Technical Report

UCAM-CL-TR-960
ISSN 1476-2986

Number 960

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Data summarizations for scalable, robust and privacy-aware learning in high dimensions

Dionysis Manousakas

September 2021

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted October 2020 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Darwin College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

The advent of large-scale datasets has offered unprecedented amounts of information for building statistically powerful machines, but, at the same time, also introduced a remarkable computational challenge: how can we efficiently process massive data? This thesis presents a suite of data reduction methods that make learning algorithms scale on large datasets, via extracting a succinct model-specific representation that summarizes the full data collection—a *coreset*. Our frameworks support by design datasets of arbitrary dimensionality, and can be used for general purpose Bayesian inference under real-world constraints, including privacy preservation and robustness to outliers, encompassing diverse uncertainty-aware data analysis tasks, such as density estimation, classification and regression.

We motivate the necessity for novel data reduction techniques in the first place by developing a reidentification attack on coarsened representations of private behavioural data. Analysing longitudinal records of human mobility, we detect privacy-revealing structural patterns, that remain preserved in reduced graph representations of individuals' information with manageable size. These unique patterns enable mounting linkage attacks via structural similarity computations on longitudinal mobility traces, revealing an overlooked, yet existing, privacy threat.

We then propose a scalable variational inference scheme for approximating posteriors on large datasets via learnable weighted *pseudodata*, termed pseudocoresets. We show that the use of pseudodata enables overcoming the constraints on minimum summary size for given approximation quality, that are imposed on all existing Bayesian coreset constructions due to data dimensionality. Moreover, it allows us to develop a scheme for pseudocoresets-based summarization that satisfies the standard framework of differential privacy by construction; in this way, we can release reduced size privacy-preserving representations for sensitive datasets that are amenable to arbitrary post-processing.

Subsequently, we consider summarizations for large-scale Bayesian inference in scenarios when observed datapoints depart from the statistical assumptions of our model. Using robust divergences, we develop a method for constructing coresets resilient to model misspecification. Crucially, this method is able to automatically discard outliers from the generated data summaries. Thus we deliver robustified scalable representations for inference, that are suitable for applications involving contaminated and unreliable data sources.

We demonstrate the performance of proposed summarization techniques on multiple parametric statistical models, and diverse simulated and real-world datasets, from

music genre features to hospital readmission records, considering a wide range of data dimensionalities.

Acknowledgements

This work contains the outcomes of prolonged research endeavours which would not have been accomplished without my interaction with many outstanding collegues. I express my deep gratitude to my supervisor, Cecilia Mascolo, for generously granting me vast stretches of freedom to pursue my own research direction, her unabated trust in my work, and her encouragement and advice over the ups and downs of my PhD. I'm also grateful to Trevor Campbell for his bold and intellectually stimulating collaboration: my perspective on science has been crucially enriched by his adherence to first principles and his enthusiasm on research. My close collaboration with Alastair Beresford was most valuable over the time I started delving into the subject of data privacy—also, Borja Balle's teaching on differential privacy has been instrumental in unveiling the wonders and beauty of the field. I'm indebted to my labmates at the Mobile Systems Group, Nokia Bell Labs, Max Planck Institute, and the rest of my collaborators: Shubham Aggarwal, Sourav Bhattacharya, Alberto Jesús Coca, Abir De, Manuel Gomez Rodriguez, Fahim Kawsar, Nic Lane, Akhil Mathur, Chulhong Min, Alberto Gil Ramos, and Rik Sarkar. My warm thanks to Michael Schaub and Kostas Kyriakopoulos: their mentorship over my first awkward steps in academic inquiry, provided me with solid foundations which were substantial over my PhD. Thanks to Matthias Grossglauser and Amanda Prorok for carefully examining this thesis, contributing their valuable suggestions and hosting an enjoyable viva.

I gratefully acknowledge the financial support received from Nokia Bell Labs, Lundgren Fund, Darwin College Cambridge, and the Department of Computer Science & Technology that backed my research.

Writings of Sebald, Bolaño, Benjamin, Parra, Herbert, activities at Darwin College Film Club, and university Canoeing and Tango societies kept me reinvigorating company over rest breaks. Antonis, Jenny and Panos made my time enjoyable supporting me with their friendship at all times—Pano, your light-spirited theory on information overflow has been a paradoxical source of inspiration over my research on coresets. Finally, my deepest thanks to Mum, Dad, Lena and Memos for keeping a shelter of unwavering love and support alive through all my successes and failures.

Table of contents

Chapter 1

Introduction

Machine learning pervades most modeling and decision-making tools of modern society: scientists rely on the wealth of stored medical records to decipher the underlying causes of diseases, web-scale recommender systems learn from users' experience to suggest music, movies, and products tailored to our habits, and driving-intelligence systems are capable to navigate self-driving cars in complex, never-seen-before environments.

From the statistical point of view, Bayesian modeling offers a powerful unifying framework where experts and practitioners alike can leverage domain-specific knowledge, learn from new observations, share statistical strength across components of hierarchical models, and take advantage of predictions which can account for model uncertainty. Having access to larger datasets is invaluable for statistical models, as it allows more insights into the process that gives rise to the data.

At the same time, handling massive-scale datasets in machine learning instigates a number of computational, societal, and statistical reliability challenges. First, beyond basic statistical settings, performing inference—i.e. computing expectations of interest under posterior distributions updated in the light of new observations—does not scale to large datasets; hence, learning in most interesting models requires additional effort from the data analyst to explore the statistical-computational trade-off of the problem, and turn to a suitable approximate inference method instead. Apart from addressing *scalability*, modern approximate inference methods should be also able to offer guarantees of convergence to the exact posterior distribution given sufficient computational resources, admit efficient quality measuring, and work seamlessly in high dimensions, where many of modern large-scale data live (e.g. genes, or social networks).

Secondly, a large fraction of modern massive-scale machine learning applications involves observations stemming from privacy-sensitive data domains, for example health records or behavioural studies. The sensitive information content of such sources makes crucial for data contributors that inference methods satisfy formal guarantees of *statistical privacy.* To this end, the gold standard is relying on the established framework of differential privacy: the existing toolset of privatising mechanisms and tight privacy loss estimation techniques, reinforced by the massive population sizes of modern datasets, allow statistically protecting individual information, yet extracting accurate insights about the population under study.

Thirdly, real-world big data are often highly heterogeneous, contain outliers and noise, or might be subject to data poisoning. The afore-mentioned phenomena are typically expressed as patterns which cannot be fully captured within the parametric assumptions of the statistical model. As a result, standard Bayesian inference techniques, which do not take extra care to downweight the contributions of outlying datapoints, lack *robustness* and, attempting to describe the full set of observations, might eventually yield unreliable posteriors.

How should we develop methods for large-scale data analysis that sufficiently address the problem of scalability, while formally preserving privacy and enhancing inferential results with robustness against mismatching observations? When faced with a dataset too large to be processed all at once, an obvious approach is to retain only a representative part of it. In this thesis, we build on the *data summarization* idea, which is validated by a critical insight in our massive-scale learning setup: when fitting a parametric probabilistic model on a large dataset, much of the data is redundant. Therefore, compressing the dataset under the strategic criterion of maximally reducing redundancy with respect to a given statistical model, opens an avenue for scalable data analysis without substantially sacrificing the accuracy of methods. The data summarization method of choice in this work is constructing *coresets*: small, weighted collections of points in the data space that can succintly and parsimoniously represent the complete dataset in a problem-dependent way.

Data Summarization and Differential Privacy. The aim of summarization is ostensibly in accord with the requirements of privacy, making it a good candidate to build privacy-preserving methods: informally, in both cases the target is to ensure encoding the prevailing patterns of the dataset, without revealing information about any individual datapoint in particular. However, an intricacy lies in that releasing part of the data, though perfectly acceptable for the purposes of coresets, directly breaches privacy, as it obviously exposes the full private information of the summarizing datapoints. Private coresets construction forms a challenging problem of releasing data in the *non-interactive*, or *offline* setting—namely in scenarios where a data owner aims to publicly release randomised privacy-preserving reductions of their data to third-parties, without knowing what statistics might be computed next. Differentially private schemes for coresets applicable in computational geometry already exist in the literature (Feldman et al., 2009; 2017). In the area of machine learning, the idea of releasing private dataset compressions via synthetic datapoints has been pursued in kernel mean embeddings (Balog et al., 2018) and compressive learning (Schellekens et al., 2019), with the utility of the private method scaling adversely with data dimension. Work limited to sparse regression (Zhou et al., 2007) has considered the high-dimensional data setting and proposed a method that compresses data via random linear or affine transformations. Nevertheless, none of these approaches is directly applicable to summarising for general-purpose Bayesian inference.

Data Summarization and Outliers Detection. Several approximate inference methods have proved brittle to observations that *"deviate markedly from other members of the sample"* (Grubbs, 1969). Outliers are a common complication emerging in real-world problems, attributed to limited precision, noise, uncertainty and adversarial behaviour often arising over data collection procedures. Since the pioneering work of Tukey (1960) and de Finetti (1961), discerning outliers has concerned the research community for over 60 years, shaping the area of robust statistics (Huber and Ronchetti, 2009). To this end, non-parametric distance-based techniques are a predominant approach that decouples outliers' detection from statistical assumptions regarding the data generating distribution, hence this paradigm has found broad applicability in machine learning and data mining. On the other hand, scaling distance computation to massive datasets is particularly resource intensive, while, further to computational intractability, distance-based analysis in high dimensions faces complications due to the *curse of dimensionality* (Donoho, 2000; Vershynin, 2018; Wainwright, 2019). Summarization has been leveraged for the purposes of outlier detection in non-probabilistic clustering in prior work by Lucic et al. (2016a). In the case of Bayesian learning, addressing inference on contaminated data via summarization critically relies on using as criterion of the coreset quality a robustified posterior, that is by definition insensitive to small deviations in the data space. Then the intuition used is that adding an outlier on a summary comprised of a majority of inliers will have an insignificant impact on the quality of the robust posterior defined on the summary points; hence, greedy incremental schemes of summarization can handily reject outlying observations while efficiently compressing the dataset.

1.1 Thesis statement and main contributions

The focus of this thesis is the development of scalable tools for data analysis on privacy-sensitive and vulnerable to contamination big data. We claim the following statement:

Automated methods for general-purpose probabilistic inference are typically computationally prohibitive in settings involving massive-scale and high-dimensional data. In contrast, designing principled dataset summarization algorithms enables scaling up learning methods in this data realm, achieving realiable inference results, and addressing concerns of privacy and robustness. Notably, the latter can be achieved without having a substantial bearing on the automation and complexity of the summarization methods.

Relying on coreset-based dataset summarizations as our fundamental framework for scalability, we adopt a two-pronged approach to tackle each of the aforementioned challenges, and design efficient algorithms that outperform state-of-the-art solutions for the posed problems.

In particular, the goals of this dissertation are to:

1. Identify threats in commonly adopted practices for releasing privacy-sensitive datasets via anonymized coarsened representations of the data.

2. Propose novel principled methods that can directly address real-word considerations of privacy and robustness when performing inference via summarization, without increasing the corresponding computational and memory footprint compared to the existing state-of-the-art methods.

The central contributions of the thesis are the following:

- We analyse the anonymity of individual data in a large-scale behavioural study, and develop a *reidentification attack that exploits structural patterns' similarity* to link users' records in the absence of identifiers in their state space.

- We introduce a novel variational formulation for Bayesian coresets construction that utilises approximations within a *family of efficient variational distributions* with learnable weights and locations of *pseudodata* as variational parameters. Leveraging the use of learnable pseudodata, we show that our variational formulation enables substantially more rapid improvement in summarization quality for high-dimensional data in the small coresets' regime, compared to existing coreset schemes that are constrained to use points from the original dataset. We provide an efficient black-box batch optimization scheme that can attain a good approximate posterior within the above mentioned variational family, and use standard randomization tools to yield differentially private versions of this variational posterior for privacy-preserving data analysis.

- We review Bayesian coresets' behaviour in corrupted datasets and show deficiencies of standard constructions when dealing with outliers and poisoning. Using tools from robust divergences, we propose approximate inference within a *robustified family of sparse variational approximations* for reliable summarization in the presence of data contamination. We develop a black-box incremental optimization scheme for constructing an approximation within this variational family, and evaluate its applicability in scenarios of summarization both over datapoints and over data minibatches.

A recurring theme in our approach is to exploit inherent data redundancy, in order to simultaneously achieve efficient data analysis and satisfy the objectives of privacy and robustness. Importantly, the computation of redundancy is adapted to the statistical model used to describe the data via the likelihood function, offering increased efficiency for the purposes of learning—as our methods, guided by the data likelihood function, manage to preserve reliable *approximate sufficient statistics* of the full data collection, despite retaining only a tiny fraction of it. Directly randomizing the sufficient statistics computation via a differentially private mechanism addresses formally the protection of privacy, and allows us to avoid adding more noise than necessary, as we only have to hide the part of individual datapoints' information which is passed to the sufficient statistics instead of their full information. On the robustness front, our framework identifies datapoints that deviate from our statistical assumptions and downweights their contribution over inference on the dataset, distilling them in this way from the extracted summary. Overall, our methods indicate that privacy and robustness on both counts are in accordance to the fundamental problem that data summarization aims to resolve: encapsulating aggregate information for a statistical model of interest, while limiting the impact of each individual datapoint's particulars.

1.2 Organization of the dissertation

The remainder of the dissertation is organized as follows.

Chapter 2 introduces relevant background and concepts used throughout the thesis.

Chapter 3 sheds light into the anonymity properties of a large-scale longitudinal mobility dataset, revealing a realistic privacy threat that survives in a release of sensitive structured data, despite anonymizing and coarsening individual behavioural records.

Chapter 4 presents a general-purpose variational inference algorithm that allows scaling up Bayesian inference in big and high-dimensional datasets via a coreset representation that relies on learnable synthetic datapoints (PSVI). Additionally, it develops a differentially private construction for this coreset (DP-PSVI).

Chapter 5 proposes a sparse variational approximation for robust generalized Bayesian posteriors using β-divergence, that can yield reliable summarizations for large-scale datasets in the presence of extensive contamination (β-Cores).

Finally, Chapter 6 concludes the thesis by summarizing our results and discussing future research directions.

This thesis covers material from the following publications:

> D. Manousakas, C. Mascolo, A. R. Beresford, D. Chan and N. Sharma (2018). "Quantifying privacy loss of human mobility graph topology". *Proceedings on Privacy Enhancing Technologies* 2018.3, pp. 5–21 (Chapter 3)
>
> D. Manousakas, Z. Xu, C. Mascolo and T. Campbell (2020). "Bayesian Pseudocoresets". *Advances in Neural Information Processing Systems* (Chapter 4)
>
> D. Manousakas and C. Mascolo (2021). "β-Cores: Robust Large-Scale Bayesian Data Summarization in the Presence of Outliers". *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (Chapter 5)

In addition, the following paper was written during my PhD but is not discussed in this thesis:

> S. Bhattacharya, D. Manousakas, A. G. C. Ramos, S. I. Venieris, N. D. Lane and C. Mascolo (2020). "Countering Acoustic Adversarial Attacks in Microphone-equipped Smart Home Devices". *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.2, pp. 1–24

Chapter 2

Background Material

This chapter aims to set the context for the remainder of this thesis. Various concepts pertaining to this thesis, including Bayesian inference, exponential family distributions and differential privacy, are briefly introduced in the following.

2.1 Comparing probability distributions

Throughout the thesis we focus primarily on probability spaces equipped with measures that are absolutely continuous w.r.t. some base measure, corresponding to the Lebesgue and counting measure respectively when considering continuous and discrete mappings from the sample space. This allows us to simplify notation and adapt the definitions presented in this section to normalised probability densities.

A critical component in constructing and evaluating inference algorithms is using a *divergence measure*, that captures informatively how similar two probability distributions are. Statistical divergences are relaxations of distance functions, that (*i*) are always non-negative, and (*ii*) equal zero iff their arguments are identical—albeit they do not necessarily satisfy symmetry in their arguments, or the triangle inequality, hence not having to be a metric by virtue of definition.

The most commonly used divergence measure in approximate inference—which will directly serve to define the objective quantifying the inferential quality of our sparse approximations in Chapters 4 and 5—is the *Kullback-Leibler* (KL) divergence, also named *relative entropy* (Kullback and Leibler, 1951; Kullback, 1959). For a continuous random variable θ and probability density functions π_1 and π_2, the KL divergence is defined as

$$\mathrm{D_{KL}}\left(\pi_1 || \pi_2\right) := \int \pi_1(\theta) \log \frac{\pi_1(\theta)}{\pi_2(\theta)} d\theta. \tag{2.1}$$

In particular, for two d-dimensional Gaussian distributions $\mathcal{N}_1(\mu_1, \Sigma_1)$ and $\mathcal{N}_2(\mu_2, \Sigma_2)$, the KL divergence is computable in closed form as follows

$$\mathrm{D_{KL}}\left(\pi_1 || \pi_2\right) = \frac{1}{2}\left[\log \frac{|\Sigma_2|}{|\Sigma_1|} - d + \mathrm{tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1)\right]. \tag{2.2}$$

Divergence	$\phi(\xi)$
Kullback-Leibler	$\xi \log \xi$
β-divergence	$\frac{1}{\beta(\beta+1)}\xi^{\beta+1} - \frac{1}{\beta}\xi + \frac{1}{\beta+1}, \quad \beta > 0$

Table 2.1: Convex functions used for reductions of relative entropy and density power to Bregman divergences on the domain of probability density functions.

In data setups that are likely to be contaminated by outliers, we get substantial inferential performance improvements when enhancing our algorithms with statistical *robustness*. Relying on the KL divergence cannot sufficiently address this concern, as this divergence attaches great importance to correctly capturing the tail behaviour of the observations. A robustified divergence, termed *β-divergence* or *density power divergence*, was instead proposed in (Basu et al., 1998; Eguchi and Kano, 2001), that is able to downweight outlying datapoints. Considering again the densities π_1, π_2, the β-divergence is defined as

$$\mathrm{D}_\beta\left(\pi_1 || \pi_2\right) := \frac{1}{\beta(\beta+1)} \int \left(\pi_1(\theta)^{1+\beta} - (\beta+1)\pi_1(\theta)\pi_2(\theta)^\beta + \beta\pi_2(\theta)^{1+\beta}\right) d\theta, \tag{2.3}$$

for $\beta \in \mathbb{R} \setminus \{-1, 0\}$.

One can easily show that the β-divergence converges to the KL divergence when $\beta \to 0$. Both divergences are asymmetric and do not satisfy the triangle inequality. Moreover, both divergences are instances of the family of Bregman divergences (Banerjee et al., 2005; Cichocki and Amari, 2010; Amari, 2016), i.e. a class of dissimilarity measures that can be expressed as $d_\phi(p, q) = \phi(p) - \phi(q) - \langle \nabla\phi(q), p - q\rangle$ using a strictly convex, differentiable function $\phi : \mathcal{K} \to \mathbb{R}$, for all p, q in a convex set $\mathcal{K} \subseteq \mathbb{R}^d$. In the case of two probability density functions π_1, π_2 the Bregman divergence admits the form $D_\phi(\pi_1, \pi_2) = \displaystyle\int \left[\phi(\pi_1(\theta)) - \phi(\pi_2(\theta)) - \phi'(\pi_2(\theta))(\pi_1(\theta) - \pi_2(\theta))\right] d\theta$. The convex functions defining the corresponding divergences are presented in Table 2.1.

2.2 Exponential families

The exponential family (Wainwright and Jordan, 2008) is a broad class of probability distributions, sharing a set of important properties that facilitate tractable inference. Exponential family members include numerous well-known distributions, such as the Poisson distribution, the Gamma distribution, and the Gaussian or normal distribution.

Definition 1 (Exponential family)**.** A collection of densities π, with respect to a base measure ν indexed by a vector of parameters θ, is an *exponential family* of densities if it can be written as

$$\pi_\theta(x) = h(x) \exp\left(\langle \theta, t(x)\rangle - Z(\theta)\right). \tag{2.4}$$

We call $t(x) : \mathcal{X} \to \mathbb{R}^d$ the *sufficient statistics* of the data, $h(x)$ the *base density* and

$$Z(\theta) := \log \int e^{\langle \theta, t(x) \rangle} h(x) \nu(dx) \tag{2.5}$$

the *log-partition function.*

The parameter space of interest, referred to as the *natural parameter space*, is the space $\Omega \subseteq \mathbb{R}^d$ that contains all θ such that $Z(\theta)$ is finite. We say that a family is *regular* if Ω is open.

An important property of exponential family densities is that the derivatives of the log-partition function Z are related to the moments of the sufficient statistics as follows.

Proposition 2 (Derivatives of the log-partition function via expected statistics)**.** *For a regular exponential family of densities in the form of Eq.* (2.4)*, the log-partition function has derivatives of all orders on its domain* Ω*, while for the first two derivatives hold the following*

$$\nabla Z(\theta) = \mathbb{E}_\theta[t(x)] \tag{2.6}$$

and

$$\nabla^2 Z(\theta) = \mathrm{Cov}_\theta[t(x)] := \mathbb{E}_\theta[t(x)t(x)^T] - \mathbb{E}_\theta[t(x)]\mathbb{E}_\theta[t(x)]^T. \tag{2.7}$$

Proposition 2 allows efficient approximations for the gradient and Hessian of Z using empirical estimates of the first two moments of the sufficient statistic; we take advantage of this property in the variational inference schemes to be introduced in Chapters 4 and 5.

2.3 Probabilistic learning at a glance

Bayesian probabilistic modeling provides a principled framework for learning from observed data, incorporating expert knowledge, handling model uncertainty and drawing coherent inferences in a unified way, following the languange of probability theory.

In (parametric) Bayesian learning settings we are generally given a set of observations $x = \{x_1, ..., x_N\} \subseteq \mathcal{X}$, and aim to find a vector of random variables θ parameterising an assumed probabilistic model that *is likely to* explain them. In the Bayesian paradigm, we first assume a *prior* distribution over the parameters $\pi_0(\theta)$, that encodes our beliefs about the uncertainty in θ before observing any data. Once the data are taken into account, our beliefs shoud be updated accordingly, in order to better describe the observed distribution. For this purpose a *likelihood* function $\pi(x|\theta)$ needs to be defined; the likelihood quantifies the probability of the observations under the assumed statistical model for parameters set to θ. Combining the above distributions we are ready to formulate *Bayes' theorem*, the fundamental rule which gives the *posterior* beliefs for our parameters updated in light of the observed data

$$\pi(\theta|x) = \frac{\pi(x|\theta)\pi_0(\theta)}{\pi(x)}. \tag{2.8}$$

Henceforth any quantity of interest $g(\cdot)$ involving the assumed probabilistic model is calculated using expectations under the posterior—which is considered to be the complete information about θ given the data x—as follows

$$\mathbb{E}_{\theta\sim\pi(\theta|x)}\left[g(\theta)\right] := \int g(\theta)\pi(\theta|x)d\theta. \tag{2.9}$$

Computing Eq. (2.9) is known as doing *inference* on our statistical model.

A key challenge in computing the posterior according to Eq. (2.8) is evaluating the normalizer, called *marginal likelihood* (or *model evidence*), which in a continuous parametric space takes the form

$$\pi(x) = \int \pi(x|\theta)\pi_0(\theta)d\theta. \tag{2.10}$$

Marginalising, i.e. computing the integral of Eq. (2.10), can be done using analytical tools for a number of simple Bayesian models—some of which will be discussed in the remainder, including Gaussian mean inference, Bayesian and neural linear regression—where the likelihood is conjugate to the prior. However, for the vast majority of interesting statistical models marginalization cannot be done in closed form and should be approximated instead. Aiming to address such cases, approximate Bayesian inference has emerged as an active research area for many decades. In the remainder of the section we present an overview of existing approaches addressing approximate inference that are relevant to our algorithms. For a more detailed exposure, including methods beyond the scope of this thesis (e.g. expectation propagation), cf. (Bishop, 2006; Murphy, 2012; Angelino et al., 2016).

2.3.1 Laplace's method

Point estimates of θ, obtained for example via *maximum a posteriori* or *maximum likelihood estimation*, are cheap to compute, as they correspond to solutions of optimization problems involving only the unnormalised RHS of Eq. (2.8)—on the other hand, they cannot capture the uncertainty of our posterior beliefs. Laplace's method (MacKay, 2003) is an approximate inference scheme that makes a first step towards uncertainty awareness, offering a non-degenerate, yet inexpensive to compute, approximate posterior for θ.

Let us write the posterior of Eq. (2.8) in the following equivalent form

$$\pi(\theta|x) = \frac{1}{Z}e^{-E(\theta)}, \tag{2.11}$$

where $E(\theta) := -\log\pi(\theta, x)$ is called the *energy function*, and Z is the unknown normalization constant. Taking the Taylor series expansion of θ (up to order 2) around the mode $\hat{\theta} := \arg\min_{\theta} E(\theta)$, we obtain the approximation $\hat{\pi}(\theta, x) := e^{-E(\hat{\theta})}\exp\Big((\theta-\hat{\theta})^T\Lambda(\theta-\hat{\theta})\Big)$

where $\Lambda := -\nabla^2 E(\theta)\Big|_{\theta=\hat{\theta}}$. Hence we have

$$\pi(\theta|X) \approx \frac{1}{Z}\hat{\pi}(\theta, x) \propto \mathcal{N}(\hat{\theta}, \Lambda^{-1}), \tag{2.12}$$

i.e. the posterior can be approximated by a (unimodal) Gaussian, where the mean corresponds to the minimum of the energy function and the covariance is the negative Hessian of the energy function evaluated on the mean. Clearly, using standard numerical optimization routines, e.g. quasi-Newton methods, we can achieve fast convergence to $\hat{\theta}$.

Laplace approximations will be used as coarse posterior approximations over our coreset summary constructions.

2.3.2 Sampling methods

In the absence of analytical formulae, integrals in the form of Eq. (2.9) can be approximated via empirical averaging, using samples from the target posterior distribution

$$\int g(\theta)\pi(\theta|x)d\theta \approx \frac{1}{S}\sum_{s=1}^{S} g(\theta_s), \quad (\theta_s)_{s=1}^{S} \overset{\text{i.i.d.}}{\sim} \pi(\theta|x). \tag{2.13}$$

Markov Chain Monte Carlo (MCMC), the workhorse of approximate Bayesian inference, is a framework of established tools that pursue the above idea efficiently (Geyer, 1992; Gilks, 2005; Robert and Casella, 2005).

MCMC offers approximations to expectations w.r.t. intractable probability distributions via simulating an ergodic random walk in the state space of the model, which admits the true posterior as its stationary distribution. As implied by the strong law of large numbers, the MC estimate—formed using (effectively independent) samples from the stationary distribution—converges to the true expectation almost surely as $s \to \infty$; this property makes MCMC methods theoretically appealing, as it endows the estimators with strong *asymptotic exactness* guarantees. Moreover, if g is a real function, using the central limit theorem, it can be shown that the standard error of a MC estimator scales asymptotically as $O(\frac{1}{\sqrt{S}})$, independently of the dimension of θ. Differing in the way that the Monte Carlo chain is constructed, as well as the offered level of automation, several methods of MCMC inference have emerged, including the Metropolis-Hastings (Andrieu et al., 2003), the Hamiltonian Monte Carlo (Neal, 2011), and the No-U-Turn-Sampler (NUTS) (Hoffman and Gelman, 2014). NUTS will be used as a reference method to evaluate summarization performance in part of our experiments over Chapters 4 and 5.

