View metadata, citation and similar papers at core.ac.uk

brought to you by CO

provided by Helsingin yliopiston digitaalinen a

UNIVERSITY OF HELSINKI

Faculty of Science

Department of Mathematics and Statistics

# Privacy-aware variational inference

**Joonas Jälkö**

MSc Thesis

**October 6, 2017**

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

| Tiedekunta/Osasto — Fakultet/Sektion — Faculty | Laitos — Institution — Department |
|---|---|
| Faculty of Science | Department of Mathematics and Statistics |

Tekijä — Författare — Author
Joonas Jälkö

Työn nimi — Arbetets titel — Title

Privacy-aware variational inference

Oppiaine — Läroämne — Subject
Mathematics

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
|---|---|---|
| Master's Thesis | September 2017 | 50 s. |

Tiivistelmä — Referat — Abstract

This thesis focuses on privacy-preserving statistical inference. We use a probabilistic point of view of privacy called *differential privacy*. Differential privacy ensures that replacing one individual from the dataset with another individual does not affect the results drastically. There are different versions of the differential privacy. This thesis considers the $\epsilon$-differential privacy also known as the pure differential privacy, and also a relaxation known as the $(\epsilon, \delta)$-differential privacy.

We state several important definitions and theorems of DP. The proofs for most of the theorems are given in this thesis. Our goal is to build a general framework for privacy preserving posterior inference. To achieve this we use an approximative approach for posterior inference called *variational Bayesian* (VB) methods. We build the basic concepts of variational inference with certain detail and show examples on how to apply variational inference.

After giving the prerequisites on both DP and VB we state our main result, the *differentially private variational inference* (DPVI) method. We use a recently proposed *doubly stochastic variational inference* (DSVI) combined with *Gaussian mechanism* to build a privacy-preserving method for posterior inference. We give the algorithm definition and explain its parameters.

The DPVI method is compared against the state-of-the-art method for DP posterior inference called the *differentially private stochastic gradient Langevin dynamics* (DP-SGLD). We compare the performance on two different models, the logistic regression model and the Gaussian mixture model. The DPVI method outperforms DP-SGLD in both tasks.

Avainsanat — Nyckelord — Keywords
Differential privacy, Variational Bayesian methods, Machine learning, Bayesian inference

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpulan tiedekirjasto

Muita tietoja — Övriga uppgifter — Additional information

# Acknowledgements

This thesis was written while working as a research assistant at the Helsinki Institute for Information Technology HIIT. I would like to thank both my supervisors Dr. Onur Dikmen and asst. prof. Antti Honkela for introducing me to the interesting topics of differential privacy and variational Bayesian methods and for giving me great ideas and guidance through the production of this thesis. I want to thank asst. prof. Antti Honkela especially for his great help on the implementation side of this project and Dr. Onur Dikmen for all the interesting conversations concerning differential privacy.

# Contents

# Chapter 1

# Introduction

Statistical methods are nowadays applied in many fields of science such as biology (Vittinghoff et al., 2011), physics (Baldi et al., 2014) and behavioural sciences (Vogelstein et al., 2014) to name a few and also in corporate world, for example in finance (Ticknor, 2013). With increasing computational resources we are able to solve difficult tasks in statistical inference. Problems such as regression, classification and clustering in which we want to learn to predict our model or learn some underlying structure of the data have gained a lot of attention. These kind of methods that apply statistical inference into learning tasks using computational resources are called machine learning in general.

Using more data usually leads to better generalisation and accuracy in machine learning. Consider a query where students of a class are asked to release their weight in order to compute the average weight of students of that age. If we have a class of 10 students we probably will get a worse estimate for average weight of children of that age than from a class of size 20. However by increasing the number of test subjects we also compromise the privacy of more students. If all but one of the students are willing to release their weight publicly it is obvious that releasing the average weight will also reveal the weight of the one student that did not want his or her weight to be released. This toy example might not look that worrying, but in more general learning tasks these kind of privacy breaches might lead to more alarming outcomes. Therefore it is important to provide privacy guarantees for test subjects.

From the above example, it is obvious that a simple anonymization scheme of removing the names of individuals in the study is not enough to ensure the privacy. There have been many different definitions of anonymity during years e.g., *k-anonymity* (Samarati and Sweeney, 1998) and *$\ell$-diversity* (Machanavajjhala et al., 2007) to name a couple. Differential privacy (DP) (Dwork and Roth, 2014) gives a mathematical definition of privacy. It has many nice properties, one of which is that it is immune to any side information of test subjects. Differential privacy is based on a probabilistic view of anonymity. It ensures

that replacing one individual from the dataset with another individual does not affect the results drastically. This can be accomplished through adding stochasticity at different levels of the estimation process, such as adding noise to data itself (input perturbation, Dwork, 2006), changing the objective function to be optimised or how it is optimised (objective perturbation, Chaudhuri et al., 2011), releasing the estimates after adding noise (output perturbation, Dwork, 2006) or by sampling from a distribution based on utility or goodness of the alternatives (exponential mechanism, McSherry and Talwar, 2007).

In recent years differential privacy has gained a lot of attention. It has been applied to many of standard machine learning approaches, such as objective-perturbation-based logistic regression (Chaudhuri and Monteleoni, 2008), regression using functional mechanism (Zhang et al., 2012) to name a few. However privacy-preserving Bayesian inference (e.g. , Williams and McSherry, 2010; Zhang et al., 2014) has only recently started attracting more interest. Dimitrakakis et al. (2014) showed that under certain assumptions the posterior distribution is differentially private. Although the result is mathematically elegant it lacks a certain generality. Methods based on this approach suffer from the major weakness that the privacy guarantees are only valid for samples drawn from the exact posterior which is usually impossible to guarantee in practice. Recent methods by Zhang et al. (2016), Foulds et al. (2016) and Honkela et al. (2016) are based on perturbing the sufficient statistics. These methods provide good accuracy, but are limited to the models that come from exponential family of distributions. The sufficient statistic perturbation approach was recently also applied to variational inference (Park et al., 2016), which is again applicable to models where non-private inference can be performed by accessing the sufficient statistics.

Wang et al. (2015) propose a simple gradient based method for posterior sampling. This method, called DP-SGLD, achieves differential privacy by gradient perturbation with stochastic gradient Markov chain Monte Carlo (MCMC) sampling. Privacy guarantees are achieved automatically when the log-probability of the model is Lipschitz continuous. This approach works in principle for arbitrary models, but because of the gradient perturbation mechanism, each MCMC iteration will consume some privacy budget, hence limiting the number of iterations which may lead to stopping before convergence.

The goal of this thesis is to provide a general framework for inferring an approximation of the posterior distribution in a differentially private manner. We achive this by applying two recent ideas from differential privacy and from variational Bayesian methods. Variational inference seems preferable to stochastic gradient MCMC here because a good optimiser should be able to make better use of the limited gradient evaluations and the variational approximation provides a very efficient approximation of the posterior distribution. The recently proposed doubly stochastic variational inference (Titsias and Lázaro-Gredilla, 2014) and the further streamlined automatic differentiation variational inference (ADVI) method (Kucukelbir et al., 2017) provide a generic variational infer-

ence method also applicable to non-conjugate models. These approaches apply a series of transformations and approximations so that the variational distributions are Gaussian and can be optimised by stochastic-gradient-based methods. Here, we propose differentially private variational inference (DPVI) based on gradient clipping and perturbation as well as double stochasticity.

We start this thesis by introducing the basic definitions and techniques in differential privacy. After that we move into the variational inference and explain the most crucial concepts of it. We then use the results of Chapters 2 and 3 to formulate the DPVI. In Chapter 5 we make a thorough case study on the Bayesian logistic regression model with comparisons to the non-private case under different design decisions for DPVI. We also test the performance of DPVI on a Gaussian mixture model.

# Chapter 2

# Differential privacy

In order to provide privacy, we first need to define what do we mean by privacy and how can we measure it. We can see privacy as anonymity of an individual. Giving privacy guarantees to subject of a study means that we need to assure them that the data they are providing for the study cannot be traced back from the results of the study. Simple anonymization schemes such as masking the names of our subjects will fail (see e.g., Narayanan and Shmatikov, 2008) because a possible adversary can combine some prior information to trace back these masked attributes. In this thesis we use the definition presented by Dwork (2006) called *differential privacy*, that gives a probabilistic point of view on anonymity. Rather than masking some attributes of our dataset we ask, how can we make individuals' contribution to learning results indistinguishable from each other. Differential privacy is a strong privacy guarantee. We start with the definition of $\epsilon$-*differential privacy* and then move to a relaxation called $(\epsilon, \delta)$-*differential privacy*.

## 2.1 $\epsilon$-differential privacy

Before we give the first definition of privacy, we define *adjacency* and *randomized algorithm*.

**Definition 2.1.** We call two datasets $x, x'$ adjacent if they differ at most in one entry i.e.

$$\max(|x \setminus x'|, |x' \setminus x|) = 1.$$

We denote adjacency between $x$ and $x'$ with $x \sim x'$.

Our definition of adjacency is flexible in sense that number of entries in $x$ and $x'$ need not to be the same.

**Definition 2.2.** We call an algorithm $\mathcal{A} : \mathcal{D} \to \mathbb{R}^k$ randomized if $\mathcal{A}$ is a random variable with probability triple $(\mathbb{R}^k, \mathcal{B}(\mathbb{R}^k), \mu)$ where $\mathcal{B}(\mathbb{R}^k)$ denotes the Borel $\sigma$-algebra of $\mathbb{R}^k$ and $\mu$ is a probability measure that is parametrized by the input of $\mathcal{A}$.

**Definition 2.3.** A randomized algorithm $\mathcal{A}$ is $\epsilon$-differentially private if for all pairs of adjacent datasets $x, x'$, and for every $S \subset \mathrm{im}(\mathcal{A})$

$$\frac{\Pr(\mathcal{A}(x) \in S)}{\Pr(\mathcal{A}(x') \in S)} \leq e^{\epsilon}. \tag{2.4}$$

Definition 2.3 assures that changing any element of dataset affects the probability of $\mathcal{A}$'s output at most by a factor $e^{\epsilon}$. We measure the privacy with parameter $\epsilon$ and smaller $\epsilon$ values provide more strict privacy guarantees.

Ratio on left hand side of (2.4) is an important quantity when considering differential privacy. We call its logarithm the *privacy loss*.

**Definition 2.5.** Let $\mathcal{A}$ be a randomized algorithm and $x, x'$ two adjacent databases, then privacy loss of $\mathcal{A}$ is

$$\mathcal{L}_{\mathcal{A}(x)||\mathcal{A}(x')} = \ln\left(\frac{\Pr(\mathcal{A}(x) = \xi)}{\Pr(\mathcal{A}(x') = \xi)}\right).$$

As we mentioned before, the definition of adjacency is flexible in terms of dataset size. This gives us two possible interpretations on $\epsilon$-differential privacy. These are called unbounded and bounded differential privacy when number of dataset entries differ by one and when the number of entries are the same, respectively. These two definitions are not necessarily equivalent. It is quite clear that if there exists a null element for query, bounded version implies the unbounded one, but it is possible that unbounded version is not, at least with same $\epsilon$, bounded. However we can get the following result.

**Theorem 2.6.** *Unbounded version of $\epsilon$-differential privacy implies bounded version of $2\epsilon$-differential privacy.*

*Proof.* Assume $x' = x \cup i$ and $x'' = x' \setminus k$, where $i$ and $k$ are some individuals data and $i \neq k$. Now $x$ and $x''$ have the same number of data entries and differ by one element. Assume that $\mathcal{A}$ is a randomized algorithm that provides unbounded $\epsilon$-differential privacy. Now for all $S \in \mathrm{im}(A)$ we get

$$\Pr(\mathcal{A}(x) \in S) \leq e^{\epsilon} \Pr(\mathcal{A}(x') \in S) \leq e^{\epsilon}(e^{\epsilon} \Pr(\mathcal{A}(x'') \in S)) = e^{2\epsilon} \Pr(\mathcal{A}(x'') \in S).$$

It is clear that this holds for any two different entries $i$ and $k$ and so $\mathcal{A}$ provides bounded $2\epsilon$-differential privacy. □

To get some insight on $\epsilon$-differential privacy let us have a look on following example (Dwork and Roth, 2014, page 30),

**Example 2.7.** Consider a study where respondents will give "yes" or "no" answer to some question of interest. To ensure the privacy and provide plausible deniability for respondents, we use the following algorithm $\mathcal{A}$ in the study

1. Flip a coin.
2. If tails, then respond truthfully.
3. If heads, flip again and respond "Yes" if heads and "No" otherwise.

Now we ask what kind of privacy this algorithm provides. Let $\mathbf{x}$ and $\mathbf{y}$ be the truthful answers to our study that differ only in one element, let us say in $i$th element. Let $\mathbf{z}$ be the output of algorithm $\mathcal{A}$. Clearly $p(\text{Output} = \text{"Yes"}|\text{Input} = \text{"Yes"}) = 1/2 + 1/4 = 3/4$, $p(\text{Output} = \text{"Yes"}|\text{Input} = \text{"No"}) = 1/4$ and similarly for "No" answers. Consider $z_i =$"Yes"

$$\frac{\Pr(\mathcal{A}(\mathbf{x}) = \mathbf{z})}{\Pr(\mathcal{A}(\mathbf{y}) = \mathbf{z})} = \frac{\Pr(\mathcal{A}(x_i) = z_i)}{\Pr(\mathcal{A}(y_i) = z_i)} = \begin{cases} \frac{p(\text{Output="Yes"}|\text{Input="Yes"})}{p(\text{Output="Yes"}|\text{Input="No"})} = 3 \\ \frac{p(\text{Output="Yes"}|\text{Input="No"})}{p(\text{Output="Yes"}|\text{Input="Yes"})} = \frac{1}{3} \end{cases}$$

$$\Rightarrow \frac{\Pr(\mathcal{A}(\mathbf{x}) = \mathbf{z})}{\Pr(\mathcal{A}(\mathbf{y}) = \mathbf{z})} \leq 3.$$

Above holds similarly for $z_i$="No", so we get that $\mathcal{A}$ is $\ln 3$-differentially private.

