Differentially Private Variational Inference for Non-conjugate Models

Joonas Jälkö **Onur Dikmen** **Antti Honkela***
Helsinki Institute for Information Technology (HIIT), Department of Computer Science,
University of Helsinki, Finland
{joonas.jalko, onur.dikmen, antti.honkela}@helsinki.fi

Abstract

Many machine learning applications are based on data collected from people, such as their tastes and behaviour as well as biological traits and genetic data. Regardless of how important the application might be, one has to make sure individuals' identities or the privacy of the data are not compromised in the analysis. Differential privacy constitutes a powerful framework that prevents breaching of data subject privacy from the output of a computation. Differentially private versions of many important Bayesian inference methods have been proposed, but there is a lack of an efficient unified approach applicable to arbitrary models. In this contribution, we propose a differentially private variational inference method with a very wide applicability. It is built on top of doubly stochastic variational inference, a recent advance which provides a variational solution to a large class of models. We add differential privacy into doubly stochastic variational inference by clipping and perturbing the gradients. The algorithm is made more efficient through privacy amplification from subsampling. We demonstrate the method can reach an accuracy close to non-private level under reasonably strong privacy guarantees, clearly improving over previous sampling-based alternatives especially in the strong privacy regime.

*AH is also with the Department of Mathematics and Statistics and Department of Public Health, University of Helsinki.

1 INTRODUCTION

Using more data usually leads to better generalisation and accuracy in machine learning. With more people getting more tightly involved in the ubiquitous data collection, privacy concerns related to the data are becoming more important. People will be much more willing to contribute their data if they can be sure that the privacy of their data can be protected.

Differential privacy (DP) (Dwork et al., 2006; Dwork and Roth, 2014) is a strong framework with strict privacy guarantees against attacks from adversaries with side information. The main principle is that the output of an algorithm (such as a query or an estimator) should not change much if the data for one individual are modified or deleted. This can be accomplished through adding stochasticity at different levels of the estimation process, such as adding noise to data itself (input perturbation), changing the objective function to be optimised or how it is optimised (objective perturbation), releasing the estimates after adding noise (output perturbation) or by sampling from a distribution based on utility or goodness of the alternatives (exponential mechanism).

A lot of ground-breaking work has been done on privacy-preserving versions 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. Privacy-preserving Bayesian inference (e.g. (Williams and McSherry, 2010; Zhang et al., 2014)) has only recently started attracting more interest. The result of Dimitrakakis et al. (2014) showing that the posterior distribution is under certain assumptions differentially private is mathematically elegant, but does not lead to practically useful algorithms. 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. Methods based on perturbation of data

sufficient statistics (Zhang et al., 2016; Foulds et al., 2016; Honkela et al., 2016) are asymptotically efficient, but they are only applicable to exponential family models which limits their usefulness. 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 sufficient statistics.

General differentially private Bayesian inference can be realised most easily using the gradient perturbation mechanism. This was first proposed by Wang et al. (2015), who combine differential privacy by gradient perturbation with stochastic gradient Markov chain Monte Carlo (MCMC) sampling. This approach works in principle for arbitrary models, but because of the gradient perturbation mechanism each MCMC iteration will consume some privacy budget, hence severely limiting the number of iterations that can be run which can cause difficulties with the convergence of the sampler.

Our goal in this work is to apply the gradient perturbation mechanism to devise a generic differentially private variational inference method. 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 summary of the posterior. 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 inference 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 ascent. Here, we propose differentially private variational inference (DPVI) based on gradient clipping and perturbation as well as double stochasticity. 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 with a Gaussian mixture model.

2 BACKGROUND

2.1 DIFFERENTIAL PRIVACY

Differential privacy (DP) (Dwork et al., 2006; Dwork and Roth, 2014) is a framework that provides mathematical formulation for privacy that enables proving strong privacy guarantees.

Definition 1 (ϵ-Differential privacy)**.** *A randomised algorithm $\mathcal{A}$ is ϵ-differentially private if for all pairs of adjacent data sets, i.e., differing only in one data sample, x, x', and for all sets $S \subset im(\mathcal{A})$*

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

There are two different variants depending on which data sets are considered adjacent: in *unbounded DP* data sets x, x' are adjacent if x' can be obtained from x by adding or removing an entry, while in *bounded DP* x, x' are adjacent if they are of equal size and equal in all but one of their elements (Dwork and Roth, 2014). The definition is symmetric in x and x' which means that in practice the probabilities of obtaining a specific output from either algorithm need to be similar. The privacy parameter ϵ measures the strength of the guarantee with smaller values corresponding to stronger privacy.

ϵ-DP defined above, also known as *pure DP*, is sometimes too inflexible and a relaxed version called (ϵ, δ)-DP is often used instead. It is defined as follows:

Definition 2 ((ϵ, δ)-Differential privacy)**.** *A randomised algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all pairs of adjacent data sets x, x' and for every $S \subset im(\mathcal{A})$*

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

It can be shown that (ϵ, δ)-DP provides a probabilistic ϵ-DP guarantee with probability $1 - \delta$ (Dwork and Roth, 2014).