The computation of bounds on the number of MCMC iterations required until we obtain a satisfactory posterior approximation can hardly be automated, as they are highly problem-specific, and in practice heuristics are used to decide when sampling should stop. Typically each sample requires at least one evaluation of a function proportional to π, scaling at cost $\Theta(N)$ which becomes a burden in big data applications—on this account, methods operating on data subsets have been proposed, including (Welling and Teh, 2011; Bardenet et al., 2014; Korattikara et al., 2014). Despite these shortcomings, in settings where data are high-dimensional, and likelihood surface lacks structure that could be exploited over inference, MCMC remains the gold standard for practitioners.

2.3.3 Variational inference

Variational inference (VI) (Jordan et al., 1999; Blei et al., 2017) takes a fundamentally different approach to addressing approximate inference. The problem formulation underpinning all VI methods is to find a member q^* within a family of tractable probability densities Q that most closely matches our true posterior π (typically in the KL-sense)

$$q^*(\theta; x) := \arg\min_{q \in Q} \mathrm{D}_{\mathrm{KL}} \left(q(\theta) || \pi(\theta|x) \right). \tag{2.14}$$

In this way, Bayesian posterior inference gets reduced into an optimization problem; hence, techniques allowing scaling up optimization (e.g. random subsampling) can in principle be applied in VI methods, enabling scalable inference of approximate posteriors (Hoffman et al., 2013).

We note in passing that, in classical Variational Bayes schemes, expanding the KL divergence according to Eq. (2.1) makes the log-evidence appear in the objective

$$\mathrm{D}_{\mathrm{KL}} \left(q(\theta) || \pi(\theta|x) \right) = \mathbb{E}_{\theta \sim q} \left[\log q(\theta) \right] - \mathbb{E}_{\theta \sim q} \left[\log \pi(x, \theta) \right] + \log \pi(x). \tag{2.15}$$

Since this term is not a function of q, it can be subtracted and the problem is reformulated as minimizing the remaining two terms, the negation of which is known as the *evidence lower bound* (ELBO)

$$q^*(\theta; x) := \arg\min_{q \in Q} \left(-\mathrm{ELBO}(q, x) \right), \qquad \mathrm{ELBO}(q, x) := \mathbb{E}_q \left[\log \pi(x, \theta) \right] - \mathbb{E}_q \left[\log q(\theta) \right]. \tag{2.16}$$

Via Jensen's inequality, the ELBO can be shown to be a lower bound of the marginal log-likelihood of x as expectation w.r.t. q. As opposed to MCMC methods, theoretical guarantees for inferential results of the solution to Eq. (2.14) can only be obtained for a few simple statistical models for the following main reasons: optimization methods in typically non-convex landscapes can often converge to bad local optima; also, depending on the statistical divergence and variational family used, VI might return miscalibrated posterior variance estimates (Bishop, 2006, Chapter 10).

The simplest family Q that can be used for VI is the *mean-field variational family* which relies on the simplifying assumption of independence among the coordinates of θ, i.e. $q(\theta) := \Pi_{d=1}^{D} q_d(\theta_d)$. Our VI schemes in Chapters 4 and 5 propose approximations within the *exponential family* instead, which generally allow less restricted posteriors. Additionally, they can circumvent the use of ELBO, and instead be directly applied on the original KL minimizing variational formulation of Eq. (2.14), since MC estimates of the gradient of the intractable log-evidence term can be extracted as per Proposition 2.

2.3.4 Bayesian coresets

Owing to their requirement for multiple evaluations of the data (log-)likelihood—a computation scaling at $\Theta(N)$—MCMC and VI methods quickly become prohibitively expensive in the large-data regime. Various stochastic schemes have been proposed to circumvent this computation, evaluating the likelihood on random data minibatches: despite achieving computational savings and often being straightforward to implement, such schemes rarely offer guarantees on posterior approximation quality, and lack a rigorous principle over the minibatch selection step, hence retaining part of the redundancy of the full data collection in the extracted samples.

Bayesian coresets (Huggins et al., 2016; Campbell and Broderick, 2018; Campbell and Beronov, 2019; Campbell and Broderick, 2019; Zhang et al., 2021a) make the assumption that the full dataset has some degree of inherent redundancy, and put forth the idea of scaling up inference via the application of a preprocessing step where *part of the data gets retained under the criterion of likelihood approximation.* In the spirit of the first coresets proposed in the field of computational geometry (Feldman and Langberg, 2011), initial construction schemes for coreset-based inference (Huggins et al., 2016; Lucic et al., 2017) utilize *importance sampling* according to the datapoints' sensitivity, i.e. a non-negative quantity measuring the redundancy of each of the datapoints w.r.t. the statistical model of interest. Although providing theoretical guarantees for the approximation quality achieved by the coreset, importance sampling based constructions have typically two shortcomings: (i) they rely on efficiently computable upper bounds of the sensitivity, and (ii) they do not have a sense of a residual posterior error, hence are limited by common MC rates in approximating the full data likelihood, offering error $\epsilon = O(\frac{1}{\sqrt{M}})$ for coreset size M.

Reformulating coreset construction as sparse function approximation in a Hilbert space (*Hilbert coresets*), Campbell and Broderick (2018, 2019) introduced alternative optimization formulations for the problem. They showed that using inner-product inducing norms can lead to faster incremental construction schemes that, critically, can guide next datapoint selection by the direction of greatest impovement. Moreover, they made use of a coarse posterior approximation and random projections to efficiently compute Hilbert norms that capture the divergence between the coreset and the true posterior, and proposed faster sparse constructions under polytope and hypersphere constraints.

In more recent work, Campbell and Beronov (2019) casted Bayesian coresets to a problem of sparse variational inference within an exponential family, named *Riemannian coresets.* Riemannian coresets removed the requirement for fixing a coarse posterior that appears when computing the norm in practical Hilbert coreset constructions, achieving full automation and improvement of approximation quality (measured through the KL divergence) over a larger range of summary sizes.

2.4 Robust inference

In this section, adopting an optimization perspective of Bayesian inference, we present robustness limitations of the standard Bayesian posterior on big data, and outline existing generalizations of the posterior that aim to robustify inference w.r.t. mismatches between observed data and modelling assumptions. Setting these robustified posteriors as the target of our coreset approximations, in Chapter 5 we will successfully address scenarios of large-scale inference under model misspecification.

2.4.1 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 $x := (x_n)_{n=1}^N \in \mathcal{X}^N$. Here we equivalently rewrite Eq. (2.8) as

$$\pi(\theta|x) = \frac{1}{Z'}\pi(x|\theta)\pi_0(\theta), \tag{2.17}$$

where Z' is a normalization constant corresponding to the (typically intractable) marginal likelihood term $\pi(x)$. When the datapoints x are conditionally independent given θ, the likelihood function 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 a convex optimization problem over the density space was introduced by Williams (1980) and Zellner (1988), and used in various subsequent works including (Zhu et al., 2014; Dai et al., 2016; Futami et al., 2018). Concretely, Eq. (2.17) can be recovered by solving the problem

$$\arg\min_{q(\theta)\in\mathcal{P}} \left(\mathrm{D_{KL}}\left(q(\theta)||\pi_0(\theta)\right) - \sum_{n=1}^{N}\left[\int q(\theta)\log\pi(x_n|\theta)d\theta\right]\right), \tag{2.18}$$

where $\mathcal{P}$ is the valid density space, while the Bayesian posterior can be expressed as

$$\pi(\theta|x) = \frac{1}{Z'}\exp\left(-\mathrm{d_{KL}}\left(\hat{\pi}(x)||\pi(x|\theta)\right)\right)\pi_0(\theta). \tag{2.19}$$

In the last expression, $\hat{\pi}(x) := \frac{1}{N}\sum_{n=1}^N \delta(x - x_n)$ is the empirical distribution of the observed datapoints and δ is the Dirac delta function. The exponent $\mathrm{d_{KL}}\left(\hat{\pi}(x)||\pi(x|\theta)\right) := -\sum_{n=1}^N \log\pi(x_n|\theta)$ 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 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 admit densities π_θ and π_{obs} respectively, we can rewrite an approximation of Eq. (2.19) via the KL divergence as in (Miller and Dunson, 2019)

$$\begin{aligned}\pi(\theta|x) \propto \exp\left(\sum_{n=1}^{N} \log \pi(x_n|\theta)\right) \pi_0(\theta) &\doteq \exp\left(N \int \pi_{\text{obs}} \log \pi_\theta\right) \pi_0(\theta) \\ &:= \exp\left(-N \mathrm{D}_{\mathrm{KL}}\left(\pi_{\text{obs}}||\pi_\theta\right)\right) \pi_0(\theta), \end{aligned} \tag{2.20}$$

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

2.4.2 Robustified generalized Bayesian posteriors

Robust inference methods aim to adapt Eqs. (2.17) to (2.19) 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 (Eguchi and Kano, 2001; Fujisawa and Eguchi, 2008; Dawid et al., 2016; Jewson et al., 2018), Bayesian updates rely on utilising robust divergences instead of the KL divergence, to express the losses over the observed data.

From the definition of KL divergence Eq. (2.1), we can equivalently rewrite Eq. (2.18) as

$$\arg \min_{q(\theta) \in \mathcal{P}} \Big(\mathrm{D}_{\mathrm{KL}}\left(q(\theta)||\pi_0(\theta)\right) + N \mathbb{E}_{q(\theta)}\left[\mathrm{D}_{\mathrm{KL}}\left(\hat{\pi}(x)||\pi(x|\theta)\right)\right]\Big), \tag{2.21}$$

namely inference corresponds to maximizing the expected likelihood of the observations, under a regularizer that aims to keep the posterior q close to the prior π_0. As mentioned in Section 2.1, a popular choice for enhancing inferential robustness is to replace the KL divergence—computed via the expected likelihood arising in the second term of Eq. (2.21)—with the *β-divergence* (Futami et al., 2018; Knoblauch et al., 2018). This yields the following posterior for θ (Ghosh and Basu, 2016; Knoblauch et al., 2018)

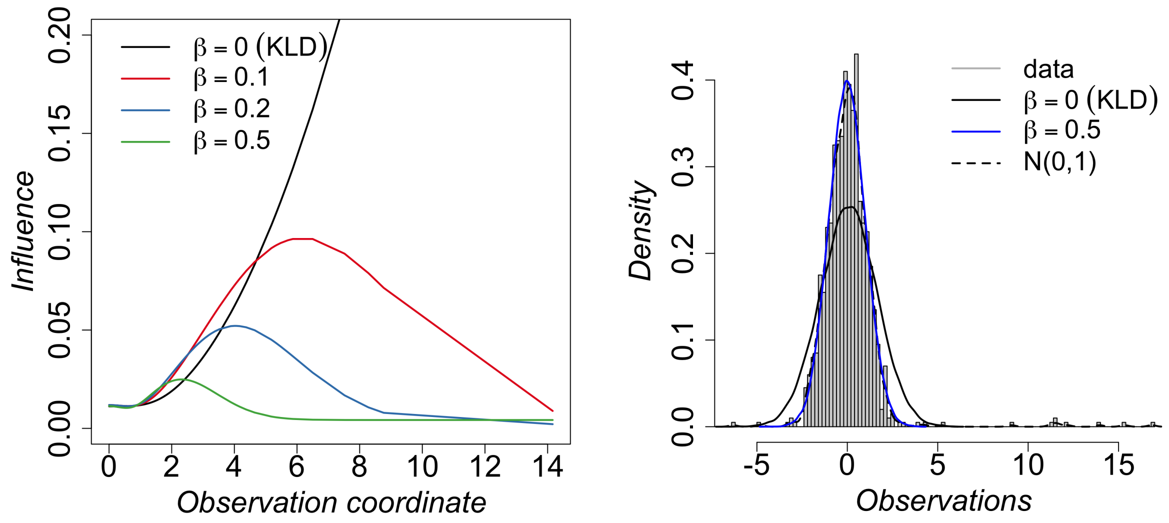
$$\pi_\beta(\theta|x) \propto \exp\Big(-\mathrm{d}_\beta\left(\hat{\pi}(x)||\pi(x|\theta)\right)\Big) \pi_0(\theta), \tag{2.22}$$

where

$$\mathrm{d}_\beta\left(\hat{\pi}(x)||\pi(x|\theta)\right) := \sum_{n=1}^{N} \underbrace{\left(-\frac{\beta+1}{\beta} \pi(x_n|\theta)^\beta + \int_{\mathcal{X}} \pi(\chi|\theta)^{1+\beta} d\chi\right)}_{:=f_n(\theta)}, \tag{2.23}$$

with $\beta > 0$. In the remainder of the thesis we refer to quantities defined in Eqs. (2.22) and (2.23) 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.

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



(a) (b)

Figure 2.1: Effects of altering the statistical divergence when conducting inference on datasets containing outliers. (a) Influence of individual datapoints under the Kullback-Leibler and the β-divergence: the concavity of influence under the β-divergence illustrates the robustness of the inferred posterior to outliers. (b) Posterior estimates of Gaussian density on observations containing a small fraction for outliers under classical and robustified inference.

When $\beta \to 0$, the Bayes' posterior of Eqs. (2.17) and (2.19) 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 respective expression for individual terms of the β-likelihood[2] is (Basu et al., 1998)

$$f_n(\theta) := -\frac{\beta+1}{\beta}\pi(y_n|x_n,\theta)^\beta + \int_{\mathcal{Y}} \pi(\psi|x_n,\theta)^{1+\beta} d\psi. \tag{2.24}$$

Illustrations In the remainder of this section we illustrate the effects of adapting the used statistical divergence when doing inference on a dataset that contains outliers. In a similar vein to (Jewson et al., 2018), we juxtapose the inference results of classical and robust posterior on simple statistical models aiming to fit a Gaussian probability distribution of unknown mean and variance $\mathcal{N}(\mu, \sigma)$ to one-dimensional observations.

Fig. 2.1a demonstrates the *influence* of individual observations with varying magnitude on the inferred posterior. The influence is measured using the Fisher–Rao metric introduced in (Kurtek and Bharath, 2015). For this experiment, $10K$ observations were sampled from a Student $t(3)$ distribution, while observations with negative coordinates were omitted from the presented plot due to symmetry. We can notice that the KL divergence allows unbounded influence, indicating the brittleness of inference on the tails of the observed distribution. In contrast, moving away from the mean, individual datapoints' influence

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

under the β-divergences is initially characterised by a regime of increase until reaching a maximum (which depends on the selected robustness hyperparameter), succeeded by attenuation down to zero at the tails of the data distribution. At the same time, this experiment makes clear that for decision problems critically relying on the tail information of the observations, KL might be the divergence of choice, as the density power divergence would downweight the importance of datapoints lying far from the mean.

Fig. 2.1b shows the posterior density estimation for classical and robustified Bayesian inference on $1K$ datapoints sampled from a contaminated distribution $0.99 \times \mathcal{N}(0,1) + 0.01 \times \mathcal{N}(5,25)$. The posterior under the KL divergence tries to explain the long tails of the observations—which are the effects of the contaminating component—eventually overestimating the variance of the data distribution. On the other hand, using the density power divergence with $\beta = 0.5$ over inference allows us to declare the long tails as outliers, and provides more accurate modeling of the inliers' component.

2.5 Representing data

Extracting a relevant feature representation is an important step in the context of statistical pattern recognition. For this purpose a feature map

$$\phi : \mathcal{X} \to \mathcal{H}, \tag{2.25}$$

is sought which transforms the datapoints from the original data space $\{x_n\}_{n=1}^N$, $x_n \in \mathcal{X}$, into *feature representations* in a Hilbert space $\{\phi(x_n)\}_{n=1}^N$, $\phi(x_n) \in \mathcal{H}$. Then the patterns of interest can be revealed via applications of inner products in the Hilbert space $\langle A, \phi(x) \rangle_{\mathcal{H}}$. There is an extensive literature on constructing data representations; for the purposes of this thesis, in the remainder of the section we focus on two of them: kernel methods and random projections.

2.5.1 Kernels

The main tool in kernel methods (Schölkopf et al., 2002) is the *kernel function* defined below.

Definition 3 (Kernel function)**.** A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive semidefinite *kernel function*, or *kernel*, if for all $N > 1$, $x_1, \ldots, x_N \in \mathbb{R}$, and $c_1, \ldots, c_N \in \mathbb{R}$

$$\sum_{i,j=1}^{N} c_i c_j k(x_i, x_j) \geq 0. \tag{2.26}$$

Every kernel is associated with a feature map ϕ as follows.

Definition 4 (Kernel representation)**.** A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel iff there exists a Hilbert space $\mathcal{H}$ and a feature map $\phi : \mathcal{X} \to \mathcal{H}$ such that for all $x, x' \in \mathcal{X}$

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}. \tag{2.27}$$

Feature map ϕ endows each datapoint $x \in \mathcal{X}$ with a kernel representation $\phi(x)$.

A kernel representation might be lacking an explicit closed form, but can always be accessed via the inner product of Eq. (2.27), which is the central object of interest in learning with kernels.

Examples of widely-used kernel functions include:

- The (inhomogeneous) polynomial kernel $k(x, x') = (\langle x, x' \rangle + c)^d$, where $c \geq 0, d \in \mathbb{N}$.
- The Gaussian kernel $k(x, x') = \exp(-\gamma ||x - x'||_2^2)$.
- Radial Basis Function (RBF) kernels $k(x, x') = f(d(x, x'))$, where d is a metric on $\mathcal{X}$ and f is a function on $\mathbb{R}^+$.

Kernel methods induce *non-parametric* representations on the data, i.e. when given a set with N datapoints of dimension d, kernels effectively map each datapoint to an N-dimensional representation.

2.5.2 Finite-dimensional random projections

Kernel methods appeal to large-scale learning due to their non-parametric nature: their representation power scales with the number of datapoints, hence they can learn complex, highly non-linear structure from the data; however, their time and memory cost scales adversely with the dataset size. Random features (Rahimi and Recht, 2008) remedy poor complexity scaling issues via utilising *parametric finite-dimensional* data representations. We motivate this concept via an application arising in Hilbert coreset constructions (Campbell and Broderick, 2019).

Denote by $f_n(\theta) := \sum_{n=1}^N \log \pi(x_n|\theta)$ the log-likelihood function of a dataset $x := (x_n)_{n=1}^N$, and by $f(\theta, w) := \sum_{n=1}^N w_n \log \pi(x_n|\theta)$ the corresponding log-likelihood of a Hilbert coreset $(w_n, x_n)_{n=1}^N$ constructed on the data, where $(w_n)_{n=1}^N$ is a vector of sparse, non-negative weights—using the simplified notation $f(\theta)$ for the full data log-likelihood. The quality of posterior approximation that this coreset offers can be quantified using an L^2 norm on the log-likelihoods under a weighting distribution $\hat{\pi}$ that has the same support with the true posterior π

$$||f(\theta, w) - f(\theta)||_{\hat{\pi},2} := \mathbb{E}_{\hat{\pi}} \left[(f(\theta) - f(\theta, w))^2 \right], \tag{2.28}$$

and induced inner product

$$\langle f_n(\theta), f_m(\theta) \rangle_{\hat{\pi},2} := \mathbb{E}_{\hat{\pi}} \left[f_n(\theta), f_m(\theta) \right]. \tag{2.29}$$

The weighting distribution $\hat{\pi}$ can be selected from a set of cheap posterior approximations, for example using Laplace's method, or running a few rounds of an MCMC algorithm. In the general case, the norm of Eq. (2.29) is not available in closed form, hence a random projection can be used instead to approximate it according to the following steps:

1. Sample J values for θ from the weighting distribution $(\hat{\theta}_j)_{j=1}^J \overset{\text{i.i.d.}}{\sim} \hat{\pi}$.

2. For $n = 1 \dots N$ compute a J-dimensional projection $\hat{f}_n(\theta) := \sqrt{\frac{1}{J}}[f_n(\hat{\theta}_1) \dots f_n(\hat{\theta}_J)]$.

In this way we get an unbiased finite-dimensional estimator of the inner products

$$\langle f_n(\theta), f_m(\theta) \rangle_{\hat{\pi},2} \approx \hat{f}_n(\theta)^T \hat{f}_m(\theta). \tag{2.30}$$

2.6 Differential privacy

Differential privacy (DP) (Dwork et al., 2006c; Dwork and Roth, 2014) is a formal framework quantifying the privacy threat that exists in observing the output of a data analysis task carried out on a sensitive database, due to changing an individual entry of its input. The central model of DP considers a setting where the database is held by a trusted curator; and an untrusted analyst sends statistical queries to the curator and receives public responses via randomized algorithms, or *mechanisms*: DP enforces a stability property on the output distribution of these mechanisms that limits the disclosure of information about any individual record within the database, offering strong indistinguishability guarantees regardless of the side information that the analyst might possess (even when the analyst knows all other records of the database).

DP definition requires a notion of *neighboring* databases. To define distance between two databases $x, x' \in \mathcal{X}$ of size N we use the Hamming distance

$$D_H(x, x') := \#\{n = 1, \dots, N : x_n \neq x'_n\}. \tag{2.31}$$

We call the databases adjacent, denoted $x \approx x'$, iff $D_H(x, x') = 1$.

Definition 5 (Differential Privacy)**.** Fix $\varepsilon \geq, \delta \geq 0$. A mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is (ε, δ)-differentially private if for all adjacent datasets $x \approx x'$ and each event $A \subseteq \mathcal{Y}$, $\mathbb{P}[\mathcal{M}(x) \in A] \leq e^{\varepsilon} \mathbb{P}[\mathcal{M}(x') \in A] + \delta$.

Definition 5 with $\delta = 0$, known as *pure DP*, requires that if we perturb a database by a single datapoint, the output of the algorithm should not differ much, with the privacy risk being controlled by the parameter ε. A weaker definition of DP allows that the guarantee of Definition 5 gets broken with probability $\delta > 0$. This corresponds to the notion of *(ε, δ)-approximate differential privacy.* The latter generally allows more tools for tighter privacy analysis over repeated access to the data, and will be the definition applied on our privacy-preserving summarization scheme in Chapter 4. In practice, $\varepsilon \leq 0.1$ and $\delta \approx 1/N^{\omega(1)}$ are typically considered good values for the privacy parameters.

The most common mechanisms that enable releasing numerical queries f under DP rely on randomization via injecting additive noise. The amount of noise is calibrated to the *global sensitivity* of the query, which is defined as

$$\Delta_p(f) := \max_{x \approx x'} ||f(x) - f(x')||_p. \tag{2.32}$$

To achieve (ε, δ)-DP one can use the Gaussian Mechanism, which returns

$$f(x) + Z, \quad Z \sim \mathcal{N}(0, \sigma^2 I), \quad \text{where } \sigma \geq \frac{\sqrt{2\log(1.25/\delta)}\,\Delta_2(f)}{\varepsilon}. \tag{2.33}$$

DP is equipped with a suite of properties that facilitate reasoning about privacy guarantees over complicated analysis tasks on a sensitive data collection in a modular fashion. In the remainder we review a fraction of them which are frequently encountered in machine learning settings.

A useful fact about DP algorithms is that a data analyst cannot weaken their privacy guarantees by doing any computation on their output that does not depend on the private input itself.

Proposition 6 (Robustness to Post-Processing (Dwork and Roth, 2014))**.** *Let $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ be (ε, δ)-DP and $\psi : \mathcal{Y} \to \mathcal{Y}'$ be any function. Then $\psi \circ \mathcal{M} : \mathcal{X} \to \mathcal{Y}'$ is (ε, δ)-DP.*

Moreover, running a mechanism on a random subset of the datapoints implies stronger privacy compared to running the mechanism on the full database.

Proposition 7 (Privacy Amplfication via Random Sampling (Kasiviswanathan et al., 2011; Beimel et al., 2013))**.** *Let $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ be (ε, δ)-DP with $\varepsilon \leq 1$ and $\upsilon : \mathcal{X} \to \mathcal{X}$, a random sampler returning a random ratio q of the datapoints. Then $\mathcal{M} \circ \upsilon : \mathcal{X} \to \mathcal{Y}$ is $(O(q\varepsilon), q\delta)$-DP.*

DP composition theorems accumulate the total privacy cost over the application of a sequence of mechanisms. The moments accountant is a recently proposed technique, that allows computing tight bounds for ε and δ, offering the following guarantees:

Proposition 8 (Moments Accountant (Abadi et al., 2016))**.** *Given $0 < \varepsilon < 1$ and $0 < \delta < 1$, to ensure $(\varepsilon, T\delta' + \delta)$-DP over the composition of T mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_T$, it suffices that each $\mathcal{M}_i$ is (ε', δ')-DP, where $\varepsilon' = \frac{\varepsilon}{2\sqrt{2T\log(2/\delta)}}$ and $\delta' = \frac{\delta}{T}$.*

The above tools are required for carrying out the privacy analysis of the *subsampled Gaussian mechanism* (Abadi et al., 2016), which will be used for privatising the variational inference scheme introduced in Chapter 4.

Chapter 3

Quantifying Privacy Loss of Human Mobility Graph Topology

In this chapter, we present a case study on population scale empirical data, which demonstrates that releases of deidentified and reduced representations of structured individual records might still breach the privacy of information-contributing participants. This analysis motivates the necessity of developing new formal privacy-preserving frameworks for scalable learning via data summarization, which is further studied in Chapter 4.

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 chapter 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 Motivation & contributions

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 (Zang and Bolot, 2011; de Montjoye et al., 2013).

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 dataset released for the research track of the Nokia Mobile Data Challenge,[1] where visited places were represented by random integers (Laurila et al., 2012); 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 (Wagner et al., 2014). Moreover, temporal resolution may also be deliberately decreased to improve anonymization (Gruteser and Grunwald, 2003), since previous work has demonstrated that sparsity in the temporal evolution of mobility can cause privacy breaches (de Montjoye et al., 2013).

In this chapter, *we examine the degree to which reduced representations of 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 to what extent these graphs reflect user-specific behavioural attributes that could act as a fingerprint, perhaps allowing the reidentification of the individual they represent. In particular, we show how graph kernel distance functions (Vishwanathan et al., 2010) 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 effectively 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 (De Mulder et al., 2008; Golle and Partridge, 2009; Gambs et al., 2014; Naini et al., 2016), 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 and coarse duration of stays.*

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

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

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 examined data reductions contain 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 indistinguishably cluster users into larger groups with similar mobility; such an approach may then be able to offer better anonymity guarantees.

A summary of the contributions of this chapter is as follows:

- 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 in state spaces 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 (Aggarwal and Yu, 2008), and hence mobility data are often vulnerable to deanonymization attacks (Narayanan and Shmatikov, 2008). 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 and Bolot (2011) analysed nationwide call-data records (*CDR*s) and showed that releasing the N most frequently visited places—so called *top*$-N$ data—correlated with publicly released side information, resulted in privacy risks, even for small values of Ns. This finding underlines the need for reductions in spatial or temporal data fidelity before publication. Further, de Montjoye et al. (2013) 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. They found that four random spatio-temporal points suffice to uniquely identify 95% of the traces. They also observed 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. De Mulder et al. (2008) developed a reidentification technique by building a Markov model for each individual in the training set, and then using this to reidentify individuals in the test set by likelihood maximisation. Similarly, Gambs et al. (2014) used Markov chains to model mobility traces in support of reidentification.

