% most frequent transitions in the respective observation window. The networks show a high degree of similarity between the mobility profiles of the same user over the two observation periods. Moreover, the presence of single directed edges in the profile of **user 2** forms a discriminative pattern that allows us to distinguish **user 2** from **user 1**.

Threat Model

We assume an adversary has access to a *set of mobility networks* $G \in \mathcal{G}_{\text{training}}$ *with disclosed identities (or labels)* $l_G \in \mathcal{L}$ and a *set of mobility networks* $G' \in \mathcal{G}_{\text{test}}$ *with undisclosed identities* $l_{G'} \in \mathcal{L}$.[3]

We define a normalised similarity metric among the networks $K : \mathcal{G}_{\text{training}} \times \mathcal{G}_{\text{test}} \to \mathcal{R}^+$. We hypothesize that a training and test mobility network belonging to the same person have common or similar connectivity patterns, thus a high degree of similarity.

The intention of an adversary is to deanonymize a given test network $G' \in \mathcal{G}_{\text{test}}$, by appropriately defining a vector of probabilities over the possible identities in $\mathcal{L}$.

[3]Generally we can think of $l_{G'} \in \mathcal{J} \supset \mathcal{L}$ and assign some fixed probability mass to the labels $l_{G'} \in \mathcal{J} \setminus \mathcal{L}$. However, here we make the *closed world assumption* that the training and test networks come from the same population. We make this assumption for two reasons: first, it is a common assumption in works on deanonymization and, second, we cannot directly update our beliefs on $l_{G'} \in \mathcal{J} \setminus \mathcal{L}$ by observing samples from $\mathcal{L}$.

An **uninformed adversary** has *no information* about the networks of the population and in the absence of any other side knowledge, the prior belief of the adversary about the identity of G' is a uniform distribution over all possible identities:

$$P(l_{G'} = l_{G_i}) = 1/|\mathcal{L}|, \text{ for every } G_i \in \mathcal{G}_{\text{training}}. \tag{3.7}$$

An **informed adversary** has *access to the population of training networks* and can compute the pairwise similarities of G' with each $G_i \in \mathcal{G}_{\text{training}}$ using a kernel function K. Hence the adversary can update her belief for the possible identities in $\mathcal{L}$ according to the values of K. Therefore, when the adversary attempts to deanonymize identities in the data, she assigns probabilities that follow a *non-decreasing function* of the computed pairwise similarity of each label. Denoting this function by f, we can write the updated adversarial probability estimate for each identity as follows:

$$P(l_{G'} = l_{G_i} | \mathcal{G}_{\text{training}}, K) = \frac{f(K(G_i, G'))}{\sum_{j \in \mathcal{L}} f(K(G_j, G'))}, \quad \text{for every } G_i \in \mathcal{G}_{\text{training}}. \tag{3.8}$$

Privacy Loss

In the case of the uninformed adversary, the true label for any user is expected to have rank $|\mathcal{L}|/2$. Under this policy, the amount of privacy for each user is proportional to the size of the population.

In the case of the informed adversary, knowledge of $\mathcal{G}_{\text{training}}$ and the use of K will induce some positive *privacy loss* which will result in the expected rank of user to be smaller than $|\mathcal{L}|/2$. The privacy loss can be quantified as follows:

$$PL(G'; \mathcal{G}_{\text{training}}, K) = \frac{P(l_{G'} = l_{G'_{\text{true}}} | \mathcal{G}_{\text{training}}, K)}{P(l_{G'} = l_{G'_{\text{true}}})} - 1 \tag{3.9}$$

A privacy loss equal to zero reflects no information gain compared to an uninformed adversary with no access to graphs with disclosed identities.

Let us assume that the users of our population generate distinct mobility networks. As will be supported with empirical evidence in the next section, this is often the case in real-world *cid* datasets of few thousand users even for small networks sizes (e.g. for top-20 networks in our dataset). Under the above premise, the maximal privacy loss occurs when the presented test network is an identical copy of a training network of the same user which exists in the data of the adversary, i.e. $G' \in \mathcal{G}_{\text{training}}$. This corresponds to a user deterministically

repeating her mobility patterns over the observation period recorded in the test network. In such a scenario, we could think that isomorphism tests are the most natural way to compute similarity; however, isomorphism tests will be useless in real-world scenarios, since the stochastic nature and noise inherent in the mobility networks of a user would make them non-isomorphic. Maximal privacy loss reflects the discriminative ability of the kernel and cannot be exceeded in real-world datasets, where the test networks are expected to be noisy copies of the training networks existing in our system. The step of comparing with the set of training networks adds computational complexity of $\mathcal{O}(|\mathcal{G}_{\text{training}}|)$ to the similarity metric cost.

Moreover, our framework can naturally facilitate incorporating new data to our beliefs when multiple examples per individual exist in the training dataset. For example, when multiple instances of mobility networks per user are available, we can use $k-$nearest neighbors techniques in the comparison of distances with the test graph.

3.4 Data for Analysis

In this section we present an exploratory analysis of the dataset used in our experiments, highlighting statistical properties of the data and empirical results regarding the structural anonymity of the generated state connectivity networks.

3.4.1 Data Description

We evaluate our methodology on the Device Analyzer dataset [113]. Device Analyzer contains records of smartphone usage collected from over 30 000 study participants around the globe. Collected data include information about system status and parameters, running background processes, cellular connectivity and wireless connectivity. For privacy purposes, released *cid* information is given a unqiue pseudonym separately for each user and contains no geographic, or semantic, information concerning the location of users. Thus we cannot determine geographic proximity between the nodes and the location data of two users cannot be directly aligned.

For our experiments, we analysed *cid* information collected from 1 500 handsets with the largest recorded location datapoints in the dataset. Figure 3.4a shows the observation period for these handsets; note that the mean is greater than one year but there is lot of variance across the population. We selected these 1 500 handsets in order to examine the re-identifiability of devices with rich longitudinal mobility profiles. This allowed us to study the various attributes of individual mobility affecting privacy in detail. As mentioned in the previous section, the cost of computing the adversarial posterior probability for the deanonymization of a given unlabeled network scales linearly with the population size.

Networks	# of networks	Num. of nodes, avg.	Edges, avg.	Density, avg.	Avg. clust. coef.	Diameter, avg.	Avg. short. path	Recurrence rate (%)
top−50 locations	1500	49.92 ± 1.26	236.55 ± 78.14	0.19 ± 0.06	0.70 ± 0.07	3.42 ± 0.86	1.93 ± 0.20	84.7 ± 5.6
top−100 locations	1500	98.33 ± 7.93	387.05 ± 144.73	0.08 ± 0.03	0.60 ± 0.10	4.67 ± 1.48	2.33 ± 0.40	78.3 ± 7.8
top−200 locations	1500	179.23 ± 37.82	548.21 ± 246.11	0.04 ± 0.02	0.47 ± 0.12	7.52 ± 4.21	3.07 ± 1.18	73.0 ± 9.9
full	1500	334.60 ± 235.81	741.64 ± 527.28	0.02 ± 0.02	0.33 ± 0.09	15.98 ± 10.18	4.84 ± 2.93	68.8 ± 12.3

Table 3.1: Summary statistics of mobility networks in the Device Analyzer dataset.

3.4.2 Mobility Networks Construction

We began by selecting the optimal order of the network representations derived from the mobility trajectories of the 1 500 handsets selected from the Device Analyzer dataset. We first parsed the *cid* sequences from the mobility trajectories into mobility networks. In order to remove *cid*s associated with movement, we only defined nodes for *cid*s which were visited by the handset for at least 15 minutes. Movements from one *cid* to another were then recorded as edges in the mobility network.

As outlined in Section 3.1, we analysed the pathways of the Device Analyzer dataset during the entire observation period, applying the model selection method [97] of Scholtes. [4] This method tests graphical models of varying orders and selects the optimal order by balancing the model complexity and the explanatory power of observations.

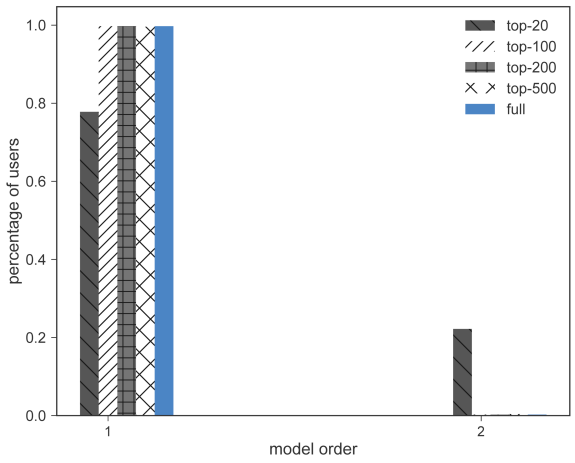
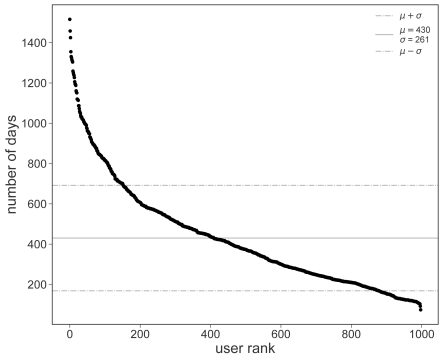


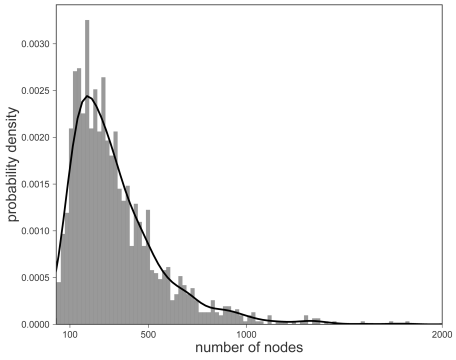
Figure 3.3: Optimal order for increasing number of locations.

We tested higher-order models up to order three. In the case of top−20 mobility networks, we found routine patterns in the mobility trajectories were best explained with models of order two for more than 20% of the users. However, when considering top−100, top-200, top-500 and full mobility networks, we found that the optimal model for our dataset has order one for more than 99% of the users; see Figure 3.3. In other words, when considering

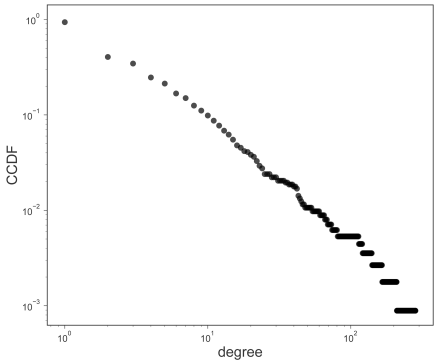
[4] `https://github.com/IngoScholtes/pathpy`



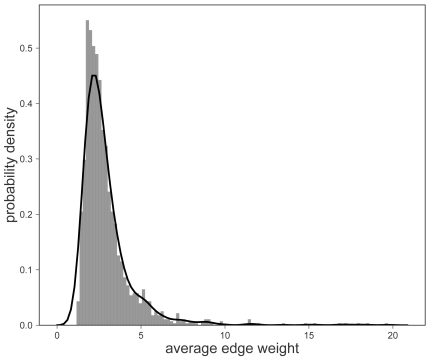
(a) Observation period duration distribution.



(b) Normalized histogram and probability density estimate of network size for the full mobility networks over the population.



(c) Complementary cumulative distribution function (*CCDF*) for the node degree in the mobility network of a typical user from the population, displayed on log-log scale.



(d) Normalized histogram and probabilty density of average edge weight over the networks.

Figure 3.4: Empirical statistical findings of the Device Analyzer dataset.

mobility trajectories which visit less frequent locations in the graph, the overall increase in likelihood of the data for higher-order models cannot compensate for the complexity penalty induced by the larger state space. So while there might still be regions in the graph which are best represented by a higher-order model, the optimal order describing the entire graph is one. Therefore we use a model of order one in the rest of this paper.

3.4.3 Data Properties and Statistics

In Table 1 we provide a statistical summary of the original and the pruned versions of the mobility networks. We observe that allowing more locations in the network implies an increase in the variance of their statistics and leads to smaller density, larger diameter and larger average shortest-path values.

A *recurrent edge traversal* in a mobility network occurs when a previously traversed edge is traversed for a second or subsequent time. We then define *recurrence rate* as the percentage of edge traversals which are recurrent. We find that mobility networks display a high recurrence rate, varying from 68.8% on average for full networks to 84.7% for the top-50 networks, indicating that the mobility of the users is mostly comprised of repetitive transitions between a small set of nodes in a mobility network.

Figure 3.4b displays the normalized histogram and probability density estimate of network size for full mobility networks. We observe that sizes of few hundred nodes are most likely in our dataset, however mobility networks of more than 1 000 nodes also exist. Reducing the variance in network size will be proved helpful in cross-network similarity metrics, hence we also consider truncated versions of the networks.

As shown in Figure 3.4c, the parsed mobility network of a typical user is characterized by a *heavy-tailed degree distribution*. We observe that a small number of locations have high degree and correspond to dominant states for a person's mobility routine, while a large number of locations are only visited a few times throughout the entire observation period and have a small degree.

Figure 3.4d shows that the estimated probability distribution of average edge weight. This peaks in the range from two to four, indicating that many transitions captured in the full mobility network are rarely repeated. However, most of the total weight of the network is attributed to the tail of this distribution, which corresponds to the edges that the user frequently repeats.

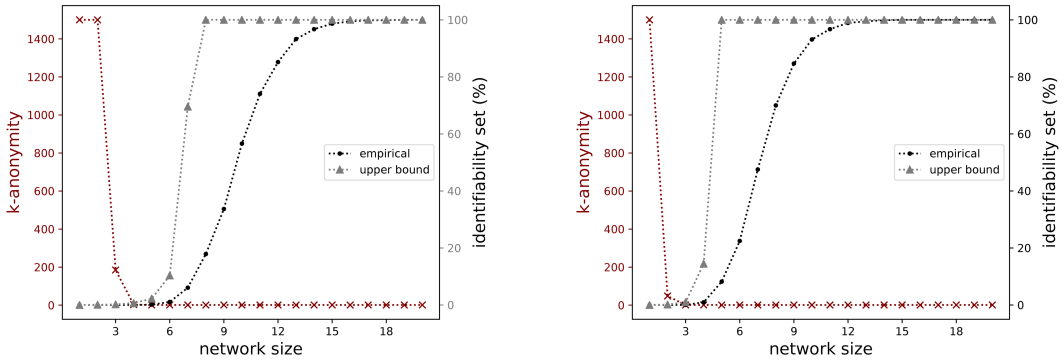
3.4.4 Anonymity Clusters on Top$-N$ Networks

We examine to what extent the heterogeneity of users mobility behaviour can be expressed in the topology of the state connectivity networks. For this purpose, we generate the isomorphism classes of the top$-N$ networks of our dataset for increasing network size N. We then compute the graph $k-$anonymity of the population and the corresponding identifiability set. This analysis demonstrates empirically the privacy implications of releasing anonymized users pathway information at increasing levels of granularity.

Before presenting our findings on the Device Analyzer dataset, we will perform a theoretical upper bound analysis on the identifiability of a population, by finding the maximum number of people that can be distinguished by networks of size N. This corresponds to the number of non-isomorphic graphs with N nodes.

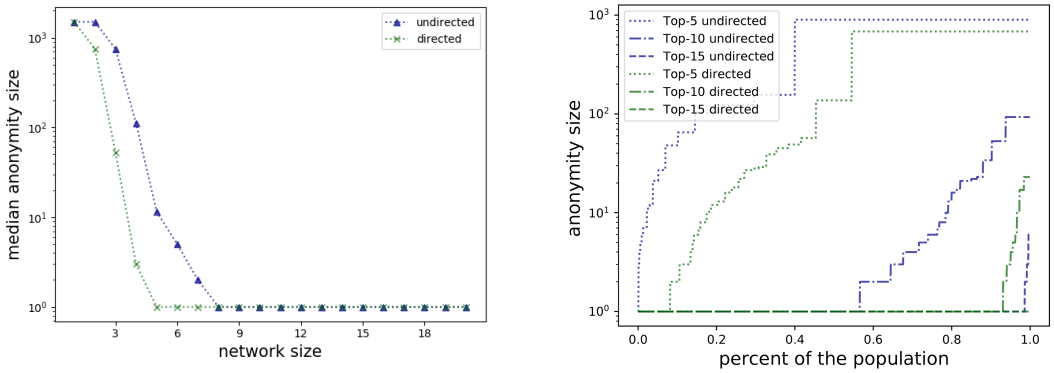
Currently the most of efficient way of enumerating non-isomorphic graphs is by using McKay's algorithm [76], implemented in the package `nauty`.[5] Table 2 presents the enumeration for undirected and directed non-isomorphic graphs of increasing size. We observe that there exist 12 346 undirected graphs with 8 nodes and 9 608 directed graphs with 5 nodes. In other

[5] `http://pallini.di.uniroma1.it/`



(a) Undirected top$-N$ networks. (b) Directed top$-N$ networks.

Figure 3.5: Identifiability set and $k-$anonymity for undirected and directed top$-N$ mobility networks for increasing number of nodes. Displayed is also the theoretical upper bound of identifiability for networks with N nodes.



(a) Median anonymity size. (b) Cumulative distribution of the anonymity size.

Figure 3.6: Anonymity size statistics over the population of top$-N$ mobility networks for increasing network size.

N	4	5	6	7	8	9
# undirected	11	34	156	1044	12346	274668
N	4	5		6		7
# directed	2128	9608		1540944		882033440

Table 3.2: Sequences of non-isomorphic graphs for undirected and directed graphs of increasing size.

words, finding the top-8 places for each person is the smallest number which could produce unique graphs for each person in our sample of 1 500 individuals; this reduces to 5 when directionality is taken into account. Moreover, we find that top-12 undirected and top-8

directed networks are sufficient to enable each human on the planet to be represented by a different graph, assuming world population of 7.6B.

Next we present the results of our analysis on the Device Analyzer data. As observed in Figure 3.5, *sparsity arises in a mobility network even for very small* N. In particular, in the space of undirected $top{-}4$ location networks, there is already a cluster with only 3 members, while for all $N > 4$ there always exist isolated isomorphic clusters. $k{-}anonymity$ decreases to 1 even for $N = 3$ when considering directionality. Moreover, the *identifiability set* dramatically increases with the size of network: approximately 60% of the users are uniquely identifiable from their top$-$10 location network. This percentage increases to 93% in directed networks. For the entire population of the 1 500 users, we find that 15 and 19 locations suffice to form uniquely identifiable directed and undirected networks respectively.

The difference between our empirical findings and our theoretical analysis suggests that large parts of the top$-N$ networks are common to many people. This can be attributed to patterns that are widely shared (e.g. the trip from work to home, and from home to work).

Figure 3.6 shows some additional statistics of the anonymous isomorphic clusters formed for varying network sizes. Median anonymity becomes one for network sizes of five and eight in directed and undirected networks respectively; see Figure 3.6a. In Figure 3.6b we observe that the population arranges into clusters with small anonymity even for very small network sizes: around 5% of the users have at most 10-anonymity when considering only five locations in their network, while this percentage increases to 80% and 100% for networks with 10 and 15 locations. This result confirms that anonymity is even harder when the directionality of edges are provided, since the space of directed networks is much larger than the space of the undirected networks with the same number of nodes.

The above empirical results indicate that the diversity of individuals mobility is reflected in the network representations we use, thus we can meaningfully proceed to discriminative tasks on the population of mobility networks.

3.5 Evaluation of Privacy Loss in Longitudinal Mobility Traces

In this section we empirically quantify the privacy leakage implied by the information of longitudinal mobility networks for the population of users in the Device Analyzer dataset. For this purpose we undertake experiments in graph matching using different kernel functions, and assume an adversary has access to a variety of mobility network information.

3.5.1 Experimental Setup

For our experiments we split the *cid* sequences of each user into two sets: the *training* sequences where user identities are disclosed to the adversary, and the *test* sequences where user identities are undisclosed to the adversary but are used to quantify the success of the

adversarial attack. Therefore each user has two mobility networks: one derived from the training sequences, and one derived from the test sequences. The objective of the adversary is to successfully match every test mobility network with the training mobility network representing the same underlying user. To do so, the adversary computes the pairwise distances between training mobility networks and test mobility networks. We partitioned *cid* sequences of each user by time, placing all *cid*s before the partition point in the training set, and all *cid*s after into the test set. We choose the partition point separately for each user as a random number from the uniform distribution with range 0.3 to 0.7.

3.5.2 Mobility Networks & Kernels

We computed the pairwise distances between training and test mobility networks using kernels from the categories described in Section 3.3. Node attributes are supported in the computation of Weisfeiler-Lehman and Shortest-Path kernel. Thus we augmented the individual mobility networks with categorical features to add some information about the different roles of nodes in users mobility routine. Such attributes are computed independently for each user on the basis of the topological information of each network. After experimenting with several schemes, we obtained the best performance on the kernels when dividing locations into three categories with respect to the frequency in which each node is visited by the user. Concretely, we computed the distribution of users' visits to locations and added the following values to the nodes:

$$a_{c=3}\left(v_i^u\right) = \begin{cases} 3, & \text{if} v_i^u \in \text{top-20\% locations of } u \\ 2, & \text{if} v_i^u \notin \text{top-20\%locations of } u \text{and } v_i^u \in \text{top} - 80\% \text{locations} \\ 1, & \text{otherwise.} \end{cases}$$

This scheme allowed a coarse, yet informative, characterisation of locations in users networks, which was robust to the variance in the frequency of visits between the two observation periods. In addition, we removed 40% of edges with the smallest edge weights and retained only the largest connected component for each user.

Due to its linear complexity, computation of the Weisfeiler-Lehman kernel could scale over entire mobility networks. However, we had to reduce the network size in order to apply the Shortest-Path kernel. This was done using top$-N$ networks for varying size N.

