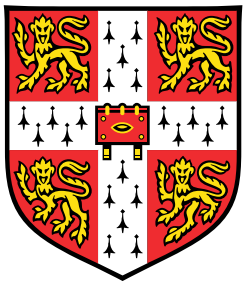
Differential Privacy & Approximate Bayesian Inference



Mrinank Sharma

Supervisor: Dr Richard E. Turner

Department of Engineering
University of Cambridge

I hereby declare that, except where specifically indicated, the work submitted herin is my own original work.

This report is submitted for the degree of
Master of Engineering

Pembroke College May 2019

Acknowledgements

I would like to acknowledge and thank my supervisor, Rich, who has aided me greatly and continually encourage me throughout this project. Additionally, I would like to thank Siddharth who has been involved in this project from the beginning and has provided valuable feedback, helped me work through technical difficulties and provided insight. I would also like to thank Thang for providing the software which much of this project is built upon.

Technical Abstract

Machine learning models have recently found use in contexts in which large quantities of private, sensitive data are used to make decisions and predictions, namely healthcare and public policy. It is essential to ensure that an adversary is unable to infer private sensitive information from model predictions. Differential Privacy, originally developed in cryptography literature, has become a mathematical standard for quantifying the level of privacy offered by parties processing data by introducing carefully calibrated noise when data queries are made. A particular area of interest is the federated learning context in which a number of clients, each of which holds a local dataset, interact with a central parameter server in order to train a shared model. The aim of this project is to develop techniques to allow for differentially private federated Bayesian learning.

Partitioned Variational Inference (PVI) is a recently developed framework which enables variational inference to be applied in the federated learning context. In this report, we present two adaptations to the PVI algorithm for parametric families which incorporate differential privacy techniques in order to protect the privacy of individuals. The first adaptation, data-point level DP-PVI, limits and obscures the contribution of each individual data-point held by every client and is the natural technique to apply when the data from each client could relate to a number of individuals and does not require encryption and authentication schemes. The second adaptation, dataset level DP-PVI, limits and obscures the contribution that a client's entire dataset has on the global model parameters and is natural when the dataset from each client relates to one individual only but requires message encryption. Both of these techniques rely on clipping, which limits how much individual data-points (or data-sets) are able to influence outcomes. The algorithms developed are fully general and could be applied to a large range of probabilistic models.

The case study of Bayesian linear regression, using a Gaussian prior and likelihood, is developed to investigate the performance and understand the properties of these algorithms. We find that across both techniques, extreme care must be taken in choosing algorithm hyper-parameters, specifically the clipping bound which can introduce significant bias if chosen inappropriately. We find the dataset level DP-PVI algorithm can give very close to non-private performance and is the more promising of the approaches applied. Furthermore, we note that the choice of parametrisation is very important and has significant implications for the outcomes of this algorithm.

Table of contents

Chapter 1

Introduction

Machine learning methods are trained on datasets, leveraging information within the dataset in order to make predictions about previously unseen data and make decisions. Recently, such methods have found use in scenarios where the data used is personal and sensitive, one example being the use of human genomic data to predict drug sensitivity (**?**). Typically, data relating to individuals in training datasets are *anonymised*, for example, by removing all identifiable information (such as names, addresses, etc) and replacing this information with an anonymous identifier. However, **?** show that anonymisation is insufficient, partially due to the availability of *auxiliary information* i.e. additional, publicly available information. When Netflix released a dataset in 2006 containing movie ratings for approximately 500000 subscribers with names replaced with identifiers, the data could be combined with public ratings on Internet Movie Database (IMDb) to identify movie ratings of two users. Intuitively, data-points about individuals are highly dimensional meaning that anonymisation is insufficient, a further example being that even when sharing DNA sequence data without identifiers, it is possible to recover particular surnames using additional meta-data (**?**). Machine learning approaches have also been used in public policy, for instance geographically placing refugees to optimise their overall employment rate (**?**).

? have shown that *model inversion attacks* are possible, where an adversary seeks to learn information about training data given model predictions. In particular, a neural network for facial recognition which returned confidence values was exploited in order to recover the image of a training set participant. Whilst more sophisticated attacks will be required for algorithms which do not provide confidence information, it is therefore possible for an adversary to recover anonymised training data-points and then de-anonymise this information using auxiliary information.

Federated learning, which performs global model training using a large quantity of decentralised data, is a particularly interesting context, especially with the increasing availability and affordability of mobile smart-phones. These approaches are also applicable for Internet of Things (IoT) devices. Additionally, federated learning schemes reduce power consumption

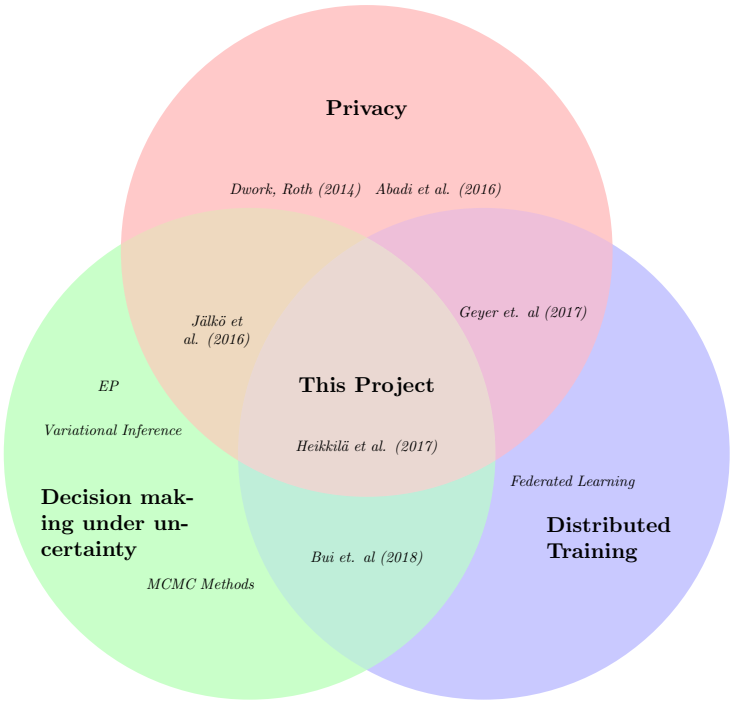


Fig. 1.1 Project aims, including other work within this area.

and intuitively give stronger privacy by removing the requirement of transferring entire local datasets (**?**).

It is often desirable to performance *Bayesian Inference* when possible. Bayesian approaches incorporate prior information about unknown parameters in order to mathematically quantify the uncertainty in parameter estimates and predictions. This allows the application of decision theory, thus providing a framework to make optimal decisions under uncertainty (**?**).

This project lies in the intersection of privacy, Bayesian inference and federated learning. There has been limited previous work in the intersection of these yields, and current techniques are only applicable for exponential conjugate models (**?**). The aim of this project is to develop a generalised method to enable Bayesian inference to be performed using complex models in contexts where data is distributed over a number of clients whilst also providing privacy guarantees for each client.

Chapter 2

Literature Review

2.1 Differential Privacy

2.1.1 Preliminaries

Differential privacy is one mathematical technique, widely adopted by the community, which formalises privacy and is able to numerically quantify the level of privacy that some method provides. This is particularly useful not only from the point of view of a designer who is able to quantify the trade-off between privacy and utility but also from the point of view of a client whose data we seek to protect; a client would be able to compare the protection offered by rival companies, or parameter settings.

Definition 2.1.1 (ϵ-Differential Privacy) *A randomised algorithm, $\mathcal{A}$, is said to be ϵ-differentially private if for any possible subset of outputs, S, and for all pairs of datasets, $(\mathcal{D}, \mathcal{D}')$, which differ in one entry only, the following inequality holds:*

$$Pr(\mathcal{A}(\mathcal{D}) \in S) \leq e^{\epsilon} Pr(\mathcal{A}(\mathcal{D}') \in S) \tag{2.1}$$

Noting that since this definition is symmetric across datasets, this requirement effectively enforces that:

$$e^{-\epsilon} \leq \frac{\Pr(\mathcal{A}(\mathcal{D}) \in S)}{\Pr(\mathcal{A}(\mathcal{D}') \in S)} \leq e^{\epsilon} \tag{2.2}$$

Thus, it can be seen that this enforces privacy in the sense that the output probability densities ought to be similar (with the ratio of density values close to one) for datasets which are also similar (i.e. differ in only entry only) (**?**). ϵ, a positive quantity, quantifies the level of privacy provided, with smaller values of epsilon corresponding to stronger privacy guarantees.

It is difficult to choose and interpret the value of ϵ. **?** provide a principled technique to choose ϵ for surveys based on economic analysis based on estimating the monetary cost of

leaking private data. This is less appropriate for machine learning methods, but we note that common values of ϵ found in the literature (as found in **?**) do not exceed $\epsilon = 10$.

In practice, differential privacy is achieved by bounding the contribution that any single data-point may have on the outcome and then adding noise which scales with the maximum magnitude of this contribution. The adversary would then be unable to determine whether a particular output was simply due to noise or due to the contribution of a specific data-point. This is related to the concept of plausible denialability; a client could simply claim that a certain outcome only occurred due to the noise applied rather than their specific contribution.

Differential privacy is suitable to quantify privacy as it is immune to post-processing; an adversary is only able to make the output of a differentially private algorithm less private (i.e. increase the value of ϵ) with additional knowledge of the dataset (**?**). Additional knowledge of the dataset could refer to either specific information about data-points in the dataset or aggregate information about a collection of data-points in the dataset.

Often, the above definition of differential privacy is slackened by introducing an extra privacy variable.

Definition 2.1.2 ((ϵ, δ)-Differential Privacy) *A randomised algorithm, $\mathcal{A}$, is said to be (ϵ, δ) differentially private if for any possible subset of outputs, S, and for all datasets, $(\mathcal{D}, \mathcal{D}')$, which differ in one entry only, the following inequality holds:*

$$Pr(\mathcal{A}(\mathcal{D}) \in S) \leq e^{\epsilon} Pr(\mathcal{A}(\mathcal{D}') \in S) + \delta \tag{2.3}$$

Similar to ϵ, larger values of δ correspond to weaker privacy guarantees and $\delta = 0$ corresponds to a pure ϵ-differentially private algorithm. Note that it can be shown that (ϵ, δ)-DP is equivalent to a probabilistic pure ϵ-DP guarantee with probability $1 - \delta$ (**?**).

A useful quantity regarding functions which operate on data is the ℓ_2 sensitivity.

Definition 2.1.3 (ℓ_2 Sensitivity) *The ℓ_2 sensitivity of function $f : \mathcal{D} \to \mathbb{R}^n$, is denoted as $\Delta_2(f)$ and is defined as:*

$$\Delta_2(f) = \max_{\mathcal{D}, \mathcal{D}'} ||f(\mathcal{D}) - f(\mathcal{D}')||_2 \tag{2.4}$$

where $\mathcal{D}$ and $\mathcal{D}'$ differ in one entry only.

For a function with a given ℓ_2 sensitivity, the Gaussian mechanism can be used to provide an (ϵ, δ)-differential privacy guarantee (**?**).

Theorem 2.1.4 (Gaussian Mechanism) *Let $f : \mathcal{D} \to \mathbb{R}^n$ be a function with ℓ_2 sensitivity $\Delta_2(f)$. Releasing $f(\mathcal{D}) + \eta$ is (ϵ, δ)-differentially private where $\eta \sim \mathcal{N}(0, \sigma^2)$ when:*

$$\sigma^2 > 2\ln(1.25/\delta)\Delta_2^2(f)/\epsilon^2 \tag{2.5}$$

with $\epsilon \in (0, 1)$.

Typically, a particular process may be repeatedly applied to a dataset; a simple example being that in many machine learning problems, a loss function can be composed into a sum over data-points. Gradient descent techniques repeatedly compute the gradient of the loss over subsets of data-points and use this estimate to adjust model parameters to reduce the loss (**?**). A key reason that differential privacy is an appropriate way of measuring privacy is that it can be *composed*; when a randomised algorithm with some privacy guarantee is repeatedly applied to a dataset, it is possible to convert the individual privacy guarantees into an overall guarantee (note that the privacy guarantee refers to a (ϵ, δ) value). There exist many schemes which bound the values of ϵ and δ in these circumstances. The *Moments Accountant* is an advanced technique which is able to 'account' for and track the total privacy expenditure of a technique by tracking the moments of the privacy loss and provides tight upper bounds on the values of ϵ and δ.

2.1.2 The Moments Accountant

The Moments Accountant was proposed in (**?**). Here, we provide a summary of its computation following the original paper.

Definition 2.1.5 (Privacy Loss) *For neighbouring datasets, $\mathcal{D}$ and $\mathcal{D}'$, a stochastic algorithm, $\mathcal{A}$ and some outcome, S, define the privacy loss at S as:*

$$c(S; \mathcal{A}, \mathcal{D}, \mathcal{D}') \triangleq \log \frac{\Pr[\mathcal{A}(\mathcal{D}) = S]}{\Pr[\mathcal{A}(\mathcal{D}') = S]} \tag{2.6}$$

If the probability of an observed algorithm outcome is very different across neighbouring datasets, observing this outcome reveals information about the dataset. This intuition matches the above definition, as this situation corresponds to a large (absolute) value of c. Note that since the algorithms we use are stochastic, the privacy loss is itself a random variable. It is clear that an $(\epsilon, 0)$ privacy guarantee directly corresponds to claiming that the privacy loss random variable will never exceed ϵ in absolute value.

Definition 2.1.6 (λth Moment of $\mathcal{A}$) *For algorithm $\mathcal{A}$, a quantity of interest is the maximum value of the log of the moment generating function of the privacy loss:*

$$\alpha_{\mathcal{A}}(\lambda) \triangleq \max_{\mathcal{D}, \mathcal{D}'} \ln \mathbb{E}_{S \sim \mathcal{A}(\mathcal{D})} \Big\{ \exp \lambda c(S; \mathcal{A}, \mathcal{D}, \mathcal{D}') \Big\} \tag{2.7}$$

Theorem 2.1.7 (Properties of $\alpha_{\mathcal{A}}(\lambda)$) *$\alpha_{\mathcal{A}}(\lambda)$ exhibits two important properties:*

1. ***Composability:*** *If the algorithm $\mathcal{A}$ consists of a sequence of adaptive steps, $\mathcal{M}_1, \ldots, \mathcal{M}_k$, for any λ:*

$$\alpha_{\mathcal{A}}(\lambda) \leq \sum_{i=1}^{k} \alpha_{\mathcal{M}_i}(\lambda) \tag{2.8}$$

2. ***Tail Bound:*** *For any $\epsilon > 0$ and λ, the stochastic algorithm, $\mathcal{A}$, is (ϵ, δ)-differentially private for*

$$\delta \leq \exp(\alpha_{\mathcal{A}}(\lambda) - \lambda\epsilon) \tag{2.9}$$

The Moments Accountant scheme computes $\alpha_{\mathcal{M}_i}(\lambda)$ for each step for several values of λ and uses the composability property to compute the aggregate loss. In typical applications, either a target value for ϵ or δ is fixed at some value, and the tail bound property is applied to compute the best possible value of the other privacy variable, using one of the following equations:

$$\delta = \min_{\lambda} \exp(\alpha_{\mathcal{A}}(\lambda) - \lambda\epsilon) \tag{2.10}$$

$$\epsilon = \min_{\lambda} \frac{1}{\lambda}(\alpha_{\mathcal{A}}(\lambda) - \ln\delta) \tag{2.11}$$

The minimum is taken as the tail bound property gives upper bounds on the privacy variables, but smaller values of ϵ and δ correspond to tighter bounds on the privacy loss and thus stronger privacy guarantees.

Let $\mathcal{D} = \{\boldsymbol{x}_i\}$ and $\mathcal{D}' = \mathcal{D} \cup \boldsymbol{x}'$ where $\boldsymbol{x} \in \mathcal{X}$. Let $f : \mathcal{X} \to \mathbb{R}^n$ with $||f(\cdot)||_2 \leq \Delta$. Consider the following mechanism:

$$\mathcal{M}(\mathcal{D}) = \sum_{i \in J} f(\boldsymbol{x}_i) + \sigma\Delta\boldsymbol{\eta} \tag{2.12}$$
$$\eta_i \sim \mathcal{N}(0, 1)$$

where J is a subset of indices where each index is chosen independently with probability q. Without loss of generality, let $f(\boldsymbol{x}_i) = \boldsymbol{0}$ and $f(\boldsymbol{x}') = \Delta \cdot \boldsymbol{e}_1$ where $\boldsymbol{e}_1$ is a unit vector. Then $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$ are distributed identically other than the coordinate corresponding to $\boldsymbol{e}_1$. Considering only this direction, the output densities for the mechanism are:

$$p(\mathcal{M}(\mathcal{D})_1 = z) \sim \mu_0(z) \triangleq \mathcal{N}(z|0, \Delta^2\sigma^2) \tag{2.13}$$

$$p(\mathcal{M}(\mathcal{D})'_1 = z) \sim \mu_1(z) \triangleq q \cdot \mathcal{N}(z|\Delta, \Delta^2\sigma^2) + (1-q) \cdot \mathcal{N}(z|0, \Delta^2\sigma^2) \tag{2.14}$$

since the probability of $\boldsymbol{x}'$ being selected is q. Then, by definition, $\alpha_{\mathcal{M}}(\lambda)$ can be computed as:

$$\alpha_{\mathcal{M}}(\lambda) = \ln\max(E_1, E_2) \tag{2.15}$$

$$E_1 = \mathbb{E}_{z\sim\mu_0}\left[\left(\frac{\mu_0(z)}{\mu_1(z)}\right)^{\lambda}\right] \tag{2.16}$$

$$E_2 = \mathbb{E}_{z\sim\mu_1}\left[\left(\frac{\mu_1(z)}{\mu_0(z)}\right)^{\lambda}\right] \tag{2.17}$$

Each integral can be evaluated using numerical integration. Both integrals must be considered since the maximum of the log moments of the privacy loss must be found, and both $\mathcal{D}$ and $\mathcal{D}'$ could be considered as the 'original' dataset.

Note: the computed values of ϵ and δ are independent of the data used for the mechanism and can be pre-computed given fixed values of Δ, q and either δ or ϵ.

2.1.3 Differentially Private Stochastic Gradient Descent

? propose differentially private stochastic gradient descent by adapting stochastic gradient descent to use the Gaussian mechanism with additional sub-sampling to compute the gradient of some loss function.

Algorithm **??** outlines the differentially private stochastic gradient descent method. Note that the gradient clipping used in this method effectively limits the contribution of each data-points towards the gradient, bounding the ℓ_2 sensitivity and enabling the Gaussian mechanism with sub-sampling to be applied. Noting that the noise added scales with the clipping bound, large values of the clipping bound correspond to strong, but noisy gradient signals. A suggested setting for the clipping bound, c_t, is the median of the norms of unclipped gradients throughout the course of training. Furthermore, **?** suggest using a learning rate which starts off at a high value large and is reduced over time. Additionally, it is remarked that $L \simeq \sqrt{N}$ is an appropriate setting for the lot size; larger values of L improve the signal-to-noise ratio of the gradient estimate but increase the privacy cost.