Naini et al. (2016) explored the privacy impact of releasing statistics of individuals' mobility traces in the form of histograms, instead of their actual location information. They demonstrated that even this statistical information suffices to successfully recover the identity of individuals in datasets of few hundred people, via matching labeled and unlabeled histograms of a population. Other researchers have investigated the privacy threats stemming from information sharing on location-based social networks, including the impact of location semantics on the difficulty of reidentification (Rossi et al., 2015) and location inference (Ağır et al., 2016).

All the above-mentioned previous work assumes that locations are expressed using a universal set of symbols or global identifiers, 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 anonymized separately for each individual. Lin et al. (2015) presented a user verification method in this setting. It is based on statistical profiles of individual indoor and outdoor mobility, including cell tower ID and WiFi access point information. In contrast, here 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 principles (Sweeney, 2002). However, recent results have questioned this assumption. Xu et al. (2017) 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. Similarly, Pyrgelis et al. (2017) performed a set of inference attacks on aggregate location time-series data and detected serious privacy loss, even when individual data are perturbed by differentially private mechanisms before aggregation.

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 (Narayanan and Shmatikov, 2009; Sharad and Danezis, 2014), or quantify the leakage of private information of a graph node via its neighbors (Zheleva and Getoor, 2009).

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 set 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 (Dwork et al., 2006c), are not directly applicable. However, metrics related to structural similiarity, including $k-$anonymity, can be seamlessly generalized in this framework.

3.2.3 Approximate graph matching

The problem of matching graphs (or networks) according to their structural similarity has emerged in research under disparate contexts and treatments. To clearly position our formulation in the related literature, we first draw a distinction between two primary instantiations of the problem: (*i*) *Graph matching* (or *graph alignment*) is the problem of finding a bijection of node sets across graphs, that typically correspond to distorted versions of the same underlying graph. (*ii*) *Graph set matching* (or *graph comparison*) is the problem of uncovering members corresponding to the same entity across two graph datasets that are assumed to form two distorted subsets of the same population of underlying graphs. The data linkability question considered in the context of our work is an instance of the latter problem.

Exact graph matching is equivalent to the problem of graph isomorphism, which admits no known polynomial algorithm (although is broadly conjectured not to belong to the family of NP-Hard problems (Schöning, 1988)). Approximate network alignment admits different solutions, depending on the given information about the graph (e.g. whether the graph nodes are labeled, or whether alignment for a subset of nodes is known). Kazemi et al. (2015) proposed a percolation-based algorithm that, leveraging a partially correct seed of node matches, can rapidly expand it to larger matching sets. Pedarsani et al. (2013) used a seedless Bayesian approach assuming a distortion model which describes how observations were obtained from the original graph. Via introducing additional heuristic functions on the results of alignment, graph alignment methods can produce distances applicable to graph comparison—for instance, Mishinev (2020) proposed a normalized edge overlap metric that allowed transforming the previous two methods into a network distance function.

Graph set matching can be approached via computations of a domain specific similarity metric applicable on graphs, that attributes large values to similarly looking graphs and small values to graphs that look dissimilar. For the purposes of supervised learning or data linkage, this metric can be relaxed to not strictly satisfy the mathematical definition of a distance metric, e.g. not obey the triangle inequality, as long as it reasonably captures a quantification of structural similarity. In a recent work, Chowdhury and Mémoli (2019) expanded the machinery of optimal transport to the problem of graph set matching: endowing graphs with probability measures, allowed them to define a pseudometric on the space of directed, weighted networks using an efficiently computable approximation of the optimal transportation function between graphs. Alternative long-standing approaches to graph comparison are based on network motifs and frequent subgraph mining methods (Milo et al., 2002; Yan and Han, 2002), which unfortunately have worst-case complexity scaling exponentially with graph size. Graph kernels (Vishwanathan et al., 2010), which will be the toolbox used in our graph comparison problem, achieve an efficient compromise, as they are restricted to measure similarity using graph substructures which are computable in polynomial time. Especially in the bioinformatics literature, graphlets (Pržulj, 2007; Shervashidze et al., 2009), i.e. small connected non-isomorphic graphs, are commonly selected as the substructures of choice, as they enable reasonable representation of the local structure in *unlabeled* networks. As graphlet kernels do not support labeled nodes and scale polynomially with the degree of the nodes, in Section 3.5 we focus our experimentation on kernels capturing shortest-path and subgraph isomorphism information.

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 that 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 (Yen et al., 2012; Olejnik et al., 2014) or smartphone application usage sequences (Welke et al., 2016).

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 (Sweeney, 2002), which we summarize as follows:

Definition 9 (**$k-$anonymity**)**.** A microdata release of statistics, containing separate entries for a number of individuals in the population, satisfies the $k-$anonymity property, iff 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 chapter 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 (Shokri et al., 2010), motivating the definition of $l-$diversity (Machanavajjhala et al., 2007) and $t-$closeness (Li et al., 2007). 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 10 (**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 11 (**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* (Pfitzmann and Hansen, 2010) as follows:

Definition 12 (**Identifiability Set**)**.** *Identifiability set* is the percentage of the population which is uniquely identified given their top$-N$ network.

Definition 13 (**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 sequence of the states is 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 (Xu et al., 2016; Scholtes, 2017) 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 14 (**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 $e_{ij} = \left(v_i^u, v_j^u\right) \in E^u$ represents the information that u had at least one recorded transition from v_i^u to v_j^u.

Definition 15 (**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 $W^u : E^u \rightarrow \mathbb{R}^+$. 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 that 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 (Kang et al., 2005). 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 study 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 practical 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 $G, G' \in \mathcal{G}$. To this end, according to the definitions of Section 2.5.1, we will use graph kernel functions $K(G, G') : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{R}^+$ (Vishwanathan et al., 2010), and their corresponding feature maps $\phi(G)$.

In order to ensure the result from the kernel lies in the interval $[-1, 1]$, 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 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.

3.3.3.1 Kernels on degree distribution

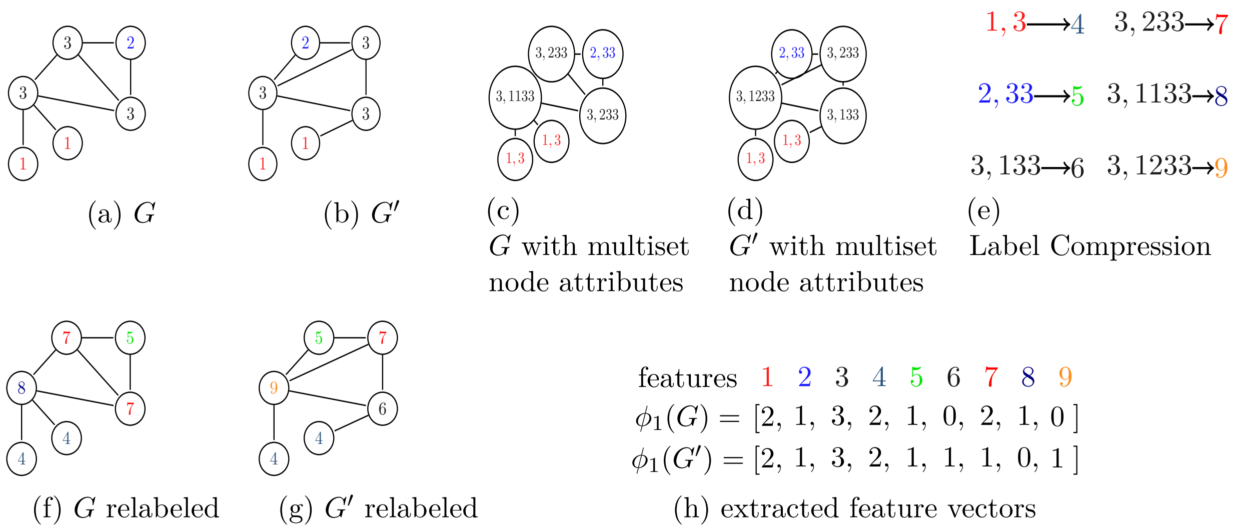
The degree distribution of nodes in the graph can be used to quantify the similarity between two graphs. For example, we can use a histogram of weighted or unweighted node degree as a feature vector. We can then compute the pairwise distance of two graphs by taking either the inner product of the feature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

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

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

3.3.3.2 Kernels 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* (Haussler, 1999). 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


1, 3⟶4
3, 233⟶7
2, 33⟶5
3, 1133⟶8
3, 133⟶6
3, 1233⟶9
features 1 2 3 4 5 6 7 8 9
$\phi_1(G) = [2, 1, 3, 2, 1, 0, 2, 1, 0]$
$\phi_1(G') = [2, 1, 3, 2, 1, 1, 1, 0, 1]$

(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.

map vectors

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

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

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

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 (Borgwardt and Kriegel, 2005) expresses the similarity between two graphs by counting the co-occurring shortest paths in the graphs. It can be written in the form of Eq. (3.3), where each element $s_i \in \mathcal{S}$ is a triplet $\left(a^i_{\text{start}}, a^i_{\text{end}}, n\right)$, 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. (2011) proposed an efficient method to construct a graph kernel utilizing the Weisfeiler-Lehman (*WL*) test of isomorphism (Weisfeiler and Lehman, 1968). The idea of the *WL* kernel is to measure co-occurrences 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 Fig. 3.1.

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

$$K_{WL}^{h}(G, G') = \left\langle \phi_h(G), \phi_h(G') \right\rangle, \tag{3.5}$$

where $\phi_h(G)$ and $\phi_h(G')$ are the vectors of labels extracted after running h steps of the computation (Fig. 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 (DKs) are a unified framework that takes into account similarity relations at the level of atomic substructures in the kernel computation (Yanardag and Vishwanathan, 2015). Hence, these kernels can quantify *similar substructure* co-occurrence, offering more robust feature representations. DKs are based on computing the following inner product:

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

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 semidefinitive 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 (Mikolov et al., 2013).

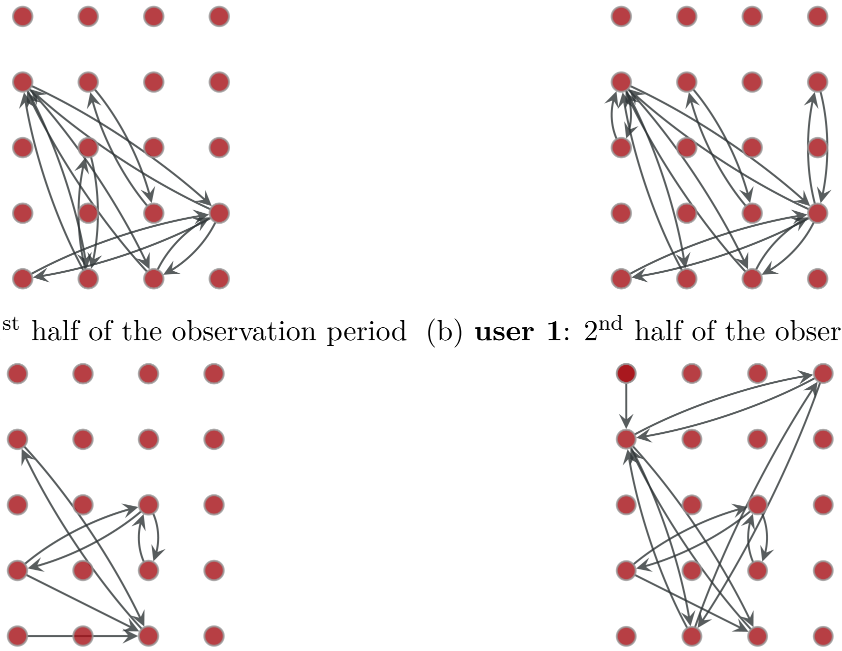
3.3.4 Deanonymization of user mobility networks and privacy leakage evaluation

3.3.4.1 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



(a) **user 1**: 1st half of the observation period (b) **user 1**: 2nd half of the observation period

(c) **user 2**: 1st half of the observation period (d) **user 2**: 2nd 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 highest 10th percentile of 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**.

deanonymization of mobility traces. Empirical evidence for our hypothesis is shown in Fig. 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.

3.3.4.2 Threat model

We assume that 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* $l_{G'} \in \mathcal{L}$.

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 $\mathcal{L}$.

We define a normalised similarity metric among the networks $K : \mathcal{G}_{\text{training}} \times \mathcal{G}_{\text{test}} \rightarrow \mathcal{R}^+$. 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 $\mathcal{L}$.

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:

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

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:

$$P_K\left(l_{G'} = l_{G_i} | \mathcal{G}_{\text{training}}\right) := \frac{f\left(K(G_i, G')\right)}{\sum\limits_{j \in \mathcal{L}} f\left(K(G_j, G')\right)}, \text{ for every } G_i \in \mathcal{G}_{\text{training}}. \tag{3.8}$$

3.3.4.3 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 non-negative *privacy loss* which will result in the expected rank of user to be smaller than $|\mathcal{L}|/2$. The privacy loss (PL) can be quantified as follows:

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

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 network 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, on top of their high computational cost, 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 $O(|\mathcal{G}_{\text{training}}|)$ 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.

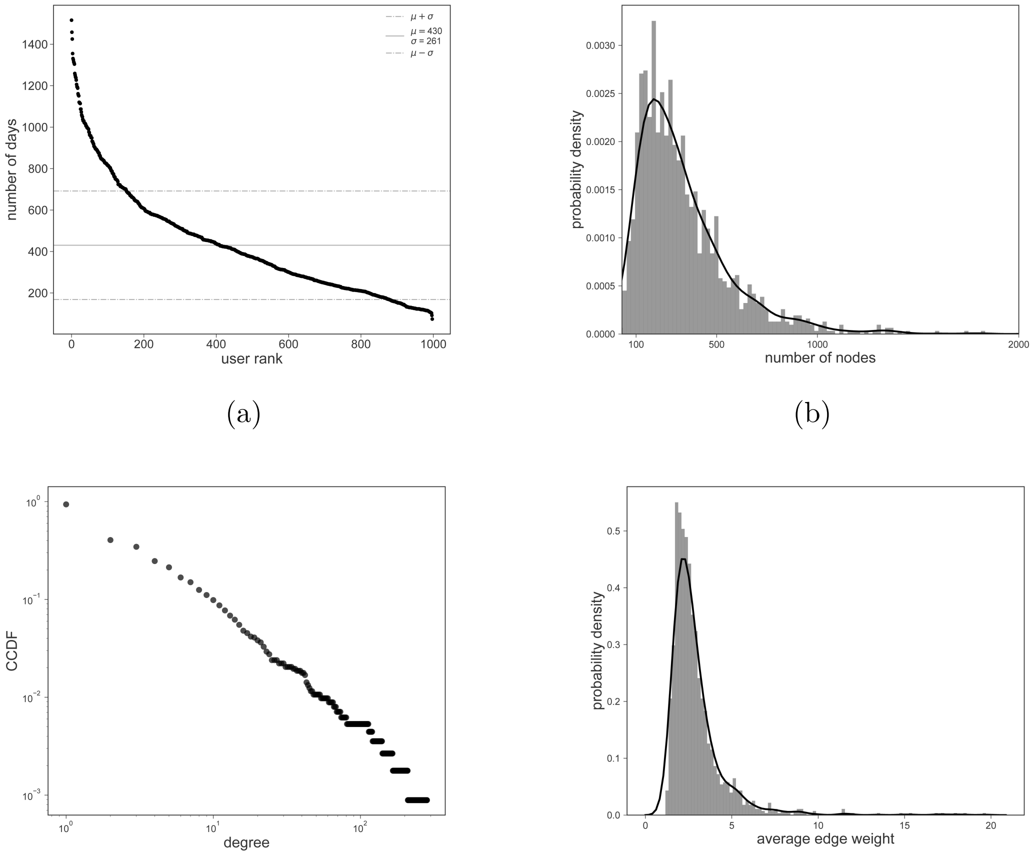
3.4.1 Data description

We evaluate our methodology on the Device Analyzer dataset (Wagner et al., 2014). 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 and wireless connectivity. For privacy purposes, released *cid* information is given a unique 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 number of recorded location datapoints in the dataset. Fig. 3.3a 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 reidentifiability 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.

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.



(a) (b)

(c) (d)

Figure 3.3: Empirical statistical findings of the Device Analyzer dataset. (a) Distribution of the observation period duration. (b) Normalized histogram and empirical 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.

As outlined in Section 3.3.1, we analysed the pathways of the Device Analyzer dataset during the entire observation period, applying the model selection method of Scholtes (2017).[3] This method tests graphical models of varying orders and selects the optimal order by balancing the model complexity and the explanatory power of observations.

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 Fig. 3.4. In other words, when considering 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. Hence, while there might still be regions in the graph which are best represented by a higher-order model, the optimal

[3] https://github.com/IngoScholtes/pathpy

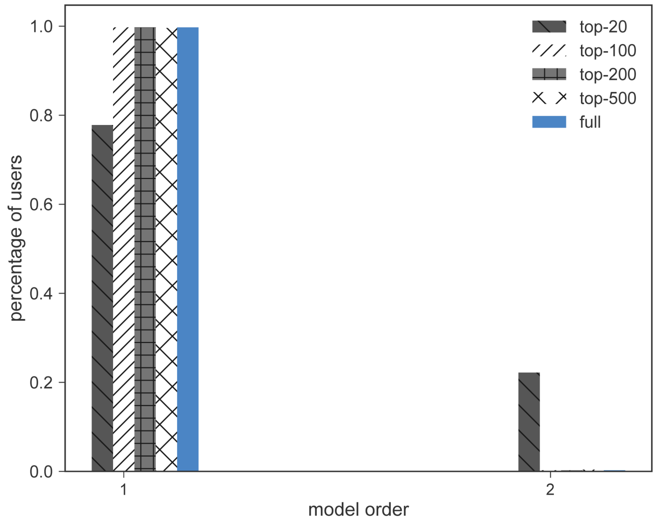


Figure 3.4: Optimal order for increasing number of locations.

Networks	# of	Num. of	Edges,	Density,	Avg.	Diameter,	Avg.	Recurrence
	networks	nodes, avg.	avg.	avg.	clust. coef.	avg.	short. path	rate (%)
top−50 locations	1, 500	49.9 ± 1.3	236.6 ± 78.1	0.19 ± 0.06	0.70 ± 0.07	3.42 ± 0.86	1.93 ± 0.20	84.7 ± 5.6
top−100 locations	1, 500	98.3 ± 7.9	387.1 ± 144.7	0.08 ± 0.03	0.60 ± 0.10	4.67 ± 1.48	2.33 ± 0.40	78.3 ± 7.8
top−200 locations	1, 500	179.2 ± 37.8	548.2 ± 246.1	0.04 ± 0.02	0.47 ± 0.12	7.52 ± 4.21	3.07 ± 1.18	73.0 ± 9.9
full	1, 500	334.6 ± 235.8	741.6 ± 527.3	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.

order describing the entire graph is one. Therefore we use a model of order one in the rest of this chapter.

3.4.3 Data properties and statistics

In Table 3.1 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.

Fig. 3.3b 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.

N	4	5	6	7	8	9
# undirected	11	34	156	1,044	12,346	274,668
N	4	5		6		7
# directed	2,128	9,608		1,540,944		882,033,440

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

As shown in Fig. 3.3c, 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.

Fig. 3.3d shows 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

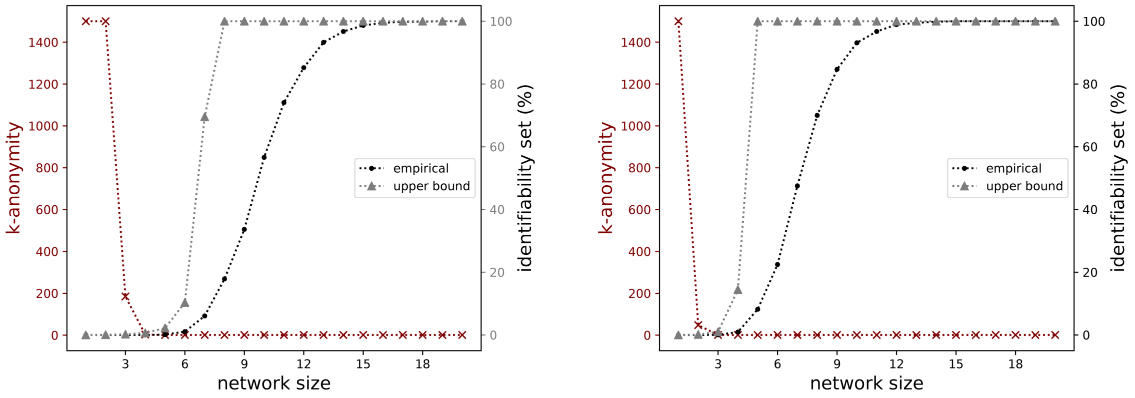
We examine to what extent the heterogeneity of users' mobility behaviour can be expressed in the topology of the state connectivity networks. For this purpose, we generate the isomorphism classes of the top$-N$ networks of our dataset for increasing network size N. We then compute the graph $k-$anonymity 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 efficient way of enumerating non-isomorphic graphs is by using the algorithm of McKay and Piperno (2014), implemented in the package `nauty`.[4] Table 3.2 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 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 Fig. 3.5, *sparsity arises in a mobility network even for very small N*. In particular, in

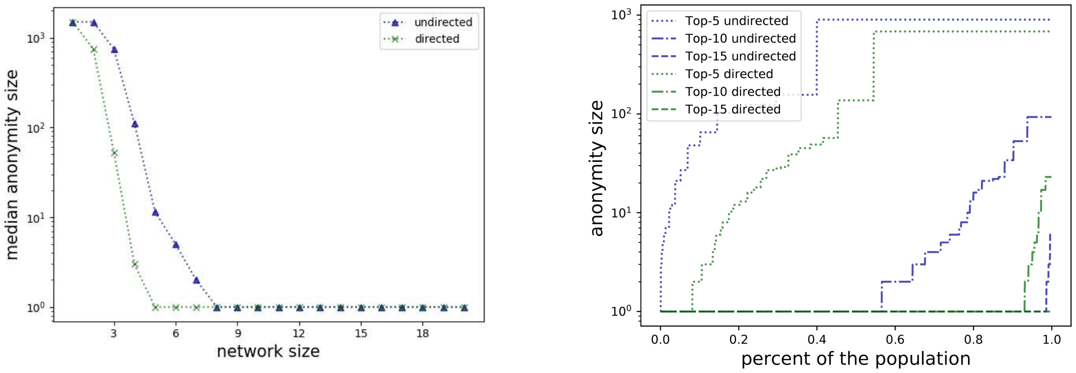
[4] `http://pallini.di.uniroma1.it/`



(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.

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).

Fig. 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 Fig. 3.6a. In Fig. 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 is 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 set 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 users' identities are disclosed to the adversary, and the *test* sequences where user identities are undisclosed to the adversary, and 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 subsequent *cid*s 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, in this part of the study, 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:

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

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, the 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

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
- *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 Fig. 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 substructure 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.

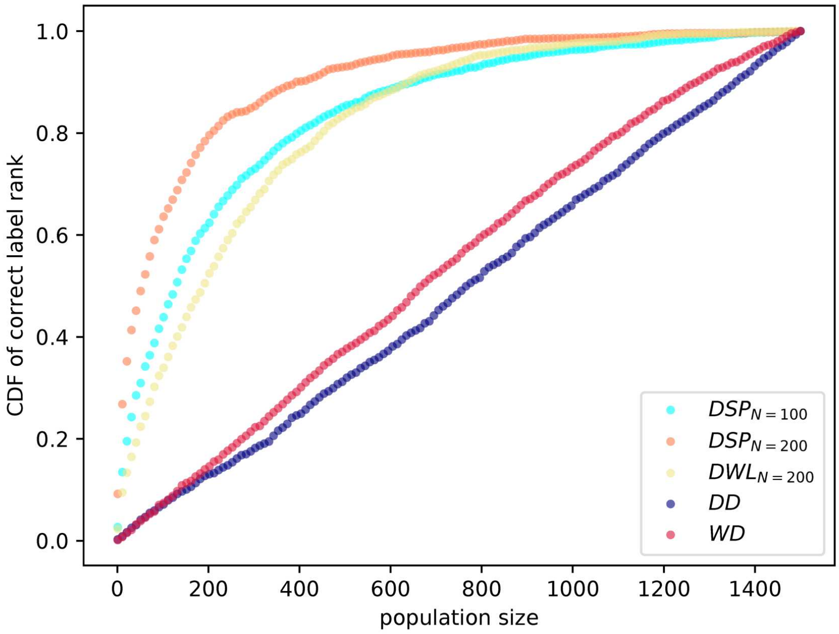


Figure 3.7: *CDF* of true rank over the population according to different kernels.

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 locations. The deep variant can also account for variation at 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 10^3 were also computed for the unweighted and weighted degree distributions and passed through a Gaussian RBF kernel. We can see that the unweighted degree distribution *DD* gives almost a random ranking, as this kernel produces a very high-dimensional mapping, which 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, which partly mitigate dimensionality effects, did not significantly improve the performance and are not presented for brevity.

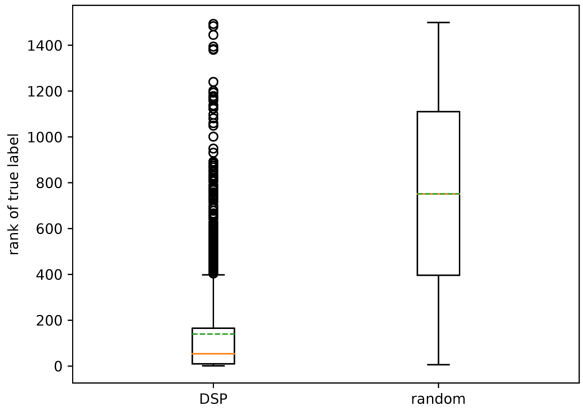


Figure 3.8: Boxplot of rank for the true labels of the population according to a Deep Shortest-Path kernel and to a random ordering.

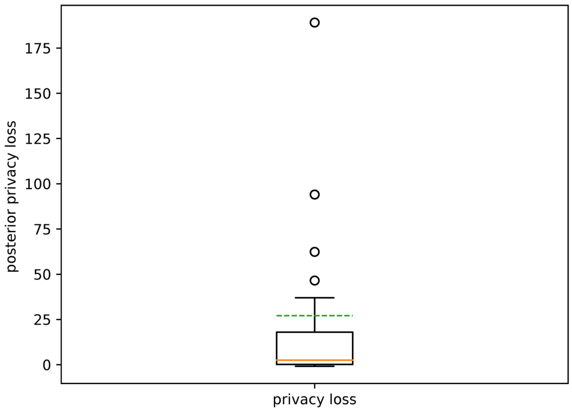


Figure 3.9: Privacy loss over the test data of our population for an adversary adopting the informed policy of (3.10). Median privacy loss is 2.52.

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:

- **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. the Weisfeiler-Lehman and the 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 Fig. 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 3.3.4.2, applying some simple probabilistic policy that uses pairwise similarity information. For example, let us examine the privacy loss implied by the update rule in (3.8) for function f:

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

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. Fig. 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*, therefore enhancing 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 querying their kernel matrix. Thus studying the anonymity associated with kernels is valuable for ensuring that such learning systems do not leak the privacy of the original data.

3.6 Summary & discussion

In this chapter we have shown that the mobility networks of individuals exhibit significant diversity, and the topology of the mobility network itself, without explicit privacy-revealing 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 retrieve the same user from a dataset recorded at a different 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. Therefore, our method is immune to commonly adopted privacy intending practices of geographic information masking or removal, and temporal cloaking, and thus it may lead to new mobility deanonymization attacks.

