Journal of Artificial Intelligence Research 68 (2020) 109-157 Submitted 02/2019; published 05/2020

Variational Bayes In Private Settings (VIPS)

Mijung Park MIJUNG.PARK@TUEBINGEN.MPG.DE
Max Planck Institute for Intelligent Systems,
Department of Computer Science, University of Tübingen,
Max-Planck-Ring 4, 72076 Tüinbingen, Germany

James Foulds JFOULDS@UMBC.EDU
Department of Information Systems,
University of Maryland, Baltimore County,
ITE 447, 1000 Hilltop Circle, Baltimore, MD 21250, USA

Kamalika Chaudhuri KAMALIKA@CS.UCSD.EDU
Department of Computer Science,
University of California, San Diego,
EBU3B 4110, University of California, San Diego, CA 92093, USA

Max Welling M.WELLING@UVA.NL
Amsterdam Machine Learning LAB (AMLAB),
Informatics Institute, University of Amsterdam,
Science Park 904 1098 XH Amsterdam, the Netherlands

Abstract

Many applications of Bayesian data analysis involve sensitive information such as personal documents or medical records, motivating methods which ensure that privacy is protected. We introduce a general privacy-preserving framework for Variational Bayes (VB), a widely used optimization-based Bayesian inference method. Our framework respects differential privacy, the gold-standard privacy criterion, and encompasses a large class of probabilistic models, called the *Conjugate Exponential* (CE) family. We observe that we can straightforwardly privatise VB's approximate posterior distributions for models in the CE family, by perturbing the expected sufficient statistics of the complete-data likelihood. For a broadly-used class of non-CE models, those with binomial likelihoods, we show how to bring such models into the CE family, such that inferences in the modified model resemble the private variational Bayes algorithm as closely as possible, using the Pólya-Gamma data augmentation scheme. The iterative nature of variational Bayes presents a further challenge since iterations increase the amount of noise needed. We overcome this by combining: (1) an improved composition method for differential privacy, called the *moments accountant*, which provides a tight bound on the privacy cost of multiple VB iterations and thus significantly decreases the amount of additive noise; and (2) the privacy amplification effect of subsampling mini-batches from large-scale data in stochastic learning. We empirically demonstrate the effectiveness of our method in CE and non-CE models including latent Dirichlet allocation, Bayesian logistic regression, and sigmoid belief networks, evaluated on real-world datasets.

1. Introduction

Bayesian inference, which reasons over the uncertainty in model parameters and latent variables given data and prior knowledge, has found widespread use in data science application

domains in which privacy is essential, including text analysis (Blei, Ng & Jordan, 2003), medical informatics (Husmeier, Dybowski & Roberts, 2006), and MOOCS (Piech et al., 2013). In these applications, the goals of the analysis must be carefully balanced against the privacy concerns of the individuals whose data are being studied (Daries et al., 2014). The recently proposed *Differential Privacy* (DP) formalism provides a means for analyzing and controlling this trade-off, by quantifying the privacy "cost" of data-driven algorithms (Dwork, McSherry, Nissim & Smith, 2006). In this work, we address the challenge of performing Bayesian inference in private settings, by developing an extension of the widely used *Variational Bayes* (VB) algorithm that preserves differential privacy. We provide extensive experiments across a variety of probabilistic models which demonstrate that our algorithm is a practical, broadly applicable, and statistically efficient method for privacy-preserving Bayesian inference.

The algorithm that we build upon, variational Bayes, provides an optimisation-based alternative to Markov Chain Monte Carlo (MCMC) simulation methods for Bayesian inference, and as such, frequently has faster convergence properties than corresponding MCMC methods. While the variational Bayes algorithm proves its usefulness by successfully solving many statistical problems, the standard form of the algorithm unfortunately cannot guarantee privacy for each individual in the dataset.

Differential Privacy (DP) (Dwork, McSherry, et al., 2006) is a formalism which can be used to establish such guarantees. An algorithm is said to preserve differential privacy if its output is sufficiently noisy or random to obscure the participation of any single individual in the data. By showing that an algorithm satisfies the differential privacy criterion, we are guaranteed that an adversary cannot draw new conclusions about an individual from the output of the algorithm, by virtue of his/her participation in the dataset. However, the injection of noise into an algorithm, in order to satisfy the DP definition, generally results in a loss of statistical efficiency, as measured by accuracy per sample in the dataset. To design practical differentially private machine learning algorithms, the central challenge is to design a noise injection mechanism such that there is a good trade-off between privacy and statistical efficiency.

Iterative algorithms, such as variational Bayes, pose a further challenge when developing a differentially private algorithm: each iteration corresponds to a query to the database which must be privatised, and the number of iterations required to guarantee accurate posterior estimates causes high cumulative privacy loss. To compensate for the loss, one needs to add a significantly higher level of noise to the quantity of interest. We overcome these challenges in the context of variational Bayes by using the following key ideas:

- **Perturbation of the expected sufficient statistics**: Building on the work of Park, Foulds, Chaudhuri & Welling (2017), when models are in the *Conjugate Exponential* (CE) family, we privatise variational posterior distributions simply by perturbing the expected sufficient statistics of the complete-data likelihood. This allows us to make effective use of the *per iteration privacy budget* in each step of the algorithm.

- **Refined composition analysis**: In order to use the privacy budget more effectively *across many iterations*, we calculate the cumulative privacy cost by using an improved composition analysis, the *Moments Accountant* (MA) method of Abadi et al. (2016). This method maintains a bound on the log of the moment generating function of the

privacy loss incurred by applying multiple mechanisms to the dataset, i.e. one mechanism for each iteration of a machine learning algorithm. This allows for allocating a higher budget per-iteration than standard methods.

- **Privacy amplification effect from subsampling of large-scale data**: Processing the entire dataset in each iteration is extremely expensive in the big data setting, and is not possible in the case of streaming data, for which the size of the dataset is assumed to be infinite. Stochastic learning algorithms provide a scalable alternative by performing parameter updates based on subsampled mini-batches of data, and this has proved to be highly advantageous for large-scale applications of variational Bayes (Hoffman, Blei, Wang & Paisley, 2013). This subsampling procedure, in fact, has a further benefit of *amplifying privacy*. Our results confirm that *subsampling works synergistically in concert with moments accountant composition* to make effective use of an overall privacy budget (Wang, Balle & Kasiviswanathan, 2019). While there are several prior works on differentially private algorithms in stochastic learning (e.g., Bassily, Smith & Thakurta, 2014; Song, Chaudhuri & Sarwate, 2013; Wang, Fienberg & Smola, 2015; Wang et al., 2019; Wang, Lei & Fienberg, 2016; Wu, Kumar, Chaudhuri, Jha & Naughton, 2016), the use of privacy amplification due to subsampling, together with MA composition, has not been used in the context of variational Bayes before (Abadi et al., 2016 used this approach for privacy-preserving deep learning).
- **Data augmentation for the non-CE family models**: For widely used non-CE models with binomial likelihoods such as logistic regression, we exploit the *Pólya-Gamma* data augmentation scheme (Polson, Scott & Windle, 2013) to bring such models into the CE family, such that inferences in the modified model resemble our private variational Bayes algorithm as closely as possible. Unlike recent work which involves perturbing and clipping gradients for privacy (Jälkö, Dikmen & Honkela, 2017), our method uses an improved composition method, and also maintains the closed-form updates for the variational posteriors and the posterior over hyper-parameters, and results in an algorithm which is more faithful to the standard CE variational Bayes method. Several papers have used the Pólya-Gamma data augmentation method in order to perform Bayesian inference, either exactly via Gibbs sampling (Polson et al., 2013), or approximately via variational Bayes (Gan, Henao, Carlson & Carin, 2015). However, this augmentation technique has not previously been used in the context of differential privacy.

Taken together, these ideas result in an algorithm for privacy-preserving variational Bayesian inference that is both practical and broadly applicable. Our private VB algorithm makes effective use of the privacy budget, both per iteration and across multiple iterations, and with further improvements in the stochastic variational inference setting. Our algorithm is also general, and applies to the broad class of CE family models, as well as non-CE models with binomial likelihoods. We present extensive empirical results demonstrating that our algorithm can preserve differential privacy while maintaining a high degree of statistical efficiency, leading to practical private Bayesian inference on a range of probabilistic models. Our code is available at `https://github.com/mijungi/vips_code`.

We organise the remainder of this paper as follows. First, we review relevant background information on differential privacy, privacy-preserving Bayesian inference, and variational

Bayes in Sec. 2. We then introduce our novel general framework of private variational Bayes in Sec. 3 and illustrate how to apply that general framework to the latent Dirichlet allocation model in Sec. 4. In Sec. 5, we introduce the Pólya-Gamma data augmentation scheme for non-CE family models, and we then illustrate how to apply our private variational Bayes algorithm to Bayesian logistic regression (Sec. 6) and sigmoid belief networks (Sec. 7). Lastly, we summarise our paper and provide future directions in Sec. 8.

2. Background

We begin with some background information on differential privacy, general techniques for designing differentially private algorithms and the composition techniques that we use. We then provide related work on privacy-preserving Bayesian inference, as well as the general formulation of the variational inference algorithm. Readers who are already knowledgeable in these areas may safely skip over the following, but may wish to refer back as needed for the relevant notation.

2.1 Differential Privacy

Differential Privacy (DP) is a formal definition of the privacy properties of data analysis algorithms (Dwork, McSherry, et al., 2006; Dwork, et al., 2014).

Definition 1 (Differential Privacy) *A randomized algorithm $\mathcal{M}(\mathcal{D})$ is said to be (ϵ, δ)-differentially private if*

$$P(\mathcal{M}(\mathcal{D}) \in \mathcal{S}) \leq \exp(\epsilon) P(\mathcal{M}(\mathcal{D}') \in \mathcal{S}) + \delta \tag{1}$$

for all measurable subsets $\mathcal{S}$ of the range of $\mathcal{M}$ and for all datasets $\mathcal{D}$, $\mathcal{D}'$ differing by a single entry.

In this paper, we assume that the entry difference incurs by replacing an entry with a different value (the "*replace-one*" version of Def. 1).[1] Note that an entry usually corresponds to a single individual's private value. We denote this as $\mathrm{d}(\mathcal{D}, \mathcal{D}') \leq 1$, where $\mathrm{d}(\mathcal{D}, \mathcal{D}')$ is the number of entries which differ between $\mathcal{D}$ and $\mathcal{D}'$. If $\delta = 0$, the algorithm is said to be ϵ-differentially private, and if $\delta > 0$, it is said to be *approximately* differentially private.

Intuitively, the definition states that the probability of any event does not change very much when a single individual's data is modified, thereby limiting the amount of information that the algorithm reveals about any one individual. We observe that $\mathcal{M}$ is a randomized algorithm, and randomization is typically achieved by either adding external noise, or by subsampling.

1. We use this version of differential privacy in order to make use of Wang et al. (2019)'s analysis of the "privacy amplification" effect of subsampling for the moments accountant. Privacy amplification is also possible with the "*include/exclude*' definition of DP, in which differing by a single entry refers to the inclusion or exclusion of that entry in the dataset; cf. Abadi et al. (2016) for an analysis of the specific case of MA using the Gaussian mechanism.

2.2 Designing Differentially Private Algorithms

There are several standard approaches for designing differentially-private algorithms – see Dwork & Roth (2014) and Sarwate & Chaudhuri (2013) for surveys. The classical approach is output perturbation, where the idea is to add noise to the output of a function computed on sensitive data. The most common form of output perturbation is the *global sensitivity method* by Dwork, McSherry, et al. (2006), where the idea is to calibrate the noise added to the global sensitivity of the function.

2.2.1 The Global Sensitivity Mechanism

The global sensitivity of a one-dimensional[2] function F of a dataset $\mathcal{D}$ is defined as the maximum amount (over all datasets $\mathcal{D}$) by which F changes when the private value of a single individual in $\mathcal{D}$ changes. Specifically,

$$\Delta F = \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |F(\mathcal{D}) - F(\mathcal{D}')|,$$

where $\mathcal{D}$ is allowed to vary over the entire data domain, and $|F(\cdot)|$ can correspond to either the L_1 norm or the L_2 norm of the function, depending on the noise mechanism used.

In this paper, we consider a specific form of the global sensitivity method, called the *Gaussian mechanism* (Dwork, Kenthapadi, McSherry, Mironov & Naor, 2006), where Gaussian noise calibrated to the global sensitivity in the L_2 norm is added. Specifically, for a function F with global sensitivity ΔF, we output:

$$F(\mathcal{D}) + Z, \quad \text{where } Z \sim N(0, \sigma^2) \ , \ \sigma^2 \geq 2\log(1.25/\delta)(\Delta F)^2/\epsilon^2 \ ,$$

and where ΔF is computed using the L_2 norm, and is referred to as the *L2 sensitivity* of the function F. The privacy properties of this method are illustrated by Dwork & Roth (2014); Bun & Steinke (2016).

2.2.2 Other Mechanisms

A variation of the global sensitivity method is the *smoothed sensitivity method* (Nissim, Raskhodnikova & Smith, 2007), where the standard deviation of the added noise depends on the dataset itself, and is less when the dataset is *well-behaved*. Computing the smoothed sensitivity in a tractable manner is a major challenge, and efficient computational procedures are known only for a small number of relatively simple tasks.

A second, different approach is the *exponential mechanism* (McSherry & Talwar, 2007), a generic procedure for privately solving an optimisation problem where the objective depends on sensitive data. The exponential mechanism outputs a sample drawn from a density concentrated around the optimal value; this method too is computationally inefficient for large domains. A third approach is *objective perturbation* (Chaudhuri, Monteleoni & Sarwate, 2011), where an optimisation problem is perturbed by adding a (randomly drawn) term and its solution is output; while this method applies easily to convex optimisation problems such as those that arise in logistic regression and SVM, it is unclear how to apply it to more complex optimisation problems that arise in Bayesian inference.

2. An extension to a multi-dimensional function is straightforward.

A final approach for designing differentially private algorithms is sample-and-aggregate (Nissim et al., 2007), where the goal is to boost the amount of privacy offered by running private algorithms on subsamples, and then aggregating the result. The key privacy mechanism we use in this paper is a number of iterations of the Subsampled Gaussian Mechanism – which is a combination of output perturbation along with sample-and-aggregate.

2.3 Composition

An important property of differential privacy which makes it conducive to real applications is *composition*, which means that the privacy guarantees decay gracefully as the same private dataset is used in multiple releases. This property allows us to easily design private versions of iterative algorithms by making each iteration private, and then accounting for the privacy loss incurred by a fixed number of iterations.

The first composition result was established by Dwork, Kenthapadi, et al. (2006); Dwork, McSherry, et al. (2006), who showed that differential privacy composes *linearly*; if we use differentially private algorithms $A_1, \ldots, A_k$ with privacy parameters $(\epsilon_1, \delta_1), \ldots, (\epsilon_k, \delta_k)$ then the resulting process is $(\sum_k \epsilon_k, \sum_k \delta_k)$-differentially private. This result was improved by Dwork, Rothblum & Vadhan (2010) to provide a better rate for (ϵ, δ)-differential privacy. Kairouz, Oh & Viswanath (2017) improved this even further, and provided a characterization of optimal composition for any differentially private algorithm.

2.3.1 The Moments Accountant Method

In this paper, we use the *Moments Accountant* (MA) composition method due to Abadi et al. (2016) for accounting for privacy loss incurred by successive iterations of an iterative mechanism, as further refined by Wang et al. (2019). We choose this method as it provides a more refined privacy analysis than Dwork et al. (2010), and applies more generally than zCDP composition (Bun & Steinke, 2016). Moreover, unlike the results of Kairouz et al. (2017), this method is tailored to specific differentially private algorithms – which results in tighter privacy accounting.