Algorithm 1 Differentially Private Stochastic Gradient Descent (DP-SGD)

Input: Dataset $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, Loss function $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_i \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}_i)$
Parameters: Learning Rate α_t, Clipping Bound c_t, Lot Size L, DP Noise Scale σ_t, Num. Iterations T

Initialise $\boldsymbol{\theta}_0$ randomly.
for $t = 1, \ldots, T$ **do**
  Take a random sample L_t with sampling probability $q = L/N$
  for all $i \in L_t$ **do** ▷ Compute Gradient
    $\boldsymbol{g}_t(\boldsymbol{x}_i) \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}_i)$
    $\tilde{\boldsymbol{g}}_t(\boldsymbol{x}_i) \leftarrow \boldsymbol{g}_t(\boldsymbol{x}_i) / \max(||\boldsymbol{g}_t(\boldsymbol{x}_i)||_2 / c_t, 1)$ ▷ Gradient Clipping
  end for
  $\tilde{\boldsymbol{g}}_t \leftarrow \frac{1}{L}[\sum_{i \in L_t} \boldsymbol{g}_t(\boldsymbol{x}_i) + \sigma_t c_t \boldsymbol{z}]$ where $z_i \sim \mathcal{N}(0, 1)$ ▷ Perturb Clipped Gradient
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \alpha_t \tilde{\boldsymbol{g}}_t$
end for
Output $\boldsymbol{\theta}_T$ and calculate (ϵ, δ) using the Moments Accountant.

This technique has been applied to deep neural networks (**?**). However, this algorithm can be used as a building block to create other differentially private techniques, and indeed this technique has been used to perform variational inference for non-conjugate models (**?**).

2.2 Federated Learning

2.2.1 Partitioned Variational Inference

Partitioned Variational Inference (PVI) is a general framework which encompasses many variational Bayesian techniques. Assume that the dataset, $\mathcal{D}$, has been partitioned into M shards i.e. $\mathcal{D} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M\}$, where $\boldsymbol{X}_i = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_i}\}$. A probabilistic model with parameters $\boldsymbol{\theta}$ has been suggested to model this data with a known prior, $p(\boldsymbol{\theta})$, and likelihood function, $p(\boldsymbol{x}|\boldsymbol{\theta})$. The aim of Bayesian inference is to calculate the posterior density over the parameters, $p(\boldsymbol{\theta}|\mathcal{D})$, but in general, it is not possible to compute this distribution. In variational Bayesian methods, a variational distribution, $q(\boldsymbol{\theta})$, is used to approximate the posterior. In this report, we refer to $q(\boldsymbol{\theta})$ as either the variational distribution or the approximate posterior. In PVI, this distribution takes the form:

$$q(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{m=1}^{M} t_m(\boldsymbol{\theta}) \simeq \frac{1}{\mathcal{Z}} p(\boldsymbol{\theta}) \prod_{m=1}^{M} p(\boldsymbol{X}_m|\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathcal{D}) \tag{2.18}$$

Note that the variational distribution includes the product of terms corresponding to each client ($t_m(\boldsymbol{\theta})$), each of which approximates the (un-normalised) likelihood $p(\boldsymbol{X}_m|\boldsymbol{\theta})$. We refer to $t_m(\boldsymbol{\theta})$ as the approximate likelihood for client m. Additionally, note that the variational distribution does not include a normalising constant.

Algorithm 2 Partitioned Variational Inference (PVI)

Input: Partitioned Dataset $\mathcal{D} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M\}$, Prior $p(\boldsymbol{\theta})$, Family of Distributions $\mathcal{Q}$

1: Initialise approximate likelihood:

$$t_m^{(0)}(\boldsymbol{\theta}) \leftarrow 1 \ \forall m \tag{2.19}$$

2: Initialise approximate posterior:

$$q^{(0)}(\boldsymbol{\theta}) \leftarrow p(\boldsymbol{\theta}) \tag{2.20}$$

3: **for** $i = 1, 2, \ldots,$ until convergence **do**

4: $b_i \leftarrow$ index of next approximate likelihood to update

5: Compute the new approximate likelihood:

$$q^{(i)}(\boldsymbol{\theta}) \leftarrow \underset{q(\boldsymbol{\theta}) \in \mathcal{Q}}{\operatorname{argmax}} \int q(\boldsymbol{\theta}) \ln \frac{q^{(i-1)}(\boldsymbol{\theta}) p(\boldsymbol{X}_{b_i}|\boldsymbol{\theta})}{q(\boldsymbol{\theta}) t_{b_i}^{(i-1)}(\boldsymbol{\theta})} \, d\boldsymbol{\theta} \tag{2.21}$$

6: Update the approximate likelihood for the updated factor:

$$t_{b_i}^{(i)}(\boldsymbol{\theta}) \leftarrow \frac{q^{(i)}(\boldsymbol{\theta})}{q^{(i-1)}(\boldsymbol{\theta})} t_{b_i}^{(i-1)}(\boldsymbol{\theta}) \tag{2.22}$$

7: **end for**

When partitioned variational inference is used in the federated learning context, each client stores its own data partition, $\boldsymbol{X}_m$, and communicates with a central parameter server which stores the global approximate posterior, $q(\boldsymbol{\theta})$. At each iteration, each client maximises a *local* free energy (**?**).

Definition 2.2.1 (Local Free Energy) *The local free energy is defined as:*

$$\mathcal{F}_{b_i}^{(i)}(q(\boldsymbol{\theta})) = \int q(\boldsymbol{\theta}) \ln \frac{q^{(i-1)}(\boldsymbol{\theta}) p(\boldsymbol{X}_{b_i}|\boldsymbol{\theta})}{q(\boldsymbol{\theta}) t_{b_i}^{(i-1)}(\boldsymbol{\theta})} \, d\boldsymbol{\theta} \tag{2.23}$$

It can be shown that the local free energy optimisation (Eq. **??**) is equivalent to the following $\mathcal{KL}$ optimisation:

$$q^{(i)}(\boldsymbol{\theta}) \leftarrow \underset{q(\boldsymbol{\theta}) \in \mathcal{Q}}{\operatorname{argmax}} \mathcal{KL}(q(\boldsymbol{\theta}) || \hat{p}_{b_i}^{(i)}(\boldsymbol{\theta})) \tag{2.24}$$

where $\hat{p}_{b_i}^{(i)}(\boldsymbol{\theta})$ is known as the *tilted distribution* and is defined as:

$$\hat{p}_{b_i}^{(i)}(\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}_{b_i}^{(i)}} \frac{q^{(i-1)}(\boldsymbol{\theta})}{t_{b_i}^{(i-1)}(\boldsymbol{\theta})} p(\boldsymbol{X}_{b_i}|\boldsymbol{\theta}) \tag{2.25}$$

$$= \frac{1}{\mathcal{Z}_{b_i}^{(i)}} p(\boldsymbol{\theta}) p(\boldsymbol{X}_{b_i}|\boldsymbol{\theta}) \prod_{m \neq b_i} t_m^{(i-1)}(\boldsymbol{\theta}) \tag{2.26}$$

which can be interpreted as a local estimate of the posterior for client b_i, using the exact local likelihood and approximate likelihood terms for the data partitions held at other clients. The free energy could be maximised in a number of ways, including analytical updates or gradient based methods (such as applying Algorithm **??** on the negative local free energy).

In standard variational inference, a global free energy that depends on the entire dataset is maximised.

Definition 2.2.2 (Global Free Energy) *The global free energy is defined as:*

$$\mathcal{F}(q(\boldsymbol{\theta})) = \int q(\boldsymbol{\theta}) \ln \frac{p(\boldsymbol{\theta}) \prod_{m=1}^{M} p(\boldsymbol{X}_m|\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \, d\boldsymbol{\theta} \tag{2.27}$$

Maximising this free energy is equivalent to minimising the $\mathcal{KL}$ divergence between the approximate posterior and the true posterior, $\mathcal{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathcal{D}))$ (**?**).

Now we return to the partitioned variational inference setting.

Theorem 2.2.3 (Properties of PVI) *Consider the approximate posterior at convergence, $q^*(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{m=1}^{M} t_m^*(\boldsymbol{\theta})$. Then:*

1. *The sum of local free energies is the global free energy:*

$$\sum_{m=1}^{M} \mathcal{F}_m(q^*(\boldsymbol{\theta})) = \mathcal{F}(q^*(\boldsymbol{\theta})) \tag{2.28}$$

2. *Optimising the local free energies optimises the global free energy:*

$$q^*(\theta) = \underset{q(\boldsymbol{\theta})\in\mathcal{Q}}{\operatorname{argmax}} \mathcal{F}_m(q(\boldsymbol{\theta})) \ \forall m \Rightarrow q^*(\theta) = \underset{q(\boldsymbol{\theta})\in\mathcal{Q}}{\operatorname{argmax}} \mathcal{F}(q(\boldsymbol{\theta})) \tag{2.29}$$

The above properties of PVI suggest that the algorithm should yield a good approximate posterior distribution (**?**).

2.3 Additional Work

? develop differentially private Bayesian learning on distributed data for exponential conjugate family models. The posterior distributions obtained for these models depend on aggregated *sufficient statistics* i.e. these statistics capture all available information about the parameters of the model in question. The approach taken in this paper is to calculate aggregate noisy sufficient statistics across a number of computation nodes, effectively applying the Gaussian mechanism with a distributed noise scheme. This can not be applied to other models.

? develop a technique to perform federated learning which does not protect single data-points held by each client but rather the entire dataset held by each client. This is performed by choosing a sub-set of clients to use to refine the model. Model updates are then calculated from each client, rescaled if required so the ℓ_2 norm of each of the clients update is less than some value, S. This fixes the ℓ_2-sensitivity, allowing the Gaussian mechanism (with noise scaling with S) to be applied centrally.

Chapter 3

Differentially Private Partitioned Variational Inference

In this chapter, we construct different forms of *Differentially Private Partitioned Variational Inference (DP-PVI)*, considering the full problem context and implications with regards to strength of privacy protection. This is a key contribution of this work.

3.1 Context

The *adversary* aims to use accessible information to glean private and sensitive information which the machine learning model has been trained with. Recalling the PVI setting, there are a number of clients with access to their own local data shards containing sensitive information. In the fully general context, it is assumed that:

- The adversary has unlimited access to the trained variational distribution, $q(\boldsymbol{\theta})$ (as well as the approximate posterior throughout the course of training).
- The adversary is able to intercept messages between the central parameter server and each client.
- The adversary is able to plant concealed 'enemy' clients or coerce existing clients to reveal their local data and/or plant data which is to be used to train the global model.
- The adversary is able to masquerade as the parameter server, not only sending each client messages but also reading messages from all of the clients.

Fig. 3.1 summarises the fully generalised context of this project.

Different methods of adapting PVI will have different implications as to which parties are or are not trusted. There may be asymmetric privacy guarantees, conferring different levels of privacy depending on what the specific adversary is able to access. We will remark on these implications in the following sections.

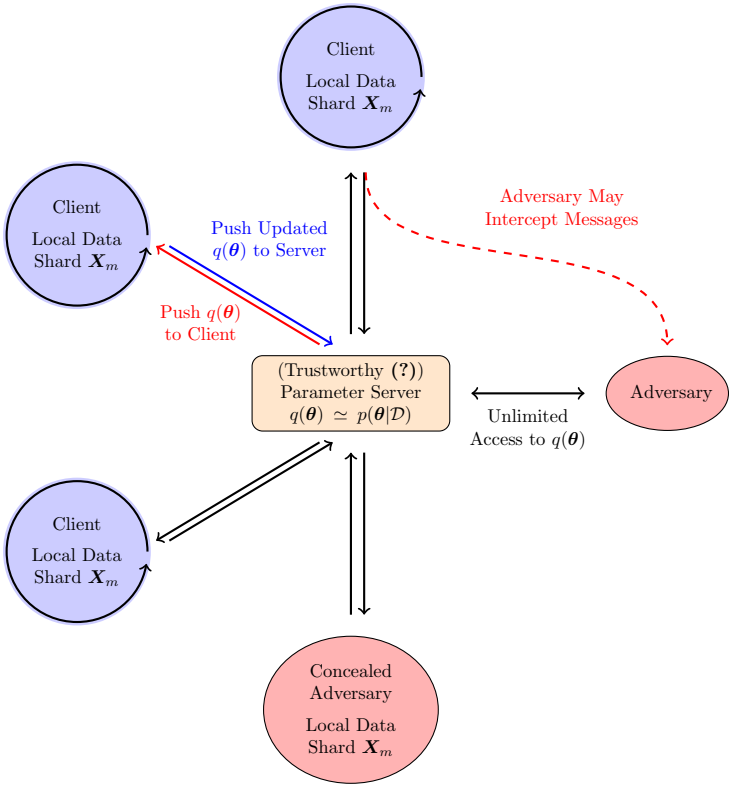


Fig. 3.1 Generalised Project Context Summary: each clients has local, sensitive data. We assume that the adversary has unlimited access to the global model, $q(\boldsymbol{\theta})$, and can intercept messages. A client may not know whether the parameter server is trustworthy.

3.2 Forms of DP-PVI

3.2.1 Datapoint Level DP-PVI

A simple method to construct DP-PVI is to perform the local free energy optimisation (Eq. **??**) in a privacy preserving manner, for example using DP-SGD (Algorithm **??**) with the loss function given by the negative local free energy. If this scheme were to be used, the client is protected against all parties as any external communication, namely variational distribution updates, will have been produced using a method which protects the data-points in each shard.

For clarity, assume that a data-shard is comprised of a number of data-points i.e. $\boldsymbol{X}_m = \{\boldsymbol{x}_1^{(m)}, \ldots, \boldsymbol{x}_{N_m}^{(m)}\}$. This scheme would limit the contribution of each $\boldsymbol{x}_i^{(m)}$ to the update of the variational distribution from each client and add noise corresponding to the the maximum possible contribution of each data-point.

Advantages of this scheme include there being no requirement for encryption on outgoing client messages and no authentication requirements i.e. the client would not need to ensure that it is indeed communicating with the a trustworthy parameter server. Additionally, each client is able to tune individual privacy settings depending on the level of protection desired and considerations from their local dataset.

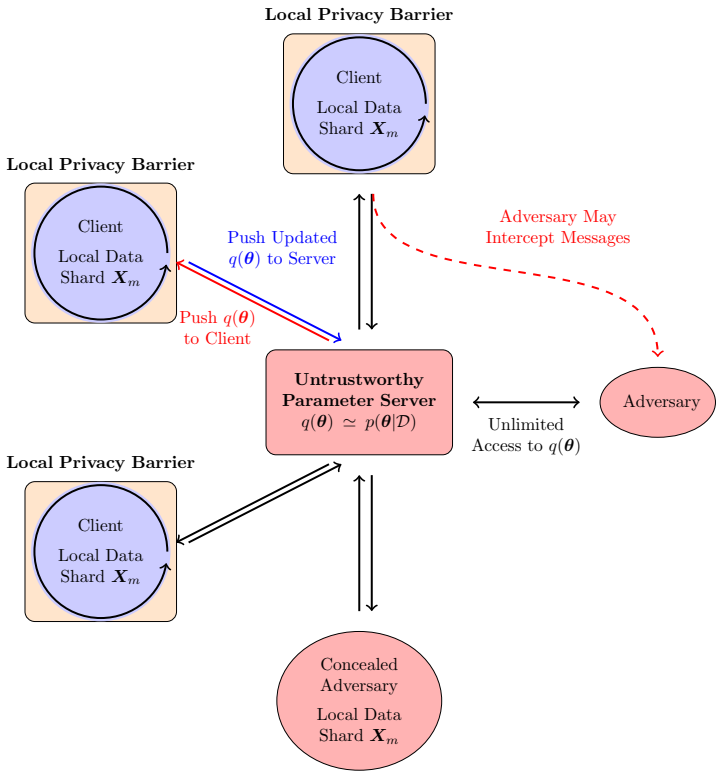


Fig. 3.2 Privacy barriers assumed by the datapoint level DP-PVI algorithm.

Fig. 3.2 summarises the privacy and trust barriers implied by the data-point level DP-PVI algorithm.

3.2.2 Dataset Level DP-PVI

Recalling the fundamental definition of differential privacy (Definition **??**), an alternative method of constructing DP-PVI is to consider one 'entry' of the dataset as an **entire data shard**, limiting the total contribution of each data shard to the variational distribution and then including corrupting noise.

Assume that the variational distribution is parametrised with parameters $\boldsymbol{\lambda} \in \Lambda$ and denote this variational distribution as $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$. We now present the Dataset Level DP-PVI algorithm, detailed in Algorithm 3, which is a key contribution of this work.

It is instructive to consider the effective update rule at the parameter server. Noting that the sum of Gaussian random variables is also Gaussian, Equations (3.4) and (3.6) are equivalent to the following update rule:

$$\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}^{(i-1)} + \alpha \cdot \left[\sum_{j=1}^{M} \frac{\Delta \boldsymbol{\lambda}_j}{\max(1, ||\Delta \boldsymbol{\lambda}_j||_2)} \right] + \sigma C \boldsymbol{z}$$
$$\text{with } z_k \overset{iid}{\sim} \mathcal{N}(0,1) \tag{3.7}$$

Algorithm 3 Dataset Level DP-PVI

Input: Partitioned Dataset $\mathcal{D} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M\}$, Prior $\boldsymbol{\lambda}^{(p)}$, Learning Rate $\alpha \in [0, 1]$, Clipping Bound C, DP Noise Scale σ

Initialise approximate likelihoods:

$$t_m^{(0)}(\boldsymbol{\theta}) \leftarrow 1 \ \forall m \tag{3.1}$$

Initialise parameters of variational distribution:

$$\boldsymbol{\lambda}^{(0)} \leftarrow \boldsymbol{\lambda}^{(p)} \tag{3.2}$$

for $i = 1, 2, \ldots,$ until convergence **do**
  for $j = 1, 2, \ldots, M$ **do** ▷ For each client
    Compute new parameters for this client:

$$\boldsymbol{\lambda}_j \leftarrow \underset{\boldsymbol{\lambda} \in \Lambda}{\operatorname{argmax}} \int q(\boldsymbol{\theta}|\boldsymbol{\lambda}) \ln \frac{q(\boldsymbol{\theta}|\boldsymbol{\lambda}^{(i-1)}) p(\boldsymbol{X}_j|\boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{\lambda}) t_j^{(i-1)}(\boldsymbol{\theta})} \, d\boldsymbol{\theta} \tag{3.3}$$

    $\Delta\boldsymbol{\lambda}_j \leftarrow \boldsymbol{\lambda}_j - \boldsymbol{\lambda}^{(i-1)}$
    Clip and corrupt update:

$$\tilde{\Delta}\boldsymbol{\lambda}_j = \alpha \cdot \left[\frac{\Delta\boldsymbol{\lambda}_j}{\max(1, ||\Delta\boldsymbol{\lambda}_j||_2 / C)} + \frac{\sigma C}{\sqrt{M}} \boldsymbol{z} \right]$$
$$\text{where } z_i \overset{iid}{\sim} \mathcal{N}(0, 1) \tag{3.4}$$

    $\boldsymbol{\lambda}_j \leftarrow \boldsymbol{\lambda}^{(i)} + \tilde{\Delta}\boldsymbol{\lambda}_j$
    Update the approximate likelihood:

$$t_j^{(i)}(\boldsymbol{\theta}) \leftarrow \frac{q(\boldsymbol{\theta}|\boldsymbol{\lambda}_j)}{q(\boldsymbol{\theta}|\boldsymbol{\lambda}^{(i-1)})} t_j^{(i-1)}(\boldsymbol{\theta}) \tag{3.5}$$

  end for
  Compute new global parameters:

$$\boldsymbol{\lambda}^{(i)} = \boldsymbol{\lambda}^{(i-1)} + \sum_{j=1}^{M} \tilde{\Delta}\boldsymbol{\lambda}_j \tag{3.6}$$

  Update privacy cost using the Moments Accountant.
end for

which uses the Gaussian mechanism to produce a differentially private estimate of the parameter update and applies a partial update. Unlike in gradient descent methods, the value of the learning rate is meaningful as a full parameter update (neglectinag clipping) would give the current (noisy) best guess of the optimal parameter settings whilst in gradient descent methods, the learning rate controls step sizes without, *a priori*, any indication of the optimal step size. Note that the learning rate is in the range $[0, 1]$.