3.5.3 Evaluation & Discussion

We evaluated graph kernels functions from the following categories:

- DSP_N: Deep Shortest-Path kernel on top$-N$ network
- DWL_N: Deep Weisfeiler-Lehman kernel on top$-N$ network

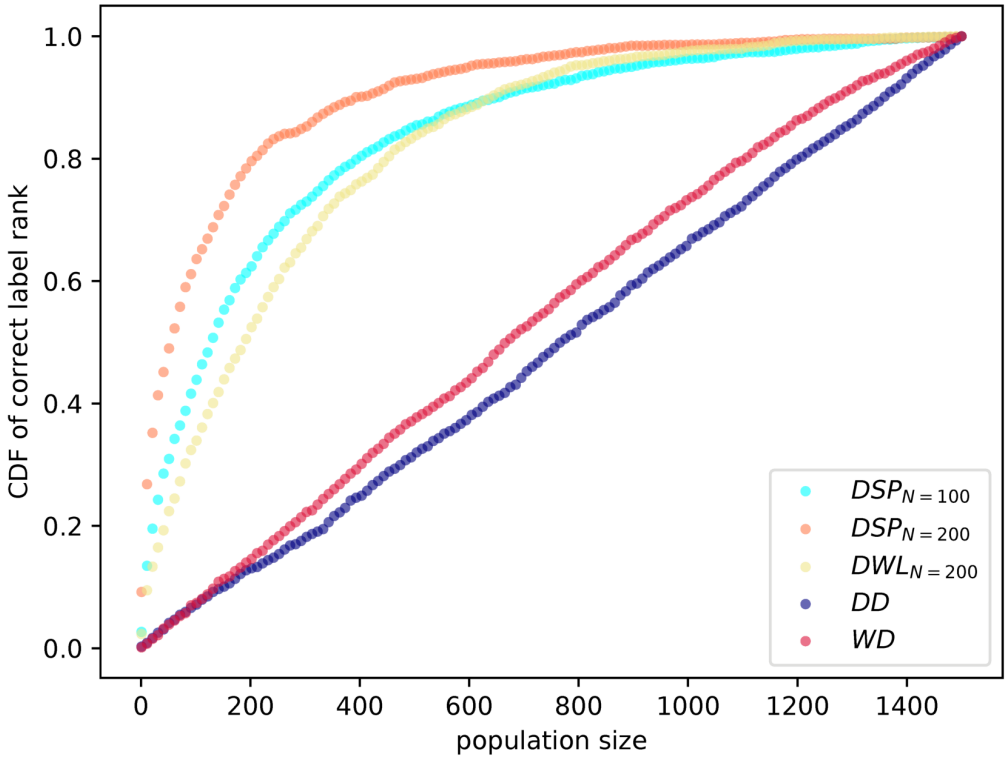


Figure 3.7: *CDF* of true rank over the population according to different kernels.

- *DD*: Degree Distribution kernel through Gaussian RBF
- *WD*: Weighted Degree distribution through Gaussian RBF

The Cumulative Density Functions (*CDF*s) of the true label rank for the best performing kernel of each category are presented in Figure 3.7.

If mobility networks are unique, an *ideal retrieval mechanism* would correspond to a curve that reaches 1 at rank one, indicating a system able to correctly deanonymize all traces by matching the closest training graph. This would be the case when users training and test networks are identical, thus the knowledge of the latter implies maximum privacy loss.

Our baseline, *random*, is a strategy which reflects the policy of an adversary with *zero knowledge* about the mobility networks of the users, who simply returns uniformly random orderings of the labels. The *CDF* of true labels rank for *random* lies on the diagonal line. We observe that atomic structure based kernels significantly outperform the random baseline performance by defining a meaningful similarity ranking across the mobility networks.

The best overall performance is achieved by the *DSP* kernel on graphs pruned to 200 nodes. In particular, this kernel places the true identity among the closest 10 networks for

10% of the individuals, and among the closest 200 networks for 80% of the population. The Shortest-Path kernel has an intuitive interpretation in the case of mobility networks, since its atomic substructures take into account the hop distances among the locations in a user's mobility network and the popularity categories of the departing and arrival location. The deep variant can also account for variation in the level of such substructures, which are more realistic when considering the stochasticity in the mobility patterns inherent to our dataset.

The best performance of the Weisfeiler-Lehman kernel is achieved by its deep variant for $h = 2$ iterations of the *WL* test for a mobility network pruned to 200 nodes. This phenomenon is explainable via the statistical properties of the mobility networks. As we saw in Section 3.4.3, the networks display power law degree distribution and small diameters. Taking into account the steps of the *WL* test, it is clear that these topological properties will lead the node relabeling scheme to cover the entire network after a very small number of iterations. Thus local structural patterns will be described by few features produced in the first iterations of the test. Furthermore, the feature space of the kernel increases very quickly as a function of h, which leads to sparsity and low levels of similarity over the population of networks.

Histograms of length 1 000 were also computed for the unweighted and weighted degree distributions and passed through a Gaussian RBF kernel. We can see that the degree distribution gives almost a random ranking, as it is heavily dependent on the network size. When including the normalized edge weights, the *WD* kernel only barely outperforms a random ranking. Repetitions on pruned versions did not improve the performance and are not presented for brevity.

Based on the insights obtained from our experiment, we can make the following observations with respect to attributes of individual mobility and their impact on the identifiability of networks:

- **Location pruning**: Reducing the number of nodes (locations) in a mobility network does not necessarily make it more privacy-preserving. On the contrary, if location pruning is done by keeping the most frequently visited locations, it can enhance reidentification. In our experiments we obtain similar, or even enhanced, performance for graph kernels when applying them on increasingly pruned networks with size down to 100 locations.

- **Transition pruning**: Including very rare transitions in longitudinal mobility does not add discriminative information. We consistently obtained better results when truncating the long tail of edge weight distribution, which led us to analyze versions of the networks where 40% of the weakest edges were removed.

- **Frequency information of locations**: The frequency of visits to nodes in the mobility network allows better ranking by kernels which support node attributes, e.g.

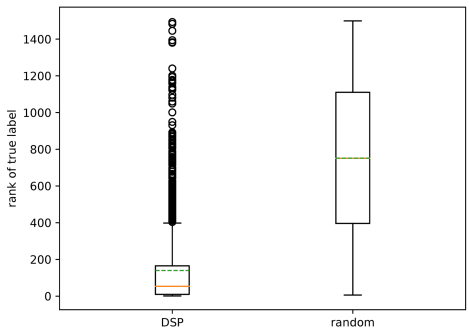


Figure 3.8: Boxplot of rank for the true labels of the population according to a Deep Shortest-Path kernel and to a random ordering.

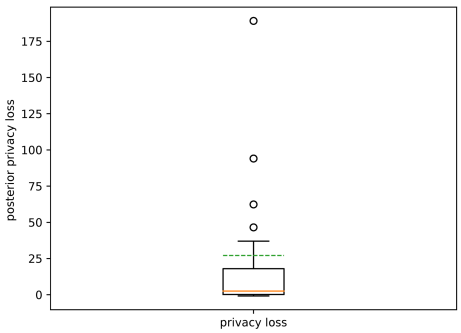


Figure 3.9: Privacy loss over the test data of our population for an adversary adopting the informed policy of (3.10). Median privacy loss is 2.52.

Weisfeiler-Lehman and Shortest-Path kernel. This information should follow a coarse scheme, in order to compensate for the temporal variation of location popularity in mobility networks.

- **Directionality of transitions**: Directionality generally enhances the identifiability of networks and guides the similarity computation when using Shortest-Path kernels.

3.5.4 Quantification of Privay Loss

The Deep Shortest-Path kernel on top-200 networks offers the best ranking of identities for the test networks. As observed in Figure 3.8, the mean of the true rank has been shifted from 750 to 140 for our population. In addition, the variance is much smaller: approximately 218, instead of 423 for the random ordering.

The obtained ordering implies a significant decrease in user privacy, since the ranking can be leveraged by an adversary to determine the most likely matches between a training mobility network and a test mobility network. The adversary can estimate the true identity of a given test network G', as suggested in Section 3.3.4, applying some simple probabilistic policy that uses pairwise similarity information. For example, let us examine the privacy loss implied by update rule in (3.8) for function f:

$$f\big(K_{\mathrm{DSP}}(G_i, G')\big) = \frac{1}{rank\big(K_{DSP}(G_i, G')\big)}. \tag{3.10}$$

This means that the adversary updates her probability estimate for the identity corresponding to a test network, by assigning to each possible identity a probability that is inversely proportional to the rank of the similarity between the test network and the training network corresponding to the identity.

From equation (3.9), we can compute the induced privacy loss for each test network, and the statistics of privacy loss over the networks of the Device Analyzer population. Figure 3.9 demonstrates considerable privacy loss with a median of 2.52. This means that the informed adversary can achieve a median deanonymization probability 3.52 times higher than an uninformed adversary. Moreover, the positive mean of privacy loss (≈ 27) means that the probabilities of the true identities of the test networks have, on average, much higher values in the adversarial estimate compared to the uninformed random strategy. Hence, revealing the kernel values makes an adversarial attack easier.

3.5.5 Defense Mechanisms

The demonstrated privacy leakage motivates the quest for defense mechanisms against this category of attacks. There are a variety of techniques which we could apply in order to *reduce the recurring patterns of an individual's mobility network over time* and *decrease the diversity of mobility networks across a population*, and therefore enhance the privacy inherent in these graphs. Examples include noise injection on network structure via several strategies: randomization of node attributes, perturbations of network edges or node removal. It is currently unclear how effective such techniques will be, and what trade-off can be achieved between utility in mobility networks and the privacy guarantees offered to individuals whose data the graphs represent. Moreover, it seems appropriate to devise kernel-agnostic techniques, suitable for generic defense mechanisms. For example, it is of interest to assess the resistance of our best similarity metric to noise, as the main purpose of deep graph kernels is to be robust to small dissimilarities at the substructure level.

We think this study is important for one further reason: kernel-based methods allow us to apply a rich toolbox of learning algorithms without accessing the original datapoints, or their feature vectors, but instead by using their kernel matrix. Thus studying the anonymity associated with kernels is valuable for ensuring that such learning systems do not leak privacy of the original data. We leave this direction to future work.

3.6 Conclusions & Future Work

In this paper we have shown that the mobility networks of individuals exhibit significant diversity and the topology of the mobility network itself, without labels, may be unique and therefore uniquely identifying.

An individual's mobility network is dynamic over time. Therefore, an adversary with access to mobility data of a person from one time period cannot simply test for graph isomorphism to find the same user in a dataset recorded at a later point in time. Hence we proposed graph kernel methods to detect structural similarities between two mobility networks, and thus provide the adversary with information on the likelihood that two mobility

networks represent the same individual. While graph kernel methods are imperfect predictors, they perform significantly better than a random strategy and therefore our approach induces significant privacy loss. Our approach does not make use of geographic information or fine-grained temporal information and therefore it is immune to commonly adopted privacy-preserving techniques of geographic information removal and temporal cloaking, and thus our method may lead to new mobility deanonymization attacks.

Moreover, we find that reducing the number of nodes (locations) or edges (transistions between locations) in a mobility network does not necessarily make the network more privacy-preserving. Conversely, the frequency of node visits and the direction of transition in a mobility network does enhance the identifiablility of a mobility network for some graph kernel methods. We provide empirical evidence that neighborhood relations in the high-dimensional spaces generated by deep graph kernels remain meaningful for our networks [11]. Further work is needed to shed more light on the geometry of those spaces in order to derive the optimal substructures and dimensionality required to support best graph matching. More work is also required to understand the sensitivity of our approach to the time period over which mobility networks are constructed. There is also an opportunity to explore better ways of exploiting pairwise distance information.

Apart from emphasizing the vulnerability of popular anonymization techniques based on user-specific location pseudonymization, our work provides insights into network features that can facilitate the identifiability of location traces. Our framework also opens the door to new anonymization techniques that can apply structural similarity methods to individual traces in order to cluster people with similar mobility behaviour. This approach may then support statistically faithful population mobility studies on mobility networks with $k-$anonymity guarantees to participants.

Apart from graph kernel similarity metrics, tools for network deanonymization can also be sought in the direction of graph mining: applying heavy subgraph mining techniques [13] or searching for persistent cashcades [79]. Frequent substructure pattern mining (gSpan [126]) and discriminative frequent subgraph mining (CORK [108]) techniques can also be considered.

Our methodology is, in principle, applicable to all types of data where individuals transition between a set of discrete states. Therefore, one of our immediate goals is to evaluate the performance of such retrieval strategies on different categories of datasets, such as web browsing histories or smartphone application usage sequences.

A drawback of our current approach is that it cannot be directly used to mimic individual or group mobility by synthesizing traces. Fitting a generative model on mobility traces and then defining a kernel on this model may provide better anonymity, and therefore privacy, and it would also support the generation of artificial traces which mimic the mobility of users.

Chapter 4

Bayesian Pseudocoresets

Standard Bayesian inference algorithms are prohibitively expensive in the regime of modern large-scale data. Recent work has found that a small, weighted subset of data (a *coreset*) may be used in place of the full dataset during inference, taking advantage of data redundancy to reduce computational cost. However, this approach has limitations in the increasingly common setting of sensitive, high-dimensional data. Indeed, we prove that there are situations in which the Kullback-Leibler (KL) divergence between the *optimal* coreset and the true posterior grows with data dimension; and as coresets include a subset of the original data, they cannot be constructed in a manner that preserves individual privacy. We address both of these issues with a single unified solution, *Bayesian pseudocoresets*—a small weighted collection of synthetic "pseudodata"—along with a variational optimization method to select both pseudodata and weights. The use of pseudodata (as opposed to the original datapoints) enables both the summarization of high-dimensional data and the differentially private summarization of sensitive data. Real and synthetic experiments on high-dimensional data demonstrate that Bayesian pseudocoresets achieve significant improvements in posterior approximation error compared to traditional coresets, and that pseudocoresets provide privacy without a significant loss in approximation quality.

4.1 Introduction

Large-scale data—which has become the norm in many scientific and commercial applications of statistical machine learning—creates an inherently difficult setting for the modern data analyst. Exploring such data is difficult because it cannot all be obtained and directly visualized at once; one is typically limited to accessing potentially nonrepresentative random subsets of data. Exploring models is similarly hard, as training even a single model can be a computationally expensive, slow, and unreliable process. And as many sources of large-scale data contain sensitive information about individuals (e.g., electronic health records and social

network data), these challenges are coupled with growing privacy concerns that preclude direct access to individual datapoints completely.

Large-scale data does offer one reprieve to the analyst: it often exhibits a significant degree of redundancy. Most data are not unique or particularly informative for modelling and exploration. Based on this notion, data summarization methods have been developed that provide the practitioner with a compressed—but still statistically representative—version of the large dataset for analysis. Summarizations have been developed for a variety of purposes, e.g., reducing the cost of computing with kernel matrices via Nyström-type approximations [27, 80, 5] or sparse pseudo-input parameterizations [103], Bayesian inference [53, 52, 17, 18, 16], maximum likelihood parameter estimation [30, 74], linear regression [134, 47], geometric shape approximation [2], clustering [35, 71, 7, 15], and dimensionality reduction [37].

A common form of summarization is that of a sparse, weighted subset of the original dataset—a *coreset* [2]. Coresets have two distinct advantages over other possible summarization modalities: they are easily interpreted, and can often be used as the input to standard data analysis algorithms without modification. But as the dimensionality of a dataset grows, its constituent datapoints tend to become more "unique" and cannot represent one another well. Indeed, in the context of Bayesian inference—the focus of the present work—we show that the *optimal* coreset posterior approximation to the true posterior has KL divergence that scales with the dimension of the data in a simple problem setting (Proposition 8). Furthermore, directly releasing a subset of the original data precludes any possibility of individual privacy under the current standard of differential privacy [32, 33]. Past work addresses this issue in the context of clustering and computational geometry [36, 38]—with the remarkable property that the privatized coreset may be queried *ad infinitum* without loss of privacy—but no such method exists for Bayesian posterior inference.

In this work, we develop a novel technique for data summarization in the context of Bayesian inference under the constraints that the method is scalable and easy to use, creates an intuitive summarization, applies to high-dimensional data, and enables privacy control. Inspired by past work [74, 134, 103], instead of using constituent datapoints, we use synthetic *pseudodata* to summarize the large dataset, resulting in a *pseudocoreset*. We show that in the high-dimensional problem setting of Proposition 8, the optimal pseudocoreset with just one pseudodata point recovers the exact posterior, a significant improvement upon the optimal standard coreset of any size. As in past work on Bayesian coresets [16], we formulate pseudocoreset construction as variational inference, and provide a stochastic optimization method. As a consequence of the use of pseudodata—as well as privacy-preserving stochastic gradient descent mechanisms [1, 86, 57]—we show that our method can easily be modified to output a privatized pseudocoreset. The paper concludes with experimental results demonstrating the performance of pseudocoresets on real and synthetic data.

4.2 Bayesian Coresets

In this work, the goal is to approximate expectations under a density $\pi(\theta)$, $\theta \in \Theta$ expressed as the product of N potentials $(f(x_n, \theta))_{n=1}^N$ and a base density $\pi_0(\theta)$:

$$\pi(\theta) := \frac{1}{Z} \exp\left(\sum_{n=1}^{N} f(x_n, \theta)\right) \pi_0(\theta). \tag{4.1}$$

In the setting of Bayesian inference with conditionally independent data, the potentials are data log-likelihoods, i.e. $f(x_n, \theta) := \log p(x_n|\theta)$, π_0 is the prior density, π is the posterior, and Z is the marginal likelihood of the data. Rather than working directly with $\pi(\theta)$ for posterior inference—which requires a $\Theta(N)$ computation per evaluation—a Bayesian coreset approximation of the form

$$\pi_w(\theta) := \frac{1}{Z(w)} \exp\left(\sum_{n=1}^{N} w_n f(x_n, \theta)\right) \pi_0(\theta) \tag{4.2}$$

for $w \in \mathbb{R}^N$, $w \geq 0$ may be used in most popular posterior inference schemes [84, 64, 91]. If the number of nonzero entries $\|w\|_0$ of w is small, this results in a significant reduction in computational burden. Recent work has formulated the problem of constructing a Bayesian coreset of size $M \in \mathbb{N}$ as sparse variational inference [16],

$$w^\star = \underset{w \in \mathbb{R}^N}{\arg\min}\, \mathrm{D}_{\mathrm{KL}}\left(\pi_w || \pi_1\right) \qquad \text{s.t.} \quad w \geq 0,\ \|w\|_0 \leq M, \tag{4.3}$$

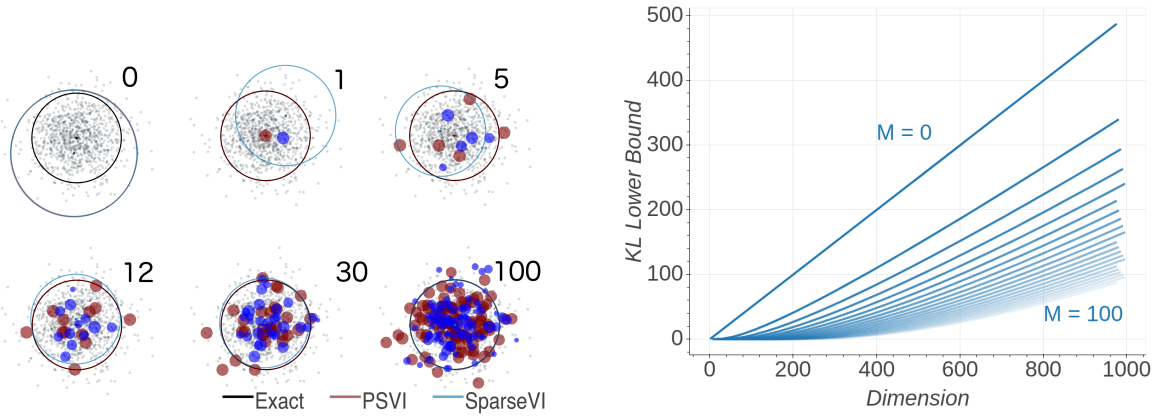
and showed that the objective can be minimized using stochastic estimates of $\nabla_w \mathrm{D}_{\mathrm{KL}}\left(\pi_w||\pi_1\right)$ based on samples from the coreset posterior π_w.

4.2.1 High-dimensional data

Coresets, as formulated in Eq. (4.3), are limited to using the original datapoints themselves to summarize the whole dataset. Proposition 8 shows that this is problematic when summarizing high-dimensional data; in the common setting of posterior inference for a Gaussian mean, the KL divergence $\mathrm{D}_{\mathrm{KL}}\left(\pi_{w^\star}||\pi_1\right)$ of the *optimal* coreset of any size scales with the dimension of the data. The proof may be found in Section 4.6.

Proposition 8. *Suppose we use* $(X_n)_{n=1}^N \overset{i.i.d.}{\sim} \mathcal{N}(0, I)$ *in* $\mathbb{R}^d$ *to perform posterior inference in a Bayesian model with prior* $\mu \sim \mathcal{N}(0, I)$ *and likelihood* $(X_n)_{n=1}^N \overset{i.i.d.}{\sim} \mathcal{N}(\mu, I)$*. Then* $\forall M < d$ *and* $\delta \in [0, 1]$*, with probability at least* $1 - \delta$ *the optimal size-*M *coreset* $w^\star$ *satisfies*

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{w^\star}||\pi_1\right) \geq \frac{1}{2} \frac{N - M}{1 + N} F_{d-M}^{-1}\left(\delta \binom{N}{M}^{-1}\right), \tag{4.4}$$



(a) (b)

Figure 4.1: Gaussian mean inference under pseudocoreset (PSVI) against standard coreset (SparseVI) summarization for $N = 1,000$ datapoints. (a) Progression of PSVI vs. SparseVI construction for coreset sizes $M = 0, 1, 5, 12, 30, 100$, in 500 dimensions (displayed are datapoint projections on 2 random dimensions). PSVI and SparseVI coreset predictive 3σ ellipses are displayed in red and blue respectively, while the true posterior 3σ ellipse is shown in black. PSVI has the ability to immediately move pseudopoints towards the true posterior mean, while SparseVI has to add a larger number of existing points in order to obtain a good posterior approximation. See Fig. 4.2 for the quantitative KL comparison. (b) Optimal coreset KL divergence lower bound from Proposition 8 as a function of dimension with $\delta = 0.5$, and coreset size M evenly spaced from 0 to 100 in increments of 5.

where F_k is the CDF of a χ^2 random variable with k degrees of freedom.