2.1.1 Gaussian mechanism

There are many possibilities how to make algorithm differentially private. In this paper we use *objective perturbation*. We use the *Gaussian mechanism* as our method for perturbation. Dwork and Roth (2014, Theorem 3.22) state that given query f with ℓ_2-sensitivity of $\Delta_2(f)$, releasing $f(x)+\eta$, where $\eta \sim N(0, \sigma^2)$, is (ϵ, δ)-DP when

$$\sigma^2 > 2\ln(1.25/\delta)\Delta_2^2(f)/\epsilon^2. \quad (1)$$

The important ℓ_2-sensitivity of a query is defined as:

Definition 3 (ℓ_2-sensitivity)**.** *Given two adjacent data sets x, x', ℓ_2-sensitivity of query f is*

$$\Delta_2(f) = \sup_{\substack{x,x' \\ ||x-x'||=1}} ||f(x) - f(x')||_2.$$

2.1.2 Composition theorems

One of the very useful features of DP compared to many other privacy formulations is that it provides a very natural way to study the privacy loss incurred by repeated use of the same data set. Using an algorithm on a data

set multiple times will weaken our privacy guarantee because of the potential of each application to leak more information. The DP variational inference algorithm proposed in this paper is iterative, so we need to use composition theorems to bound the total privacy loss.

The simplest basic composition Dwork and Roth (2014) shows that a k-fold composition of an (ϵ, δ)-DP algorithm provides $(k\epsilon, k\delta)$-DP. More generally releasing joint output of k algorithms $\mathcal{A}_i$ that are individually (ϵ_i, δ_i)-DP will be $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$-DP. Under pure ϵ-DP when $\delta_1 = \cdots = \delta_k = 0$ this is the best known composition that yields a pure DP algorithm.

Moving from the pure ϵ-DP to general (ϵ, δ)-DP allows a stronger result with a smaller ϵ at the expense of having a larger total δ on the composition. This trade-off is characterised by the Advanced composition theorem of Dwork and Roth (2014, Theorem 3.20), which becomes very useful when we need to use data multiple times

Theorem 1 (Advanced composition theorem)**.** *Given algorithm $\mathcal{A}$ that is (ϵ, δ)-DP and $\delta' > 0$, k-fold composition of algorithm $\mathcal{A}$ is $(\epsilon_{tot}, \delta_{tot})$-DP with*

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

$$\delta_{tot} = k\delta + \delta'. \quad (3)$$

The theorem states that with small loss in δ_{tot} and with small enough ϵ, we can provide more strict ϵ_{tot} than just summing the ϵ. This is obvious by looking at the first order expansion for small ϵ of

$$\epsilon_{tot} \approx \sqrt{2k \ln(1/\delta')}\epsilon + k\epsilon^2.$$

2.1.3 Privacy amplification

We use a stochastic gradient algorithm that uses subsampled data while learning, so we can make use of the amplifying effect of the subsampling on privacy. This *Privacy amplification theorem* (Li et al., 2012) states that if we run (ϵ, δ)-DP algorithm $\mathcal{A}$ on randomly sampled subset of data with uniform sampling probability $q > \delta$, privacy amplification theorem states that the subsampled algorithm is $(\epsilon_{amp}, \delta_{amp})$-DP with

$$\epsilon_{amp} = \min(\epsilon, \log(1 + q(e^\epsilon - 1))) \quad (4)$$

$$\delta_{amp} = q\delta, \quad (5)$$

assuming $\log(1 + q(e^\epsilon - 1)) < \epsilon$.

2.1.4 Moments accountant

The moments accountant proposed by Abadi et al. (2016) is a method to accumulate the privacy cost that provides a tighter bound for ϵ and δ than the previous composition approaches. The moments accountant incorporates both the composition over iterations and privacy amplification due to subsampling into a single bound given by the following Theorem.

Theorem 2. *There exist constants c_1 and c_2 so that given the sampling probability $q = L/N$ and the number of steps T, for any $\epsilon < c_1 q^2 T$, a DP stochastic gradient algorithm that clips the ℓ_2 norm of gradients to C and injects Gaussian noise with standard deviation $2C\sigma$ to the gradients, is (ϵ, δ)-DP for any $\delta > 0$ under bounded DP if we choose*

$$\sigma \geq c_2 \frac{q\sqrt{T \log(2/\delta)}}{\epsilon}. \quad (6)$$

Proof. Abadi et al. (2016) show that injecting gradient noise with standard deviation $C\sigma$ where σ satisfies the inequality (6) yields an $(\epsilon, \frac{1}{2}\delta)$-DP algorithm under unbounded DP. This implies that adding noise with standard deviation $2C\sigma$ yields an $(\frac{1}{2}\epsilon, \frac{1}{2}\delta)$-DP algorithm under unbounded DP.

This proves the theorem as any $(\frac{1}{2}\epsilon, \frac{1}{2}\delta)$ unbounded DP algorithm is an (ϵ, δ) bounded DP algorithm. This follows from the fact that the replacement of an element in the data set can be represented as a composition of one addition and one removal of an element. □

Similar bounds can also be derived using concentrated DP (Dwork and Rothblum, 2016; Bun and Steinke, 2016).

We use the implementation of Abadi et al. (2016) to compute the total ϵ privacy cost with a given δ-budget, standard deviation σ of noise applied in Gaussian mechanism and subsampling ratio q.

In our experiments we report results using both the advanced composition theorem with privacy amplification as well as the moments accountant.