The rationale behind distributing the central noise across each client and performing clipping locally is to ensure that each client is able to accurately track their contribution to the global variational distribution; if clipping and noise corruption occurred centrally, when calculating the new global parameters, the local approximate likelihood terms would become out-of-sync with the global variational distribution, resulting incorrect parameter values and poor model performance. This is similar to the technique used in (**?**), with the key difference that this occurs within each client.

A variant on this scheme would instead have each client transmit exact parameter updates to the central parameter server. Clipping and noise corruption would then occur at the server, **which would be required to recalculate the approximate likelihood term for each client**. Whilst the clipping of the update from each client naturally corresponds to an approximate likelihood which the server would then have to recalculate, there is freedom in precisely how the noise is incorporated into the client approximate likelihoods.

Note that Eq. (3.4) is also applying the Gaussian mechanism to the update from each client as the clipping fixes the ℓ_2 sensitivity. Recalling Eq. (**??**), it can be seen that for fixed δ, the division of the variance by M weakens the privacy guarantee by a factor of $\sqrt{M}$. Thus for large M, the messages from client to server will effectively be insecure, meaning that **this proposal relies on the existence of secure encryption methods** to prevent the adversary intercepting these messages.

The distribution of noise has additional consequences. Firstly, updates from several clients must be performed at once; if updates are performed using messages from one client only, it is possible that other clients will be able to infer sensitive information from the variational distribution update. There is freedom in choosing how many clients participate at each update. Noting that the variance of noise added is independent of the number of clients, updating using a smaller number of clients corresponds to smaller signal-to-noise ratios but the update from each client will take into account the information from other clients better (as the updates from each client will affect the updates which other clients provide). Secondly, since each client is aware of the noise they have contributed, they would be able to remove this noise from the global parameter update. Therefore, from the point of view of a client, the variance of the noise protecting other clients is reduced by a factor of $\frac{M-1}{M}$. For a large number of clients, **assuming that each concealed adversary is not in collusion**, this will have a small effect on the privacy guarantees provided. However, this raises privacy concerns in cases where it is possible that the adversary is able to conceal

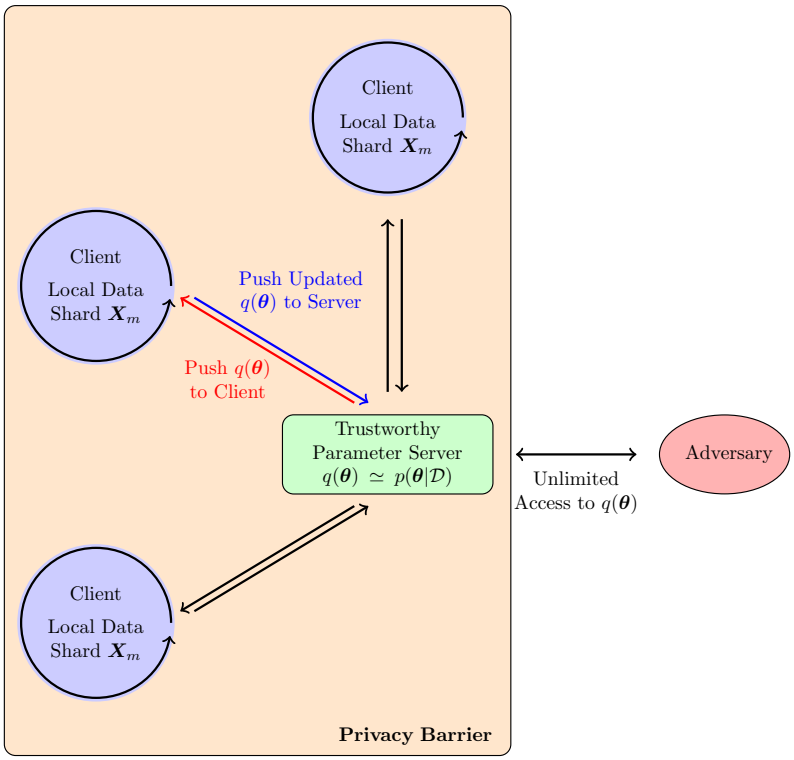


Fig. 3.3 Privacy Barriers assumed by the dataset level DP-PVI algorithm.

themselves and place multiple clients in the system. For instance, an incredibly simplified example is a situation where the clients consist of one genuine client and a large number of concealed adversaries in collusion. In this case, the adversaries would be able to collude to remove almost all of the noise added by the parameter server rendering the sensitive information of the genuine client insecure.

Additionally, we remark that this techniques requires that each client is able to ensure the authenticity of the parameter server; if the adversary masqueraded as the parameter server, it could simply inform each client that there are a very large number of clients, after which access to each message will enable the adversary to recover sensitive information about each client. Therefore, it is suggested that this technique should only be applied in situations where it is difficult for an adversary to conceal 'enemy' clients.

Fig. 3.3 summaries the trust and privacy barrier assumptions implied by the dataset level DP-PVI algorithm.

3.2.3 Comparison of Dataset and Datapoint Level Protection

There are significant differences between which parties are assumed to be trustworthy between the dataset and data-point level protection schemes. In practice, if choosing between these schemes, this will be a key factor in assessing which scheme is appropriate.

Besides the differences in implied trust and privacy barriers, there is a crucial difference in the type of protection offered by each scheme. The data-point level protection scheme

considers neighbouring datasets to be those which have differing individual data-points whilst the dataset level allows for differences in an entire data-shard. We now mathematically compare the protection offered by these schemes.

Let $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^N$ and let $\mathcal{D}^{(t)}$ denote a dataset which differs from $\mathcal{D}$ in t entries. Consider the definition of data-point level differential privacy i.e. assume that neighbouring datasets have one value of $\boldsymbol{x}_i$ that differs:

$$\begin{aligned}
\Pr(\mathcal{A}(\mathcal{D}) \in S) &\leq e^{\epsilon}\Pr(\mathcal{A}(\mathcal{D}^{(1)}) \in S) + \delta \\
&\leq e^{2\epsilon}\Pr(\mathcal{A}(\mathcal{D}^{(2)}) \in S) + e^{\epsilon}\delta + \delta \\
&\vdots \\
&\leq e^{t\epsilon}\Pr(\mathcal{A}(\mathcal{D}^{(t)}) \in S) + \delta\Big[1 + e^{\epsilon} + e^{2\epsilon} + \ldots + e^{(t-1)\epsilon}\Big] \\
&\leq e^{t\epsilon}\Pr(\mathcal{A}(\mathcal{D}^{(t)}) \in S) + \delta\frac{1-e^{t\epsilon}}{1-e^{\epsilon}} \qquad (3.8)
\end{aligned}$$

Therefore, if we protect each data-point with (ϵ, δ) differentially privacy, a group of t data-points is protected with $(t\epsilon, \delta(1-e^{t\epsilon})/(1-e^{\epsilon}))$ differential privacy. The dataset level DP-PVI approach can be regarded as providing an (ϵ, δ) guarantee on any changes within each shard which is itself made up of N data-points. Whilst the above analysis cannot be used to convert the group (ϵ, δ) pair into a (ϵ, δ) pair at the data-point level, we can interpret a dataset level (ϵ, δ) guarantee as being 'stronger' than the equivalent data-point level protection; the data-point level protection schemes (probabilistically) bounds the magnitude of the privacy loss for events which are a subset of the events considered by the dataset level protection scheme. Intuitively, it ought to be more difficult to disguise changes of an entire data-shard compared to a single data-point, matching the above analysis which shows that data-point level (ϵ, δ) protection corresponds to a weaker privacy guarantee.

Additionally, we remark that in cases where the each data-point in a client's data shard corresponds to an individual, the data-point level differential privacy approach is the more natural way to quantify privacy offered; in this case, the (ϵ, δ) guarantee corresponds directly to outcomes being 'similar' for datasets which neighbour in the sense that only the specific information about a single individual is different. Similarly for the case where an entire data shard corresponds to a single user, the data-point level approach seems to be excessive and the dataset level approach is more natural.

Furthermore, note that the dataset level DP-PVI scheme applies the same clipping bound to all of the workers and applies corrupting noise with a fixed standard deviation. In the case where the data is distributed differently at each client (i.e. inhomogeneous), the magnitude of parameter updates from each worker could be very different and applying the same clipping bound to each of the workers creates an inefficiency. In contrast, relevant parameters can be tuned for each worker individually for the data-point level DP-PVI algorithm, meaning that this technique is better suited for inhomogeneous data.

Chapter 4

Case Study: Bayesian Linear Regression

Here, both dataset and data-point level DP-PVI is applied to a Gaussian linear regression model. Whilst this model is simple, it remains a useful case study in order to understand the properties of these algorithms and assess whether these techniques provide reasonable performance.

4.1 Preliminaries

4.1.1 Model Definition

Data at each client is generated according to:

$$y_i = \theta x_i + \epsilon_i \qquad \epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_e^2) \tag{4.1}$$

where θ is a fixed, unknown parameter which is the same for every client. σ_e is assumed known. Each client has observations, $\boldsymbol{X}_m = \{(x_i^{(m)}, y_i^{(m)})\}_{i=1}^{N_m} = (\boldsymbol{x}_m, \boldsymbol{y}_m)$. Denote the entire dataset as $\mathcal{D} = \{\boldsymbol{X}_m\}_{m=1}^{M}$. A Gaussian prior is placed on θ:

$$p(\theta) = \mathcal{N}(\theta | \mu_\theta, \sigma_\theta^2) \tag{4.2}$$

The approximate likelihood factors take the form:

$$t_i(\theta) \propto \mathcal{N}(\mu_i, \sigma_i^2) \tag{4.3}$$

meaning that the approximate posterior, $q(\theta)$, is a Gaussian distribution. The aim is to find a $q(\theta)$ which is a good approximation to $p(\theta|\mathcal{D})$. Note that this posterior distribution is also a Gaussian distribution.

It is useful to express the univariate Gaussian distribution using *natural parameters*.

$$\begin{aligned}\mathcal{N}(x|\mu,\sigma^2) &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\Big(-\frac{1}{2\sigma^2}(x-\mu)^2\Big)\\ &= \frac{1}{\sqrt{2\pi}}\exp\Bigg(\underbrace{\begin{bmatrix}1/\sigma^2\\ \mu/\sigma^2\end{bmatrix}}_{\boldsymbol{\eta}}\cdot\underbrace{\begin{bmatrix}-x^2/2\\ x\end{bmatrix}}_{\boldsymbol{T}(x)}-(\frac{\mu^2}{2\sigma^2}-\ln\sigma)\Bigg)\\ &= h\exp[\boldsymbol{\eta}\cdot\boldsymbol{T}(x)-A(\boldsymbol{\eta})] \end{aligned} \tag{4.4}$$

with

$$A(\boldsymbol{\eta}) = \frac{\eta_2^2}{2\eta_1}-\frac{1}{2}\ln\eta_1 \tag{4.5}$$

η_1 and η_2 are known as the natural parameters of the distribution while $T(x)_1$ and $T(x)_2$ are known as the sufficient statistics. This representation allows the functional forms of the products and quotients of Gaussian distributions to be written straightforwardly:

$$\mathcal{N}(x|\boldsymbol{\eta}_1)\cdot\mathcal{N}(x|\boldsymbol{\eta}_2) \propto \mathcal{N}(x|\boldsymbol{\eta}_1+\boldsymbol{\eta}_2) \tag{4.6}$$

$$\mathcal{N}(x|\boldsymbol{\eta}_1)/\mathcal{N}(x|\boldsymbol{\eta}_2) \propto \mathcal{N}(x|\boldsymbol{\eta}_1-\boldsymbol{\eta}_2) \tag{4.7}$$

which may be seen by direct substitution. Note that the parameter $\eta_1 = 1/\sigma^2$ is known as the *precision*.

4.1.2 Analytical Update Equations

For this model, the update equations given by Equations (**??**) and (**??**) are analytical.

Recall Eq. (**??**) which reformulates the free energy maximisation as a $\mathcal{KL}$ minimisation between q and the tilted distribution. Let m be the client index. This $\mathcal{KL}$ is minimised when $q(\theta)$ is exactly equal to $\hat{p}_m^{(i)}(\theta)$. The tilted distribution takes the form

$$\begin{aligned}\hat{p}_m^{(i)}(\theta) &\propto \frac{q^{(i-1)}(\theta)}{t_m^{(i-1)}(\theta)}p(\boldsymbol{X}_m|\theta)\\ &\propto \underbrace{\mathcal{N}(\theta|\boldsymbol{\eta}_q^{(i-1)}-\boldsymbol{\eta}_m^{(i-1)})}_{\text{'prior'}}\overbrace{p(\boldsymbol{X}_m|\theta)}^{\text{likelihood}} = \mathcal{N}(\theta|\tilde{\boldsymbol{\eta}}) \end{aligned} \tag{4.8}$$

with

$$\tilde{\eta}_1 = \eta_{q,1}^{(i-1)} - \eta_{m,1}^{(i-1)} + \frac{\boldsymbol{x}_m^T \boldsymbol{x}_m}{\sigma_e^2} \tag{4.9}$$

$$\tilde{\eta}_2 = \eta_{q,2}^{(i-1)} - \eta_{m,2}^{(i-1)} + \frac{\boldsymbol{x}_m^T \boldsymbol{y}_m}{\sigma_e^2} \tag{4.10}$$

since this is equivalent to standard Bayesian linear regression. The equations for the exact posterior have been applied precisely (**?**). Thus, the free energy maximisation step is equivalent to setting $\boldsymbol{\eta}_q^{(i)} = \tilde{\boldsymbol{\eta}}$ as defined above.

The natural parameters of the approximate likelihood term are now straightforward to calculate as follows:

$$\boldsymbol{\eta}_m^{(i)} = \boldsymbol{\eta}_q^{(i)} + \boldsymbol{\eta}_m^{(i-1)} - \boldsymbol{\eta}_q^{(i-1)} \tag{4.11}$$

Note that since the approximate posterior, $q(\theta)$, must normalise, there is no need to keep track of the scale factors of the approximate likelihood terms; only the functional dependence upon θ must be stored.

4.1.3 Gradient of Local Free Energy

We now derive the gradient of the local free energy, defined in Eq. (**??**), with respect to the mean and variance of $q(\theta)$. For client m, the free energy is written:

$$\begin{aligned}\mathcal{F}_m^{(i)}(q(\theta)) &= \int q(\theta) \ln \frac{q^{(i-1)}(\theta) p(\boldsymbol{X}_m|\theta)}{q(\theta) t_m^{(i-1)}(\theta)} \, d\theta \\ &= \mathcal{H}[q] + \int q(\theta) \ln p(\boldsymbol{X}_m|\theta) \, d\theta + \int q(\theta) \ln \mathcal{C} \cdot \mathcal{N}(\theta|\boldsymbol{\eta}_q^{(i-1)} - \boldsymbol{\eta}_m^{(i-1)}) \, d\theta \end{aligned} \tag{4.12}$$

where $\mathcal{H}[q]$ is the *differential entropy* of q and $\mathcal{C}$ represents some constant (which does not have a fixed value in the following analysis). The differential entropy for a Gaussian random variable has a simple analytical form:

$$\mathcal{H}[q] = \frac{1}{2}(1 + \ln 2\pi\sigma_q^2) \tag{4.13}$$

(**?**) and thus its gradients are straightforward. By substitution, the likelihood term can be written as follows:

$$\begin{aligned}\int q(\theta) \ln p(\boldsymbol{X}_m|\theta) \, d\theta &= \mathcal{C} - \frac{1}{2\sigma_e^2} \int q(\theta) \left\{ \theta^2 \boldsymbol{x}_m^T \boldsymbol{x}_m - 2\theta \boldsymbol{x}_m^T \boldsymbol{y}_m \right\} d\theta \\ &= \mathcal{C} - \frac{1}{2\sigma_e^2} \left\{ (\mu_q^2 + \sigma_q^2) \boldsymbol{x}_m^T \boldsymbol{x}_m - 2\mu_q \boldsymbol{x}_m^T \boldsymbol{y}_m \right\} \end{aligned} \tag{4.14}$$

where the first and second moments of the Gaussian distribution have been directly substituted in. The gradients of this term are straightforward to evaluate. Let $\tilde{\mu}$ and $\tilde{\sigma}^2$ denote the mean and variance corresponding to $\boldsymbol{\eta}_q^{(i-1)} - \boldsymbol{\eta}_m^{(i-1)}$. Then, the final term can be written as:

$$\begin{aligned}\int q(\theta) \ln \mathcal{C} \cdot \mathcal{N}(\theta | \boldsymbol{\eta}_q^{(i-1)} - \boldsymbol{\eta}_m^{(i-1)}) \, d\theta &= \mathcal{C} - \frac{1}{2\tilde{\sigma}^2} \int q(\theta) \{\theta^2 - 2\tilde{\mu}\theta\} \, d\theta \\ &= \mathcal{C} - \frac{1}{2\tilde{\sigma}^2} \{\sigma_q^2 + \mu_q^2 - 2\tilde{\mu}\mu_q\} \end{aligned} \tag{4.15}$$

Combining the above expressions and taking derivatives yields the gradients of the local free energy:

$$\frac{\partial \mathcal{F}_m^{(i)}(q(\theta))}{\partial \mu_q} = -\frac{1}{\sigma_e^2}(\boldsymbol{x}_m^T \boldsymbol{x}_m \mu_q - \boldsymbol{x}_m^T \boldsymbol{y}_m) - \frac{1}{\tilde{\sigma}^2}(\mu_q - \tilde{\mu}) \tag{4.16}$$

$$\frac{\partial \mathcal{F}_m^{(i)}(q(\theta))}{\partial \sigma_q^2} = \frac{1}{2\sigma_q^2} - \frac{1}{2\sigma_e^2}(\boldsymbol{x}_m^T \boldsymbol{x}_m) - \frac{1}{2\tilde{\sigma}^2} \tag{4.17}$$

which can then be used in a gradient descent scheme. Note that since we seek to maximise the local free energy, we would perform gradient descent using the negative of the above gradients.