Next we define an important quantity called $\ell_1$-sensitivity.

**Definition 2.8.** The $\ell_1$-sensitivity $\Delta_1 f$ of a function $f : \mathcal{D} \to \mathbb{R}^k$ is defined as

$$\Delta_1 f = \sup_{x \sim x'} ||f(x) - f(x')||_1 = \sup_{x \sim x'} \sum_{i=1}^{k} |f(x)_i - f(x')_i|$$

Sensitivity plays a key role in privacy calculations, because it tells us how much our query results differ in the worst case on any two adjacent datasets.

Algorithm $\mathcal{A}$ in Example 2.7 is an illustrating example of a randomized algorithm that provides differential privacy. We could ask, what kinds of randomized algorithms provide differential privacy? Answer is simply all kinds of randomized algorithms that add uncertainty on the output. However for an arbitrary randomized algorithm, calculation of exact privacy budget, i.e., providing a bound on $\epsilon$, can be difficult. Next we introduce a probability distribution that we can use to achieve differential privacy.

**Definition 2.9.** Random variable $X$ is distributed according to Laplace distribution centered at 0 with scale parameter $b$ if it has density

$$p(x \mid b) = \frac{1}{2b} \exp\left\{-\frac{|x|}{b}\right\}.$$

With random variables drawn from Laplace distribution we can build an randomized algorithm. We call this method the *Laplace mechanism*.

**Definition 2.10.** Laplace mechanism for any function $f : \mathcal{D} \to \mathbb{R}^k$ is defined as

$$\mathcal{M}_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, \ldots, Y_k)$$

where $Y_i \sim \text{Lap}(\Delta f/\epsilon)$.

This mechanism gives us a way to provide $\epsilon$-differential privcy on a query with known sensitivity.

**Theorem 2.11.** *Laplace mechanism provides $\epsilon$-differential privacy.*

*Proof.* Assume $\mathbf{x}, \mathbf{x}' \in \mathcal{D}$, $\mathbf{x} \sim \mathbf{x}'$ and $f : \mathcal{D} \to \mathbb{R}^k$ with $\ell_1$-sensitivity $\Delta f$. Denote the probability density function of $\mathcal{M}(\mathbf{x}, f(\cdot), \epsilon)$ as $p_{\mathbf{x}}$ and density of adjacent pair with $p_{\mathbf{x}'}$. Now releasing some $z \in \mathbb{R}^k$ given $\mathbf{x}$ means that $(Y_1, \ldots, Y_k) = f(\mathbf{x}) - z$, where $Y_i \sim \text{Lap}(\Delta f/\epsilon)$, $\forall i$. So we get

$$\begin{aligned}
p_{\mathbf{x}}(z) &= \prod_{i=1}^{k} \frac{\epsilon}{2\Delta f} \exp\left(-\frac{\epsilon |f(\mathbf{x})_i - z_i|}{\Delta f}\right) \\
\Rightarrow \frac{p_{\mathbf{x}}(z)}{p_{\mathbf{x}'}(z)} &= \prod_{i=1}^{k} \frac{\exp\left(-\frac{\epsilon |f(\mathbf{x})_i - z_i|}{\Delta f}\right)}{\exp\left(-\frac{\epsilon |f(\mathbf{x}')_i - z_i|}{\Delta f}\right)} \\
&= \prod_{i=1}^{k} \exp\left(\frac{\epsilon}{\Delta f}(|f(\mathbf{x}')_i - z_i| - |f(\mathbf{x})_i - z_i|)\right) \\
&\leq \exp\left(\frac{\epsilon}{\Delta f} \sum_{i=1}^{k} |f(\mathbf{x}')_i - f(\mathbf{x})_i|\right) = \exp\left(\frac{\epsilon}{\Delta f} ||f(\mathbf{x}') - f(\mathbf{x})||_1\right) \\
&\leq \exp(\epsilon).
\end{aligned}$$

□

## 2.2 $(\epsilon, \delta)$-differential privacy

Sometimes we do not require as strict privacy guarantee as $\epsilon$-differential privacy provides. We give a useful relaxation called $(\epsilon, \delta)$-differential privacy.

**Definition 2.12.** A randomized algorithm $\mathcal{A}$ is $(\epsilon, \delta)$-differentially private if for all pairs of adjacent databases $x, x'$ and for every $S \subset \text{im}(\mathcal{A})$

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

The relaxation constant $\delta$ can be seen as a probability of a privacy breach. We can easily see, that small positive $\epsilon$ and $\delta$ values provide better privacy. It is immediate that every $\epsilon$-differentially private algorithm is also $(\epsilon, \delta)$-differentially private with any $\delta \in [0, 1)$[1]. Unfortunately this relation only works in one direction as we can show with simple example.

**Example 2.13.** Consider algorithm $\mathcal{A}$ that is $\ln(e^{\epsilon} + \delta)$-differentially private. Now $\mathcal{A}$ is clearly $(\epsilon, \delta)$-differentially private because

$$\Pr(\mathcal{A}(x) \in S) \leq \Pr(\mathcal{A}(x') \in S)(e^{\epsilon} + \delta) \leq e^{\epsilon} \Pr(\mathcal{A}(x') \in S) + \delta.$$

However since $\ln(e^{\epsilon} + \delta) > \epsilon$ for all $\delta > 0$, algorithm $\mathcal{A}$ is not $\epsilon$-differentially private.

In the next lemma we show how privacy loss and $(\epsilon, \delta)$-differential privacy are linked.

**Lemma 2.14.**

*(1) If the privacy loss of algorithm $\mathcal{A}$ is bounded by $\epsilon$ with probability at least $1 - \delta$, then $\mathcal{A}$ preserves $(\epsilon, \delta)$-differential privacy.*

*(2) If the algorithm $\mathcal{A}$ preserves $(\epsilon, \delta)$-differential privacy, then probability of the absolute value of privacy loss exceeding $2\epsilon$ is bounded by $\frac{2\delta}{\epsilon e^{\epsilon}}$.*

*Proof of (1).* Let us first assume that privacy loss is bounded by $\epsilon$ with probability greater than $1 - \delta$. Define a set $B$ as a region where $\epsilon$-differential privacy is breached

$$\mathcal{L}_{\mathcal{A}(x)||\mathcal{A}(x')}(B) > \epsilon.$$

By assumption $\Pr(\mathcal{A}(x) \in B) < \delta$. Now by law of total probability for all $S \in \text{im}(\mathcal{A})$ we have

$$\begin{aligned} \Pr(\mathcal{A}(x) \in S) &= \Pr(\mathcal{A}(x) \in B \cap S) + \Pr\big(\mathcal{A}(x) \in B^C \cap S\big) \\ &\leq \Pr(\mathcal{A}(x) \in B) + e^{\epsilon} \Pr\big(\mathcal{A}(x') \in B^C \cap S\big) \\ &\leq e^{\epsilon} \Pr(\mathcal{A}(x') \in S) + \delta \end{aligned}$$

thus $\mathcal{A}$ is $(\epsilon, \delta)$-differentially private. □

[1]The case $\delta = 0$ corresponds to the $\epsilon$-differential privacy.

*Proof of (2).* Assume now that $\mathcal{A}$ preserves $(\epsilon, \delta)$-differential privacy. Consider a set $B'$ defined as

$$B' = \left\{ o : e^{2\epsilon} \Pr(\mathcal{A}(x') = o) < \Pr(\mathcal{A}(x) = o) \right\}.$$

Now we get

$$\Pr(\mathcal{A}(x) \in B') > e^{2\epsilon} \Pr(\mathcal{A}(x') \in B') \geq e^{\epsilon}(\epsilon + 1) \Pr(\mathcal{A}(x') \in B').$$

Using above and Definition 2.12 we get

$$\begin{aligned} &\delta \geq \Pr(\mathcal{A}(x) \in B') - e^{\epsilon} \Pr(\mathcal{A}(x') \in B') \geq e^{\epsilon}\epsilon \Pr(\mathcal{A}(x') \in B') \\ &\Rightarrow \Pr(\mathcal{A}(x') \in B') \leq \frac{\delta}{e^{\epsilon}\epsilon}. \end{aligned}$$

Similar result holds also for set

$$B'' = \left\{ o : e^{-2\epsilon} \Pr(\mathcal{A}(x') = o) > \Pr(\mathcal{A}(x) = o) \right\}$$

so now we can easily see that the claim holds. □

As for $\epsilon$-differential privacy, also for $(\epsilon, \delta)$-differential privacy it holds that unbounded definition implies the bounded definition with small loss in parameters. Following the same kind of reasoning we get the following result.

**Theorem 2.15.** *Unbounded $(\epsilon, \delta)$-differential privacy implies bounded $(2\epsilon, \delta(e^{\epsilon} + 1))$-differential privacy. The factor $e^{\epsilon} + 1 \approx 2$ with small epsilon values.*

*Proof.* Assume algorithm $\mathcal{A}$ preserves unbounded $(\epsilon, \delta)$-differential privacy. As in the proof of Theorem 2.6 we assume $x' = x \cup i$ and $x'' = x' \setminus k$, where $i$ and $k$ are some individuals data and $i \neq k$. Again applying privacy bound twice we get

$$\begin{aligned} \Pr(\mathcal{A}(x) \in S) &\leq e^{\epsilon} \Pr(\mathcal{A}(x') \in S) + \delta \leq e^{\epsilon}(e^{\epsilon} \Pr(\mathcal{A}(x'') \in S) + \delta) + \delta \\ &= e^{2\epsilon} \Pr(\mathcal{A}(x'') \in S) + \delta(1 + e^{\epsilon}). \end{aligned}$$

□

Similar to the $\ell_1$-sensitivity (Definition 2.8) we define a quantity called $\ell_2$-sensitivity.

**Definition 2.16.** The $\ell_2$-sensitivity of a function $f : \mathcal{D} \to \mathbb{R}^k$, is defined as

$$\Delta_2 f = \max_{x \sim x'} ||f(x) - f(x')||_2.$$

Now we can construct a mechanism, that is based on adding zero mean Gaussian noise to the query with known $\ell_2$-sensitivity, which provides $(\epsilon, \delta)$-differential privacy.

**Definition 2.17** (Gaussian Mechanism)**.** Given a query $f : \mathcal{D} \to \mathbb{R}^k$ with $\ell_2$-sensitivity $\Delta_2 f$ and a privacy budget $(\epsilon, \delta)$ where $\epsilon \in (0, 1)$, $\delta < 0.5$ and $\epsilon > \delta$, we define the Gaussian mechanism as

$$\mathcal{M}_G(x, f(\cdot), \epsilon, \delta) = f(x) + (Y_1, \ldots, Y_k),$$

where $Y_1 \sim N(0, \sigma^2)$ with $\sigma > \sqrt{2\ln(1.25/\delta)}\Delta f/\epsilon$.