The bound in Proposition 8 depends on d through the χ^2 distribution inverse CDF. Although difficult to see directly, the bound is reasonably large for typical values of N, M, d, δ, and increasing linearly in d; Fig. 4.1b visualizes the value of the lower bound as a function of dimension d for various coreset sizes M. Note that the above bound requires the data to be high-dimensional such that $d > M$; if $d \leq M$ the proof technique in Section 4.6 results in a vacuous $\mathrm{D}_{\mathrm{KL}}\left(\pi_{w^\star} || \pi_1\right) = 0$ lower bound.

4.3 Bayesian Pseudocoresets

Proposition 8 shows that there is room for improvement in coreset construction in the high-dimensional data regime. Indeed, consider again the same problem setting; the coreset posterior distribution is a Gaussian with mean μ_w and covariance Σ_w,

$$\Sigma_w = \left(1 + \sum_{n=1}^{N} w_n\right)^{-1} I \qquad \mu_w = \Sigma_w \sum_{n=1}^{N} w_n X_n. \tag{4.5}$$

Examining Eq. (4.5), we can replicate any coreset posterior exactly by using a single synthetic *pseudodata* point $U = \left(\sum_{n=1}^N w_n\right)^{-1} \sum_{n=1}^N w_n X_n$ with weight $\sum_{n=1}^N w_n$. In particular, the true posterior is equivalent to the posterior conditioned on the single pseudodata point $U = \frac{1}{N}\sum_{n=1}^N X_n$ with weight N (with corresponding KL divergence equal to 0). This is not surprising; the mean of the data is precisely a sufficient statistic for the data in this simple setting. However, it does illustrate that carefully-chosen pseudodata may be able to represent the overall dataset—as "approximate sufficient statistics"—far better than any reasonably small collection of the original data. This intuition has been used before, e.g., for scalable Gaussian process inference [103, 109], privacy-preserving compression in linear regression [134], herding [121, 19, 55], and deep generative models [110].

In this section, we extend the realm of applicability of pseudopoint compression methods to the general class of Bayesian posterior inference problems with conditionally independent data, resulting in *Bayesian pseudocoresets*. Building on recent work [16], we formulate pseudocoreset construction as a variational inference problem where both the weights and pseudopoint locations are parameters of the variational posterior approximation, and develop a stochastic algorithm to solve the optimization.

4.3.1 Pseudocoreset variational inference

A Bayesian pseudocoreset takes the form

$$\pi_{u,w}(\theta) = \frac{1}{Z(u,w)} \exp\left(\sum_{m=1}^M w_m f(u_m, \theta)\right) \pi_0(\theta), \tag{4.6}$$

where $u := (u_m)_{m=1}^M$ are M pseudodata points $u_m \in \mathbb{R}^d$, $(w_m)_{m=1}^M$ are nonnegative weights, $f : \mathbb{R}^d \times \Theta \to \mathbb{R}$ is a potential function parametrized by a pseudodata point, and $Z(u, w)$ is the corresponding normalization constant rendering $\pi_{u,w}$ a probability density. In the setting of Bayesian posterior inference, u_m will take the same form as the data, while the potentials are the log-likelihood functions, i.e., $f(u_m, \theta) = \log p(u_m|\theta)$. We construct a coreset by minimizing the KL divergence over both the pseudodata locations and weights,

$$u^\star, w^\star = \underset{u \in \mathbb{R}^{d\times M}, w \in \mathbb{R}_+^M}{\arg\min} \mathrm{D}_{\mathrm{KL}}\left(\pi_{u,w} || \pi\right). \tag{4.7}$$

As opposed to previous Bayesian coreset construction optimization problems [17, 18, 16], we do not need an explicit sparsity constraint; the coreset size is limited to M directly through the selection of the number of pseudodata and weights.

Denote the vectors of original data potentials $f(\theta) \in \mathbb{R}^N$ and synthetic pseudodata potentials $\tilde{f}(\theta) \in \mathbb{R}^M$ as $f(\theta) := [f_1(\theta) \dots f_N(\theta)]^T$ and $\tilde{f}(\theta) := [f(u_1, \theta) \dots f(u_M, \theta)]^T$ respectively, where we suppress the (θ) for brevity where clear from context. Denote $\mathbb{E}_{u,w}$ and $\mathrm{Cov}_{u,w}$ to be the expectation and covariance operator for the pseudocoreset posterior $\pi_{u,w}$. Then we

may write the KL divergence in Eq. (4.7) as

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}\left(\pi_{u,w}||\pi\right) =& \mathbb{E}_{u,w}[\log \pi_{u,w}(\theta)] - \mathbb{E}_{u,w}[\log \pi(\theta)] && (4.8)\\ =& \log Z(1) - \log Z(u,w) - 1^T \mathbb{E}_{u,w}[f] + w^T \mathbb{E}_{u,w}[\tilde{f}], && (4.9) \end{aligned}$$

where $1 \in \mathbb{R}^N$ is the vector of all 1 entries, and $w \in \mathbb{R}^M$ is the vector of pseudocoreset weights.

As we will employ gradient descent steps as part of our algorithm to minimize the variational objective over the parameters u, w, we need to evaluate the derivative of the KL divergence Eq. (4.9). Despite the presence of the intractable normalization constants and expectations, we show in Section 4.7 that gradients can be expressed using moments of the pseudodata and original data potential vectors. In particular, the gradients of the KL divergence with respect to the weights w and to a single pseudodata location u_m are

$$\nabla_w \mathrm{D}_{\mathrm{KL}} = -\operatorname{Cov}_{u,w}[\tilde{f}, f^T 1 - \tilde{f}^T w], \qquad \nabla_{u_m} \mathrm{D}_{\mathrm{KL}} = -w_m \operatorname{Cov}_{u,w}\left[h(u_m), f^T 1 - \tilde{f}^T w\right], \tag{4.10}$$

where $h(\cdot, \theta) := \nabla_u f(\cdot, \theta)$, and the θ argument is again suppressed for brevity.

4.3.2 Stochastic optimization

The gradients in Eq. (4.10) involve expectations of (gradient) log-likelihoods from the model. Although there are a few particular Bayesian models where these can be evaluated in closed-form (e.g. the synthetic experiment in Section 4.4; see also Section 4.8.1), this is not usually the case. In order to make the proposed pseudocoreset method broadly applicable, in this section we develop a black-box stochastic optimization scheme (Algorithm 1) for Eq. (4.7).

To initialize the pseudocoreset, we subsample M datapoints from the large dataset and reweight them to match the overall weight of the full dataset,

$$u_m \leftarrow x_{b_m}, \quad w_m \leftarrow N/M, \quad m = 1, \ldots, M \tag{4.11}$$

$$\mathcal{B} \sim \mathsf{UnifSubset}\left([N], M\right), \quad \mathcal{B} := \{b_1, \ldots, b_M\}. \tag{4.12}$$

After initializing the pseudodata locations and weights, we simultaneously optimize Eq. (4.7) over both. Each optimization iteration $t \in \{1, \ldots, T\}$ consists of a stochastic gradient descent step with a learning rate $\gamma_t \propto t^{-1}$,

$$w_m \leftarrow \max\left(0, w_m - \gamma_t (\hat{\nabla}_w)_m\right), \quad u_m \leftarrow u_m - \gamma_t \hat{\nabla}_{u_m}, \quad 1 \le m \le M. \tag{4.13}$$

The stochastic gradient estimates $\hat{\nabla}_w \in \mathbb{R}^M$ and $\hat{\nabla}_{u_m} \in \mathbb{R}^d$ are based on $S \in \mathbb{N}$ samples $\theta_s \overset{\text{i.i.d.}}{\sim} \pi_{u,w}$ from the coreset approximation and a minibatch of $B \in \mathbb{N}$ datapoints from the

Algorithm 1 Pseudocoreset Variational Inference

procedure PSVI($f(\cdot,\cdot), \pi_0, x, M, B, S, T, (\gamma_t)_{t=1}^{\infty}$)
  ▷ Initialize the pseudocoreset using a uniformly chosen subset of the full dataset
  $N \leftarrow$ # datapoints in x, $\mathcal{B} \sim \mathsf{UnifSubset}([N], M)$, $\mathcal{B} := \{b_1, \ldots, b_M\}$
  $u_m \leftarrow x_{b_m}$, $w_m \leftarrow {}^{N}/_{M}$, $m = 1, \ldots, M$
  for $t = 1, \ldots, T$ **do**
    ▷ Take S samples from current pseudocoreset posterior
    $(\theta)_{s=1}^{S} \overset{\text{i.i.d.}}{\sim} \pi_{u,w}$ where $\pi_{u,w}(\theta) \propto \exp\left(\sum_{m=1}^{M} w_m f(u_m, \theta)\right) \pi_0(\theta)$
    $\mathcal{B} \sim \mathsf{UnifSubset}([N], B)$ ▷ Obtain a minibatch of B datapoints from the full dataset
    for $s = 1, \ldots, S$ **do** ▷ Compute (gradient) log-likelihood discretizations
      $g_s \leftarrow \left(f(x_b, \theta_s) - {}^{1}/_{S} \sum_{s'=1}^{S} f(x_b, \theta_{s'})\right)_{b \in \mathcal{B}} \in \mathbb{R}^B$
      $\tilde{g}_s \leftarrow \left(f(u_m, \theta_s) - {}^{1}/_{S} \sum_{s'=1}^{S} f(u_m, \theta_{s'})\right)_{m=1}^{M} \in \mathbb{R}^M$
      for $m = 1, \ldots, M$ **do**
        $\tilde{h}_{m,s} \leftarrow \nabla_u f(u_m, \theta_s) - {}^{1}/_{S} \sum_{s'=1}^{S} \nabla_u f(u_m, \theta_{s'}) \in \mathbb{R}^d$
    $\hat{\nabla}_w \leftarrow -{}^{1}/_{S} \sum_{s=1}^{S} \tilde{g}_s \left({}^{N}/_{B} g_s^T 1 - \tilde{g}_s^T w\right)$ ▷ Compute Monte-Carlo gradients for w and $(u_m)_{m=1}^{M}$
    for $m = 1, \ldots, M$ **do**
      $\hat{\nabla}_{u_m} \leftarrow -w_m {}^{1}/_{S} \sum_{s=1}^{S} \tilde{h}_{m,s} \left({}^{N}/_{B} g_s^T 1 - \tilde{g}_s^T w\right)$
    $w \leftarrow \max(w - \gamma_t \hat{\nabla}_w, 0)$ ▷ Take stochastic gradient step in w and $(u_m)_{m=1}^{M}$
    for $m = 1, \ldots, M$ **do**
      $u_m \leftarrow u_m - \gamma_t \hat{\nabla}_{u_m}$
  return w, $(u_m)_{m=1}^{M}$

full dataset,

$$\hat{\nabla}_w := -\frac{1}{S} \sum_{s=1}^{S} \tilde{g}_s \left(\frac{N}{B} g_s^T 1 - \tilde{g}_s^T w\right), \tag{4.14}$$

$$\hat{\nabla}_{u_m} := -w_m \frac{1}{S} \sum_{s=1}^{S} \tilde{h}_{m,s} \left(\frac{N}{B} g_s^T 1 - \tilde{g}_s^T w\right), \tag{4.15}$$

where

$$\begin{aligned} \tilde{h}_{m,s} &:= \nabla_u f(u_m, \theta_s) - \frac{1}{S} \sum_{s'=1}^{S} \nabla_u f(u_m, \theta_{s'}), \qquad g_s := \left(f(\theta_s) - \frac{1}{S} \sum_{s'=1}^{S} f(\theta_{s'})\right)\Big|_{\mathcal{B}} \\ \tilde{g}_s &:= \tilde{f}(\theta_s) - \frac{1}{S} \sum_{s'=1}^{S} \tilde{f}(\theta_{s'}), \qquad\qquad \mathcal{B} \sim \mathsf{UnifSubset}([N], B), \end{aligned} \tag{4.16}$$

and $(\cdot)|_{\mathcal{B}}$ denotes restriction of a vector to only those indices in $\mathcal{B} \subset [N]$. Crucially, note that this computation does not scale with N, but rather with the number of coreset points M, the sample and minibatch sizes S and B, and the dimension d. Obtaining $\theta_s \overset{\text{i.i.d.}}{\sim} \pi_{u,w}$ efficiently via Markov chain Monte Carlo sampling algorithms [50, 56] is (roughly) $O(M)$ per sample,

because the coreset is always of size M; and we need not compute the entire vector $g_s \in \mathbb{R}^N$ per sample s, but rather only those $B \ll N$ indices in the minibatch $\mathcal{B}$, resulting in a cost of $O(B)$. Aside from that, all computations involving $\tilde{g}_s \in \mathbb{R}^M$ and $\tilde{h}_{m,s} \in \mathbb{R}^d$ are at most $O(Md)$. Each of these computations are repeated S times over the coreset posterior samples.

4.3.3 Differentially Private Scheme

Beyond better summarizations of high-dimensional data, pseudocoresets enable the generation of a data summarization that ensures the statistical privacy of individual datapoints under the model of (approximate) *differential privacy*. In this setting, a trusted curator holds an aggregate dataset of N datapoints, $x \in \mathcal{X}^N$, $\mathcal{X} \subseteq \mathbb{R}^d$, and builds and releases a pseudocoreset (u, w), $u \in \mathcal{X}^M$, $w \in \mathbb{R}_+^M$ via a randomized mechanism satisfying Definition 9 [**?** 31].

Definition 9 ((ε, δ)-Differentially Private Coreset)**.** Fix $\varepsilon \geq 0, \delta \in [0, 1]$. A pseudocoreset construction algorithm $\mathcal{M} : \mathcal{X}^N \to \mathbb{R}_+^M \times \mathcal{X}^M$ is (ε, δ)-differentially private if for every pair of adjacent datasets $x \approx x'$ and all events $A \subseteq \mathbb{R}_+^M \times \mathcal{X}^M$, $\mathbb{P}[\mathcal{M}(x) \in A] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(x') \in A] + \delta$.

We consider two datasets x, x' as adjacent (denoted $x \approx x'$) if x' can be obtained from x by adding or removing an element. ε controls the effect that removal or addition of an element can have on the output distribution of $\mathcal{M}$, while δ captures the failure probability, and is preferably $o(1/N)$.

In this section, we develop a differentially private version of pseudocoreset construction. Beyond modifying our initialization scheme, private pseudocoreset construction comes as natural extension of Algorithm 1 via replacing gradient computation involving points of the true dataset with its differentially private counterpart.

Pseudodata points initialization In the standard (nonprivate) pseudocoreset construction (Algorithm 1), pseudopoints are initialized from the dataset itself, incurring a privacy penalty. In differentially private pseudocoreset construction, we simply initialize pseudopoints by generating synthetic data from the statistical model at no privacy cost.

Optimization Examining lines 4–19 of Algorithm 1, the only steps that involve handling the original data occur at lines 8, 12, and 14, when we use the minibatch subsample to compute log-likelihoods and gradients. Due to the post-processing property of differential privacy [33], all of the other computations in Algorithm 1 (e.g. sampling from the pseudocoreset posterior, computing pseudopoint log-likelihoods, etc.) incur no privacy cost. Therefore, we need only to control the influence of private data entering the gradient computation through the vector of $(g_s^T 1)_{s=1}^S$ terms.

To accomplish this we do repeated applications of the *subsampled Gaussian mechanism*, since this also allows us to use a *moments accountant* technique to keep tight estimates

of privacy parameters [1, 118]. As in the nonprivate scheme, in each optimization step we uniformly subsample a minibatch $\mathcal{B} = \{x_1, \ldots, x_B\}$ of private datapoints. We then replace the $g_s^T 1$ term in lines 12 and 14 with a randomized privatization:

$$\text{replace} \quad (g_s^T 1)_{s=1}^S \quad \text{with} \quad Z + \sum_{i=1}^{B} \frac{G_i}{\max\left(1, \frac{||G_i||_2}{C}\right)}, \qquad Z \sim \mathcal{N}(0, \sigma^2 C^2 I), \tag{4.17}$$

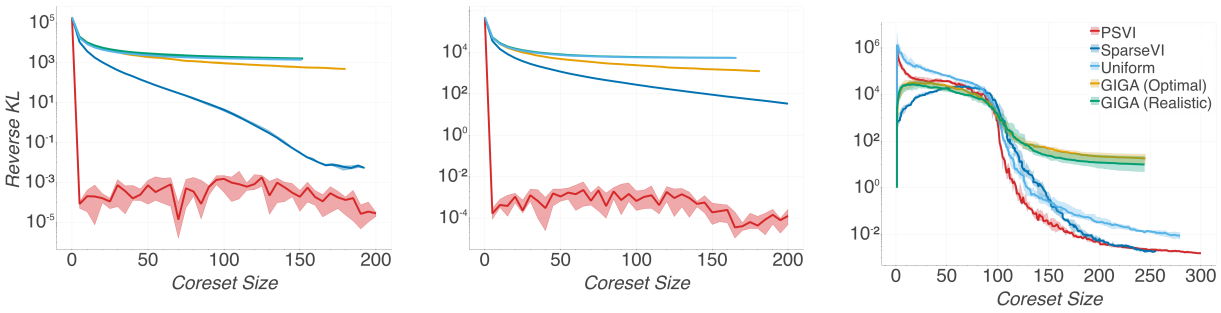
where $G_i := \left(f(x_i, \theta_s) - \frac{1}{S}\sum_{s'=1}^S f(x_i, \theta_{s'})\right)_{s=1}^S \in \mathbb{R}^S \; \forall x_i \in \mathcal{B}$, and $C, \sigma > 0$ are parameters controlling the amount of privacy. This modification to Algorithm 1 has been shown in past work to obtain the privacy guarantee provided in Corollary 10; crucially, the privacy cost of our construction is independent of the pseudocoreset size. It also does not introduce any significant amount of additional computation. No sensitivity computation for privatisation noise calibration is required, as boundedness is enforced via clipping in Eq. (4.17). Finally, a manageable number of privacy specific hyperparameters is introduced: the clipping bound C and noise level σ.

Corollary 10 ([1])**.** *There exist constants c_1, c_2 such that Algorithm 1 modified per Eq.* (4.17) *is (ε, δ)-differentially private for any $\varepsilon < c_1 q^2 T$, $\delta > 0$, and $\sigma \geq c_2 q \sqrt{T \log(1/\delta)}/\varepsilon$, where $q := \frac{B}{N}$ is the fraction of data in a minibatch and T is the number of optimization steps.*

4.4 Experimental Results

In this section, we evaluate the posterior approximation quality achieved by pseudocoreset sparse VI (PSVI) compared against uniform random subsampling (`Uniform`), Hilbert coresets (`GIGA` [17]) and SparseVI greedy coreset construction [16]. For black-box constructions of SparseVI and PSVI we used $S = 100$ Monte Carlo samples per gradient estimation. For `GIGA` we used a 100-dimensional random projection from a Gaussian approximate posterior $\hat{\pi}$ with two choices for mean and covariance: one set to the exact posterior (`Optimal`), which is not tractable to obtain in practice and forms an optimistic estimate of achievable approximation quality; and one with mean and covariance set to a random point on the interpolant between the prior and the exact posterior point estimates, and subsequently corrupted with 75% additive relative noise (`Realistic`). Notably, Hilbert coresets and SparseVI develop incremental schemes for construction, while PSVI relies on batch optimization with random initialization (Algorithm 1), and does not use any information from pseudocoresets of smaller size. An incremental scheme for SparseVI is included in Section 4.8. Code for the presented experiments is available at anonymous_public_repo.

Gaussian mean inference We first evaluate the performance of PSVI on a synthetic dataset of $N = 10^3$ datapoints, where we aim to infer the posterior mean $\theta \sim \mathcal{N}(\mu_0, \Sigma_0)$ of a



(a) Gauss. mean inf., $d = 200$ (b) Gauss. mean inf., $d = 500$ (c) Bayes. lin. regr., $d = 100$

Figure 4.2: Comparison of coreset approximate posterior quality for experiments on synthetic datasets over 10 trials. Solid lines display the median KL divergence, with shaded areas showing 25$^{\text{th}}$ and 75$^{\text{th}}$ percentiles of KL divergence. In Fig. 4.2c, KL divergence is normalized by the prior.

d-dimensional Gaussian conditioned on Gaussian observations $(X_n)_{n=1}^N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\theta, \Sigma)$. In this example, the exact pseudocoreset posterior for any set of weights $(w_m)_{m=1}^M$ and pseudopoint locations $(u_m)_{m=1}^M$ is available in closed-form:

$$\Sigma_{u,w} = (\Sigma_0^{-1} + \sum_{m=1}^{M} w_m \Sigma^{-1})^{-1} \qquad \mu_{u,w} = \Sigma_{u,w}(\Sigma_0^{-1}\mu_0 + \Sigma^{-1} \sum_{m=1}^{M} w_m u_m). \tag{4.18}$$

Using the exact posterior, we derive the exact moments used in the gradient formulae from Eq. (4.10) in closed form (see Section 4.8.1),

$$\begin{aligned} &\mathrm{Cov}_{u,w}[f_n, f_m] = v_n^T \Psi v_m + {}^1\!/_2 \operatorname{tr} \Psi^T \Psi, \quad \mathrm{Cov}_{u,w}[\tilde{f}_n, f_m] = \tilde{v}_n^T \Psi v_m + {}^1\!/_2 \operatorname{tr} \Psi^T \Psi, \\ &\mathrm{Cov}_{u,w}[h(u_i), f_n] = Q^{-T} \Psi v_n, \qquad \mathrm{Cov}_{u,w}[h(u_i), \tilde{f}_n] = Q^{-T} \Psi \tilde{v}_n, \end{aligned} \tag{4.19}$$

where Q is the Cholesky decomposition of Σ (i.e. $\Sigma = QQ^T$), $\Psi := Q^{-1}\Sigma_{u,w}Q^{-T}$, $v_n := Q^{-1}(x_n - \mu_{u,w})$, and $\tilde{v}_m := Q^{-1}(u_m - \mu_{u,w})$. We vary the pseudocoreset size from $M = 1$ to 200, and set the total number of iterations to $T = 500$. We use learning rates $\gamma_t(M) = \alpha(M)t^{-1}$, where $\alpha(M) = 1$ for SparseVI and $\alpha(M) = \max(1.1 - 0.005M, 0.2)$ for PSVI. As verified in Figs. 4.2a and 4.2b, Hilbert coresets provide poor quality summarizations in the high-dimensional regime, even for large coreset sizes. Despite showing faster decrease of approximation error for a larger range of coreset sizes, SparseVI is also fundamentally limited by the use of the original datapoints, per Proposition 8. Furthermore, we observe that the quality of all previous coreset methods when $d = 500$ is significantly lower compared to $d = 200$. On the other hand, the KL divergence for PSVI decreases significantly more quickly, giving a near perfect approximation for true posterior with a single pseudodata point regardless of data dimension. As shown earlier in Fig. 5.1a, PSVI has the capacity to move the pseudodata points in order to capture the true posterior very efficiently.