The moments accountant method is based on the concept of a *privacy loss random variable*, which allows us to consider the entire spectrum of likelihood ratios $\frac{P(\mathcal{M}(\mathcal{D})=o)}{P(\mathcal{M}(\mathcal{D}')=o)}$ induced by a privacy mechanism $\mathcal{M}$. Specifically, the privacy loss random variable corresponding to a mechanism $\mathcal{M}$, datasets $\mathcal{D}$ and $\mathcal{D}'$, and an auxiliary parameter[3] w is a random variable defined as follows:

$$L_{\mathcal{M}}(\mathcal{D}, \mathcal{D}', w) := \log \frac{P(\mathcal{M}(\mathcal{D}, w) = o)}{P(\mathcal{M}(\mathcal{D}', w) = o)}, \quad \text{with likelihood } P(\mathcal{M}(\mathcal{D}, w) = o),$$

where o lies in the range of $\mathcal{M}$. Observe that if $\mathcal{M}$ is $(\epsilon, 0)$-differentially private, then the absolute value of $L_{\mathcal{M}}(\mathcal{D}, \mathcal{D}', w)$ is at most ϵ with probability 1.

The moments accountant method exploits properties of this privacy loss random variable to account for the privacy loss incurred by applying mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_t$ successively to a dataset $\mathcal{D}$; this is done by bounding properties of the log of the moment generating

3. The the auxiliary parameter could be, for instance, in the adaptive data analysis, the output of all the previous mechanisms $\mathcal{M}_{1,\cdots,k-1}$, as the input to the kth mechanism $\mathcal{M}_k$.

function of the privacy loss random variable. Specifically, the log moment function $\alpha_{\mathcal{M}_t}$ of a mechanism $\mathcal{M}_t$ is defined as:

$$\alpha_{\mathcal{M}_t}(\lambda) = \sup_{\mathcal{D},\mathcal{D}',w} \log \mathbb{E}[\exp(\lambda L_{\mathcal{M}_t}(\mathcal{D},\mathcal{D}',w))], \tag{2}$$

where $\mathcal{D}$ and $\mathcal{D}'$ are datasets that differ in the private value of a single person.[4] Abadi et al. (2016) showed that if $\mathcal{M}$ is the combination of mechanisms $(\mathcal{M}_1,\ldots,\mathcal{M}_k)$ where each mechanism adds independent noise, then, its log moment generating function $\alpha_{\mathcal{M}}$ has the property that:

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum_{t=1}^{k} \alpha_{\mathcal{M}_t}(\lambda) \ . \tag{3}$$

Additionally, given a log moment function $\alpha_{\mathcal{M}}$, the corresponding mechanism $\mathcal{M}$ satisfies a range of privacy parameters (ϵ,δ) connected by the following equations. For any chosen target $\epsilon > 0$, $\mathcal{M}$ is (ϵ,δ)-DP for

$$\delta = \min_{\lambda} \exp(\alpha_{\mathcal{M}}(\lambda) - \lambda\epsilon) \ , \tag{4}$$

and as a consequence of the above, for any chosen target δ $\mathcal{M}$ is (ϵ,δ)-DP for

$$\epsilon = \min_{\lambda} \frac{\log(1/\delta) + \alpha_{\mathcal{M}_t}(\lambda)}{\lambda}. \tag{5}$$

These properties immediately suggest a procedure for tracking privacy loss incurred by a combination of mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$ on a dataset. For each mechanism $\mathcal{M}_t$, first compute the log moment function $\alpha_{\mathcal{M}_t}$; for certain mechanisms such as the Gaussian mechanism this can be done by simple algebra. Next, compute a bound on $\alpha_{\mathcal{M}}$ for the combination $\mathcal{M} = (\mathcal{M}_1,\ldots,\mathcal{M}_k)$ from (3), and finally, recover the privacy parameters of $\mathcal{M}$ by either finding the best δ for a target ϵ using (4), or the best ϵ for a target δ using (5).

2.3.2 Moments Accountant for the Subsampled Gaussian Mechanism

The key privacy tool that we used in this paper is the composition of multiple iterations of the Subsampled Gaussian Mechanism. The privacy loss of this procedure was originally analyzed by Abadi et al. (2016), but in this paper we instead make use of a more recently proposed version of the Moments Accountant that allows for more precise privacy calculations, the Analytical Moments Accountant (Wang et al., 2019).

The Subsampled Gaussian Mechanism, under the analysis of Wang et al. (2019), works as follows. Given a dataset $\mathcal{D}$ with N datapoints and a sampling proportion υ, in each iteration, we draw a fresh independent random sample of υN points from the entire dataset $\mathcal{D}$. These υN points are drawn *without replacement* and are hence distinct.[5] We then

4. In this paper, we will interchangeably denote expectations as $\mathbb{E}[\cdot]$ or $\langle\cdot\rangle$.
5. Note however that successive iterations may involve overlapping samples – as these are drawn independently from the entire dataset. Also note that the original Moments Accountant analysis of Abadi et al. (2016) used a different subsampling procedure: *Poisson subsampling*, in which each data point is included in the subsample according to an independent Bernoulli trial with a fixed probability. Poisson subsampling leads to subsamples of different sizes, which is not concordant with the usual practice of minibatch stochastic gradient descent; the *subsampling with replacement* procedure of Wang et al. (2019) does not have this issue.

compute a function F (to be specified later) on these υN points, and privatize F using the Gaussian mechanism.

How do we compute the log-moment of this mechanism? Theorem 9 of Wang et al. (2019) provides an analytical expression for computing the log-moment of the Subsampled Gaussian Mechanism as a function of the sampling proportion, and the log-moments of the Gaussian Mechanism. Their overall approach is to upper bound the privacy parameters of subsampled mechanisms which satisfy Rényi Differential Privacy (RDP), a generalization of Differential Privacy which has a direct correspondence to the log moment generating function $\alpha_{\mathcal{M}}$. The calculation of the log moment generating function of the Subsampled Gaussian Mechanism is a special case of their analysis. We refer the reader to Wang et al. (2019) for details.[6]

Since all log-moments of the Gaussian Mechanism can be calculated directly by simple algebra (Abadi et al., 2016; Wang et al., 2019), this gives an analytical way to calculate the log-moment of a single iteration of the Subsampled Gaussian Mechanism. Successive iterations are then composed by adding up the log-moments as described in Section 2.3.1.

2.4 Privacy-Preserving Bayesian Inference

Privacy-preserving Bayesian inference is a new research area which is currently receiving a lot of attention. Dimitrakakis, Nelson, Mitrokotsa & Rubinstein (2014) showed that Bayesian posterior sampling is automatically differentially private, under a mild sensitivity condition on the log likelihood. This result was independently discovered by Wang et al. (2015), who also showed that the Stochastic Gradient Langevin Dynamics (SGLD) Bayesian inference algorithm (Welling & Teh, 2011) automatically satisfies approximate differential privacy, due to the use of Gaussian noise in the updates.

As an alternative to obtaining privacy "for free" from posterior sampling, Foulds, Geumlek, Welling & Chaudhuri (2016) studied a Laplace mechanism approach for exponential family models, and proved that it is asymptotically efficient, unlike the former approach. Independently of this work, Zhang, Rubinstein & Dimitrakakis (2016) proposed an equivalent Laplace mechanism method for private Bayesian inference in the special case of beta-Bernoulli systems, including graphical models constructed based on these systems. Foulds et al. (2016) further analyzed privacy-preserving MCMC algorithms, via both exponential and Laplace mechanism approaches.

In terms of approximate Bayesian inference, Jälkö et al. (2017) recently considered privacy-preserving variational Bayes via perturbing and clipping the gradients of the variational lower bound. However, this work focuses its experiments on logistic regression, a model that does not have latent variables. Given that most latent variable models consist of at least as many latent variables as the number of datapoints, the long vector of gradients (the concatenation of the gradients with respect to the latent variables; and with respect to the model parameters) in such cases is expected to typically require excessive amounts of additive noise. Furthermore, our approach, using the data augmentation scheme (see Sec. 5) and moment perturbation, yields closed-form posterior updates (posterior distributions both for latent variables and model parameters) that are closer to the spirit of the original

6. For code implementing the Analytical Moments Accountant analysis, see `https://github.com/yuxiangw/autodp/tree/master/autodp`.

variational Bayes method, for both CE and non-CE models, as well as an improved composition analysis using moments accountant. On the other hand, the method of Jälkö et al. (2017) applies to a more general class of non-conjugate non-CE models, such as those that may be encountered in a black-box VI setting.

In recent work, Barthe et al. (2016) designed a *probabilistic programming* language for designing privacy-preserving Bayesian machine learning algorithms, with privacy achieved via input or output perturbation, using standard mechanisms.

Lastly, although it is not a fully Bayesian method, it is worth noting the differentially private expectation maximisation algorithm developed in our previous work (Park et al., 2017), which also involves perturbing the expected sufficient statistics for the complete-data likelihood. The major difference between our current and previous work is that EM is not (fully) Bayesian, i.e., EM outputs the *point estimates* of the model parameters; while VB outputs the posterior distributions (or those quantities that are necessary to do Bayesian predictions, e.g., expected natural parameters and expected sufficient statistics). Furthermore, Park et al. (2017) dealt with only CE family models for obtaining the closed-form MAP estimates of the parameters; while our approach here encompasses both CE and non-CE family models, using the Polya-Gamma data augmentation technique which is particularly well-suited for fully Bayesian inference (Polson et al., 2013). Finally, we demonstrated our differentially private EM method on small- to medium-sized datasets (Park et al., 2017), which do not require stochastic learning; while our method here addresses the scenario of stochastic learning which is essential in the era of big data (Hoffman et al., 2013).

2.5 Variational Bayes

Variational Bayes is an inference algorithm which has origins in the closely related *Expectation Maximisation* (EM) algorithm (Dempster, Laird & Rubin, 1977), although there are important differences between these methods. As mentioned above, VB aims to approximate fully Bayesian inference, while EM does not. It will nevertheless be convenient to frame our discussion on VB in the context of EM, following Beal (2003). The EM algorithm seeks to learn the parameters of models with latent variables. Since directly optimising the log likelihood of observations under such models is intractable, EM introduces a lower bound on the log likelihood by rewriting it in terms of auxiliary probability distributions over the latent variables, and using a Jensen's inequality argument. EM proceeds by iteratively alternating between improving the bound via updating the auxiliary distributions (the E-step), and optimising the lower bound with respect to the parameters (the M-step).

Alternatively, EM can instead be understood as an instance of the *variational method*. The general idea of the variational method is to cast a quantity of interest as an optimisation problem. *Variational inference* is the class of techniques which solve inference problems in probabilistic models using variational methods (Jordan, Ghahramani, Jaakkola & Saul, 1999; Wainwright & Jordan, 2008).[7] In the variational inference interpretation of EM, both the E- and M-steps are viewed as maximising the same joint objective function: a reinterpretation of the bound as a *variational* lower bound which is related to a quantity known as the

7. Note that the "variational" terminology comes from the calculus of variations, which is concerned with finding optima of functionals. This pertains to probabilistic inference, since distributions are functions, and we aim to optimise over the space of possible distributions. However, it is typically not necessary to use the calculus of variations when deriving variational inference algorithms in practice.

variational free energy in statistical physics, and to the KL-divergence (Neal & Hinton, 1998).[8]

This interpretation opens the door to extensions in which simplifying assumptions are made on the optimization problem, thereby trading improved computational tractability against tightness of the bound. Such simplifications, for instance assuming that the auxiliary distributions are fully factorized, make feasible a fully Bayesian extension of EM, called *Variational Bayesian* EM (VBEM) (Beal, 2003). While VBEM has a somewhat similar algorithmic structure to EM, it aims to compute a fundamentally different object: an approximation to the entire posterior distribution, instead of a point estimate of the parameters. We collectively refer to VBEM and an intermediary method between it and EM, called *Variational* EM (VEM), as *variational Bayes*.[9]

Our discussion is focused on the Variational Bayesian EM (VBEM) algorithm variant. The goal of the algorithm is to compute an approximation q to the posterior distribution over latent variables and model parameters for models where exact posterior inference is intractable. This should be contrasted to VEM and EM, which aim to compute a point estimate of the parameters. VEM and EM can both be understood as special cases, in which the set of q distributions is constrained such that the approximate posterior is a Dirac delta function (Beal, 2003). We therefore include them within the definition of VB. See Blei, Kucukelbir & McAuliffe (2017) for a recent review on variational Bayes, Beal (2003) for more detailed derivations, and see Appendix Sec. 6 for more information on the relationship between EM and VBEM. We summarize the most important notational symbols for the VB algorithm, and its stochastic variant, in Table 1.

2.5.1 High-Level Derivation of VB

Consider a generative model that produces a dataset $\mathcal{D} = \{\mathcal{D}_n\}_{n=1}^N$ consisting of N (conditionally) independent identically distributed (*i.i.d.*) items ($\mathcal{D}_n$ denotes the nth input/output pair $\{\mathbf{x}_n, y_n\}$ for supervised learning and the nth vector output $\mathbf{y}_n$ for unsupervised learning), generated using a set of latent variables $\boldsymbol{l} = \{\boldsymbol{l}_n\}_{n=1}^N$. The generative model provides $p(\mathcal{D}_n|\boldsymbol{l}_n, \boldsymbol{m})$, where $\boldsymbol{m}$ are the model parameters. We also consider the prior distribution over the model parameters $p(\boldsymbol{m})$ and the distribution over the latent variables $p(\boldsymbol{l}|\boldsymbol{m})$, which are assumed to be conditionally independent given $\boldsymbol{m}$.

Variational Bayes recasts the task of approximating the posterior $p(\boldsymbol{l}, \boldsymbol{m}|\mathcal{D})$ as an optimisation problem: making an approximating distribution $q(\boldsymbol{l}, \boldsymbol{m})$, which is called the *variational distribution*, as similar as possible to the posterior, by minimising some distance (or divergence) between them. The algorithm optimises over $q(\boldsymbol{l}, \boldsymbol{m})$ based on the input dataset $\mathcal{D}$, and outputs the (hopefully) optimal variational distribution $q(\boldsymbol{l}, \boldsymbol{m})$, typically via a set of parameters which encode $q(\boldsymbol{l}, \boldsymbol{m})$.

8. See Appendix Sec. 6 for a detailed derivation.
9. Variational EM uses simplifying assumptions on the auxiliary distribution over latent variables, as in VBEM, but still aims to find a point estimate of the parameters, as in EM. See Sec. 2.5 for more details on variational Bayes, and Appendix Sec. 6 for more details on EM.

Symbol	Definition
$\mathcal{D}$	Dataset containing N data points, $\{\mathbf{x}_n, y_n\}$ or $\{\mathbf{y}_n\}$
n	Indexes the nth data point when used as a subscript
p	Probability distribution of the assumed generative model
$\boldsymbol{l}$	Latent variables per data point in the model p
$\boldsymbol{m}$	Model parameters for p
q	Variational distribution
$\mathcal{L}(q)$	Evidence lower bound (ELBO) objective function for learning q
$q^*(\boldsymbol{l}_n)$	Optimal coordinate-ascent update for $q(\boldsymbol{l}_n)$
$q^*(\boldsymbol{m})$	Optimal coordinate-ascent update for $q(\boldsymbol{m})$
H	Entropy (or differential entropy)
$\mathbf{n}(\boldsymbol{m})$	Natural parameters
$\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)$	Sufficient statistics
$\bar{\mathbf{n}}$	Expected natural parameters (under the variational distribution q)
$\bar{\mathbf{s}}(\mathcal{D})$	Expected sufficient statistics (under the variational distribution q)
$\bar{\mathbf{s}}$	Stochastic estimate of $\bar{\mathbf{s}}(\mathcal{D})$
S	Mini-batch size
J	Number of minibatches to process
ρ_t	Step size at iteration t of stochastic VB
υ	Subsampling rate

Table 1: Summary of the main notation for VB (top) and stochastic VB (bottom).