**Lemma 2.18.** *For Gaussian random variable $X$ with zero mean and variance $\sigma^2$*

$$\Pr(X > y) \leq \frac{\sigma}{y\sqrt{2\pi}} \exp\bigl(-y^2/2\sigma^2\bigr)$$

*holds that for all $y > 0$.*

*Proof.* Denote the probability density function of standard normal distribution with $\phi(x)$

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\bigl(-x^2/2\bigr).$$

We clearly see that $\phi'(x) = -x\phi(x)$. Assuming $y > 0$, integration by parts yields

$$\begin{aligned}
\Pr(x > y) &= \int_y^\infty \phi(x)\mathrm{d}x = -\int_y^\infty \frac{\phi'(x)}{x}\mathrm{d}x \\
&= -\left[\frac{\phi(x)}{x}\right]_y^\infty - \int_y^\infty \underbrace{\frac{\phi(x)}{x^2}}_{\geq 0}\mathrm{d}x \\
&\leq \frac{\phi(y)}{y}.
\end{aligned}$$

Now if $X \sim N(0, \sigma)$ we know that $X/\sigma \sim N(0, 1)$ and therefore

$$\begin{aligned}
\Pr_{x\sim X}(x > y) &= \Pr_{x\sim X/\sigma}\left(x > \frac{y}{\sigma}\right) \\
&\leq \frac{\phi(y/\sigma)}{(y/\sigma)} \\
&= \frac{\sigma}{y\sqrt{2\pi}} \exp\bigl(-y^2/2\sigma^2\bigr).
\end{aligned}$$

□

**Theorem 2.19.** *Gaussian mechanism provides $(\epsilon, \delta)$-differential privacy.*

*Proof.* We show the proof for case $k = 1$. It can be shown that for the absolute value of privacy loss (see Definition 2.5) of the Gaussian mechanism we get

$$\left|\mathcal{L}_{\mathcal{M}_G(x,f(\cdot),\epsilon,\delta)||\mathcal{M}_G(x',f(\cdot),\epsilon,\delta)}\right| \leq \left|\frac{1}{2\sigma^2}(2x\Delta_2 f + \Delta_2^2 f)\right|, \tag{2.20}$$

where $f$ is the object we want to perturb, $\Delta_2 f$ is the $\ell_2$-sensitivity of $f$ and $x$ is a Gaussian random variable with zero mean and $\sigma^2$ variance. Now, according to part 1 of Lemma 2.14 it is sufficient to show that (2.20) is bounded by $\epsilon$ with probability at least $1 - \delta$. Privacy loss is bounded by $\epsilon$ when $x < \sigma^2\epsilon/\Delta_2 f - \Delta_2 f/2$. So all we need to show is that

$$\Pr\big(|x| \geq \sigma^2\epsilon/\Delta_2 f - \Delta_2 f/2\big) < \delta.$$

Because $x$ is a Gaussian random variable with $\mathbb{E}(x) = 0$, we can only consider the tail of the distribution and require

$$\Pr\big(x \geq \sigma^2\epsilon/\Delta_2 f - \Delta_2 f/2\big) < \delta/2. \tag{2.21}$$

Now we can use Lemma 2.18 with $y = \sigma^2\epsilon/\Delta_2 f - \Delta_2 f/2$ to show (2.21). We require

$$\frac{\sigma}{y\sqrt{2\pi}}\exp\big(-y^2/2\sigma^2\big) < \delta/2 \Leftrightarrow \ln(y/\sigma) + y^2/2\sigma^2 > \ln\Big(2/\sqrt{2\pi}\delta\Big)$$
$$\Rightarrow \ln\big((\sigma^2\epsilon/\Delta_2 f - \Delta_2 f/2)/\sigma\big) + (\sigma^2\epsilon/\Delta_2 f - \Delta_2 f/2)^2/2\sigma^2 > \ln\Big(2/\sqrt{2\pi}\delta\Big).$$

Let us first consider the term inside the logarithm of the leftmost term in the above inequality. Denote it as $g(\sigma)$. We can easily see that since $\epsilon$ and $\Delta_f$ are positive, $g'(\sigma) > 0$, $\forall\sigma > 0$. We can solve the region where $g(\sigma) \geq 1$ and therefore its logarithm is positive.

$$g(\sigma) = 1 \Leftrightarrow \sigma^2\frac{\epsilon}{\Delta_2 f} - \sigma - \frac{\Delta_2 f}{2} = 0$$
$$\Rightarrow \sigma = \frac{(1 \pm \sqrt{1 + 2\epsilon})}{2}\frac{\Delta_2 f}{\epsilon}.$$

The other solution for $\sigma$ is clearly negative, so we consider only the case

$$\sigma \geq \frac{(1 + \sqrt{1 + 2\epsilon})}{2}\frac{\Delta_2 f}{\epsilon}.$$

Now it is easy to verify that if $\epsilon \geq \delta$, choosing $\sigma > \sqrt{2\ln(1.25/\delta)}\Delta_2 f/\epsilon$ yields $g(\sigma) > 1$ and therefore $\log(g(\sigma)) \geq 0$. Because we have proved that the first term in the inequality of interest is positive, it is sufficient to show that

$$h(\sigma) = (\sigma^2\epsilon/\Delta_2 f - \Delta_2 f/2)^2/2\sigma^2 > \ln\Big(2/\sqrt{2\pi}\delta\Big). \tag{2.22}$$

We can now see that $h(\sigma) = g^2(\sigma)/2$ and therefore in the range we are considering, $\sigma > \sqrt{2\ln(1.25/\delta)}\Delta_2 f/\epsilon$ the derivative $h'(\sigma) = g'(\sigma)g(\sigma)$ is positive. Again it is numerically easy to ensure that choosing $\sigma > \sqrt{2\ln(1.25/\delta)}\Delta_2 f/\epsilon$ satisfies the required condition and we conclude the proof. The proof for multidimensional version follows almost immediately from the 1-d case (see Dwork and Roth, 2014, Appendix A). □

## 2.3 Techniques for differential privacy

We have given the definition of differential privacy and shown an example of an algorithm that provides differential privacy. In this section we introduce some basic techniques used to assure differential privacy in general databases. Next we will give a toy example of a database and a query.

**Example 2.23.** In this example we are considering a diabetes study, where we have five individuals. The first column of the following table is the identity of an individual and the second column indicates if the individual has diabetes, denoted as 1, or not, denoted as 0.

| | |
|---|---|
| Alice | 0 |
| Bob | 1 |
| Charles | 0 |
| Diana | 1 |
| Eric | 1 |

Consider an adversary who wants to know if Diana has diabetes or not. Let us assume that there has been a similar study a year before that did not involve Diana but other participants were involved. Assume that an adversary knows that the status of other participants has not changed since the last study. Our goal is to release the number of diabetics in our study. Given the result of the previous study and the result of the same study but now Diana involved will compromise Diana's privacy because the results will differ by one and the adversary will know that Diana has diabetes.

In order to provide differential privacy we need to inject noise into our computations somehow. There are several methods for adding stochasticity to our inference. In this thesis we present input, output and objective perturbation. Besides these basic methods that are all based on additive noise there are methods such as exponential mechanism (McSherry and Talwar, 2007) and plausible deniability scheme, seen in Example 2.7, that provide differential privacy.

Input perturbation (Dwork, 2006) is the easiest way to achieve privacy. It adds noise to our pure dataset and we can release the noisy data to third party. Output perturbation

(Dwork, 2006) adds noise to the results of a query given pure data. Objective perturbation (Chaudhuri et al., 2011) adds noise to some task that we are performing on our dataset. In contrast to input perturbation, in neither output nor objective perturbation we are able to release the data to third party but only the results of a given query. Although it makes the release of data is possible, input perturbation is a very crude way of providing differential privacy, because of the noise level we are adding will affect the inference on this data a lot.

To summarise above methods, let us denote $Z$ as a random vector or scalar depending on the query we are considering:

**Input perturbation**: Given pure data $\mathbf{x}$ we release $\mathbf{x} + Z$

**Output perturbation**: Given query $f$ and data $\mathbf{x}$ we release $f(\mathbf{x}) + Z$

**Objective perturbation**: Given task $g$ and data $\mathbf{x}$ we release $g(\mathbf{x} + Z)$

## 2.4 Composition and enhancing privacy guarantees

Differential privacy is a strong privacy guarantee. It is in fact so strong that given output of an $\epsilon$- or $(\epsilon, \delta)$-differentially private algorithm, an adversary can do whatever he or she likes on the result and can never weaken the privacy guarantee. This property is called immunity to *post-processing*.

**Proposition 2.24.** *Given an $(\epsilon, \delta)$-differentially private algorithm $\mathcal{A}$ and any function $f$, the composition $f(\mathcal{A})$ is still $(\epsilon, \delta)$-differentially private.*

*Proof.*

$$\begin{aligned}\Pr(f(\mathcal{A}(x)) \in S) &= \Pr\big(\mathcal{A}(x) \in f^{-1}(S)\big) \\ &\leq e^{\epsilon} \Pr\big(\mathcal{A}(x') \in f^{-1}(S)\big) + \delta \\ &= e^{\epsilon} \Pr(f(\mathcal{A}(x')) \in S) + \delta.\end{aligned}$$

□

Immunity to post-processing is very useful because after we have provided differential privacy we can do anything we want on the results, if we do not access the data anymore.

Sometimes we do not need to use the whole dataset to respond to a query with sufficient accuracy. If we assume that all individuals of a dataset are somewhat similar we may just use a subset of the whole dataset to respond our query. Li et al. (2012) showed that using only a subset of the whole dataset actually amplifies the privacy guarantee of an unbounded $(\epsilon, \delta)$-differentially private algorithm.

**Theorem 2.25** (Privacy amplification)**.** *Consider an unbounded $(\epsilon, \delta)$-differentially private algorithm $\mathcal{A}$. Using a subsample of size $qN$, where $N$ is the size of our whole dataset, as an input for $\mathcal{A}$ preserves unbounded $(\epsilon', \delta')$-differential privacy where*

$$\epsilon' = \log(1 + q(e^\epsilon - 1))$$
$$\delta' = q\delta.$$

*Proof.* Consider datasets $x$ and $x'$ such that $x' = x \setminus \{i\}$. Denote the subsample with $T$ and the distribution of $\mathcal{A}(x)$ with $\Pr_x$. For the subsampled input on $\mathcal{A}$ it clearly holds that $\Pr_x(S|i \notin T) = \Pr_{x'}(S)$ and $\Pr_x(S|i \in T) \leq e^\epsilon \Pr_{x'}(S) + \delta$ for all $S \in \text{im}(\mathcal{A})$. Using the law of total probability we get

$$\begin{aligned}\Pr_x(S) &= (1-q)\Pr_x(S|i \notin T) + q\Pr_x(S|i \in T) \\ &\leq (1-q)\Pr_{x'}(S) + qe^\epsilon \Pr_{x'}(S) + q\delta \\ &= (1 + q(e^\epsilon - 1))\Pr_{x'}(S) + q\delta.\end{aligned}$$

□

So far we have considered the case where individuals give their data to a curator who performs a query and provides some privacy guarantees. In real life however we admit our data to several different queries. It is possible for an adversary to weaken the privacy guarantees of a differentially private query by using the results of these previous queries, even if they all are differentially private. Differential privacy has a composing property, which means that $\epsilon$'s and $\delta$'s add up. The basic version of this is called the *basic composition theorem*. It gives a way of computing the privacy cost of multiple differentially private queries applied on the same dataset.

**Theorem 2.26** (Basic composition theorem)**.** *Given algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_n$ which are $(\epsilon_1, \delta_1), \ldots, (\epsilon_n, \delta_n)$ differentially private, respectively, the composition of these algorithms is $(\sum_{i=1}^n \epsilon_i, \sum_{i=1}^n \delta_i)$-differentially private.*

*Proof.* (See Dwork and Roth, 2014, Appendix B). □

Although basic composition theorem gives us a way to accumulate the privacy cost on multiple queries it is rather crude, and given that $\epsilon_i = \epsilon$ with $\forall i$ the privacy cost increases linearly. Dwork et al. (2010) showed that we can enhance the $\epsilon$ part of total privacy cost by sacrificing the $\delta$ part by a small additional factor. This enhancement is known as the *advanced composition theorem*.

**Theorem 2.27** (Advanced composition theorem)**.** *Let $\mathcal{A}^k$ be a $k$-fold composition of $(\epsilon, \delta)$-differentially private algorithms. Algorithm $\mathcal{A}^k$ then provides $(\epsilon', k\delta + \delta')$-differential privacy with*

$$\epsilon' = \sqrt{2k\ln(1/\delta')}\epsilon + k\epsilon(e^\epsilon - 1)$$

*and* $\delta' \in [0, 1]$ *arbitrary.*

*Proof.* (See Dwork et al., 2010, Appendix B). □

Now let us take a look at an example where composition theorems become useful.

**Example 2.28.** Consider a study where we want to minimize the sum of squared error between $y_i$ and $wx_i$ where $y_i, x_i, w \in \mathbb{R}$. Our cost function is

$$S(w) = \sum_{i=1}^{N} (y_i - wx_i)^2.$$

This particular problem has an analytical solution

$$\hat{w} = \left( \sum_{i=1}^{N} x_i^2 \right)^{-1} \sum_{i=1}^{N} x_i y_i.$$

If we had information that both $x_i, y_i$ are bounded we could easily calculate the sensitivity of $\hat{w}$ and just release the perturbed least squares solution. Instead of using the analytical solution we use a gradient-based method to minimize $S(w)$. Non-private updates are

$$w_n = w_{n-1} - \eta S'(w_{n-1})$$
$$S'(w_{n-1}) = -\sum_{i=1}^{N} 2x_i(y_i - w_{n-1}x_i),$$

where $\eta > 0$ is a learning rate. In order to preserve privacy we bound the absolute value of each individual contribution to the sum in $S'$ by $C$. This kind of *clipping* solution gives us a way to compute the sensitivity of $S'$. After clipping, the $\ell_1$ sensitivity between adjacent datasets $D, D'$ becomes

$$\Delta S' = \left| \sum_{i \in D} c(2x_i(y_i - w_{n-1}x_i)) - \sum_{j \in D'} c(2x_j(y_j - w_{n-1}x_j)) \right| \leq 2C$$

where $c(\cdot)$ denotes clipping. Because of the iterative use of data we need to use composition theorems to compute the privacy cost. At each update we perturb the gradient of $S'$ by adding Laplacian noise to it.

Figure 2.1 shows the approximation of $w$ given by above-described gradient-based algorithm applied to a synthetic dataset with 100 samples. Clipping value $C$ was set to 0.5 and for each $\epsilon$ value the algorithm was run for 500 iterations. We can clearly see that noise level used in BCT curve perturbs the learning too much. However allowing small relaxation to BCT budget as $\delta = 10^{-6}$ our results improve a lot.