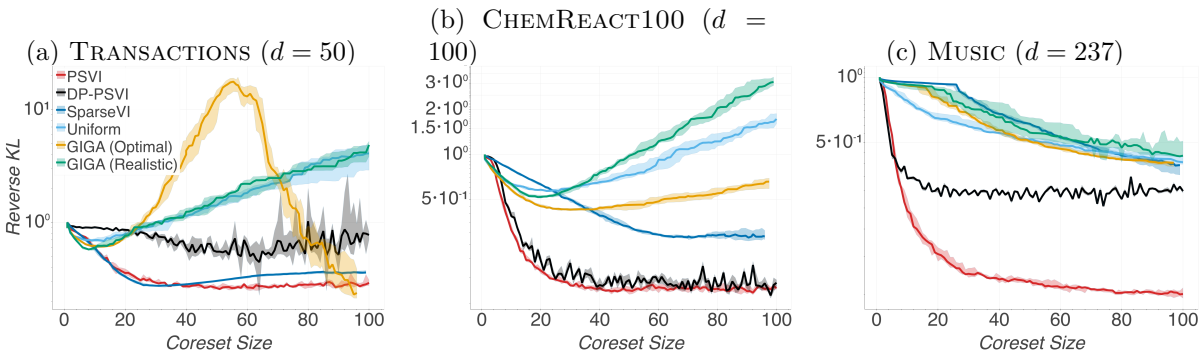


Figure 4.3: Comparison of (pseudo)coreset approximate posterior quality vs coreset size for logistic regression over 10 trials on 3 large-scale datasets. Presented differentially private pseudocoresets correspond to $(0.2, 1/N)$-DP. Reverse KL divergence is displayed normalized by the prior.

Bayesian linear regression In the second experiment, we use a set of $N = 2,000$ 101-dimensional datapoints $(x_n, y_n)_{n=1}^N$ generated as follows: $(x_n)_{n=1}^N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$, $(y_n)_{n=1}^N \sim [1, x_n]^T\theta + \epsilon_n$, $(\epsilon_n)_{n=1}^N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$, and aim to infer $\theta \in \mathbb{R}^{101}$. We assume a prior $\theta \sim \mathcal{N}(\mu_0, \sigma_0^2 I)$, where μ_0, σ_0^2 are the dataset empirical mean and second moment, and set the noise parameter σ to the variance of $(y_n)_{n=1}^N$. We apply stochastic optimization for PSVI construction (also see Section 4.8.2). We use learning rates $\gamma_t = t^{-1}$ for SparseVI, and $\gamma_t = 0.1t^{-1}$ for PSVI, $B = 200$, $T = 1000$, while selection step for SparseVI is carried out over the full dataset. Fig. 4.2c shows that Hilbert coresets cannot improve posterior approximation in this setting with 100 random projections (see Section 4.8.2), while PSVI achieves the fastest decay rate over sizes $100 \leq M \lessapprox 250$, surpassing SparseVI.

Bayesian logistic regression Finally, we compare (pseudo)coreset construction methods on Bayesian logistic regression applied to 3 large (8.4–100K datapoints, 50–237 dimensions) datasets. For brevity, equations and gradients for the logistic regression model, additional experiments on 3 smaller-scale datasets, full dataset descriptions, hyperparameter selection, time performance evaluation and results on an incremental scheme for pseudocoreset construction are deferred to Section 4.8.3. For PSVI and SparseVI we use minibatch size $B = 200$, number of gradient updates $T = 500$, and learning rate schedules $\gamma_t = \alpha t^{-1}$. For TRANSACTIONS, CHEMREACT100 and MUSIC, α is respectively set to 0.1, 0.1, 1 for SparseVI, and 1, 10, 10 for PSVI. In the selection step of SparseVI we use a uniform subsample of 1,000 datapoints. For the differentially private pseudocoreset constructions (`DP-PSVI`), we use a subsampling ratio $q = 2 \times 10^{-3}$. At each iteration we adapt the clipping norm value C to the median norm of $(f(u_m, \theta_s) - \frac{1}{S}\sum_{s'=1}^S f(u_m, \theta_{s'}))_{s=1}^S$ computed over pseudodata point values u_m, and use noise level $\sigma = 5$. We initialise each pseudocoreset of size M via sampling $(x_m)_{m=1}^M \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$, and sampling θ, $(y_m)_{m=1}^M$ from the statistical model.

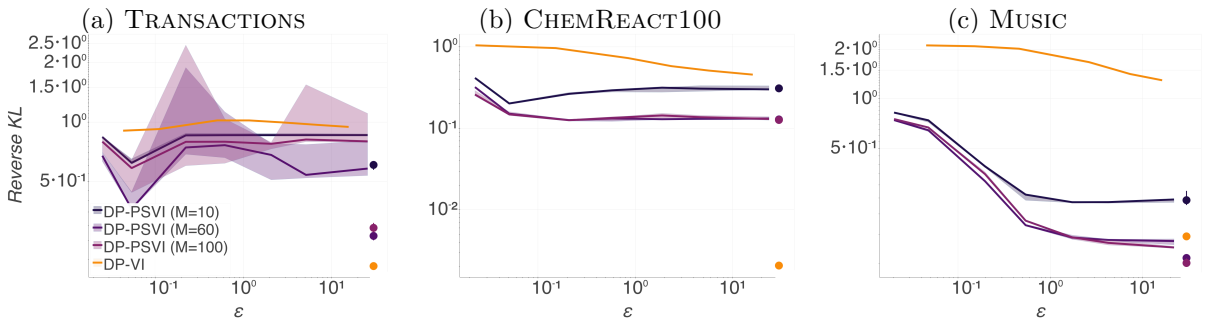


Figure 4.4: Approximate posterior quality over decreasing differential privacy guarantees for private pseudocoresets of varying size plotted against private variational inference [57]. δ is always kept fixed at $1/N$. Markers on the right end of each plot display the errorbar of approximation achieved by the corresponding nonprivate posteriors. Results are displayed over 5 trials for each construction.

Results presented in Fig. 4.3 demonstrate that PSVI achieves consistently the smallest posterior approximation error in the small coreset size regime, offering improvement compared to SparseVI and being competitive with `GIGA (Optimal)`, without the requirement for specifying a weighting function. In Fig. 4.3a, for $M \geq d$ `GIGA (Optimal)` follows a much steeper decrease in KL divergence, reflecting the dependence of its approximation quality on dataset dimension per Proposition 8. In contrast, PSVI typically reaches its minimum at $M < d$. The difference in approximation quality becomes clearer in higher dimensions (e.g. Music, where $d = 237$). Perhaps surprisingly, the private pseudocoreset construction has only marginally worse approximation quality compared to nonprivate PSVI and generally achieves better peformance in comparison to the other state-of-the-art nonprivate coreset constructions. In Fig. 4.4 we present achieved posterior approximation quality via `DP-PSVI`, against a competitive state-of-the-art method (`DP-VI`, [57]). For logistic regression, `DP-VI` infers an approximate posterior from the family of Gaussians with diagonal covariance via ADVI [64], followed by an additional Laplace approximation. Note that by design, `DP-VI` is constrained by the usual Gaussian variational approximation, while `DP-PSVI` is more flexible and can approach the true posterior as M increases—this effect is reflected in nonprivate posteriors as well as data dimensionality grows (see for example Fig. 4.4c). Indeed, we verify that in the high-privacy regime `DP-PSVI` for sufficient pseudocoreset size (which is typically small for tested real-world datasets) offers posterior approximation with better KL divergence compared to `DP-VI`. Our findings indicate that private PSVI offers efficient releases of big data via informative pseudopoints, which enable arbitrary post processing (e.g. running any *nonprivate* black-box algorithm for Bayesian inference), under strong privacy guarantees and without reducing the quality of inference.

4.5 Conclusion

We introduced a new variational formulation for Bayesian coreset construction, which yields efficient summarizations for big and high-dimensional datasets via simultaneously learning pseudodata points locations and weights. We proved limitations of existing variational formulations for coresets and demonstrated that they can be resolved with our new methodology. We proposed an efficient construction scheme via black-box stochastic optimization and showed how it can be adapted for differentially private Bayesian summarization. Finally, we demonstrated the applicability of our methodology on synthetic and real-world datasets, and practical statistical models.

Broader Impact

Pseudocoreset variational inference is a general-purpose Bayesian inference algorithm, hence shares implications mostly encountered in approximate inference methods. For example, replacing the full dataset with a pseudocoreset has the potential to cause inferential errors; these can be partially tempered by using a pseudocoreset of larger size. Note also that the optimization algorithm in this work aims to reduce KL divergence: however the proposed variational objective might be misleading in many applications and lead to incorrect conclusions in certain statistical models (e.g. point estimates and uncertainties might be far off despite KL being almost zero [54]). Moreover, Bayesian inference in general is prone to model misspecification. Therefore, a pseudocoreset summarization based on a wrong statistical model will lead to non-representative compression for inferential purposes. Constructing the coreset on a statistical model suited for robust inference instead of the original one [78, 117], can offer protection against modelling mismatches. Naturally, the utility of generated dataset summary becomes task-dependent, as it has been optimized for a specific learning objective, and cannot be fully transferable to multiple different inference tasks on the same dataset.

Our learnable pseudodata are also generally not as interpretable as the points of previous coreset methods, as they are not real data. And the level of interpretability is model specific. This creates a risk of misinterpretation of pseudocoreset points in practice. On the other hand, our optimization framework does allow the introduction of interpretability constraints (e.g. pseudodata sparsity) to explicitly capture interpretability requirements.

Pseudocoreset-based summarization is susceptible to reproducing potential biases and unfairness existing in the original dataset. Majority-group datapoints in the full dataset which capture information relevant to the statistical task of interest are expected to remain over-represented in the learned summary; while minority-group datapoints might be eliminated, if their distinguishing features are not related to inference. Amending the initialization step to contain such datapoints or using a prior that strongly favors a debiased version of the dataset could both mitigate these concerns; but more study is warranted.

4.6 Technical Results and Proofs

4.6.1 Proof of Proposition 8

In the setting of Proposition 8, both the exact posterior and the coreset posterior are multivariate Gaussian distributions, denoted as $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_w, \Sigma_w)$ respectively. The mean and covariance are

$$\Sigma_1 = \frac{1}{1+N} I_d, \quad \mu_1 = \Sigma_1 \left(\sum_{n=1}^{N} X_n \right), \tag{4.20}$$

and

$$\Sigma_w = \frac{I_d}{1 + \left(\sum_{n=1}^{N} w_n \right)}, \quad \mu_w = \Sigma_w \left(\sum_{n=1}^{N} w_n X_n \right). \tag{4.21}$$

Proof of Proposition 8. By Eqs. (4.20) and (4.21),

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}\left(\pi_w || \pi_1\right) =& \frac{1}{2}\left[\log \frac{|\Sigma_1|}{|\Sigma_w|} - d + \operatorname{tr}\left(\Sigma_1^{-1}\Sigma_w\right) (\mu_1 - \mu_w)^T \Sigma_1^{-1} (\mu_1 - \mu_w)\right] \\ =& \frac{1}{2}\left[-d \log\left(\frac{1+N}{1+\sum_{n=1}^{N} w_n}\right) - d + d\left(\frac{1+N}{1+\sum_{n=1}^{N} w_n}\right) + (\mu_1 - \mu_w)^T \Sigma_1^{-1} (\mu_1 - \mu_w)\right]. \end{aligned} \tag{4.22}$$

Note that $\forall x > 0, x - 1 \geq \log x$, implying that

$$d \log\left(\frac{1+N}{1+\sum_{n=1}^{N} w_n}\right) - d + d\left(\frac{1+N}{1+\sum_{n=1}^{N} w_n}\right) > 0.$$

Thus,

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_w || \pi_1\right) \geq \frac{1}{2} (\mu_1 - \mu_w)^T \Sigma_1^{-1} (\mu_1 - \mu_w). \tag{4.23}$$

Suppose we pick a set $\mathcal{I} \subseteq [N]$, $|\mathcal{I}| = M$ of active indices n where the optimal $w_n \geq 0$, and enforce that all others $n \notin \mathcal{I}$ satisfy $w_n = 0$. Then denoting

$$Y = [X_n : n \notin \mathcal{I}] \in \mathbb{R}^{d \times (N-M)}, \quad X = [X_n : n \in \mathcal{I}] \in \mathbb{R}^{d \times M}, \tag{4.24}$$

we have that for any $w \in \mathbb{R}_+^M$ for those indices $\mathcal{I}$,

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_w || \pi_1\right) \geq \frac{1}{2(N+1)} 1^T Y^T Y 1 + 1^T Y^T X \left(\frac{1}{N+1} - \frac{w}{1+1^T w}\right) \tag{4.25}$$

$$+ \frac{N+1}{2}\left(\frac{1}{N+1} - \frac{w}{1+1^T w}\right)^T X^T X \left(\frac{1}{N+1} - \frac{w}{1+1^T w}\right). \tag{4.26}$$

Relaxing the nonnegativity constraint, replacing $w/(1+1^T w)$ with a generic $x \in \mathbb{R}^M$, and noting that $X^T X$ is invertible almost surely when $M < d$, we can optimize this expression yielding a lower bound on the optimal KL divergence using active index set $\mathcal{I}$,

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{w^\star_{\mathcal{I}}} || \pi_1\right) \geq \frac{1^T Y^T \left(I - X(X^T X)^{-1} X^T\right) Y 1}{2(N+1)}. \tag{4.27}$$

The numerator is the squared norm of $Y1$ minus its projection onto the subspace spanned by the M columns of X. Since $Y1 \sim \mathcal{N}(0, (N-M)I)$, $Y1 \in \mathbb{R}^d$ is an isotropic Gaussian, then its projection into the orthogonal complement of any fixed subspace of dimension M is also an isotropic Gaussian of dimension $d - M$ with the same variance. Since the columns of X are also independent and isotropic, its column subspace is uniformly distributed. So therefore, for each possible choice of $\mathcal{I}$

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{w^\star_{\mathcal{I}}} || \pi_1\right) \geq \frac{N-M}{2(N+1)} Z_{\mathcal{I}}, \quad Z_{\mathcal{I}} \sim \chi^2(d-M). \tag{4.28}$$

Note that the $Z_{\mathcal{I}}$ will have dependence across the $\binom{N}{M}$ different choices of index subset $\mathcal{I}$. Thus, the probability that *all* $Z_{\mathcal{I}}$ are large is

$$\mathbb{P}\left(\min_{\mathcal{I} \subseteq [N], |\mathcal{I}| = M} Z_{\mathcal{I}} > \epsilon\right) \geq 1 - \binom{N}{M} \mathbb{P}\left(Z_{\mathcal{I}} \leq \epsilon\right) \tag{4.29}$$

$$= 1 - \binom{N}{M} F_{d-M}(\epsilon), \tag{4.30}$$

where F_k is the CDF for the χ^2 distribution with k degrees of freedom. The result follows. □

4.7 Gradient Derivations

Throughout, expectations and covariances over the random parameter θ with no explicit subscripts are taken under pseudocoreset posterior $\pi_{u,w}$. We also interchange differentiation and integration without explicitly verifying that sufficient conditions to do so hold.

4.7.1 Weights gradient

First, we compute the gradient with respect to weights vector $w \in \mathbb{R}^M_+$, which is written as

$$\nabla_w \mathrm{D}_{\mathrm{KL}} = -\nabla_w \log Z(u,w) - \nabla_w \mathbb{E}[f(\theta)^T 1] + \nabla_w \mathbb{E}[\tilde{f}(\theta)^T w]. \tag{4.31}$$

For any function $a : \Theta \to \mathbb{R}$, we have that

$$\nabla_w \mathbb{E}\left[a(\theta)\right] = \int \nabla_w \left(\exp\left(w^T \tilde{f}(\theta) - \log Z(u,w)\right)\right) a(\theta) \pi_0(\theta) \mathrm{d}\theta \tag{4.32}$$

$$=\mathbb{E}\left[\left(\tilde{f}(\theta) - \nabla_w \log Z(u,w)\right) a(\theta)\right]. \tag{4.33}$$

Next, we compute the gradient of the log normalization constant via

$$\nabla_w \log Z(u,w) = \int \frac{1}{Z(u,w)} \nabla_w \left(\exp\left(w^T \tilde{f}(\theta)\right)\right) \pi_0(\theta) \mathrm{d}\theta \tag{4.34}$$

$$=\mathbb{E}\left[\tilde{f}(\theta)\right]. \tag{4.35}$$

Combining, we have

$$\nabla_w \mathbb{E}\left[a(\theta)\right] =\mathbb{E}\left[\left(\tilde{f}(\theta) - \mathbb{E}\left[\tilde{f}(\theta)\right]\right) a(\theta)\right]. \tag{4.36}$$

Subtracting $0 = \mathbb{E}\left[a(\theta)\right] \mathbb{E}\left[\tilde{f}(\theta) - \mathbb{E}\left[\tilde{f}(\theta)\right]\right]$ yields

$$\nabla_w \mathbb{E}\left[a(\theta)\right] = \mathrm{Cov}\left[\tilde{f}(\theta), a(\theta)\right]. \tag{4.37}$$

The gradient with respect to w in Eq. (4.10) follows by substituting $1^T f(\theta)$ and $w^T \tilde{f}(\theta)$ for $a(\theta)$ and using the product rule.

4.7.2 Location gradients

Here we take the gradient with respect to a single pseudopoint $u_i \in \mathbb{R}^d$. First note that

$$\nabla_{u_i} \mathrm{D}_{\mathrm{KL}} = -\nabla_{u_i} \log Z(u,w) - \nabla_{u_i} \mathbb{E}[f(\theta)^T 1] + \nabla_{u_i} \mathbb{E}[\tilde{f}(\theta)^T w]. \tag{4.38}$$

For any function $a(u,\theta) : \mathbb{R}^{d\times M} \times \Theta \to \mathbb{R}$, we have

$$\nabla_{u_i} \mathbb{E}\left[a(u,\theta)\right] = \int \nabla_{u_i} \Big(\exp\left(w^T \tilde{f}(\theta) - \log Z(u,w)\right) a(u,\theta)\Big) \pi_0(\theta) \mathrm{d}\theta. \tag{4.39}$$

Using the product rule and recalling from the main text that $h(\cdot,\theta) := \nabla_u f(\cdot,\theta)$,

$$\nabla_{u_i} \mathbb{E}\left[a(u,\theta)\right] = \mathbb{E}\left[\nabla_{u_i} a(u,\theta)\right] + \mathbb{E}\left[a(u,\theta)\left(w_i h(u_i,\theta) - \nabla_{u_i} \log Z(u,w)\right)\right]. \tag{4.40}$$

Taking the gradient of the log normalization constant using similar techniques,

$$\nabla_{u_i} \log Z(u,w) = w_i \mathbb{E}\left[h(u_i,\theta)\right]. \tag{4.41}$$

Combining,

$$\nabla_{u_i} \mathbb{E}\left[a(u,\theta)\right] = \mathbb{E}\left[\nabla_{u_i} a(u,\theta)\right] + w_i \mathbb{E}\left[a(u,\theta)\left(h(u_i,\theta) - \mathbb{E}\left[h(u_i,\theta)\right]\right)\right]. \tag{4.42}$$

Subtracting $0 = \mathbb{E}\left[a(u,\theta)\right] \mathbb{E}\left[\left(h(u_i,\theta) - \mathbb{E}\left[h(u_i,\theta)\right]\right)\right]$ yields

$$\nabla_{u_i} \mathbb{E}\left[a(u,\theta)\right] = \mathbb{E}\left[\nabla_{u_i} a(u,\theta)\right] + w_i \,\mathrm{Cov}\left[a(u,\theta), h(u_i,\theta)\right]. \tag{4.43}$$

The gradient with respect to u_i in Eq. (4.10) follows by substituting $f(\theta)^T 1$ and $\tilde{f}(\theta)^T w$ for $a(u,\theta)$.

4.8 Details on Experiments

4.8.1 Gaussian mean inference

Let the coreset posterior have mean $\mu_{u,w}$ and covariance matrix $\Sigma_{u,w}$. Throughout, expectations and covariances over the random parameter θ with no explicit subscripts are taken under pseudocoreset posterior $\pi_{u,w}$. Define $\Psi := Q^{-1}\Sigma_{u,w}Q^{-T}$, $v_n := Q^{-1}(x_n - \mu_{u,w})$, $\tilde{v}_n := Q^{-1}(u_n - \mu_{u,w})$, and Q to be the Cholesky decomposition of Σ, i.e. $\Sigma := QQ^T$. In order to compute the gradients in Eq. (4.10), we need expressions for $\mathrm{Cov}[f_n, f_m]$, $\mathrm{Cov}[\tilde{f}_n, f_m]$, $\mathrm{Cov}[h(u_i), f_n]$, and $\mathrm{Cov}[h(u_i), \tilde{f}_n]$.