Moreover, we find that reducing the number of edges (transitions between locations) in a mobility network does not necessarily make the network more privacy-preserving, while user anonymity is violated even when reducing the number of nodes (locations). Conversely, releasing the frequency of node visits and the direction of transitions in a mobility network does aid the identifiablility of a mobility network for adversaries applying graph kernel similarity metrics on identified historical data. We provide empirical evidence that neighborhood relations in the high-dimensional spaces generated by the tested deep graph kernels remain meaningful for our dataset of networks (Beyer et al., 1999). 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 set 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.

Beyond 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 securing $k-$anonymity guarantees for 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 (Bogdanov et al., 2011), or searching for persistent cascades (Morse et al., 2016). Frequent substructure pattern mining (e.g. gSpan, Yan and Han (2002)) and discriminative frequent subgraph mining (e.g. CORK, Thoma et al. (2010)) techniques can also be considered.

Our methodology is, in principle, applicable to all types of data where individuals transition amongst a set of discrete states. Therefore, the performance of such retrieval strategies can also be evaluated 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 (Song et al., 2011) 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

In Chapter 2, we exposed the prohibitive computational limitations of Bayesian inference in the regime of modern large-scale data, and discussed coreset-based summarization as a viable solution for scalable approximate inference under statistical guarantees. In Chapter 3, we considered a case study on a massive high-dimensional dataset capturing longitudinal mobility information of a population, and quantified the privacy loss incurred via coarse representations of the datapoints that can be used for fast data analysis. Motivated by the quest for scalable learning methods on sensitive data, in this chapter we propose *pseudocoreset variational inference*, a general-purpose approximate inference method designed to enable scalable inference on high-dimensional datasets, under the guarantees of approximate differential privacy.

We begin by investigating the shortcomings of existing Bayesian coreset constructions in the increasingly common setting of sensitive, high-dimensional data. In particular, we prove that there are situations in which the Kullback-Leibler 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 reduction compared to traditional coresets, and that pseudocoresets provide privacy without a significant loss in approximation quality.

4.1 Related work & contributions

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 datapoints are not unique or particularly informative for modeling 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 (Drineas and Mahoney, 2005; Musco and Musco, 2017; Agrawal et al., 2019) or sparse pseudo-input parameterizations for Gaussian processes (Williams and Seeger, 2001; Csató and Opper, 2002; Snelson and Ghahramani, 2005; Titsias, 2009), Bayesian inference (Huggins et al., 2016; Huggins et al., 2017; Campbell and Broderick, 2018; 2019), maximum likelihood parameter estimation (DuMouchel et al., 1999; Madigan et al., 2002), linear regression (Zhou et al., 2007; Guhaniyogi and Dunson, 2015), geometric shape approximation (Agarwal et al., 2005), clustering (Feldman et al., 2011; Bachem et al., 2015; Braverman et al., 2016; Lucic et al., 2016b), and dimensionality reduction (Feldman et al., 2016).

A common form of summarization is that of a sparse, weighted subset of the original dataset—a *coreset* (Agarwal et al., 2005). 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 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 16). Furthermore, directly releasing a subset of the original data precludes any possibility of individual privacy under the current standard of differential privacy (Dwork et al., 2006c; Dwork and Roth, 2014). Past work addresses this issue in the context of clustering and computational geometry (Feldman et al., 2009; 2017)—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 chapter, 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 (Madigan et al., 2002; Snelson and Ghahramani, 2005; Zhou et al., 2007; Titsias, 2009), 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 16, 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 (Campbell and Beronov, 2019), we formulate pseudocoreset construction as variational inference, and provide a stochastic optimization method (Section 4.3). As a consequence of the use of pseudodata—as well as privacy-preserving stochastic gradient descent mechanisms (Abadi et al., 2016; Jälkö et al., 2017; Park et al., 2020)—we show that our method can easily be modified to output a privatized pseudocoreset. The chapter concludes with experimental results demonstrating the performance of pseudocoresets on real and synthetic data (Section 4.4).

4.2 Existing Bayesian coresets

Our 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. $f(x_n, \theta) := \log \pi(x_n|\theta)$, 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

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

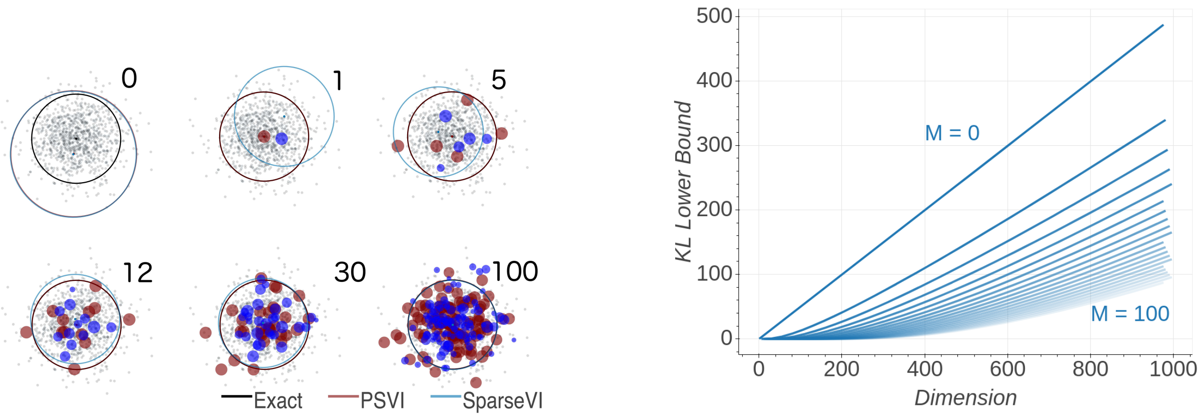
for $w \in \mathbb{R}^N$, $w \geq 0$ may be used in most popular posterior inference schemes (Neal, 2011; Ranganath et al., 2014; Kucukelbir et al., 2017). If the number of nonzero entries $\|w\|_0$ of w is small, this results in a significant reduction in computational burden. Recent work has formulated the problem of constructing a Bayesian coreset of size $M \in \mathbb{N}$ as sparse variational inference (Campbell and Beronov, 2019),

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

and showed that the objective can be minimized using stochastic estimates of $\nabla_w \mathrm{D_{KL}}\left(\pi_w || \pi\right)$ 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 16 shows that this is problematic when summarizing high-dimensional data; in the common setting of posterior inference for a Gaussian mean, the KL divergence $\mathrm{D_{KL}}\left(\pi_{w^\star} || \pi\right)$ of the *optimal* coreset of any size scales with the dimension of the data. The proof may be found in Appendix A.1.



(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. 4.2b for the quantitative KL comparison. (b) Optimal coreset KL divergence lower bound from Proposition 16 as a function of dimension with $\delta = 0.5$, and coreset size M evenly spaced from 0 to 100 in increments of 5.

Proposition 16. *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*

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

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

The bound in Proposition 16 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 used in Appendix A.1 results in a vacuous $\mathrm{D_{KL}}\left(\pi_{w^\star} || \pi\right) = 0$ lower bound.

4.3 Bayesian pseudocoresets

Proposition 16 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,

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

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), indicating the absence of a lower bound for the KL divergence of the optimal coreset defined on pseudodata in the setting of Proposition 16 *regardless of data dimensionality.*

Corollary 17. *Suppose the same setting with Proposition 16. The optimal size-M pseudocoreset $(u^\star, w^\star)$ defined on pseudodata $u_1, \dots, u_M \in \mathbb{R}^d$ achieves $\mathrm{D}_{\mathrm{KL}}\left(\pi_{u^\star,w^\star} || \pi\right) = 0$, for any size $M \geq 1$ and any data dimension d.*

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 (Snelson and Ghahramani, 2005; Titsias, 2009), privacy-preserving compression in linear regression (Zhou et al., 2007), herding (Welling, 2009; Chen et al., 2010; Huszár and Duvenaud, 2012), and deep generative models (Tomczak and Welling, 2018).

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 (Campbell and Beronov, 2019), 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. $f(u_m, \theta) = \log \pi(u_m|\theta)$. We construct a coreset by minimizing the KL divergence over both the pseudodata locations and weights,

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

As opposed to previous Bayesian coreset construction optimization problems (Campbell and Broderick, 2018; Campbell and Beronov, 2019; Campbell and Broderick, 2019), 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

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

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. (4.8). Despite the presence of the intractable normalization constants and expectations, we show in Appendix A.2 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

$$\nabla_w \mathrm{D_{KL}} = -\,\mathrm{Cov}_{u,w}\left[\tilde{f}, f^T 1 - \tilde{f}^T w\right], \quad \nabla_{u_m} \mathrm{D_{KL}} = -w_m\,\mathrm{Cov}_{u,w}\left[h(u_m), f^T 1 - \tilde{f}^T w\right], \tag{4.9}$$

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. (4.9) 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 4.4.1; see also Appendix A.3.1), 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).

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 $x$,    $\mathcal{B} \sim \mathsf{UnifSubset}\left([N], M\right)$,    $\mathcal{B} := \{b_1, \dots, b_M\}$
    $u_m \leftarrow x_{b_m}$,    $w_m \leftarrow N/M$,    $m = 1, \dots, M$
    for $t = 1, \dots, T$ do
            ▷ Take $S$ samples from current pseudocoreset posterior
        $(\theta)_{s=1}^{S} \overset{\text{i.i.d.}}{\sim} \pi_{u,w}$ where $\pi_{u,w}(\theta) \propto \exp\left(\sum_{m=1}^{M} w_m f(u_m, \theta)\right) \pi_0(\theta)$
        $\mathcal{B} \sim \mathsf{UnifSubset}\left([N], B\right)$   ▷ Obtain a minibatch of $B$ points from the full data
        for $s = 1, \dots, S$ 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 $m = 1, \dots, M$ 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 $w$
        for $m = 1, \dots, M$ do                                          and $(u_m)_{m=1}^{M}$
            $\hat{\nabla}_{u_m} \leftarrow -w_m 1/S \sum_{s=1}^{S} \tilde{h}_{m,s} \left(N/B\, g_s^T 1 - \tilde{g}_s^T w\right)$
        $w \leftarrow \max(w - \gamma_t \hat{\nabla}_w, 0)$                    ▷ Take stochastic gradient step in $w$
        for $m = 1, \dots, M$ do                                          and $(u_m)_{m=1}^{M}$
            $u_m \leftarrow u_m - \gamma_t \hat{\nabla}_{u_m}$
    return $w$, $(u_m)_{m=1}^{M}$
```

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

$$u_m \leftarrow x_{b_m}, \quad w_m \leftarrow N/M, \quad m = 1, \dots, M \tag{4.10}$$

$$\mathcal{B} \sim \mathsf{UnifSubset}\left([N], M\right), \quad \mathcal{B} := \{b_1, \dots, b_M\}. \tag{4.11}$$

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

$$w_m \leftarrow \max\left(0, w_m - \gamma_t (\hat{\nabla}_w)_m\right), \quad u_m \leftarrow u_m - \gamma_t \hat{\nabla}_{u_m}, \quad 1 \leq m \leq M. \tag{4.12}$$

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 full dataset,

$$\hat{\nabla}_w := -\frac{1}{S} \sum_{s=1}^{S} \tilde{g}_s \left(\frac{N}{B} g_s^T 1 - \tilde{g}_s^T w\right), \tag{4.13}$$

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

where

$$\begin{aligned}\tilde{h}_{m,s} &:= \nabla_u f(u_m, \theta_s) - \frac{1}{S}\sum_{s'=1}^{S} \nabla_u f(u_m, \theta_{s'}), \qquad & g_s &:= \left(f(\theta_s) - \frac{1}{S}\sum_{s'=1}^{S} f(\theta_{s'})\right)\Bigg|_{\mathcal{B}}, \\ \tilde{g}_s &:= \tilde{f}(\theta_s) - \frac{1}{S}\sum_{s'=1}^{S} \tilde{f}(\theta_{s'}), & \mathcal{B} &\sim \mathsf{UnifSubset}\left([N], B\right),\end{aligned} \tag{4.15}$$

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 (Hoffman and Gelman, 2014; Jacob et al., 2020) 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 is 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 18 (Dwork et al., 2006a; b).

Definition 18 ((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 $A \subseteq \mathbb{R}^M_+ \times \mathcal{X}^M$, $\mathbb{P}[\mathcal{M}(x) \in A] \leq e^{\varepsilon}\mathbb{P}[\mathcal{M}(x') \in A] + \delta$.

As in Section 2.6, we consider two datasets x, x' as adjacent (denoted $x \approx x'$) if their Hamming distance equals 1, i.e. 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.

4.3.3.1 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.

4.3.3.2 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 (Dwork and Roth, 2014), 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 (Abadi et al., 2016; Wang et al., 2019). 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:

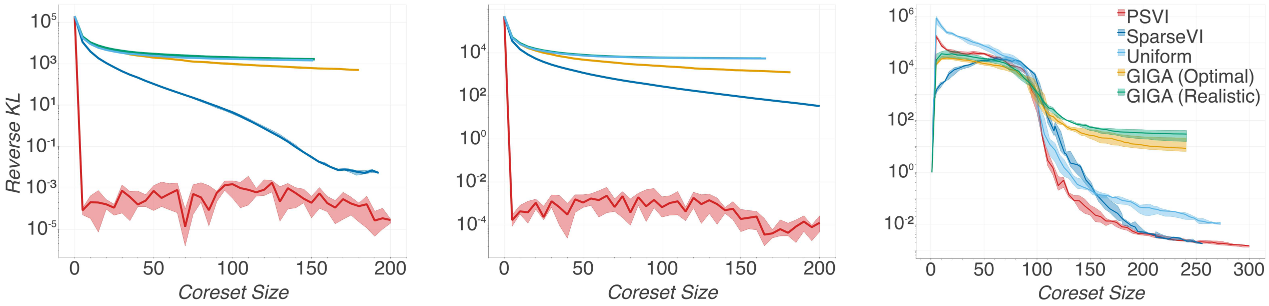
$$\text{replace} \quad (g_s^T 1)_{s=1}^S \quad \text{with} \quad Z + \sum_{i=1}^{B} \frac{G_i}{\max\left(1, \frac{||G_i||_2}{C}\right)}, \qquad Z \sim \mathcal{N}(0, \sigma^2 C^2 I), \tag{4.16}$$

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 19; 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. (4.16). Finally, a manageable number of privacy specific hyperparameters is introduced: the clipping bound C and noise level σ.

Corollary 19 (Abadi et al. (2016))**.** *There exist constants c_1, c_2 such that Algorithm 1 modified per Eq. (4.16) 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 VI (PSVI) compared against uniform random subsampling (Uniform), Hilbert coresets (GIGA, Campbell and Broderick (2018)) and SparseVI greedy coreset construction (Campbell and Beronov, 2019). 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



(a) Gaussian mean inference, $d = 200$

(b) Gaussian mean inference, $d = 500$

(c) Bayesian linear regression, $d = 100$

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

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 Appendix A.3.

4.4.1 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 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:

$$\Sigma_{u,w} = (\Sigma_0^{-1} + \sum_{m=1}^{M} w_m \Sigma^{-1})^{-1} \qquad \mu_{u,w} = \Sigma_{u,w}(\Sigma_0^{-1}\mu_0 + \Sigma^{-1} \sum_{m=1}^{M} w_m u_m). \tag{4.17}$$

Using the exact posterior, we derive the exact moments used in the gradient formulae from Eq. (4.9) in closed form (see Appendix A.3.1),

$$\begin{aligned} \mathrm{Cov}_{u,w}[f_n, f_m] &= v_n^T \Psi v_m + {}^1\!/_2 \operatorname{tr} \Psi^T \Psi, & \mathrm{Cov}_{u,w}[\tilde{f}_n, f_m] &= \tilde{v}_n^T \Psi v_m + {}^1\!/_2 \operatorname{tr} \Psi^T \Psi, \\ \mathrm{Cov}_{u,w}[h(u_i), f_n] &= Q^{-T} \Psi v_n, & \mathrm{Cov}_{u,w}[h(u_i), \tilde{f}_n] &= Q^{-T} \Psi \tilde{v}_n, \end{aligned} \tag{4.18}$$

where Q is the lower triangular matrix of 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 16. 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 the true posterior with a single pseudodata point regardless of data dimension. As shown earlier in Fig. 4.1a, PSVI has the capacity to move the pseudodata points in order to capture the true posterior very efficiently.

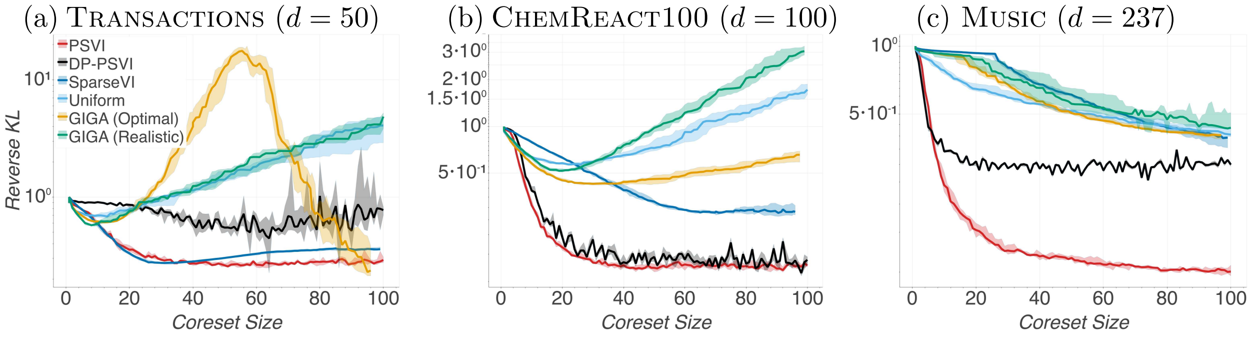
4.4.2 Bayesian linear regression

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 Appendix A.3.2.1). 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 Appendix A.3.2.2), while PSVI achieves the fastest decay rate over sizes $100 \leq M < 250$, surpassing SparseVI.

4.4.3 Bayesian logistic regression

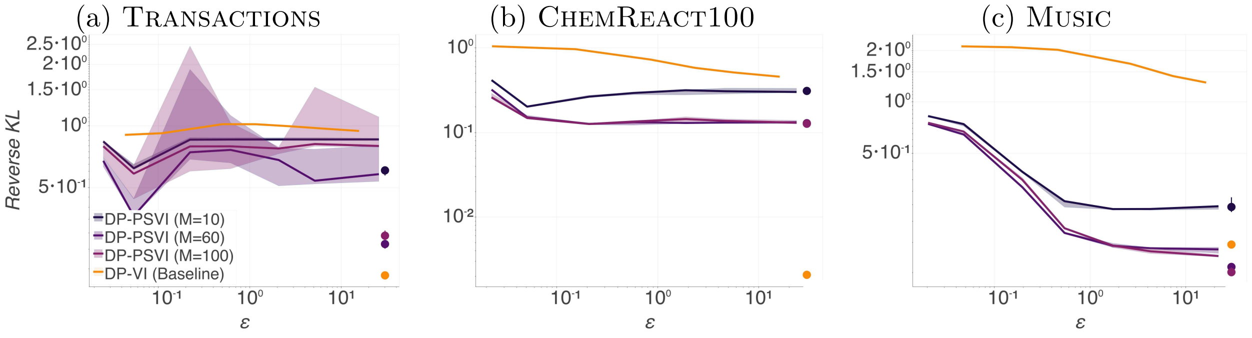
Finally, we compare (pseudo)coreset construction methods on Bayesian logistic regression applied to 3 large (8.4–100K datapoints, 50–237 dimensions) datasets. For brevity, equations and gradients for the logistic regression model, additional experiments on 3 smaller-scale datasets, full dataset descriptions, hyperparameter selection, time performance evaluation and results on an incremental scheme for pseudocoreset construction are deferred to Appendix A.3.3. 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 $\sigma = 5$. Our hyperparameters choice implies privacy parameters $\varepsilon = 0.2$ and $\delta = 1/N$ for each of the datasets. 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.

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



(a) Transactions ($d = 50$) (b) ChemReact100 ($d = 100$) (c) Music ($d = 237$)

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.



(a) Transactions (b) ChemReact100 (c) Music

Figure 4.4: Approximate posterior quality over decreasing differential privacy guarantees for private pseudocoresets of varying size (DP-PSVI) plotted against private variational inference (DP-VI, Jälkö et al. (2017)). 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.

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 16. 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 the achieved posterior approximation quality via DP-PSVI, against a competitive state-of-the-art method for general-purpose private inference (DP-VI, Jälkö et al., 2017). The plots display the behaviour of methods over a wide range of ε values, achieved using varying levels of privatization noise, and δ always set to $1/N$. For logistic regression, DP-VI infers an approximate posterior from the family of Gaussians with diagonal covariance via ADVI (Kucukelbir et al., 2017), 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 Summary & discussion

In this chapter, 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.

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 (Huggins et al., 2020)). 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 (Wang et al., 2017; Miller and Dunson, 2019), can offer protection against modeling mismatches, and will be the subject of the following chapter. 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.

Chapter 5

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

In Chapter 4, we proposed a novel Bayesian coreset construction that addresses scalability to dataset size and dimensionality, along with privacy preservation requirements, often arising in large-scale inference. In this chapter, we design one more coreset construction that aims to resolve another frequently occurring challenge in probabilistic inference over real-world datasets, namely robustness to model misspecification.

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 classical Bayesian inference depends critically on whether observations conform with the assumed data generating model, which is impossible to guarantee in practice. In this chapter, 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 generalized 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 and unstructured data contamination. 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 Related work & contributions

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 remedy various practical problems. Hence, recent years have witnessed a surge of interest in scaling up inference in the large-data regime via stochastic methods, relying on random minibatch access to the dataset (Welling and Teh, 2011; Hoffman et al., 2013; Angelino et al., 2016). 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 (Ghorbani and Zou, 2019).

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 (Zheng et al., 2008; Zhuang et al., 2015). 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 (Frénay and Verleysen, 2013), and necessitates laborious data cleansing (Lewis et al., 2004; Paschou et al., 2010), 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 (Barreno et al., 2010; Biggio et al., 2012; Li et al., 2016; Koh and Liang, 2017; Steinhardt et al., 2017; Ghorbani and Zou, 2019), realised for example via data generation from fake accounts. Outliers detection is challenging, particularly in high dimensions (Lucic et al., 2016a; Diakonikolas et al., 2019; Dickens et al., 2020). 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 (Sheng et al., 2008; Whitehill et al., 2009; Raykar et al., 2010; Karger et al., 2011; Liu et al., 2012; Zhang et al., 2016). 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 chapter, 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 (Huggins et al., 2016; Campbell and Beronov, 2019; Campbell and Broderick, 2019) 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 (Berger et al., 1994) have sought solutions to mismatches between available observations and the assumed data generating model, via proposing heavy-tailed data likelihood functions (Huber and Ronchetti, 2009; Ríos Insúa and Ruggeri, 2012) and localization (de Finetti, 1961; Wang and Blei, 2018), using robust statistical divergences (Futami et al., 2018; Knoblauch et al., 2018; Miller and Dunson, 2019), employing robust gradient estimates over Langevin Monte Carlo methods (Bhatia et al., 2019), or inferring datapoints' specific importance weights (Wang et al., 2017). 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 remainder of the chapter is organized as follows. In Section 5.2 we introduce our proposed method for scalable robust inference, providing an incremental black-box construction for sparse approximations of the β-posterior. In Section 5.3 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 5.4 we provide conclusions and discuss extensions of our method.

5.2 Method

In this section we present β-Cores, our unified solution to the robustness and scalability challenges of large-scale Bayesian inference. Section 5.2.1 introduces the main quantity of interest in our inference method, and shows how it addresses the exposed issues. Section 5.2.2 presents an iterative algorithm that allows efficient approximate computations of our posterior.

5.2.1 Sparse β-posterior

Scaling up the computation of the robust β-posterior defined in Eq. (2.22) in the regime of massive datasets for non-conjugate models is challenging: similarly to the standard Bayesian posterior Eq. (2.17), applying Markov chain Monte Carlo methods to sample from the β-posterior, implies a computational cost scaling at order $\Theta(N)$.

Bayesian coresets (Huggins et al., 2016; Campbell and Broderick, 2019) 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. (2.23) in Eq. (2.22), 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

$$\pi_{\beta,w}(\theta|x) = \frac{1}{Z(\beta, w)} \exp\left(\sum_{n=1}^N w_n f_n(\theta)\right) \pi_0(\theta), \tag{5.1}$$

where $(f_n(\theta))_{n=1}^N$ correspond to the β-likelihood terms, π_0 is the prior, and $Z(\beta, w)$ is the marginal likelihood (which in the general case corresponds to an intractable constant). In the case of the β-posterior on the full dataset Eq. (2.22), we have $w = 1 \in \mathbb{R}^N$; 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 model-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. (2.20) can carry over in coresets posterior.

The recent formulation of Riemannian coresets (Campbell and Beronov, 2019) has framed the problem of coreset construction as variational inference 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

$$w^\star = \arg\min_{w \in \mathbb{R}^N} \mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w} || \pi_\beta\right) \quad \text{s.t.} \quad w \geq 0,\ ||w||_0 \leq M. \tag{5.2}$$

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

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w} || \pi_\beta\right) := \mathbb{E}_{\beta,w}\left[\log \frac{\pi_{\beta,w}}{\pi_\beta}\right]. \tag{5.3}$$

In our formulation it is easy to observe that posteriors of Eq. (5.1) form a set of *exponential family distributions* (Wainwright and Jordan, 2008), with natural parameters $w \in \mathbb{R}_{\geq 0}^N$, sufficient statistics $(f_n(\theta))_{n=1}^N$, and log-partition function $\log Z(\beta, w)$. Following Campbell and Beronov (2019), the objective can be expanded as

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w} || \pi_\beta\right) = \log Z(\beta) - \log Z(\beta, w) - \sum_{n=1}^N \mathbb{E}_{\beta,w}\left[f_n(\theta) - w_n f_n(\theta)\right], \tag{5.4}$$

and minimized via gradient descent on w. The gradient of the objective of Eq. (5.4) can be derived in closed form, as

$$\nabla_w \mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w} || \pi_\beta\right) = -\,\mathrm{Cov}_{\beta,w}\left[f, (1-w)^T f\right], \tag{5.5}$$

Algorithm 2 Incremental construction of sparse β-posterior