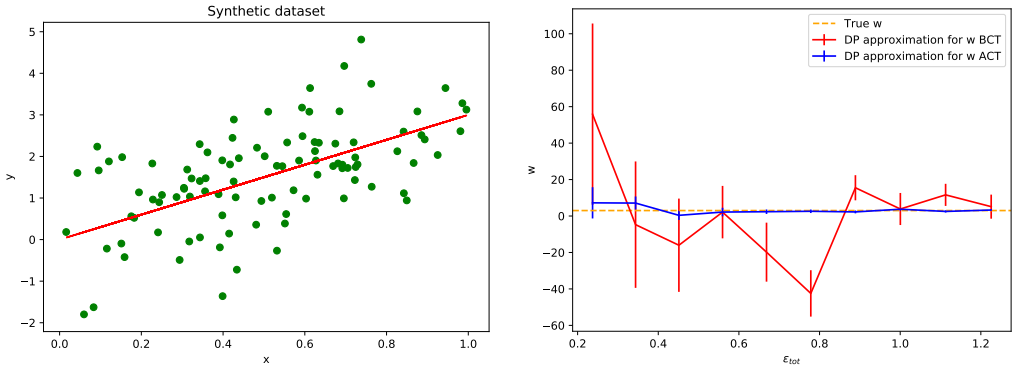


Figure 2.1: On the left the synthetic dataset with true fit. On the right comparison between Advance composition theorem (ACT) and Basic composition theorem (BCT). The curve shows mean of 20 runs of algorithm for both ACT and BCT with errorbars denoting the standard error of mean. The $\delta'$ of ACT was set to 1e-6.

There has been a lot of research on composing differentially private queries. Kairouz et al. (2015) showed improved results on both $\epsilon$ and $\delta$ part of the privacy budget compared to the ACT. Another improvement on composing private queries called *Moments accountant* (MA) was introduced by Abadi et al. (2016). Moments accountant is a privacy accounting method tailored to Gaussian mechanism. It provides strong composition result when randomized algorithm takes as input just a fraction of the whole dataset i.e. it takes advantage of subsampling through privacy amplification. The proof can be found in the appendix of Abadi et al. (2016). Next example illustrates the power of MA and privacy amplification.

**Example 2.29.** Consider a task where we use Gaussian mechanism to provide differential privacy. Let us assume that our dataset consists of $N = 1000$ samples and we use subsampling with $q = 0.005$. We compared the level of perturbation needed for different number of data passes to achieve $(1.0, 10^{-3})$-differential privacy. Figure 2.2 shows the $\sigma$ required in each iteration to maintain aforementioned privacy guarantee. We can see that increasing the number of data passes ten fold, does not affect the $\sigma$ of the MA much whereas $\sigma$ of the amplified version of ACT is increased from 2 to 6.

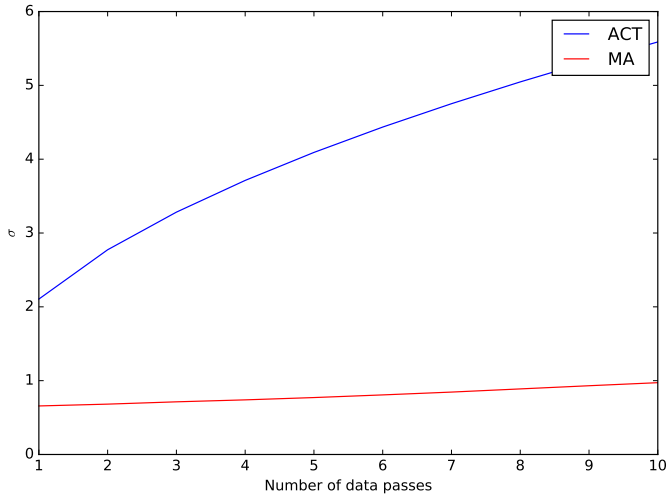


Figure 2.2: Comparison of the level of perturbation between the moments accountant and the amplified version of advanced composition theorem. The privacy budget for the Gaussian mechanism is $(1.0, 10^{-3})$ and the subsampling ratio $q$ is set to 0.005.

# Chapter 3

# Variational Bayesian methods

> *An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.*
>
> John Tukey

Consider a probabilistic model that consists of observed random variables $\mathbf{X}$ and unobservable random variables $\mathbf{Z}$ that include model parameters and latent variables. We denote the prior distribution of the parameters with $p(\mathbf{Z})$ and likelihood of $\mathbf{X}$ given $\mathbf{Z}$ with $p(\mathbf{X}|\mathbf{Z})$. In Bayesian learning tasks we would like to infer the posterior distribution of $\mathbf{Z}$ i.e. probability density of $\mathbf{Z}$ given data denoted with $p(\mathbf{Z}|\mathbf{X})$. Applying Bayes' rule and marginalization we get the posterior as

$$p(\mathbf{Z}|\mathbf{X}) = \frac{p(\mathbf{X},\mathbf{Z})}{p(\mathbf{X})} = \frac{p(\mathbf{X},\mathbf{Z})}{\int_{\mathrm{Supp}(\mathbf{Z})} p(\mathbf{X},\mathbf{Z})\mathrm{d}\mathbf{Z}}, \tag{3.1}$$

where $p(\mathbf{X},\mathbf{Z}) = p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})$ denotes the joint probability between observed and unobserved random variables. Following example shows how to analytically infer the posterior distribution for a certain model.

**Example 3.2.** Consider Gaussian linear regression model

$$\begin{aligned} y_n &\sim \mathcal{N}(wx_n, \tau) \\ w, \tau &\sim \mathrm{NINV}(0, \lambda_0, a_0, b_0), \end{aligned}$$

where NINV denotes normal-inverse-gamma. The likelihood and prior densities are defined as

$$\begin{aligned} p(\mathbf{y}|w,\tau) &= \prod_{n=1}^{N} p(y_n|w,\tau) = \prod_{n=1}^{N} \sqrt{\frac{1}{2\pi\tau}} \exp\left(-\frac{1}{2\tau}(y_n - wx_n)^2\right) \\ &= \left(\frac{1}{2\pi\tau}\right)^{N/2} \exp\left(-\frac{1}{2\tau}\sum_{n=1}^{N}(y_n - wx_n)^2\right) \\ p(w,\tau) &= \sqrt{\frac{\lambda_0}{\tau 2\pi}}\frac{b_0^{a_0}}{\Gamma(a_0)}\left(\frac{1}{\tau}\right)^{a_0+1} \exp\left(-\frac{2b_0 + \lambda_0 w^2}{2\tau}\right). \end{aligned}$$

Let us calculate the posterior. We start by calculating $p(\mathbf{y})$

$$p(\mathbf{y}) = \int_{\mathrm{Supp}(w)\times\mathrm{Supp}(\tau)} p(\mathbf{y}|w,\tau)p(w,\tau)\mathrm{d}\tau\mathrm{d}w \tag{3.3}$$

$$\propto \int_{\mathbb{R}}\int_{\mathbb{R}^+} \tau^{-(N+1)/2-a_0-1} \exp\left(-\frac{1}{2\tau}\left(2b_0 + \lambda_0 w^2 + \sum_{n=1}^{N}(y_n - wx_n)^2\right)\right)\mathrm{d}\tau\mathrm{d}w. \tag{3.4}$$

The terms inside the exponential in (3.4) can be rewritten as

$$2b_0 + \sum_{n=1}^{N} y_n^2 - \left(\sum_{n=1}^{N} x_n^2 + \lambda_0\right) r^2 + \left(\sum_{n=1}^{N} x_n^2 + \lambda_0\right)(w - r)^2,$$

where

$$r = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_{n=1}^{N} x_n^2 + \lambda_0}.$$

We can see that integrand in (3.4) is just unnormalized normal-inverse-gamma density. Therefore $p(\mathbf{y})$ becomes

$$p(\mathbf{y}) = (2\pi)^{-N/2}\frac{b_0^{a_0}}{\Gamma(a_0)}\frac{\Gamma(a_1)}{b_1^{a_1}}\sqrt{\frac{\lambda_0}{\lambda_1}},$$

with

$$\begin{aligned} a_1 &= a_0 + N/2 \\ b_1 &= 2b_0 + \sum_{n=1}^{N} y_n^2 - \left(\sum_{n=1}^{N} x_n^2 + \lambda_0\right) r^2 \\ \lambda_1 &= \left(\sum_{n=1}^{N} x_n^2 + \lambda_0\right) \\ \mu_1 &= r \end{aligned}$$

Finally we get the posterior $p(w, \tau|\mathbf{y})$ by plugging $p(\mathbf{y})$ into (3.1).

Unfortunately, the posterior distribution is not always analytically available. There are models where the integral in the denominator of (3.1) is intractable, and therefore direct posterior calculation is impossible. One example of such models is the previous Gaussian linear regression example with a slight modification. This will be seen in Example 3.16. When analytical posterior is intractable we can still approximate it. The approximation can be done by using Markov chain Monte Carlo (MCMC) methods that draw samples from the posterior. Because MCMC methods rely on sampling they can be computationally inefficient especially for large scale problems. Also the convergence of the chain can be difficult to check. Instead of focusing to MCMC methods we use Variational Bayesian (VB) methods to approximate the intractable posterior distribution.

## 3.1 Variational Inference

We start our variational Bayes method consideration with formulation of an important quantity called *evidence lower bound* (ELBO). First we observe that we can rewrite the log-evidence $\ln p(\mathbf{X})$ in following way.

**Theorem 3.5.** *For arbitrary probability density function $q(\mathbf{Z})$, the following holds*

$$\ln p(\mathbf{X}) = \int q(\mathbf{Z}) \ln \left( \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right) \mathrm{d}\mathbf{Z} - \int q(\mathbf{Z}) \ln \left( \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right) \mathrm{d}\mathbf{Z}. \tag{3.6}$$

*Proof.*

$$\begin{aligned}
&\int q(\mathbf{Z}) \ln \left( \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right) \mathrm{d}\mathbf{Z} - \int q(\mathbf{Z}) \ln \left( \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right) \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) \left( \ln \left( \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right) - \ln \left( \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right) \right) \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) \ln \left( \frac{p(\mathbf{X}, \mathbf{Z})}{p(\mathbf{Z}|\mathbf{X})} \right) \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) \ln \left( \frac{p(\mathbf{Z}|\mathbf{X}) p(\mathbf{X})}{p(\mathbf{Z}|\mathbf{X})} \right) \mathrm{d}\mathbf{Z} = \ln p(\mathbf{X})
\end{aligned}$$

□

We identify the second term on the RHS of (3.6) as the Kullback-Leibler (KL) divergence between $q(\mathbf{Z})$, that we call the *variational distribution*, and the true posterior $p(\mathbf{Z}|\mathbf{X})$. We denote the first term on the RHS of (3.6) with $\mathcal{L}(q)$.

The following lemma is one of our key components for formulating the variational distribution $q(\mathbf{Z})$.

**Lemma 3.7.** *KL-divergence $KL(p_0 \,||\, p_1)$ between any two densities $p_0$ and $p_1$ is non-negative and zero exactly when $p_0 = p_1$.*

*Proof.*

$$\begin{aligned}
\mathrm{KL}(p_0 \,||\, p_1) &= \int_{\mathbb{R}} p_0(x) \ln \frac{p_0(x)}{p_1(x)} \mathrm{d}x \\
&= -\int_{\mathbb{R}} p_0(x) \ln \frac{p_1(x)}{p_0(x)} \mathrm{d}x \\
&\geq -\int_{\mathbb{R}} p_0(x) \left( \frac{p_1(x)}{p_0(x)} - 1 \right) \mathrm{d}x = 0
\end{aligned}$$

where the inequality follows from the fact that $\ln x$ is a concave function that intersects line $x - 1$ only in $x = 1$ and therefore $\ln x \leq x - 1$. It is clear that when $p_0 = p_1$, the KL-divergence vanishes. □

The KL-divergence is a measure of the difference between two probability distributions (see e.g. Cover and Thomas, 2012) so we can gain approximations of $p(\mathbf{Z}|\mathbf{X})$ by minimizing the divergence between the variational and the true posterior. However this is not feasible because we need the true posterior. Recall that we wrote the log-evidence $\ln p(\mathbf{X})$ as

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q(\mathbf{Z}) \,||\, p(\mathbf{Z}))$$

now from non-negativity of KL-divergence we get

$$\ln p(\mathbf{X}) \geq \mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left( \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right) \mathrm{d}\mathbf{Z}. \tag{3.8}$$

Because of the above inequality, $\mathcal{L}(q)$ is called the evidence lower bound. Assume now that the variational distribution $q(\mathbf{Z})$ factorizes in the following way

$$q(\mathbf{Z}) = \prod_{i=1}^{M} q_i(\mathbf{Z}_i), \tag{3.9}$$

where $\mathbf{Z}_i$ are the of elements of $\mathbf{Z}$. Substituting (3.9) into $\mathcal{L}(q)$ in (3.8) yields

(3.10)

$$\mathcal{L}(q) = \int \prod_i q_i(\mathbf{Z}_i) \left( \ln p(\mathbf{X}, \mathbf{Z}) - \sum_i \ln q_i(\mathbf{Z}_i) \right) \mathrm{d}\mathbf{Z}$$

$$= \int q_j(\mathbf{Z}_j) \left( \int \ln p(\mathbf{X}, \mathbf{Z}) \prod_{i \neq j} q_i(\mathbf{Z}_i) \, \mathrm{d}\mathbf{Z}_i \right) \mathrm{d}\mathbf{Z}_j - \int q_j(\mathbf{Z}_j) \ln q_j(\mathbf{Z}_j) \, \mathrm{d}\mathbf{Z}_j + \text{const} \tag{3.11}$$

$$= \int q_j(\mathbf{Z}_j) \ln \tilde{p}(\mathbf{X}, \mathbf{Z}_j) \mathrm{d}\mathbf{Z}_j - \int q_j(\mathbf{Z}_j) \ln q_j(\mathbf{Z}_j) \, \mathrm{d}\mathbf{Z}_j + \text{const} \tag{3.12}$$

Where const denotes a constant w.r.t. distribution $q_j$. We have used notation

$$\ln \tilde{p}(\mathbf{X}, \mathbf{Z}_j) = \mathbb{E}_{i \neq j} \left[ \ln p(\mathbf{X}, \mathbf{Z}) \right] + \text{const} \tag{3.13}$$

where the notation $\mathbb{E}_{i \neq j}$ is adopted from Bishop (2006) and it denotes the expectation w.r.t. the variational distribution $q$ over variables $\mathbf{Z}_i$ such that $i \neq j$ i.e., $\prod_{i \neq j} q_i(\mathbf{Z}_i)$. From (3.12) we recognize negative KL divergence between $q_j(\mathbf{Z}_j)$ and $\tilde{p}(\mathbf{X}, \mathbf{Z}_j)$ so maximizing $\mathcal{L}(q)$ is equivalent to minimizing this KL divergence w.r.t $q_j$ while keeping $q_{i \neq j}$ fixed. This minimum occurs when $q_j(\mathbf{Z}_j) = \tilde{p}(\mathbf{X}, \mathbf{Z}_j)$ so expression for optimal solution $q_j^*(\mathbf{Z}_j)$ is given by

$$\ln q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i \neq j} \left[ \ln p(\mathbf{X}, \mathbf{Z}) \right] + \text{const} \tag{3.14}$$

which finally yields

$$q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i \neq j} \left[ \ln p(\mathbf{X}, \mathbf{Z}) \right])}{\int \exp(\mathbb{E}_{i \neq j} \left[ \ln p(\mathbf{X}, \mathbf{Z}) \right]) \mathrm{d}\mathbf{Z}_j}. \tag{3.15}$$

Next we show an example of how to use variational inference.