Following [16], we have that

$$\mathrm{Cov}[f_n, f_m] = v_n^T \Psi v_m + \frac{1}{2}\operatorname{tr}\Psi^T\Psi \tag{4.44}$$

$$\mathrm{Cov}[\tilde{f}_n, f_m] = \tilde{v}_n^T \Psi v_m + \frac{1}{2}\operatorname{tr}\Psi^T\Psi. \tag{4.45}$$

We now evaluate the remaining covariance $\mathrm{Cov}[h(u_i), f_m]$; the derivation of $\mathrm{Cov}[h(u_i), \tilde{f}_m]$ follows similarly. We begin by explicitly evaluating the log likelihood gradient and its expectation,

$$h(u_i) = -\Sigma^{-1}(u_i - \theta) \tag{4.46}$$

$$\mathbb{E}\left[h(u_i)\right] = -\Sigma^{-1}(u_i - \mu_{u,w}), \tag{4.47}$$

and again following [16], we have (up to a constant) that

$$f_n = -\frac{1}{2}(x_n - \theta)^T\Sigma^{-1}(x_n - \theta) \tag{4.48}$$

$$\mathbb{E}\left[f_n\right] = -\frac{1}{2}\operatorname{tr}\Psi - \frac{1}{2}\|v_n\|^2. \tag{4.49}$$

Thus using the above definitions,

$$\mathbb{E}\left[h(u_i)\right]\mathbb{E}\left[f_n\right] = \frac{(\operatorname{tr}\Psi + \|v_n\|^2)}{2}Q^{-T}\tilde{v}_i. \tag{4.50}$$

Next,

$$\mathbb{E}\left[h(u_i)f_n\right] = \frac{1}{2}\Sigma^{-1}\mathbb{E}\left[(u_i - \theta)(x_n - \theta)^T\Sigma^{-1}(x_n - \theta)\right]. \tag{4.51}$$

Defining $z \sim \mathcal{N}(0, \Psi)$, and using the above definitions,

$$\mathbb{E}\left[h(u_i)f_n\right] = \frac{1}{2}Q^{-T}\mathbb{E}\left[(\tilde{v}_i - z)(v_n - z)^T(v_n - z)\right]. \tag{4.52}$$

Evaluating the expectation, noting that odd order moments of z are equal to 0,

$$\mathbb{E}\left[h(u_i)f_n\right] = \frac{\|v_n\|^2 + \operatorname{tr}\Psi}{2}Q^{-T}\tilde{v}_i + Q^{-T}\Psi v_n. \tag{4.53}$$

Therefore,

$$\mathrm{Cov}[h(u_i), f_n] = Q^{-T}\Psi v_n, \tag{4.54}$$

and likewise,

$$\mathrm{Cov}[h(u_i), \tilde{f}_n] = Q^{-T}\Psi \tilde{v}_n. \tag{4.55}$$

4.8.2 Bayesian linear regression

Model and gradients details

Here we present the terms involving pseudodata points—the corresponding expressions for original datapoints are the same, after replacing u_m with x_m.

For individual points, dropping normalization constants, we get log-likelihood terms of the form

$$f_m(\theta) = -\frac{1}{2\sigma^2}\left(y_m - \theta^T u_m\right)^2. \tag{4.56}$$

Hence, we obtain for the pseudocoreset posterior

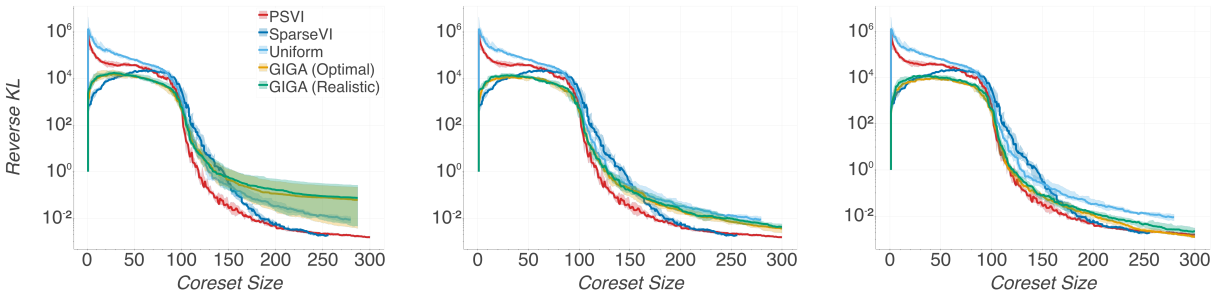
$$\pi_{u,w} = \mathcal{N}(\mu_{u,w}, \Sigma_{u,w}), \quad \text{where} \tag{4.57}$$

$$\Sigma_{u,w} = (\sigma_0^{-2} I + \sigma^{-2} \sum_{m=1}^{M} w_m u_m u_m^T)^{-1}, \quad \mu_{u,w} = \Sigma_{u,w}(\sigma_0^{-2} I \mu_0 + \sigma^{-2} \sum_{m=1}^{M} w_m y_m u_m). \tag{4.58}$$

To scale up computation on large datasets, in our experiment we made use of stochastic gradients for black-box construction of PSVI and SparseVI. Beyond the expressions for individual log-likelihood and (pseudo)coreset posteriors presented above, for pseudocoreset construction we also need the expression for log-likelihood gradient with respect to the pseudodata points, for which we can immediately see that $\nabla_{u_m} f(u_m, \theta) = \frac{1}{\sigma^2}(y_m - \theta^T u_m)\theta$. Over our experiment, we optimized initial learning rates for SparseVI and PSVI via a grid search over $\{0.1, 1, 10\}$.

Additional plots

Here we present some more plots demonstating the dependence of Hilbert coresets approximation quality on the number of random dimensions in the Bayesian linear regression setting presented in Fig. 4.2c. We remind that dimension used at this experiment and throughout the entire experiments section was set to 100. Increasing this number is typically expensive to obtain in practice. As demonstrated in Fig. 4.5, getting higher projection dimension enables better posterior approximation in the problem, while `GIGA (Optimal)` starts offering better quality of approximation than `GIGA (Realistic)`. However, PSVI remains competitive in the



project. dim. = 200 project. dim. = 2,000 project. dim. = 10,000

Figure 4.5: Comparison of Hilbert coresets performance on Bayesian linear regression experiment for increasing projection dimension (over 10 trials).

small coreset regime even for Hilbert coresets with extremely large projection dimensionality, demonstrating the information-geometric limitations that Hilbert coreset constructions are known to face [16].

4.8.3 Bayesian Logistic Regression

Model

In logistic regression we have a set of datapoints $(x_n, y_n)_{n=1}^N$ each corresponding to a feature vector $x_n \in \mathbb{R}^d$ and a label $y_n \in \{-1, 1\}$. Datapoints are assumed to be generated according to following statistical model

$$y_n | x_n, \theta \sim \mathsf{Bern}\left(\frac{1}{1+e^{-z_n\theta}}\right) \quad z_n := \begin{bmatrix} x_n \\ 1 \end{bmatrix}. \tag{4.59}$$

The aim of inference is to compute the posterior over the latent parameter $\theta = [\theta_0 \dots \theta_d]^T \in \mathbb{R}^{d+1}$. Log-likelihood of each datapoint can be expressed as

$$\begin{aligned} f_n(x_n, y_n|\theta) =& \mathbb{1}[y_n = -1] \log\left(1 - \frac{1}{1+e^{-z_n^T\theta}}\right) - \mathbb{1}[y_n = 1] \log\left(1 + e^{-z_n^T\theta}\right) \\ =& -\log\left(1 + \exp(-y_n z_n^T \theta)\right). \end{aligned} \tag{4.60}$$

Hence in pseudocoreset construction we can optimize pseudodata point locations with respect to continuous variable x_n, using the gradient

$$\nabla_{x_n} f_n = \frac{e^{-y_n z_n^T\theta}}{1+e^{-y_n z_n^T\theta}} y_n \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}. \tag{4.61}$$

Datasets description

For logistic regression experiments, we used subsampled and full versions of datasets presented in Table 4.1: a synthetic dataset with $x \in \mathbb{R}^2$ sampled i.i.d. from a $\mathcal{N}(0, I)$ and

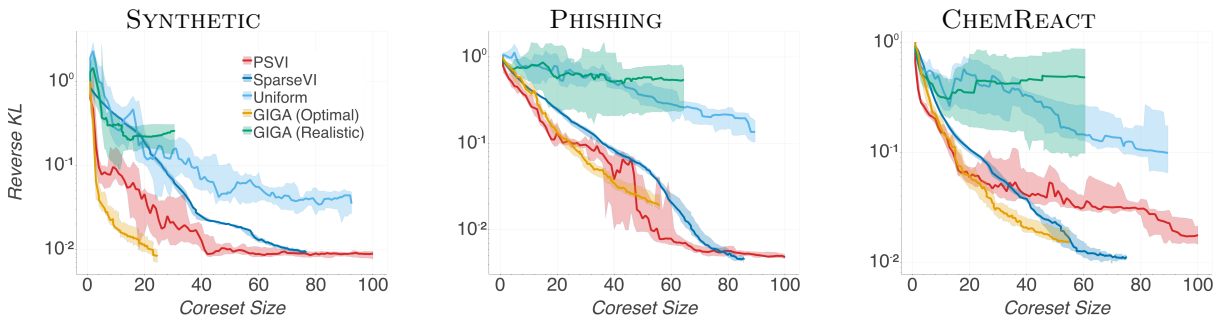


Figure 4.6: Comparison of (pseudo)coreset approximate posterior quality vs coreset size for logistic regression over 10 trials.

$y \in \{-1, 1\}$ sampled from respective logistic likelihood with $\theta = [3, 3, 0]^T$ (SYNTHETIC); a phishing websites dataset reduced to $D = 10$ via PCA (PHISHING); a chemical reactivity dataset with real-valued features corresponding to its first 10 and 100 principal components (CHEMREACT and CHEMREACT100 respectively); a dataset with 50 real-valued features associated with whether each of $100K$ customers of a bank will make a specific transaction (TRANSACTIONS); and a dataset for music analysis, where we consider "classical vs all" genre classification task (MUSIC). Original versions of the four latter datasets are available online respectively at https://www.csie.ntu.edu.tw/˜cjlin/libsvm tools/datasets/binary.html, http://komarix.org/ac/ds, https://www.kaggle.com/c/santander-customer- transaction-prediction/data, and https://github.com/mdeff/fma.

Dataset name	N	D
SYNTHETIC	500	2
PHISHING	500	10
CHEMREACT	500	10
TRANSACTIONS	100,000	50
CHEMREACT100	26,733	100
MUSIC	8,419	237

Table 4.1: Details for datasets used in logistic regression experiments.

Small-scale experiments

In the small-scale experiment, the number of overall gradient updates was set to $T = 1,500$, while minibatch size was set to $B = 400$. Learning rate schedule for SparseVI and PSVI was $\gamma_t = 0.1t^{-1}$. Results presented in Fig. 4.6 indicate that PSVI achieves superior quality to SparseVI for small coreset sizes, and is competitive to `GIGA (Optimal)`, while the latter unrealistically uses true posterior samples to tune a weighting function required over construction.

Reproducibility of Bayesian Logistic Regression experiment

In this subsection we provide additional details for reproducibility of the experimental setup for the Bayesian Logistic Regression experiment presented in Section 4.4.

Posterior approximation metrics, coreset gradients and learning rates Posterior approximation quality was estimated via computing KL divergence between Gaussian distributions fitted on coreset and full data posteriors via Laplace approximation. For both SparseVI and PSVI, gradients were estimated using samples drawn from a Laplace approximation fitted on current coreset weights and points. To optimize initial learning rates for SparseVI and PSVI, we did a grid search over $\{0.1, 1, 10\}$.

Differential privacy loss accounting and hyperparameter selection In the differential privacy experiment, we were not concerned with the extra privacy cost of hyperparameter optimization task. Estimation of differential privacy cost at all experiments was based on TensorFlow privacy implementation of moments accountant for the subsampled Gaussian mechanism[1]. For `DP-PSVI` we used the best learning hyperparameters found for PSVI on the corresponding dataset. The demonstrated range of privacy budgets was generated by decreasing the variance σ of additive Gaussian noise and keeping the rest of hyperparameters involved in privacy accounting fixed. Regarding `DP-VI`, over our experiments we also kept subsampling ratio fixed. We based our implementation of `DP-VI` on authors code,[2] adapting noise calibration according to the adjacency relation used in Section 4.3.3, and the standard differential privacy definition [33]. In our experiment, we used the AdaGrad optimizer [29], with learning rate 0.01, number of iterations $2,000$, and minibatch size 200. Gradient clipping values for `DP-VI` results presented in Fig. 4.4, for Transactions, ChemReact100, and Music datasets were tuned via grid search over $\{1, 5, 10, 50\}$. The values of gradient clipping constant giving best privacy profiles for each dataset, used in Fig. 4.4, were 10, 5, and 5 respectively.

Additional Plots

Evaluation of CPU time requirements Experiments were performed on a CPU cluster node with a 2x Intel Xeon Gold 6142 and 12GB RAM. In the case of PSVI the computation of coreset sizes from 1 to 100 was parallelized per single size over 32 cores in total. Fig. 4.7 shows posterior approximation error vs required CPU time for all coreset construction algorithms over logistic regression on the small-scale and large-scale datasets. As opposed to existing incremental coreset construction schemes, batch construction of PSVI reduces the dependence between coreset size and processing cost: for SparseVI $\Theta(M^2)$ gradient computations are

[1] https://github.com/tensorflow/privacy
[2] https://github.com/DPBayes/DPVI-code

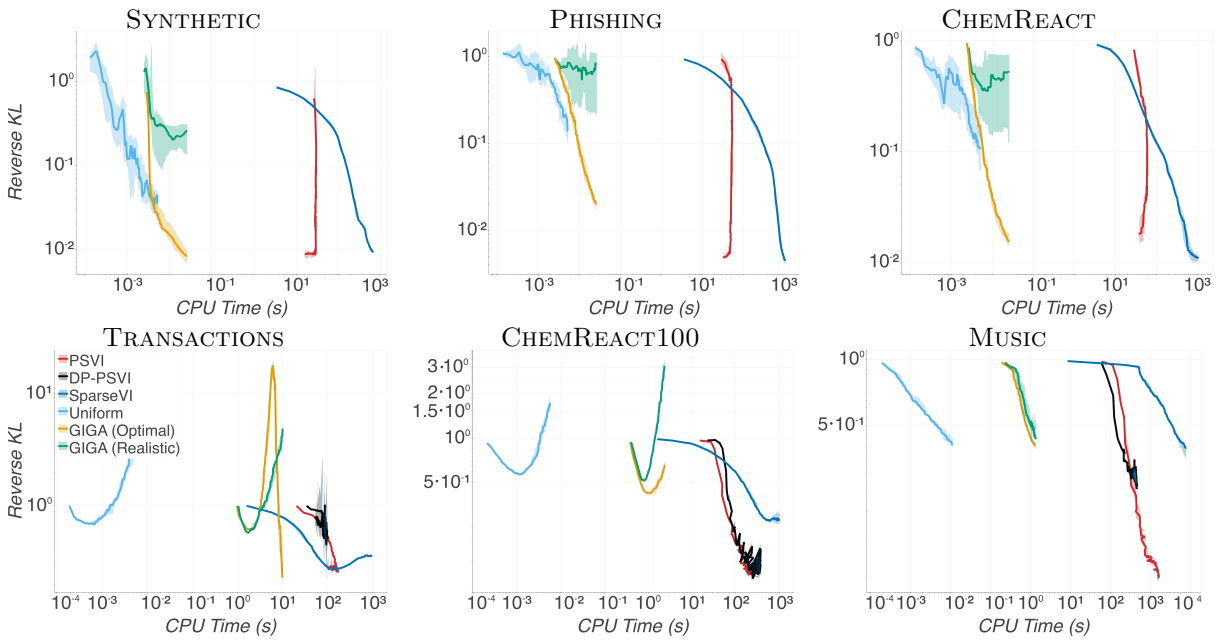


Figure 4.7: Comparison of PSVI and SparseVI approximate posterior quality vs CPU time requirements for logistic regression experiment of Section 4.4.

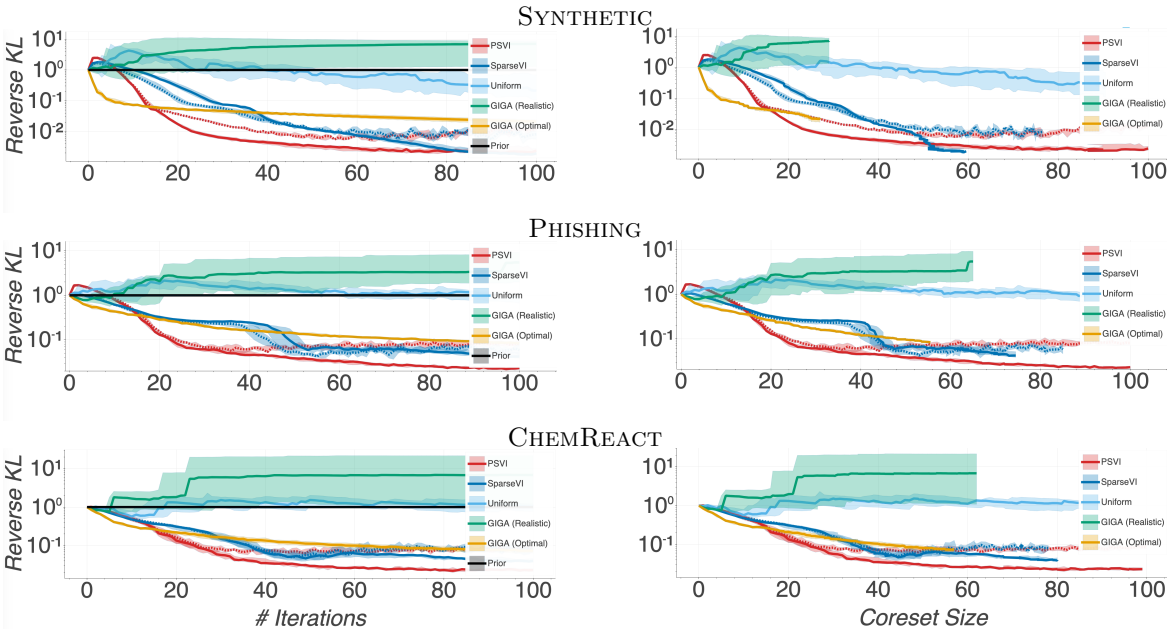


Figure 4.8: Comparison of incremental PSVI and SparseVI approximate posterior quality vs iterations of incremental construction (*left*) and coreset size (*right*) for logistic regression on small-scale experiment. With dashed lines is displayed the posterior quality achieved by incremental PSVI and SparseVI constructions using gradients computed on data subsets of size 256.

required, as this method builds up a coreset one point at a time; in contrast, PSVI requires $\Theta(M)$ gradients since it learns all pseudodata points jointly. Although each gradient step of PSVI is more expensive, practically this implies a steeper decrease in approximation error over processing time compared to SparseVI. In the case of differentially private PSVI, some extra CPU requirements are added due to the subsampled Gaussian mechanism computations.

Incremental scheme for pseudocoreset construction We also experimented with an *incremental scheme for pseudocoreset* construction. According to this scheme, pseudodata points are added sequentially to the pseudocoreset. Similarly to SparseVI, in the beginning of each coreset iteration, we initialize a new pseudodata point at the true datapoint which maximizes correlation with current residual approximation error. Next, we jointly optimize the most recently added pseudodata point location, along with the pseudocoreset weights vector, over a gradient descent loop. As opposed to batch construction, for large coreset sizes the incremental scheme for PSVI does not achieve savings in CPU time compared to SparseVI.

We evaluated coreset construction methods on Bayesian logistic regression. We used $M = 100$ iterations for construction, $S = 100$ Monte Carlo samples per gradient estimation, $T = 100$ iterations for optimization, and learning rate $\gamma_t \propto 0.5t^{-1}$. Coreset posterior samples over the course of construction for SparseVI and incremental PSVI were drawn from a Laplace approximation using current coreset weights and points. We implemented SparseVI and incremental PSVI via computing gradients on the full dataset, as well as using stochastic gradients on subsets of size $B = 256$ for lowering computational cost.

Results presented in Fig. 4.8 demonstrate that incremental PSVI achieves consistently the smallest posterior approximation error, offering improvement compared to SparseVI and even achieving better performance than `GIGA (Optimal)`. We observe that stochastic gradients implementation (dashed lines) reaches a plateau at higher values of KL compared to full gradients (solid lines), but still achieves performance comparable with `GIGA (Optimal)`.

Chapter 5

β-Cores: Robust Large-Scale Bayesian Data Summarization in the Presence of Outliers

Modern machine learning applications should be able to address the intrinsic challenges arising over inference on massive real-world datasets, including scalability and robustness to outliers. Despite the multiple benefits of Bayesian methods (such as uncertainty-aware predictions, incorporation of experts knowledge, and hierarchical modeling), the quality of classic Bayesian inference depends critically on whether observations conform with the assumed data generating model, which is impossible to guarantee in practice. In this work, we propose a variational inference method that, in a principled way, can simultaneously scale to large datasets, and robustify the inferred posterior with respect to the existence of outliers in the observed data. Reformulating Bayes theorem via the β-divergence, we posit a robustified pseudo-Bayesian posterior as the target of inference. Moreover, relying on the recent formulations of Riemannian coresets for scalable Bayesian inference, we propose a sparse variational approximation of the robustified posterior and an efficient stochastic black-box algorithm to construct it. Overall our method allows releasing cleansed data summaries that can be applied broadly in scenarios including structured data corruption. We illustrate the applicability of our approach in diverse simulated and real datasets, and various statistical models, including Gaussian mean inference, logistic and neural linear regression, demonstrating its superiority to existing Bayesian summarization methods in the presence of outliers.