2.2 VARIATIONAL BAYES

Variational Bayes (VB) methods (Jordan et al., 1999) provide a way to approximate the posterior distribution of latent variables in a model when the true posterior is intractable. True posterior $p(\boldsymbol{\theta}|\mathbf{x})$ is approximated with a variational distribution $q_{\boldsymbol{\xi}}(\boldsymbol{\theta})$ that has a simpler form than the posterior, obtained generally by removing some dependencies from the graphical model such as the fully-factorised form $q_{\boldsymbol{\xi}}(\boldsymbol{\theta}) = \prod_d q_{\boldsymbol{\xi}_d}(\theta_d)$. $\boldsymbol{\xi}$ are the variational parameters and their optimal values $\boldsymbol{\xi}^*$ are obtained through minimising the Kullback-Leibler (KL) divergence between $q_{\boldsymbol{\xi}}(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\mathbf{x})$. This is also equiv-

alent to maximising the *evidence lower bound* (ELBO)

$$\mathcal{L}(q_{\boldsymbol{\xi}}) = \int q_{\boldsymbol{\xi}}(\boldsymbol{\theta}) \ln \left(\frac{p(\mathcal{D}, \boldsymbol{\theta})}{q_{\boldsymbol{\xi}}(\boldsymbol{\theta})} \right)$$
$$= -\mathrm{KL}(q_{\boldsymbol{\xi}}(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta})) + \sum_{i=1}^{B} \langle \ln p(x_i|\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\xi}}(\boldsymbol{\theta})},$$

where $\langle \rangle_{q_{\boldsymbol{\xi}}(\boldsymbol{\theta})}$ is an expectation taken w.r.t $q_{\boldsymbol{\xi}}(\boldsymbol{\theta})$ and the observations $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are assumed to be exchangeable under our model.

When the model is in the conjugate exponential family (Ghahramani and Beal, 2001) and $q_{\boldsymbol{\xi}}(\boldsymbol{\theta})$ is factorised, the expectations that constitute $\mathcal{L}(q_{\boldsymbol{\xi}})$ are analytically available and each $\boldsymbol{\xi}_d$ is updated iteratively by fixed point iterations. Most popular applications of VB fall into this category, because handling of the more general case involves more approximations, such as defining another level of lower bound to the ELBO or estimating the expectations using Monte Carlo integration.

2.2.1 Doubly stochastic variational inference

An increasingly popular alternative approach is the doubly stochastic variational inference framework proposed by Titsias and Lázaro-Gredilla (2014). The framework is based on stochastic gradient optimisation of the ELBO. The expectation over $q_{\boldsymbol{\xi}}(\boldsymbol{\theta})$ is evaluated using Monte Carlo sampling. Exchanging the order of integration (expectation) and differentiation and using the reparametrisation trick to represent for example samples from a Gaussian approximation $q_{\boldsymbol{\xi}_i}(\boldsymbol{\theta}_i) = N(\boldsymbol{\theta}_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ as $\boldsymbol{\theta}_i = \boldsymbol{\mu}_i + \boldsymbol{\Sigma}_i^{1/2}\mathbf{z},\ \mathbf{z} \sim N(0, I)$, it is possible to obtain stochastic gradients of the ELBO which can be fed to a standard stochastic gradient ascent (SGA) optimisation algorithm.

For models with exchangeable observations, the ELBO objective can be broken down to a sum of terms for each observation:

$$\mathcal{L}(q_{\boldsymbol{\xi}}) = -\mathrm{KL}(q_{\boldsymbol{\xi}}(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta})) + \sum_{i=1}^{N} \langle \ln p(x_i|\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\xi}}(\boldsymbol{\theta})}$$
$$= \sum_{i=1}^{N} \left(\langle \ln p(x_i|\boldsymbol{\theta}) \rangle_{q_{\boldsymbol{\xi}}(\boldsymbol{\theta})} - \frac{1}{N} \mathrm{KL}(q_{\boldsymbol{\xi}}(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta})) \right)$$
$$=: \sum_{i=1}^{N} \mathcal{L}_i(q_{\boldsymbol{\xi}}).$$

This allows considering mini batches of data at each iteration to handle big data sets, which adds another level of stochasticity to the algorithm.

The recently proposed Automatic Derivation Variational Inference (ADVI) framework (Kucukelbir et al., 2017) unifies different classes of models through a transformation of variables and optimises the ELBO using stochastic gradient ascent (SGA). Constrained variables are transformed into unconstrained ones and their posterior is approximated by Gaussian variational distributions, which can be a product of independent Gaussians (mean-field) or larger multivariate Gaussians. Expectations in the gradients are approximated using Monte Carlo integration and the ELBO is optimised iteratively using SGA.

Algorithm 1 DPVI

Input: Data set $\mathcal{D}$, sampling probability q, number of iterations T, SGA step size η, Clipping threshold c_t and initial values $\boldsymbol{\xi}_0$.
for $t \in [T]$ **do**
  Pick random sample U from $\mathcal{D}$ with sampling probability q
  Calculate the gradient $g_t(x_i) = \nabla \mathcal{L}_i(q_{\boldsymbol{\xi}_t})$ for each $i \in U$
  Clip and sum gradients:
  $\tilde{g}_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{||g_t(x_i)||_2)}{c_t})$
  $\tilde{g}_t \leftarrow \sum_i \tilde{g}_t(x_i)$
  Add noise: $\tilde{g}_t \leftarrow \tilde{g}_t + \mathcal{N}(0, 4c_t^2\sigma^2\mathbf{I})$
  Update AdaGrad parameter. $G_t \leftarrow G_{t-1} + \tilde{g}_t^2$
  Ascent: $\boldsymbol{\xi}_t \leftarrow \boldsymbol{\xi}_{t-1} + \eta\tilde{g}_t/\sqrt{G_t}$
end for