```
procedure $\beta$-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
            ▷ Take S samples from current coreset posterior
        $(\theta)_{s=1}^S \overset{\text{i.i.d.}}{\sim} \pi_{\beta,w} \propto \exp\left(w^T f\right) \pi_0(\theta)$
            ▷ Obtain a minibatch of $B$ datapoints from the full dataset
        $\mathcal{B} \sim \mathsf{UnifSubset}\left([N], B\right)$
            ▷ Compute the $\beta$-likelihood vectors over the coreset and minibatch datapoints for each sample
        $g_s \leftarrow \left(f(x_m, \theta_s, \beta) - \frac{1}{S}\sum_{r=1}^S f(x_m, \theta_r, \beta)\right)_{m\in\mathcal{I}} \in \mathbb{R}^M$
        $g'_s \leftarrow \left(f(x_b, \theta_s, \beta) - \frac{1}{S}\sum_{r=1}^S f(x_b, \theta_r, \beta)\right)_{b\in\mathcal{B}} \in \mathbb{R}^B$
            ▷ Get empirical estimates of correlation over the coreset and minibatch datapoints
        $\widehat{\mathrm{Corr}} \leftarrow \mathrm{diag}\left[\frac{1}{S}\sum_{s=1}^S g_s {g_s}^T\right]^{-\frac{1}{2}} \left(\frac{1}{S}\sum_{s=1}^S g_s \left(\frac{N}{B} 1^T g'_s - w^T g_s\right)\right) \in \mathbb{R}^M$
        $\widehat{\mathrm{Corr}}' \leftarrow \mathrm{diag}\left[\frac{1}{S}\sum_{s=1}^S g'_s {g'_s}^T\right]^{-\frac{1}{2}} \left(\frac{1}{S}\sum_{s=1}^S g'_s \left(\frac{N}{B} 1^T g'_s - w^T g_s\right)\right) \in \mathbb{R}^B$
            ▷ Add next datapoint via correlation maximization
        $n^\star \leftarrow \arg\max_{n\in[m]\cup[B]} \left(\left|\widehat{\mathrm{Corr}}\right| \cdot \mathbb{1}[n \in \mathcal{I}] + \widehat{\mathrm{Corr}}' \cdot \mathbb{1}[n \notin \mathcal{I}]\right), \quad \mathcal{I} \leftarrow \mathcal{I} \cup \{n^\star\}$
        for $t = 1, \ldots, T$ do  ▷ Optimize weights vector via projected gradient descent
            $(\theta)_{s=1}^S \overset{\text{i.i.d.}}{\sim} \pi_{\beta,w}(\theta) \propto \exp\left(w^T f\right) \pi_0(\theta)$
            $\mathcal{B} \sim \mathsf{UnifSubset}\left([N], B\right)$
                ▷ Compute gradient terms discretizations over the coreset and minibatch datapoints for each sample
            for $s = 1, \ldots, S$ do
                $g_s \leftarrow \left(f(x_m, \theta_s, \beta) - \frac{1}{S}\sum_{r=1}^S f(x_m, \theta_r, \beta)\right)_{m\in\mathcal{I}} \in \mathbb{R}^M$
                $g'_s \leftarrow \left(f(x_b, \theta_s, \beta) - \frac{1}{S}\sum_{r=1}^S f(x_b, \theta_r, \beta)\right)_{b\in\mathcal{B}} \in \mathbb{R}^B$
                ▷ Compute MC gradients for variational parameters
            $\hat{\nabla}_w \leftarrow -\frac{1}{S}\sum_{s=1}^S g_s \left(\frac{N}{B} 1^T g'_s - w^T g_s\right)$
                ▷ Take a projected stochastic gradient step
            $w \leftarrow \max(w - \gamma_t \hat{\nabla}_w, 0)$
    return $w$
```

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

5.2.2 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. (5.5) 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 iterates. 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 incremental construction 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.

The optimization problem of Eq. (5.2) is intractable due to the cardinality constraint; hence, our incremental scheme takes the approach of approximating the solution to the original problem via solving a sequence of interleaved combinatorial and continuous optimization problems as follows:

For $i \in \{1, \dots, M\}$:

Next datapoint selection (Combinatorial optimization)

$$m^{\star} = \arg\min_{m\in[N]} \mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w\leftarrow w\cup\{x_m\}}||\pi_{\beta}\right) \tag{5.6}$$

Coreset points reweighting (Continuous optimization)

$$w^{\star} = \arg\min_{w\in\mathbb{R}^N_{\geq 0}} \mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta,w}||\pi_{\beta}\right) \tag{5.7}$$

In Eq. (5.6) we have introduced the notation $\pi_{\beta,w\leftarrow w\cup\{x_m\}}$ to consider the coreset expansion that assigns potentially non-zero weight to a datapoint x_m.

5.2.2.1 Next datapoint selection

We first select the next datapoint to include in our coreset summary Eq. (5.6), via a greedy selection criterion. Although maximizing the decrease in KL locally via Eq. (5.5), seems to be the natural greedy choice here, this would incur the impractical cost of resampling from the coreset posterior *for all potential expansions* of the coreset with a new datapoint. Moreover, even if we can tolerate this cost, adding a single unweighted datapoint is likely to induce a negligible effect on the coreset posterior, especially in massive dataset settings. Submodularity of the objective would be a clearly attractive property, as it could possibly point to a cheap greedy strategy with provable suboptimality guarantees—however, our analysis in Appendix B.2 demonstrates that this property is generally not satisfied for our problem.

Hence, considering that the weight of the active support for the updated coreset will be optimized in the subsequent step Eq. (5.7) of the algorithm, an efficient method for informative datapoint selection can be based on adding a datapoint that correlates well with the direction of residual error. Thus we finally rely instead on the following correlation maximization criterion:

$$x_m = \arg\max_{x_n\in\mathcal{I}\cup\mathcal{B}} \begin{cases} \left|\mathrm{Corr}_{\beta,w}\left[f_n, \frac{N}{B}1^T f - w^T f\right]\right| & w_n > 0 \\ \mathrm{Corr}_{\beta,w}\left[f_n, \frac{N}{B}1^T f - w^T f\right] & w_n = 0, \end{cases} \tag{5.8}$$

where we denoted by $\mathcal{I}$ the set of coreset points. Eq. (5.8) additionally allows us to expand the information-geometric interpretation of Riemannian coresets presented in Campbell and Beronov (2019) in our construction. This criterion is invariant to scaling each potential f_n by any positive constant, and selects the potential that has the largest correlation with the current residual error $\frac{N}{B}1^T f - w^T f$. The correlations for coreset and minibatch datapoints are empirically approximated as in lines 8 and 9 of Algorithm 2 respectively.

5.2.2.2 Coreset points reweighting

After adding a new datapoint to the summary, we optimize Eq. (5.7), updating the coreset weight vector $w \in \mathbb{R}_{\geq 0}$ via T steps of projected stochastic gradient descent, for which we use the Monte Carlo estimate of Eq. (5.5) 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.

Choice of the robustness hyperparameter value. Selecting a proper value for β when doing inference using power divergences can be treated as an instance of hyperparameter optimization. Prior knowledge on the expected subspace for the inliers of a data analysis task at hand can be leveraged in order to specify a reasonable value for the hyperparameter β *a priori* (recall from Fig. 2.1a that β controls the distance from population's sufficient statistic where the maximum of the concave data influence function is located). Earlier work in robust Bayesian inference has considered automating the selection of this value in the light of observations, using cross-validation (Futami et al., 2018), or via performing on-line gradient descent on the expected predictive loss (Knoblauch et al., 2018). In a similar vein, for the purposes of variational inference using other parameterised families of divergence functions, such as the α- and f-divergence, recent approaches for adaptive learning of optimal hyperparameters have relied on controlling the variance of Monte Carlo estimates used in variational inference (Wang et al., 2018), and on gradient descent based meta-learning techniques (Zhang et al., 2021b).

5.3 Experiments & applications

We examine the inferential results achieved by our method under 3 statistical models, in scenarios capturing different types of mismatch between modeling assumptions and reality. The data contamination models used in the following experiments are reminiscent of *Huber's ϵ-contamination model* (Huber, 1992), which postulates that observed data are generated from a mixture of distributions of the form $(1-\epsilon)\cdot G+\epsilon\cdot Q$, where $\epsilon\in(0,1)$, G is a distribution of inliers captured by the assumed statistical model, and Q is an arbitrary distribution of outliers. This model has found use in several recent studies on robust statistical estimators suitable for underlying data distributions with minimal assumptions (Wei and Minsker, 2017; Chen et al., 2018).

β-Cores is compared against a uniform random sampling baseline, and stochastic batch implementations of two existing Riemannian coreset methods:

(i) SparseVI (Campbell and Beronov, 2019), 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 (Manousakas et al., 2020), the method introduced in Chapter 4, 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. From Section 5.3.1 to Section 5.3.4, the values for β are selected via cross-validation on a held-out dataset. For consistency with the compared baselines, we evaluate inference results obtained by β-Cores using the classical Bayesian posterior from Eq. (2.17) conditioned on the corresponding robustified data summary. Additional details on used benchmark datasets are presented in Appendix B.3.

5.3.1 Simulated Gaussian mean inference under stuctured data contamination

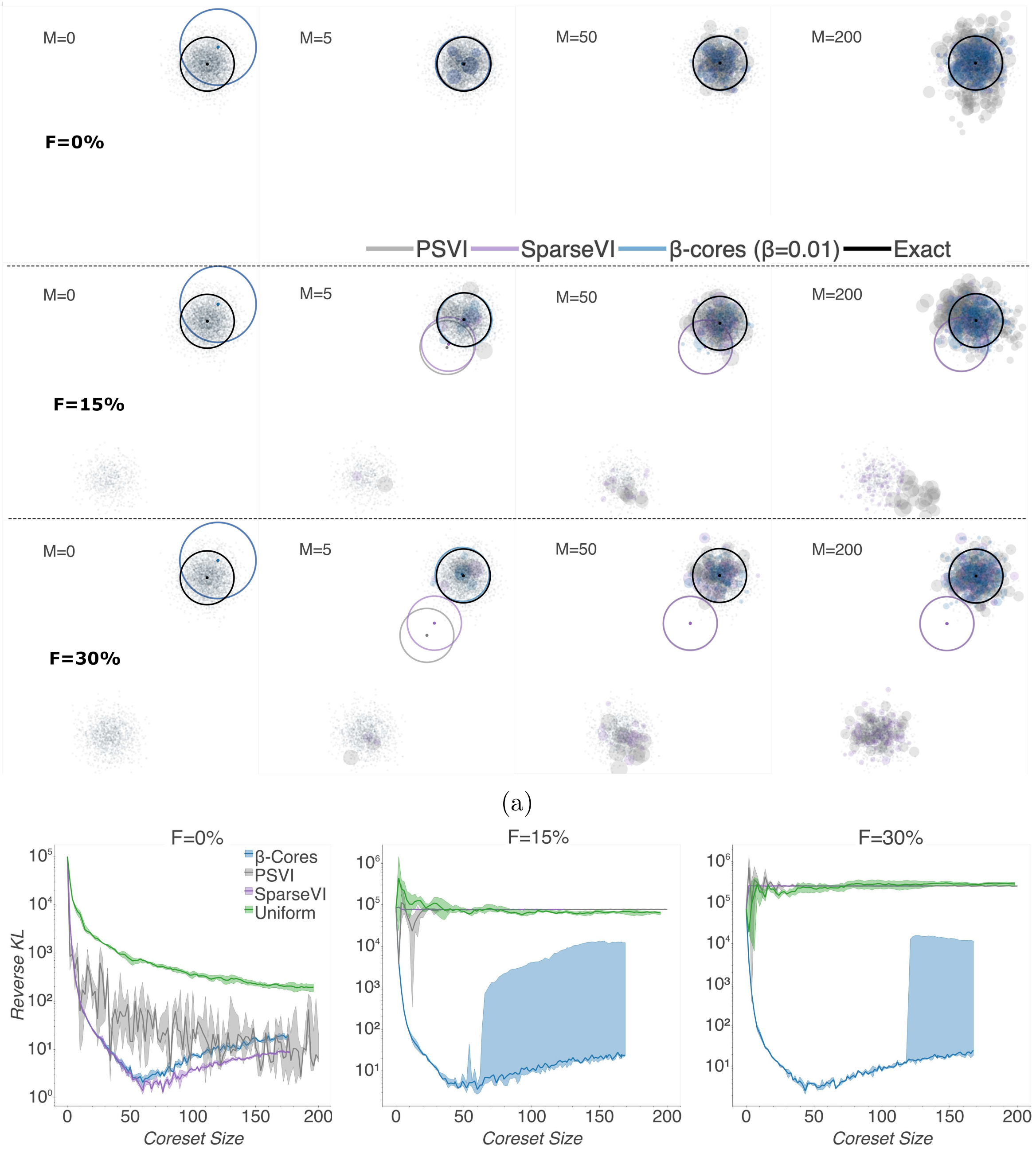
In the first experiment we study how β-Cores behaves in the setting of mean inference on synthetic d-dimensional data, sampled i.i.d. from a normal distribution with known covariance,

$$\theta\sim\mathcal{N}\left(\mu_0,\Sigma_0\right),\qquad x_n\overset{\text{i.i.d.}}{\sim}\mathcal{N}(\theta,\Sigma),\qquad n=1,\ldots,N. \tag{5.9}$$

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 contamination 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.



(a)

(b)

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. Classic 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 (the latter computed on clean data), averaged over 5 trials. Solid lines display the median KL divergence, with shaded areas showing 25th and 75th percentiles of KL divergence.

All computations involved in the coreset construction and posterior evaluation in this experiment can be performed in closed form. We apply the minibatch scheme of Algorithm 2, sampling from the exact coreset posterior over gradient estimation. The used (β-)likelihood equations are outlined in Appendix B.1.1. For all coreset methods, constructions are repeated for up to $M = 200$ iterations, with learning rate $\gamma_t = t^{-1}$. 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. Moreover, as opposed to the Gaussian experiment of the previous chapter, here we select a simpler hyperparameter selection scheme with constant initial learning rate over the entire range of coreset sizes, which in our settings allows SparseVI and β-Cores to reach their maximum posterior approximation quality at approximately 60 coreset points, and causes a slight increase in KL beyond this size.

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, and will be further discussed in Section 5.3.5.

5.3.2 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 Appendix B.3). 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,

$$y_n | x_n, \theta \sim \mathsf{Bern}\left(\frac{1}{1 + e^{-z_n^T \theta}}\right), \qquad z_n := \begin{bmatrix} x_n \\ 1 \end{bmatrix}. \tag{5.10}$$

The closed form of β-likelihood terms required in our construction is computed in Appendix B.1.2.

Data corruption is simulated by generating unstructured outliers in the input and output space similarly to (Futami et al., 2018): 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 the Laplace approximation to efficiently draw samples from the (non-conjugate) coreset posterior, while over evaluation coreset posterior samples are obtained via NUTS (Hoffman and Gelman, 2014). We evaluate the accuracy over the test set, predicting labels according to the maximum log-likelihood rule for θs sampled from the coreset posterior distribution. The 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 and learning rates γ_t were chosen via cross-validation.

Fig. 5.2 illustrates that β-Cores shows competitive performance with the classical 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 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).

5.3.3 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

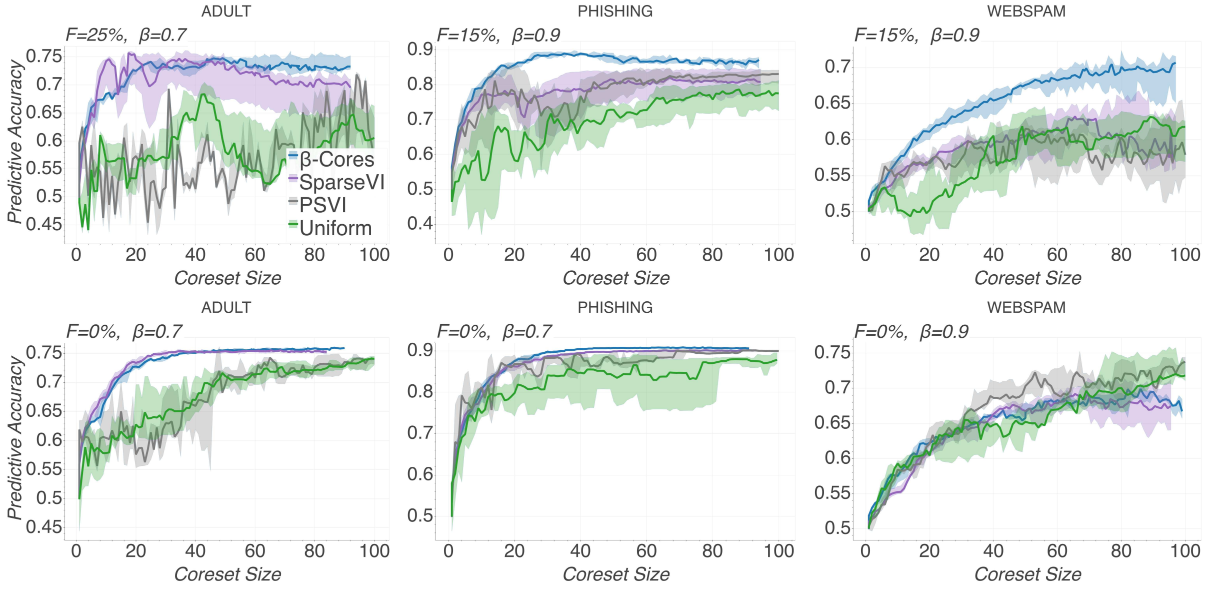


Figure 5.2: Predictive accuracy vs coreset size for logistic regression experiments over 10 trials on 3 large-scale datasets. Solid lines display the median accuracy, with shaded areas showing 25th and 75th percentiles. Dataset corruption rate F, and β value used in β-Cores for each experiment are shown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

regression on the representation of the last layer of a deterministic neural network feature extractor (Snoek et al., 2015; Riquelme et al., 2018; Pinsler et al., 2019). The corresponding statistical model is as follows

$$(y_n)_{n=1}^N = \theta^T z(x_n) + \epsilon_n, \quad (\epsilon_n)_{n=1}^N \sim \mathcal{N}(0, \sigma^2). \tag{5.11}$$

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 Appendix B.1.3. Input and output related outliers are simulated as in Section 5.3.2, 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 Appendix B.3). 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 (Duchi et al., 2010; McMahan and Streeter, 2010; Duchi et al., 2011). 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

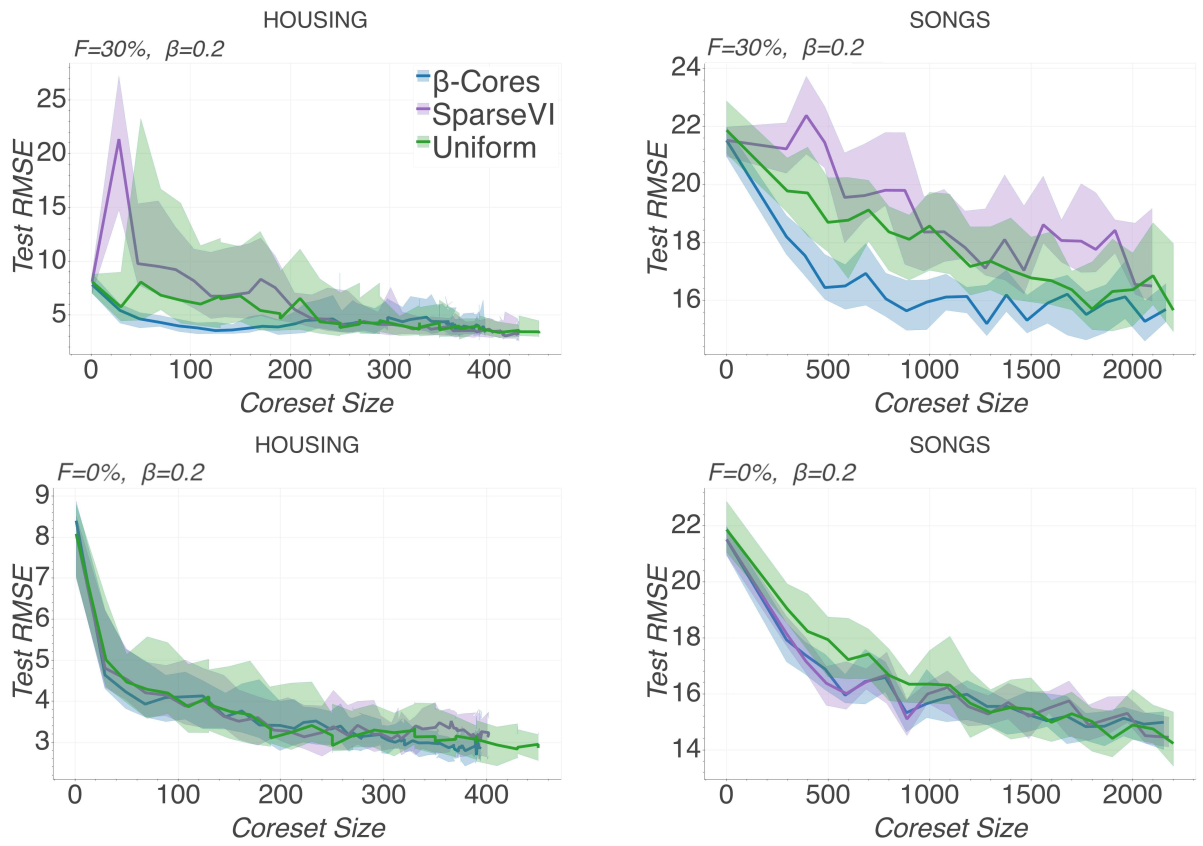


Figure 5.3: Test RMSE vs coreset size for neural linear regression experiments averaged over 30 trials. Solid lines display the median RMSE, with shaded areas showing 25th and 75th percentiles. Dataset corruption rate F, and β value used in β-Cores for each experiment are shown on the figures. The bottom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

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.

5.3.4 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 accessed over inference. Budgeted inference can be motivated by several practical considerations: 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 (Balle et al., 2019). 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 (Pinsler et al., 2019; Vahidian et al., 2020), and hence should be prioritised over data collection.

In this study we use a subset of more than $60K$ datapoints from the HospitalReadmissions dataset (for further details see Appendix B.3). 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 5.3.2, 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 ranks. Shapley value is a concept originating in cooperative game theory (Shapley, 1953), which has recently found applications in data valuation and outliers detection (Ghorbani and Zou, 2019). 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 estimator which samples $5K$ possible permutations of groups, and for each permutation it computes marginals for coalitions formed by the first 20 groups.[1]

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 the 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.

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

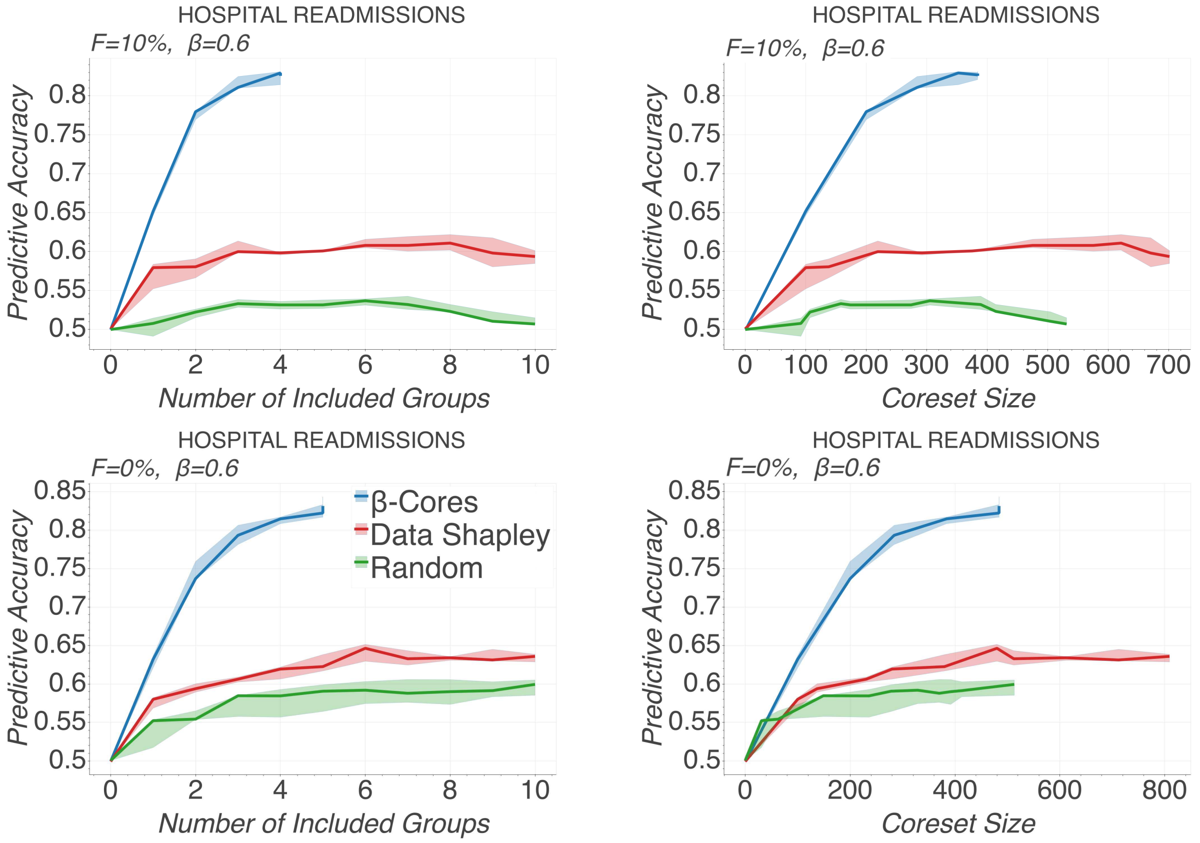


Figure 5.4: Predictive accuracy against number of groups (left) and number of datapoints (right) selected for inference. Compared group selection shemes are β-Cores, selection according to Shapley values based ranking, and random selection. The experiment is repeated over 5 trials, on a contaminated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

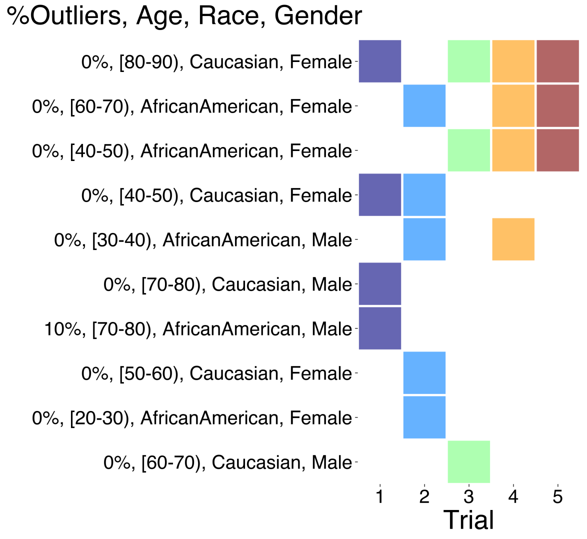


Figure 5.5: Attributes of selected groups after running 10 iterations of β-Cores with $\beta = 0.6$ on the contaminated HospitalReadmissions dataset (repeated over 5 random trials).

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, hence the later are favoured. 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 retain redundancy in the selected data subset.

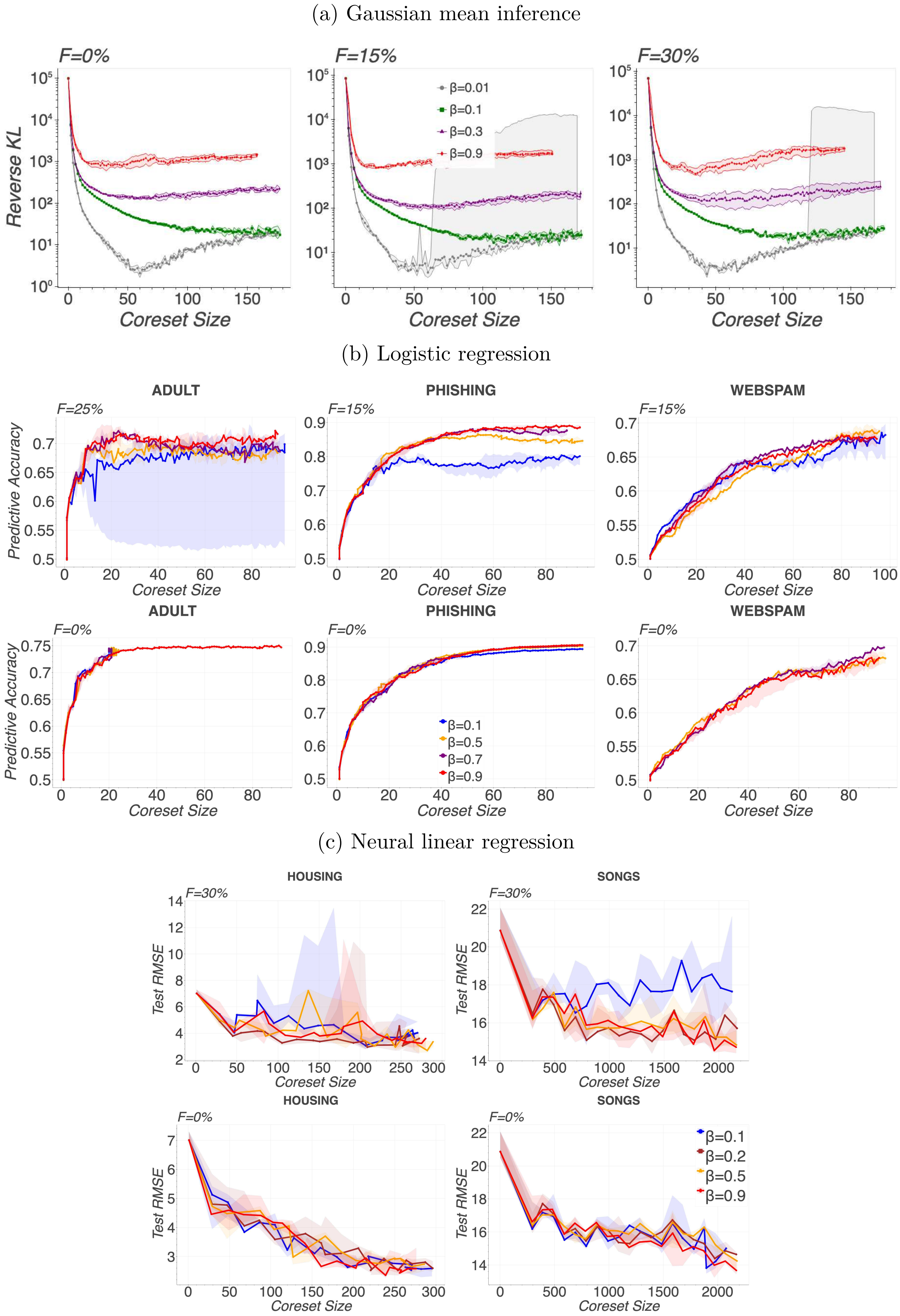
5.3.5 Effects of varying the robustness hyperparameter

In this section we perform an empirical analysis of the effects on robustness of inference that can be caused by varying the value of the divergence hyperparameter $\beta \in (0, 1)$. As observed in Fig. 5.6a, in the case of Gaussian mean inference under structured contamination, setting β to large values ($\beta \geq 0.3$) implies more conservative summarization schemes and more rigid coreset posteriors, that do not allow achieving optimal approximation quality; however these scheme also enable maintaining similar performance and small variance across trials for increasing size of the contaminated component. For smaller βs, the KL divergence between the approximate and the true posterior can reach lower minima; nonetheless, eventually the coreset quality might present larger variance, as the summarization becomes prone to adding outliers. At the remaining experiments, Figs. 5.6b and 5.6c, where inference takes place in the presence of unstructured outliers, the effects of varying the robustness hyperparameter are less pronounced. More noticeably, the remark of increased variance for small β remains valid with observable effects both in the logistic and the neural linear regression experiments.

5.4 Summary & discussion

In this chapter, 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 approximate inference under the β-divergence, we developed a greedy 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.

Further directions include developing more methods for adaptive 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). Moreover, future experimentation may consider stronger adversarial settings where summaries are initialized to data subsets that already contain outliers.



(a) Gaussian mean inference

(b) Logistic regression

(c) Neural linear regression

Figure 5.6: Predictive performance of β-Cores for varying values of the robustness hyperparameter β. At each experiment, results are averaged over 5 trials. Solid lines display the median of the predictive metric, with shaded areas showing the corresponding 25th and 75th percentiles.

Chapter 6

Conclusions

In this thesis, we have presented three original pieces of work drawing on one of the fundamental research problems in large-scale machine learning: *finding scalable dataset reductions under constraints commonly arising in real-world data analysis applications.* Our premise has been that principled dataset summarization methods can be harsenessed to enable efficient approximations for the purposes of large-scale data analysis without compromising requirements of privacy and robustness. In this section, we briefly recap our key contributions and suggest directions for future research.

6.1 Summary

6.1.1 Privacy loss of coarsened structured data

Reducing the information content and removing explicit identifiers from sensitive datasets prior to public release offers an illusion of privacy. In Chapter 3, we examined a large collection of longitudinal mobility traces recorded by smartphone devices. We converted each pseudonymised user trace record to a truncated graph, which retained the transition patterns among user's most frequent locations, and generated such representations over two different time windows spanning the entire period of tracking. Computing structural similarities via graph kernels allowed us to delevop a linkage attack, that was able to reidentify the anonymized mobility graphs at a 3.5× higher median success rate compared to random guessing. Our finding stressed that pseudonymisation and coarsening of data cannot protect data subjects against adversaries with access to the infomation of (nearly uniquely) identifying substructures—hence, further elaborating on data reduction techniques that adhere to formal privacy guarantees is required.

6.1.2 Privacy-preserving Bayesian coresets in high dimensions

In Chapter 4, we developed a novel construction for Bayesian coresets. We extended the existing sparse variational inference framework by introducing a richer family of scalable posterior approximations which, instead of points of the original dataset, makes use of learnable pseudodata that act as variational parameters optimized to summarize the full data likelihood. Our variational approximation enabled effective summarization that is not limited by data dimensionality, unlike previous constructions. Moreover, our coreset construction is amenable both to an incremental, as well as a batch black-box optimization scheme, offering computational time savings compared to state-of-the-art sparse VI methods. Finally, the use of synthetic data, combined with the subsampled Gaussian mechanism, allowed us to yield differentially private dataset summarization. We demonstrated applications of inference over a diverse set of Bayesian models, including Gaussian mean estimation, linear and logistic regression, showing the advantages in data posterior approximation offered by our approach.

6.1.3 Robust Bayesian coresets under misspecification