**Example 3.16.** Consider Gaussian linear regression model as in Example 3.2, but with slight modification

$$\begin{aligned} y_n &\sim \mathcal{N}(wx_n, \tau) \\ w, \tau &\sim \text{NINV}(0, \lambda, a_0, b_0) \\ \lambda &\sim \text{Gam}(\alpha, \beta). \end{aligned}$$

Densities are given as

$$\begin{aligned} p(\mathbf{y}; w, \tau) &= \left( \frac{1}{2\pi\tau} \right)^{N/2} \exp\left( -\frac{1}{2\tau} \sum_{n=1}^{N} (y_n - wx_n)^2 \right) \\ p(w, \tau; 0, \lambda, a_0, b_0) &= \sqrt{\frac{\lambda}{\tau 2\pi}} \frac{b_0^{a_0}}{\Gamma(a_0)} \left( \frac{1}{\tau} \right)^{a_0+1} \exp\left( -\frac{2b_0 + \lambda w^2}{2\tau} \right) \\ p(\lambda; \alpha, \beta) &= \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp(-\beta\lambda). \end{aligned}$$

The joint density function $p(\mathbf{y}, w, \tau, \lambda)$ becomes

$$p(\mathbf{y}, w, \tau, \lambda) = C_1 \tau^{-\frac{N}{2} - a_0 - 1 - \frac{1}{2}} \exp\left( -\frac{1}{2\tau} \left( 2b_0 + \lambda w^2 + \sum_n (y_n - wx_n)^2 \right) - \beta\lambda \right) \lambda^{\alpha + \frac{1}{2} - 1}.$$

In order to compute the analytical posterior we need to marginalize the above density w.r.t. $w, \tau$ and $\lambda$. However, after marginalization over $\lambda$, we get something relatively complicated

$$p(\mathbf{y}, w, \tau) = C_2 \tau^{-\frac{N}{2}-a_0-1-\frac{1}{2}} \exp\left(-\frac{1}{2\tau}\left(2b_0 + \sum_n (y_n - wx_n)^2\right)\right)\left(\frac{w^2}{2\tau} + \beta\right)^{-\alpha+\frac{1}{2}}.$$

Now instead of calculating the analytical posterior, we approximate the posterior $p(w, \tau, \lambda|x)$ with variational distribution $q(w, \tau, \lambda)$, which we assume factorizes as $q(w, \tau)q(\lambda)$. Using (3.14), we get

$$\begin{aligned}
\ln q^*(\lambda) &= \mathbb{E}_{w,\tau}[\ln p(\mathbf{y}, w, \tau, \lambda)] + \text{const} \\
&= \ln p(\lambda; \alpha, \beta) + \underbrace{\mathbb{E}_{w,\tau}[\ln p(\mathbf{y}; w, \tau)]}_{\text{const. w.r.t. } \lambda} + \mathbb{E}_{w,\tau}[\ln p(w, \tau; 0, \lambda, a_0, b_0)] + \text{const} \\
&= (\alpha - 1)\ln\lambda - \beta\lambda + \frac{1}{2}\ln\lambda - \mathbb{E}_{w,\tau}\left[\lambda\frac{w^2}{2\tau} - \left(\frac{1}{2} + a_0 - 1\right)\ln\tau\right] + \text{const} \\
&= \left(\alpha - \frac{1}{2}\right)\ln\lambda - \lambda\left(\beta + \frac{1}{2}\mathbb{E}_{w,\tau}\left[\frac{w^2}{\tau}\right]\right) + \text{const}.
\end{aligned}$$

So we see, that $q^*(\lambda)$ takes form of gamma distribution with parameters

$$\begin{aligned}
\alpha_t &= \alpha + \frac{1}{2} \\
\beta_t &= \beta + \frac{1}{2}\mathbb{E}_{w,\tau}\left[\frac{w^2}{\tau}\right].
\end{aligned}$$

Next we calculate $q^*(w, \tau)$.

$$\begin{aligned}
\ln q^*(w, \tau) &= \mathbb{E}_\lambda[\ln p(\mathbf{y}, w, \tau, \lambda)] + \text{const} \\
&= \ln p(\mathbf{y}; w, \tau) + \mathbb{E}_\lambda[\ln p(w, \tau; 0, \lambda, a_0, b_0)] + \underbrace{\mathbb{E}_\lambda[\ln p(\lambda; \alpha, \beta)]}_{\text{const. w.r.t } w,\tau} + \text{const} \\
&= -\frac{N}{2}\ln\tau - \frac{1}{2\tau}\sum_{n=1}^{N}(y_n - wx_n)^2 - (\frac{1}{2} + a_0 + 1)\ln\tau - \frac{w^2}{2\tau}\mathbb{E}_\lambda[\lambda] - \frac{b_0}{\tau} + \text{const} \\
&= -\ln\tau\left(\frac{N+1}{2} + a_0 + 1\right) - \frac{1}{2\tau}\left(\sum_{n=1}^{N}(y_n - wx_n)^2 + w^2\mathbb{E}_\lambda[\lambda] + 2b_0\right) + \text{const}.
\end{aligned}$$

We can rearrange the term inside the exponential as we did in Example 3.2 and we get

yet another normal-inverse-gamma distribution with hyperparameters

$$
\begin{aligned}
a_t &= a_0 + N/2 \\
b_t &= 2b_0 + \sum_{n=1}^{N} y_n^2 - \lambda_t \mu_t^2 \\
\lambda_t &= \left( \sum_{n=1}^{N} x_n^2 + \mathbb{E}_\lambda[\lambda] \right) \\
\mu_t &= \frac{\sum y_n x_n}{\lambda_t}.
\end{aligned}
$$

The remaining two expectations $\mathbb{E}_\lambda[\lambda]$ and $\mathbb{E}_{w,\tau}\left[\frac{w^2}{\tau}\right]$ become

$$
\begin{aligned}
\mathbb{E}_\lambda[\lambda] &= \frac{a_t}{b_t}, \\
\mathbb{E}_{w,\tau}\left[\frac{w^2}{\tau}\right] &= \frac{1}{\lambda_t} + \frac{\mu_t a_t}{b_t}.
\end{aligned}
$$

We first initialize one of these to some value and then update each parameter iteratively.

## 3.2 Reparameterization trick

In the beginning of this chapter we gave analytically intractable posteriors as a motivation for variational learning. We saw in example 3.16 how we can maximize the lower bound using (3.14). However, analytical solutions for expectations taken in (3.14) are generally intractable. For example with the Bayesian logistic regression, model given in Example 3.19, we run into trouble while trying to use the standard VB approach. This happens often with non-conjugate models.

There are methods to overcome the analytically intractable expectations in the variational treatment of this model, for example local variational approach (Bishop, 2006) or augmentation (Polson et al., 2013). Kingma and Welling (2013) showed a method to overcome the difficulty of evaluating analytical expectations w.r.t. the variational distribution. We begin by assuming that the variational distribution of parameter vector $\mathbf{w}$ takes the form of a multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and a diagonal covariance matrix $\boldsymbol{\sigma}^2\mathbf{I}$.

$$
\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I}).
$$

By properties of normal distribution, we can write $\mathbf{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\odot$ denotes elementwise multiplication. This *reparametrization* of $\mathbf{w}$ with $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$

is a key component in our inference. We recall that our aim is to maximize the ELBO which takes the form

$$\mathcal{L}(q) = -\mathrm{KL}(q(\mathbf{w};\boldsymbol{\xi}) \,||\, p(\mathbf{w})) + \mathbb{E}_{q(\mathbf{w};\boldsymbol{\xi})}\left[\log p(\mathcal{D}|\mathbf{w})\right]. \tag{3.17}$$

Both terms on the right-hand side are expectations w.r.t. the variational distribution, which we assumed to be a multivariate normal distribution. These expectations may still be intractable, but we can approximate them using *Monte Carlo integration*.

**Definition 3.18** (Monte Carlo integration)**.** Given a function $f$ and a probability measure $\mu(\mathrm{d}x)$, the integral

$$\int_\Omega f(x)\mu(\mathrm{d}x)$$

can be approximated with Monte Carlo integral

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i), \text{ where } x_i \sim \mu.$$

Using Monte Carlo (MC) integration we can now write both of these expectations in terms of $\boldsymbol{\epsilon}$ from which we can sample. After the MC integration our ELBO is now a function of variational parameters $\boldsymbol{\xi} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$.

Next we show an example of how to use the reparametrization trick to make analytically intractable ELBO available.

**Example 3.19.** Consider Bayesian logistic regression model that was mentioned in the beginning of this Section

$$\begin{aligned} P(y|\mathbf{x},\mathbf{w}) &= \sigma(y\mathbf{w}^T\mathbf{x}) \\ p(\mathbf{w}) &= \mathcal{N}(\mathbf{w};\mathbf{w}_0,\mathbf{S}_0) \\ \sigma(x) &= \frac{1}{1+e^{-x}}. \end{aligned}$$

The sigmoid $\sigma(\cdot)$ makes both the analytical posterior and the standard variational approach difficult. Assume now that the variational distribution takes the form of a multivariate Gaussian distribution $q(\mathbf{w};\boldsymbol{\xi}) = \boldsymbol{\phi}(\mathbf{w};\boldsymbol{\mu},\boldsymbol{\sigma}^2\mathbf{I})$. We approximate now the variational expectation of likelihood with MC integration.

$$\mathbb{E}_{q(\mathbf{w};\boldsymbol{\xi})}\left[\log p(y_i|\mathbf{w})\right] \approx \frac{1}{L}\sum_{l=1}^{L} \log p(y_i|\mathbf{w}_l),$$

where $\mathbf{w}_l$ is a sample from the $N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$. Because we want to optimize the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ we rewrite $\mathbf{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I})$. Now we can rewrite (3.8) as

$$\mathcal{L}(\boldsymbol{\xi}; y_i) \approx -\mathrm{KL}(q(\mathbf{w}; \boldsymbol{\xi}) \,||\, p(\mathbf{w})) + \frac{1}{L} \sum_{l=1}^{L} \log \sigma(y_i(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_l)^T \mathbf{x}_i)$$

The negative KL-divergence of multivariate normal distribution (Duchi, 2007) becomes

$$-\mathrm{KL}(q(\mathbf{w}; \boldsymbol{\xi}) \,||\, p(\mathbf{w})) = \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2\right).$$

So the approximate ELBO becomes:

$$\tilde{\mathcal{L}}_i = \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2\right) + \frac{1}{L} \sum_{l=1}^{L} \log \sigma(y_i(\boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\epsilon}_l)\mathbf{x}_i).$$

Partial derivatives w.r.t $\boldsymbol{\mu}$s components yield

$$\frac{\partial \mathcal{L}_i}{\partial \mu_k} = -\mu_k + \frac{1}{L} \sum_{l=1}^{L} y_i \mathbf{x}_{i_k} (1 - \sigma(y_i(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_l)^T \mathbf{x}_i))$$

and w.r.t to $\boldsymbol{\sigma}$s components

$$\frac{\partial \mathcal{L}_i}{\partial \sigma_k} = \frac{1}{\sigma_k} - \sigma_k + \frac{1}{L} \sum_{l=1}^{L} y_i \boldsymbol{\epsilon}_{l_k} \mathbf{x}_{i_k} (1 - \sigma(y_i(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_l)^T \mathbf{x}_i)).$$

Now we can apply any gradient based-optimizer to our problem and obtain an approximation of the posterior distribution.