3 DIFFERENTIALLY-PRIVATE VARIATIONAL INFERENCE

Differentially-private variational inference (DPVI) is based on clipping of the contributions of individual data samples to the gradient, $g_t(x_i) = \nabla \mathcal{L}_i(q_{\boldsymbol{\xi}_t})$, at each iteration t of the stochastic optimisation process and perturbing the total gradient. Each $g_t(x_i)$ is clipped in order to calculate gradient sensitivity. Gradient contributions from all data samples in the mini batch are summed and perturbed with Gaussian noise $\mathcal{N}(0, 4c_t^2\sigma^2\mathbf{I})$. The algorithm is presented in Algorithm 1. The algorithm is very similar to the one used for deep learning by Abadi et al. (2016). DPVI can be easily implemented using automatic differentiation software such as Autograd, or incorporated even more easily into automatic inference engines, such as the ADVI implementations in PyMC3 (Salvatier et al., 2016) or Edward (Tran et al., 2017) which also provide subsampling. Python implementation of DPVI based on Autograd can be found at `https://github.com/DPBayes/DPVI-code`.

The sampling frequency q for subsampling within the data set, total number of iterations T and the variance σ^2 of Gaussian noise are important design parameters that

determine the privacy cost. c_t is chosen before learning, and does not need to be constant. After clipping $||g_t(x_i)||_2 \leq c_t$, $\forall i \in U$. Clipping gradients too much will affect accuracy, but on the other hand large clipping threshold will cause large amount of noise to sum of gradients. Parameter q determines how large subsample of the training data we use to for gradient ascent. Small q values enable privacy amplification but may need a need larger T. For a very small q when the mini batches consist of just a few samples, the added noise will dominate over the gradient signal and the optimisation will fail. While in our experiments q was fixed, we could also alter the q during iteration. Next, we show that Algorithm 1 is differentially private and explain how to calculate its privacy budget.

Theorem 3. *Algorithm 1 is (ϵ, δ)-differentially private.*

Proof. The algorithm applies Gaussian mechanism on the total gradient of a subsample of the dataset at every iteration t. If each iteration is $(\epsilon_{\text{subs}}, \delta_{\text{subs}})$-DP on the subsample, then it is $(\epsilon_{\text{iter}}, \delta_{\text{iter}})$-DP w.r.t. the whole data set due to privacy amplification theorem with

$$\epsilon_{\text{iter}} = \log(1 + q\,(e^{\epsilon_{\text{subs}}} - 1))\,. \tag{7}$$

Overall algorithm is (ϵ, δ)-DP with

$$\epsilon = \sqrt{2T\ln(1/\delta')}\epsilon_{\text{iter}} + T\epsilon_{\text{iter}}(e^{\epsilon_{\text{iter}}} - 1)\,, \tag{8}$$

due to T-fold composition of $(\epsilon_{\text{iter}}, \delta_{\text{iter}})$-DP iterations. Determining δ, δ' and σ before running the algorithm, leads to $\delta_{\text{iter}} = (\delta - \delta')/T$ and $\delta_{\text{subs}} = \delta_{\text{iter}}/q$. Then, evaluation of the total privacy budget in (8) is straightforward using (7) with

$$\epsilon_{\text{subs}} = \sqrt{2\ln(1.25/(\delta_{\text{subs}}))}/\sigma\,.$$

This is simpler to show using the moments accountant instead of advanced composition and privacy amplification theorems. Given δ and σ, Algorithm 1 is readily (ϵ, δ)-DP, where ϵ (is the smallest value that) satisfies the conditions in Theorem 2. □

3.1 MODELS WITH LATENT VARIABLES

The simple approach in Algorithm 1 will not work well for models with latent variables. This is because the main gradient contributions to latent variables come from only a single data point, and the amount of noise that would need to be injected to mask the contribution of this point as needed by DP would make the gradient effectively useless.

One way to deal with the problem is to take the EM algorithm view (Dempster et al., 1977) of latent variables as a hidden part of a larger complete data set and apply the DP protection to summaries computed from the complete data set. In this approach, which was also used by Park et al. (2016), no noise would be injected to the updates of the latent variables but the latent variables would never be released.

An alternative potentially easier way to avoid this problem is to marginalise out the latent variables if the model allows this. As the DPVI framework works for arbitrary likelihoods we can easily perform inference even for complicated marginalised likelihoods. This is a clear advantage over the VIPS framework of Park et al. (2016) which requires conjugate exponential family models.

3.2 SELECTING THE ALGORITHM HYPERPARAMETERS

The DPVI algorithm depends on a number of parameters, the most important of which are the gradient clipping threshold c_t, the data subsampling ratio q and the number of iterations T. Together these define the total privacy cost of the algorithm, but it is not obvious how to find the optimal combination of these under a fixed privacy budget. Unfortunately the standard machine learning hyperparameter adaptation approach of optimising the performance on a validation set is not directly applicable, as every test run would consume some of the privacy budget. Developing good heuristics for parameter tuning is thus important for practical application of the method.

Out of these parameters, the subsampling ratio q seems easiest to interpret. The gradient that is perturbed in Algorithm 1 is a sum over qN samples in the mini batch. Similarly the standard deviation of the noise injected with the moments accountant in Eq. (6) scales linearly with q. Thus the signal-to-noise ratio for the gradients will be independent of q and q can be chosen to minimise the number of iterations T.