The terminology *VB* is often assumed to refer to the standard case, in which the divergence to minimise is the KL-divergence from $p(\boldsymbol{l}, \boldsymbol{m}|\mathcal{D})$ to $q(\boldsymbol{l}, \boldsymbol{m})$,

$$\begin{aligned} D_{KL}(q(\boldsymbol{l}, \boldsymbol{m}) \| p(\boldsymbol{l}, \boldsymbol{m}|\mathcal{D})) &= \mathbb{E}_q\Big[\log \frac{q(\boldsymbol{l}, \boldsymbol{m})}{p(\boldsymbol{l}, \boldsymbol{m}|\mathcal{D})}\Big] \\ &= \mathbb{E}_q[\log q(\boldsymbol{l}, \boldsymbol{m})] - \mathbb{E}_q[\log p(\boldsymbol{l}, \boldsymbol{m}|\mathcal{D})] \\ &= \mathbb{E}_q[\log q(\boldsymbol{l}, \boldsymbol{m})] - \mathbb{E}_q[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})] + \log p(\mathcal{D}) \ . \end{aligned} \tag{6}$$

The arg min of Eq. 6 with respect to $q(\boldsymbol{l}, \boldsymbol{m})$ does not depend on the constant $\log p(\mathcal{D})$. Minimising it is therefore equivalent to maximizing

$$\mathcal{L}(q) \triangleq \mathbb{E}_q[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})] - \mathbb{E}_q[\log q(\boldsymbol{l}, \boldsymbol{m})] = \mathbb{E}_q[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})] + H(q) \ , \tag{7}$$

where $H(q)$ is the entropy (or differential entropy) of $q(\boldsymbol{l}, \boldsymbol{m})$. The entropy of $q(\boldsymbol{l}, \boldsymbol{m})$ rewards simplicity, while $\mathbb{E}_q[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})]$, the expected value of the complete data log-likelihood under the variational distribution, rewards accurately fitting to the data. We can alternatively derive $\mathcal{L}(q)$ as a lower bound on the log of the marginal probability of the data $p(\mathcal{D})$ (the *evidence*),

$$\begin{aligned} \log p(\mathcal{D}) &= \log\Big(\int d\boldsymbol{l}\ d\boldsymbol{m}\ p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})\Big) = \log\Big(\int d\boldsymbol{l}\ d\boldsymbol{m}\ p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D}) \frac{q(\boldsymbol{l}, \boldsymbol{m})}{q(\boldsymbol{l}, \boldsymbol{m})}\Big) \\ &= \log\Big(\mathbb{E}_q\Big[\frac{p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})}{q(\boldsymbol{l}, \boldsymbol{m})}\Big]\Big) \geq \mathbb{E}_q\Big[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D}) - \log q(\boldsymbol{l}, \boldsymbol{m})\Big] = \mathcal{L}(q) \ , \end{aligned} \tag{8}$$

where we have made use of Jensen's inequality. Due to Eq. 8, $\mathcal{L}(q)$ is sometimes referred to as a variational lower bound, and in particular, the *Evidence Lower Bound* (ELBO), since it is a lower bound on the log of the model evidence $p(\mathcal{D})$.

The VBEM algorithm maximises $\mathcal{L}(q)$ via coordinate ascent over parameters encoding $q(\boldsymbol{l}, \boldsymbol{m})$, called the variational parameters. The E-step optimises the variational parameters pertaining to the latent variables $\boldsymbol{l}$, and the M-step optimises the variational parameters pertaining to the model parameters $\boldsymbol{m}$. These updates are iterated until convergence. Under certain assumptions, these updates have a convenient form, as we will describe below.

2.5.2 Mean-Field VB

Since the KL-divergence is 0 if and only if the two distributions are the same, maximising $\mathcal{L}(q)$ will result in $q(\boldsymbol{l}, \boldsymbol{m}) = p(\boldsymbol{l}, \boldsymbol{m}|\mathcal{D})$ when the optimisation problem is unconstrained. In practice, however, we restrict $q(\boldsymbol{l}, \boldsymbol{m})$ to a tractable subset of possible distributions. A common choice for the tractable subset is the set of fully factorised distributions:

$$q(\boldsymbol{l}, \boldsymbol{m}) = q(\boldsymbol{l})q(\boldsymbol{m}) = q(\boldsymbol{m}) \prod_{n=1}^{N} q(\boldsymbol{l}_n) \ . \tag{9}$$

Under this assumption, known as the *mean-field assumption* by analogy to a related technique for modeling behaviors of particles in statistical physics, the method is referred to as *mean-field variational Bayes*. The ELBO then has the form

$$\mathcal{L}(q(\boldsymbol{l})q(\boldsymbol{m})) = \int d\boldsymbol{m}\ d\boldsymbol{l}\ q(\boldsymbol{l})q(\boldsymbol{m}) \log \frac{p(\boldsymbol{l}, \mathcal{D}, \boldsymbol{m})}{q(\boldsymbol{m}) \prod_{n=1}^{N} q(\boldsymbol{l}_n)} \ . \tag{10}$$

It is possible to derive the general structure of coordinate ascent updates to optimize the ELBO without making any further distributional assumptions or assuming a specific parameterisation for $q(\boldsymbol{l})$ or $q(\boldsymbol{m})$. After some algebra (Beal, 2003), we can isolate the terms in the ELBO relating to latent variable $\boldsymbol{l}_n$ and rewrite it as

$$\begin{aligned} \mathcal{L}(q(\boldsymbol{l})q(\boldsymbol{m})) &= -D_{KL}(q_n(\boldsymbol{l}_n) \| c_n(\boldsymbol{l}_n)) + \text{const}, \\ c_n(\boldsymbol{l}_n) &\triangleq \exp(\mathbb{E}_{q_{\neg n}}[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})])/Z_n \ , \\ Z_n &= \int d\boldsymbol{l}_n \exp(\mathbb{E}_{q_{\neg n}}[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})]) \ , \end{aligned} \tag{11}$$

where $q_{\neg n}$ is the variational distribution $q(\boldsymbol{l})q(\boldsymbol{m})$ excluding variable $\boldsymbol{l}_n$, and $c_n(\boldsymbol{l}_n)$ is a distribution over $\boldsymbol{l}_n$ with normalizing constant Z_n. KL-divergences are minimised by setting the two distributions to be equal, so the optimal $q(\boldsymbol{l}_n)$ for the coordinate ascent update, denoted $q^*(\boldsymbol{l}_n)$, is equal to $c_n(\boldsymbol{l}_n)$, i.e.

$$q^*(\boldsymbol{l}_n) \propto \exp(\mathbb{E}_{q_{\neg n}}[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})]) \ . \tag{12}$$

By an identical argument, we also obtain the optimal coordinate-ascent update for $q(\boldsymbol{m})$ as

$$q^*(\boldsymbol{m}) \propto \exp(\mathbb{E}_{q(\boldsymbol{l})}[\log p(\boldsymbol{l}, \boldsymbol{m}, \mathcal{D})]) \ . \tag{13}$$

2.5.3 VB for CE Models

Mean-field VB simplifies further when the model falls into the Conjugate-Exponential (CE) class of models, which satisfy two conditions (Beal, 2003):

(1) The complete-data likelihood is in the exponential family:

$$p(\mathcal{D}_n, \boldsymbol{l}_n|\boldsymbol{m}) = g(\boldsymbol{m})f(\mathcal{D}_n, \boldsymbol{l}_n)\exp(\mathbf{n}(\boldsymbol{m})^\top \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)), \tag{14}$$

(2) The prior over $\boldsymbol{m}$ is conjugate to the complete-data likelihood:

$$p(\boldsymbol{m}|\tau, \boldsymbol{\nu}) = h(\tau, \boldsymbol{\nu})g(\boldsymbol{m})^\tau \exp(\boldsymbol{\nu}^\top \mathbf{n}(\boldsymbol{m})), \tag{15}$$

where the natural parameters (to be inferred) and sufficient statistics (a function of the data and the latent variables to be inferred) of the complete-data likelihood are denoted by $\mathbf{n}(\boldsymbol{m})$ and $\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)$, respectively, and g, f, h are some known functions. The hyperparameters (that need to be tuned) are denoted by τ (a scalar) and $\boldsymbol{\nu}$ (a vector).

A large class of models fall in the CE family. Examples include linear dynamical systems and switching models; Gaussian mixtures; factor analysis and probabilistic PCA; Hidden Markov Models (HMM) and factorial HMMs; and discrete-variable belief networks. Some models that are widely used but not in the CE family are: Markov Random Fields (MRFs) and Boltzmann machines; logistic regression; sigmoid belief networks; and Independent Component Analysis (ICA). We illustrate how best to bring such models into the CE family in a later section.

The VB algorithm for a CE family model can be derived by plugging the CE modeling assumptions (1) and (2) into the generic mean-field updates in Eq. 12 and Eq. 13:

$$q^*(\boldsymbol{l}_n) \propto f(\mathcal{D}_n, \boldsymbol{l}_n)\exp(\mathbb{E}_{q(\boldsymbol{m})}[\mathbf{n}(\boldsymbol{m})]^\top \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)) \ , \tag{16}$$

$$\begin{aligned} q^*(\boldsymbol{m}) &\propto g(\boldsymbol{m})^\tau \exp(\boldsymbol{\nu}^\top \mathbf{n}(\boldsymbol{m})) \prod_{n=1}^{N} \Big(g(\boldsymbol{m})\exp(\mathbf{n}(\boldsymbol{m})^\top \mathbb{E}_{q(\boldsymbol{l}_n)}[\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)])\Big) \\ &\propto g(\boldsymbol{m})^{\tau+N} \exp\Big(\big(\boldsymbol{\nu} + \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{l}_n)}[\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)]\big)^\top \mathbf{n}(\boldsymbol{m})\Big) \ . \end{aligned} \tag{17}$$

The optimal coordinate-ascent updated variational distributions are thus in the exponential family. We can see that $q^*(\boldsymbol{m})$ is in the same exponential family as $p(\boldsymbol{m}|\tau, \boldsymbol{\nu})$, and is hence normalized by the multiplicative constant $h(\tau + N, \boldsymbol{\nu} + \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{l}_n)}[\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)])$. If we further assume that $p(\boldsymbol{l}_n|\boldsymbol{m})$ is an exponential family distribution, then $q^*(\boldsymbol{l}_n)$ is in, the same exponential family as $p(\boldsymbol{l}_n|\boldsymbol{m})$, e.g. if $p(\boldsymbol{l}_n|\boldsymbol{m})$ is a Gaussian then so is $q^*(\boldsymbol{l}_n)$ (Blei et al., 2017; Hoffman et al., 2013). The optimal natural parameters for $q^*(\boldsymbol{l}_n)$, called "*local parameters*" by Blei et al. (2017); Hoffman et al. (2013), are $\mathbb{E}_{q(\boldsymbol{m})}[\mathbf{n}(\boldsymbol{m})]$, and the optimal natural parameters for $q^*(\boldsymbol{m})$ (called "*global parameters*") are $\boldsymbol{\nu} + \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{l}_n)}[\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)]$.

Crucially, these updates can be calculated **based on the expected sufficient statistics and the expected natural parameters under the variational distribution**, and these are hence the quantities that the algorithm outputs and maintains in each iteration. We will make use of this fact in our privacy-preserving VB algorithm, which we introduce in the next section. Let $\bar{\mathbf{n}}$ be an estimate of the expected natural parameters under the variational distribution, and let $\bar{\mathbf{s}}(\mathcal{D})$ be an estimate of the expected sufficient statistics under

the variational distribution, which can be randomly initialized. The VB algorithm for CE models is the following two-step procedure.

(1) First, given expected natural parameters $\bar{\mathbf{n}}$, the E-step computes:

$$q(\boldsymbol{l}) = \prod_{n=1}^{N} q(\boldsymbol{l}_n) \propto \prod_{n=1}^{N} f(\mathcal{D}_n, \boldsymbol{l}_n) \exp(\bar{\mathbf{n}}^\top \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)) = \prod_{n=1}^{N} p(\boldsymbol{l}_n | \mathcal{D}_n, \bar{\mathbf{n}}). \tag{18}$$

Using $q(\boldsymbol{l})$, it outputs expected sufficient statistics, the expectation of $\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)$ with probability density $q(\boldsymbol{l}_n)$: $\bar{\mathbf{s}}(\mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{l}_n)}[\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)]$.

(2) Second, given expected sufficient statistics $\bar{\mathbf{s}}(\mathcal{D})$, the M-step computes:

$$q(\boldsymbol{m}) = h(\tilde{\tau}, \tilde{\boldsymbol{\nu}}) g(\boldsymbol{m})^{\tilde{\tau}} \exp(\tilde{\boldsymbol{\nu}}^\top \mathbf{n}(\boldsymbol{m})), \text{ where } \tilde{\tau} = \tau + N,\ \tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} + N\bar{\mathbf{s}}(\mathcal{D}). \tag{19}$$

Using $q(\boldsymbol{m})$, it outputs expected natural parameters $\bar{\mathbf{n}} = \mathbb{E}_{q(\boldsymbol{m})}[\mathbf{n}(\boldsymbol{m})]$.

2.5.4 Stochastic VB for CE Models

The VB update introduced in Eq. 18 is inefficient for large data sets because we need to optimise the variational posterior over the latent variables corresponding to each data point before re-estimating the variational posterior over the parameters. For more efficient learning, we adopt stochastic variational inference, which uses stochastic optimisation to fit the variational distribution over the parameters. We repeatedly subsample the data to form noisy estimates of the natural gradient of the variational lower bound, and we mix these estimates[10] with a decreasing step-size ρ_t, as in Hoffman et al. (2013)[11]. In particular, $\rho_t = (\tau_0 + t)^{-\kappa}$, where t denotes the optimization step; $\kappa \in (0.5, 1]$ controls how quickly the old information is forgotten; and τ_0 down-weights early iterations. The stochastic variational Bayes algorithm is summarised in Algorithm 1.

Algorithm 1 (Stochastic) Variational Bayes for CE family distributions

Input: Data $\mathcal{D}$. Define $\rho_t = (\tau_0 + t)^{-\kappa}$ and mini-batch size S.
Output: Expected natural parameters $\bar{\mathbf{n}}$ and expected sufficient statistics $\bar{\mathbf{s}}$.
for $t = 1, \ldots, J$ **do**
Draw a minibatch of S datapoints, without replacement.
(1) E-step: Given the expected natural parameters $\bar{\mathbf{n}}$, compute $q(\boldsymbol{l}_n)$ for $n = 1, \ldots, S$.
Output the expected sufficient statistics $\bar{\mathbf{s}} = \frac{1}{S} \sum_{n=1}^{S} \langle \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n) \rangle_{q(\boldsymbol{l}_n)}$.
(2) M-step: Given $\bar{\mathbf{s}}$, compute $q(\boldsymbol{m})$ by $\tilde{\boldsymbol{\nu}}^{(t)} = \boldsymbol{\nu} + N\bar{\mathbf{s}}$. Set $\tilde{\boldsymbol{\nu}}^{(t)} \leftarrow\!\!\rightarrow (1-\rho_t)\tilde{\boldsymbol{\nu}}^{(t-1)} + \rho_t \tilde{\boldsymbol{\nu}}^{(t)}$.
Output the expected natural parameters $\bar{\mathbf{n}} = \langle \mathbf{n}(\boldsymbol{m}) \rangle_{q(\boldsymbol{m})}$.
end for

10. This mixing step is in the M-step in Algorithm 1, where we first compute $\tilde{\boldsymbol{\nu}}^{(t)}$ then mix it with the previous step's estimate $\tilde{\boldsymbol{\nu}}^{(t-1)}$ via $\tilde{\boldsymbol{\nu}}^{(t)} \leftarrow\!\!\rightarrow (1-\rho_t)\tilde{\boldsymbol{\nu}}^{(t-1)} + \rho_t \tilde{\boldsymbol{\nu}}^{(t)}$.
11. When optimising over a probability distribution, the Euclidean distance between two parameter vectors is often a poor measure of the dissimilarity of the distributions. The natural gradient of a function accounts for the information geometry of its parameter space, using a Riemannian metric to adjust the direction of the traditional gradient, which results in a faster convergence than the traditional gradient (Hoffman et al., 2013).