In Chapter 5, we designed a Bayesian coreset construction suitable for summarizing datasets that potentially depart from our statistical model assumptions—as often can be the case in practice, due to observations containing outliers, and/or being subjected to contamination. We proposed an incremental scheme that attains a sparse approximation of a robust generalized Bayesian posterior defined via the β-divergence, while discerning and retaining a representative small part of the data inliers instead of the full dataset. Further to offering scalability and reducing data redundancy, our scheme provided a unified and highly-automated solution to the important question of detecting and removing harmful datapoints prior to inference. We evaluated our technique on clean and contaminated data over a range of applications, including Gaussian mean inference, Bayesian linear regression, neural linear regression, and selection of informative data subpopulations' combinations, demonstrating reliable posteriors and predictive performance in all examined test cases.

6.2 Future research directions

The summarization frameworks presented in this dissertation allow numerous probabilistic models to be tractably and reliably deployed in practice. Yet they allude to a realm of so far unexplored research questions, some of which we overiew in the remainder of this section, thus concluding the thesis.

6.2.1 Coresets for models with structured likelihoods

Our variational formulations for coreset construction Eqs. (4.7) and (5.2) use the assumption that the data likelihood function gets factorised as a product of individual datapoint potentials. To the best of our knowledge, the idea of constructing coresets has not yet been used for inference in models with structured likelihood functions, including time-series and point processes. Recent results on parameter estimation for Hawkes processes using uniform downsampling (Li and Ke, 2019) indicate important improvements in efficiency when learning in massive temporal event sequences via reducing data, even without explicitly optimizing for redundancy in the extracted data subsets.

6.2.2 Implicit differential privacy amplification of data-dependent compressions

In Chapter 4 we presented an optimization scheme that yields Bayesian coreset constructions under explicit differential privacy quarantees. As discussed in Section 2.6, a known result in DP literature is that incorporating random sampling in data analysis has implicit privacy amplification effects, i.e. that an algorithm has higher privacy guarantees when run on a random subset of the datapoints instead of the full dataset (Li et al., 2012; Beimel et al., 2013; Bassily et al., 2014; Abadi et al., 2016). More recently, Balle et al. (2018) presented a unifying methodology that utilises couplings and divergences to reason about DP amplification effects of several random sampling methods (including Poisson subsampling and sampling with/without replacement), under different data neighbouring relations.

Existing research makes a common assumption that simplifies privacy analysis, but is violated in the case of coresets: the sampling distribution is data-independent. It remains an open question whether generalizations of existing approaches can be used to argue about implicit DP amplification when replacing a privacy-sensitive dataset with a coreset—in primitive schemes, coreset construction simply takes the form of importance sampling (Bachem et al., 2017). Investigating DP amplification under data-dependent sampling is a direction with far-reaching implications, that can contribute to tighter privacy analysis, not only in the case of coresets, but more broadly in all machine learning applications involving importance sampling, which is already a cornerstone of many state-of-the-art stochastic learning methods.

6.2.3 Human-centric summaries for scalable inference

In Chapter 4 we presented a method utilising learnable batches of pseudodata to summarize a much larger dataset. Naturally this coreset construction bears the potential of reducing the interpretability of learned pseudodata, since coreset points are now not a subset of the original dataset, but rather the result of a likelihood-specific optimization routine. To remedy related concerns, further interpretability constraints can be explicitly incorporated in the optimization formulation of pseudocoreset variational inference of Eq. (4.7).

Beyond the quest of interpretability, additional research is required in examining other desiderata in human-centric inference. To name a few, deletion-robustness is often sought or imposed on methods for large-scale data analysis (Mirzasoleiman et al., 2017; Ginart et al., 2019): user's *right-to-be-forgotten* is related to imposing bounds on the effects of removing an individual datapoint from an existing dataset, and can be approximately satisfied under differential privacy. Moreover, group *fairness* is one more topic that necessitates further investigation: without special treatment, reducing datasets will potentially transfer existing inequalities across groups in the derived summary, hence a different construction may be sought when aiming to ameliorate unfairness in scalable inference.

6.2.4 Compressing datasets for meta-learning

A distinguishing feature of human intelligence is the ability to adaptively learn new tasks on the basis of prior acquired experience, rather than learning each new task from scratch. Although Bayesian coresets have been originally proposed as an approach for efficient model-specific inference, it seems reasonable to inquire whether sparse dataset summaries can be also useful in meta-learning, i.e. settings where we aim to learn over a variety of tasks using few training examples per task. Recent work has shown that model-agnostic meta-learning (Finn et al., 2017) admits reformulations as a hierarchical Bayesian model, and gets performance improvements via expressing uncertainty (Finn et al., 2018; Grant et al., 2018). Apart from offering another avenue for scalability in meta-learning, extracting versatile summaries from a universe of data domains simulates more closely the situations that a human faces when organizing experience and knowledge for learning in the real world; hence, designing coresets in this context could contribute novel insights into the nature of general intelligence.

Appendix A

Supplement for Bayesian Pseudocoresets

A.1 Proof of Proposition 16

In the setting of Proposition 16, 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

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

and

$$\Sigma_w = \frac{I_d}{1 + \left(\sum_{n=1}^{N} w_n \right)}, \quad \mu_w = \Sigma_w \left(\sum_{n=1}^{N} w_n X_n \right). \tag{A.2}$$

Proof of Proposition 16. By Eqs. (A.1) and (A.2),

$$\begin{aligned} \mathrm{D_{KL}}\left(\pi_w || \pi\right) =& \frac{1}{2} \left[\log \frac{|\Sigma_1|}{|\Sigma_w|} - d + \operatorname{tr}\left(\Sigma_1^{-1} \Sigma_w \right) + (\mu_1 - \mu_w)^T \Sigma_1^{-1} (\mu_1 - \mu_w) \right] \\ =& \frac{1}{2} \left[-d \log \left(\frac{1+N}{1 + \sum_{n=1}^{N} w_n} \right) - d + d \left(\frac{1+N}{1 + \sum_{n=1}^{N} w_n} \right) + (\mu_1 - \mu_w)^T \Sigma_1^{-1} (\mu_1 - \mu_w) \right]. \end{aligned} \tag{A.3}$$

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

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

Thus,

$$\mathrm{D_{KL}}\left(\pi_w || \pi\right) \geq \frac{1}{2} (\mu_1 - \mu_w)^T \Sigma_1^{-1} (\mu_1 - \mu_w). \tag{A.4}$$

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

$$Y = [X_n : n \notin \mathcal{I}] \in \mathbb{R}^{d\times(N-M)}, \quad X = [X_n : n \in \mathcal{I}] \in \mathbb{R}^{d\times M}, \tag{A.5}$$

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

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}\left(\pi_w \| \pi\right) \geq & \frac{1}{2(N+1)} 1^T Y^T Y 1 + 1^T Y^T X \left(\frac{1}{N+1} - \frac{w}{1+1^T w}\right) \\ & + \frac{N+1}{2}\left(\frac{1}{N+1} - \frac{w}{1+1^T w}\right)^T X^T X \left(\frac{1}{N+1} - \frac{w}{1+1^T w}\right). \end{aligned} \tag{A.6}$$

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 $\mathcal{I}$,

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{w_{\mathcal{I}}^\star} \| \pi\right) \geq \frac{1^T Y^T \left(I - X(X^T X)^{-1} X^T\right) Y 1}{2(N+1)}. \tag{A.7}$$

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 $\mathcal{I}$

$$\mathrm{D}_{\mathrm{KL}}\left(\pi_{w_{\mathcal{I}}^\star} \| \pi\right) \geq \frac{N-M}{2(N+1)} Z_{\mathcal{I}}, \quad Z_{\mathcal{I}} \sim \chi^2(d-M). \tag{A.8}$$

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

$$\begin{aligned} \mathbb{P}\left(\min_{\mathcal{I}\subseteq[N], |\mathcal{I}|=M} Z_{\mathcal{I}} > \epsilon\right) \geq & 1 - \binom{N}{M} \mathbb{P}\left(Z_{\mathcal{I}} \leq \epsilon\right) \\ = & 1 - \binom{N}{M} F_{d-M}(\epsilon), \end{aligned} \tag{A.9}$$

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

A.2 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.

A.2.1 Weights gradient

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

$$\nabla_w \mathrm{D_{KL}} = -\nabla_w \log Z(u,w) - \nabla_w \mathbb{E}[f(\theta)^T 1] + \nabla_w \mathbb{E}[\tilde{f}(\theta)^T w]. \tag{A.10}$$

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

$$\begin{aligned}\nabla_w \mathbb{E}\left[a(\theta)\right] =& \int \nabla_w \left(\exp\left(w^T \tilde{f}(\theta) - \log Z(u,w)\right)\right) a(\theta)\pi_0(\theta)\mathrm{d}\theta \\ =& \mathbb{E}\left[\left(\tilde{f}(\theta) - \nabla_w \log Z(u,w)\right) a(\theta)\right].\end{aligned} \tag{A.11}$$

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

$$\begin{aligned}\nabla_w \log Z(u,w) =& \int \frac{1}{Z(u,w)} \nabla_w \left(\exp\left(w^T \tilde{f}(\theta)\right)\right) \pi_0(\theta)\mathrm{d}\theta \\ =& \mathbb{E}\left[\tilde{f}(\theta)\right].\end{aligned} \tag{A.12}$$

Combining, we have

$$\nabla_w \mathbb{E}\left[a(\theta)\right] = \mathbb{E}\left[\left(\tilde{f}(\theta) - \mathbb{E}\left[\tilde{f}(\theta)\right]\right) a(\theta)\right]. \tag{A.13}$$

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

$$\nabla_w \mathbb{E}\left[a(\theta)\right] = \mathrm{Cov}\left[\tilde{f}(\theta), a(\theta)\right]. \tag{A.14}$$

The gradient with respect to w in Eq. (4.9) follows by substituting $1^T f(\theta)$ and $w^T \tilde{f}(\theta)$ for $a(\theta)$ and using the product rule.

A.2.2 Location gradients

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

$$\nabla_{u_i} \mathrm{D_{KL}} = -\nabla_{u_i} \log Z(u,w) - \nabla_{u_i} \mathbb{E}[f(\theta)^T 1] + \nabla_{u_i} \mathbb{E}[\tilde{f}(\theta)^T w]. \tag{A.15}$$

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

$$\nabla_{u_i} \mathbb{E}\left[a(u,\theta)\right] = \int \nabla_{u_i} \left(\exp\left(w^T \tilde{f}(\theta) - \log Z(u,w)\right) a(u,\theta)\right) \pi_0(\theta)\mathrm{d}\theta. \tag{A.16}$$

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

$$\nabla_{u_i} \mathbb{E}\left[a(u,\theta)\right] = \mathbb{E}\left[\nabla_{u_i} a(u,\theta)\right] + \mathbb{E}\left[a(u,\theta)\left(w_i h(u_i,\theta) - \nabla_{u_i} \log Z(u,w)\right)\right]. \tag{A.17}$$

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

$$\nabla_{u_i} \log Z(u,w) = w_i \mathbb{E}\left[h(u_i,\theta)\right]. \tag{A.18}$$

Combining,

$$\nabla_{u_i} \mathbb{E}\left[a(u,\theta)\right] = \mathbb{E}\left[\nabla_{u_i} a(u,\theta)\right] + w_i \mathbb{E}\left[a(u,\theta)\left(h(u_i,\theta) - \mathbb{E}\left[h(u_i,\theta)\right]\right)\right]. \tag{A.19}$$

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

$$\nabla_{u_i} \mathbb{E}\left[a(u, \theta)\right] = \mathbb{E}\left[\nabla_{u_i} a(u, \theta)\right] + w_i \operatorname{Cov}\left[a(u, \theta), h(u_i, \theta)\right]. \tag{A.20}$$

The gradient with respect to u_i in Eq. (4.9) follows by substituting $f(\theta)^T 1$ and $\tilde{f}(\theta)^T w$ for $a(u, \theta)$.

A.3 Details on experiments

A.3.1 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 lower triangular matrix of the Cholesky decomposition of Σ, i.e. $\Sigma := QQ^T$. In order to compute the gradients in Eq. (4.9), we need expressions for $\operatorname{Cov}[f_n, f_m]$, $\operatorname{Cov}[\tilde{f}_n, f_m]$, $\operatorname{Cov}[h(u_i), f_n]$, and $\operatorname{Cov}[h(u_i), \tilde{f}_n]$. Following Campbell and Beronov (2019), we have that

$$\operatorname{Cov}[f_n, f_m] = v_n^T \Psi v_m + \frac{1}{2} \operatorname{tr} \Psi^T \Psi \tag{A.21}$$

$$\operatorname{Cov}[\tilde{f}_n, f_m] = \tilde{v}_n^T \Psi v_m + \frac{1}{2} \operatorname{tr} \Psi^T \Psi. \tag{A.22}$$

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

$$h(u_i) = -\Sigma^{-1}(u_i - \theta) \tag{A.23}$$

$$\mathbb{E}\left[h(u_i)\right] = -\Sigma^{-1}(u_i - \mu_{u,w}), \tag{A.24}$$

We have (up to a constant) that

$$f_n = -\frac{1}{2}(x_n - \theta)^T \Sigma^{-1}(x_n - \theta) \tag{A.25}$$

$$\mathbb{E}\left[f_n\right] = -\frac{1}{2} \operatorname{tr} \Psi - \frac{1}{2}\|v_n\|^2. \tag{A.26}$$

Thus using the above definitions,

$$\mathbb{E}\left[h(u_i)\right] \mathbb{E}\left[f_n\right] = \frac{(\operatorname{tr} \Psi + \|v_n\|^2)}{2} Q^{-T} \tilde{v}_i. \tag{A.27}$$

Next,

$$\mathbb{E}\left[h(u_i) f_n\right] = \frac{1}{2} \Sigma^{-1} \mathbb{E}\left[(u_i - \theta)(x_n - \theta)^T \Sigma^{-1}(x_n - \theta)\right]. \tag{A.28}$$

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

$$\mathbb{E}\left[h(u_i) f_n\right] = \frac{1}{2} Q^{-T} \mathbb{E}\left[(\tilde{v}_i - z)(v_n - z)^T (v_n - z)\right]. \tag{A.29}$$

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

$$\mathbb{E}\left[h(u_i)f_n\right] = \frac{\|v_n\|^2 + \operatorname{tr}\Psi}{2} Q^{-T}\tilde{v}_i + Q^{-T}\Psi v_n. \tag{A.30}$$

Therefore,

$$\operatorname{Cov}[h(u_i), f_n] = Q^{-T}\Psi v_n, \tag{A.31}$$

and likewise,

$$\operatorname{Cov}[h(u_i), \tilde{f}_n] = Q^{-T}\Psi \tilde{v}_n. \tag{A.32}$$

A.3.2 Bayesian linear regression

A.3.2.1 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

$$f_m(\theta) = -\frac{1}{2\sigma^2}\left(y_m - \theta^T u_m\right)^2. \tag{A.33}$$

Hence, we obtain for the pseudocoreset posterior

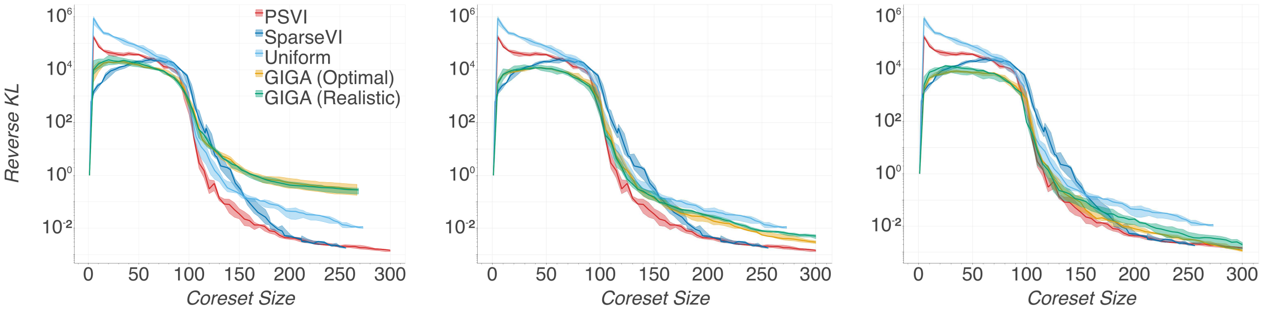
$$\pi_{u,w} = \mathcal{N}(\mu_{u,w}, \Sigma_{u,w}), \quad \text{where} \tag{A.34}$$

$$\Sigma_{u,w} = \left(\sigma_0^{-2} I + \sigma^{-2}\sum_{m=1}^{M} w_m u_m u_m^T\right)^{-1}, \quad \mu_{u,w} = \Sigma_{u,w}\left(\sigma_0^{-2} I \mu_0 + \sigma^{-2}\sum_{m=1}^{M} w_m y_m u_m\right). \tag{A.35}$$

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 $\{0.1, 1, 10\}$.

A.3.2.2 Additional plots

Here we present some more plots demonstating the dependence of Hilbert coresets' approximation quality on the dimension of random projections in the Bayesian linear regression setting presented in Fig. 4.2c. We remind that the 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. A.1, getting higher projection dimension enables better posterior approximation in the problem for both GIGA (Optimal) and GIGA (Realistic). However, PSVI remains competitive in the



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

Figure A.1: 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 (Campbell and Beronov, 2019).

A.3.3 Bayesian logistic regression

A.3.3.1 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

$$y_n | x_n, \theta \sim \mathsf{Bern}\left(\frac{1}{1+e^{-z_n^T\theta}}\right) \quad z_n := \begin{bmatrix} x_n \\ 1 \end{bmatrix}. \tag{A.36}$$

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

$$\begin{aligned} f_n := f(x_n, y_n|\theta) =& \mathbb{1}[y_n = -1] \log\left(1 - \frac{1}{1+e^{-z_n^T\theta}}\right) - \mathbb{1}[y_n = 1] \log\left(1 + e^{-z_n^T\theta}\right) \\ =& -\log\left(1 + \exp(-y_n z_n^T \theta)\right). \end{aligned} \tag{A.37}$$

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

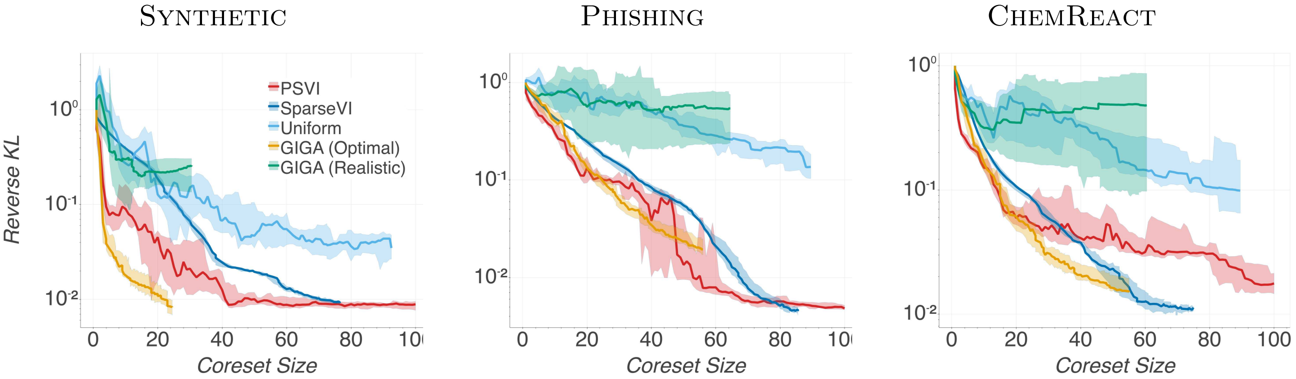
$$\nabla_{x_n} f_n = \frac{e^{-y_n z_n^T\theta}}{1+e^{-y_n z_n^T\theta}} y_n \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}. \tag{A.38}$$

A.3.3.2 Datasets description

For logistic regression experiments, we used subsampled and full versions of datasets presented in Table A.1: a synthetic dataset with $x \in \mathbb{R}^2$ sampled i.i.d. from a $\mathcal{N}(0, I)$ and $y \in \{-1, 1\}$ 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

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 A.1: Details for datasets used in logistic regression experiments.


Synthetic

Phishing

ChemReact

Figure A.2: Comparison of (pseudo)coreset approximate posterior quality vs coreset size for logistic regression over 10 trials.

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/libsvmtools/datasets/binary.html, http://komarix.org/ac/ds, https://www.kaggle.com/c/santander-customer-transaction-prediction/data, and https://github.com/mdeff/fma.

A.3.3.3 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. A.2 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.

A.3.3.4 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.

A.3.3.4.1 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 $\{0.1, 1, 10\}$.

A.3.3.4.2 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 the 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 (Dwork and Roth, 2014). In our experiment, we used the AdaGrad optimizer, 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.

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

A.3.3.5 Additional plots

A.3.3.5.1 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. A.3 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 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.

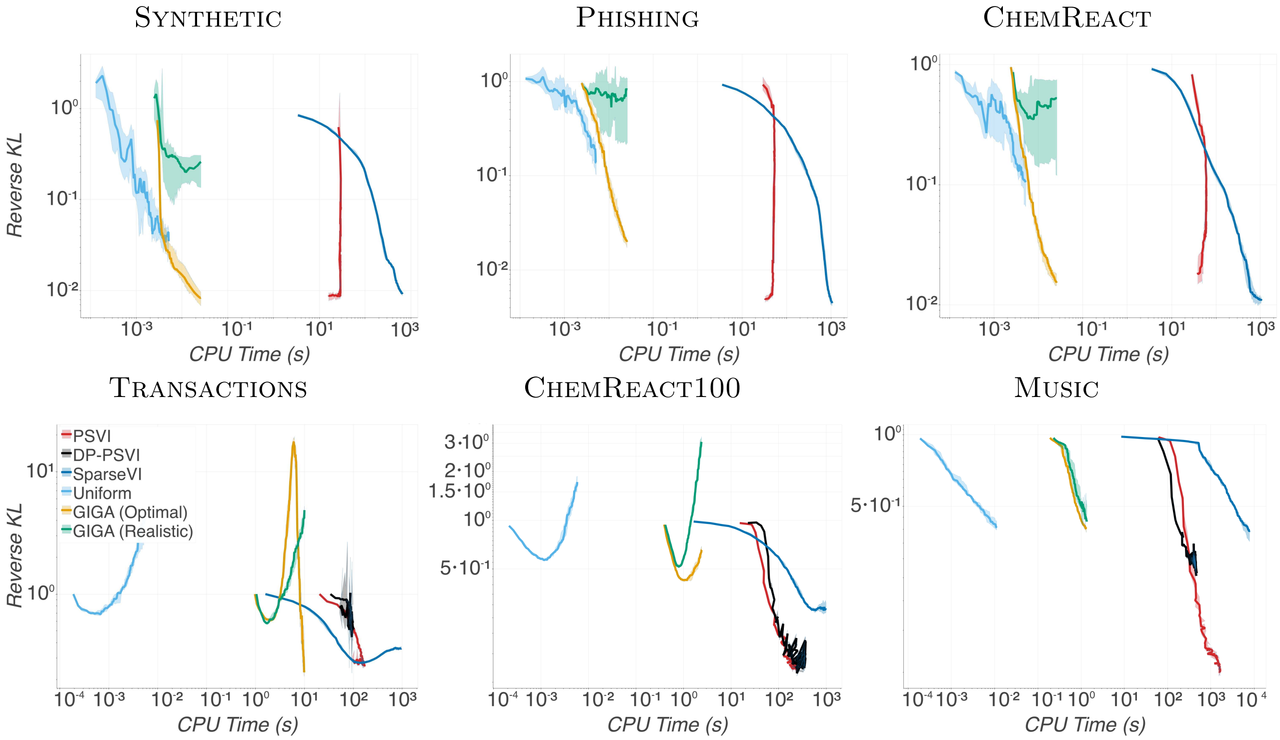


Figure A.3: Comparison of PSVI and SparseVI approximate posterior quality vs CPU time requirements for logistic regression experiment of Section 4.4.

A.3.3.5.2 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. A.4 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).

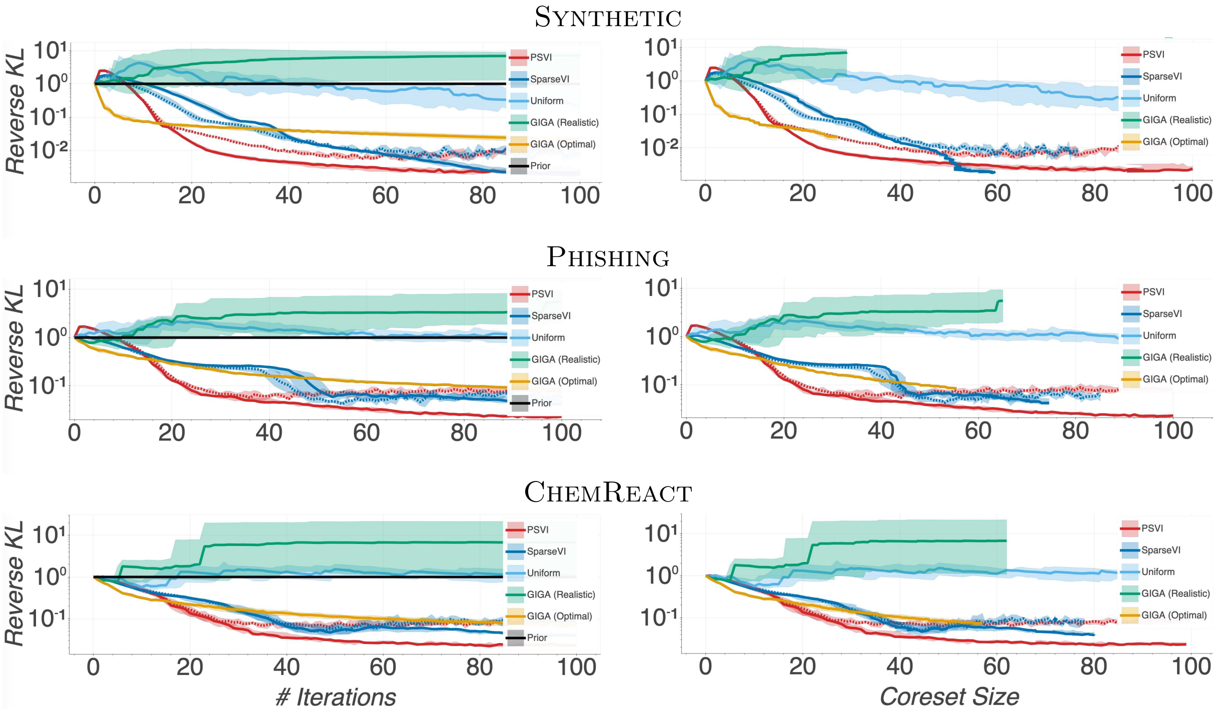


Figure A.4: Comparison of incremental PSVI and SparseVI approximate posterior quality vs iterations of incremental construction (*left*) and coreset size (*right*) for logistic regression on small-scale experiment. With dashed lines is displayed the posterior quality achieved by incremental PSVI and SparseVI constructions using gradients computed on random data subsets of size 256.

Appendix B

Supplement for β-Cores

B.1 Models

In this section we present the derivations of β-likelihood terms Eqs. (2.23) and (2.24) required over the β-Cores constructions for the statistical models of our experiments.

B.1.1 Gaussian likelihoods

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

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

and, by simple calculus (see also Samek et al. (2013)),

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

Hence, omitting the constant term due to the shift-invariance of potentials entering Algorithm 2, we get up to proportionality

$$f_n(\mu) \propto \frac{1}{\beta} \exp\left(-\frac{\beta}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right). \tag{B.3}$$

B.1.2 Logistic regression likelihoods

Log-likelihood terms of individual datapoints are given as follows

$$\log \pi(y_n|x_n,\theta) = -\log\left(1+e^{-y_n z_n^T\theta}\right). \tag{B.4}$$

Substituting to Eq. (2.24), for the β-likelihood terms we get

$$f_n(\theta) \propto -\frac{1}{\beta}\left(1+e^{-y_n z_n^T\theta}\right)^{-\beta} + \frac{1}{\beta+1}\left(\left(1+e^{-z_n^T\theta}\right)^{-(\beta+1)} + \left(1+e^{z_n^T\theta}\right)^{-(\beta+1)}\right). \tag{B.5}$$

B.1.3 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

$$f_n(\theta) = -\frac{1}{2\sigma^2}\left(y_n - \theta^T z(x_n)\right)^2. \tag{B.6}$$

Assuming a prior $\theta \sim \mathcal{N}(\mu_0, \sigma_0^2 I)$, the coreset posterior is a Gaussian $\pi_w(\theta) = \mathcal{N}(\mu_w, \Sigma_w)$, with mean and covariance computable in closed form as follows

$$\Sigma_w := \left(\sigma_0^{-2} I + \sigma^{-2} \sum_{m=1}^{M} w_m z(x_m) z(x_m)^T\right)^{-1}, \tag{B.7}$$

$$\mu_w := \Sigma_w \left(\sigma_0^{-2} I \mu_0 + \sigma^{-2} \sum_{m=1}^{M} w_m y_m z(x_m)\right). \tag{B.8}$$

By substitution to Eq. (2.24) and omitting constants, the β-likelihood terms for our adaptive basis linear regression are written as

$$f_n(\theta) \propto e^{-\beta\left(y_n - \theta^T z(x_n)\right)^2/(2\sigma^2)}. \tag{B.9}$$

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

$$\begin{aligned}\pi(y_t|x_t, \mathcal{D}) &\approx \pi(y_t|x_t, \mathcal{C}) \\ &= \int \pi(y_t|x_t, \theta)\pi(\theta|\mathcal{C}) d\theta. \end{aligned} \tag{B.10}$$

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

$$\pi(y_t|x_t, \mathcal{C}) = \mathcal{N}\left(y_t; \mu_w^T z(x_t), \sigma^2 + z(x_t)^T \Sigma_w z(x_t)\right). \tag{B.11}$$

B.2 Characterization of Riemannian coresets' combinatorial optimization objective

When optimizing a set function, the property of submodularity is often deemed appealing as it can allow using fast greedy selection policies with provable suboptimality guarantees (Nemhauser et al., 1978; Bach, 2013). The optimization problem corresponding to our next datapoint selection step of Eq. (5.6) can be equivalently rewritten as follows

$$m^\star = \arg\max_{m \in [N]} -\mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta, w \leftarrow w \cup \{x_m\}} || \pi_\beta\right). \tag{B.12}$$