5.1 Introduction

Machine learning systems perpetually collect growing datasets, such as product reviews, posting activity on social media, users feedback on services, or insurance claims. The rich

information content of such datasets has opened up an exciting potential to tackle various practical problems. Hence, recent years have witnessed a surge of interest in scaling up inference in the large-data regime via stochastic and batch methods [6, 49, 122]. Most of related approaches have treated datapoints indiscriminantly; nevertheless, it is well known that not all datapoints contribute equally valuable information for a given target task [43].

Datasets collected in modern applications contain redundant input samples that reflect very similar statistical patterns, or multiple copies of identical observations. Often input aggregates subpopulations emanating from different distributions [133, 135]. Moreover, the presence of outliers is a ubiquitous challenge, attributed to multiple causes. In the first place, noise is inherent in most real-world data collection procedures, creating systematic outliers: crowdsourcing is prone to mislabeling [39] and necessitates laborious data cleansing [66, 87], while measurements commonly capture sensing errors and system failures. Secondly, outliers can be generated intentionally from information contributing parties, who aim to compromise the functionality of the application through data poisoning attacks [8, 12, 67, 62, 105, 43], realised for example via data generation from fake accounts. Outliers detection is challenging, particularly in high dimensions [25, 26]. Proposed solutions often are model-specific, and include dedicated learning components which increase the time complexity of the application, involve extensive hyperparameter tuning, introduce data redundancies, or require model retraining [100, 123, 92, 60, 70, 131]. On the other hand, operating on a corrupted dataset is brittle, and can decisively degrade the predictive performance of downstream statistical tasks, deceptively underestimate model uncertainty and lead to incorrect decisions.

In this work, we design an integrated approach for inference on massive scale observations that can jointly address scalability and data cleansing for complex Bayesian models, via robust data summarization. Our method inherits the full set of benefits of Bayesian inference and works for any model with tractable likelihood function. At the same time, it maintains a high degree of automation with no need for manual data inspection, no additional computational overhead due to robustification, and can tolerate a non-constant number of corruptions. Moreover, our work points to a more efficient practice in large-scale data acquisition, filtering away less valuable samples, and indicating the regions of the data space that are most beneficial for our inference task.

Our solution can be regarded as an extension of Bayesian coreset methods that can encompass robustified inference. Bayesian coresets [53, 18, 16] have been recently proposed as a method that enables Bayesian learning at scale via substituting the complete dataset over inference with an informative sparse subset thereof. Robustified Bayesian inference methods [10] have sought solutions to mismatches between available observations and the assumed data generating model, via proposing heavy-tailed data likelihood functions [51, 93] and localization [22, 115], using robust statistical divergences [41, 61, 78], or inferring datapoints-specific importance weights [117]. Here, we cast coreset construction in the framework of robustified inference, introducing β-*Cores*, a method that learns sparse variational approximations of

the full data posterior under the β-divergence. In this way, we are able to yield summaries of large data that are distilled from outliers, or data subpopulations departing from our statistical model assumptions. Importantly, β-Cores can act as a preprocessing step, and the learned data summaries can subsequently be given as input to any ordinary or robustified black-box inference algorithm.

The rest of this paper is organized as follows. In Sections 5.2 and 5.3 we introduce necessary concepts from Bayesian inference, and present our proposed method. In Section 5.4 we expose experimental results on simulated and real-world benchmark datasets: we consider diverse statistical models and scenarios of extensive data contamination, and demonstrate that, in contrast to existing summarization algorithms, our method is able to maintain reliable predictive performance in the presence of structured and unstructured outliers. Finally, in Section 5.5 we provide conclusions and discuss future works.

5.2 Preliminaries

In this section, we introduce the required concepts from Bayesian inference, present robustness limitations of standard posterior on big data, and outline existing generalizations of the posterior that aim to robustify inference with respect to data mismatch.

5.2.1 Standard Bayesian inference and lack of robustness in the large-data regime

In the context of Bayesian inference, we are interested in updating our beliefs about a vector of random variables $\theta \in \Theta$, initially expressed through a prior distribution $\pi_0(\theta)$, after observing a set of datapoints $x := (x_n)_{n=1}^N \in \mathcal{X}^N$. Posterior on θ can be computed via the application of Bayes rule

$$\pi(\theta|x) = \frac{1}{Z'}\pi(x|\theta)\pi_0(\theta), \tag{5.1}$$

where Z' is a (typically intractable) normalization constant, and $\pi(x|\theta)$ is the likelihood of our observations according to an assumed statistical model. When datapoints are conditionally independent given θ—which is the primary focus of this work—likelihood gets factorized as $\pi(x|\theta) = \Pi_{n=1}^N \pi(x_n|\theta)$. An equivalent formulation of the Bayesian posterior as a solution to an optimization problem was proposed by Zellner [130], which is written as

$$\pi(\theta|x) = \frac{1}{Z'}\exp\left(-\mathrm{d}_{\mathrm{KL}}\left(\hat{\pi}(x)||\pi(x|\theta)\right)\right)\pi_0(\theta). \tag{5.2}$$

In the above, $\hat{\pi}(x)$ is the empirical distribution of the observed datapoints. The exponent $\mathrm{d}_{\mathrm{KL}}\left(\hat{\pi}(x)||\pi(x|\theta)\right) := -\sum_{n=1}^N \log \pi(x_n|\theta)$ corresponds (up to a constant) to the *cross-entropy*, which is equal to the empirical average of negative log-likelihoods of the datapoints, and

quantifies the expected loss incurred by our estimates for the model parameters θ over the available observations, under the *Kullback-Leibler (KL) divergence*.

When N is large, the Bayesian posterior is strongly affected by perturbations in the observed data space. To develop an intuition on this, assuming that the true and observed data distributions have densities π_θ and π_{obs} respectively, we can rewrite an approximation of Eq. (5.2) via the KL divergence ($\mathrm{D_{KL}}$) as [78]

$$\pi(\theta|x) \propto \exp\left(\sum_{n=1}^{N} \log \pi(x_n|\theta)\right) \pi_0(\theta) \doteq \exp\left(N \int \pi_{\text{obs}} \log \pi_\theta\right) \pi_0(\theta) \tag{5.3}$$

$$:= \exp\left(-N \mathrm{D_{KL}}\left(\pi_{\text{obs}} || \pi_\theta\right)\right) \pi_0(\theta), \tag{5.4}$$

where $\doteq$ denotes agreement to first order in exponent.[1] Hence, due to the large N in the exponent, small changes to π_{obs} will have a large impact on the posterior.

5.2.2 Robustified posteriors

Robust inference methods aim to adapt Eq. (5.1) to formulations that can address the case of observations departing from model assumptions, as often happening in practice, e.g. due to misspecified shapes of data distributions and number of components, or due to the presence of outliers. In such formulations [21, 58, 40, 34], Bayesian updates rely on utilising robust divergences instead of the KL divergence, to express the losses over the data.

A popular choice [41, 61] for enhancing robustness of inference is replacing the log-likelihood terms arising in Eq. (5.2) with the *β-divergence* (or *density power divergence*) [9, 20], which yields the following posterior for θ [44, 61]

$$\pi_\beta(\theta|x) \propto \exp\left(-\mathrm{d}_\beta\left(\hat{\pi}(x) || \pi(x|\theta)\right)\right) \pi_0(\theta), \tag{5.5}$$

where

$$\mathrm{d}_\beta\left(\hat{\pi}(x) || \pi(x|\theta)\right) := -\sum_{n=1}^{N} \underbrace{\left(\frac{\beta+1}{\beta} \pi(x_n|\theta)^\beta + \int_{\mathcal{X}} \pi(\chi|\theta)^{1+\beta} d\chi\right)}_{:= f_n(\theta)}, \tag{5.6}$$

with $\beta > 0$. We refer to quantities defined in Eqs. (5.5) and (5.6) as the *β-posterior* and *β-likelihood* respectively. Noticeably, the individual terms $f_n(\theta)$ of the β-likelihood allow attributing *different strength of influence to each of the datapoints*, depending on their accordance with the model assumptions. As densities get raised to a suitable power β, outlying observations are exponentially downweighted. When $\beta \to 0$, Eq. (5.2) is recovered and all datapoints are treated equally.

In the presentation above we focused on modeling observations $(x_n)_{n=1}^N$ (unsupervised learning). In the case of supervised learning on data pairs $(x_n, y_n)_{n=1}^N \in (\mathcal{X} \times \mathcal{Y})^N$, the

[1] i.e. $a_n \doteq b_n$ iff $(1/n)\log(a_n/b_n) \to 0$

respective expression for individual terms of β-likelihood[2] is [9]

$$f_n(\theta) := -\frac{\beta+1}{\beta}\pi(y_n|x_n,\theta)^\beta + \int_{\mathcal{Y}} \pi(\psi|x_n,\theta)^{1+\beta} d\psi. \tag{5.7}$$

5.3 Method

In this section we discuss β-Cores, our unified solution to the robustness and scalability challenges of large-scale Bayesian inference. Section 5.3.1 introduces the main quantity of interest in our inference method, and shows how it addresses the exposed issues. Section 5.3.2 presents an iterative algorithm that allows efficient approximate computations of our posterior.

5.3.1 Sparse β-posterior

Scaling up the computation of Eq. (5.5) in the regime of massive datasets for non-conjugate models is challenging: similarly to Eq. (5.1), applying Markov chain Monte Carlo (MCMC) methods to sample from the β-posterior, implies a computational cost scaling at order $\Theta(N)$.

Bayesian coresets [53, 18] have been recently proposed as a method to circumvent the computational cost for the purposes of approximate inference via summarizing the original dataset $(x_n)_{n=1}^N$ with a small learnable subset of weighted datapoints $(x_m, w_m)_{m=1}^M$, where $(w_m)_{m=1}^M \in \mathbb{R}_+^M$, $M \ll N$. Substituting Eq. (5.6) in Eq. (5.5), allows us to explicitly introduce a weights vector $w \in \mathbb{R}_{\geq 0}^N$ in the posterior, and rewrite the latter in the general form

$$\pi_{\beta,w}(\theta|x) = \frac{1}{Z(\beta,w)} \exp\left(\sum_{n=1}^N w_n f_n(\theta)\right) \pi_0(\theta). \tag{5.8}$$

In the case of the β-posterior on the full dataset Eq. (5.5), we have $w = \mathbb{1} \in \mathbb{R}^N$; for coreset posteriors this vector acts as a learnable parameter and attains a non-trivial sparse value, with non-zero entries corresponding to the elements of the full dataset that are selected over the summarization.

Although Bayesian coresets can dramatically reduce inference time, they inherit the susceptibility of Bayesian posterior to data mismatch in the large data regime: even though the number of points used in inference gets reduced, these points are now weighted, hence the remark of Eq. (5.4) can carry over in coresets posterior.

The recent formulation of Riemannian coresets [16] has framed the problem of coreset construction as Variational Inference (VI) in a sparse exponential family. Our method provides a natural extension of this framework to robust divergences. Here we aim to approximate data posterior via a *sparse β-posterior*, which can be expressed as follows

$$w^* = \arg\min_{w\in\mathbb{R}^N} \mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w}||\pi_\beta\right) \quad \text{s.t.} \quad w \geq 0,\ ||w||_0 \leq M, \tag{5.9}$$

[2] In this context for simplicity we use notation $f_n(\cdot)$ to denote $f(y_n|x_n,\cdot)$.

Algorithm 2 Incremental construction of sparse β-posterior

procedure β-CORES($f, \pi_0, x, M, B, S, T, (\gamma_t)_{t=1}^{\infty}, \beta$)
  $w \leftarrow \mathbf{0} \in \mathbb{R}^M, \quad g \leftarrow \mathbf{0} \in \mathbb{R}^{S\times M}, \quad g' \leftarrow \mathbf{0} \in \mathbb{R}^{S\times B}, \quad \mathcal{I} \leftarrow \emptyset$
  for $m = 1, \ldots, M$ **do**
    $(\theta)_{s=1}^{S} \overset{\text{i.i.d.}}{\sim} \pi_{\beta,w} \propto \exp\left(w^T f\right) \pi_0(\theta)$
    $\mathcal{B} \sim \mathsf{UnifSubset}\left([N], B\right)$
    $g_s \leftarrow \left(f(x_m, \theta_s, \beta) - \frac{1}{S}\sum_{r=1}^{S} f(x_m, \theta_r, \beta)\right)_{m\in\mathcal{I}} \in \mathbb{R}^M$
    $g'_s \leftarrow \left(f(x_b, \theta_s, \beta) - \frac{1}{S}\sum_{r=1}^{S} f(x_b, \theta_r, \beta)\right)_{b\in\mathcal{B}} \in \mathbb{R}^B$
    $\widehat{\mathrm{Corr}} \leftarrow \mathrm{diag}\left[\frac{1}{S}\sum_{s=1}^{S} g_s g_s{}^T\right]^{-\frac{1}{2}} \left(\frac{1}{S}\sum_{s=1}^{S} g_s \left(\frac{N}{B} 1^T g'_s - w^T g_s\right)\right) \in \mathbb{R}^M$
    $\widehat{\mathrm{Corr}}' \leftarrow \mathrm{diag}\left[\frac{1}{S}\sum_{s=1}^{S} g'_s g'_s{}^T\right]^{-\frac{1}{2}} \left(\frac{1}{S}\sum_{s=1}^{S} g'_s \left(\frac{N}{B} 1^T g'_s - w^T g_s\right)\right) \in \mathbb{R}^B$
    $n^\star \leftarrow \arg\max_{n\in[m]\cup[B]} \left(\left|\widehat{\mathrm{Corr}}\right| \cdot \mathbb{1}[n \in \mathcal{I}] + \widehat{\mathrm{Corr}}' \cdot \mathbb{1}[n \notin \mathcal{I}]\right), \quad \mathcal{I} \leftarrow \mathcal{I} \cup \{n^\star\}$
    for $t = 1, \ldots, T$ **do**
      $(\theta)_{s=1}^{S} \overset{\text{i.i.d.}}{\sim} \pi_{\beta,w}(\theta) \propto \exp\left(w^T f\right) \pi_0(\theta)$
      $\mathcal{B} \sim \mathsf{UnifSubset}\left([N], B\right)$
      for $s = 1, \ldots, S$ **do**
        $g_s \leftarrow \left(f(x_m, \theta_s, \beta) - \frac{1}{S}\sum_{r=1}^{S} f(x_m, \theta_r, \beta)\right)_{m\in\mathcal{I}} \in \mathbb{R}^M$
        $g'_s \leftarrow \left(f(x_b, \theta_s, \beta) - \frac{1}{S}\sum_{r=1}^{S} f(x_b, \theta_r, \beta)\right)_{b\in\mathcal{B}} \in \mathbb{R}^B$
      $\hat{\nabla}_w \leftarrow -\frac{1}{S}\sum_{s=1}^{S} g_s \left(\frac{N}{B} 1^T g'_s - w^T g_s\right)$
      $w \leftarrow \max(w - \gamma_t \hat{\nabla}_w, 0)$
  return w

In the following we denote expectations and covariances under $\theta \sim \pi_{\beta,w}(\theta|x)$ as $\mathbb{E}_{\beta,w}$ and $\mathrm{Cov}_{\beta,w}$ respectively. Then the KL divergence is written as

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w}||\pi\right) := \mathbb{E}_{\beta,w}\left[\log \frac{\pi_{\beta,w}}{\pi_\beta}\right]. \tag{5.10}$$

In our formulation it is easy to observe that posteriors of Eq. (5.8) form a set of *exponential family distributions* [114], with natural parameters $w \in \mathbb{R}^N_{\geq 0}$, sufficient statistics $(f_n(\theta))_{n=1}^N$, and log-partition function $\log Z(\beta, w)$. Following [16], the objective can be expanded as

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w}||\pi\right) = \log Z(\beta) - \log Z(\beta, w) \tag{5.11}$$

$$- \sum_{n=1}^{N} \mathbb{E}_{\beta,w}\left[f_n(\theta) - w_n f_n(\theta)\right], \tag{5.12}$$

and minimized via gradient descent on w. The gradient of the objective of Eq. (5.12) can be derived in closed form, as

$$\nabla_w \mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w}||\pi\right) = -\mathrm{Cov}_{\beta,w}\left[f, (1-w)^T f\right], \tag{5.13}$$

where $f := [f_1(\theta) \ldots f_N(\theta)]^T$.

5.3.2 Black-box stochastic scheme for incremental coreset construction

To scale up coreset construction on massive datasets we use stochastic gradient descent on minibatches $\mathcal{B} \sim \mathsf{UnifSubset}([N], B)$, with $B \ll N$. The covariance of Eq. (5.13) required

for exact gradient computation of the variational objective is generally not available in analytical form. Hence, for our black-box coreset construction we approximate this quantity via Monte Carlo estimates, using samples of the unknown parameters from the coreset posterior. These samples can be efficiently obtained with complexity $O(M)$ (not scaling with dataset size N) due to the sparsity of the coreset posterior over the procedure. The proposed black-box construction makes no assumptions on the statistical model other than having tractable β-likelihoods. We employ a two-step incremental scheme, with complexity of order $O\left(M(M+B)ST\right)$, where S is the number of samples from the coreset posterior, and T is the total number of iterations over coreset points weights optimization. The full incremental construction is outlined in Algorithm 2.

Next datapoint selection

We first select the next datapoint to include in our coreset summary, via a greedy selection criterion. Although maximizing decrease in KL locally via Eq. (5.13), seems to be the natural greedy choice here, using the information-geometric argument presented in [16], we use instead the following correlation maximization criterion:

$$x_m = \arg\max_{x_m \in \mathcal{I} \cup \mathcal{B}} \begin{cases} \left|\mathrm{Corr}_{\beta,w}\left[f_m, \frac{N}{B} 1^T f - w^T f\right]\right| & w_m > 0 \\ \mathrm{Corr}_{\beta,w}\left[f_m, \frac{N}{B} 1^T f - w^T f\right] & w_m = 0, \end{cases} \tag{5.14}$$

where we denoted by $\mathcal{I}$ the set of coreset points. The correlations for coreset and minibatch datapoints are empirically approximated as in lines 8 and 9 of Algorithm 2 respectively.

Coreset points reweighting

After adding a new datapoint we update the coreset weight vector $w \in \mathbb{R}_{\geq 0}$ via T steps of projected stochastic gradient descent, using the Monte Carlo estimate of Eq. (5.13) per line 17 of Algorithm 2.

Summarization of observations groups and batches. Apart from working at the individual datapoints level, our scheme also enables summarizing batches and groups of observations. Acquiring efficiently informative batches of datapoints can replace random minibatch selection commonly used in stochastic optimization for large-scale model training. This extension can also be quite useful in situations where datapoints are partitioned in clusters, e.g. according to demographic information. For example, when gender and age features are available in datasets capturing users movies habits, collected datapoints can be binned accordingly, and our group summarization technique will allow extracting informative combinations of demographic groups that can jointly summarize the entire population's information. The robustness properties of β-Cores in such applications can aid removing group bias, and rejecting groups with large fractions of outliers. Algorithm 2 is again directly

applicable, where g_s vectors are now summed over the corresponding datapoints of each batch or group.

5.4 Experiments & Applications

We examine the inferential results achieved by our method under 3 statistical models, in scenarios capturing different types of data mismatch with reality. β-Cores is compared against a uniformly random sampling baseline, and stochastic batch implementations of two existing Riemannian coreset methods: (i) SparseVI [16], which builds up a coreset according to an incremental scheme similar to ours, considering the standard likelihood function terms evaluated on the dataset points, and (ii) PSVI [75], which runs a batch optimization on a set of pseudopoints, and uses standard likelihood evaluations to jointly learn the pseudopoints weights and locations so that the extracted summary resembles the statistics of the full dataset.

We default the number of iterations in the optimization loop over gradient-based coreset constructions to $T = 500$, using a learning rate $\gamma_t \propto t^{-1}$ and $S = 100$ random projections per gradient computation. For consistency with the compared baselines, we evaluate inference results obtained by β-Cores using the classical Bayesian posterior from Eq. (5.1) conditioned on the corresponding robustified data summary. Additional details on used benchmark datasets are presented in Section 5.7. Code is available at https://github.com/dionman/beta-cores.

5.4.1 Simulated Gaussian Mean Inference under Stuctured Data Contamination

In this experiment we study how β-Cores behaves in the setting of mean inference on synthetic d-dimensional data, sampled i.i.d. from a normal distribution with known covariance,

$$\theta \sim \mathcal{N}(\mu_0, \Sigma_0), \qquad x_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\theta, \Sigma), \qquad n = 1, \ldots, N. \tag{5.15}$$

In the presented results, we use priors $\mu_0 = \mathbf{0}$ and $\Sigma_0 = I$, dimensionality $d = 20$ and dataset size $N = 5,000$.

We consider the case of structured data corruption existing in the observations, simulated as follows: Observed datapoints are typically sampled from a Gaussian $\mathcal{N}(\mathbf{1}, I)$. At a percentage $F\%$, data collection fails; in this case, datapoints are collected from a shifted Gaussian $\mathcal{N}(\mathbf{10}, I)$. Consequently, the observed dataset forms a Gaussian mixture with two components; however, our statistical model assumes only a single Gaussian.

All computations involved in the coreset construction and posterior evaluation in this experiment can be performed in closed form [16]. We apply the batch scheme of Algorithm 2, sampling from the exact coreset posterior over gradient estimation. The used (β-)likelihood

equations are outlined in Section 5.6.1. For all coreset methods, constructions are repeated for up to $M = 200$ iterations, with $\gamma_t = t^{-1}$. Notice that our setting does not imply that maximum summary size contains 200 datapoints: often over the iterations an already existing summary point may be selected again, resulting in smaller coresets.