The number of iterations T is potentially more difficult to determine as it needs to be sufficient but not too large. The moments accountant is somewhat forgiving here as its privacy cost increases only in proportion to $\sqrt{T}$. In practice one may need to simply pick T believed to be sufficiently large and hope for the best. Poor results in the end likely indicate that the number of samples in the data set may be insufficient for good results at the given level of privacy.

The gradient clipping threshold c_t may be the most difficult parameter to tune as that depends strongly on the details of the model. Fortunately our results do not seem overly sensitive to using the precisely optimal value of c_t. Developing good heuristics for choosing c_t is an important objective for future research. Still, the same problem

is shared by every DL method based on gradient perturbation including the deep learning work of Abadi et al. (2016) and the DP stochastic gradient MCMC methods of Wang et al. (2015). In the case of stochastic gradient MCMC this comes up through selecting a bound on the parameters to bound the Lipschitz constant appearing in the algorithm. A global Lipschitz constant for the ELBO would naturally translate to a c_t guaranteed not to distort the gradients, but as noted by Abadi et al. (2016), it may actually be good to clip the gradients to make the method more robust against outliers.

4 EXPERIMENTS

4.1 LOGISTIC REGRESSION

We tested DPVI with two different learning tasks. Lets first consider model of logistic regression using the Abalone and Adult data sets from the UCI Machine Learning Repository (Lichman, 2013) for the binary classification task. Our model is:

$$P(y|\mathbf{x}, \mathbf{w}) = \sigma(y\mathbf{w}^T\mathbf{x})$$
$$p(\mathbf{w}) = N(\mathbf{w}; \mathbf{w}_0, \mathbf{S}_0),$$

where $\sigma(x) = 1/(1 + \exp(-x))$.

For Abalone, individuals were divided into two classes based on whether individual had less or more than 10 rings. The data set consisted of 4177 samples with 8 attributes. We learned a posterior approximation for $\mathbf{w}$ using ADVI with SGA using Adagrad optimiser (Duchi et al., 2011) and sampling ratio $q = 0.02$. The posterior approximation $q(\mathbf{w})$ was Gaussian with a diagonal covariance. Classification was done using an additional Laplace approximation. Before training, features of the data set were normalised by subtracting feature mean and dividing by feature standard deviation. Training was done with 80% of data.

The other classification dataset "Adult" that we used with logistic regression consisted of 48842 samples with 14 attributes. Our classification task was to predict whether or not an individual's annual income exceeded \$50K. The data were preprocessed similarly as in Abalone: we subtracted the feature mean and divided by the standard deviation of each feature. We again used 80% of the data for training the model.

We first compared the classification accuracy of models learned using two variants of DPVI with the moments accountant and advanced composition accounting as well as DP-SGLD of Wang et al. (2015). The classification results for Abalone and Adult are shown in Fig. 1. We used $q = 0.05$ in Abalone corresponding to mini batches of 167 points and $q = 0.005$ in Adult corresponding to mini batches of 195 points. With Abalone the algorithm was run for 1000 iterations and with Adult for 2000 iterations. Clipping threshold were 5 for Abalone and 75 for Adult. Both results clearly show that even under comparable advanced composition accounting used by DP-SGLD, DPVI consistently yields significantly higher classification accuracy at a comparable level of privacy. Using the moments accountant further helps in obtaining even more accurate results at comparable level of privacy. DPVI with the moments accountant can reach classification accuracy very close to the non-private level already for $\epsilon < 0.5$ for both data sets.

4.1.1 The effect of algorithm hyperparameters

We further tested how changing the different hyperparameters of the algorithm discussed in Sec. 3.2 affects the test set classification accuracy of models learned with DPVI with the moments accountant.

Fig. 2 shows the results when changing the data subsampling rate q. The result confirms the analysis of Sec. 3.2 that larger q tend to perform better than small q although there is a limit how small values of ϵ can be reached with a larger q.

Fig. 3 shows corresponding results when changing the gradient clipping threshold c_t. The results clearly show that too strong clipping can hurt the accuracy significantly. Once the clipping is sufficiently mild the differences between different options are far less dramatic.

4.2 GAUSSIAN MIXTURE MODEL

We also tested the performance of DPVI with a Gaussian mixture model. For K components our model is

$$\pi_k \sim \text{Dir}(\alpha)$$
$$\mu^{(k)} \sim \text{MVNormal}(\mathbf{0}, \mathbf{I})$$
$$\tau^{(k)} \sim \text{Inv-Gamma}(1, 1)$$

with the likelihood

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

Unlike standard variational inference that augments the model with indicator variables denoting the component responsible for generating each sample, we performed the inference directly on the mixture likelihood. This lets us avoid having to deal with latent variables that would otherwise make the DP inference more complicated.

The posterior approximation $q(\pi, \mu, \tau) = q(\pi)q(\mu)q(\tau)$ was fully factorised. $q(\pi)$ was parametrised using soft-