4.1.4 Assessing Performance

Performance of the DP-PVI technique algorithm is evaluated by computing the $\mathcal{KL}$ divergence between the approximate posterior produced by the algorithm and the posterior obtained non-privately using the entire combined dataset, computed using the exact analytical equations i.e. $\mathcal{KL}(q(\theta)||p(\theta|\mathcal{D}))$.

In practice, the corrupting noise applied means that the server parameters oscillate around. Typically, we average the $\mathcal{KL}$ divergence across the final ten iterations to reduce the variance of the performance metric used.

We note that this performance metric is not perfect. Intuitively, differential privacy seeks to limit (and obscure) the contribution that any one data-point has on the resulting model. We may expect to reduce this *overfitting*, which occurs when a machine learning model has been influenced 'too much' by a specific data-point. Therefore, we may expect differential privacy techniques to reduce this and improve *model generalisation*, that is, prediction performance on unseen data. We also note that this approach may also encourage under-fitting. For this case study, since such a simple linear probabilistic model only has one free parameter (and so is unlikely to over-fit) and the data was generated using a linear model, the $\mathcal{KL}$ divergence is a suitable metric. However, it is important to note that in general, this metric is imperfect.

4.1.5 Data Generation

For a fixed value of θ and σ_e, the data shard held at each client is generated using Eq. (4.1) with:

$$x_i \overset{iid}{\sim} \mathcal{N}(0,1) \tag{4.18}$$

For the purposes of this project, the dataset is *homogeneous* i.e. the underlying data distribution is the same for each of the clients. The number of clients is denoted as M and each client generates ρ data-points.

4.2 Datapoint Level DP-PVI

4.2.1 DP-SGD

With the gradient of the free energy calculated in the previous section, the free energy maximisation step PVI (Algorithm **??**) can be performed by applying DP-SGD (Algorithm **??**) to minimise negative local free energy to yield an algorithm which is differentially private which respect to each data-point in each data shard.

In our implementation, gradient descent is performed on the mean and the log of the variance in order to ensure that the variance remains positive; otherwise, large learning rates can give negative variances, after which gradient calculations are meaningless and the algorithm fails. Writing $\hat{\sigma}^2 = \ln \sigma_q^2$, the gradient of the free energy with respect to the log of the variance is written:

$$\frac{\partial \mathcal{F}_m^{(i)}(q(\theta))}{\partial \hat{\sigma}_q^2} = \frac{\partial \mathcal{F}_m^{(i)}(q(\theta))}{\partial \sigma_q^2} \cdot \exp(\hat{\sigma}_q^2) \tag{4.19}$$

Additionally, we apply a gradient thresholding step on the gradient of the precision which limits the magnitude of the gradient to a set value. This avoids problems with relatively large learning rates causing instability during the course of training.

Fig. 4.1 outlines typical results for this approach produced by hand tuning hyper-parameters. We remark that the form of noise involved in DP-SGD is problematic as there is no guarantee that the scale of gradients of different parameters is similar but the noise applied is isotropic, yielding different signal-to-noise ratios (SNRs) for each parameter. It is reassuring to see that this approach yields $\mathcal{KL}$ divergences which are small ($\sim 10^{-3}$), showing that we are able to achieve performance effectively equivalent to the non-private approach. Unfortunately, the privacy cost is very high, requiring $\epsilon \simeq 500$ for reasonable performance which far exceeds the maximum values found in literature ($\epsilon \simeq 10$).

Fig. 4.2 plots the privacy cost assuming a Gaussian mechanism with a varying sampling probability and shows there is a very large dependence of ϵ upon q. This figure suggests that

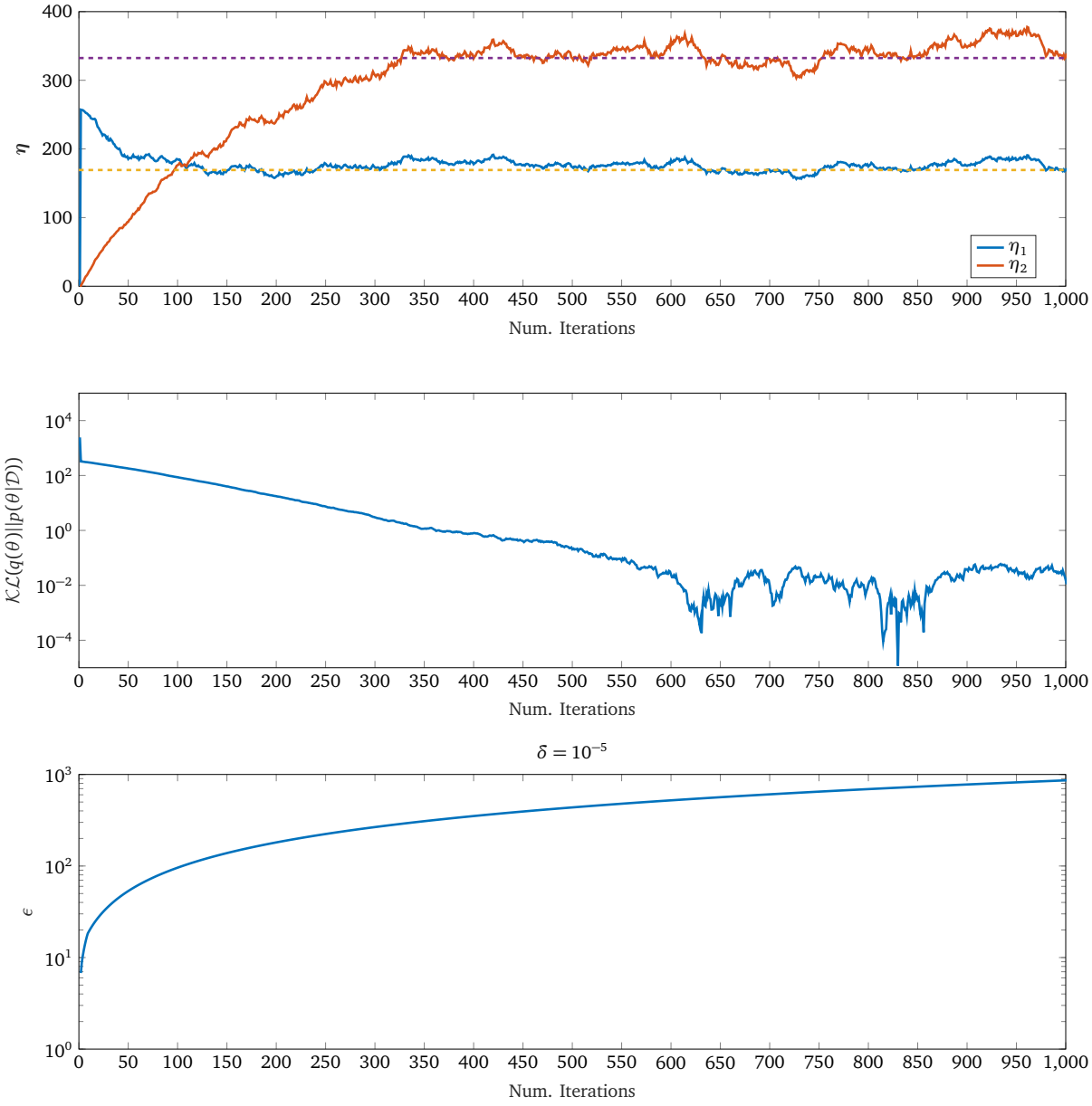


Fig. 4.1 Data-point level DP-PVI implemented using DP-SGD. First plot outlines evolution of the natural parameters stored at the central parameter server as the number of iterations increases. The $\mathcal{KL}$ divergence between $q(\theta)$ and the true posterior is also plotted (middle sub-plot), as is the privacy guarantee, ϵ, with $\delta = 10^{-5}$ fixed (bottom sub-plot). Parameters used: $\theta = 2$, $\sigma_e = 0.5$, $\mu_\theta = 0$, $\sigma_\theta = 5$. Learning rate $\alpha = 10^{-5}$, $L = 1$, $C = 10$, $\sigma = 1$. 5 workers with $\rho = 10$ points per worker. 50 iterations of DP-SGD performed locally for one server update.

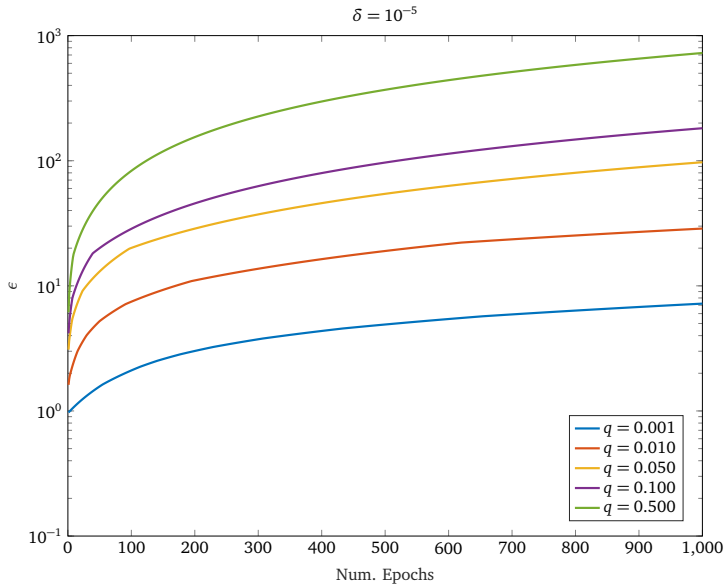


Fig. 4.2 Privacy cost for different sampling probabilities, $q = L/N$, as a function of the number of epochs assuming a mechanism which sub-samples from each dataset. An epoch is defined as the mechanism having been run processed $\frac{1}{q}$ times i.e. N data-points having been visited.

such an approach can only give strong privacy protection when there are a large number of data-points within each data shard which would enable a small value of q to also give a reliable gradient estimate. This is particularly difficult in the federated learning context as the data is distributed across clients, but will be appropriate in certain cases.

4.2.2 Analytical Updates

An alternative approach to perform the local free energy maximisation is to adapt the exact analytical equations to form a differentially private mechanism. Equations (4.9) and (4.10) can be modified as follows:

$$\ell_{m,n} = \sqrt{(x_{m,n}^2)^2 + (x_{m,n}y_{m,n})^2} \tag{4.20}$$

$$\tilde{\eta}_1 = \eta_{q,1}^{(i-1)} - \eta_{m,1}^{(i-1)} + \frac{1}{\sigma_e^2} \max\left\{0, \left[\sum_{n=1}^{N_m} \frac{x_{m,n}^2}{\max(1, \ell_{m,n}/C)} + \sigma C z_1\right]\right\} \tag{4.21}$$

$$\tilde{\eta}_2 = \eta_{q,2}^{(i-1)} - \eta_{m,2}^{(i-1)} + \frac{1}{\sigma_e^2}\left[\sum_{n=1}^{N_m} \frac{x_{m,n}y_{m,n}}{\max(1, \ell_{m,n}/C)} + \sigma C z_2\right] \tag{4.22}$$

$$z_j \overset{iid}{\sim} \mathcal{N}(0,1) \tag{4.23}$$

which bounds the ℓ_2 sensitivity by using clipping and applies the Gaussian mechanism with noise scaled according to the clipping bound, C. Unlike the gradient descent case, a mechanism employing sub-sampling is not appropriate here. A gradient produced by averaging the gradients of a subset of points remains an appropriate gradient estimate,

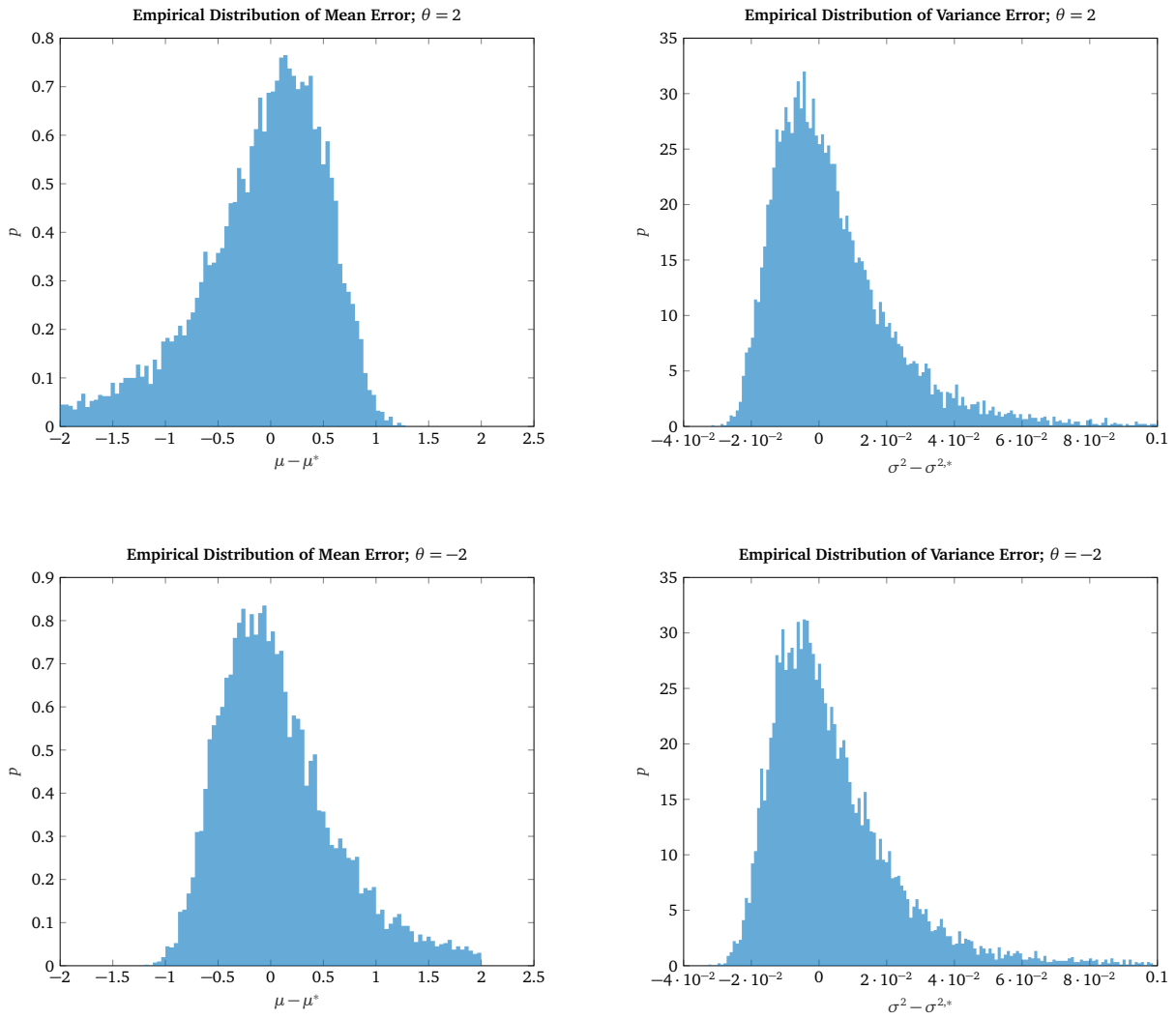


Fig. 4.3 Simulation on introduced bias on titled posterior calculation. Prior, $p(\theta) = \mathcal{N}(0, 5)$ with $\sigma_e = 0.5$, 10 data-points spaced uniformly in $[-1, 1]$. $N = 10000$ draws of differential privacy corrupting noise (with $\sigma = 1$) used to produce empirical distributions. Clipped neglected.

but computing the exact analytical equation with a subset of data-points does not yield an appropriate update. In Eq. (4.21), an additional clipping step is applied to ensure the contribution from the corrupted term remains positive (otherwise the precision may become negative which causes numerical problems) which is valid as differential privacy is immune to post-processing.

Neglecting the clipping to ensure the the precision remains positive, it is worth noting that the above update equations provide unbiased estimates for the natural parameters of the tilted posterior. However, this does not correspond to unbiased estimates of the mean and variance of the titled posterior. Fig. 4.3 shows that, assuming no clipping occurs, unbiasedly estimating the natural parameters of a Gaussian distribution gives an estimate with mean biased towards zero and an overestimation of the variance.

Since *a priori*, it is not known what the true ranges of each value of x and y will be, the choice of clipping bound, C, is crucial. If chosen too small, the solution obtained by each maximisation is incorrect and bias is introduced into the results. However, if chosen too large, parameter updates are dominated by noise. This issue can be avoided by rescaling the range and mean of numerical data, but this would also have to be done privately and is complicated by the fact that the data is distributed across client. This is not considered in this report.

Fig. 4.4 shows results obtained using this approach with a ($\epsilon \simeq 10, \delta = 10^{-5}$) privacy guarantee when varying the clipping bound. The DP noise scale, σ, is not varied as it can be immediately interpreted as controlling the trade-off between the signal-to-noise ratio and the privacy guarantee provided. It is immediately clear that increasing the clipping bound value tends to increase the variance of the $\mathcal{KL}$ divergence, likely due to the standard deviation of the corrupting noise scaling with the clipping bound. We also note that these standard deviations are very high and inspecting the bottom left figure, there are very large differences in performance for the same parameter settings across datasets. The median being significantly lower than the mean for all values of C shows that performance can be incredibly poor for certain datasets, which is far from optimal. Curiously, the best median performance achieved corresponds to $C = 0.25$, the smallest value used, which gives a median $\mathcal{KL}$ of approximately 22. Fig. 4.5 shows private posteriors and true posteriors corresponding to this level of $\mathcal{KL}$ divergence, showing reasonable performance. It is worth noting that the private posteriors underestimate the precision of the true posterior and have a mean value closer to zero, almost acting as a sort of regularisation. This method performs worse than the DP-SGD approach (which achieves effectively the non-private solution), but gives a much stronger privacy guarantee; $\epsilon \simeq 10$ is significantly stronger than $\epsilon \simeq 500$ and is also a value found in the literature. Additionally, we note that increasing the value of the clipping bound tends to decrease performance as it corresponds to adding significantly larger values of noise.

The value of θ has a number of implications for the performance of the algorithm. Inspecting the top left plot, for moderate values of C, the ratio between the true precision and obtained precision decreases as θ rises, which can be understood as follows. Assuming fixed values of $\boldsymbol{x}$ and σ_e, changing θ directly affects η_2 but does not affect η_1. Therefore a larger value of θ corresponds to a larger value of η_2 and thus a larger value of $\ell_{m,n}$, meaning that the clipping applied is more aggressive. This yields a smaller value for the precision and a more distorted approximate solution, increasing the $\mathcal{KL}$ divergence.