3. Variational Bayes In Private Settings (VIPS) for the CE Family

To create an extension of variational Bayes which preserves differential privacy, we need to inject noise into the algorithm. The design choices for the noise injection procedure must be carefully made, as they can strongly affect the statistical efficiency of the algorithm, in terms of its accuracy versus the number of samples in the dataset. We start by introducing our problem setup.

3.1 Problem Setup

A naive way to privatise the VB algorithm is by perturbing both $q(\boldsymbol{l})$ and $q(\boldsymbol{m})$. Unfortunately, this is impractical, due to the excessive amounts of additive noise (recall: we typically have as many latent variables as the number of datapoints). We propose to perturb the expected sufficient statistics *only*. What follows next explains why this makes sense.

While the VB algorithm is being run, the places where the algorithm needs to look at the data are (1) when computing the variational posterior over the latent variables $q(\boldsymbol{l})$; and (2) when computing the expected sufficient statistics $\bar{\mathbf{s}}(\mathcal{D})$ using this computed $q(\boldsymbol{l})$. These are all happening during the E-step. In our proposed approach, we compute $q(\boldsymbol{l})$ behind the *privacy wall* (see below), and compute the expected sufficient statistics using $q(\boldsymbol{l})$, as shown in Fig. 1. Before outputting the expected sufficient statistics, we perturb each coordinate of the expected sufficient statistics to compensate the maximum difference in $\langle \mathbf{s}_l(\mathcal{D}_n, \boldsymbol{l}_n) \rangle_{q(\boldsymbol{l}_n)}$ caused by both $\mathcal{D}_n$ and $q(\boldsymbol{l}_n)$. The perturbed expected sufficient statistics then dictate the expected natural parameters in the M-step. Hence we do not need an additional step to add noise to $q(\boldsymbol{m})$.

The reason we neither perturb nor output $q(\boldsymbol{l})$ for training data is that we do not need $q(\boldsymbol{l})$ itself most of the time. For instance, when computing the predictive probability for test datapoints $\mathcal{D}_{tst}$, we need to perform the E-step to obtain the variational posterior for the test data $q(\boldsymbol{l}_{tst})$, which is a function of the test data and the expected natural parameters $\bar{\mathbf{n}}$, given as

$$p(\mathcal{D}_{tst}|\mathcal{D}) = \int d\boldsymbol{m} d\boldsymbol{l}_{tst}\; p(\mathcal{D}_{tst}|\boldsymbol{l}_{tst}, \boldsymbol{m})\; q(\boldsymbol{l}_{tst}; \mathcal{D}_{tst}, \bar{\mathbf{n}})\; q(\boldsymbol{m}; \tilde{\boldsymbol{\nu}}), \tag{20}$$

where the dependence on the training data $\mathcal{D}$ is implicit in the approximate posteriors $q(\boldsymbol{m})$ through $\tilde{\boldsymbol{\nu}}$; and the expected natural parameters $\bar{\mathbf{n}}$. Hence, outputting the perturbed sufficient statistics and the expected natural parameters suffice for protecting the privacy of individuals in the training data. Furthermore, the M-step can be performed based on the (privatised) output of the E-step, without querying the data again, so we do not need to add any further noise to the M-step to ensure privacy, due to the fact that differential privacy is immune to data-independent post-processing. To sum up, we provide our problem setup as below.

1. Privacy wall: We assume that the sensitive training dataset is only accessible through a sanitising interface which we call a *privacy wall*. The training data stay behind the privacy wall, and adversaries have access to the outputs of our algorithm only, i.e., no direct access to the training data, although they may have further prior knowledge on some of the individuals that are included in the training data.

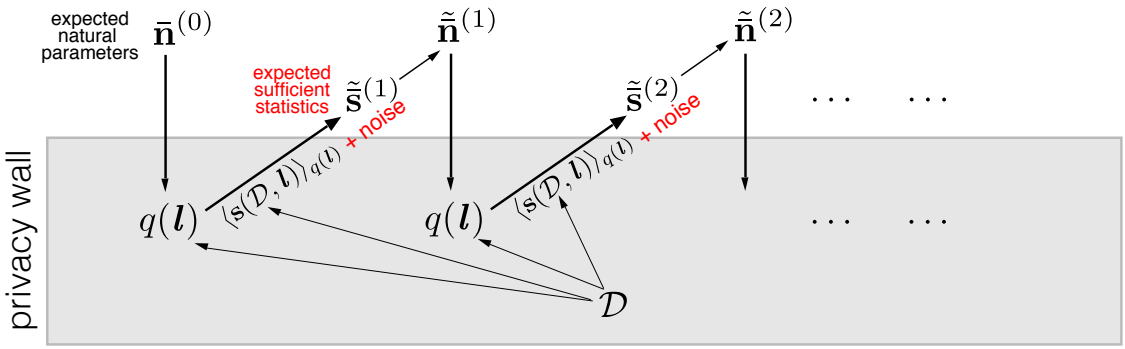

Figure 1: Schematic of VIPS. Given the initial expected natural parameters $\bar{\mathbf{n}}^{(0)}$, we compute the variational posterior over the latent variables $q(\boldsymbol{l})$. Since $q(\boldsymbol{l})$ is a function of not only the expected natural parameters but also the data $\mathcal{D}$, we compute $q(\boldsymbol{l})$ behind the privacy wall. Using $q(\boldsymbol{l})$, we then compute the expected sufficient statistics. Note that we neither perturb nor output $q(\boldsymbol{l})$ itself. Instead, when we noise up the expected sufficient statistics before outputting, we add noise to each coordinate of the expected sufficient statistics in order to compensate the maximum difference in $\langle \mathbf{s}_l(\mathcal{D}_n, \boldsymbol{l}_n) \rangle_{q(\boldsymbol{l}_n)}$ caused by both $\mathcal{D}_n$ and $q(\boldsymbol{l}_n)$. In the M-step, we compute the variational posterior over the parameters $q(\boldsymbol{m})$ using the perturbed expected sufficient statistics $\tilde{\bar{\mathbf{s}}}^{(1)}$. Using $q(\boldsymbol{m})$, we compute the expected natural parameters $\tilde{\bar{\mathbf{n}}}^{(1)}$, which is already perturbed since it is a function of $\tilde{\bar{\mathbf{s}}}^{(1)}$. We continue performing these two steps until convergence.

2. Training phase: Our differentially private VB algorithm releases the perturbed expected natural parameters and perturbed expected sufficient statistics in every iteration. Every release of a perturbed parameter based on the training data triggers an update in the log moment functions (see Sec 3.2 on how these are updated). At the end of the training phase, the final privacy parameters are calculated based on (4).

3. Test phase: Test data are public (or belong to users), i.e., outside the privacy wall. Bayesian prediction on the test data is possible using the released expected natural parameters and expected sufficient statistics (given as Eq. 20). Note that we do not consider protecting the privacy of the individuals in the test data.

3.2 How to Compute the Log Moment Functions?

Recall that to use the moments accountant method, we need to compute the log moment functions $\alpha_{\mathcal{M}_t}$ for each individual iteration t. An iteration t of VIPS randomly subsamples a υ fraction of the dataset, without replacement, and uses it to compute the sufficient statistics which is then perturbed via the Gaussian mechanism with some variance. How the log moment function is computed depends on the sensitivity of the sufficient statistics as well as the underlying mechanism; we next discuss each of these aspects.

3.2.1 Sensitivity Analysis

Suppose there are two neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$, where $\mathcal{D}'$ has one entry difference compared to $\mathcal{D}$ (i.e., by replacing one entry with a different value).

We denote the vector of expected sufficient statistics by $\bar{\mathbf{s}}(\mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N}\bar{\mathbf{s}}(\mathcal{D}_n) = [M_1, \cdots, M_L]$, where each expected sufficient statistic M_l is given by

$$M_l = \frac{1}{N}\sum_{n=1}^{N}\langle \mathbf{s}_l(\mathcal{D}_n, \boldsymbol{l}_n)\rangle_{q(\boldsymbol{l}_n)} \ . \tag{21}$$

When computing the sensitivity of the expected sufficient statistics, we assume, without loss of generality, the last entry is the one which differs between $\mathcal{D}$ and $\mathcal{D}'$. Under this assumption and the *i.i.d.* assumption on the likelihood, given the current expected natural parameters $\bar{\mathbf{n}}$, all $q(\boldsymbol{l}_n)$ and $\mathbf{s}_l(\mathcal{D}_n, \boldsymbol{l}_n)$ for $n \in \{1, \ldots, N-1\}$ evaluated on the dataset $\mathcal{D}$ are the same as $q(\boldsymbol{l}'_n)$ and $\mathbf{s}_l(\mathcal{D}'_n, \boldsymbol{l}'_n)$ evaluated on the dataset $\mathcal{D}'$. Hence, the sensitivity is given by

$$\Delta M_l = \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |M_l(\mathcal{D}) - M_l(\mathcal{D}')| \leq \max_{\mathcal{D}_N, q(\boldsymbol{l}_N)} \frac{2}{N}|\mathbb{E}_{q(\boldsymbol{l}_N)}\mathbf{s}_l(\mathcal{D}_N, \boldsymbol{l}_N)|. \tag{22}$$

See Appendix Sec. 1 for the proof.

As in many existing works (e.g., Chaudhuri et al., 2011; Kifer et al., 2012, among many others), we also assume that the dataset is pre-processed such that the L_2 norm of any $\mathcal{D}_n$ is less than 1. Furthermore, we choose $q(\boldsymbol{l}_n)$ such that its support is bounded. Under these conditions, each coordinate of the expected sufficient statistics $\langle \mathbf{s}_l(\mathcal{D}_n, \boldsymbol{l}_n)\rangle_{q(\boldsymbol{l}_n)}$ has limited sensitivity. We will add noise to each coordinate of the expected sufficient statistics to compensate this bounded maximum change in the variational E-step.

3.2.2 Log Moment Function of the Subsampled Gaussian Mechanism

Observe that iteration t of our algorithm subsamples a υ fraction of the dataset, computes the sufficient statistics based on this subsample, and perturbs it using the Gaussian mechanism with variance $\Delta^2\sigma^2 I_d$, where Δ is the L2-sensitivity. Here, the subsampling is done by uniformly sampling S datapoints from the total N training samples without replacement, which makes the sampling rate $\upsilon = S/N$.

From Proposition 1.6 of Bun & Steinke (2016) along with simple algebra, the log moment function of the Gaussian Mechanism $\mathcal{M}$ is $\alpha_{\mathcal{M}}(\lambda) = \frac{\lambda(\lambda+1)\Delta^2}{2\sigma^2}$. In this case, with a pre-defined σ and a fixed Δ, we compute the log moments over some values of λ and obtain the corresponding privacy loss by converting the log moments to (ϵ, δ) that satisfies (4).

To compute the log moment function for the *Subsampled Gaussian Mechanism*, we use the Analytical Moments Accountant Method that was proposed by the recent work of Wang et al. (2019).[12] A more detailed description of this method is provided in Section 2.3.2.

Note that our algorithms are run for a pre-specified number of iterations, and with a pre-specified σ, and a fixed subsampling rate υ. Hence, we compute the privacy loss at the end of training once *post hoc* using the analytical moments accountant method. This

12. Code is available at `https://github.com/yuxiangw/autodp/tree/master/autodp`.

is different from the case where a data-dependent adaptive analysis such as that of Rogers, Roth, Ullman & Vadhan (2016) is required.

Also, note that when using the subsampled data per iteration, the sensitivity analysis has to be modified as now the query is evaluated on a smaller dataset. Hence, the $1/N$ factor has to be changed to $1/S$ in Eq. 22, as the two subsampled datasets can have maximally a single datapoint difference out of S number of data samples:

$$\Delta M_l \leq \max_{\mathcal{D}_S, q(\boldsymbol{l}_S)} \tfrac{2}{S} |\mathbb{E}_{q(\boldsymbol{l}_S)} \mathbf{s}_l(\mathcal{D}_S, \boldsymbol{l}_S)| \ . \tag{23}$$

Algorithm 2 summarizes our VIPS algorithm that performs differentially private stochastic variational Bayes for CE family models.

Algorithm 2 Private VIPS for CE family distributions

Input: Data $\mathcal{D}$. Define $\rho_t = (\tau_0 + t)^{-\kappa}$, noise variance σ^2, mini-batch size S, and maximum iterations J.

Output: Perturb expected natural parameters $\tilde{\bar{\mathbf{n}}}$ and expected sufficient statistics $\tilde{\bar{\mathbf{s}}}$.

Compute the L2-sensitivity Δ of the expected sufficient statistics.

for $t = 1, \ldots, J$ **do**

Draw a minibatch of S datapoints, without replacement.

(1) E-step: Given the expected natural parameters $\bar{\mathbf{n}}$, compute $q(\boldsymbol{l}_n)$ for $n = 1, \ldots, S$. Perturb each coordinate of $\bar{\mathbf{s}} = \frac{1}{S} \sum_{n=1}^{S} \langle \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n) \rangle_{q(\boldsymbol{l}_n)}$ by adding $\mathcal{N}(0, \sigma^2 \Delta^2 I)$ noise, and output $\tilde{\bar{\mathbf{s}}}$.

(2) M-step: Given $\tilde{\bar{\mathbf{s}}}$, compute $q(\boldsymbol{m})$ by $\tilde{\boldsymbol{\nu}}^{(t)} = \boldsymbol{\nu} + N\tilde{\bar{\mathbf{s}}}$. Set $\tilde{\boldsymbol{\nu}}^{(t)} \leftarrow\!\!\rightarrow (1-\rho_t)\tilde{\boldsymbol{\nu}}^{(t-1)} + \rho_t \tilde{\boldsymbol{\nu}}^{(t)}$. Output the expected natural parameters $\tilde{\bar{\mathbf{n}}} = \langle \mathbf{n}(\boldsymbol{m}) \rangle_{q(\boldsymbol{m})}$.

end for

Compute the privacy loss $(\epsilon_{tot}, \delta_{tot})$ using the analytical moments account method (Wang et al., 2019).

4. VIPS for Latent Dirichlet Allocation

Here, we illustrate how to use the general framework of VIPS for CE family in the example of the Latent Dirichlet Allocation (LDA) topic model (Blei et al., 2003). The LDA model takes as input a corpus of text documents, and aims to identify the semantic themes ("*topics*") which are present in the corpus. A *topic* is a discrete distribution over all of the words in the vocabulary, where the high probability words are assumed to be semantically coherent representatives of a particular theme. For example, a topic on baseball might give high probability to the words *pitcher*, *bat*, and *base*. By finding the topics in the text corpus, LDA automatically provides a "birds-eye view" semantic summary of the dataset which can be interpreted by human analysts. The model can also be used for dimensionality reduction on the documents themselves, by representing each document according to extent to which the inferred topics are present.