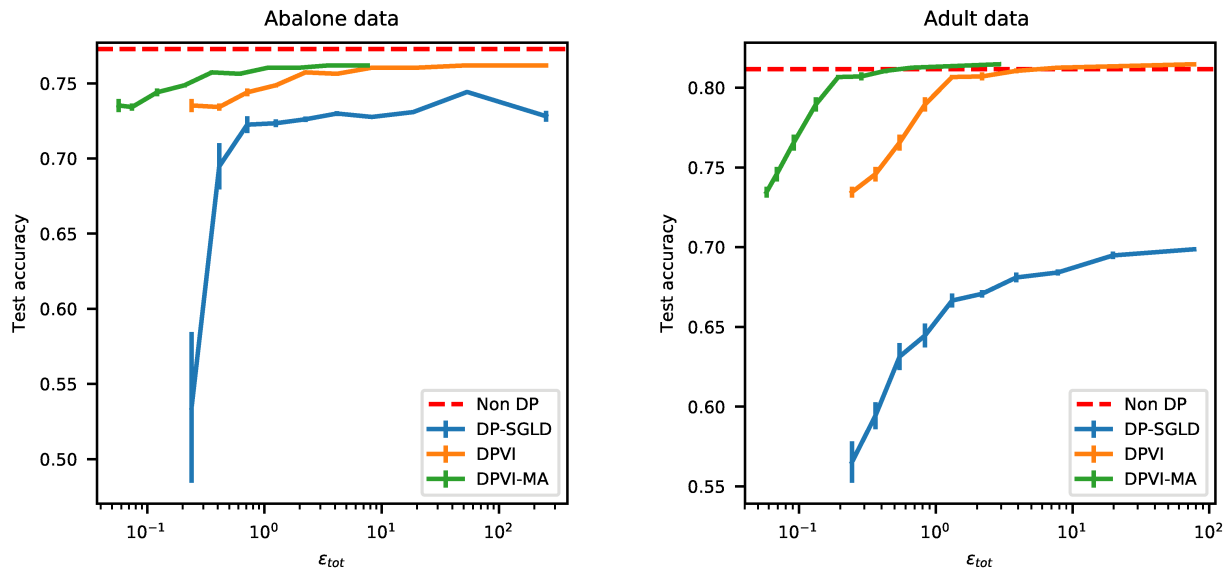

Figure 1: Comparison of binary classification accuracies using the Abalone data set (left) and the Adult data set (right). The figure shows test set classification accuracies of non-private logistic regression, two variants of DPVI with the moments accountant and advanced composition accounting and DP-SGLD of Wang et al. (2015). The curve shows the mean of 10 runs of both algorithms with error bars denoting the standard error of the mean.

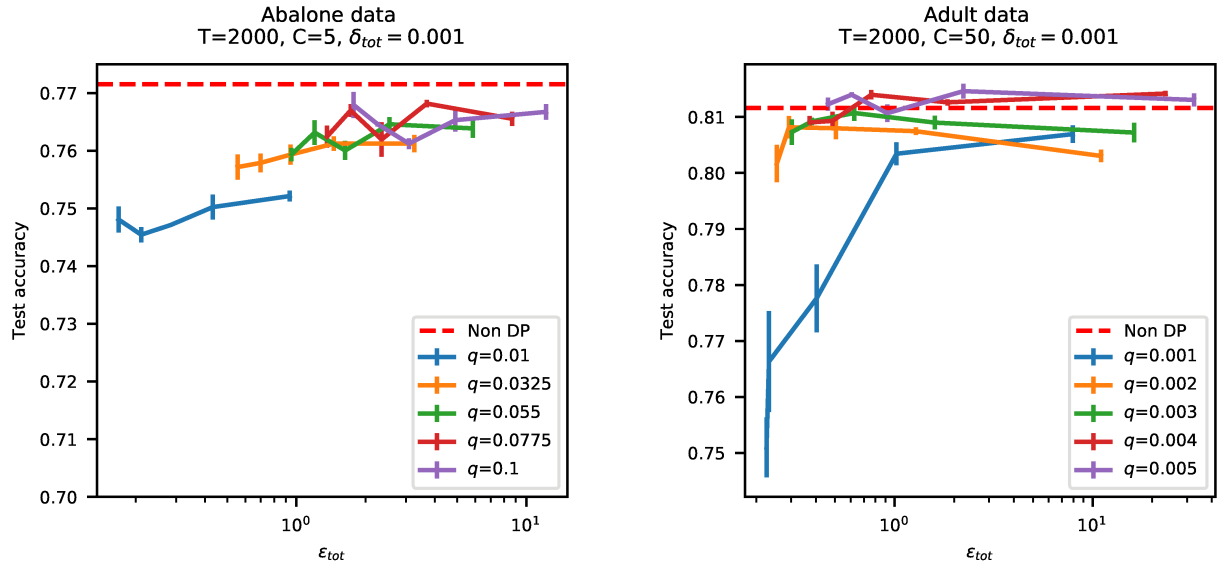

Figure 2: Accuracy vs. total ϵ in Abalone (left) and Adult (right) data sets with several data subsampling ratios q in DPVI with the moments accountant. The curve shows the mean of 10 runs of the DP algorithm with error bars denoting the standard error of the mean. Note that the y-axis scale covers a much smaller range than in Fig. 1.

max transformation from a diagonal covariance Gaussian while $q(\mu)$ was Gaussian with a diagonal covariance and $q(\tau)$ was log-normal with a diagonal covariance.