Table B.1: Logistic regression datasets

Dataset	d	N_{train}	N_{test}	#Pos. test data
Adult (Kohavi, 1996)	10	30,162	7,413	3,700
Phishing (Dua and Graff, 2017)	10	8,844	2,210	1,230
WebSpam (Wang et al., 2012)	127	126,185	13,789	6,907
HospitalReadmissions (Strack et al., 2014)	10	55,163	6,079	3,044

Hence—ignoring the coreset datapoints' reweighting step which is treated separately—it is of interest to characterize the properties of the objective function $d : 2^{|\mathcal{X}|} \to \mathbb{R}_{\leq 0}$

$$d(S) := -\mathrm{D}_{\mathrm{KL}}\left(\pi_{\beta, \frac{N}{M}\mathcal{I}_S} || \pi_\beta\right), \tag{B.13}$$

where S is set of M datapoints appearing with non-zero weight in the coreset.

Below we give a tight condition for submodularity via second-order differences, which captures its characteristic property of *diminishing returns* for increasing set size.

Definition 20 (Submodularity)**.** The set function d is submodular if and only if for all $S \subseteq \mathcal{X}$ and $x_j, x_k \in \mathcal{X} \setminus S$, we have $d(S \cup \{x_j\}) - d(S) \geq d(S \cup \{x_j, x_k\}) - d(S \cup \{x_k\})$.

In the next proposition, we demonstrate a problem instance where the necessary and sufficient condition of Definition 20 is violated for d considered in Eq. (B.13), hence proving that our objective is *non-submodular* under no further assumptions.

Proposition 21. *The set function d of Eq.* (B.13) *is non-submodular.*

Proof. For convenience let's focus on the case of Gaussian mean inference for the classical Bayesian posterior ($\beta \to 0$), where the objective can be handily written in closed form. Similar arguments will in principle carry over for arbitrary βs and statistical models. We recall from Eq. (A.3) that

$$\begin{aligned} d(S) &= -\frac{1}{2}\left[-d \log\left(\frac{1+N}{1+\frac{N}{M}||\mathcal{I}_S||_1}\right) - d + d\left(\frac{1+N}{1+\frac{N}{M}||\mathcal{I}_S||_1}\right) + (1+N)(\mu_1 - \mu_w)^T(\mu_1 - \mu_w)\right] \\ &= -\frac{1}{2}(1+N)||\mu_1 - \mu_w||_2^2, \end{aligned} \tag{B.14}$$

where

$$\mu_1 = \frac{1}{1+N}\sum_{n=1}^{N} x_n, \quad \mu_w = \frac{1}{1+N}\frac{N}{M}\sum_{x_i \in S} x_i. \tag{B.15}$$

Let's consider a set of observations containing two mirrored datapoints $x_0, -x_0$, such that $x_0 \neq \mu_1$. Then clearly

$$\begin{aligned} &d(S \cup \{x_0\}) - d(S) - d(S \cup \{x_0, -x_0\}) + d(S \cup \{-x_0\}) \\ =&d(S \cup \{x_0\}) + d(S \cup \{-x_0\}) - 2d(\mathcal{X}) = d(S \cup \{x_0\}) + d(S \cup \{-x_0\}) < 0, \end{aligned} \tag{B.16}$$

where we have used the fact that $d(S) = d(\mathcal{X}) = 0$. □

Table B.2: Neural linear regression datasets

Dataset	d	N_{train}	N_{test}
Housing (Dua and Graff, 2017)	13	446	50
Songs (Dua and Graff, 2017)	90	463,711	51,534

B.3 Datasets details

The benchmark datasets used in logistic regression (including subpopulations' selection) and neural linear regression experiments are detailed in Tables B.1 and B.2 respectively, 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 B.1).

Original versions of the six benchmark datasets were respectively downloaded from the following URLs: http://archive.ics.uci.edu/ml/datasets/Adult, https://archive.ics.uci.edu/ml/datasets/Phishing+Websites, https://www.cc.gatech.edu/projects/doi/WebbSpam Corpus.html, https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+ years+1999-2008, https://archive.ics.uci.edu/ml/machine-learning-databases/housing, and https://archive.ics.uci.edu

List of figures

List of tables

Nomenclature

Acronyms/Abbreviations

i.i.d.	Independent and identically distributed
w.r.t.	with respect to
CCDF	Complementary Cumulative Density Function
CDF	Cumulative Density Function
cf.	confer
cit.	cited
DP	Differentially Private
e.g.	exempli gratia
ELBO	Evidence Lower Bound
etc.	et cetera
i.e.	id est
iff	if and only if
KL	Kullback-Leibler
MC	Monte Carlo
NUTS	No-U-Turn Sampler
PCA	Principal Components Analysis
PL	Privacy Loss
PSVI	Pseudocoresets Sparse Variational Inference
RANDOM	Random Sampling Coreset
RBF	Radial Basis Function
RHS	right hand side

RMSE	Root-Mean-Square Error
s.t.	such that
VI	Variational Inference

Roman Symbols

$\mathcal{D}$	Dataset
$\mathcal{H}$	Hilbert space
$\mathcal{X}^N$ / $(\mathcal{X} \times \mathcal{Y})^N$	Data space of N unlabeled/labeled observations

Greek Symbols

$\delta(x)$	Dirac delta function: equals $+\infty$ iff $x = 0$, otherwise 0
ϵ	a random variable
Θ	Asymptotically tight upper and lower bound of complexity
ε	a very small non-negative constant
O	Upper bound of complexity

Other Symbols

$[N]$	$[1, \ldots, N]$
#	Number of
$\circ$	Composition of functions
$:=$	Defined as
$\langle \cdot\ ,\ \cdot \rangle$	Inner product
$\mathbb{P}$	Probability
D_β	β-divergence
d_β	β-cross-entropy
D_{KL}	Kullback-Leibler divergence
d_{KL}	Cross-entropy
$\propto$	Proportional to
$\sim$	Distributed as
$\mathbb{Z}$	Integer numbers

$\mathbb{N}$	Natural numbers
$\mathbb{R}$	Real numbers

Superscripts

$\widehat{\ }$	Empirical estimate
$\widetilde{\ }$	Computed on pseudodata

Distributions

χ^2	Chi-square distribution
$\mathsf{UnifSubset}$	Uniform subset distribution
$\mathcal{N}$	Normal distribution
$t(\nu)$	Student's t-distribution with ν degrees of freedom
Bern	Bernoulli distribution
Unif	Uniform distribution

Operators

$\mathbb{E}$	Expectation
Corr	Correlation
Cov	Covariance
diag	Matrix diagonal
tr	Matrix trace
Var	Variance

Mobile Data Abbreviations

CDR	Call Data Record
cid	Cell tower identifier
ID	Identifier
MAC	Media Access Control

Graphs

$\mathcal{G}$, G, V, E	Space of graphs, Graph, Vertices, Edges
DK	Deep graph kernels
SP	Shortest Path
WL	Weisfeiler-Lehman

Bibliography

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K. and Zhang, L. (2016). "Deep Learning with Differential Privacy". *ACM SIGSAC Conference on Computer and Communications Security* (cit. on pp. 30, 59, 65, 91).

Agarwal, P. K., Har-Peled, S., Varadarajan, K. R., et al. (2005). "Geometric approximation via coresets". *Combinatorial and computational geometry* 52, pp. 1–30 (cit. on p. 58).

Aggarwal, C. C. and Yu, P. S. (2008). "A General Survey of Privacy-Preserving Data Mining Models and Algorithms". Vol. 34 (cit. on p. 33).

Ağır, B., Huguenin, K., Hengartner, U. and Hubaux, J.-P. (2016). "On the privacy implications of location semantics". *Proceedings on Privacy Enhancing Technologies* 2016.4 (cit. on p. 34).

Agrawal, R., Campbell, T., Huggins, J. and Broderick, T. (2019). "Data-dependent compression of random features for large-scale kernel approximation". *International Conference on Artificial Intelligence and Statistics* (cit. on p. 58).

Amari, S.-i. (2016). *Information Geometry and Its Applications.* Springer (cit. on p. 18).

Andrieu, C., De Freitas, N., Doucet, A. and Jordan, M. I. (2003). "An introduction to MCMC for machine learning". *Machine Learning* 50.1-2, pp. 5–43 (cit. on p. 21).

Angelino, E., Johnson, M. J. and Adams, R. P. (2016). "Patterns of Scalable Bayesian Inference". *Found. Trends Mach. Learn.* (cit. on pp. 20, 72).

Bach, F. R. (2013). "Learning with Submodular Functions: A Convex Optimization Perspective". *Found. Trends Mach. Learn.* 6.2-3, pp. 145–373 (cit. on p. 104).

Bachem, O., Lucic, M. and Krause, A. (2015). "Coresets for nonparametric estimation—the case of DP-means". *International Conference on Machine Learning* (cit. on p. 58).

Bachem, O., Lucic, M. and Krause, A. (2017). *Practical Coreset Constructions for Machine Learning.* arXiv: 1703.06476 (cit. on p. 91).

Balle, B., Barthe, G. and Gaboardi, M. (2018). "Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences". *Advances in Neural Information Processing Systems* (cit. on p. 91).

Balle, B., Barthe, G., Gaboardi, M. and Geumlek, J. (2019). "Privacy Amplification by Mixing and Diffusion Mechanisms". *Advances in Neural Information Processing Systems.* Vol. 32 (cit. on p. 84).

Balog, M., Tolstikhin, I. and Schölkopf, B. (2018). "Differentially private database release via kernel mean embeddings". *International Conference on Machine Learning* (cit. on p. 12).

Banerjee, A., Merugu, S., Dhillon, I. S. and Ghosh, J. (2005). “Clustering with Bregman Divergences”. *J. Mach. Learn. Res.* 6, pp. 1705–1749 (cit. on p. 18).

Bardenet, R., Doucet, A. and Holmes, C. (2014). “Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach”. *International Conference on Machine Learning* (cit. on p. 21).

Barreno, M., Nelson, B., Joseph, A. D. and Tygar, J. D. (2010). “The security of machine learning”. *Machine Learning* (cit. on p. 72).

Bassily, R., Smith, A. and Thakurta, A. (2014). “Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds”. *IEEE Annual Symposium on Foundations of Computer Science* (cit. on p. 91).

Basu, A., Harris, I. R., Hjort, N. L. and Jones, M. C. (Sept. 1998). “Robust and efficient estimation by minimising a density power divergence”. *Biometrika* 85.3, pp. 549–559 (cit. on pp. 18, 26).

Beimel, A., Nissim, K. and Stemmer, U. (2013). “Characterizing the sample complexity of private learners”. *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pp. 97–110 (cit. on pp. 30, 91).

Berger, J. O., Moreno, E., Pericchi, L. R., Bayarri, M. J., Bernardo, J. M., Cano, J. A., De la Horra, J., Martín, J., Ríos Insúa, D., Betrò, B., et al. (1994). “An overview of robust Bayesian analysis”. *Test* 3.1, pp. 5–124 (cit. on p. 73).

Beyer, K. S., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). “When Is ”Nearest Neighbor” Meaningful?” *Proceedings of the 7th International Conference on Database Theory.* Springer-Verlag (cit. on p. 54).

Bhatia, K., Ma, Y.-A., Dragan, A. D., Bartlett, P. L. and Jordan, M. I. (2019). *Bayesian Robustness: A Nonasymptotic Viewpoint.* arXiv: 1907.11826 (cit. on p. 73).

Bhattacharya, S., Manousakas, D., Ramos, A. G. C., Venieris, S. I., Lane, N. D. and Mascolo, C. (2020). “Countering Acoustic Adversarial Attacks in Microphone-equipped Smart Home Devices”. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.2, pp. 1–24 (cit. on p. 15).

Biggio, B., Nelson, B. and Laskov, P. (2012). “Poisoning Attacks against Support Vector Machines”. *Proceedings of the 29th International Coference on International Conference on Machine Learning* (cit. on p. 72).

Bishop, C. M. (2006). *Pattern recognition and machine learning.* Springer (cit. on pp. 20, 22).

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017). “Variational Inference: A Review for Statisticians”. *Journal of the American Statistical Association* 112.518, pp. 859–877 (cit. on p. 22).

Bogdanov, P., Mongiovì, M. and Singh, A. K. (2011). “Mining Heavy Subgraphs in Time-Evolving Networks”. *Proceedings of the 11th IEEE International Conference on Data Mining* (cit. on p. 54).

Borgwardt, K. M. and Kriegel, H.-P. (2005). “Shortest-Path Kernels on Graphs”. *Proceedings of the Fifth IEEE International Conference on Data Mining* (cit. on p. 40).

Braverman, V., Feldman, D. and Lang, H. (2016). *New frameworks for offline and streaming coreset constructions*. arXiv: 1612.00889 (cit. on p. 58).

Campbell, T. and Beronov, B. (2019). “Sparse Variational Inference: Bayesian Coresets from Scratch”. *Advances in Neural Information Processing Systems* (cit. on pp. 23, 59, 61, 62, 65, 73, 74, 77, 78, 96, 98).

Campbell, T. and Broderick, T. (2018). “Bayesian Coreset Construction via Greedy Iterative Geodesic Ascent”. *International Conference on Machine Learning* (cit. on pp. 23, 58, 62, 65).

Campbell, T. and Broderick, T. (2019). “Automated Scalable Bayesian Inference via Hilbert Coresets”. *Journal of Machine Learning Research* 20.15 (cit. on pp. 23, 28, 58, 62, 73).

Chen, M., Gao, C. and Ren, Z. (2018). “Robust covariance and scatter matrix estimation under Huber’s contamination model”. *The Annals of Statistics* 46.5, pp. 1932–1960 (cit. on p. 78).

Chen, Y., Welling, M. and Smola, A. (2010). “Super-Samples from Kernel Herding”. *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence* (cit. on p. 61).

Chowdhury, S. and Mémoli, F. (2019). “The Gromov–Wasserstein distance between networks and stable network invariants”. *Information and Inference: A Journal of the IMA* 8.4, pp. 757–787 (cit. on p. 36).

Cichocki, A. and Amari, S.-i. (2010). “Families of Alpha- Beta- and Gamma- Divergences: Flexible and Robust Measures of Similarities”. *Entropy* 12.6, pp. 1532–1568 (cit. on p. 18).

Csató, L. and Opper, M. (2002). “Sparse on-line Gaussian processes”. *Neural computation* 14.3, pp. 641–668 (cit. on p. 58).

Dai, B., He, N., Dai, H. and Song, L. (2016). “Provable bayesian inference via particle mirror descent”. *Artificial Intelligence and Statistics* (cit. on p. 24).

Dawid, A. P., Musio, M. and Ventura, L. (2016). “Minimum scoring rule inference”. *Scandinavian Journal of Statistics* 43.1, pp. 123–138 (cit. on p. 25).

de Finetti, B. (1961). “The Bayesian approach to the rejection of outliers”. *Proceedings of the fourth Berkeley Symposium on Probability and Statistics* (cit. on pp. 13, 73).

de Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M. and Blondel, V. D. (2013). “Unique in the Crowd: The privacy bounds of human mobility.” *Scientific reports* 3 (cit. on pp. 32, 34).

De Mulder, Y., Danezis, G., Batina, L. and Preneel, B. (2008). “Identification via location-profiling in GSM networks”. *Proceedings of the 2008 ACM Workshop on Privacy in the Electronic Society* (cit. on pp. 32, 34).

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Steinhardt, J. and Stewart, A. (2019). “Sever: A Robust Meta-Algorithm for Stochastic Optimization”. *Proceedings of the 36th International Conference on Machine Learning* (cit. on p. 72).

Dickens, C., Meissner, E., Moreno, P. G. and Diethe, T. (2020). *Interpretable Anomaly Detection with Mondrian Pólya Forests on Data Streams*. arXiv: 2008.01505 (cit. on p. 72).

Donoho, D. L. (2000). “High-dimensional data analysis: The curses and blessings of dimensionality”. *AMS Math Challenges Lecture* 1 (cit. on p. 13).

Drineas, P. and Mahoney, M. (2005). “On the Nyström method for approximating a Gram matrix for improved kernel-based learning”. *Journal of Machine Learning Research* 6 (cit. on p. 58).

Dua, D. and Graff, C. (2017). *UCI Machine Learning Repository* (cit. on pp. 105, 106).

Duchi, J., Hazan, E. and Singer, Y. (2010). “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. *The 23rd Conference on Learning Theory* (cit. on p. 82).

Duchi, J., Hazan, E. and Singer, Y. (2011). “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. *Journal of Machine Learning Research* 12.61, pp. 2121–2159 (cit. on p. 82).

DuMouchel, W., Volinsky, C., Johnson, T., Cortes, C. and Pregibon, D. (1999). “Squashing flat files flatter”. *ACM Conference on Knowledge Discovery and Data Mining* (cit. on p. 58).

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I. and Naor, M. (2006a). “Our Data, Ourselves: Privacy via Distributed Noise Generation”. *International Conference on The Theory and Applications of Cryptographic Techniques* (cit. on p. 64).

Dwork, C., McSherry, F., Nissim, K. and Smith, A. (2006b). “Calibrating Noise to Sensitivity in Private Data Analysis”. *Conference on Theory of Cryptography* (cit. on p. 64).

Dwork, C., McSherry, F., Nissim, K. and Smith, A. (2006c). “Calibrating noise to sensitivity in private data analysis”. *Theory of Cryptography Conference*. Springer, pp. 265–284 (cit. on pp. 29, 35, 58).

Dwork, C. and Roth, A. (2014). “The algorithmic foundations of differential privacy”. *Foundations and Trends in Theoretical Computer Science* 9.3–4 (cit. on pp. 29, 30, 58, 65, 100).

Eguchi, S. and Kano, Y. (2001). *Robustifying maximum likelihood estimation*. Tech. rep. (cit. on pp. 18, 25).

Feldman, D., Faulkner, M. and Krause, A. (2011). “Scalable training of mixture models via coresets”. *Advances in Neural Information Processing Systems* (cit. on p. 58).

Feldman, D., Fiat, A., Kaplan, H. and Nissim, K. (2009). “Private Coresets”. *ACM Symposium on Theory of Computing* (cit. on pp. 12, 58).

Feldman, D. and Langberg, M. (2011). “A Unified Framework for Approximating and Clustering Data”. *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing* (cit. on p. 23).

Feldman, D., Volkov, M. and Rus, D. (2016). “Dimensionality reduction of massive sparse datasets using coresets”. *Advances in Neural Information Processing Systems* (cit. on p. 58).

Feldman, D., Xiang, C., Zhu, R. and Rus, D. (2017). “Coresets for Differentially Private k-Means Clustering and Applications to Privacy in Mobile Sensor Networks”. *International Conference on Information Processing in Sensor Networks* (cit. on pp. 12, 58).

Finn, C., Abbeel, P. and Levine, S. (2017). “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. *International Conference on Machine Learning* (cit. on p. 92).

Finn, C., Xu, K. and Levine, S. (2018). “Probabilistic model-agnostic meta-learning”. *Advances in Neural Information Processing Systems* (cit. on p. 92).

Frénay, B. and Verleysen, M. (2013). “Classification in the presence of label noise: a survey”. *IEEE transactions on neural networks and learning systems* 25.5 (cit. on p. 72).

Fujisawa, H. and Eguchi, S. (2008). “Robust Parameter Estimation with a Small Bias against Heavy Contamination”. *J. Multivar. Anal.*, pp. 2053–2081 (cit. on p. 25).

Futami, F., Sato, I. and Sugiyama, M. (2018). “Variational Inference based on Robust Divergences”. *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics* (cit. on pp. 24, 25, 73, 77, 81).

Gambs, S., Killijian, M.-O. and Núñez Del Prado Cortez, M. (2014). “De-anonymization Attack on Geolocated Data”. *J. Comput. Syst. Sci.* 80 (cit. on pp. 32, 34).

Geyer, C. J. (Nov. 1992). “Practical Markov Chain Monte Carlo”. *Statist. Sci.* 7.4, pp. 473–483 (cit. on p. 21).

Ghorbani, A. and Zou, J. (2019). “Data Shapley: Equitable Valuation of Data for Machine Learning”. *Proceedings of the 36th International Conference on Machine Learning* (cit. on pp. 72, 84).

Ghosh, A. and Basu, A. (2016). “Robust Bayes estimation using the density power divergence”. *Annals of the Institute of Statistical Mathematics* 68.2, pp. 413–437 (cit. on p. 25).

Gilks, W. R. (2005). “Markov Chain Monte Carlo”. *Encyclopedia of biostatistics* 4 (cit. on p. 21).

Ginart, A., Guan, M., Valiant, G. and Zou, J. Y. (2019). “Making AI Forget You: Data Deletion in Machine Learning”. *Advances in Neural Information Processing Systems* (cit. on p. 91).

Golle, P. and Partridge, K. (2009). “On the anonymity of home/work location pairs”. *International Conference on Pervasive Computing*. Springer (cit. on p. 32).

Grant, E., Finn, C., Levine, S., Darrell, T. and Griffiths, T. L. (2018). “Recasting Gradient-Based Meta-Learning as Hierarchical Bayes”. *International Conference on Learning Representations* (cit. on p. 92).

Grubbs, F. E. (1969). “Procedures for detecting outlying observations in samples”. *Technometrics* 11.1, pp. 1–21 (cit. on p. 12).

Gruteser, M. and Grunwald, D. (2003). “Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking”. *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*. ACM (cit. on p. 32).

Guhaniyogi, R. and Dunson, D. (2015). “Bayesian compressed regression”. *Journal of the American Statistical Association* 110.512 (cit. on p. 58).

Haussler, D. (1999). *Convolution kernels on discrete structures*. Tech. rep. Department of Computer Science, University of California at Santa Cruz (cit. on p. 39).

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013). “Stochastic Variational Inference”. *Journal of Machine Learning Research* 14, pp. 1303–1347 (cit. on pp. 22, 72).

Hoffman, M. D. and Gelman, A. (2014). “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. *Journal of Machine Learning Research* 15.1, pp. 1593–1623 (cit. on pp. 21, 64, 81).

Huber, P. J. (1992). “Robust estimation of a location parameter”. *Breakthroughs in statistics*. Springer, pp. 492–518 (cit. on p. 78).

Huber, P. J. and Ronchetti, E. M. (2009). *Robust statistics; 2nd ed.* Wiley Series in Probability and Statistics (cit. on pp. 13, 73).

Huggins, J., Campbell, T. and Broderick, T. (2016). “Coresets for Scalable Bayesian Logistic Regression”. *Advances in Neural Information Processing Systems* (cit. on pp. 23, 58, 73).

Huggins, J., Campbell, T., Kasprzak, M. and Broderick, T. (2020). “Validated Variational Inference via Practical Posterior Error Bounds”. *International Conference on Artificial Intelligence and Statistics* (cit. on p. 69).

Huggins, J., Adams, R. and Broderick, T. (2017). “PASS-GLM: polynomial approximate sufficient statistics for scalable Bayesian GLM inference”. *Advances in Neural Information Processing Systems* (cit. on p. 58).

Huszár, F. and Duvenaud, D. (2012). “Optimally-Weighted Herding is Bayesian Quadrature”. *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence* (cit. on p. 61).

Jacob, P. E., O’Leary, J. and Atchadé, Y. F. (2020). “Unbiased Markov chain Monte Carlo methods with couplings”. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 82.3, pp. 543–600 (cit. on p. 64).

Jälkö, J., Dikmen, O. and Honkela, A. (2017). “Differentially Private Variational Inference for Non-conjugate Models”. *Uncertainty in Artificial Intelligence* (cit. on pp. 59, 68).