Symbol	Definition
	Indices and counts
D	Number of documents
N	Number of word tokens in a document
K	Number of topics
V	Vocabulary size
d	Indexes the dth document (indices for LDA may be sub- or super-scripts)
n	Indexes the nth word in a document
k	Indexes the kth topic
v	Indexes the vth vocabulary word
$\mathbf{1}_L$	A vector of ones of length L, for any integer L
	Input data and hyperparameters
$\mathcal{D}$	Dataset containing D documents $\{\mathcal{D}_d\}$
$\mathbf{w}_{dn}$	Indicator vector of length V for the nth word in document d
$\mathbf{n}_d$	Vector of word counts for document d
η	Dirichlet prior hyperparameter for topics (scalar)
α	Dirichlet prior hyperparameter for documents' topic distributions (scalar)
	Quantities to be inferred
$\boldsymbol{\beta}$	Topics
$\boldsymbol{\theta}$	Distribution over topics for documents
$\mathbf{z}_{dn}$	Indicator vector of length K, topic assignment of the nth word in document d
$\boldsymbol{\phi}$	Variational parameters for topic assignments
$\boldsymbol{\gamma}$	Variational parameters for distributions over topics
$\boldsymbol{\lambda}$	Variational parameters for topics
$\bar{\mathbf{s}}$	Expected sufficient statistics, a matrix of size $K \times V$
$\tilde{\bar{\mathbf{s}}}$	Privatized expected sufficient statistics

Table 2: Summary of the main notation for LDA.

4.1 Model Specifics for LDA

In the LDA topic model, we observe a corpus of D documents $\mathcal{D}_d$, where each observed word is represented by an indicator vector $\mathbf{w}_{dn}$ (nth word in the dth document) of length V, and where V is the number of terms in a fixed vocabulary set (see Table 2 for a summary of the main notation used in this section). Given the corpus, the model infers K latent topics $\boldsymbol{\beta}_k$, which are discrete distributions over the vocabulary, and discrete distributions over topics $\boldsymbol{\theta}_d$ for each document $\mathcal{D}_d$. Each word $\mathbf{w}_{dn}$ is given a topic assignment latent variable $\mathbf{z}_{dn}$, represented by an indicator vector of length K. Let $\mathbf{1}_L$ be a vector of ones of length L, for any integer L. The LDA model posits that the generative process for the corpus as follows:

- Draw topics $\boldsymbol{\beta}_k \sim$ Dirichlet $(\eta \mathbf{1}_V)$, for $k = \{1, \ldots, K\}$, where η is a scalar hyperparameter.
- For each document $\mathcal{D}_d, d \in \{1, \ldots, D\}$

- Draw topic proportions $\boldsymbol{\theta}_d \sim \text{Dirichlet}\,(\alpha \mathbf{1}_K)$, where α is a scalar hyperparameter.
- For each word $n \in \{1, \ldots, N\}$
  - Draw topic assignments $\mathbf{z}_{dn} \sim \text{Discrete}(\boldsymbol{\theta}_d)$
  - Draw word $\mathbf{w}_{dn} \sim \text{Discrete}(\boldsymbol{\beta}_{\mathbf{z}_{dn}})$.

The LDA model falls in the CE family, viewing $\mathbf{z}_{d,1:N}$ and $\boldsymbol{\theta}_d$ as two types of latent variables: $\boldsymbol{l}_d = \{\mathbf{z}_{d,1:N}, \boldsymbol{\theta}_d\}$, and $\boldsymbol{\beta}$ as model parameters $\boldsymbol{m} = \boldsymbol{\beta}$.[13] The conditions for CE are satisfied: (1) the complete-data likelihood per document is in exponential family:

$$p(\mathbf{w}_{d,1:N}, \mathbf{z}_{d,1:N}, \boldsymbol{\theta}_d | \boldsymbol{\beta}) \propto f(\mathcal{D}_d, \mathbf{z}_{d,1:N}, \boldsymbol{\theta}_d) \exp(\sum_n \sum_k [\log \boldsymbol{\beta}_k]^\top [\mathbf{z}_{dn}^k \mathbf{w}_{dn}]), \tag{24}$$

where $f(\mathcal{D}_d, \mathbf{z}_{d,1:N}, \boldsymbol{\theta}_d) \propto \exp([\alpha \mathbf{1}_K]^\top [\log \boldsymbol{\theta}_d] + \sum_n \sum_k \mathbf{z}_{dn}^k \log \boldsymbol{\theta}_d^k)$; and (2) we have a conjugate prior over $\boldsymbol{\beta}_k$:

$$p(\boldsymbol{\beta}_k | \eta \mathbf{1}_V) \propto \exp([\eta \mathbf{1}_V]^\top [\log \boldsymbol{\beta}_k]), \tag{25}$$

for $k = \{1, \ldots, K\}$. For simplicity, we assume hyperparameters α and η are set manually.

Under the LDA model, we assume the variational posteriors are given by

- Discrete : $q(\mathbf{z}_{dn}^k | \boldsymbol{\phi}_{dn}^k) \propto \exp(\mathbf{z}_{dn}^k \log \boldsymbol{\phi}_{dn}^k)$, with variational parameters for capturing the posterior topic assignment,

$$\boldsymbol{\phi}_{dn}^k \propto \exp(\langle \log \boldsymbol{\beta}_k \rangle_{q(\boldsymbol{\beta}_k)}{}^\top \mathbf{w}_{dn} + \langle \log \boldsymbol{\theta}_d^k \rangle_{q(\boldsymbol{\theta}_d)}). \tag{26}$$

- Dirichlet : $q(\boldsymbol{\theta}_d | \boldsymbol{\gamma}_d) \propto \exp(\boldsymbol{\gamma}_d{}^\top \log \boldsymbol{\theta}_d)$, where $\boldsymbol{\gamma}_d = \alpha \mathbf{1}_K + \sum_{n=1}^N \langle \mathbf{z}_{dn} \rangle_{q(\mathbf{z}_{dn})}$,

where these two distributions are computed in the E-step behind the privacy wall. By comparing the exponential family form of the LDA model (Equation 24) with the general form of the exponential family (Equation 14) and taking the expectation over the variational distribution, we observe that the expected sufficient statistics are $\bar{\mathbf{s}}_k = \frac{1}{D} \sum_d \sum_n \langle \mathbf{z}_{dn}^k \rangle_{q(\mathbf{z}_{dn})} \mathbf{w}_{dn} = \frac{1}{D} \sum_d \sum_n \boldsymbol{\phi}_{dn}^k \mathbf{w}_{dn}$ where we compute the expectation over $\mathbf{z}_{dn}^k$ using the posterior topic assignment probability we computed in Eq. 26. Then, in the M-step, we compute the posterior

- Dirichlet : $q(\boldsymbol{\beta}_k | \boldsymbol{\lambda}_k) \propto \exp(\boldsymbol{\lambda}_k{}^\top \log \boldsymbol{\beta}_k)$, where $\boldsymbol{\lambda}_k = \eta \mathbf{1}_V + \sum_d \sum_n \langle \mathbf{z}_{dn}^k \rangle_{q(\mathbf{z}_{dn})} \mathbf{w}_{dn}$.

4.2 VIPS for LDA

We follow the general framework of VIPS for differentially private LDA, with the addition of several LDA-specific heuristics which are important for good performance, described below. First, while each document originally has a different document length N_d, in order to bound the sensitivity, and to ensure that the signal-to-noise ratio remains reasonable for very short documents, we preprocess all documents to have the same fixed length N. We accomplish this by sampling N words with replacement from each document's bag of words. In our experiments, we use $N = 500$. Note that since privacy is preserved at the level of documents

13. For more detailed derivations of the exponential family form and variational posteriors for LDA, we refer the reader to Hoffman et al. (2013).

rather than at the level of words, this step does not directly impact the degree of privacy achieved, but it can reduce the relative amount of noise compared to the amount of data.

To perturb the expected sufficient statistics, which is a matrix of size $K \times V$, we add Gaussian noise to each component of this matrix:

$$\tilde{\bar{\mathbf{s}}}_k^v = \bar{\mathbf{s}}_k^v + Y_k^v, \text{ where } Y_k^v \sim \mathcal{N}(0, \sigma^2(\Delta\bar{\mathbf{s}})^2), \tag{27}$$

where $\bar{\mathbf{s}}_k^v = \frac{1}{S}\sum_d \sum_n \boldsymbol{\phi}_{dn}^k \mathbf{w}_{dn}^v$, and $\Delta\bar{\mathbf{s}}$ is the sensitivity. We then map the perturbed components to 0 if they become negative. For LDA, with a mini-batch of S documents (which changes D to S when computing the sensitivity) the worst-case sensitivity is given by

$$\Delta\bar{\mathbf{s}} = \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} \sqrt{\sum_k \sum_v (\bar{\mathbf{s}}_k^v(\mathcal{D}) - \bar{\mathbf{s}}_k^v(\mathcal{D}'))^2} \leq \max_{\boldsymbol{\phi}_{Sn}^k, \mathbf{w}_{Sn}^v} \frac{1}{S}\sum_n (\sum_k \boldsymbol{\phi}_{Sn}^k)(\sum_v \mathbf{w}_{Sn}^v) \leq \frac{N}{S}, \tag{28}$$

see Appendix Sec. 2 for proof. As we shall see, it is worthwhile to consider this sensitivity calculation more closely. We can write the mini-batch's expected sufficient statistics matrix $\bar{\mathbf{s}}$ in the form $\bar{\mathbf{s}} = \sum_d \bar{\mathbf{s}}^d$, a sum of contributions from each document d in the mini-batch. Here, document d's contribution $\bar{\mathbf{s}}^d$ is a $K \times V$ matrix which has entries $\bar{\mathbf{s}}_k^{v,d} = \frac{1}{S}\sum_n \boldsymbol{\phi}_{dn}^k \mathbf{w}_{dn}^v$. The sensitivity $\Delta\bar{\mathbf{s}}$ is equal to the worst case norm of document S's contribution, $|\bar{\mathbf{s}}^S|$, which can be seen from the proof of Eq. 28 in Appendix Sec. 2, specifically the penultimate line of Equation 74. The worst case occurs when all N words in document S have the same word type v, and are hard-assigned to some topic k in the variational distribution, i.e. $\bar{\mathbf{s}}_k^{v,S} = \frac{N}{S}$, and all other entries are 0. We can see that this is a rather extreme scenario, which typically will not occur often in practice. In our practical implementation, we improve the sensitivity by exploiting the fact that for most typical documents, the contribution's norm $|\bar{\mathbf{s}}^d|$ will be smaller than the worst case norm $\frac{N}{S}$.

Specifically, inspired by Abadi et al. (2016), we apply a norm clipping strategy, in which the per-document contributions $\bar{\mathbf{s}}^d$ are clipped (or projected) such that $|\bar{\mathbf{s}}^d| \leq a\frac{N}{S}$, for a user-specified $a \in (0, 1]$. Note that when $a = 1$, no clipping is applied. For each document in the minibatch, if this criterion is not satisfied, we project the expected sufficient statistics $\bar{\mathbf{s}}^d$ down to the required norm via

$$\bar{\mathbf{s}}^d := \frac{aN}{S}\frac{\bar{\mathbf{s}}^d}{|\bar{\mathbf{s}}^d|} \ . \tag{29}$$

For example, consider a scenario where $N = V = K = 2$, $S = 1$, $a = 0.1$, and $\bar{\mathbf{s}}^d = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$. The worst-case norm is $\frac{N}{S} = 2$, the clipping threshold is calculated as $a\frac{N}{S} = 0.2$, and the norm of $\bar{\mathbf{s}}^d$ is $|\bar{\mathbf{s}}^d| = \sqrt{2} \approx 1.41 > 0.2$. The document's expected sufficient statistics hence are clipped to

$$\bar{\mathbf{s}}^d := \frac{aN}{S}\frac{\bar{\mathbf{s}}^d}{|\bar{\mathbf{s}}^d|} = \frac{0.2}{\sqrt{(2)}}\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \ . \tag{30}$$

These clipped sufficient statistics have a norm of $|\bar{\mathbf{s}}^d| = 0.2$, as required. In this case, the pre-clipping document norm $|\bar{\mathbf{s}}^d|$ was well above the clipping threshold, and hence was

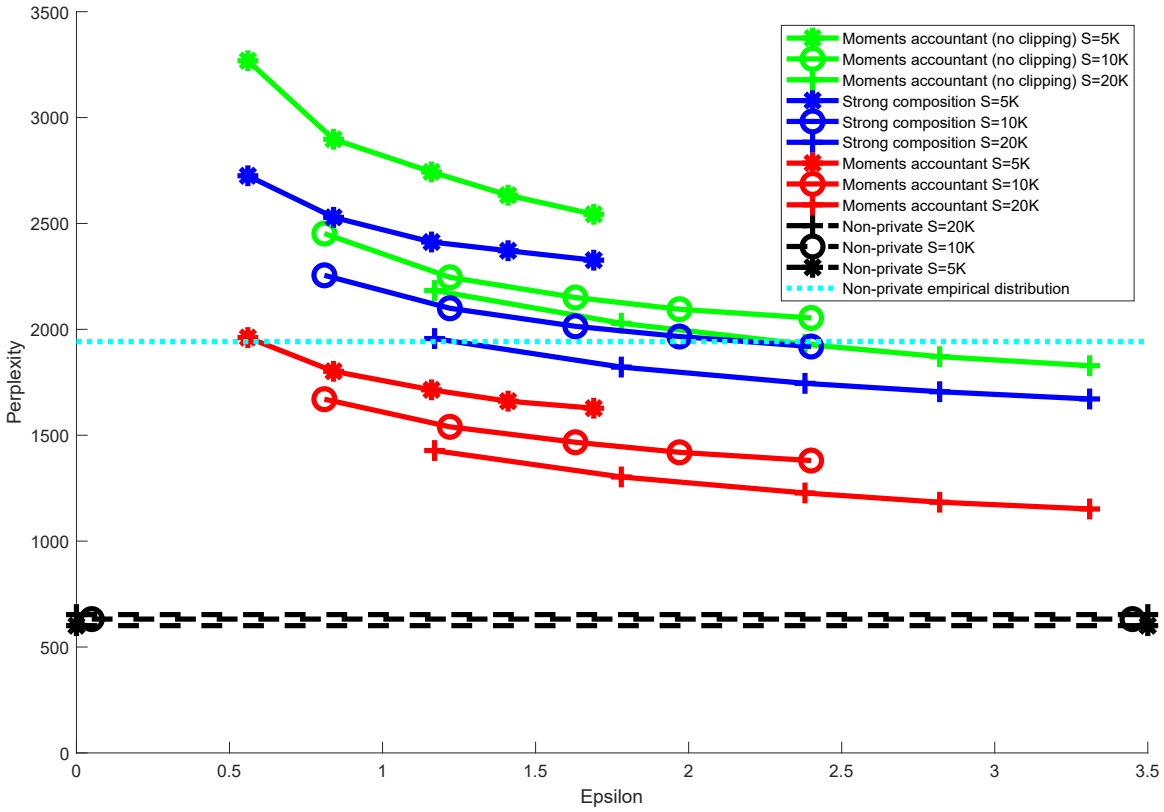

Figure 2: Epsilon versus perplexity, varying σ and S, Wikipedia data, one epoch. Perplexity is approximated using the upper bound of Hoffman et al. (2010) which is exact for the *non-private empirical distribution* baseline, hence this has a slightly unfair advantage.

substantially rescaled by the clipping procedure. However, this circumstance becomes substantially less common with a higher-dimensional $\bar{\mathbf{s}}^d$, as encountered in practice. Intuitively, for a fixed N and S, the L1 norm of $\bar{\mathbf{s}}^d$ is always N/S, but its L2 norm can become arbitrarily small as the dimensionality of $\bar{\mathbf{s}}^d$ increases. E.g., if all entries of $\bar{\mathbf{s}}^d$ are equal, the L2 norm approaches 0 in the limit as the number of entries increases. Thus, in higher dimensions, the clipping procedure can potentially affect relatively few documents, and to a lesser degree, even when a relatively aggressive value of the clipping constant a is selected. We thereby substantially reduce the amount of noise to obtain privacy, at the cost of only a small bias in the algorithm's behavior. Indeed, this is what we observe in practice, as we shall see below.