The synthetic data used in experiments was drawn from mixture of five spherical multivariate Gaussian distribution 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 2000 samples from this mixture for training the model and 100

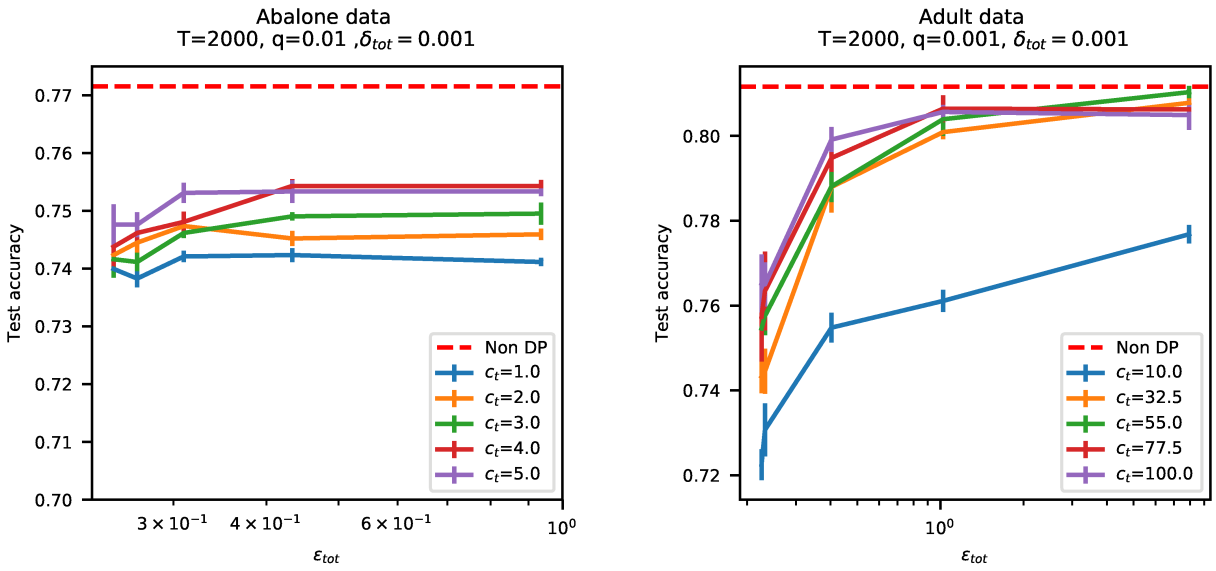

Figure 3: Accuracy vs. total ϵ in Abalone (left) and Adult (right) data sets with several gradient clipping threshold c_t values in DPVI with the moments accountant. The curve shows the mean of 10 runs of the DP algorithm with error bars denoting the standard error of the mean. Note that the y-axis scale covers a much smaller range than in Fig. 1.

samples to test the performance. We used both DPVI and DP-SGLD for this data. Performance comparison was done by computing the predictive likelihoods for both algorithms with several different epsilon values. We also show one example of approximate distribution that DPVI learns from above mixture model.

From Fig. 4 we can see that DPVI algorithm performs well compared to non-private version of DPVI even with relatively small epsilon values. We used $q = 0.003$ for DPVI and $q = 0.03$ for DP-SGLD. For DPVI algorithm number of iterations was 1000 and for DP-SGLD it was 150. Gradient clipping threshold for DPVI was set to $c_t = 1.0$. We used $\delta = 0.001$ in the predictive likelihood comparison. For DPVI predictive likelihood was approximated by Monte-Carlo integration using samples from the learned approximate posterior and for DP-SGLD by using the last 100 samples the algorithm produced. Non-private results were obtained by setting $\sigma = 0$ in DPVI, using $q = 0.003$ and running the DPVI algorithm for 2000 iterations.

Fig. 5 shows a visualisation of the mixture components learned by DPVI and DP-SGLD. Components inferred by DPVI appear much closer to ground truth than those from DP-SGLD.

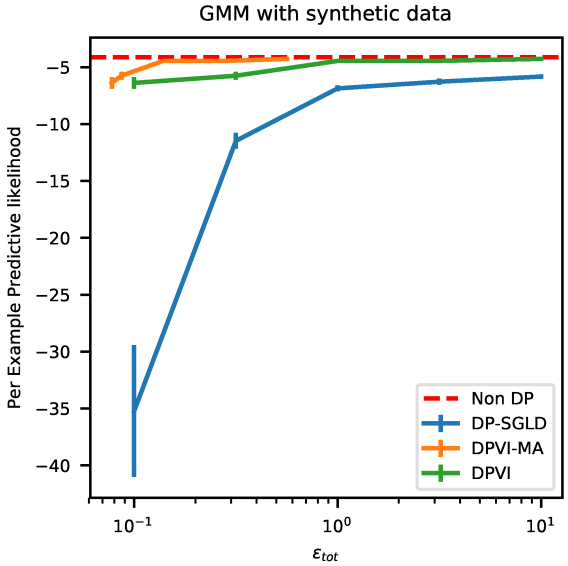

Figure 4: Per example predictive likelihood vs. ϵ. For both DP-SGLD and DPVI, lines show mean between 5 runs of algorithm with error bars denoting the standard error of mean.