## 3.3 Doubly-Stochastic Variational Inference

The reparametrization trick introduced in the previous section gives us a way to overcome intractable expectations. However, the likelihood term that appears in (3.17) can be computationally very expensive. Common approach in machine learning is to approximate this kind of costly functions that can be separated into individual contributions, by using only a random subset of the original data. Stochastic gradient descent (SGD) (Robbins and Monro, 1951) is a method that does exactly this. It iteratively updates the parameters of a cost function by taking a random minibatch of the original dataset, computes the

gradient of the cost function given this minibatch, scales the minibatch gradient by a factor of $(N/N_m)$, where $N_m$ denotes the size of minibatch, and descents in the direction given by this gradient approximation multiplied with step size. There has been lot of research on the SGD method to improve learning. Several methods have been developed to make the step size in gradient descent adaptive. In this thesis we consider a popular method called AdaGrad (Duchi et al., 2011).

Titsias and Lázaro-Gredilla (2014) introduced an idea of applying SGD into variational inference approximated with the reparametrization trick. This method is called doubly-stochastic variational inference (DSVI). The DSVI method makes the original likelihood computation less costly by computing the gradient from only a subset of the dataset.

# Chapter 4

# Differentially Private Variational Inference

Our aim is to build a general differentially private framework for posterior estimation. Motivation for such privacy preserving method is to protect the data of the individuals in our dataset. Releasing the posterior distribution without a privacy layer could enable some adversary to deduce sensitive information about the individuals back from the results.

We propose a method that is easy to implement and is applicable to a wide range of models. Differentially Private Variational Inference (DPVI) is based on automatic differentiation variational inference (ADVI) by Kucukelbir et al. (2017) that applies "black box" differentiation on doubly-stochastic variational inference introduced in Section 3.3. DPVI preserves privacy by perturbing the gradients of DSVI with Gaussian noise. The contribution of an individual's gradient is restricted by norm clipping in order to bound the sensitivity of total gradient. We use the AdaGrad to make the step size of gradient ascent adaptive. The AdaGrad was chosen because it scales each component of the gradient by the $\ell_2$-norm of previous gradient components and therefore assures that for example even if the Gaussian noise contribution happens to be large, we do not jump too far.

Even though stochastic-gradient-based optimization reduces the computational cost, stochasticity helps DPVI also in another sense. Subsampling that is used in SGD enhances the privacy guarantees through privacy amplification. This is an essential part of the idea behind DPVI.

Because of the generality of the DSVI method, DPVI is a general framework for approximate posterior inference. In our experiments we used spherical multivariate normal distributions as our approximate distribution in the reparametrization. This is mainly because using a full rank covariance matrix in the reparametrization makes the computations more costly and also it seems that the perturbation makes the full rank approximation worse than spherical approximation. However DPVI easily generalizes to any

reparametrization available.

**Algorithm 1** DPVI

**Input:** Data set $\mathcal{D}$, sampling probability $q$, number of iterations $T$, SGA step size $\eta_t$, Clipping threshold $c_t$ and initial values $\boldsymbol{\xi}_0$.

**for** $t \in [T]$ **do**

1. Pick random sample $U$ from $\mathcal{D}$ with sampling probability $q$
2. Calculate gradient of ELBO for each $i \in U$
3. Clip and sum gradients:
4. $\tilde{g}_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{||g_t(x_i)||_2)}{c_t})$
5. $\tilde{g}_t \leftarrow \sum_i \tilde{g}_t(x_i)$
6. Add noise: $\tilde{g}_t \leftarrow \tilde{g}_t + \mathcal{N}(0, 4c_t^2\sigma^2\mathbf{I})$
7. Update AdaGrad parameter. $G_t \leftarrow G_{t-1} + \tilde{g}_t^2$
8. Ascent: $\boldsymbol{\xi}_t \leftarrow \boldsymbol{\xi}_{t-1} + \eta_t \tilde{g}_t / \sqrt{G_t}$

**end for**

Next we show that the DPVI algorithm is differentially private.

**Theorem 4.1.** *Algorithm 1 preserves differential privacy.*

*Proof.* At each iteration of Algorithm 1, we compute the gradients and clip their norm using the predetermined clipping threshold $c_t$. Clipping forces the $\ell_2$ norms of each individual gradient to be less than or equal to $c_t$. Now the sensitivity of $\tilde{g}_t$ becomes

$$\Delta_2 \tilde{g}_t = \sup_{i \neq j} ||\tilde{g}_t(x_i) - \tilde{g}_t(x_j)||_2 \leq 2 \sup_i ||\tilde{g}_t(x_i)||_2 \leq 2c_t.$$

It is now clear that applying the Gaussian mechanism to each component of $\tilde{g}_t$ preserves differential privacy as we see from Theorem 2.17. □

As we can see from the description of Algorithm 1, DPVI has many parameters that affect both to the accuracy of our posterior and the privacy budget. Next we give some rules of thumb, how these parameters are expected to affect the learning. We show some examples of the effect of these parameters in the Section 5.

Maybe the most important parameter to tune is the number of iterations $T$. It is clear that it has maybe the biggest effect on convergence of parameters of DSVI. Without privacy we could just let the algorithm run until convergence and release the final parameters. However the privacy guarantees degrade as the number of iterations increase. Therefore it is very important for a curator to find a sufficiently small number of iterations to provide both good performance of the algorithm and sufficient privacy guarantees.

Because of the use of subsampling, gradients at each iteration affect the parameters as an averaged version of the gradient w.r.t the whole dataset. Using large subsample size will make the sum of gradients less vulnerable to the Gaussian noise. On the other hand, we want to use a relatively small value of $q$ to take advantage of the privacy amplification.

Another important parameter is the clipping thereshold $c_t$. We would like to preserve the original gradient as much as we could, but because the noise level depends linearly on the clipping threshold, we need to find some moderately small threshold. A good practise is to find a small $c_t$ value in the non-private case and then try to apply it to private version.

We also have parameters that affect only the performance of DPVI via the SGD. The learning rate $\eta_t$ is one of these. Although we use the AdaGrad to make the learning rate adaptive with respect to the previous gradients, we still have chosen to include this kind of parameter that controls the base level of learning rate. When it comes to calibrating the DPVI, $\eta_t$ is really an important parameter. Because of the additional differential privacy noise, it is possible that our perturbed gradient will point to some very bad direction, that it cannot ever come back again. Therefore it is important to find such $\eta_t$ so that the leaps are not too big. Our experiments, not presented in this thesis, also suggest to reset the AdaGrad parameter after some number of iterations. This is done to boost the learning a bit. However resetting the AdaGrad parameter too often will lead to bad results, because the leaps become large.

# Chapter 5

# Experiments

In this chapter we apply DPVI to two different models, logistic regression and the Gaussian mixture model.

We compared the DPVI method against the state-of-art Differentially private stochastic gradient Langevin dynamics (DP-SGLD) method by Wang et al. (2015). The DP-SGLD algorithm is a Hamiltonian Monte-Carlo-based approach to draw samples from the posterior distribution. It relies on the Lipschitz continuity of the model likelihood to control the sensitivity. The noise addition happens through the Gaussian mechanism. Because DP-SGLD is also an iterative algorithm it uses the composition property of differential privacy similar to the ACT (see Wang et al., 2015, Theorem 4) to provide exact bounds on the privacy spent.

**Algorithm 2** DP-SGLD

**Input:** Data $X$ of size $N$, Size of minibatch $\tau$, number of data passes $T$, privacy budget $(\epsilon, \delta)$, Lipschitz constant $L$, initial $\boldsymbol{\theta}_1$ and the logarithm of the prior probability density function $r$ of the model parameters $\boldsymbol{\theta}$.

**for** $t = 1 : [NT/\tau]$ **do**

1. Random sample a minibatch $S \subset [N]$ of size $\tau$.
2. Sample each coordinate of $\mathbf{z}_t$ iid from $\mathcal{N}\left(0, \frac{128NTL^2}{\tau\epsilon^2} \log\left(\frac{2.5NT}{\tau\delta}\right) \log(2/\delta)\eta_t^2 \vee \eta_t\right)$.
3. Update $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_t \left(\nabla r(\theta) + \frac{N}{\tau} \sum_{i \in S} \nabla \ell(\mathbf{x}_i | \boldsymbol{\theta})\right) + \mathbf{z}_t$.
4. Return $\boldsymbol{\theta}_{t+1}$ as a posterior sample (after a pre-defined burn-in period).
5. Increment $t \leftarrow t + 1$.

**end for**

## 5.1 Logistic regression

Consider a binary classification task. Our data consists of class labels $y_i$ and feature vector $\mathbf{x}_i$. We assume that there is a non-linear relationship between the class and the features. Recall the logistic regression model from Example 3.19.

$$\begin{aligned} \Pr(y_i|\mathbf{x}_i, \mathbf{w}) &= \sigma(y_i \mathbf{w}^T \mathbf{x}_i) \\ p(\mathbf{w}) &= \mathcal{N}(\mathbf{w}\,; \mathbf{w}_0, \mathbf{S}_0), \end{aligned}$$

where $\sigma(\cdot)$ is the sigmoid function given as

$$\sigma(x) = \frac{1}{1 + \exp(-x)}.$$

For our binary classification task, we want to separate the data into two classes according to the label $y_i$. Classification is done based on whether the probability $\Pr(y_i = 1|\mathbf{x}_i)$ exceeds 0.5. This probability is given as

$$\Pr(y_i = 1|\mathbf{x}_i) = \int \Pr(y_i = 1|\mathbf{x}_i, \mathbf{w}) p(\mathbf{w}|\mathbf{x}) \mathrm{d}\mathbf{w}. \tag{5.1}$$

In order to calculate the integral in (5.1) we need the posterior distribution for $\mathbf{w}$. As we stated in Section 3.2, direct and analytical variational posterior inferences are not possible. Using the approximate posterior $q(\mathbf{w})$ in integral (5.1) yields

$$\Pr(y_i = 1|\mathbf{x}_i) \approx \int \sigma(\mathbf{w}^T \mathbf{x}_i) q(\mathbf{w}) \mathrm{d}\mathbf{w}.$$

By the results proved in Bishop (2006, Section 4.5.2) we get

$$\Pr(y_i = 1|\mathbf{x}_i) \approx \sigma(\kappa(\sigma_a^2)\mu_a),$$

where

$$\begin{aligned} \kappa(\sigma^2) &= (1 + \pi\sigma^2/8)^{-1/2} \\ \sigma_a^2 &= \mathbf{x}_i^T \Sigma_{q(\mathbf{w})} \mathbf{x}_i \\ \mu_a &= \boldsymbol{\mu}_{q(\mathbf{w})}^T \mathbf{x}_i. \end{aligned}$$

We test DPVI on two different datasets, Abalone and Adult (Lichman, 2013). The Abalone dataset consists of 4177 samples with 8 attributes. Attributes are Sex, Length, Diameter Height, Whole weight, Shucked weight, Viscera weight and Shell. Besides these

attributes, the dataset also has an attribute Rings. We use our binary classifier to determine whether the abalones have more or less than 10 rings. Before the training, the whole dataset is divided into 80% training set and 20% test set. The dataset is normalized by subtracting feature mean and dividing by feature standard deviation.

The Adult dataset consists of 48842 samples with 14 attributes. Class labels indicate whether the annual income of an individual exceeds \$50K. We divide the dataset into training and test set and normalize it same as we did with the Abalone dataset.

Because the DP-SGLD method relies on drawing samples from the approximate posterior, the predictive probability $\Pr(y_i = 1|\mathbf{x}_i)$ was computed as

$$\Pr(y_i|\mathbf{x}_i) = \int \Pr(y_i = 1|\mathbf{x}_i, \mathbf{w}) q_{DP-SGLD}(\mathbf{w}) \mathrm{d}\mathbf{w} \approx (1/L) \sum_{l=1}^{L} \sigma(\mathbf{w}_l^T \mathbf{x}_i), \tag{5.2}$$

where $\mathbf{w}_l$ are the posterior draws of DP-SGLD after the burn-in period. In Figure 5.1 we see that DPVI outperforms the DP-SGLD method and reaches the non-private level with relatively small epsilon values. The $\delta$ is set to 0.001 in these figures. For the Abalone dataset we used $q = 0.05$ which corresponds to minibatches of size 167 and for the Adult dataset we used $q = 0.005$ corresponding to minibatches of size 195. On the Abalone data we ran the DPVI algorithm for 1000 iterations and with the Adult data for 2000 iterations. Presented results are obtained using clipping threshold 5 for the Abalone and 75 for the Adult task.

As we mentioned before, DPVI has many parameters that affect the performance. In Figure 5.2 we see how altering the subsample size affects the classification accuracy. We kept the number of iterations fixed and calculated the privacy cost for each sampling ratio $q$ using the moments accountant. We used five different noise levels to show how changing $\epsilon$ affects the classification accuracy. Because we used a fixed number of iterations for each $q$ value, the $\epsilon$ interval for each $q$ value changes. From the Abalone figure we see that $q = 0.01$, which means $\sim 41$ samples per iteration, gives decent accuracy even with small epsilon values, but is too small to overcome the effect of noise even with $\epsilon_{tot} = 1$. The largest subsampling ratio $q = 0.1$ on the other hand yields classification accuracy close to non private version, but the privacy guarantee is hurt because of smaller effect of privacy amplification. We observe the same in the experiment on the Adult data set.