After this the procedure, the sensitivity of the clipped expected sufficient statistics matrix becomes $a\Delta\bar{\mathbf{s}}$ (i.e., $a\frac{N}{S}$), and we add noise on this scale to the clipped expected sufficient statistics. We set $a = 0.1$ in our experiments, which empirically resulted in clipping being applied to around 3/4 of the documents, while improving the sensitivity by an order of magnitude. The resulting algorithm is summarised in Algorithm 3.

Algorithm 3 VIPS for LDA

Input: Data $\mathcal{D}$. Define D (documents), V (vocabulary), K (number of topics).
Define $\rho_t = (\tau_0 + t)^{-\kappa}$, mini-batch size S, hyperparameters α, η, σ^2, and a
Output: Privatised expected natural parameters $\langle \log \boldsymbol{\beta}_k \rangle_{q(\boldsymbol{\beta}_k)}$ and sufficient statistics $\tilde{\bar{\mathbf{s}}}$.
for $t = 1, \ldots, J$ **do**
Draw a minibatch of S documents, without replacement.
(1) E-step: Given expected natural parameters $\langle \log \boldsymbol{\beta}_k \rangle_{q(\boldsymbol{\beta}_k)}$
for $d = 1, \ldots, S$ **do**
while (not converged) **do**
\\Internal E-step iterations are performed behind the privacy wall:
Compute $q(\mathbf{z}_{dn}^k)$ parameterised by $\boldsymbol{\phi}_{dn}^k \propto \exp(\langle \log \boldsymbol{\beta}_k \rangle_{q(\boldsymbol{\beta}_k)}^\top \mathbf{w}_{dn} + \langle \log \boldsymbol{\theta}_d^k \rangle_{q(\boldsymbol{\theta}_d)})$.
Compute $q(\boldsymbol{\theta}_d)$ parameterised by $\boldsymbol{\gamma}_d = \alpha \mathbf{1}_K + \sum_{n=1}^N \langle \mathbf{z}_{dn} \rangle_{q(\mathbf{z}_{dn})}$.
end while
Compute per-document expected sufficient statistics $\bar{\mathbf{s}}_k^{v,d} = \frac{1}{S} \sum_n \boldsymbol{\phi}_{dn}^k \mathbf{w}_{dn}^v$.
if $|\bar{\mathbf{s}}^d| > aN/S$ **then**
$\bar{\mathbf{s}}^d := \frac{aN}{S} \frac{\bar{\mathbf{s}}^d}{|\bar{\mathbf{s}}^d|}$.
end if
end for
Compute the expected sufficient statistics $\bar{\mathbf{s}} = \sum_d \bar{\mathbf{s}}^d$.
Output the perturbed expected sufficient statistics $\tilde{\bar{\mathbf{s}}}_k^v = \bar{\mathbf{s}} + Y_k^v$, where Y_k^v is Gaussian noise given in Eq. 27, but using clipped sensitivity $a\frac{N}{S}$.
Clip negative entries of $\tilde{\bar{\mathbf{s}}}$ to 0.
(2) M-step: Given perturbed expected sufficient statistics $\tilde{\bar{\mathbf{s}}}_k$,
Compute $q(\boldsymbol{\beta}_k)$ parameterised by $\boldsymbol{\lambda}_k^{(t)} = \eta \mathbf{1}_V + D\tilde{\bar{\mathbf{s}}}_k$.
Set $\boldsymbol{\lambda}^{(t)} \leftarrow\!\!\rightarrow (1 - \rho_t)\boldsymbol{\lambda}^{(t-1)} + \rho_t \boldsymbol{\lambda}^{(t)}$.
Output expected natural parameters $\langle \log \boldsymbol{\beta}_k \rangle_{q(\boldsymbol{\beta}_k)}$.
end for
Compute the privacy loss $(\epsilon_{tot}, \delta_{tot})$ using the analytical moments account method (Wang et al., 2019).

4.3 Experiments using Wikipedia Data

We downloaded a random $D = 400,000$ documents from Wikipedia to test our VIPS algorithm. We used 50 topics and a vocabulary set of approximately 8000 terms. The algorithm was run for one epoch in each experiment.

We compared our moments accountant approach with a baseline method using the *strong composition* (Theorem 3.20 of Dwork & Roth, 2014), resulting from the max divergence of the privacy loss random variable being bounded by a total budget including an adjustable slack variable δ'', which yields $(J\epsilon'(e^{\epsilon'} - 1) + \sqrt{2J\log(1/\delta'')}\epsilon',\ \delta'' + J\delta')$-DP, where J is the number of iterations, and (ϵ', δ') is the privacy loss we allow to incur in every learning step. We also compared to a baseline where the clipping step was not performed. As our evaluation metric, we compute an upper bound on the perplexity on held-out documents. Perplexity is an information-theoretic measure of the predictive performance of probabilistic models which is commonly used in the context of language modeling (Jelinek, Mercer, Bahl

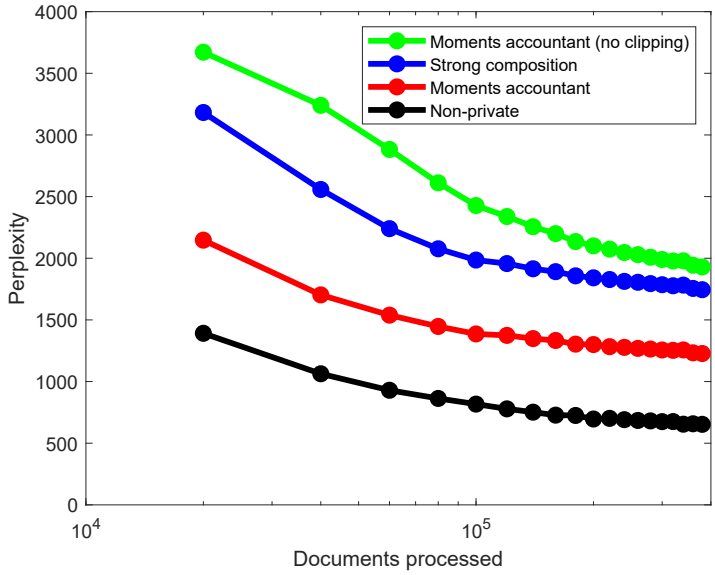

Figure 3: Perplexity per iteration. Wikipedia data, $S = 20,000$, $\epsilon = 2.38$, one epoch.

& Baker, 1977). The perplexity of a probabilistic model $p_{model}(x)$ on a test set of N data points x_i (e.g. words in a corpus) is defined as

$$b^{-\frac{1}{N}\sum_{i=1}^{N}\log_b p_{model}(x_i)}\ , \tag{31}$$

where b is generally either 2 or e, corresponding to a measurement based on either bits or nats, respectively.[14] In our case, p_{model} is the posterior predictive distribution under our variational approximation to the posterior, which is intractable to compute. Following Hoffman et al. (2010), we approximate perplexity based on the learned variational distribution, measured in nats, by plugging the ELBO into Equation 31, which results in an upper bound:

$$\begin{aligned}&\text{perplexity}(\mathcal{D}^{test}, \boldsymbol{\lambda}) \\ &\leq \exp\left[-\left(\sum_d \langle \log p(\mathbf{n}^{test}, \boldsymbol{\theta}_d, \mathbf{z}_d|\boldsymbol{\lambda})\rangle_{q(\boldsymbol{\theta}_d,\mathbf{z}_d)} - \langle \log q(\boldsymbol{\theta},\mathbf{z})\rangle_{q(\boldsymbol{\theta},\mathbf{z})}\right) / \sum_{d,n}\mathbf{n}_{d,n}^{test}\right],\end{aligned} \tag{32}$$

where $\mathbf{n}_d^{test}$ is a vector of word counts for the dth document, $\mathbf{n}^{test} = \{\mathbf{n}_d^{test}\}_{d=1}^{I}$, for I test documents. In the above, we use the $\boldsymbol{\lambda}$ that was calculated during training. We compute the posteriors over $\mathbf{z}$ and $\boldsymbol{\theta}$ by performing the first step in our algorithm using the test data

14. We can interpret $-\frac{1}{N}\sum_{i=1}^{N}\log_b p_{model}(x_i)$ as the cross-entropy between the model and the empirical distribution. This is the expected number of bits (nats) needed to encode a data point from the empirical data distribution (i.e., a word in our case), under an optimal code based on the model. Perplexity is b to the power of the cross entropy, which converts the number of bits (nats) in the encoding to the number of possible values in an encoding of that length (supposing the cross entropy were integer valued). Thus, perplexity measures the *effective vocabulary size when using the model to encode the data*, which is understood as a reflection of how confused (i.e. "perplexed") the model is.

and the perturbed sufficient statistics we obtain during training. We adapted the Python implementation by the authors of Hoffman et al. (2010) for our experiments.

Figure 2 shows the trade-off between ϵ and per-word perplexity on the Wikipedia dataset for the different methods under a variety of conditions, in which we varied the noise level $\sigma \in \{1.0, 1.1, 1.24, 1.5, 2\}$ and the minibatch size $S \in \{5,000, 10,000, 20,000\}$. The strong composition baseline was performed using the same minibatch sizes, and with σ set corresponding to the moments accountant's ϵ. We found that the moments accountant composition substantially outperformed strong composition in each of these settings. The moments accountant was able to consistently beat a simple non-private baseline, predicting based on empirical word frequencies, while strong composition only outperformed this baseline for certain parameter settings. Our proposed method, which implements the clipping procedure described in the previous section, also outperformed a baseline in which the clipping step was not applied. The no-clipping baseline, which used the moments accountant, consistently performed slightly worse than strong composition (for which clipping was applied).

Here, we used relatively large minibatches, which were necessary to control the signal-to-noise ratio in order to obtain reasonable results for private LDA. Larger minibatches thus had lower perplexity. However, due to its impact on the subsampling rate, increasing S comes at the cost of a higher ϵ for a fixed number of documents processed (in our case, one epoch). To show the convergence behavior of the algorithms, we also report the perplexity per iteration in Figure 3. We found that the gaps between methods remain relatively constant at each stage of the learning process, with the exception that the no-clipping moments accountant baseline gradually approached the performance of strong composition (with clipping). Similar results were observed for other values of σ and S.

In Table 3, for each method, we show the top 10 words in terms of assigned probabilities for 3 topics. Non-private LDA results in the most coherent words among all the methods. For the private LDA models with a total privacy budget $\epsilon = 2.38$ ($S = 20,000, \sigma = 1.24$), as we move from moments accountant to strong composition, the amount of noise added gets larger, and the topics become less coherent. We also observe that the probability mass assigned to the most probable words decreases with the noise, and thus strong composition gave less probability to the top words compared to the other methods. For comparison, we also show the results for the moments accountant without clipping. In this case, the topics were observably less coherent than for the same method where clipping was applied, and were somewhat comparable in qualitative performance to strong composition (with clipping). This was consistent with the corresponding perplexity results of these two methods, which were fairly similar at the final iteration (Figure 3).

5. VIPS for Non-CE Family

Under non-CE family models, the complete-data likelihood typically has the following form:

$$p(\mathcal{D}_n, \boldsymbol{l}_n | \boldsymbol{m}) \propto \exp(-h(\boldsymbol{m}, \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n))) \ , \tag{33}$$

which includes some function $h(\boldsymbol{m}, \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n))$ that cannot be split into two functions, where one is a function of only $\boldsymbol{m}$ and the other is a function of only $\mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)$. Hence, we cannot apply the general VIPS framework we described in the previous section to this case.

Table 3: Posterior topics from private ($\epsilon = 2.38$) and non-private LDA

Non-private		Moments Accountant		Strong Composition		Moments Accountant (no clipping)	
topic 33:		topic 33:		topic 33:			
german	0.0244	function	0.0019	resolution	0.0003	resolution	0.0003
system	0.0160	domain	0.0017	northward	0.0003	northward	0.0003
group	0.0109	german	0.0011	deeply	0.0003	messages	0.0003
based	0.0089	windows	0.0011	messages	0.0003	deeply	0.0003
science	0.0077	software	0.0010	dark	0.0003	river	0.0003
systems	0.0076	band	0.0007	research	0.0003	research	0.0003
computer	0.0072	mir	0.0006	river	0.0003	superstition	0.0003
software	0.0071	product	0.0006	superstition	0.0003	don	0.0003
space	0.0061	resolution	0.0006	don	0.0003	dark	0.0003
power	0.0060	identity	0.0005	found	0.0003	tete	0.0003
topic 35:		topic 35:		topic 35:			
station	0.0846	station	0.0318	station	0.0116	station	0.0083
line	0.0508	line	0.0195	line	0.0062	line	0.0046
railway	0.0393	railway	0.0149	railway	0.0054	french	0.0042
opened	0.0230	opened	0.0074	opened	0.0021	railway	0.0037
services	0.0187	services	0.0064	services	0.0015	opened	0.0014
located	0.0163	closed	0.0056	stations	0.0014	services	0.0011
closed	0.0159	code	0.0054	closed	0.0014	republic	0.0009
owned	0.0158	country	0.0052	section	0.0013	closed	0.0009
stations	0.0122	located	0.0051	platform	0.0012	stations	0.0007
platform	0.0109	stations	0.0051	republic	0.0010	country	0.0007
topic 37:		topic 37:		topic 37:			
born	0.1976	born	0.0139	born	0.0007	american	0.0021
american	0.0650	people	0.0096	american	0.0006	born	0.0018
people	0.0572	notable	0.0092	street	0.0005	people	0.0013
summer	0.0484	american	0.0075	charles	0.0004	notable	0.0010
notable	0.0447	name	0.0031	said	0.0004	charles	0.0005
canadian	0.0200	mountain	0.0026	events	0.0004	italian	0.0005
event	0.0170	japanese	0.0025	people	0.0003	summer	0.0005
writer	0.0141	fort	0.0025	station	0.0003	events	0.0005
dutch	0.0131	character	0.0019	written	0.0003	actor	0.0004
actor	0.0121	actor	0.0014	point	0.0003	written	0.0004

However, when the models we consider have binomial likelihoods, for instance, under negative binomial regression, nonlinear mixed-effects models, spatial models for count data, and logistic regression, we can bring the non-CE models to the CE family by adopting the Pólya-Gamma data augmentation strategy introduced by Polson et al. (2013).

Pólya-Gamma Data Augmentation Pólya-Gamma data augmentation introduces an auxiliary variable that is Pólya-Gamma distributed per datapoint, such that the log-odds can be written as a mixture of Gaussians with respect to a Pólya-Gamma distribution, as stated in Theorem 1 of Polson et al. (2013):

$$\frac{\exp(\psi_n)^{y_n}}{(1+\exp(\psi_n))^b} \quad = \quad 2^{-b}\exp((y_n-\tfrac{b}{2})\psi_n)\int_0^\infty \exp(-\tfrac{\xi_n\psi_n^2}{2})\; p(\xi_n)\; d\xi_n \tag{34}$$

where ψ_n is a linear function in model parameters $\boldsymbol{m}$, y_n is the nth observation, and ξ_n is a Pólya-Gamma random variable, $\xi_n \sim \mathrm{PG}(b,0)$ where $b>0$. For example, $\psi_n = \boldsymbol{m}^\top \mathbf{x}_n$ for models without latent variables and $\mathbf{x}_n$ is the nth input vector, or $\psi_n = \boldsymbol{m}^\top \boldsymbol{l}_n$ for models with latent variables. Note that b is set depending on which binomial model (depending on whether the likelihood is defined by logistic regression, binomial regression, negative binomial, etc.) one uses.