Worryingly, choosing C to be very large shows a large overestimation of the precision, suggesting that this approach gives a systematic bias for inappropriate parameter settings. Recalling Eq. 4.21, the cause of this bias can be seen. Assuming that $C > \ell_{m,n}\ \forall n$, the update

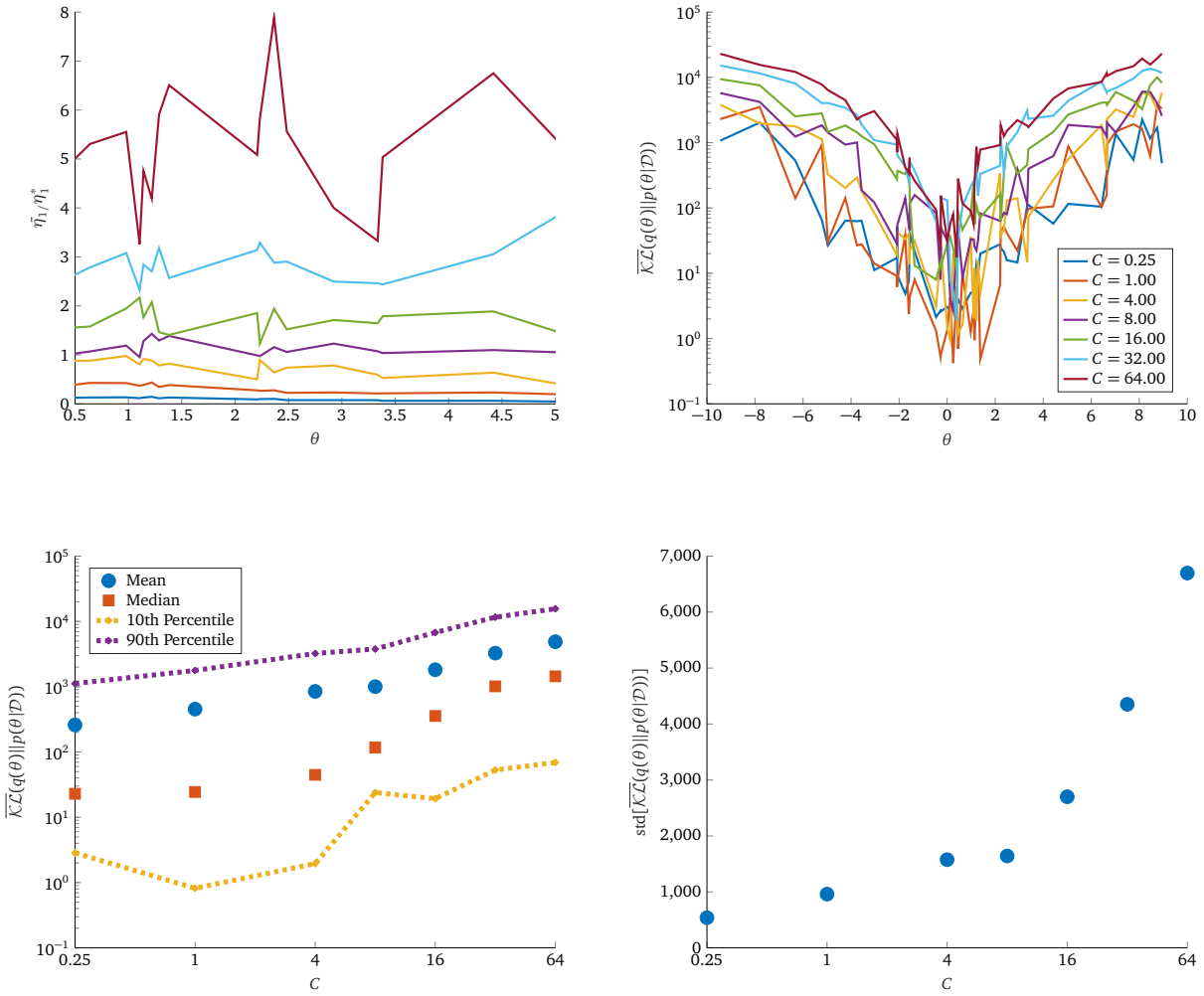


Fig. 4.4 Results obtained with data-point level differential privacy with $\epsilon_{\max} = 10$ (i.e. termination after ϵ exceeds this value). Top left plot shows ratio between the true precision and obtained precision as a function of θ, averaged over the last ten iterations of the DP-PVI algorithm. Top right plot shows $\mathcal{KL}(q||p)$ (again averaged over the ten last iterations) as a function of θ. Bottom left plots show the mean, median and chosen percentiles of this averaged $\mathcal{KL}$ divergence as a function of C (across the 50 different values of θ) and the bottom right plot shows the corresponding standard deviation. 50 random seeds used for each clipping bound value with each random seed corresponding to a specific θ sampled from the prior distribution, $p(\theta) = \mathcal{N}(0, 5)$ and specific draw of corrupting noise. $\sigma_e = 0.5$, $\sigma = 5$, $\alpha = 0.1$ with $M = 20$ workers and $\rho = 10$ data-points per worker.

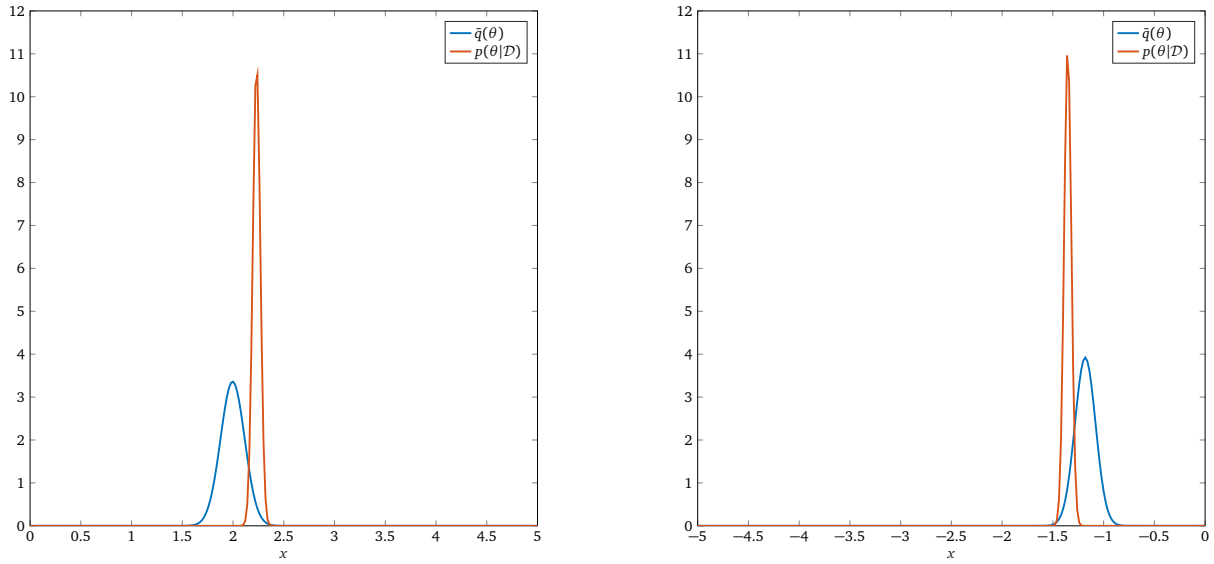


Fig. 4.5 Typical posteriors obtained using the data-point level DP-PVI algorithm with analytical clipped updates. Produced with $C = 2$, $\alpha = 0.1$, $M = 20$ workers with $\rho = 10$ data-points per worker. Averaged variational parameters across the final ten iterations used to produce above plot.

can be written as:

$$\begin{aligned}\tilde{\eta}_1 &= \eta_{q,1}^{(i-1)} - \eta_{m,1}^{(i-1)} + \frac{1}{\sigma_e^2} \underbrace{\max\left\{0, \left[\overbrace{\sum_{n=1}^{N_m} x_{m,n}^2}^{\Sigma_{xx}} + \sigma C z\right]\right\}}_{\Delta} \\ &= \eta_{q,1}^{(i-1)} - \eta_{m,1}^{(i-1)} + \frac{1}{\sigma_e^2}\Delta \end{aligned} \tag{4.24}$$

where $z \sim \mathcal{N}(0,1)$. There would be no systematic bias if $\mathbb{E}[\Delta] = \Sigma_{xx}$. Taking $\boldsymbol{X}_m$ to be fixed:

$$\Delta = \max\{0, \Sigma_{xx} + \sigma C z)\} \tag{4.25}$$

and thus the distribution of Δ can be written by inspection as follows:

$$p(\Delta = x) = \Phi\Big(\frac{-\Sigma_{xx}}{\sigma C}\Big)\delta(x) + \begin{cases} 0 & x \leq 0 \\ \mathcal{N}(x|\Sigma_{xx}, \sigma^2 C^2) & x > 0 \end{cases} \tag{4.26}$$

where $\Phi(\cdot)$ denotes the cumulative density function of the standard Gaussian distribution. This distribution clearly has mean larger than Σ_{xx} as negative contributions in the first moment from are replaced with a zero contribution due to the delta function. As the value of C increases, the negative contribution which is removed increases in value, resulting in a larger bias, which matches the obtained results.

Fig. 4.6 shows results for the same parameter settings as Fig. 4.4 maintaining clipping but removing the corrupting DP noise. In the top-left plot, the largest clipping bounds do not overestimate the precision, confirming that it is the noise causing the systematic bias in this parameter. For smaller values of the clipping bound, C, increasing the value of θ decreases the value of the precision and tends to decrease model performance as the effect of clipping becomes larger. As expected, increasing the clipping bound increases model performance when no noise is applied since the only drawback of increasing the clipping bound is the increased noise standard deviation.

Fig. 4.7 shows typical approximate posteriors obtained for the extremes of the clipping bound when no DP noise is applied; large C gives practically unmodified posteriors as expected whilst small values of C show the previously observed pattern of underestimating the precision and absolute value of θ.

Whilst further gains in terms of reduction of the privacy expenditure could likely be made by a more comprehensive search over hyper-parameters, this scheme does not give particularly good performance in practice for this model. However, it may be practical to use this in certain contexts with a small clipping bound in order to avoid the overestimation of the precision. It is then likely that the approximate posterior formed will have a larger variance and smaller mean (in terms of absolute value) than the true posterior, but since the bias appears to be systematic, the machine learning practitioner may be able to correct for this or at the minimum be aware of the consequences.

4.2.3 Hydrid Scheme

Whilst the DP-SGD scheme is able to provide performance indistinguishable from the non-private performance, it does so at very high privacy costs. On the other hand, using analytical updates to create a DP mechanism provides an approximate, biased solution but at low privacy costs. These approaches could be combined straightforwardly, using the analytical update to initialise the DP-SGD scheme in a suitable position, which would then require fewer iterations to reach the optimum solution.

It is likely that this would achieve performance close to the DP-SGD approach at significantly lower privacy costs, though it is unclear exactly how large the privacy gains would be. It is suggested that this approach, which could be applied in general for exponential conjugate models, should be investigated further.

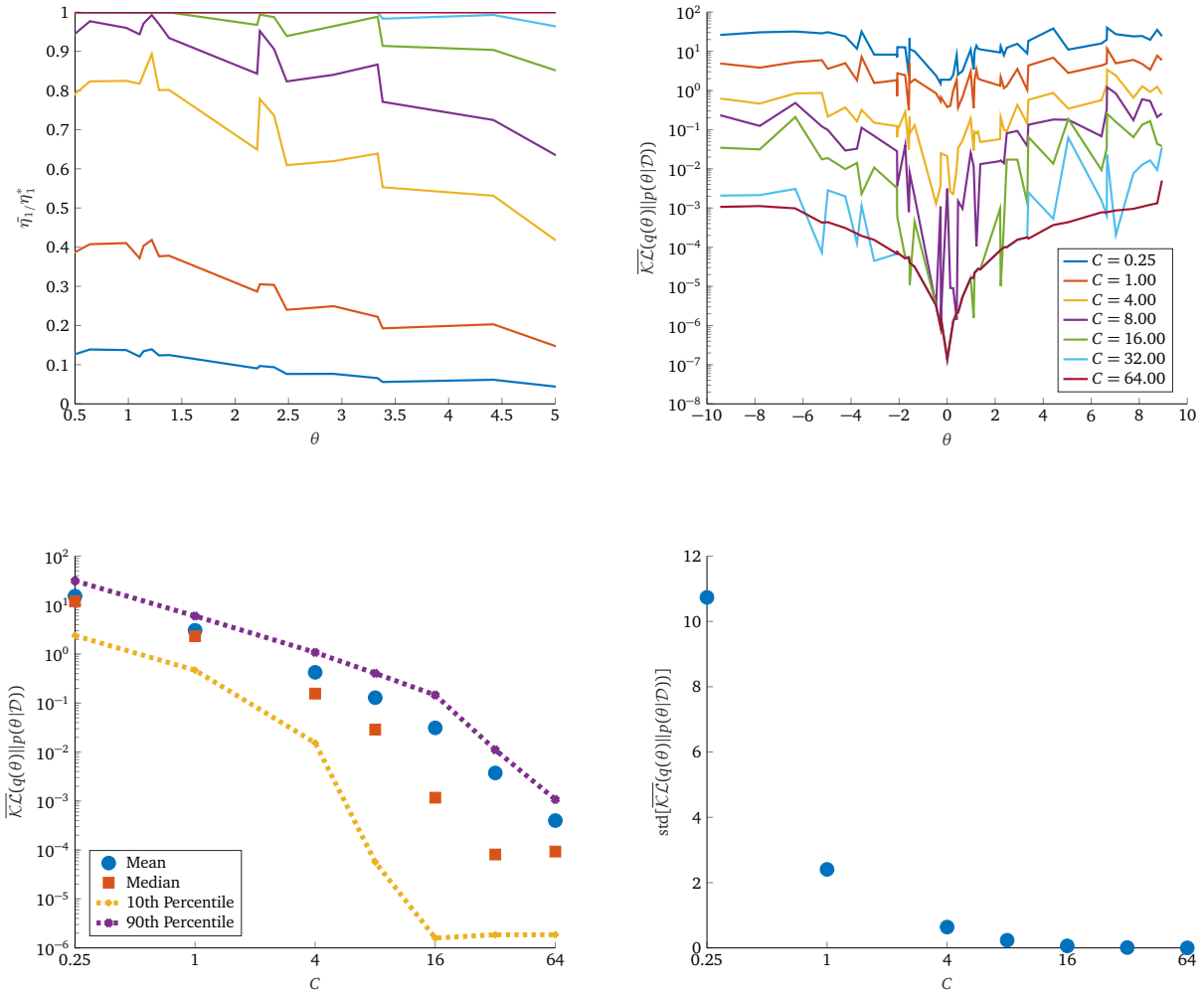


Fig. 4.6 Results obtained with data-point level differential privacy applied **without noise** when run for the same number of iterations as Fig. 4.4. Top left plot shows ratio between the true precision and obtained precision as a function of θ, averaged over the last ten iterations of the DP-PVI algorithm. Top right plot shows the $\mathcal{KL}(q||p)$ (again averaged over the ten last iterations) as a function of θ. Bottom left plots show the mean, median and chosen percentiles of this averaged $\mathcal{KL}$ divergence as a function of C (across the 50 different values of θ) and the bottom right plot shows the corresponding standard deviation. 50 random seeds used for each clipping bound value with each random seed corresponding to a specific θ sampled from the prior distribution, $p(\theta) = \mathcal{N}(0, 5)$ and specific draw of corrupting noise. $\sigma_e = 0.5$, $\alpha = 0.1$ with $M = 20$ workers and 10 data-points per worker.

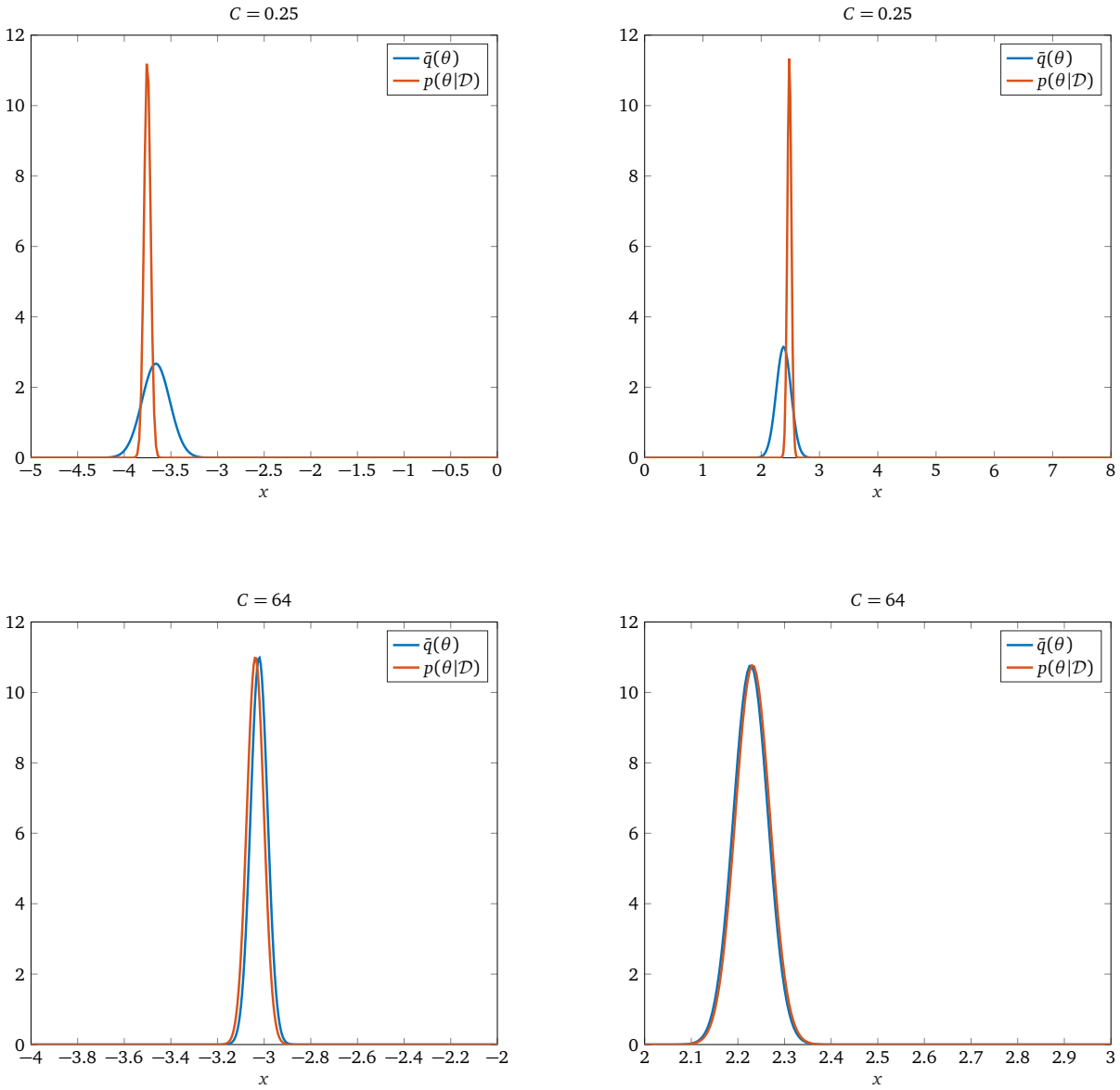


Fig. 4.7 Typical posteriors obtained using the **noiseless** data-point level DP-PVI algorithm with analytical clipped updates. Produced with $C = 2$, $\alpha = 0.1$, $M = 20$ workers with 10 data-points per worker. Averaged variational parameters across the final ten iterations used to produce above plots.