In Figure 5.3 we see how altering the clipping threshold will affect the classification accuracy. The noise level of the DPVI algorithm depends on $c_t$ which explains why smaller $c_t$ values in the Abalone experiment at some smaller values of $\epsilon$ yield better accuracies than larger values. On the other hand it is obvious that allowing gradients to have larger norms improves the performance of learning. In the Adult experiment we see that we get almost the same classification accuracy with all $c_t$ values greater than 10.0. This could

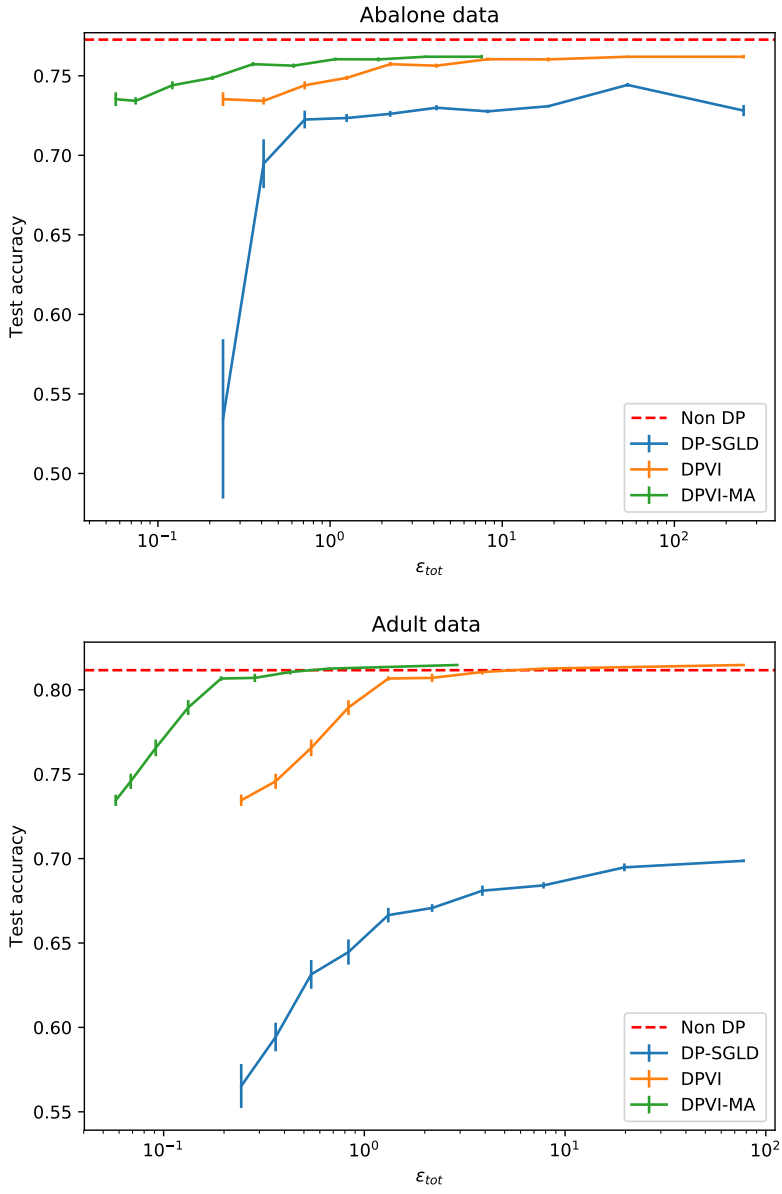


Figure 5.1: Comparison of binary classification accuracies on the Abalone data set (top) and the Adult data set (bottom). The figure shows test set classification accuracies of non-private logistic regression, two variants of the DPVI with the moments accountant (DPVI-MA) and advanced composition accounting (DPVI) and the DP-SGLD. The curve shows the mean of 10 runs of both algorithms with error bars denoting the standard error of the mean.

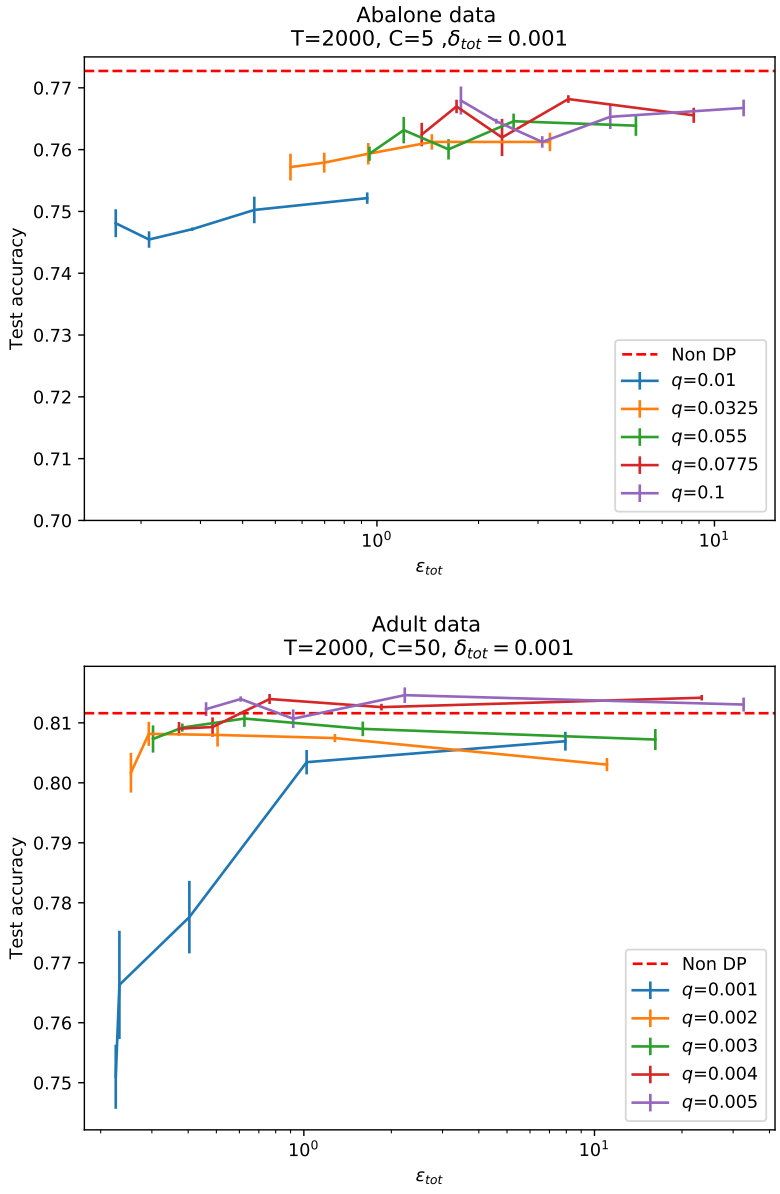


Figure 5.2: Accuracy vs. total $\epsilon$ in the Abalone (top) and Adult (bottom) datasets with several data subsampling ratios $q$ in DPVI with the moments accountant. The curve shows the mean of 10 runs of the DPVI algorithm with error bars denoting the standard error of the mean. Note that the $y$-axis scale covers a much smaller range than in Fig. 5.1.

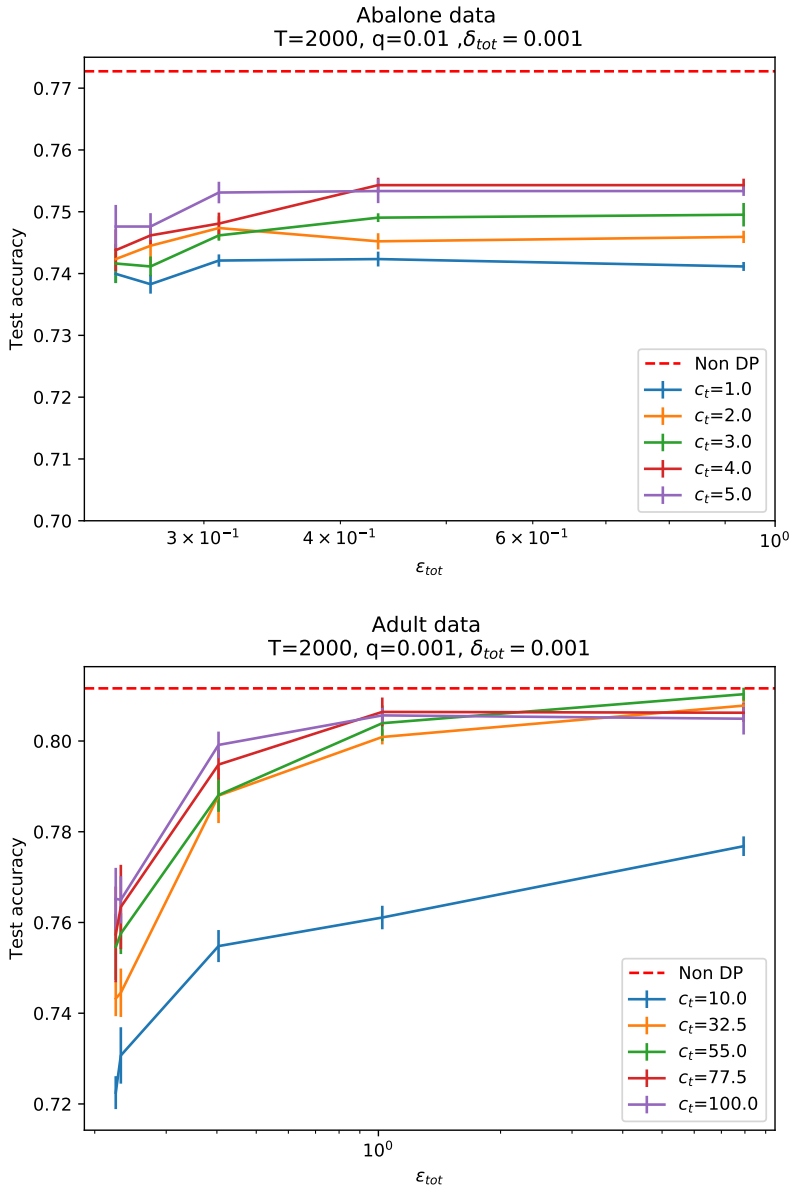


Figure 5.3: Accuracy vs. total $\epsilon$ in the Abalone (top) and the Adult (bottom) datasets with several gradient clipping threshold $c_t$ values in DPVI with the moments accountant. The curve shows the mean of 10 runs of the DP algorithm with error bars denoting the standard error of the mean. Note that the $y$-axis scale covers a much smaller range than in Fig. 5.1.

suggest that given large enough $\epsilon$, increasing the clipping threshold does not improve the learning dramatically after some value.

## 5.2 Gaussian mixture model

A mixture model is a probabilistic model to represent the data as a collection of samples from several underlying models. The Gaussian mixture model is a mixture of (multi)normally distributed random variables.

For a fixed number $K$ of mixture components our model is

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{Dir}(\alpha) \\ k &\sim \text{Cat}(\pi) \\ \mu^{(k)} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \tau^{(k)} &\sim \text{Inv-Gamma}(1, 1). \end{aligned}$$

We want to avoid inference of latent variables, in this case indicator variables $k$ denoting the mixture component responsible of producing each sample, because that would make the privacy preserving inference more complicated, we will discuss more of this in Chapter 6. Instead of augmenting the model with aforementioned latent variables we perform inference directly on the marginal likelihood. Marginalizing the latent variable $k$ from our model yields the following likelihood

$$p(\mathbf{x}_i|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\tau}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x}_i; \mu^k, \tau^{(k)}\mathbf{I}).$$

The posterior approximation $q(\pi, \mu, \tau) = q(\pi)q(\mu)q(\tau)$ is fully factorised. $q(\pi)$ is parametrised using the softmax transformation from a diagonal covariance Gaussian while $q(\mu)$ is Gaussian with a diagonal covariance and $q(\tau)$ is log-normal with a diagonal covariance.

The synthetic data used in experiments was drawn from the mixture of five spherical multivariate Gaussian distributions with means $[0, 0], [\pm 2, \pm 2]$ and covariance matrices $0.5\mathbf{I}$. Similar data has been used previously by Honkela et al. (2010) and Hensman et al. (2012). We used 1000 samples from this mixture for training the model and 100 samples to test the performance. We used both DPVI and DP-SGLD on this data. Performance comparison was done by computing the predictive likelihoods for both algorithms with several different epsilon values. We also show an example of the posterior distribution that we learn from the above mixture model by using both DPVI and DP-SGLD.

Figure 5.5 shows the results of both DPVI and DP-SGLD algortihms. Green dots represent the simulated data. Black dots represent the means of Gaussians and black

spheres the covariance structure of each Gaussian, which in this case is spherical. The other colored dots represent the posterior means we learn by using these private algorithms. The covariance structure of each component is shown in a sphere with the same color as the mean. We can see that DPVI finds one of the means almost exactly and 3 means out of 5 relatively well. The DP-SGLD algortihm on the other hand learns 2 of the 5 means with very good accuracy but for some reason misses the (-2,2) and (-2,-2) centered Gaussians almost completely.

In Figure 5.4 we can see that the DPVI algorithm performs almost as well as the non-private version of DPVI even with relatively small epsilon values. The reason why DPVI outperforms the DP-SGLD in predictive likelihood comparison could easily be explained by the behaviour we saw in Figure 5.5. The DP-SGLD algorithm for some reason tends to miss some of the mixture componens as well as the variances given by it tend to be small. Therefore it is obvious that if we cover a wider range of the support of our mixture model, we get better prediction likelihoods.