Following the above Theorem 1 (of Polson et al., 2013), for logistic regression, where we set $\psi_n = \boldsymbol{l}_n{}^\top \boldsymbol{m}$ and $b=1$, we can express the likelihood as:

$$p(\mathcal{D}_n|\boldsymbol{l}_n,\boldsymbol{m}) = 2^{-1}\exp((y_n-\tfrac{1}{2})\boldsymbol{l}_n{}^\top\boldsymbol{m})\int_0^\infty \exp(-\tfrac{1}{2}\xi_n \boldsymbol{l}_n{}^\top \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{l}_n) p(\xi_n) d\xi_n. \tag{35}$$

By introducing a variational posterior over the auxiliary variables, we introduce a new objective function (See Appendix Sec. 3 for derivation)

$$\begin{aligned}
&\mathcal{L}_n(q(\boldsymbol{m}),q(\boldsymbol{l}_n),q(\xi_n)) \\
&= \int\; q(\boldsymbol{m})q(\boldsymbol{l}_n)q(\xi_n)\log\frac{p(\mathcal{D}_n|\boldsymbol{l}_n,\xi_n,\boldsymbol{m})p(\boldsymbol{l}_n)p(\xi_n)p(\boldsymbol{m})}{q(\boldsymbol{m})q(\boldsymbol{l}_n)q(\xi_n)}, \\
&= -\log 2 + (y_n-\tfrac{1}{2})\langle \boldsymbol{l}_n\rangle_{q(\boldsymbol{l}_n)}{}^\top \langle \boldsymbol{m}\rangle_{q(\boldsymbol{m})} - \tfrac{1}{2}\langle\xi_n\rangle_{q(\xi_n)}\langle \boldsymbol{l}_n{}^\top \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{l}_n\rangle_{q(\boldsymbol{l}_n)q(\boldsymbol{m})}, \\
&\quad - \mathrm{D}_{KL}(q(\xi_n)||p(\xi_n)) - \mathrm{D}_{KL}(q(\boldsymbol{m})||p(\boldsymbol{m})) - \mathrm{D}_{KL}(q(\boldsymbol{l}_n)||p(\boldsymbol{l}_n)).
\end{aligned} \tag{36}$$

The first derivative of the lower bound with respect to $q(\xi_n)$ gives us a closed form update rule

$$\frac{\partial}{\partial q(\xi_n)}\mathcal{L}_n(q(\boldsymbol{m}),q(\boldsymbol{l}_n),q(\xi_n)) = 0, \;\; \mapsto q(\xi_n) \propto \mathrm{PG}(1, \sqrt{\langle \boldsymbol{l}_n{}^\top \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{l}_n\rangle_{q(\boldsymbol{l}_n)q(\boldsymbol{m})}}). \tag{37}$$

The rest of the updates for $q(\boldsymbol{l}_n)q(\boldsymbol{m})$ follow the standard updates under the conjugate exponential family distributions, since the lower bound to the conditional likelihood term includes only linear and quadratic terms both in $\boldsymbol{l}_n$ and $\boldsymbol{m}$. By introducing the auxiliary variables, the complete-data likelihood conditioned on ξ_n (with some prior on $\boldsymbol{l}_n$) now has the form of

$$\begin{aligned}
p(\mathcal{D}_n,\boldsymbol{l}_n|\boldsymbol{m},\xi_n) &\propto p(\mathcal{D}_n|\boldsymbol{l}_n,\boldsymbol{m},\xi_n)\; p(\boldsymbol{l}_n), \\
&\propto \exp(\mathbf{n}(\boldsymbol{m})^\top \mathbf{s}(\mathcal{D}_n,\boldsymbol{l}_n,\xi_n))\; p(\boldsymbol{l}_n),
\end{aligned} \tag{38}$$

which consists of natural parameters and sufficient statistics given by

$$\mathbf{n}(\boldsymbol{m}) = \begin{bmatrix} \boldsymbol{m} \\ \text{vec}(-\frac{1}{2}\boldsymbol{m}\boldsymbol{m}^\top) \end{bmatrix}, \quad \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n, \xi_n) = \begin{bmatrix} (y_n - \frac{1}{2})\boldsymbol{l}_n \\ \text{vec}(\xi_n \boldsymbol{l}_n \boldsymbol{l}_n^\top) \end{bmatrix}. \tag{39}$$

The above is obtained by fixing ξ_n in Eq. 35, noting that $\boldsymbol{l}_n{}^\top \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{l}_n = \sum_{ij} \boldsymbol{l}_{ni}(\boldsymbol{m}\boldsymbol{m}^\top)_{ij}\boldsymbol{l}_{nj} = \sum_{ij}(\boldsymbol{m}\boldsymbol{m}^\top)_{ij}(\boldsymbol{l}_n\boldsymbol{l}_n{}^\top)_{ij}$, collecting terms and plugging into Eq. 14. Note that now not only the latent and observed variables but also the new variables ξ_n form the complete-data sufficient statistics. The resulting variational Bayes algorithm for models with binomial likelihoods is given by

(a) Given the expected natural parameters $\bar{\mathbf{n}}$, the E-step yields:

$$q(\boldsymbol{l}, \boldsymbol{\xi}) = \prod_{n=1}^{N} q(\boldsymbol{l}_n) q(\xi_n) \propto p(\boldsymbol{\xi}) \exp\left[\int d\boldsymbol{m}\; q(\boldsymbol{m}) \log p(\mathcal{D}, \boldsymbol{l} | \boldsymbol{m}, \boldsymbol{\xi})\right],$$

$$\propto p(\boldsymbol{\xi}) p(\boldsymbol{l}) \prod_{n=1}^{N} \exp(\bar{\mathbf{n}}^\top \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n, \xi_n)),$$

$$\text{where } q(\xi_n) = \text{PG}(1, \sqrt{\langle \boldsymbol{l}_n{}^\top \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{l}_n \rangle_{q(\boldsymbol{l}_n, \boldsymbol{m})}}), \text{ and } p(\xi_n) = \text{PG}(1, 0) \tag{40}$$

$$q(\boldsymbol{l}_n) \propto p(\boldsymbol{l}_n) \exp(\bar{\mathbf{n}}^\top \langle \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n, \xi_n) \rangle_{q(\xi_n)}). \tag{41}$$

Using $q(\boldsymbol{l})q(\xi)$, it outputs $\bar{\mathbf{s}}(\mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N} \langle \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n, \xi_n) \rangle_{q(\boldsymbol{l}_n) q(\xi_n)}$.

(b) Given the expected sufficient statistics $\bar{\mathbf{s}}$, the M-step yields:

$$q(\boldsymbol{m}) \propto p(\boldsymbol{m}) \exp\left[\int d\boldsymbol{l}\; d\boldsymbol{\xi}\; q(\boldsymbol{l}) q(\boldsymbol{\xi})\; \log p(\mathcal{D}, \boldsymbol{l} | \boldsymbol{m}, \boldsymbol{\xi})\right],$$

$$\propto \exp(\mathbf{n}(\boldsymbol{m})^\top \tilde{\boldsymbol{\nu}}), \text{ where } \tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} + N\bar{\mathbf{s}}(\mathcal{D}). \tag{42}$$

Using $q(\boldsymbol{m})$, it outputs the expected natural parameters $\bar{\mathbf{n}} := \langle \mathbf{n}(\boldsymbol{m}) \rangle_{q(\boldsymbol{m})}$.

Similar to the VIPS algorithm for the CE family, perturbing the expected sufficient statistics $\bar{\mathbf{s}}(\mathcal{D})$ in the E-step suffices for privatising all the outputs of the algorithm. Algorithm 4 summarizes private stochastic variational Bayes algorithm for non-CE family with binomial likelihoods.

As a side note, variational inference algorithms typically use the variational lower bound as a stopping criterion. However, in the case of using Pólya-Gamma latent variables, one needs to draw samples from the Pólya-Gamma posterior to numerically calculate the lower bound given in Eq. 36. This might be a problem if one does not have access to the Pólya-Gamma posterior, since our algorithm only outputs the perturbed expected sufficient statistics in the E-step (not the Pólya-Gamma posterior itself). However, one could use other stopping criteria, which do not require sampling from the Pólya-Gamma posterior, e.g., calculating the prediction accuracy in the classification case.

6. VIPS for Bayesian Logistic Regression

We present an example of a non-CE family model, Bayesian logistic regression (BLR), and illustrate how to employ the VIPS framework given in Algorithm 4 in such a case.

Algorithm 4 $(\epsilon_{tot}, \delta_{tot})$-DP VIPS for non-CE family with binomial likelihoods

Input: Data $\mathcal{D}$. Define $\rho_t = (\tau_0 + t)^{-\kappa}$, mini-batch size S, maximum iterations J, and σ.
Output: Privacy-preserving expected natural parameters $\tilde{\bar{\mathbf{n}}}$ and expected sufficient stats $\tilde{\bar{\mathbf{s}}}$.
Compute the L2-sensitivity Δ of the expected sufficient statistics.
for $t = 1, \ldots, J$ **do**
Draw a minibatch of S datapoints, without replacement.
(1) E-step: Given expected natural parameters $\bar{\mathbf{n}}$, compute $q(\boldsymbol{l}_n)q(\xi_n)$ for $n = 1, \ldots, S$. Perturb each coordinate of $\bar{\mathbf{s}} = \frac{1}{S}\sum_{n=1}^{S}\langle \mathbf{s}(\mathcal{D}_n, \boldsymbol{l}_n)\rangle_{q(\boldsymbol{l}_n)q(\xi_n)}$ by adding noise drawn from $\mathcal{N}(0, \sigma^2\Delta^2)$. Output $\tilde{\bar{\mathbf{s}}}$.
(2) M-step: Given $\tilde{\bar{\mathbf{s}}}$, compute $q(\boldsymbol{m})$ by $\tilde{\boldsymbol{\nu}}^{(t)} = \boldsymbol{\nu} + N\tilde{\bar{\mathbf{s}}}$. Set $\tilde{\boldsymbol{\nu}}^{(t)} \hookleftarrow (1-\rho_t)\tilde{\boldsymbol{\nu}}^{(t-1)} + \rho_t\tilde{\boldsymbol{\nu}}^{(t)}$. Output the expected natural parameters $\tilde{\bar{\mathbf{n}}} = \langle \mathbf{n}(\boldsymbol{m})\rangle_{q(\boldsymbol{m})}$.
end for
Compute the privacy loss $(\epsilon_{tot}, \delta_{tot})$ using the analytical moments account method (Wang et al., 2019).

6.1 Model Specifics for Bayesian Logistic Regression

Under the logistic regression model with the Gaussian prior on the weights $\boldsymbol{m} \in \mathbb{R}^d$,

$$p(y_n = 1|\mathbf{x}_n, \boldsymbol{m}) = \sigma(\boldsymbol{m}^\top \mathbf{x}_n), \quad p(\boldsymbol{m}|\alpha) = \mathcal{N}(\boldsymbol{m}|0, \alpha^{-1}I), \quad p(\alpha) = \mathrm{Gam}(a_0, b_0), \tag{43}$$

where $\sigma(\boldsymbol{m}^\top \mathbf{x}_n) = 1/(1 + \exp(-\boldsymbol{m}^\top \mathbf{x}_n))$, the nth input is $\mathbf{x}_n \in \mathbb{R}^d$, and the nth output is $y_n \in \{0, 1\}$. In logistic regression, there are no latent variables. Hence, the complete-data likelihood coincides with the data likelihood,

$$p(y_n|\mathbf{x}_n, \boldsymbol{m}) \quad = \quad \frac{\exp(\psi_n)^{y_n}}{1 + \exp(\psi_n)}, \tag{44}$$

where $\psi_n = {\mathbf{x}_n}^\top \boldsymbol{m}$. Since the likelihood is not in the CE family, we use the Pólya-Gamma data augmentation trick to re-write it as

$$p(y_n|\mathbf{x}_n, \xi_n, \boldsymbol{m}) \quad \propto \quad \exp((y_n - \tfrac{1}{2}){\mathbf{x}_n}^\top \boldsymbol{m}) \exp(-\tfrac{1}{2}\xi_n {\mathbf{x}_n}^\top \boldsymbol{m}\boldsymbol{m}^\top \mathbf{x}_n) \ . \tag{45}$$

As ${\mathbf{x}_n}^\top \boldsymbol{m}\boldsymbol{m}^\top \mathbf{x}_n = \sum_{ij} \mathbf{x}_{ni}(\boldsymbol{m}\boldsymbol{m}^\top)_{ij}\mathbf{x}_{nj} = \sum_{ij}(\boldsymbol{m}\boldsymbol{m}^\top)_{ij}(\mathbf{x}_n{\mathbf{x}_n}^\top)_{ij}$, by collecting terms and plugging into Eq. 14 we observe that the data likelihood conditioned on ξ_n is

$$p(y_n|\mathbf{x}_n, \boldsymbol{m}, \xi_n) \propto \exp(\mathbf{n}(\boldsymbol{m})^\top \mathbf{s}(\mathcal{D}_n, \xi_n)), \tag{46}$$

where the natural parameters and sufficient statistics are given by

$$\mathbf{n}(\boldsymbol{m}) = \begin{bmatrix} \boldsymbol{m} \\ \mathrm{vec}(-\frac{1}{2}\boldsymbol{m}\boldsymbol{m}^\top) \end{bmatrix}, \quad \mathbf{s}(\mathcal{D}_n, \xi_n) = \begin{bmatrix} (y_n - \frac{1}{2})\mathbf{x}_n \\ \mathrm{vec}(\xi_n \mathbf{x}_n {\mathbf{x}_n}^\top) \end{bmatrix}. \tag{47}$$

Variational Bayes for Bayesian Logistic Regression Using the likelihood given in Eq. 38, we compute the posterior distribution over $\boldsymbol{m}, \alpha, \boldsymbol{\xi}$ by maximising the following

Algorithm 5 VIPS for Bayesian logistic regression

Input: Data $\mathcal{D}$. Define $\rho_t = (\tau_0 + t)^{-\kappa}$, mini-batch size S, and maximum iterations J
Output: Privatised expected natural parameters $\tilde{\bar{\mathbf{n}}}$ and expected sufficient statistics $\tilde{\bar{\mathbf{s}}}$
Using the sensitivity of the expected sufficient statistics given in Eq. 57 and Eq. 59,
for $t = 1, \ldots, J$ **do**
Draw a minibatch of S datapoints, without replacement.
(1) E-step: Given expected natural parameters $\bar{\mathbf{n}}$, compute $q(\xi_n)$ for $n = 1, \ldots, S$.
Perturb $\bar{\mathbf{s}} = \frac{1}{S}\sum_{n=1}^{S} \langle \mathbf{s}(\mathcal{D}_n, \xi_n) \rangle_{q(\xi_n)}$ by Eq. 56 and Eq. 58, and output $\tilde{\bar{\mathbf{s}}}$.
(2) M-step: Given $\tilde{\bar{\mathbf{s}}}$, compute $q(\boldsymbol{m})$ by $\tilde{\boldsymbol{\nu}}^{(t)} = \boldsymbol{\nu} + N\tilde{\bar{\mathbf{s}}}$. Set $\tilde{\boldsymbol{\nu}}^{(t)} \hookleftarrow (1-\rho_t)\tilde{\boldsymbol{\nu}}^{(t-1)} + \rho_t \tilde{\boldsymbol{\nu}}^{(t)}$.
Using $\tilde{\boldsymbol{\nu}}^{(t)}$, update $q(\alpha)$ by Eq. 53, and output $\tilde{\bar{\mathbf{n}}} = \langle \mathbf{n}(\boldsymbol{m}) \rangle_{q(\boldsymbol{m})}$.
end for
Compute the privacy loss $(\epsilon_{tot}, \delta_{tot})$ using the analytical moments account method (Wang et al., 2019).