4.3 Dataset Level DP-PVI

4.3.1 Analytical Updates

Algorithm 3 can applied to this probabilistic model directly using the analytical update equations derived previously. In our implementation, the variational distribution is parametrised using its natural parameters and thus the clipping and corruption step (Eq. 3.4) affects the precision and the product of the mean and precision directly. Additionally, the update corresponding to the precision is clipped at each worker in order to ensure that the corresponding to the precision of each approximate likelihood (i.e. $\eta_1^{(m)}$ for each client) cannot become negative.

Fig. 4.8 shows results obtained using the dataset level DP-PVI algorithm for different values of C and α. The DP noise scale, σ, is not varied as it can be immediately interpreted as controlling the trade-off between the signal-to-noise ratio and the privacy guarantee provided. We remark that if the clipping bound value is chosen too large, the value found for the precision, η_1, is biased and overestimated similar to the analytical updates for the data-point level protection. Crucially, unlike the data-point level DP-PVI algorithm, the clipping does not introduce a bias if chosen too small as **only parameter updates are clipped** rather than terms determining the exact solution. This is confirmed by inspecting the top left subplot of Fig. 4.10 which not only shows that the removal of noise removes the overestimation of the precision but also that when the clipping bound is chosen relatively small, a precision close to the true precision is also recovered. Indeed, performance is very similar for all of the clipping bounds investigated when no DP noise is applied, with very small $\mathcal{KL}$ divergences being reached. The drawback to choosing the clipping bound too small however is that the magnitude of the update at each iteration will be smaller, resulting in a larger number of iterations being required to reach convergence which would correspond to a weaker privacy guarantee.

The pattern of smaller values of θ corresponding to smaller values of the $\mathcal{KL}$ divergence is again observed, both in the noisy and noise-free cases. Whilst the magnitude of η_2 increases with θ, which means that fluctuations in η_1 give larger changes in the mean parameter as follows:

$$\mu = \frac{\eta_2}{\eta_1} \Rightarrow \frac{\partial \mu}{\partial \eta_1} \propto -\eta_2, \tag{4.27}$$

This is not sufficient to explain this behaviour as it also occurs for the noiseless example, but will contribute to the behaviour when the DP noise is applied. This behaviour can also in part be explained by the number of iterations being insufficient to reach convergence for the largest values of θ.

The chosen parameter settings have a large effect on the performance achieved. Larger values of C increase the magnitude of the corrupting noise and thus also increase the bias

of the precision parameter, worsening performance. Additionally, large values of noise make the variational distribution very unstable which makes it more difficult to choose parameter settings. If the clipping bound is relatively large, increasing the learning rate tends to decrease performance as the updates become more sensitive to the specific random noise applied.

Fig. 4.9 shows typical posteriors obtained with the settings $C = 5$ and $\alpha = 0.1$. Additionally, Fig. 4.11 shows training curves for different parameter settings where the dataset level DP-PVI algorithm has worked well. Since C and α are fixed, the parameters of the approximate posterior remain noisy and tend to fluctuate around the non-private values.

It is instructive to inspect the evolution of approximate posterior parameters where the dataset level DP-PVI algorithm has performed poorly and has consumed the privacy budget while not reaching a 'good' approximate posterior, as plotted in Fig. 4.12 for different parameter settings. Crucially, there appear to be two failure modes of this technique:

1. If the clipping bound, C, and learning rate, α, are chosen too small, the algorithm does not converge when the privacy budget is consumed. As a result, the averaged variational distribution obtained is not close to the non-private posterior, though it would appear that increasing the privacy budget would remedy this issue (as would increasing the learning rate and clipping bound).

2. If the clipping bound, C, and learning rate, α, are chosen to be too large, the values of the precision are overestimated by a large amount due to the noise random variable effectively being truncated, as previously discussed. Inspecting the bottom three plots of Fig. 4.12, the value of η_2 fluctuates around the non-private value whilst η_1 fluctuates above the non-private value. This gives poor performance. Additionally, the parameters of the variational distribution fluctuate by large amounts, meaning that performance with the parameter settings after a given number of iterations will also fluctuate a large amount.

4.3.2 Robustness Study

We investigate running the dataset DP-PVI scheme whilst varying the number of points per worker, ρ, and the clipping bound, C. From previous results, for $\rho = 10$, it would appear that $C = 5$ and $\alpha = 0.1$ is an appropriate setting for the algorithm parameters. Recalling Equations (4.9) and (4.10), we note that the local contributions from each worker scale linearly with the number of data-points at each worker. Therefore, *a priori*, we know that the magnitudes of the messages each worker will send to the parameter server scale with ρ, and therefore we choose to fix $C = 0.5\rho$, which is equivalent to $C = 5$ for $\rho = 10$ points per worker (the suggested setting). Changing the number of clients, M, does not affect the magnitude of the messages sent by each client so we set C independent of M.

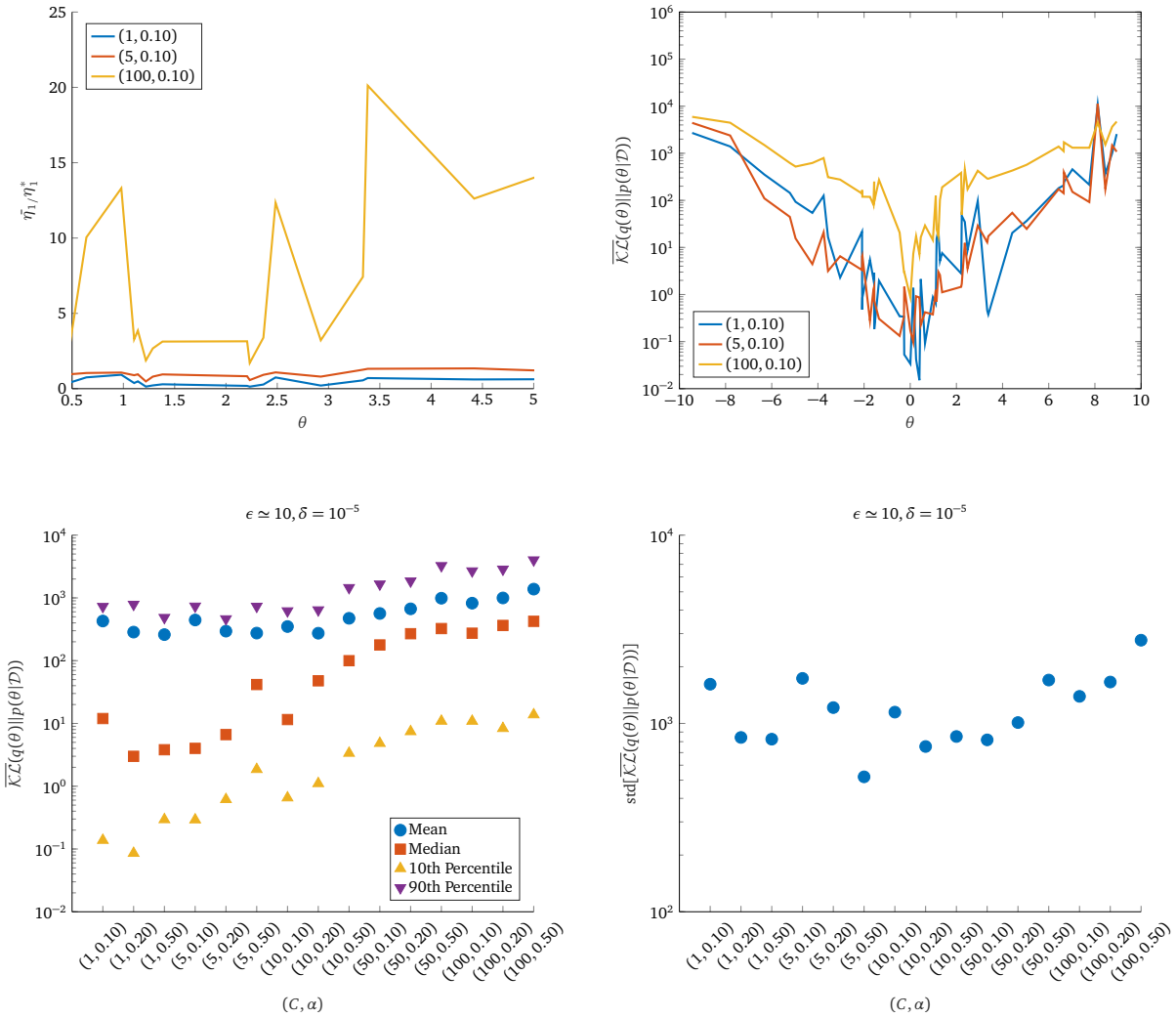


Fig. 4.8 Results obtained with dataset level differential privacy applied with $\epsilon_{\text{max}} = 10$ (termination after ϵ exceeds this value). Top left plot shows ratio between the true precision and obtained precision as a function of θ, averaged over the last ten iterations of the DP-PVI algorithm. Top right plot shows $\mathcal{KL}(q||p)$ (again averaged over the ten last iterations) as a function of θ for chosen settings of C and α. Bottom left plots show the mean, median and chosen percentiles of this averaged $\mathcal{KL}$ divergence for different algorithm settings (across the 50 different values of θ) and the bottom right plot shows the corresponding standard deviation. 50 random seeds used for each clipping bound value; each random seed corresponding to a specific θ sampled from the prior distribution, $p(\theta) = \mathcal{N}(0, 5)$, a specific draw of corrupting noise and a value of σ_e sampled from $\mathcal{U}(0.5, 2)$. $M = 20$ workers and $\rho = 10$ data-points per worker. $\sigma = 5$.

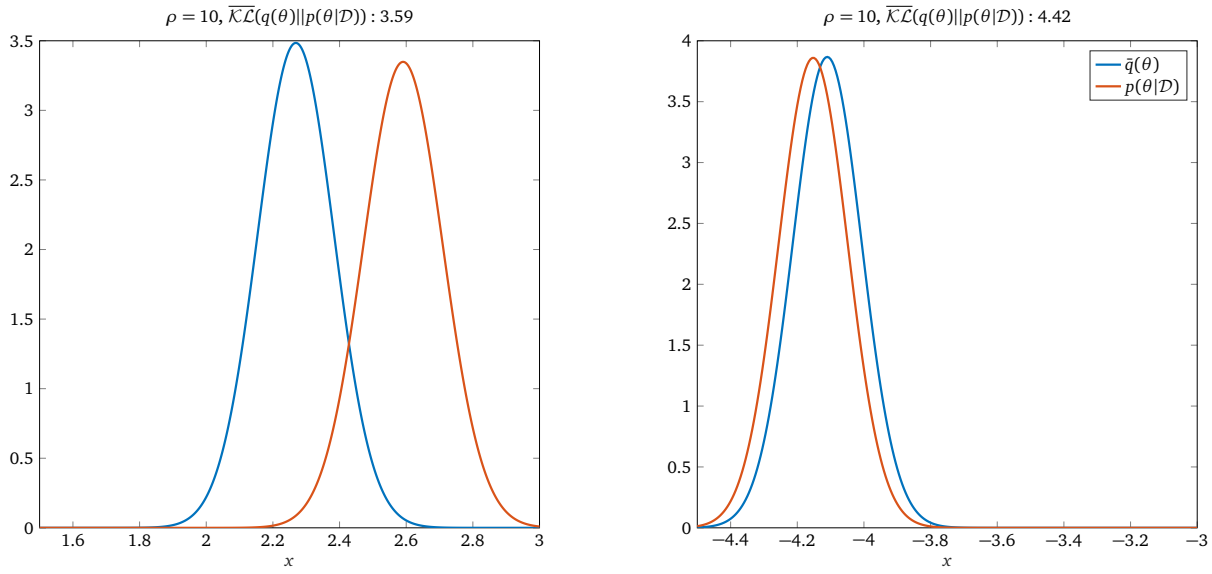


Fig. 4.9 Typical posteriors obtained the dataset level DP-PVI algorithm. Produced with $C = 5$, $\alpha = 0.1$, $M = 20$ workers with $\rho = 10$ data-points per worker. Variational distribution parameters averaged across the final ten iterations to produce plots.

Fig. 4.13 shows results obtained when varying M and ρ. Increasing the value of M improves the median performance achieved with this algorithm. Recalling Eq. (3.7) which effectively gives the total update at the parameter server, it can be seen that increasing M increases the magnitude of the update signal and thus increases the signal-to-noise ratio as the variance of the noise applied is fixed. With the exception of $M = 30$, the performance for the worst seeds tends to decrease. These seeds correspond to large values of θ for which the algorithm does not reach convergence when the privacy budget is exceeded. For a given value of θ and homogeneous data, the unclipped messages from each client are expected to be the same at each iteration. Therefore, increasing M scales the magnitude of the natural parameters linearly, which corresponds to no change in the mean but an increase in the precision. Therefore, the difference in mean between the variational distribution and non-private posterior is expected to be similar (since the privacy parameters fix the total number of iterations), but the larger values of precision cause these differences to be penalised by a larger amount. However, if convergence is reached, the variational distribution roughly reaches the non-private mean and precision. The larger value of the natural parameters gives a smaller signal-to-noise ratio (considering the DP noise), decreasing the median $\mathcal{KL}$ divergence value.

Increasing the number of points per worker tends to decrease performance in general; the median and mean values as well as the 10th and 90th percentiles of the $\mathcal{KL}$ divergence all increase. As ρ increases, the overall precision increases. This causes larger penalties in terms of the $\mathcal{KL}$ divergence for a given mismatch in the mean value, giving a large increase of $\mathcal{KL}$ for the worst datasets. Fig. 4.14 plots typical posteriors for $\rho = 10$ and $\rho = 90$ which seem to suggest that mismatch between the private and non-private mean appears

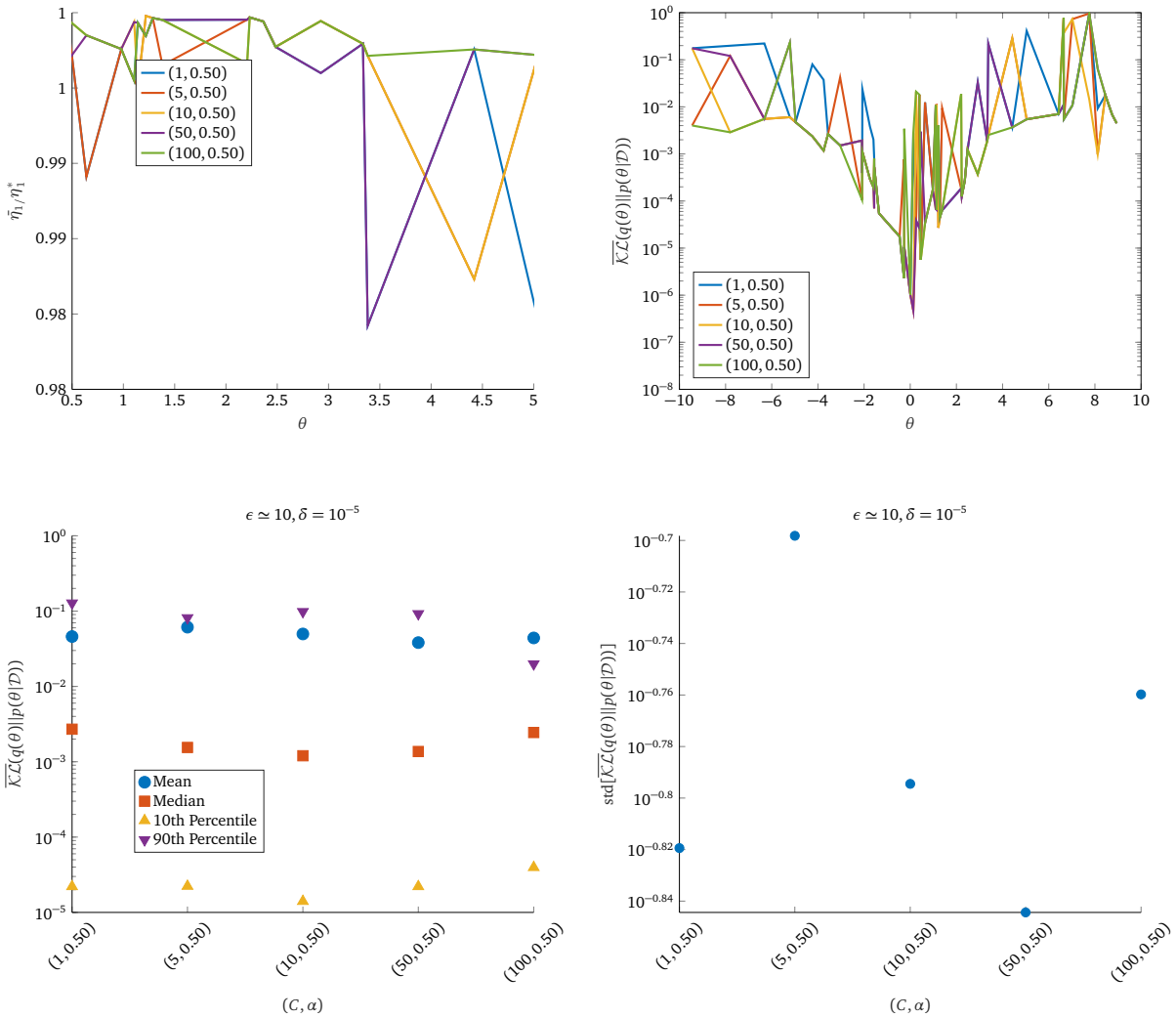


Fig. 4.10 Results obtained with the dataset level DP-PVI algorithm **without adding noise**. The number of iterations performed is increased to 1000 instead of terminating after the privacy budget is consumed. Top left plot shows ratio between the true precision and obtained precision as a function of θ, averaged over the last ten iterations of the DP-PVI algorithm. Top right plot shows the $\mathcal{KL}(q||p)$ (again averaged over the ten last iterations) as a function of θ for chosen settings of C and α. Bottom left plots show the mean, median and chosen percentiles of this averaged $\mathcal{KL}$ divergence for different algorithm settings (across the 50 different values of θ) and the bottom right plot shows the corresponding standard deviation. 50 random seeds used for each clipping bound value; each random seed corresponding to a specific θ sampled from the prior distribution, $p(\theta) = \mathcal{N}(0, 5)$, a specific draw of corrupting noise and a value of σ_e sampled from $\mathcal{U}(0.5, 2)$. $M = 20$ workers and $\rho = 10$ data-points per worker. $\sigma = 5$.