In the experiments, both DP-SGLD and DPVI used $q = 0.03$. The DP-SGLD algortihm was run for 150 iterations, whereas the DPVI was run for 1000 iterations. The gradient clipping threshold for DPVI was set to $c_t = 1.0$. For DPVI predictive likelihood was approximated by Monte-Carlo integration using samples from the learned approximate posterior. For the DP-SGLD, predictive likelihood was approximated using 100 samples after a burn-in period for this sampler as Wang et al. (2015) suggest. Non-private results were obtained by setting $\sigma = 0$ in DPVI, using $q = 0.01$ and running the algorithm for 2000 iterations.

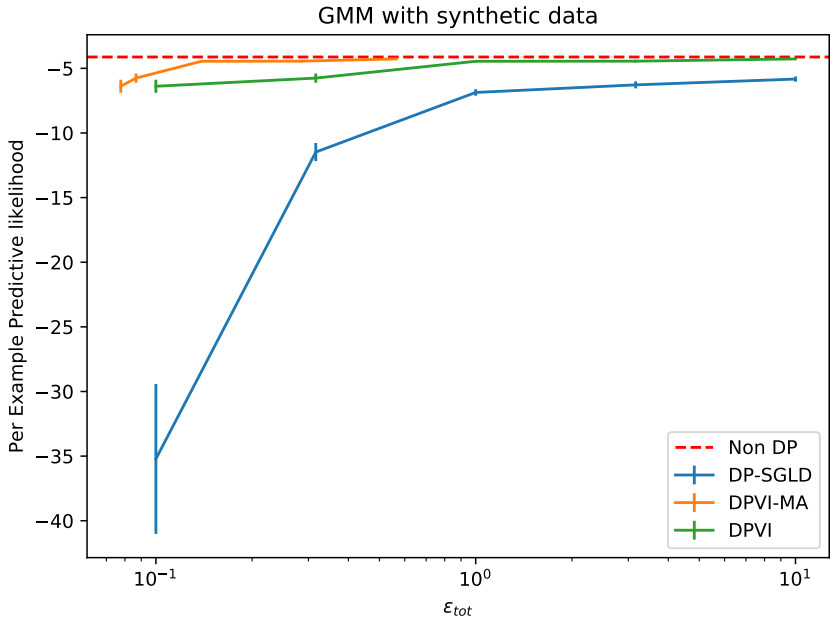


Figure 5.4: Per example predictive likelihood vs. $\epsilon$. For both the DP-SGLD and the DPVI the curves show the mean of 10 runs of the algorithm with error bars denoting the standard error of the mean.

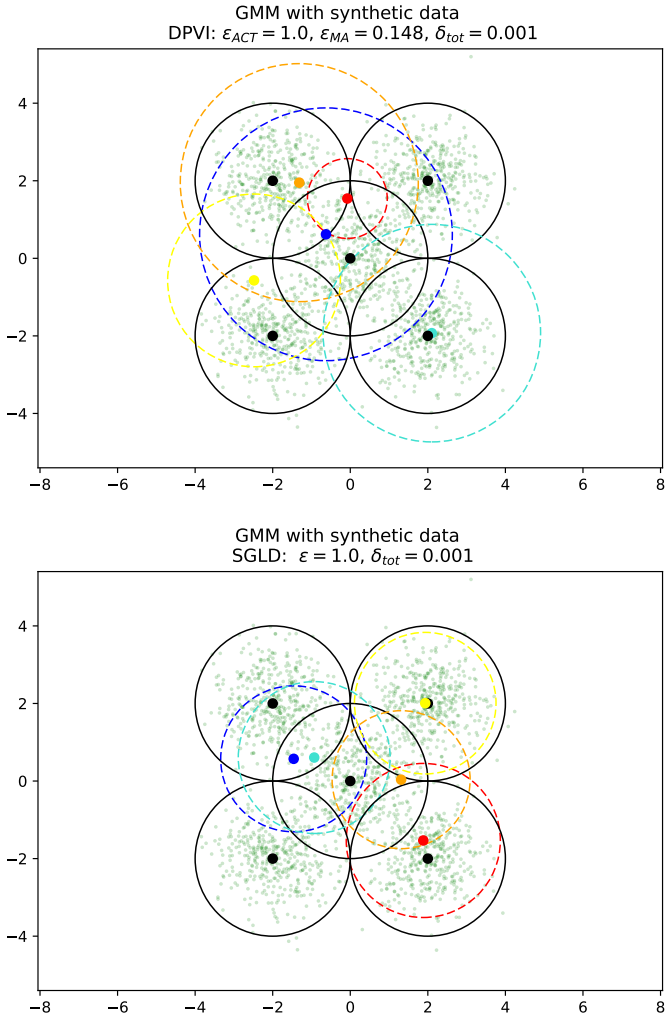


Figure 5.5: Approximate posterior predictive distribution for the Gaussian mixture model learned with DPVI (top) and DP-SGLD (bottom). The DP-SGLD distribution is formed as an average over the last 100 samples from the algorithm.

# Chapter 6

# Discussion

The DPVI method proposed in this thesis provides a general framework for privacy preserving variational inference. Because of the generality of DSVI and for the simplicity of the automated version of it, the ADVI, our method could be applied to a wide range of models. For example there is ongoing work on differentially private datasharing.

While DPVI performed well in our experiments there are still weaknesses that need to be studied in future work. As we briefly mentioned in the Gaussian mixture model experiment in Section 5, models with latent variables cause trouble for privacy. The problem arises from the concept of differential privacy. This is simply because usually there are as many latent variables as there are datapoints. Because differential privacy provides indistinguishability between the individuals in the dataset it is easy to see that if some variables are individual specific it is difficult to perturb them with strict enough privacy guarantees and still maintain a good performance in learning. Perturbing these kind of gradients would add huge amounts of noise to the latent variable updates. The problem with latent variables can be easily avoided if our model allows us to integrate out the latent variables. Another possible way to overcome such difficulty is to let a trusted curator update the latent variables given the differentially private global variables. In this setting the posterior distribution of the latent variables would never be released but it would still be part of the optimization of the global variable posterior distribution.

Another of the weaknesses is the number of parameters to be tuned. In Chapter 4 we gave some heuristics on how to tune the parameters, but in the end the only way to really tune these parameters is by experimenting on different choices. In our experiments we saw that even though the parameter tuning affects the results, the range of parameter values that provide good performance is wide. Another problem is that because we can only release the perturbed gradients, in order to maintain privacy guarantees, we cannot check the convergence of the SGD. The clipping scheme that is applied to bound the sensitivity of each individual gradient also causes trouble. One could think that upon convergence

the effect of clipping would disappear as the total gradient would tend to zero. However this is not true since the clipping scheme is applied to individual gradients that typically do not disappear even at convergence. Therefore it is possible that clipping will change the stationary points of the SGD algorithm.

While differential privacy has gained quite a lot attention in recent years, especially in the theoretical point of view, it still has not been widely applied in practice. However, recently big corporations such as Apple (Pease and Freudiger, 2016) and Google (Eland, 2015) have chosen to adopt differential privacy into some of their systems. Differential privacy has also gained attention from public authorities. The United States Census Bureau has recently awarded cooperative agreement for differential privacy research (U.S. Census Bureau, 2016). In addition, the new Europe Union wide data protection legislation GDPR (European Parliament, 2016), that will take effect in 2018 is going to change the ways companies and institutions can store data and what they can release based on the data. It is fair to say that in future differential privacy could be the industry standard for privacy protection and therefore there is demand for such frameworks as DPVI that are easy to implement and provide good performance.

# Chapter 7

# Conclusions

In this thesis we have established a method for privacy preserving variational inference. We combined modern techniques from variational Bayesian methods with the latest results on differential privacy. The main theorems and techniques used in order to establish DPVI are presented in this work. The proposed method is essentially a privacy preserving extension of the popular DSVI method and therefore generalizes for a wide range of models.

According to our experiments on two very different models, the DPVI algorithm performs with accuracy close to non-private version of the DSVI. Our method was also compared against the state-of-the-art method for privacy preserving posterior inference the DP-SGLD. Our experiments show that DPVI outperforms DP-SGLD in both of the learning tasks. For DPVI we have shown the results for two different ways of composing privacy cost, the state-of-the-art moments accountant and the amplified version of the advanced composition theorem.

# Bibliography


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`.

P Baldi, P Sadowski, and D Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5, 2014.

C M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pages 289–296, USA, 2008. Curran Associates Inc. ISBN 978-1-6056-0-949-2. URL `http://dl.acm.org/citation.cfm?id=2981780.2981817`.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *J. Mach. Learn. Res.*, 12:1069–1109, July 2011. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1953048.2021036`.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. Wiley-interscience, 2012.

Christos Dimitrakakis, Blaine Nelson, Aikaterini Mitrokotsa, and Benjamin I. P. Rubinstein. Robust and private Bayesian inference. In *ALT 2014*, volume 8776 of *Lecture Notes in Computer Science*, pages 291–305. Springer Science + Business Media, 2014.

John Duchi. Derivations for linear algebra and optimization. *Berkeley, California*, 2007.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1953048.2021068`.

C. Dwork, G. N. Rothblum, and S. Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, Oct 2010. doi: 10.1109/FOCS.2010.12.

Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052, pages 1–12, Venice, Italy, July 2006. Springer Verlag. ISBN 3-540-35907-9. URL `https://www.microsoft.com/en-us/research/publication/differential-privacy/`.

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`.

Andrew Eland. Tackling urban mobility with technology. https://europe.googleblog.com/2015/11/tackling-urban-mobility-with-technology.html, 2015. Accessed: 2017-08-01.

Council of the European Union European Parliament. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance). http://data.europa.eu/eli/reg/2016/679/oj, 2016. Accessed: 2017-08-01.

James Foulds, Joseph Geumlek, Max Welling, and Kamalika Chaudhuri. On the theory and practice of privacy-preserving Bayesian data analysis. In *Proc. 32nd Conf. on Uncertainty in Artificial Intelligence (UAI 2016)*, 2016.

James Hensman, Magnus Rattray, and Neil D. Lawrence. Fast variational inference in the conjugate exponential family. In *Advances in Neural Information Processing Systems 25*, pages 2897–2905. 2012.

Antti Honkela, Tapani Raiko, Mikael Kuusela, Matti Tornio, and Juha Karhunen. Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes. *J Mach Learn Res*, 11:3235–3268, Nov 2010.

Antti Honkela, Mrinal Das, Arttu Nieminen, Onur Dikmen, and Samuel Kaski. Efficient differentially private learning improves drug sensitivity prediction. 2016. arXiv:1606.02109.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *Proceedings of the 32Nd International Conference on International*

*Conference on Machine Learning - Volume 37*, ICML'15, pages 1376–1385. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045265.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL http://arxiv.org/abs/1312.6114.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M. Blei. Automatic differentiation variational inference. *J Mach Learn Res*, 18(14):1–45, 2017.

Ninghui Li, Wahbeh Qardaji, and Dong Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, pages 32–33, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1648-4. doi: 10.1145/2414456.2414474. URL http://doi.acm.org/10.1145/2414456.2414474.

M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

Mijung Park, James Foulds, Kamalika Chaudhuri, and Max Welling. Variational Bayes in private settings (VIPS). 2016. arXiv:1611.00340 [stat.ML].

Jessie Pease and Julien Freudiger. Engineering privacy for your users. https://developer.apple.com/videos/play/wwdc2016/709, 2016. Accessed: 2017-08-01.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951. doi: 10.1214/aoms/1177729586. URL http://dx.doi.org/10.1214/aoms/1177729586.

Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, 1998.

Jonathan L Ticknor. A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, 40(14):5501–5506, 2013.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *Proc. 31st Int. Conf. Mach. Learn. (ICML 2014)*, pages 1971–1979, 2014. URL http://jmlr.org/proceedings/papers/v32/titsias14.pdf.

Michalis K. Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1971–II–1980. JMLR.org, 2014. URL http://dl.acm.org/citation.cfm?id=3044805.3045112.

U.S. Census Bureau. Census bureau awards two cooperative agreements. https://www.census.gov/newsroom/press-releases/2016/cb16-tps141.html, 2016. Accessed: 2017-08-22.

Eric Vittinghoff, David V Glidden, Stephen C Shiboski, and Charles E McCulloch. *Regression methods in biostatistics: linear, logistic, survival, and repeated measures models*. Springer Science & Business Media, 2011.

Joshua T Vogelstein, Youngser Park, Tomoko Ohyama, Rex A Kerr, James W Truman, Carey E Priebe, and Marta Zlatic. Discovery of brainwide neural-behavioral maps via multiscale unsupervised structure learning. *Science*, 344(6182):386–392, 2014.

Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola. Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In *Proc. 32nd Int. Conf. Mach. Learn. (ICML 2015)*, pages 2493–2502, 2015.

O. Williams and F. McSherry. Probabilistic inference and differential privacy. In *Adv. Neural Inf. Process. Syst. 23*, 2010.

J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. PrivBayes: Private data release via Bayesian networks. In *SIGMOD '14*, pages 1423–1434, 2014.

Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *Proc. VLDB Endow.*, 5 (11):1364–1375, July 2012. ISSN 2150-8097. doi: 10.14778/2350229.2350253. URL `http://dx.doi.org/10.14778/2350229.2350253`.

Zuhe Zhang, Benjamin Rubinstein, and Christos Dimitrakakis. On the differential privacy of Bayesian inference. In *Proc. Conf. AAAI Artif. Intell. 2016*, 2016.