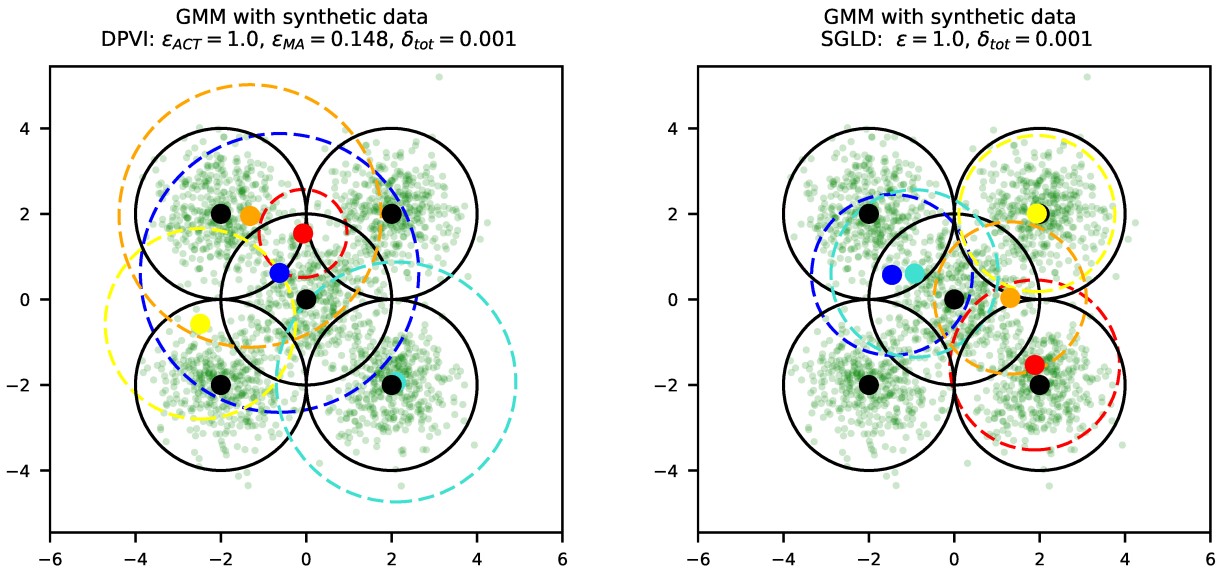

Figure 5: Approximate posterior predictive distribution for the Gaussian mixture model learned with DPVI (left) and DP-SGLD (right). The DP-SGLD distribution is formed as an average over the last 100 samples from the algorithm. Per example predictive log-likelihoods for DPVI and DP-SGLD in these experiments are -5.84 and -7.56 respectively. Predictive log-likelihoods were calculated using 100 test points.

5 DISCUSSION

Our results demonstrate that the proposed DPVI method has the potential to produce very accurate learning results, but this requires finding good values for algorithmic hyperparameters that unfortunately cannot be tuned using standard approaches without compromising the privacy. Finding good heuristics and default values for the hyperparameters is a very important avenue of future research.

It is tempting to think that the effect of gradient clipping would disappear as the algorithm converges and the total gradient becomes smaller. Unfortunately this is not true as the clipping is applied on the level of data point specific gradients which will typically not disappear even at convergence. This also means that aggressive clipping will change the stationary points of the SGA algorithm.

One way to make the problem easier to learn under DP is to simplify it for example through dimensionality reduction. This was noted for exponential family models by Honkela et al. (2016) but the same principle carries over to DPVI too. In DPVI, lower dimensional model typically has fewer parameters leading to a shorter parameter vector whose norm would thus be smaller, implying that smaller c_t is enough. This means that simpler posterior approximations such as Gaussians with a diagonal covariance may be better under DP while without the DP constraint an approximation with a full covariance would usually be better (see also Kucukelbir et al., 2017).

6 CONCLUSIONS

We have introduced the DPVI method that can deliver differentially private inference results with accuracy close to the non-private doubly stochastic variational inference and ADVI. The method can effectively harness the power of ADVI to deal with very general models instead of just conjugate exponential models and the option of using multivariate Gaussian posterior approximations for greater accuracy.

Acknowledgements

This work was funded by the Academy of Finland (Centre of Excellence COIN; and grants 278300, 259440 and 283107).

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proc. CCS 2016*, 2016. arXiv:1607.00133 [stat.ML].

M. Bun and T. Steinke. Concentrated differential pri-

vacy: Simplifications, extensions, and lower bounds. May 2016. arXiv:1605.02065 [cs.CR].

K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *Adv. Neural Inf. Process. Syst. 21*, pages 289–296, 2008.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

C. Dimitrakakis, B. Nelson, A. Mitrokotsa, and B. 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.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011.

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.

C. Dwork and G. N. Rothblum. Concentrated differential privacy. 2016. arXiv:1603.01887 [cs.DS].

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings*, pages 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

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

Z. Ghahramani and M. J. Beal. Propagation algorithms for variational Bayesian learning. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 507–513. MIT Press, 2001.

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

A. Honkela, T. Raiko, M. Kuusela, M. Tornio, and J. Karhunen. Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes. *J Mach Learn Res*, 11:3235–3268, Nov 2010.

A. Honkela, M. Das, A. Nieminen, O. Dikmen, and S. Kaski. Efficient differentially private learning improves drug sensitivity prediction. 2016. arXiv:1606.02109.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, November 1999.

A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei. Automatic differentiation variational inference. *J Mach Learn Res*, 18(14):1–45, 2017.

N. Li, W. Qardaji, and D. 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.

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

M. Park, J. Foulds, K. Chaudhuri, and M. Welling. Variational Bayes in private settings (VIPS). 2016. arXiv:1611.00340 [stat.ML].

J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, apr 2016.

M. Titsias and M. Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *Proc. 31st Int. Conf. Mach. Learn. (ICML 2014)*, pages 1971–1979, 2014.

D. Tran, M. D. Hoffman, R. A. Saurous, E. Brevdo, K. Murphy, and D. M. Blei. Deep probabilistic programming. In *International Conference on Learning Representations*, 2017.

Y. Wang, S. E. Fienberg, and A. 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.

J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. Functional mechanism: Regression analysis under differential privacy. *PVLDB*, 5(11):1364–1375, 2012.

Z. Zhang, B. Rubinstein, and C. Dimitrakakis. On the differential privacy of Bayesian inference. In *Proc. Conf. AAAI Artif. Intell. 2016*, 2016.