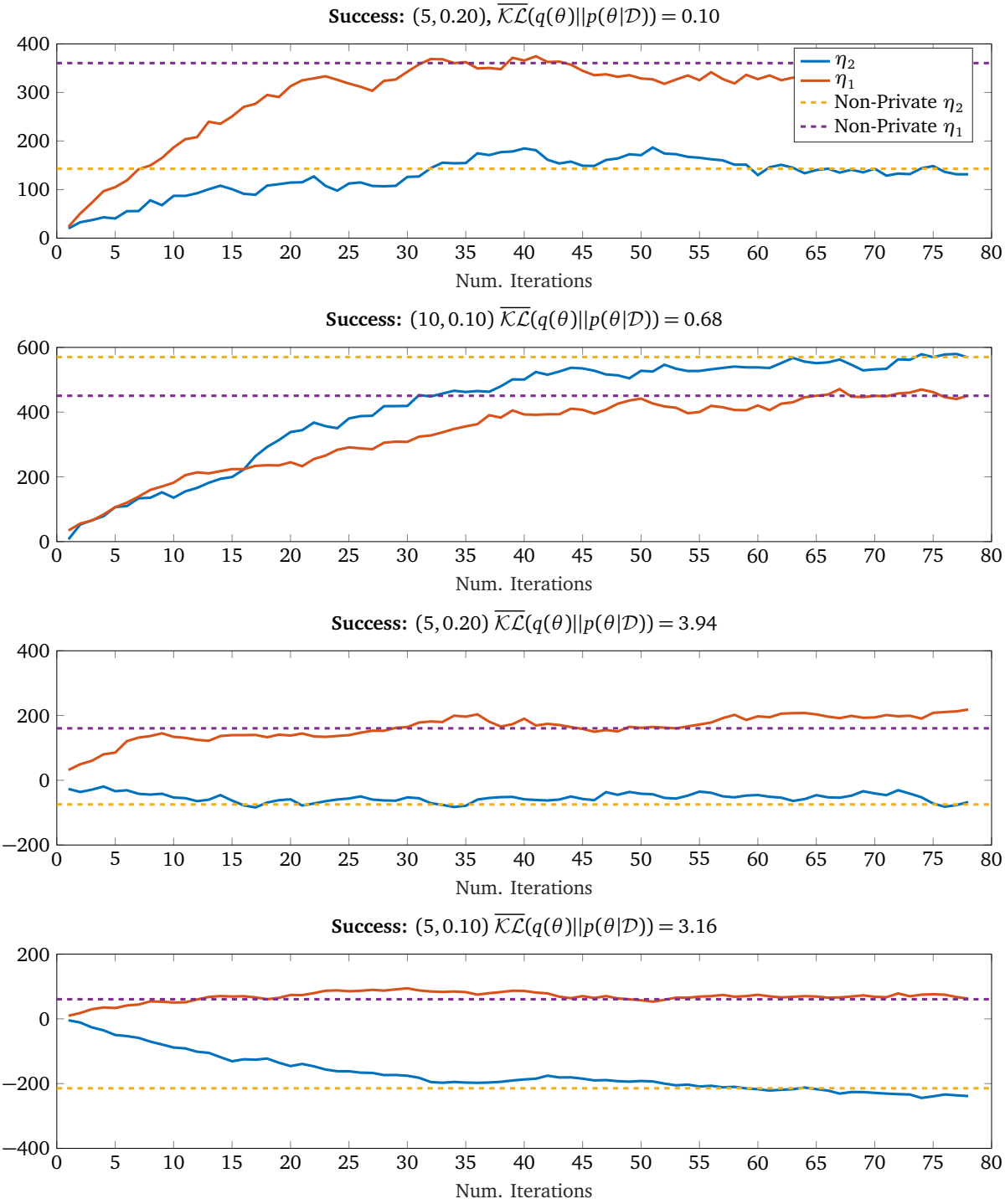


Fig. 4.11 Training curves showing the evolution of natural parameters as the number of iterations (performed at the parameter server) increases for random seeds which resulted in good approximate posteriors. Dashed lines correspond to non-private values. $M = 20$ clients, $\rho = 10$ points per worker. $\epsilon \simeq 10, \delta = 10^{-5}$. Plot title values refer to values of (C, α).

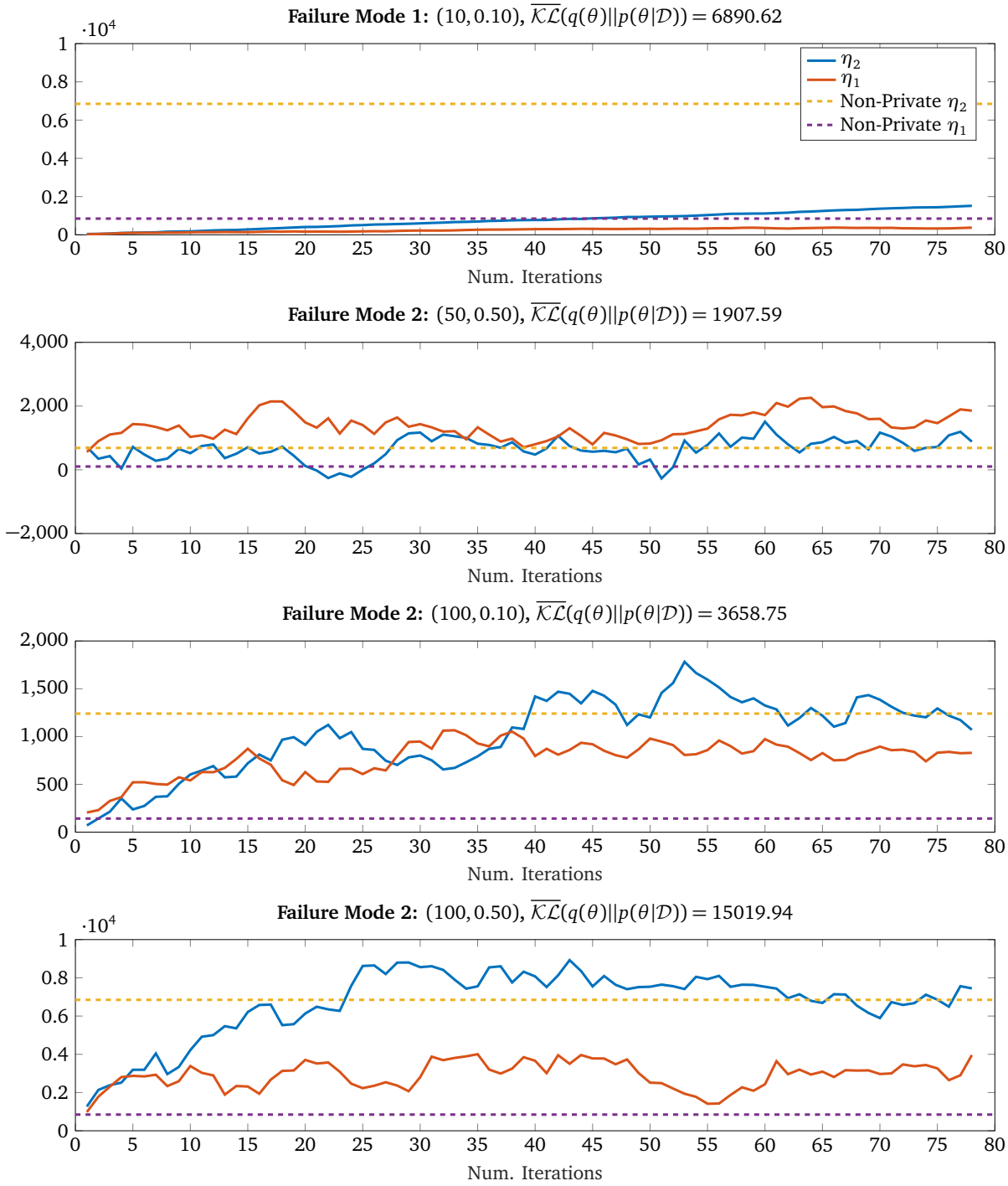


Fig. 4.12 Training curves showing the evolution of natural parameters as the number of iterations (performed at the parameter server) increases for random seeds which resulted in poor approximate posteriors. Dashed lines correspond to non-private values. $M = 20$ clients, $\rho = 10$ points per worker. $\epsilon \simeq 10, \delta = 10^{-5}$. Sub-plot titles refer to values of (C, α).

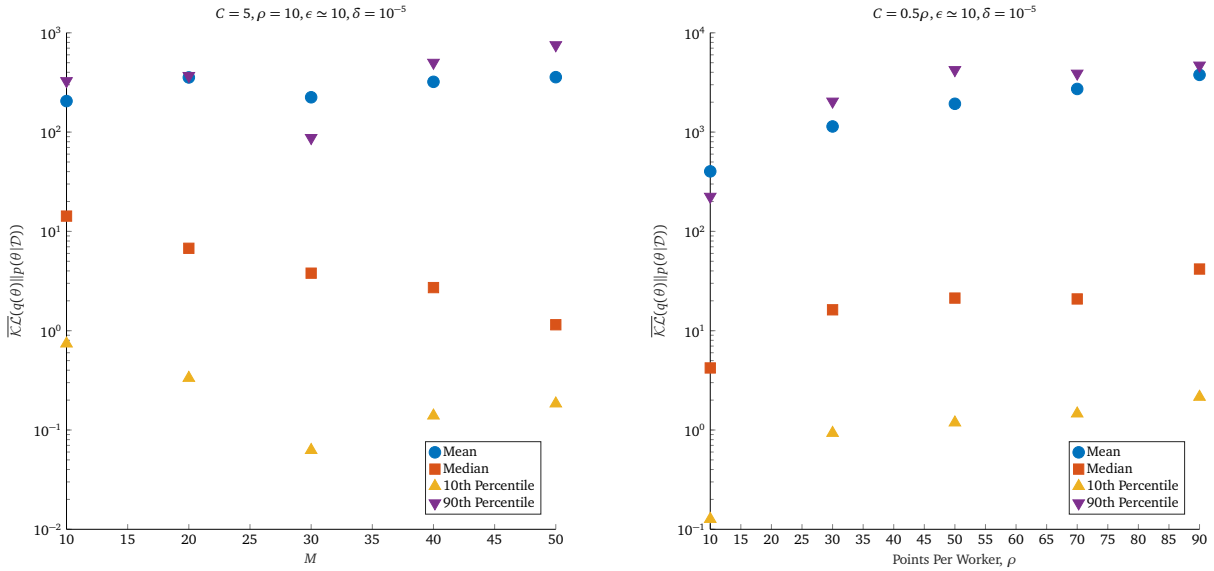


Fig. 4.13 Resulted obtained when varying ρ and M while fixing $\alpha = 0.1$ and $C = 0.5\rho$, in order to investigate how appropriate these parameter settings are across different datasets. 50 random seeds, each corresponding to a value of θ sampled from $\mathcal{N}(0, 5)$ and σ_e sampled from $\mathcal{U}(0.5, 2)$ and the corrupting noise sequence. In the left plot, $\rho = 10$ is fixed whilst M varies. The right plot corresponds to $M = 20$ being fixed whilst ρ varies. $\mathcal{KL}$ divergences are averaged across the final ten iterations for each random seed, and the mean, median and values of chosen percentiles (across the random seeds) are plotted.

to increase somewhat with ρ, perhaps due to the scaling of the clipping bound not being appropriate. The median performance achieved does not diminish significantly. It is also worth noting that a given value of ϵ corresponds to a stronger privacy protection for larger values of ρ, as now the worst case privacy loss must be bounded when a larger number of data-points change. However, it must also be noted that it is unclear whether the scaling of the clipping bound employed is optimal, and how much performance could be improved by tuning the function relating ρ to C.

Many of the properties that the dataset level DP-PVI algorithm exhibits are a direct consequence of the parametrisation employed. The magnitude of the natural parameters expected (i.e. of the non-private solution) grows as the number of the data-points increases, which can result in difficulties in choosing a clipping bound which is appropriate across a number of situations. The linear scaling suggested gives reasonable performance, but it unclear how close this scheme is to the optimum. Additionally, in certain situations, the magnitudes of each of the natural parameters can be very different (e.g. for very large values of θ). This is problematic as the Gaussian mechanism applies isotropic noise, meaning that different parameters have very different signal-to-noise ratios. We remark however freedom in choosing the parametrisation could enable less noisy estimates for certain parameters of the variational distribution, and the machine learning practitioner would be able to design the parametrisation with this in mind. Finally, since the parametrisation used has a strictly

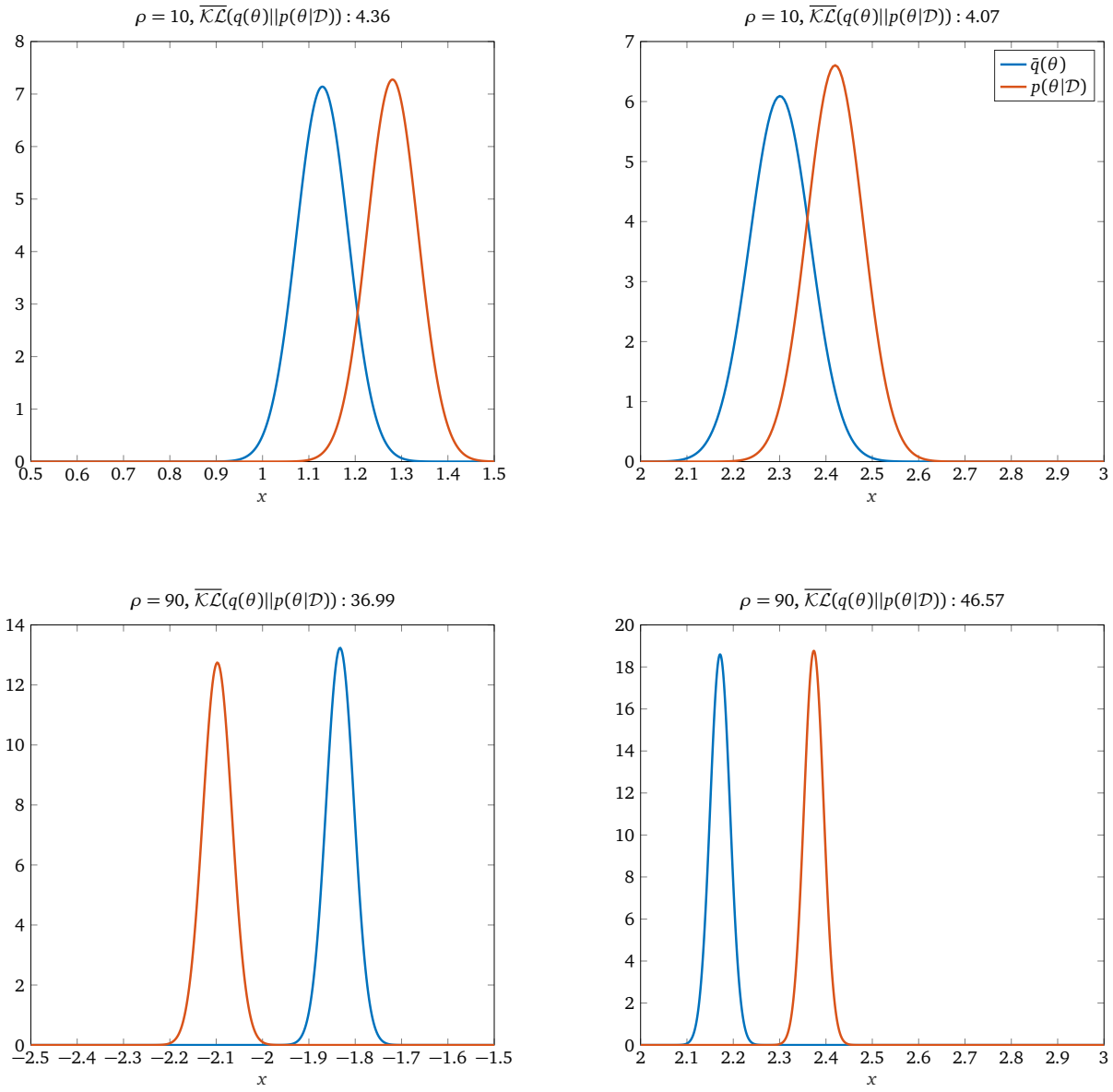


Fig. 4.14 Typical posteriors for different values of ρ obtained the dataset level DP-PVI algorithm. Produced with $C = 0.5\rho$, $\alpha = 0.1$, $M = 20$ workers. $\mathcal{KL}$ divergences reported are averaged across the final ten iterations, and the mean values of η_1 and η_2 are used to plot the approximate posteriors shown.

positive parameter (the precision), bias is introduced which could be avoided by using a parameter which must not remain positive.

4.3.3 Validity of Local Clipping

The dataset level DP-PVI scheme distributes the central Gaussian noise applied by the Gaussian mechanism in order to enable each client to keep track of their approximate likelihood term accurately. However, in the implementation used, updates to the precision are clipped locally at each client to ensure that the precision relating to the client approximate approximate likelihood term remains positive. We refer to this clipping as *precision clipping* to distinguish it from the normal ℓ_2 clipping applied. The precision clipping means that the central update for the precision takes the following form:

$$\tilde{\Delta}\eta_1 = \sum_{m=1}^{M} \max\left\{ -\eta_{m,1} \ , \ \alpha \cdot \left[\frac{\Delta\eta_{m,1}}{\max(1, ||\Delta\lambda_m||_2 / C)} + \frac{\sigma C}{\sqrt{M}} z_m \right] \right\}$$
$$\text{with } z_m \overset{iid}{\sim} \mathcal{N}(0,1) \tag{4.28}$$

which due to the additional precision clipping step is not equivalent to the central application of the Gaussian mechanism with variance σC. A consequence of this is that the claimed privacy bounds for results of the previous section are technically incorrect. However, we suggest that this could be remedied to yield very similar performance and valid privacy calculations, provided that the clipping bound has been chosen appropriately. We now quantify the probability of precision clipping being applied at the start of the dataset level DP-PVI algorithm and at convergence.