Jewson, J., Smith, J. Q. and Holmes, C. (2018). “Principles of Bayesian inference using general divergence criteria”. *Entropy* 20.6, p. 442 (cit. on pp. 25, 26).

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. and Saul, L. K. (Nov. 1999). “An Introduction to Variational Methods for Graphical Models”. *Mach. Learn.* 37.2, pp. 183–233 (cit. on p. 22).

Kang, J. H., Welbourne, W., Stewart, B. and Borriello, G. (2005). “Extracting places from traces of locations”. *ACM SIGMOBILE Mobile Computing and Communications Review* 9 (cit. on p. 38).

Karger, D. R., Oh, S. and Shah, D. (2011). “Iterative learning for reliable crowdsourcing systems”. *Advances in Neural Information Processing Systems* (cit. on p. 72).

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S. and Smith, A. (2011). “What can we learn privately?” *SIAM Journal on Computing* 40.3, pp. 793–826 (cit. on p. 30).

Kazemi, E., Hassani, S. H. and Grossglauser, M. (2015). “Growing a graph matching from a handful of seeds”. *Proceedings of the VLDB Endowment* 8.10, pp. 1010–1021 (cit. on p. 35).

Knoblauch, J., Jewson, J. E. and Damoulas, T. (2018). “Doubly Robust Bayesian Inference for Non-Stationary Streaming Data with β-Divergences”. *Advances in Neural Information Processing Systems* (cit. on pp. 25, 73, 77).

Koh, P. W. and Liang, P. (2017). “Understanding Black-box Predictions via Influence Functions”. *International Conference on Machine Learning* (cit. on p. 72).

Kohavi, R. (1996). “Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid.” *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (cit. on p. 105).

Korattikara, A., Chen, Y. and Welling, M. (2014). “Austerity in MCMC land: Cutting the Metropolis-Hastings budget”. *International Conference on Machine Learning* (cit. on p. 21).

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. and Blei, D. M. (2017). “Automatic Differentiation Variational Inference”. *Journal of Machine Learning Research* 18.14 (cit. on pp. 59, 68).

Kullback, S. (1959). *Statistics and Information Theory* (cit. on p. 17).

Kullback, S. and Leibler, R. A. (1951). “On information and sufficiency”. *The Annals of Mathematical Statistics* 22.1, pp. 79–86 (cit. on p. 17).

Kurtek, S. and Bharath, K. (2015). “Bayesian sensitivity analysis with the Fisher–Rao metric”. *Biometrika* 102.3, pp. 601–616 (cit. on p. 26).

Laurila, J. K., Gatica-Perez, D., Aad, I., Bornet, O., Do, T.-M.-T., Dousse, O., Eberle, J., Miettinen, M., et al. (2012). “The mobile data challenge: Big data for mobile computing research”. *Pervasive Computing* (cit. on p. 32).

Lewis, D. D., Yang, Y., Rose, T. G. and Li, F. (2004). “RCV1: A New Benchmark Collection for Text Categorization Research”. *Journal of Machine Learning Research* 5, pp. 361–397 (cit. on p. 72).

Li, B., Wang, Y., Singh, A. and Vorobeychik, Y. (2016). “Data poisoning attacks on factorization-based collaborative filtering”. *Advances in Neural Information Processing Systems* (cit. on p. 72).

Li, N., Li, T. and Venkatasubramanian, S. (2007). “t-closeness: Privacy beyond k-anonymity and l-diversity”. *Proceedings of the 23rd IEEE International Conference on Data Engineering* (cit. on p. 37).

Li, N., Qardaji, W. and Su, D. (2012). “On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy”. *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 32–33 (cit. on p. 91).

Li, T. and Ke, Y. (2019). “Thinning for Accelerating the Learning of Point Processes”. *Advances in Neural Information Processing Systems* (cit. on p. 90).

Lin, M., Cao, H., Zheng, V. W., Chang, K. C.-C. and Krishnaswamy, S. (2015). “Mobile user verification/identification using statistical mobility profile”. *2015 International Conference on Big Data and Smart Computing* (cit. on p. 34).

Liu, Q., Peng, J. and Ihler, A. T. (2012). “Variational Inference for Crowdsourcing”. *Advances in Neural Information Processing Systems* (cit. on p. 72).

Lucic, M., Bachem, O. and Krause, A. (2016a). “Linear-Time Outlier Detection via Sensitivity”. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (cit. on pp. 13, 72).

Lucic, M., Bachem, O. and Krause, A. (2016b). “Strong coresets for hard and soft Bregman clustering with applications to exponential family mixtures”. *International Conference on Artificial Intelligence and Statistics* (cit. on p. 58).

Lucic, M., Faulkner, M., Krause, A. and Feldman, D. (2017). “Training Gaussian mixture models at scale via coresets”. *The Journal of Machine Learning Research* 18.1, pp. 5885–5909 (cit. on p. 23).

Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M. (2007). “l-diversity: Privacy Beyond k-anonymity”. *ACM Trans. Knowl. Discov. Data* (cit. on p. 37).

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press (cit. on p. 20).

Madigan, D., Raghavan, N. and DuMouchel, W. (2002). “Likelihood-based data squashing: a modeling approach to instance construction”. *Data Mining and Knowledge Discovery* 6 (cit. on p. 58).

Manousakas, D. and Mascolo, C. (2021). “β-Cores: Robust Large-Scale Bayesian Data Summarization in the Presence of Outliers”. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (cit. on p. 15).

Manousakas, D., Mascolo, C., Beresford, A. R., Chan, D. and Sharma, N. (2018). “Quantifying privacy loss of human mobility graph topology”. *Proceedings on Privacy Enhancing Technologies* 2018.3, pp. 5–21 (cit. on p. 15).

Manousakas, D., Xu, Z., Mascolo, C. and Campbell, T. (2020). “Bayesian Pseudocoresets”. *Advances in Neural Information Processing Systems* (cit. on pp. 15, 78).

McKay, B. D. and Piperno, A. (2014). “Practical graph isomorphism, II”. *Journal of Symbolic Computation* 60, pp. 94–112 (cit. on p. 47).

McMahan, H. B. and Streeter, M. J. (2010). “Adaptive Bound Optimization for Online Convex Optimization”. *The 23rd Conference on Learning Theory* (cit. on p. 82).

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). *Efficient estimation of word representations in vector space.* arXiv: 1301.3781 (cit. on p. 41).

Miller, J. W. and Dunson, D. B. (2019). “Robust Bayesian Inference via Coarsening”. *Journal of the American Statistical Association* 114.527 (cit. on pp. 25, 69, 73).

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D. and Alon, U. (2002). “Network Motifs: Simple Building Blocks of Complex Networks”. *Science* 298.5594, pp. 824–827 (cit. on p. 36).

Mirzasoleiman, B., Karbasi, A. and Krause, A. (2017). “Deletion-robust submodular maximization: Data summarization with “the right to be forgotten””. *International Conference on Machine Learning* (cit. on p. 91).

Mishinev, E. I. (2020). *Anonymized Data Linkability via Approximate Graph Matching.* Part II Project. Department of Computer Science and Technology, University of Cambridge (cit. on p. 35).

Morse, S., Gonzalez, M. C. and Markuzon, N. (2016). “Persistent cascades: Measuring fundamental communication structure in social networks”. *2016 IEEE International Conference on Big Data* (cit. on p. 54).

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective.* MIT press (cit. on p. 20).

Musco, C. and Musco, C. (2017). “Recursive sampling for the Nyström method”. *Advances in Neural Information Processing Systems* (cit. on p. 58).

Naini, F. M., Unnikrishnan, J., Thiran, P. and Vetterli, M. (2016). “Where You Are Is Who You Are: User Identification by Matching Statistics”. *IEEE Transactions on Information Forensics and Security* 11.2 (cit. on pp. 32, 34).

Narayanan, A. and Shmatikov, V. (2008). “Robust De-anonymization of Large Sparse Datasets”. *29th IEEE Symposium on Security and Privacy* (cit. on p. 33).

Narayanan, A. and Shmatikov, V. (2009). “De-anonymizing social networks”. *30th IEEE Symposium on Security and Privacy* (cit. on p. 35).

Neal, R. (2011). “MCMC using Hamiltonian dynamics”. *Handbook of Markov chain Monte Carlo.* Ed. by S. Brooks, A. Gelman, G. Jones and X.-L. Meng. CRC Press. Chap. 5 (cit. on pp. 21, 59).

Nemhauser, G. L., Wolsey, L. A. and Fisher, M. L. (1978). “An analysis of approximations for maximizing submodular set functions - I”. *Math. Program.* 14.1, pp. 265–294 (cit. on p. 104).

Olejnik, L., Castelluccia, C. and Janc, A. (2014). "On the uniqueness of Web browsing history patterns". *Annales des Télécommunications* 69 (cit. on p. 36).

Park, M., Foulds, J. R., Chaudhuri, K. and Welling, M. (2020). "Variational Bayes In Private Settings (VIPS)". *J. Artif. Intell. Res.* 68 (cit. on p. 59).

Paschou, P., Lewis, J., Javed, A. and Drineas, P. (2010). "Ancestry informative markers for fine-scale individual assignment to worldwide populations". *Journal of Medical Genetics* (cit. on p. 72).

Pedarsani, P., Figueiredo, D. R. and Grossglauser, M. (2013). "A Bayesian method for matching two similar graphs without seeds". *2013 51st Annual Allerton Conference on Communication, Control, and Computing.* IEEE, pp. 1598–1607 (cit. on p. 35).

Pfitzmann, A. and Hansen, M. (2010). *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management* (cit. on p. 37).

Pinsler, R., Gordon, J., Nalisnick, E. and Hernández-Lobato, J. M. (2019). "Bayesian Batch Active Learning as Sparse Subset Approximation". *Advances in Neural Information Processing Systems* (cit. on pp. 82, 84).

Pržulj, N. (2007). "Biological network comparison using graphlet degree distribution". *Bioinformatics* 23.2, e177–e183 (cit. on p. 36).

Pyrgelis, A., Troncoso, C. and De Cristofaro, E. (2017). "What Does The Crowd Say About You? Evaluating Aggregation-based Location Privacy". *Proceedings on Privacy Enhancing Technologies* 2017.4, pp. 156–176 (cit. on p. 34).

Rahimi, A. and Recht, B. (2008). "Random features for large-scale kernel machines". *Advances in neural information processing systems* (cit. on p. 28).

Ranganath, R., Gerrish, S. and Blei, D. (2014). "Black Box Variational Inference". *International Conference on Artificial Intelligence and Statistics* (cit. on p. 59).

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L. and Moy, L. (2010). "Learning from crowds." *Journal of Machine Learning Research* 11.4 (cit. on p. 72).

Ríos Insúa, D. and Ruggeri, F. (2012). *Robust Bayesian Analysis.* Vol. 152. Springer Science & Business Media (cit. on p. 73).

Riquelme, C., Tucker, G. and Snoek, J. (2018). "Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling". *6th International Conference on Learning Representations* (cit. on p. 82).

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods (Springer Texts in Statistics).* Berlin, Heidelberg: Springer-Verlag. ISBN: 0387212396 (cit. on p. 21).

Rossi, L., Williams, M. J., Stich, C. and Musolesi, M. (2015). "Privacy and the City: User Identification and Location Semantics in Location-Based Social Networks". *Proceedings of the Ninth International Conference on Web and Social Media* (cit. on p. 34).

Samek, W., Blythe, D., Müller, K.-R. and Kawanabe, M. (2013). "Robust spatial filtering with beta divergence". *Advances in Neural Information Processing Systems* (cit. on p. 103).

Schellekens, V., Chatalic, A., Houssiau, F., de Montjoye, Y.-A., Jacques, L. and Gribonval, R. (2019). "Differentially private compressive k-means". *IEEE International Conference on Acoustics, Speech and Signal Processing* (cit. on p. 12).

Schölkopf, B., Smola, A. J., Bach, F., et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press (cit. on p. 27).

Scholtes, I. (2017). "When is a Network a Network?: Multi-Order Graphical Model Selection in Pathways and Temporal Networks". *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining* (cit. on pp. 38, 45).

Schöning, U. (1988). "Graph isomorphism is in the low hierarchy". *Journal of Computer and System Sciences* 37.3, pp. 312–323 (cit. on p. 35).

Shapley, L. S. (1953). "A Value for n-Person Games". *Contributions to the Theory of Games* 2.28 (cit. on p. 84).

Sharad, K. and Danezis, G. (2014). "An Automated Social Graph De-anonymization Technique". *Proceedings of the 13th ACM Workshop on Privacy in the Electronic Society* (cit. on p. 35).

Sheng, V. S., Provost, F. and Ipeirotis, P. G. (2008). "Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers". *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining* (cit. on p. 72).

Shervashidze, N., Schweitzer, P., Leeuwen, V., Jan, E., Mehlhorn, K. and Borgwardt, K. (2011). "Weisfeiler-Lehman graph kernels". *Journal of Machine Learning Research* 12, pp. 2539–2561 (cit. on p. 40).

Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K. and Borgwardt, K. (2009). "Efficient graphlet kernels for large graph comparison". *Artificial Intelligence and Statistics*, pp. 488–495 (cit. on p. 36).

Shokri, R., Troncoso, C., Diaz, C., Freudiger, J. and Hubaux, J.-P. (2010). "Unraveling an old cloak: k-anonymity for location privacy". *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society* (cit. on p. 37).

Snelson, E. and Ghahramani, Z. (2005). "Sparse Gaussian processes using pseudo-inputs". *Advances in Neural Information Processing Systems* (cit. on pp. 58, 61).

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M. and Adams, R. (2015). "Scalable Bayesian Optimization Using Deep Neural Networks". *Proceedings of the 32nd International Conference on Machine Learning* (cit. on p. 82).

Song, Y., Stolfo, S. and Jebara, T. (2011). *Markov models for network-behavior modeling and anonymization.* Tech. rep. Department of Computer Science, Columbia University (cit. on p. 55).

Steinhardt, J., Koh, P. W. W. and Liang, P. S. (2017). “Certified defenses for data poisoning attacks”. *Advances in Neural Information Processing Systems* (cit. on p. 72).

Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J. and Clore, J. N. (2014). “Impact of HbA1c measurement on hospital readmission rates: analysis of $70,000$ clinical database patient records”. *BioMed research international* (cit. on p. 105).

Sweeney, L. (2002). “k-anonymity: A model for protecting privacy”. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05, pp. 557–570 (cit. on pp. 34, 37).

Thoma, M., Cheng, H., Gretton, A., Han, J., Kriegel, H. P., Smola, A., Song, L., Yu, P. S., Yan, X. and Borgwardt, K. M. (2010). “Discriminative frequent subgraph mining with optimality guarantees”. *Statistical Analysis and Data Mining* 3.5, pp. 302–318 (cit. on p. 55).

Titsias, M. (2009). “Variational Learning of Inducing Variables in Sparse Gaussian Processes”. *International Conference on Artificial Intelligence and Statistics* (cit. on pp. 58, 61).

Tomczak, J. M. and Welling, M. (2018). “VAE with a VampPrior”. *International Conference on Artificial Intelligence and Statistics* (cit. on p. 61).

Tukey, J. W. (1960). “A survey of sampling from contaminated distributions”. *Contributions to probability and statistics*, pp. 448–485 (cit. on p. 13).

Vahidian, S., Mirzasoleiman, B. and Cloninger, A. (2020). “Coresets for Estimating Means and Mean Square Error with Limited Greedy Samples”. *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence* (cit. on p. 84).

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science.* Vol. 47. Cambridge University Press (cit. on p. 13).

Vishwanathan, S., Schraudolph, N., Kondor, R. and Borgwardt, K. (2010). “Graph Kernels”. *Journal of Machine Learning Research* 11, pp. 1201–1242 (cit. on pp. 32, 36, 39).

Wagner, D. T., Rice, A. and Beresford, A. R. (2014). “Device Analyzer: Understanding Smartphone Usage”. *Mobile and Ubiquitous Systems: Computing, Networking, and Services: 10th International Conference*, pp. 195–208 (cit. on pp. 32, 44).

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint.* Vol. 48. Cambridge University Press (cit. on p. 13).

Wainwright, M. J. and Jordan, M. I. (Jan. 2008). “Graphical Models, Exponential Families, and Variational Inference”. *Found. Trends Mach. Learn.* 1.1-2, pp. 1–305 (cit. on pp. 18, 74).

Wang, C. and Blei, D. M. (2018). “A general method for robust Bayesian modeling”. *Bayesian Analysis* (cit. on p. 73).

Wang, D., Irani, D. and Pu, C. (2012). “Evolutionary study of web spam: Webb Spam Corpus 2011 versus Webb Spam Corpus 2006”. *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing* (cit. on p. 105).

Wang, D., Liu, H. and Liu, Q. (2018). “Variational Inference with Tail-adaptive f-Divergence”. *Advances in Neural Information Processing Systems* (cit. on p. 77).

Wang, Y.-X., Balle, B. and Kasiviswanathan, S. P. (2019). “Subsampled Rényi Differential Privacy and Analytical Moments Accountant”. *International Conference on Artificial Intelligence and Statistics* (cit. on p. 65).

Wang, Y., Kucukelbir, A. and Blei, D. M. (2017). “Robust Probabilistic Modeling with Bayesian Data Reweighting”. *Proceedings of the 34th International Conference on Machine Learning* (cit. on pp. 69, 73).

Wei, X. and Minsker, S. (2017). “Estimation of the covariance structure of heavy-tailed distributions”. *Advances in Neural Information Processing Systems* (cit. on p. 78).

Weisfeiler, B. and Lehman, A. (1968). “A reduction of a graph to a canonical form and an algebra arising during this reduction”. *Nauchno-Technicheskaya Informatsia* 2.9, pp. 12–16 (cit. on p. 40).

Welke, P., Andone, I., Blaszkiewicz, K. and Markowetz, A. (2016). “Differentiating Smartphone Users by App Usage”. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (cit. on p. 36).

Welling, M. (2009). “Herding Dynamical Weights to Learn”. *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal, Quebec, Canada (cit. on p. 61).

Welling, M. and Teh, Y. W. (2011). “Bayesian learning via stochastic gradient Langevin dynamics”. *Proceedings of the 28th International Conference on Machine Learning* (cit. on pp. 21, 72).

Whitehill, J., Wu, T.-F., Bergsma, J., Movellan, J. R. and Ruvolo, P. L. (2009). “Whose vote should count more: Optimal integration of labels from labelers of unknown expertise”. *Advances in Neural Information Processing Systems* (cit. on p. 72).

Williams, C. and Seeger, M. (2001). “Using the Nyström method to speed up kernel machines”. *Advances in Neural Information Processing Systems* (cit. on p. 58).

Williams, P. M. (1980). “Bayesian conditionalisation and the principle of minimum information”. *The British Journal for the Philosophy of Science* 31.2, pp. 131–144 (cit. on p. 24).

Xu, F., Tu, Z., Li, Y., Zhang, P., Fu, X. and Jin, D. (2017). “Trajectory Recovery From Ash: User Privacy Is NOT Preserved in Aggregated Mobility Data”. *Proceedings of the 26th International Conference on World Wide Web* (cit. on p. 34).

Xu, J., Wickramarathne, T. L. and Chawla, N. V. (2016). “Representing higher-order dependencies in networks”. *Science Advances* 2.5 (cit. on p. 38).

Yan, X. and Han, J. (2002). “gSpan: Graph-Based Substructure Pattern Mining”. *Proceedings of the 2002 IEEE International Conference on Data Mining* (cit. on pp. 36, 54).

Yanardag, P. and Vishwanathan, S. V. N. (2015). “Deep Graph Kernels”. *Proceedings of the 21th ACM International Conference on Knowledge Discovery and Data Mining* (cit. on p. 41).

Yen, T.-F., Xie, Y., Yu, F., Yu, R. P. and Abadi, M. (2012). "Host Fingerprinting and Tracking on the Web:Privacy and Security Implications". *The 19th Annual Network and Distributed System Security Symposium*. Internet Society (cit. on p. 36).

Zang, H. and Bolot, J. (2011). "Anonymization of Location Data Does Not Work: A Large-scale Measurement Study". *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*. ACM (cit. on pp. 32, 33).

Zellner, A. (1988). "Optimal Information Processing and Bayes's Theorem". *The American Statistician* 42.4, pp. 278–280 (cit. on p. 24).

Zhang, J. Y., Khanna, R., Kyrillidis, A. and Koyejo, O. (2021a). "Bayesian Coresets: Revisiting the Nonconvex Optimization Perspective". *International Conference on Artificial Intelligence and Statistics* (cit. on p. 23).

Zhang, R., Li, Y., De Sa, C., Devlin, S. and Zhang, C. (2021b). "Meta-Learning Divergences for Variational Inference". *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics* (cit. on p. 77).

Zhang, Y., Chen, X., Zhou, D. and Jordan, M. I. (2016). "Spectral methods meet EM: A provably optimal algorithm for crowdsourcing". *The Journal of Machine Learning Research* 17.1, pp. 3537–3580 (cit. on p. 72).

Zheleva, E. and Getoor, L. (2009). "To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles". *Proceedings of the 18th International Conference on World Wide Web* (cit. on p. 35).

Zheng, V. W., Pan, S. J., Yang, Q. and Pan, J. J. (2008). "Transferring Multi-device Localization Models using Latent Multi-task Learning." *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence* (cit. on p. 72).

Zhou, S., Lafferty, J. and Wasserman, L. (2007). "Compressed regression". *Advances in Neural Information Processing Systems* (cit. on pp. 12, 58, 61).

Zhu, J., Chen, N. and Xing, E. P. (2014). "Bayesian Inference with Posterior Regularization and Applications to Infinite Latent SVMs". *Journal of Machine Learning Research* 15.53, pp. 1799–1847 (cit. on p. 24).

Zhuang, H., Parameswaran, A., Roth, D. and Han, J. (2015). "Debiasing Crowdsourced Batches". *Proceedings of the 21th ACM International Conference on Knowledge Discovery and Data Mining* (cit. on p. 72).