Fig. 5.1a presents the results obtained by the different coreset methods. We stress-test their performance under varying amounts of data corruption (from top to bottom, 0%, 15%, and 30% of the datapoints get replaced by outliers). We can verify that β-Cores with $\beta = 0.01$ is on par with existing Riemannian coresets in an uncontaminated dataset. Noticeably, β-Cores remains robust to high levels of structured corruption (even up to 30% of the dataset), giving reliable posterior estimates; KL divergence plots in Fig. 5.1b reconfirm the superiority of inference via β-Cores. On the other hand, in the presence of outliers, previous Riemannian coresets performance degrades quickly, offering similar posterior inference quality with random sampling. The KL divergence from the cleansed data posterior for existing summarizations and uniform sampling increases with observations failure probability, as it asymptotically converges to the Bayesian posterior computed on the corrupted dataset.

Moreover, in the case of contaminated datasets, baseline coresets are quite confident in their wrong predictive posteriors: they keep assigning the same weight to all observations and hence do not adjust their posterior uncertainty estimates, in spite of having to describe contradicting data. In contrast, β-Cores discards samples from the outlying group and can confidently explain the inliers, despite the smaller effective sample size: indeed, Fig. 5.1b shows that the achieved KL divergence from the exact posterior is at same order of magnitude regardless of failure probability.

We can however notice that, for coreset sizes growing beyond 60 points—despite remaining consistently better compared to the baselines—β-Cores starts to present some instability over trials in contaminated dataset instances. This effect is attributed to the small value of the β hyperparameter selected for the demonstration (so that this value can successfully model the case of clean data). As a result, eventually some outliers might be allowed to enter the summary for large coreset sizes. The instability can be resolved by increasing β according to the observations failure probability.

5.4.2 Bayesian Logistic Regression under Mislabeling and Feature Noise

In this section, we study the robustness achieved by β-Cores on the problem of binary classification under unreliable measurements and labeling. We test our methods on 3 benchmark datasets with varying dimensionality (10-127 dimensions, more details on the data are provided in Section 5.7). We observe data pairs $(x_n, y_n)_{n=1}^N$, where $x \in \mathbb{R}^d$, $y_n \in \{-1, 1\}$, and use the Bayesian logistic regression model to describe them,

$$y_n | x_n, \theta \sim \mathsf{Bern}\left(\frac{1}{1 + e^{-z_n^T \theta}}\right), \qquad z_n := \begin{bmatrix} x_n \\ 1 \end{bmatrix}. \tag{5.16}$$

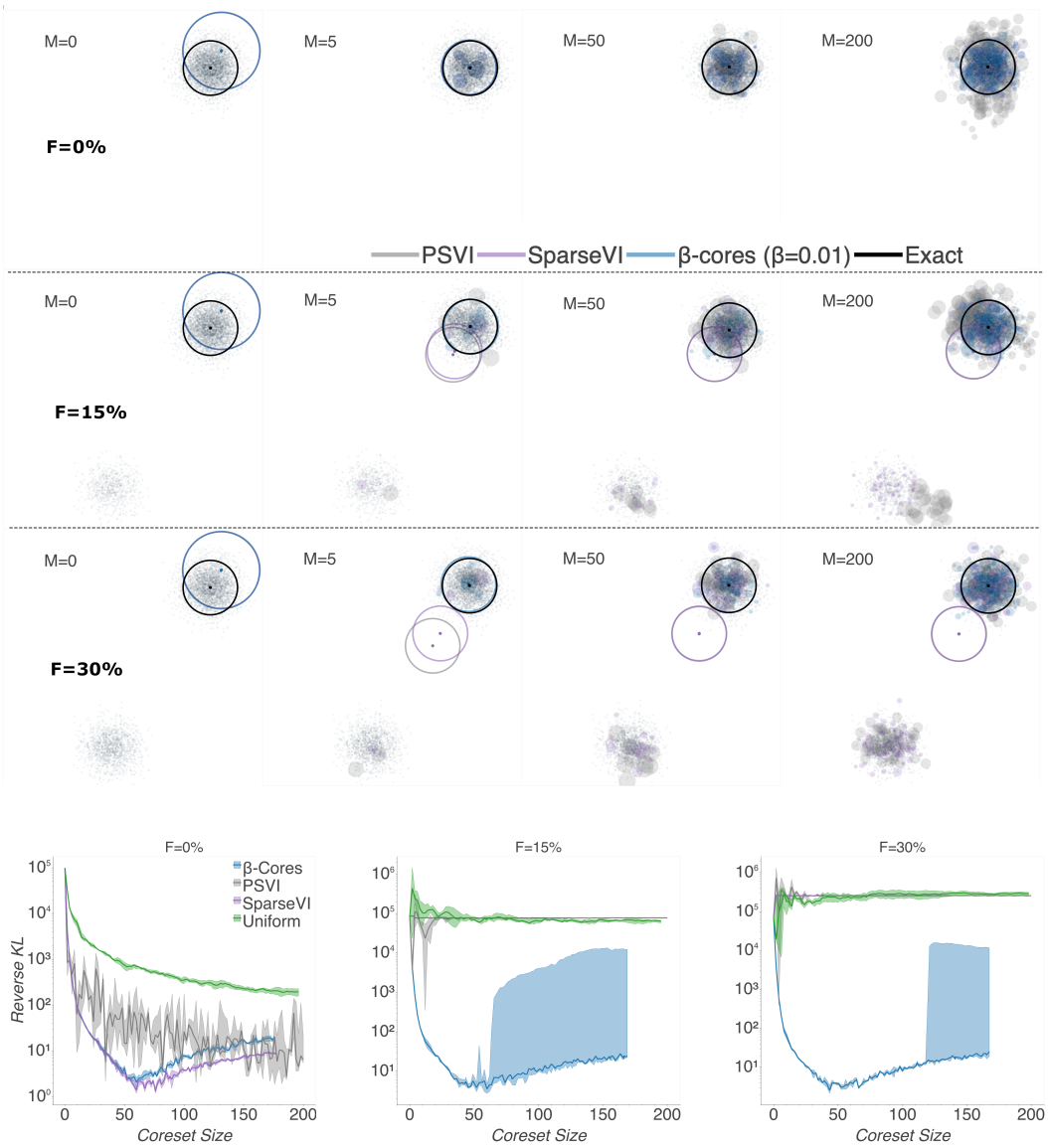


Figure 5.1: (a) Scatterplot of the observed datapoints projected on two random axes, overlaid by the corresponding coreset points and predictive posterior 3σ ellipses for increasing coreset size (from left to right). Exact posterior (illustrated in black) is computed on the dataset after removing the group of outliers. From top to bottom, the level of structured contamination increases. Classical Riemannian coresets are prone to model misspecification, adding points from the outlying component, while β-Cores adds points only from the uncontaminated subpopulation yielding better posterior estimation. (b) Reverse KL divergence between coreset and true posterior, averaged over 5 trials. Solid lines display the median KL divergence, with shaded areas showing 25^{th} and 75^{th} percentiles of KL divergence.

β-likelihood terms required in our construction are computed in Section 5.6.2.

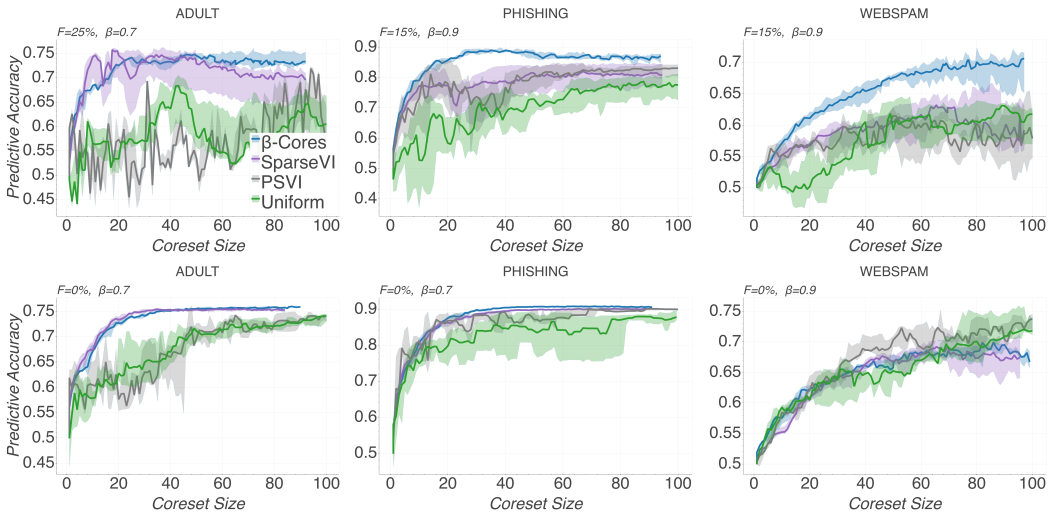


Figure 5.2: Predictive accuracy vs coreset size for logistic regression experiments over 10 trials on 3 large-scale datasets. Solid lines display the median accuracy, with shaded areas showing 25^{th} and 75^{th} percentiles. Dataset corruption rate F, and β value used in β-Cores for each experiment are shown on the figures. The bottom row plots illustrate the achieved predictive performance under no contamination.

Data corruption is simulated by generating outliers in the input and output space similarly to [41]: For corruption rate F, we sample two random subsets of size $F \cdot N$ from the training data. For the datapoints in the first subset, we replace the value of half of the features with Gaussian noise sampled i.i.d. from $\mathcal{N}(0, 5)$; for the datapoints in the other subset, we flip the binary label. Over construction we use Laplace approximation [73] to efficiently draw samples from the (non-conjugate) coreset posterior, while over evaluation coreset posterior samples are obtained via NUTS [50]. We evaluate accuracy over the test set, predicting labels according to the maximum log-likelihood rule under the posterior θ sampling distribution. Learning rate schedule was set to $\gamma_t = c_0 t^{-1}$, with c_0 set to 1 for SparseVI and β-Cores, and 0.1 for PSVI. The values for hyperparameter β and learning rates γ_t were chosen via cross-validation.

Fig. 5.2 illustrates that β-Cores shows competitive performance with the classic Riemannian coresets in the absence of data contamination (bottom row), while it consistently achieves the best predictive accuracy in corrupted datasets (top row). On the other hand, ordinary summarization techniques, although overall outperforming random sampling for small coreset sizes, soon attain degraded predictive performance on poisoned data: by construction, via increasing coreset size, Riemannian coresets are expected to converge to the Bayesian posterior computed on the corrupted dataset. All baselines present noticeable degradation in their predictive accuracy when corruption is introduced (typically more than 5%), which is not the case for our method: β-Cores is designed to support corrupted input and, for a

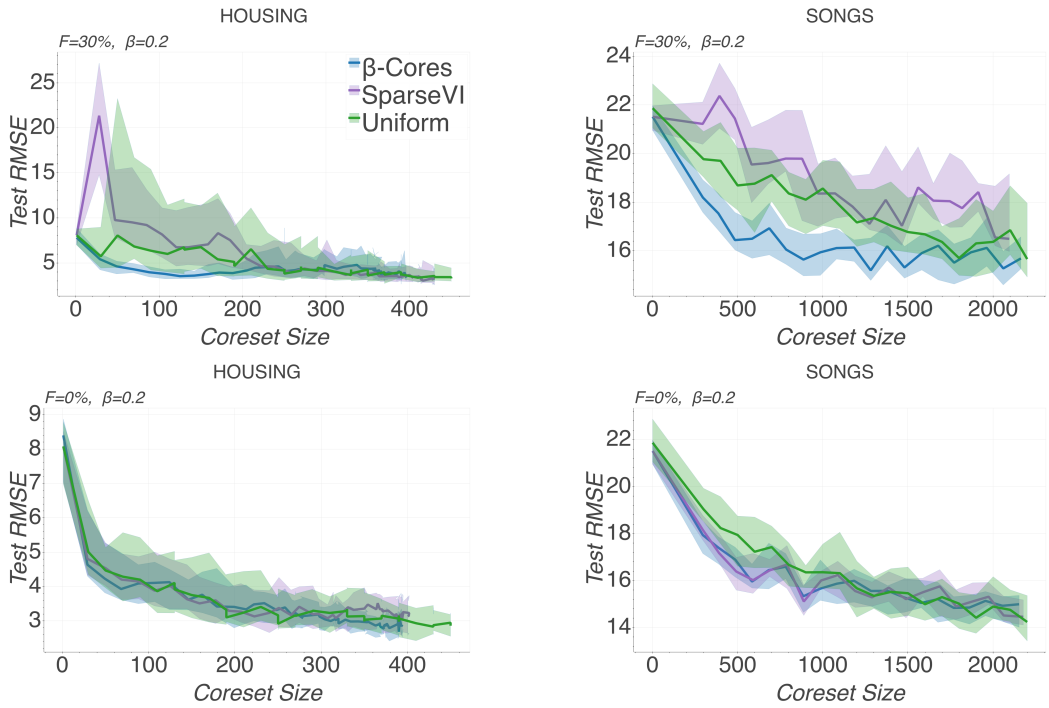


Figure 5.3: Test RMSE vs coreset size for neural linear regression experiments averaged over 30 trials. Solid lines display the median RMSE, with shaded areas showing 25$^{\text{th}}$ and 75$^{\text{th}}$ percentiles. Dataset corruption rate F, and β value used in β-Cores for each experiment are shown on the figures. The bottom row plots illustrate the achieved predictive performance under no contamination.

well-tuned hyperparameter β, maintains similar performance in the presence of outliers, while practically it can even achieve improvement (as occurring for the WebSpam data).

5.4.3 Neural Linear Regression on Noisy Data Batches

Here we use the coresets extension for batch summarization to efficiently train a neural linear model on selected data minibatches. Neural linear models perform Bayesian linear regression on the representation of the last layer of a deterministic neural network feature extractor [104, 94, 89]. The corresponding statistical model is as follows

$$(y_n)_{n=1}^N = \theta^T z(x_n) + \epsilon_n, \quad (\epsilon_n)_{n=1}^N \sim \mathcal{N}(0, \sigma^2). \tag{5.17}$$

The neural network is trained to learn an adaptive basis $z(\cdot)$ from N datapoint pairs $(x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$, which we then use to regress $(y_n)_{n=1}^N$ on $(z(x_n))_{n=1}^N$, and yield uncertainty aware estimates of θ. More details on the model-specific formulae entering coresets construction are provided in Section 5.6.3. Input and output related outliers are simulated as in Section 5.4.2, while here, for the output related outliers, y_n gets replaced by Gaussian noise. Corruption occurs over a percentage $F\%$ of the total number of minibatches of the dataset, while the

remaining minibatches are left uncontaminated. Each poisoned minibatch gets 70% of its points substituted by outliers.

We evaluate β-Cores, SparseVI and random sampling on two benchmark regression datasets (detailed in Section 5.7). All coresets are initialized to a small batch of datapoints sampled uniformly at random from the dataset inliers. Over incremental construction, we interleave each minibatch selection and weights optimization step of the coreset with a training round for the neural network, constrained on the current coreset datapoints. Each such training round consists of 10^3 minibatch gradient descent steps using the AdaGrad optimizer [29]. Our neural architecture is comprised of two fully connected hidden layers, batch normalization and ReLU activation functions. The values of coreset size at initialization, batch size added per coreset iteration, and units at each neural network hidden layer are set respectively to 20, 10 and 30 for the Housing, and 200, 100 and 100 for the Songs dataset.

Fig. 5.3 (bottom row) shows that β-Cores are competitive with the baselines in the absence of data corruption, achieving similar predictive performance over the entire range of tested coreset sizes. Under data poisoning (top row), β-Cores is the only method that offers monotonic decrease of test RMSE for increasing summary size from the beginning of the experiment. On the other hand, baselines present unreliable predictive performance for small coreset sizes: random sampling and SparseVI are both prone to including corrupted data batches, whose misguiding information gets expressed on the flexible representations learnt by the neural network, requiring a larger summary size to reach the RMSE of β-Cores.

5.4.4 Efficient Data Acquisition from Subpopulations for Budgeted Inference

We consider the scenario where a machine learning service provider aims to fit a binary classification model to observations coming from multiple subpopulations of data contributors. The provider aims to maximize the predictive accuracy of the model, while adhering to a budget on the total number of subpopulations from which data can be used over inference. Budgeted inference can be motivated by several practical requirements: First, restricting the total number of datapoints used over learning to a smaller informative subset aids scalability—which is the primary motivation for coresets. Moreover, taking decisions at the subpopulations level regarding which groups of datapoints are useful for the task, without the need to inspect datapoints individually, reduces the privacy loss incurred over the data selection stage, and can be integrated in machine learning pipelines that follow formal hierarchical privacy schemes. Finally, subpopulations valuation can guide costly experimental procedures, via inducing knowledge regarding which group combinations are most beneficial in summarizing the entire population of interest [89, 111], and hence should be prioritised over data collection.

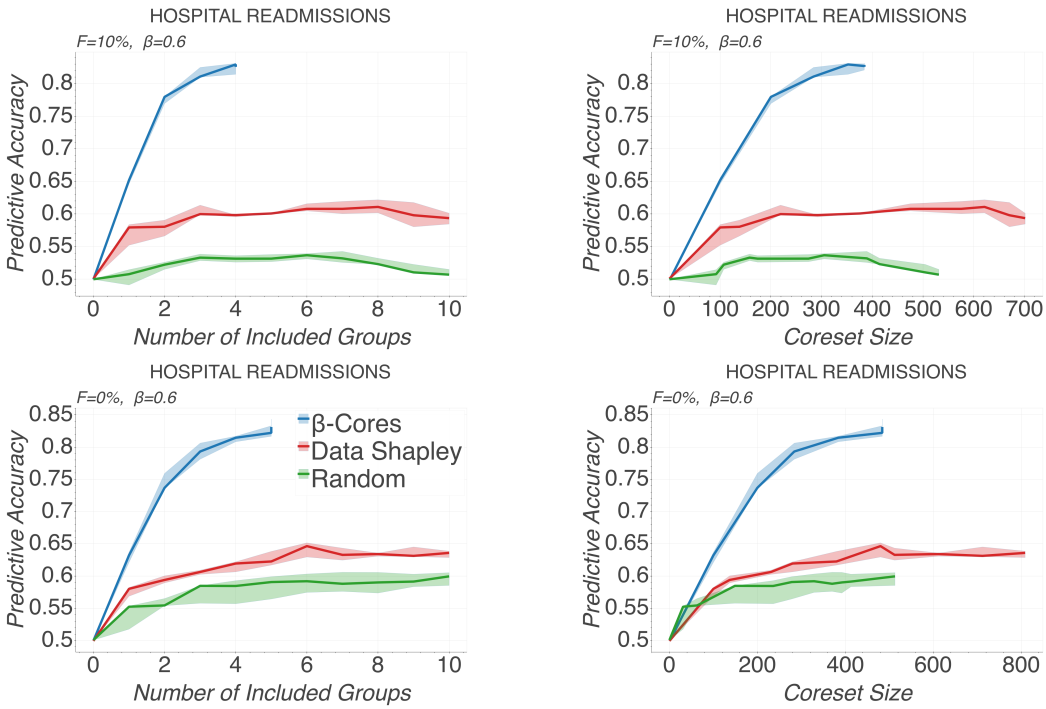


Figure 5.4: Predictive accuracy against number of groups (left) and number of datapoints (right) selected for inference. Compared group selection shemes are β-Cores, selection according to Shapley values based ranking, and random selection. The experiment is repeated over 5 trials, on a contaminated dataset containing a 10% of crafted outliers distributed non-uniformly across groups (top row), and a clean dataset (bottom row).

In this study we use a subset of more than $60K$ datapoints from the HospitalReadmissions dataset (for further details see Section 5.7). Using combinations of age, race and gender information of data contributors, we form a total of 165 subpopulations within the training dataset. Data contamination is simulated identically to the experiment of Section 5.4.2, while now we also consider the case of varying levels of contamination across the subpopulations. In particular, we form groups of roughly equal size where $0\%, 10\%$ and 20% of the datapoints get replaced by outliers—this results in getting a dataset with approximately 10% of its full set of datapoints corresponding to outliers.

We evaluate the predictive accuracy achieved by doing inference on the data subset obtained after running 10 iterations of the β-Cores extension for groups (which gives a maximum of 10 selected groups). We compare against (i) a random sampler, and (ii) a baseline which ranks all groups according to their Shapley value and selects the groups with the highest values. Shapley value is a concept originating in cooperative game theory [98], which has recently found applications in data valuation and outliers detection [43]. In the context of our experiment, it quantifies what is the marginal contribution of each group to the predictive accuracy of the model at all possible group coalitions that can be formed. As this quantity is notoriously expensive to be computed in large datasets, we use a Monte Carlo

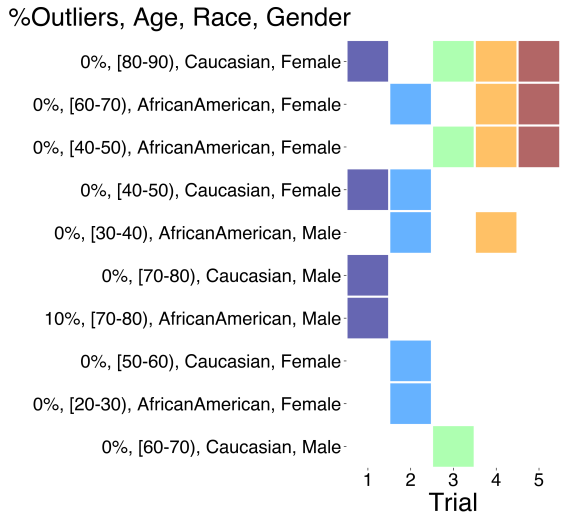


Figure 5.5: Attributes of selected groups after running 10 iterations of β-Cores with $\beta = 0.6$ on the contaminated HospitalReadmissions dataset (repeated over 5 random trials).

estimator which samples $5K$ possible permutations of groups and for each permutation it computes marginals for coalitions formed by the first 20 groups.[3]

As illustrated in Fig. 5.4, β-Cores with $\beta = 0.6$ offers the best solution to our problem, and is able to reach predictive accuracy exceeding 75% by fitting a coreset on no more than 2 groups. Fig. 5.5 displays the demographic information of selected groups. We can notice that subpopulations of female and older patients are more informative for the classification task, while Caucasian and African-American groups are preferred to smaller racial minorities. Importantly, β-Cores is able to distill clean from contaminated groups. For used β value we can see than over the set of trials only one group with outliers level of 10% is allowed to enter a summary, which already contains 3 uncontaminated groups.