Due to the properties of the parametrisation employed, the change of the precision of the variational distribution due to client m is equal to the change of the precision of the local approximate likelihood. Recalling Eq. (4.9):

$$\Delta\eta_{m,1} = \frac{\boldsymbol{x}_m^T \boldsymbol{x}_m}{\sigma_e^2} - \eta_{m,1} \tag{4.29}$$

In the absence of noise, we expect convergence when this is zero. Recalling that there are ρ data-points per client and that $\mathbb{E}(x_i^2) = 1$ (due to Eq. (4.18)), it is expected that the value of the local precision at convergence is:

$$\eta_{m,1}^* \simeq \frac{\rho}{\sigma_e^2} \tag{4.30}$$

The PVI scheme initialises $q(\theta)$ at the prior, corresponding to $\eta_{m,1}^{(0)} = 0 \ \forall m$. The probability that precision clipping occurs at initialisation is therefore equivalent to the probability that the initial update is negative. Assuming that $\Delta\eta_2 = 0$ and that $\eta_{m,1} > C$, the initial

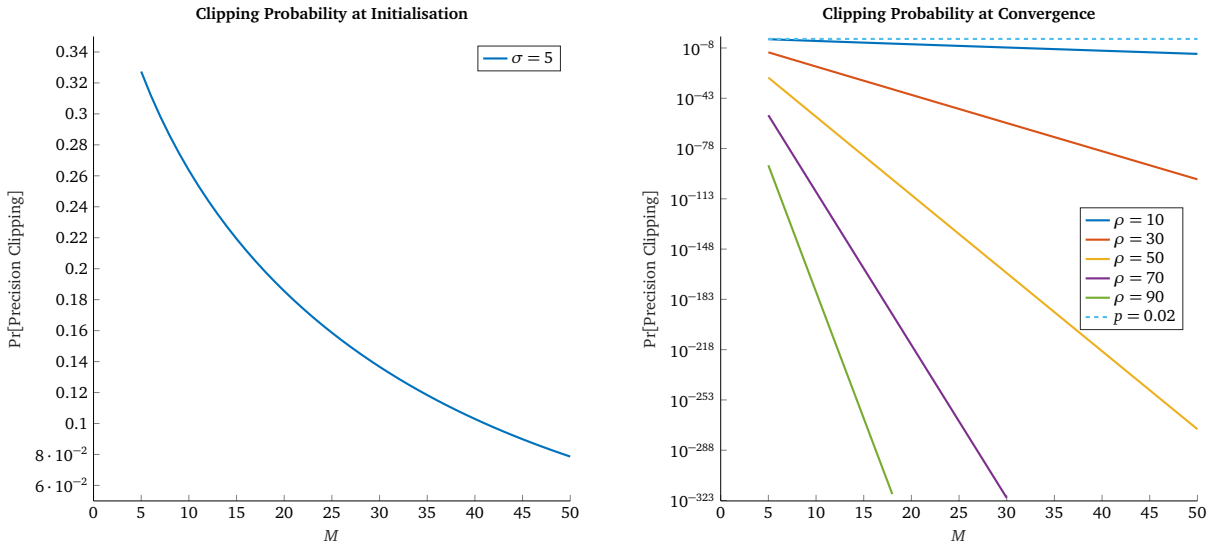


Fig. 4.15 Probability of local precision clipping occurring at the start of the dataset level DP-PVI algorithm (left sub-plot) and when close to convergence (right sub-plot) for different hyper-parameter settings. Parameters used in experiments are consistent with those elsewhere in the report. Left sub-plot: $\sigma = 5$. Right sub-plot: $C = 5, \sigma = 5, \alpha = 0.1$. $\sigma_e = 2$, which is the worst case of values considered here.

update is rescaled to C. The update random variable before precision clipping therefore is distributed according to $\mathcal{N}(\alpha C, \alpha^2\sigma^2 C^2/M)$. Therefore, the probability of precision clipping at initialisation is given by:

$$\Pr(\text{Precision Clipping at Initialisation}) \simeq \Phi\Big(\frac{-\sqrt{M}}{\sigma}\Big) \tag{4.31}$$

At convergence, assume that $\eta_{m,1} \simeq \eta^{(*)}_{m,1}$. Therefore, $\Delta\eta_{m,1} \simeq 0$ and the update random variable is approximately distributed according to $\mathcal{N}(0, \sigma^2\alpha^2 C^2/M)$. Therefore, the probability of precision clipping at convergence is:

$$\Pr(\text{Precision Clipping at Convergence}) \simeq \Phi\Big(\frac{-\rho\sqrt{M}}{\sigma_e^2\alpha\sigma C}\Big) \tag{4.32}$$

Fig. 4.15 plots the probability of additional clipping occurring in these cases due to the local precision becoming negative for the typical parameter settings used in this report.

We remark that whilst this technically means that the privacy bounds claimed by the experiments here are incorrect, it is likely that similar performance can be obtained when correcting for this issue. Simply removing the precision clipping step and would yield an algorithm with correct privacy guarantees, and the parameter server could for instance reject changes which result in a negative precision globally and roll-back the approximate likelihoods. Since the precision clipping occurs rarely at convergence (and only moderately often at initialisation), this would not result in a significantly increased privacy cost. We also note that several clients would have to provide a negative precision to cause the precision

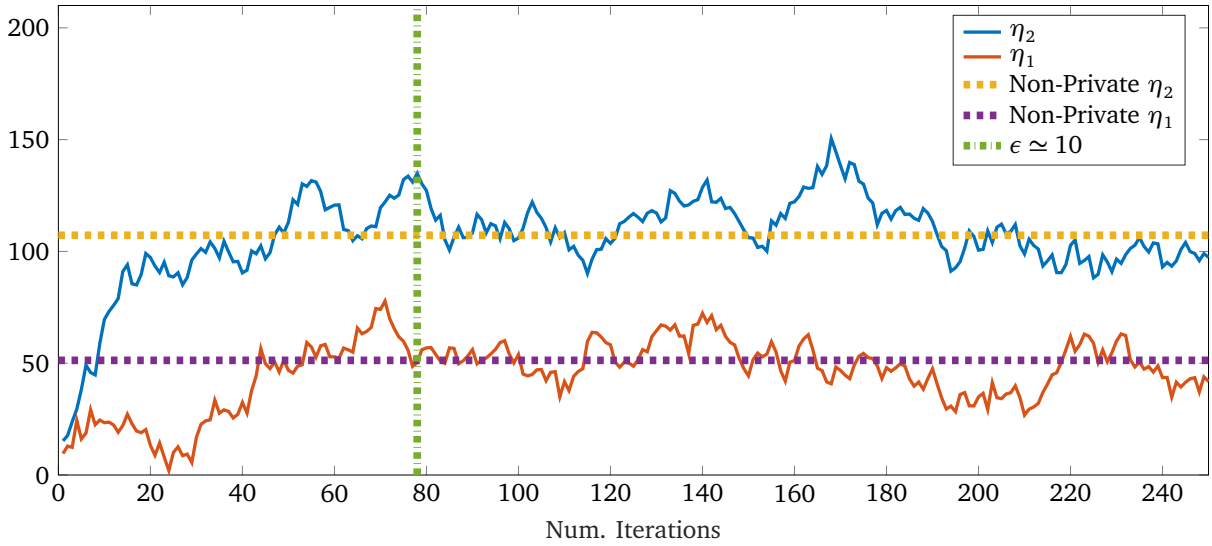


Fig. 4.16 Example dataset level DP-PVI run with local precision clipping removed. Produced with $M = 20$ workers, $\rho = 10$ points per worker, $C = 5$, $\alpha = 0.1$ and $\sigma = 5$, which are the suggested settings used for experiments. $\theta = 2$ and $\sigma_e = 2$. Additional iterations are run, vertical dashed line shows the number of iterations when $\epsilon \simeq 10$. Horizontal dashed lines refer to non-private parameter values.

at the parameter server to become negative, which is even less likely. However, it must be noted that it is essential to not choose the clipping bound value too large. This may yield invalid local approximate likelihood factors (but not variational distributions, as invalid distributions would be rejected) for the first few iterations of the dataset level DP-PVI algorithm but, on average, the parameters of the variational distribution would still move towards their optimal values. Indeed, Fig. 4.16 shows an example run with the local precision clipping removed and the central parameter server rejecting changes which result in negative precision, which, by the post-processing property of differential privacy, does not invalidate the privacy calculations performed. Note that even though the precision clipping is removed, the precision of the approximate posterior does not become negative in this example. The algorithm performs well, reaching close to the non-private solution within the privacy budget. Extending the number of iterations past the consumption of the privacy budget, it would also appear that the estimate for the precision is unbiased, which is desirable.

There are many other methods to correct for this issue. For instance, the parametrisation could be altered to use the log of the variance. Another proposal is to generate noise at the server and create an additional likelihood term at the server which would track the noise. Alternatively, noise generation could still occur at each client and the server could recalculate the approximate likelihood terms to incorporate the overall noise applied and then communicate these terms back to each client.

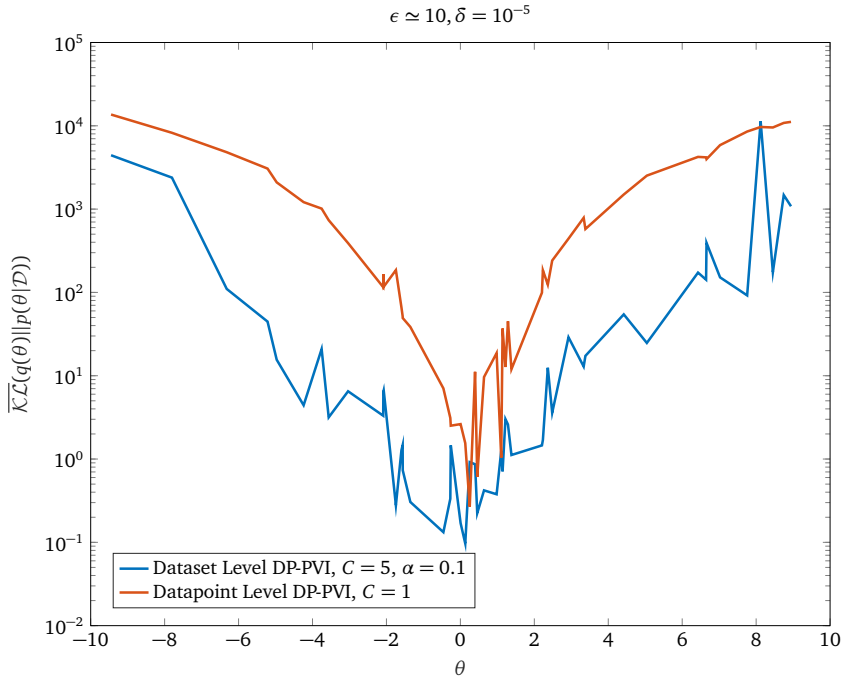


Fig. 4.17 Comparison of performance between appropriate settings for both the dataset and data-point level DP-PVI schemes. Produced with $C = 0.5\rho = 5$ for the dataset level protection and $C = 1$ for the data-point level protection. $\rho = 10$ points per worker, $M = 20$ workers with $\alpha = 0.1$. $\mathcal{KL}$ values reported are averaged across the final ten iterations. The DP noise level is fixed at $\sigma = 5$.

4.4 Comparison between Dataset and Datapoint DP-PVI

Fig. 4.17 compares the performance achieved by the dataset and data-point level DP-PVI algorithms for appropriate parameter settings using analytical updates.

It appears that the dataset level approach is more promising and it is able to achieve significantly smaller $\mathcal{KL}$ divergences. Additionally, as remarked in the previous section, it actually provides stronger privacy guarantees. It is curious that the algorithm providing stronger privacy guarantees also achieves a better performance. It is suggested that this occurs as this scheme allows for analytical updates to be performed without bias for small clipping bounds, but the analytical updates for the data-point level scheme introduce bias if the clipping bound is chosen too small. This approach is however not without drawbacks as significantly more assumptions are placed in terms of the trustworthiness of external parties. Additionally, this approach is less suitable for inhomogeneous data. See Section 3.2.3 for a more detailed discussion of the merits of these schemes.

Chapter 5

Conclusions

The existing PVI algorithm was adapted in two different ways in order to create the data-point level and dataset level DP-PVI algorithms, which each have their own advantages and disadvantages. Data-point level protection makes fewer assumptions about the trustworthiness of external parties and enables each client to choose their own level of protection whilst the dataset level protection is the natural scheme to apply when each dataset corresponds to the data of a single user, but is less appropriate when the underlying data is inhomogeneous.

These techniques were applied to a linear regression model with one parameter. With regards to data-point level protection, whilst DP-SGD was found to give close to non-private performance, the privacy guarantee achieved was poor. Clipped analytical updates were found to provide stronger privacy guarantees at the cost of bias in their variational distributions, and the choice of clipping bound is challenging in situations where we are unable to *a priori* place bounds on data variables. A hybrid scheme, initially applying the clipped analytical updated followed by DP-SGD is likely give performance close to that achieved by DP-SGD at a significantly lower privacy cost.

Dataset level DP-PVI was applied using exact analytical updates, and experiments were performed to identify a suitable value for the clipping bound. A small algorithmic modification is required to ensure that the privacy guarantees advertised by this scheme are legitimate. Unlike the data-point level protection scheme, if this bound is chosen too small no bias is introduced; only the number of iterations required to give convergence is affected. This is a very useful property. An appropriate choice of clipping bound and learning rate was identified for this scheme and these parameters were applied whilst varying the number of workers and points per worker, scaling the clipping bound with the number of points per worker. It was found that these settings were relatively robust to changes in these parameters, but they did tend to affect performance, and it is unclear the magnitude of gains which could be made by further tuning the algorithm hyper-parameters. Additionally, it was found that the choice of parametrisation is very important, and the properties of the parametrisation used in general will affect how robust a given set of parameter settings are to changes in the

context in which the algorithm is applied; for the case study employed, the restriction of one of the parameters being positive introduced bias.

For the case study considered, we found that the dataset level DP-PVI algorithm was not only able to provide a stronger privacy guarantee than that of the data-point level DP-PVI algorithm (the same (ϵ, δ) was achieved for both schemes) but also gave much better performance.

Finally, we conclude in stating that the dataset level DP-PVI algorithm shows promise, and is able to provide excellent results for certain datasets and parameter settings. A small modification in the implementation will however be required to ensure correct privacy accounting, but it is suggested that this modification should not confer a large additional privacy cost. In general, there are a very large number of hyper-parameters which must be tuned, and unlike in the standard machine learning context, performing a search using the data which we wish to train on consumes the privacy budget itself! A machine learning practitioner wishing to apply this technique must take care in the parametrisation employed for the probabilistic model which is being trained and the value of clipping bound set, whilst the learning rate and the DP noise scale can be tuned more easily.

5.1 Future Work

The application of differential privacy techniques to machine learning models is recent work, and therefore there remain a number of unanswered questions.

Performance for a chosen (ϵ, δ) privacy bound depends significantly on algorithm hyper-parameters, but it is not immediately clear how to choose these values or how changes in these values affect the optimal hyper-parameter settings, which is relevant as in practice, a designer would with choose hyper-parameter settings with a privacy guarantee in mind. In certain cases, we are unable to *a priori* bound values meaning that appropriate scales for the clipping bound are difficult to obtain, and indeed performing investigations to determine these values will consume the fixed privacy budget. It is strongly suggested that a differentially private data-whitening scheme be applied to the data to mitigate this problem, but the privacy cost of applying this scheme must be accounted for.

Additionally, it is unclear how, in general, the choice of hyper-parameters generalises across several models and several data-sets. This will be influenced largely by the choice of parametrisation; for instance, in the case study employed, the value of θ has a large influence on the magnitude of the parameters, suggesting that it would also affect the optimal parameter settings. If it were possible to find parameter settings which were very robust across a wide range of datasets, fake data with similar underlying statistics or data for which privacy is not required could be used to find an appropriate set of hyper-parameters. It is likely that applying a differentially private data-whitening mechanism would improve the generalisation of model hyper-parameters. These settings could then be used to protect the

sensitive dataset. Furthermore, a meta-analysis could be performed, for example, training a machine learning model to predict appropriate hyper-parameter settings from certain dataset characteristics.

Furthermore, we note that schemes which adaptively modify the clipping bound, learning rate and noise scale may yield improvements and reduce the importance of *a priori* choosing appropriate hyper-parameters. For the dataset DP-PVI algorithm, it is expected that updates early in the course of training are mostly moving the model towards the optimum but once the optimum is reached, the parameters of the variational distribution oscillate. Therefore, decreasing the learning rate as the number of iterations increases seems appropriate. For the DP-SGD approach, adaptive learning rate techniques which already exist could be applied directly, modifying them to be differentially private if required. It is not clear how an adaptive clipping bound scheme would work; as the results of the the analytical clipped updates for the data-point level DP-PVI scheme show, the clipping can fundamentally change the solutions obtained.

An alternative way to perform DP-PVI privately would be to construct a differentially private estimate of the local likelihood function, which would only need to be constructed when a client gathers new data. This likelihood function could then be fixed and used repeatedly in combination with exact analytical updates to yield a differentially private algorithm.

Furthermore, it is likely the applying the dataset level DP-PVI algorithm on inhomogeneous data will yield poor results since the magnitude of parameter updates may be very different. This needs to be investigated further.

References

Teppo Niinimäki, Mikko Heikkilä, Antti Honkela, and Samuel Kaski. Representation transfer for differentially private drug sensitivity prediction. *arXiv preprint arXiv:1901.10227*, 2019.

A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, May 2008. doi: 10.1109/SP.2008.33.

Melissa Gymrek, Amy L McGuire, David Golan, Eran Halperin, and Yaniv Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013.

Kirk Bansak, Jeremy Ferwerda, Jens Hainmueller, Andrea Dillon, Dominik Hangartner, Duncan Lawrence, and Jeremy Weinstein. Improving refugee integration through data-driven algorithmic assignment. *Science*, 359(6373):325–329, 2018.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 1322–1333, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3832-5. doi: 10.1145/2810103.2813677. URL http://doi.acm.org/10.1145/2810103.2813677.

Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data, Apr 2017. URL https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 978-0387-31073-2. URL http://research.microsoft.com/en-us/um/people/cmbishop/prml/.

Mikko Heikkilä, Eemil Lagerspetz, Samuel Kaski, Kana Shimizu, Sasu Tarkoma, and Antti Honkela. Differentially private bayesian learning on distributed data. In *Advances in neural information processing systems*, pages 3226–3235, 2017.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL http://dx.doi.org/10.1561/0400000042.

Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C Pierce, and Aaron Roth. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pages 398–410. IEEE, 2014.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 308–318,

New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4139-4. doi: 10.1145/2976749.2978318. URL http://doi.acm.org/10.1145/2976749.2978318.

J. Jälkö, O. Dikmen, and A. Honkela. Differentially Private Variational Inference for Non-conjugate Models. *ArXiv e-prints*, October 2016.

Thang D Bui, Cuong V Nguyen, Siddharth Swaroop, and Richard E Turner. Partitioned variational inference: A unified framework encompassing federated and continual learning. *arXiv preprint arXiv:1811.11206*, 2018.

Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

Appendix A

Appendices

A.1 Risk Assessment Retrospective

In risk assessment submitted it October, it was noted that the risks related to this project would be due to extensive computer usage. This risk assessment proved to be accurate and the appropriate steps were taken to mitigate these risks, including ensuring appropriate working conditions, suitable posture and a sensible working pattern.

A.2 Electronic Resources

Please note that all code listings for this project may be found on `GitHub` at this address: https://github.com/MrinankSharma/dp-pvi-project.

Additionally, the electronic log book used for this project can be found at the following address: https://drive.google.com/file/d/1PafemIYA0pPCeIDv2K_utI8CYklXyPmM/view?usp=sharing.