Shapley values based ranking treats outliers better than random sampling: As outliers are expected to have negative marginal contribution to predictive accuracy, their Shapley rank is generally lower compared to clean data groups. On the other hand, Shapley computation is much slower than random sampling and β-Cores, specific to the evaluation metric of interest, while Shapley values are not designed to find data-efficient combinations of groups, hence this baseline can still return redundancy in the selected data subset.

5.5 Conclusion & further directions

In this work, we proposed a general purpose framework for yielding contamination-robust summarizations of massive scale datasets for inference. Relying on recent advances in Bayesian coresets and robustified inference under the β-divergence, we developed a greedy

[3]The latter truncation is supported by the observation that marginal contributions to the predictive accuracy are diminishing as the dataset size increases.

black-box construction that efficiently shrinks big data via keeping informative datapoints, while simultaneously rejecting outliers. Finally, we presented experiments involving various statistical models, and simulated and real-world datasets, demonstrating that our methodology outperforms existing techniques in scenarios of structured and unstructured data corruption.

Our future work will be concerned with considering stronger adversarial settings where summaries are initialized to data subsets that already contain outliers. Further directions also include automating the tuning of the robustness hyperparameter β, as well as applying our techniques to more complicated statistical models, including ones with structured likelihood functions (e.g. time-series and temporal point processes).

5.6 Models

In this section we present the derivations of β-likelihood terms Eqs. (5.6) and (5.7) required over the β-Cores constructions for the statistical models of our experiments.

5.6.1 Gaussian likelihoods

For the β-likelihood terms of a multivariate normal distribution, we have

$$\pi(x|\mu,\Sigma)^{\beta} = \left((2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}\right)^{\beta} \exp\left(-\frac{\beta}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right), \tag{5.18}$$

and, by simple calculus (see also [96]),

$$\int_{\mathcal{X}} \pi(\chi|\mu,\Sigma)^{1+\beta} d\chi = \left((2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}\right)^{\beta} (1+\beta)^{-\frac{d}{2}}. \tag{5.19}$$

Hence

$$f_n(\mu) \propto \frac{1}{\beta}\left((2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}\right)^{\beta} \exp\left(-\frac{\beta}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{5.20}$$

$$- \left((2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}\right)^{\beta} (1+\beta)^{-\frac{d}{2}-1} \tag{5.21}$$

$$\propto \frac{1}{\beta}\exp\left(-\frac{\beta}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) - (1+\beta)^{-\frac{d}{2}-1}. \tag{5.22}$$

5.6.2 Logistic regression likelihoods

Log-likelihood terms of individual datapoints are given as follows

$$\log \pi(y_n|x_n,\theta) = -\log\left(1+e^{-y_n z_n^T\theta}\right). \tag{5.23}$$

Substituting to Eq. (5.7), for the β-likelihood terms we get

$$f_n(\theta) \propto -\frac{1}{\beta}\left(1+e^{-y_n z_n^T\theta}\right)^{-\beta} \tag{5.24}$$

$$+\frac{1}{\beta+1}\left(\left(1+e^{-z_n^T\theta}\right)^{-(\beta+1)}+\left(1+e^{z_n^T\theta}\right)^{-(\beta+1)}\right). \tag{5.25}$$

5.6.3 Neural linear regression likelihoods and predictive posterior

Recall that in the neural linear regression model, $\left(y_n-\theta^T z(x_n)\right)\sim\mathcal{N}(0,\sigma^2),\ n=1,\ldots,N$. Then the Gaussian log-likelihoods corresponding to individual observations (after dropping normalization constants), are written as

$$f_n(\theta)=-\frac{1}{2\sigma^2}\left(y_n-\theta^T z(x_n)\right)^2. \tag{5.26}$$

Assuming a prior $\theta\sim\mathcal{N}(\mu_0,\sigma_0^2 I)$, the coreset posterior can be computed in closed form as follows

$$\pi_w(\theta)=\mathcal{N}\left(\mu_w,\Sigma_w\right), \tag{5.27}$$

where

$$\Sigma_w:=\left(\sigma_0^{-2}I+\sigma^{-2}\sum_{m=1}^{M}w_m z(x_m)z(x_m)^T\right)^{-1}, \tag{5.28}$$

$$\mu_w:=\Sigma_w\left(\sigma_0^{-2}I\mu_0+\sigma^{-2}\sum_{m=1}^{M}w_m y_m z(x_m)\right). \tag{5.29}$$

By substitution to Eq. (5.7), the β-likelihood terms for our adaptive basis linear regression are written as

$$f_n(\theta)\propto\frac{1}{(2\pi)^{\beta/2}\sigma^{\beta}}\left(-\frac{\beta+1}{\beta}e^{-\beta\left(y_n-\theta^T z(x_n)\right)^2/(2\sigma^2)}+\frac{1}{\sqrt{1+\beta}}\right). \tag{5.30}$$

Let $\mathcal{C}$ be the output of the coreset applied on a dataset $\mathcal{D}$. Hence, in regression problems, the predictive posterior on a test data pair (x_t,y_t) via a coreset is approximated as follows

$$\pi(y_t|x_t,\mathcal{D})\approx\pi(y_t|x_t,\mathcal{C}) \tag{5.31}$$

$$=\int\pi(y_t|x_t,\theta)\pi(\theta|\mathcal{C})d\theta. \tag{5.32}$$

In the neural linear experiment, the predictive posterior is a Gaussian given by the following formula

$$\pi(y_t|x_t,\mathcal{C})=\mathcal{N}\left(y_t;\mu_w^T z(x_t),\sigma^2+z(x_t)^T\Sigma_w z(x_t)\right). \tag{5.33}$$

Table 5.1: Logistic regression datasets

Dataset	d	N_{train}	N_{test}	#Pos. test data
ADULT [63]	10	30,162	7,413	3,700
PHISHING [28]	10	8,844	2,210	1,230
WEBSPAM [116]	127	126,185	13,789	6,907
HOSPITALREADMISSIONS [106]	10	55,163	6,079	3,044

Table 5.2: Neural linear regression datasets

Dataset	d	N_{train}	N_{test}
HOUSING [28]	13	446	50
SONGS [28]	90	463,711	51,534

5.7 Datasets Details

The benchmark datasets used in logistic regression (including group selection) and neural linear regression experiments are detailed in Tables 5.1 and 5.2 respectively.[4], and include:

- a dataset used to predict whether a citizen's income exceeds $50K\$$ per year extracted from USA 1994 census data (ADULT),
- a dataset containing webpages features and a label categorizing them as phishing or not (PHISHING),
- a corpus of webpages crawled from links found in spam emails (WEBSPAM),
- a set of hospitalization records for binary prediction of readmission pertaining to diabetes patients (HOSPITALREADMISSIONS),
- a set of various features from homes in the suburbs of Boston, Massachussets used to model housing price (HOUSING), and
- a dataset used to predict the release year of songs from associated audio features (SONGS).

For ADULT, PHISHING and HOSPITALREADMISSIONS we fit our statistical models on the first 10 principal components of the datasets, while all logistic regression benchmark datasets are evaluated on balanced subsets of the test data between the two classes (see Table 5.1).

[4]The original versions of all used datasets can be accessed by following the corresponding hyperlinks in the Tables appearing in the electronic version of the paper.

Chapter 6

Conclusions & Future Directions

Bibliography

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[2] P. K. Agarwal, S. Har-Peled, K. R. Varadarajan, et al. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

[3] C. C. Aggarwal and P. S. Yu. A general survey of privacy-preserving data mining models and algorithms. volume 34. 2008.

[4] B. Ağır, K. Huguenin, U. Hengartner, and J.-P. Hubaux. On the privacy implications of location semantics. *Proceedings on Privacy Enhancing Technologies*, 2016(4), 2016.

[5] R. Agrawal, T. Campbell, J. Huggins, and T. Broderick. Data-dependent compression of random features for large-scale kernel approximation. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[6] E. Angelino, M. J. Johnson, and R. P. Adams. Patterns of scalable bayesian inference. *Foundations and Trends in Machine Learning*, 9(2-3):119–247, 2016.

[7] O. Bachem, M. Lucic, and A. Krause. Coresets for nonparametric estimation—the case of DP-means. In *International Conference on Machine Learning*, 2015.

[8] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar. The security of machine learning. *Machine Learning*, 2010.

[9] A. Basu, I. R. Harris, N. L. Hjort, and M. C. Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 09 1998.

[10] J. O. Berger, E. Moreno, L. R. Pericchi, M. J. Bayarri, J. M. Bernardo, J. A. Cano, J. De la Horra, J. Martín, D. Ríos Insua, B. Betrò, et al. An overview of robust bayesian analysis. *Test*, 3(1):5–124, 1994.

[11] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*. Springer-Verlag, 1999.

[12] B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012.

[13] P. Bogdanov, M. Mongiovì, and A. K. Singh. Mining heavy subgraphs in time-evolving networks. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, 2011.

[14] K. M. Borgwardt and H.-P. Kriegel. Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, 2005.

[15] V. Braverman, D. Feldman, and H. Lang. New frameworks for offline and streaming coreset constructions. *arXiv:1612.00889*, 2016.

[16] T. Campbell and B. Beronov. Sparse variational inference: Bayesian coresets from scratch. In *Advances in Neural Information Processing Systems*, 2019.

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

[18] T. Campbell and T. Broderick. Automated scalable bayesian inference via hilbert coresets. *Journal of Machine Learning Research*, 20(15), 2019.

[19] Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010.

[20] A. Cichocki and S. Amari. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

[21] A. P. Dawid, M. Musio, and L. Ventura. Minimum scoring rule inference. *Scandinavian Journal of Statistics*, 43(1):123–138, 2016.

[22] B. de Finetti. The bayesian approach to the rejection of outliers. In *Proceedings of the fourth Berkeley Symposium on Probability and Statistics*, 1961.

[23] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the Crowd: The privacy bounds of human mobility. *Scientific reports*, 3, 2013.

[24] Y. De Mulder, G. Danezis, L. Batina, and B. Preneel. Identification via location-profiling in GSM networks. In *Proceedings of the 2008 ACM Workshop on Privacy in the Electronic Society*, 2008.

[25] I. Diakonikolas, G. Kamath, D. Kane, J. Li, J. Steinhardt, and A. Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[26] C. Dickens, E. Meissner, P. G. Moreno, and T. Diethe. Interpretable anomaly detection with Mondrian Pólya forests on data streams. *arXiv preprint*, 2020.

[27] P. Drineas and M. Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6, 2005.

[28] D. Dua and C. Graff. UCI machine learning repository, 2017.

[29] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[30] W. DuMouchel, C. Volinsky, T. Johnson, C. Cortes, and D. Pregibon. Squashing flat files flatter. In *ACM Conference on Knowledge Discovery and Data Mining*, 1999.

[31] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *International Conference on The Theory and Applications of Cryptographic Techniques*, 2006.

[32] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[33] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4), 2014.

[34] S. Eguchi and Y. Kano. Robustifying maximum likelihood estimation. *Tokyo Institute of Statistical Mathematics, Tokyo, Japan, Tech. Rep*, 2001.

[35] D. Feldman, M. Faulkner, and A. Krause. Scalable training of mixture models via coresets. In *Advances in Neural Information Processing Systems*, 2011.

[36] D. Feldman, A. Fiat, H. Kaplan, and K. Nissim. Private coresets. In *ACM Symposium on Theory of Computing*, 2009.

[37] D. Feldman, M. Volkov, and D. Rus. Dimensionality reduction of massive sparse datasets using coresets. In *Advances in Neural Information Processing Systems*, 2016.

[38] D. Feldman, C. Xiang, R. Zhu, and D. Rus. Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks. In *International Conference on Information Processing in Sensor Networks*, 2017.

[39] B. Frénay and M. Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5), 2013.

[40] H. Fujisawa and S. Eguchi. Robust parameter estimation with a small bias against heavy contamination. *J. Multivar. Anal.*, pages 2053–2081, 2008.

[41] F. Futami, I. Sato, and M. Sugiyama. Variational inference based on robust divergences. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, 2018.

[42] S. Gambs, M.-O. Killijian, and M. Núñez Del Prado Cortez. De-anonymization attack on geolocated data. *J. Comput. Syst. Sci.*, 80, 2014.

[43] A. Ghorbani and J. Zou. Data shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[44] A. Ghosh and A. Basu. Robust bayes estimation using the density power divergence. *Annals of the Institute of Statistical Mathematics*, 68(2):413–437, 2016.

[45] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *International Conference on Pervasive Computing*. Springer, 2009.

[46] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*. ACM, 2003.

[47] R. Guhaniyogi and D. Dunson. Bayesian compressed regression. *Journal of the American Statistical Association*, 110(512), 2015.

[48] D. Haussler. Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.

[49] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[50] M. D. Hoffman and A. Gelman. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

[51] P. J. Huber and E. M. Ronchetti. *Robust statistics; 2nd ed.* Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ, 2009.

[52] J. Huggins, R. Adams, and T. Broderick. PASS-GLM: polynomial approximate sufficient statistics for scalable Bayesian GLM inference. In *Advances in Neural Information Processing Systems*, 2017.

[53] J. Huggins, T. Campbell, and T. Broderick. Coresets for scalable Bayesian logistic regression. In *Advances in Neural Information Processing Systems*, 2016.

[54] J. Huggins, T. Campbell, M. Kasprzak, and T. Broderick. Validated variational inference via practical posterior error bounds. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[55] F. Huszár and D. Duvenaud. Optimally-weighted herding is Bayesian quadrature. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012.

[56] P. Jacob, J. O'Leary, and Y. Atchadé. Unbiased Markov chain Monte Carlo with couplings. *arXiv:1708.03625*, 2017.

[57] J. Jälkö, O. Dikmen, and A. Honkela. Differentially private variational inference for non-conjugate models. In *Uncertainty in Artificial Intelligence*, 2017.

[58] J. Jewson, J. Q. Smith, and C. Holmes. Principles of bayesian inference using general divergence criteria. *Entropy*, 20(6):442, 2018.

[59] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9, 2005.

[60] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, 2011.

[61] J. Knoblauch, J. E. Jewson, and T. Damoulas. Doubly robust bayesian inference for non-stationary streaming data with β-divergences. In *Advances in Neural Information Processing Systems 31*. 2018.

[62] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[63] R. Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[64] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(14), 2017.

[65] J. K. Laurila, D. Gatica-Perez, I. Aad, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen, et al. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*, 2012.

[66] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5:361–397, 2004.

[67] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in neural information processing systems*, 2016.

[68] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, IEEE 23rd International Conference on*, 2007.

[69] M. Lin, H. Cao, V. W. Zheng, K. C. Chang, and S. Krishnaswamy. Mobile user verification/identification using statistical mobility profile. In *2015 International Conference on Big Data and Smart Computing*, 2015.

[70] Q. Liu, J. Peng, and A. T. Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems 25*. 2012.

[71] M. Lucic, O. Bachem, and A. Krause. Strong coresets for hard and soft Bregman clustering with applications to exponential family mixtures. In *International Conference on Artificial Intelligence and Statistics*, 2016.

[72] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 2007.

[73] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[74] D. Madigan, N. Raghavan, and W. DuMouchel. Likelihood-based data squashing: a modeling approach to instance construction. *Data Mining and Knowledge Discovery*, 6, 2002.

[75] D. Manousakas, D. Xu, C. Mascolo, and T. Campbell. Bayesian pseudocoresets. *arXiv preprint*, 2020.

[76] B. D. McKay and A. Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94 – 112, 2014.

[77] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[78] J. W. Miller and D. B. Dunson. Robust bayesian inference via coarsening. *Journal of the American Statistical Association*, 114(527), 2019.

[79] S. Morse, M. C. Gonzalez, and N. Markuzon. Persistent cascades: Measuring fundamental communication structure in social networks. In *2016 IEEE International Conference on Big Data*, 2016.

[80] C. Musco and C. Musco. Recursive sampling for the Nyström method. In *Advances in Neural Information Processing Systems*, 2017.

[81] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli. Where you are is who you are: User identification by matching statistics. *IEEE Transactions on Information Forensics and Security*, 11(2), 2016.

[82] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 29th IEEE Symposium on*, 2008.

[83] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Security and Privacy, 30th IEEE Symposium on*, 2009.

[84] R. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov chain Monte Carlo*, chapter 5. CRC Press, 2011.

[85] L. Olejnik, C. Castelluccia, and A. Janc. On the uniqueness of web browsing history patterns. *Annales des Télécommunications*, 69, 2014.

[86] M. Park, J. R. Foulds, K. Chaudhuri, and M. Welling. Variational bayes in private settings (VIPS). *J. Artif. Intell. Res.*, 68, 2020.

[87] P. Paschou, J. Lewis, A. Javed, and P. Drineas. Ancestry informative markers for fine-scale individual assignment to worldwide populations. *Journal of Medical Genetics*, 2010.

[88] A. Pfitzmann and M. Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, 2010.

[89] R. Pinsler, J. Gordon, E. Nalisnick, and J. M. Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems*. 2019.

[90] A. Pyrgelis, C. Troncoso, and E. De Cristofaro. What does the crowd say about you? Evaluating aggregation-based location privacy. *arXiv preprint arXiv:1703.00366*, 2017.

[91] R. Ranganath, S. Gerrish, and D. Blei. Black Box Variational Inference. In *International Conference on Artificial Intelligence and Statistics*, 2014.

[92] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(4), 2010.

[93] D. Ríos Insua and F. Ruggeri. *Robust Bayesian Analysis*, volume 152. Springer Science & Business Media, 2012.

[94] C. Riquelme, G. Tucker, and J. Snoek. Deep Bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *6th International Conference on Learning Representations*, 2018.

[95] L. Rossi, M. J. Williams, C. Stich, and M. Musolesi. Privacy and the city: User identification and location semantics in location-based social networks. In *Proceedings of the Ninth International Conference on Web and Social Media*, 2015.

[96] W. Samek, D. Blythe, K.-R. Müller, and M. Kawanabe. Robust spatial filtering with beta divergence. In *Advances in Neural Information Processing Systems*, 2013.

[97] I. Scholtes. When is a network a network?: Multi-order graphical model selection in pathways and temporal networks. In *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining*, 2017.

[98] L. S. Shapley. A Value for n-Person Games. *Contributions to the Theory of Games*, 2(28), 1953.

[99] K. Sharad and G. Danezis. An automated social graph de-anonymization technique. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. ACM, 2014.

[100] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining*, 2008.

[101] N. Shervashidze, P. Schweitzer, V. Leeuwen, E. Jan, K. Mehlhorn, and K. Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

[102] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, pages 115–118. ACM, 2010.

[103] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, 2005.

[104] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams. Scalable bayesian optimization using deep neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[105] J. Steinhardt, P. W. W. Koh, and P. S. Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, 2017.

[106] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore. Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, 2014.

[107] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[108] M. Thoma, H. Cheng, A. Gretton, J. Han, H. P. Kriegel, A. Smola, L. Song, P. S. Yu, X. Yan, and K. M. Borgwardt. Discriminative frequent subgraph mining with optimality guarantees. *Statistical Analysis and Data Mining*, 3(5):302–318, 2010.

[109] M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, 2009.

[110] J. M. Tomczak and M. Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics*, 2018.

[111] S. Vahidian, B. Mirzasoleiman, and A. Cloninger. Coresets for estimating means and mean square error with limited greedy samples. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, 2020.

[112] S. Vishwanathan, N. Schraudolph, R. Kondor, and K. Borgwardt. Graph Kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.

[113] D. T. Wagner, A. Rice, and A. R. Beresford. *Device Analyzer: Understanding Smartphone Usage*, pages 195–208. Springer International Publishing, 2014.

[114] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, Jan. 2008.

[115] C. Wang and D. M. Blei. A general method for robust Bayesian modeling. *Bayesian Analysis*, 2018.

[116] D. Wang, D. Irani, and C. Pu. Evolutionary study of web spam: Webb Spam Corpus 2011 versus Webb Spam Corpus 2006. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2012.

[117] Y. Wang, A. Kucukelbir, and D. M. Blei. Robust probabilistic modeling with bayesian data reweighting. In *International Conference on Machine Learning*, 2017.

[118] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[119] B. Weisfeiler and A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

[120] P. Welke, I. Andone, K. Blaszkiewicz, and A. Markowetz. Differentiating smartphone users by app usage. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*.

[121] M. Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[122] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011.

[123] J. Whitehill, T.-F. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, 2009.

[124] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*, 2017.

[125] J. Xu, T. L. Wickramarathne, and N. V. Chawla. Representing higher-order dependencies in networks. *Science Advances*, 2(5), 2016.

[126] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*.

[127] P. Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM International Conference on Knowledge Discovery and Data Mining*, 2015.

[128] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi. Host fingerprinting and tracking on the web:privacy and security implications. In *The 19th Annual Network and Distributed System Security Symposium*. Internet Society, 2012.

[129] H. Zang and J. Bolot. Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*. ACM, 2011.

[130] A. Zellner. Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280, 1988.

[131] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *The Journal of Machine Learning Research*, 17(1):3537–3580, 2016.

[132] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on World Wide Web*, 2009.

[133] V. W. Zheng, S. J. Pan, Q. Yang, and J. J. Pan. Transferring multi-device localization models using latent multi-task learning. In *AAAI*, 2008.

[134] S. Zhou, J. Lafferty, and L. Wasserman. Compressed regression. In *Advances in Neural Information Processing Systems*, 2007.

[135] H. Zhuang, A. Parameswaran, D. Roth, and J. Han. Debiasing crowdsourced batches. In *Proceedings of the 21th ACM International Conference on Knowledge Discovery and Data Mining*, 2015.