variational lower bound due to Eq. 36:

$$\mathcal{L}(q(\boldsymbol{m}), q(\boldsymbol{\xi}), q(\alpha)) = \sum_{n=1}^{N} \left[-\log 2 + (y_n - \tfrac{1}{2}) \mathbf{x}_n{}^\top \langle \boldsymbol{m} \rangle_{q(\boldsymbol{m})} - \tfrac{1}{2} \langle \xi_n \rangle_{q(\xi_n)} \mathbf{x}_n{}^\top \langle \boldsymbol{m}\boldsymbol{m}^\top \rangle_{q(\boldsymbol{m})} \mathbf{x}_n \right], \\ - \sum_{n=1}^{N} \mathrm{D}_{KL}(q(\xi_n) || p(\xi_n)) - \mathrm{D}_{KL}(q(\boldsymbol{m}) || p(\boldsymbol{m})) - \mathrm{D}_{KL}(q(\alpha) || p(\alpha)).$$

In the E-step, we update

$$q(\boldsymbol{\xi}) \propto p(\boldsymbol{\xi}) \prod_{n=1}^{N} \exp(\bar{\mathbf{n}}(\boldsymbol{m})^\top \mathbf{s}(\mathcal{D}_n, \xi_n)), \tag{48}$$

$$= \prod_{n=1}^{N} q(\xi_n), \text{ where } q(\xi_n) = \mathrm{PG}(1, \sqrt{\mathbf{x}_n{}^\top \langle \boldsymbol{m}\boldsymbol{m}^\top \rangle_{q(\boldsymbol{m})} \mathbf{x}_n}). \tag{49}$$

Using $q(\boldsymbol{\xi})$, we compute the expected sufficient statistics

$$\bar{\mathbf{s}} = \tfrac{1}{N} \sum_{n=1}^{N} \bar{\mathbf{s}}(\mathcal{D}_n), \text{ where } \bar{\mathbf{s}}(\mathcal{D}_n) = \begin{bmatrix} \bar{\mathbf{s}}_1(\mathcal{D}_n) \\ \bar{\mathbf{s}}_2(\mathcal{D}_n) \end{bmatrix} = \begin{bmatrix} \frac{1}{N}(y_n - \frac{1}{2})\mathbf{x}_n \\ \frac{1}{N} \mathrm{vec}(\langle \xi_n \rangle_{q(\xi_n)} \mathbf{x}_n \mathbf{x}_n{}^\top) \end{bmatrix}. \tag{50}$$

In the M-step, we compute $q(\boldsymbol{m})$ and $q(\alpha)$ by

$$q(\boldsymbol{m}) \propto p(\mathcal{D}|\boldsymbol{m}, \langle \boldsymbol{\xi} \rangle_{q(\boldsymbol{\xi})})\, p(\boldsymbol{m}|\langle \alpha \rangle_{q(\alpha)}) \propto \exp(\mathbf{n}(\boldsymbol{m})^\top \tilde{\boldsymbol{\nu}}) = \mathcal{N}(\boldsymbol{m}|\boldsymbol{\mu}_{\boldsymbol{m}}, \Sigma_{\boldsymbol{m}}), \tag{51}$$

$$q(\alpha) \propto p(\boldsymbol{m}|\alpha) p(\alpha|a_0, b_0) = \mathrm{Gamma}(a_N, b_N), \tag{52}$$

where $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} + N\bar{\mathbf{s}}$, $\boldsymbol{\nu} = [\mathbf{0}_d, \langle \alpha \rangle_{q(\alpha)} I_d]$, and

$$a_N = a_0 + \tfrac{d}{2}, \quad b_N = b_0 + \tfrac{1}{2}(\boldsymbol{\mu}_{\boldsymbol{m}}{}^\top \boldsymbol{\mu}_{\boldsymbol{m}} + \mathrm{tr}(\Sigma_{\boldsymbol{m}})). \tag{53}$$

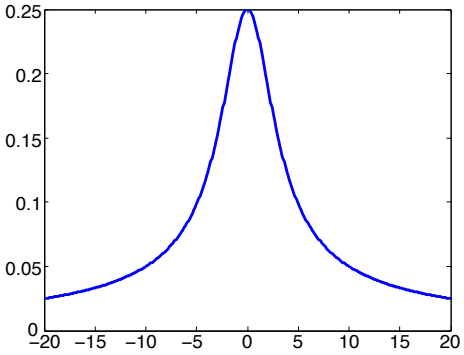

Figure 4: Posterior mean of each PG variable $\langle \xi_i \rangle$ as a function of $\sqrt{\mathbf{x}_i{}^\top \langle \boldsymbol{m}\boldsymbol{m}^\top \rangle \mathbf{x}_i}$. The maximum value of $\langle \xi_i \rangle$ is 0.25 when $\sqrt{\mathbf{x}_n{}^\top \langle \boldsymbol{m}\boldsymbol{m}^\top \rangle \mathbf{x}_n} = 0$,

Mapping from $\tilde{\boldsymbol{\nu}}$ to $(\boldsymbol{\mu}_{\boldsymbol{m}}, \Sigma_{\boldsymbol{m}})$ is deterministic, as below, where $\bar{\mathbf{s}}_1 = \sum_{n=1}^N \bar{\mathbf{s}}_1(\mathcal{D}_n)$ and $\bar{\mathbf{s}}_2 = \sum_{n=1}^N \bar{\mathbf{s}}_2(\mathcal{D}_n)$,

$$\tilde{\boldsymbol{\nu}} = \begin{bmatrix} N\bar{\mathbf{s}}_1 + \mathbf{0}_d \\ N\bar{\mathbf{s}}_2 + \langle \alpha \rangle_{q(\alpha)} I_d \end{bmatrix} = \begin{bmatrix} \Sigma_{\boldsymbol{m}}^{-1} \boldsymbol{\mu}_{\boldsymbol{m}} \\ \Sigma_{\boldsymbol{m}}^{-1} \end{bmatrix}. \tag{54}$$

Using $q(\boldsymbol{m})$, we compute expected natural parameters

$$\langle \mathbf{n}(\boldsymbol{m}) \rangle_{q(\boldsymbol{m})} = \begin{bmatrix} \langle \boldsymbol{m} \rangle_{q(\boldsymbol{m})} \\ \mathrm{vec}(-\frac{1}{2} \langle \boldsymbol{m}\boldsymbol{m}^\top \rangle_{q(\boldsymbol{m})}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{m}} \\ \mathrm{vec}(-\frac{1}{2}(\Sigma_{\boldsymbol{m}} + \boldsymbol{\mu}_{\boldsymbol{m}} \boldsymbol{\mu}_{\boldsymbol{m}}{}^\top)) \end{bmatrix}. \tag{55}$$

6.2 VIPS for BLR

Following the general framework of VIPS, we perturb the expected sufficient statistics. For perturbing $\bar{\mathbf{s}}_1$, we add Gaussian noise to each coordinate,

$$\tilde{\bar{\mathbf{s}}}_1 = \bar{\mathbf{s}}_1 + Y_{1,\ldots,d}, \text{ where } Y_i \sim^{i.i.d.} \mathcal{N}\left(0, \sigma^2 \Delta \bar{\mathbf{s}}_1^2\right) \tag{56}$$

where the sensitivity $\Delta \bar{\mathbf{s}}_1$ is given by

$$\Delta \bar{\mathbf{s}}_1 = \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}') \leq 1}} |\bar{\mathbf{s}}_1(\mathcal{D}) - \bar{\mathbf{s}}_1(\mathcal{D}')|_2 \leq \max_{\mathbf{x}_n, y_n} \tfrac{2}{N} |(y_n - \tfrac{1}{2})| \, |\mathbf{x}_n|_2 \leq \tfrac{2}{N}, \tag{57}$$

due to Eq. 22 and the assumption that the dataset is preprocessed such that any input has a maximum L2-norm of 1.

For perturbing $\bar{\mathbf{s}}_2$, we follow the *Analyze Gauss* (AG) algorithm (Dwork, Talwar, Thakurta & Zhang, 2014). We first draw Gaussian random variables $\mathbf{z} \sim \mathcal{N}\left(0, \sigma^2 (\Delta \bar{\mathbf{s}}_2)^2 I\right)$. Using $\mathbf{z}$, we construct a upper triangular matrix (including diagonal), then copy the upper part to the lower part so that the resulting matrix Z becomes symmetric. Then, we add this noisy matrix to the covariance matrix

$$\tilde{\bar{\mathbf{s}}}_2 = \bar{\mathbf{s}}_2 + Z. \tag{58}$$

The perturbed covariance might not be positive definite. In such case, we project the negative eigenvalues to some value near zero to maintain positive definiteness of the covariance matrix. The sensitivity of $\bar{\mathbf{s}}_2$ in Frobenius norm is given by

$$\Delta\bar{\mathbf{s}}_2 = \max_{\mathbf{x}_n, q(\xi_n)} \tfrac{2}{N} |\langle \xi_n \rangle_{q(\xi_n)} \mathrm{vec}(\mathbf{x}_n \mathbf{x}_n{}^\top)|_2 \leq \tfrac{1}{2N}, \tag{59}$$

due to Eq. 22 and the fact that the mean of a PG variable $\langle \xi_n \rangle$ is given by

$$\int_0^\infty \xi_n \, \mathrm{PG}(\xi_n |\, 1,\, \sqrt{\mathbf{x}_n{}^\top \langle \boldsymbol{m}\boldsymbol{m}^\top \rangle \mathbf{x}_n}) d\xi_n = \frac{1}{2\sqrt{\mathbf{x}_n{}^\top \langle \boldsymbol{m}\boldsymbol{m}^\top \rangle \mathbf{x}_n}} \tanh\left(\frac{\sqrt{\mathbf{x}_n{}^\top \langle \boldsymbol{m}\boldsymbol{m}^\top \rangle \mathbf{x}_n}}{2} \right).$$

As shown in Fig. 4, the maximum value is 0.25. Our VIPS algorithm for Bayesian logistic regression is given in Algorithm 5.

6.3 Experiments with Stroke Data

We used the Stroke dataset, which was first introduced by Letham, Rudin, McCormick & Madigan (2014) for predicting the occurrence of a stroke within a year after an atrial fibrillation diagnosis.[15] There are $N = 12,586$ patients in this dataset, and among these patients, 1,786 (14%) had a stroke within a year of the atrial fibrillation diagnosis.

Following Letham et al. (2014), we also considered all drugs and all medical conditions of these patients as candidate predictors. A binary predictor variable is used for indicating the presence or absence of each drug or condition in the longitudinal record prior to the atrial fibrillation diagnosis. In addition, a pair of binary variables is used for indicating age and gender. These totalled $d = 4,146$ unique features for medications and conditions. We randomly shuffled the data to make 5 pairs of training and test sets. For each set, we used $10,069$ patients' records as training data and the rest as test data.

Using this dataset, we ran our VIPS algorithm in batch mode, i.e., using the entire training data in each iteration, as opposed to using a small subset of data. We also ran the private and non-private Empirical Risk Minimisation (ERM) algorithms (Chaudhuri et al., 2011), in which we performed 5-fold cross-validation to set the regularisation constant given each training/test pair. As a performance measure, we calculated the *Area Under the Curve* (AUC) on each test data, given the posteriors over the latent and parameters in case of BLR and the parameter estimate in case of ERM. In Fig. 5, we show the mean and 1-standard deviation of the AUCs obtained by each method.

6.4 Experiments with Adult Data

We used the Adult data (from the UCI data repository), which consists of 48,842 data samples with 14 attributes and binary labels. We used 80% or original data for training and the rest for computing the AUC on test data. Under the logistic regression model, we compared the algorithm, DPVI[16] (Jälkö et al., 2017) to our VIPS method. In each

15. The authors extracted every patient in the MarketScan Medicaid Multi-State Database (MDCD) with a diagnosis of atrial fibrillation, one year of observation time prior to the diagnosis, and one year of observation time following the diagnosis.

16. Code is available at `https://github.com/DPBayes/DPVI-code/tree/master/uai17-code`

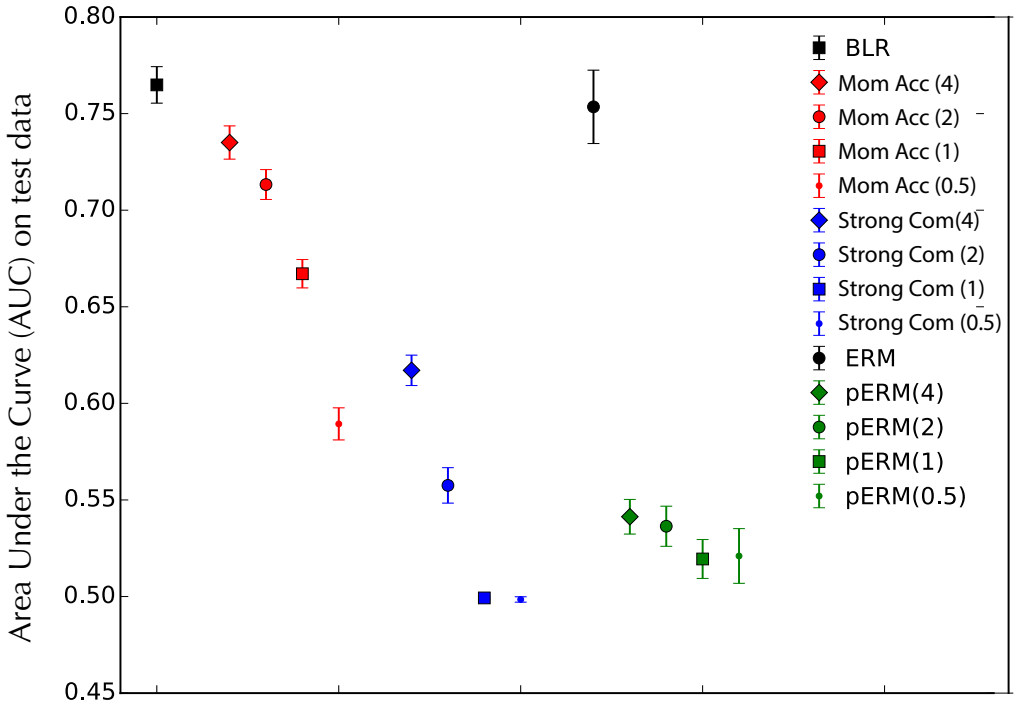

Figure 5: Stroke data. Comparison between our method and private/non-private ERM, for different $\epsilon_{tot} \in \{0.5, 1, 2, 4\}$. For the non-private methods, non-private BLR (black square marker) and ERM (black circle marker) achieved a similar AUC, which is higher than AUC obtained by any other private methods. Our method (red) under BLR with moments accountant with $\delta = 0.0001$ achieved the highest AUCs regardless of ϵ_{tot} among all the other private methods. The private version of ERM (objective perturbation, green) performed worse than BRL with strong composition as well as BRL with moments accountant. While directly comparing these to the private ERM is not totally fair since the private ERM (pERM) is ϵ-DP while others are (ϵ, δ)-DP, we show the difference between them in order to contrast the relative gain of our method compared to the existing method.

training step, we randomly selected data samples from the training with a sampling rate 0.004. We ran each of these algorithms for 100 training steps with 20 different random initializations with varying levels of noise. In Fig. 6, our VIPS algorithm outperforms DPVI regardless of the noise level we tested. The results seem to indicate that directly adding noise to the gradient with respect to the variational parameters in DPVI seems less effective than adding noise to the expected sufficient statistics computed using the Polya-Gamma random variables, in order to obtain the privacy-preserving natural parameters (variational parameters).

7. VIPS for Sigmoid Belief Networks

As the last example, we consider the Sigmoid Belief Network (SBN) model, which was first introduced by Neal (1992), for modeling N binary observations in terms of binary hidden

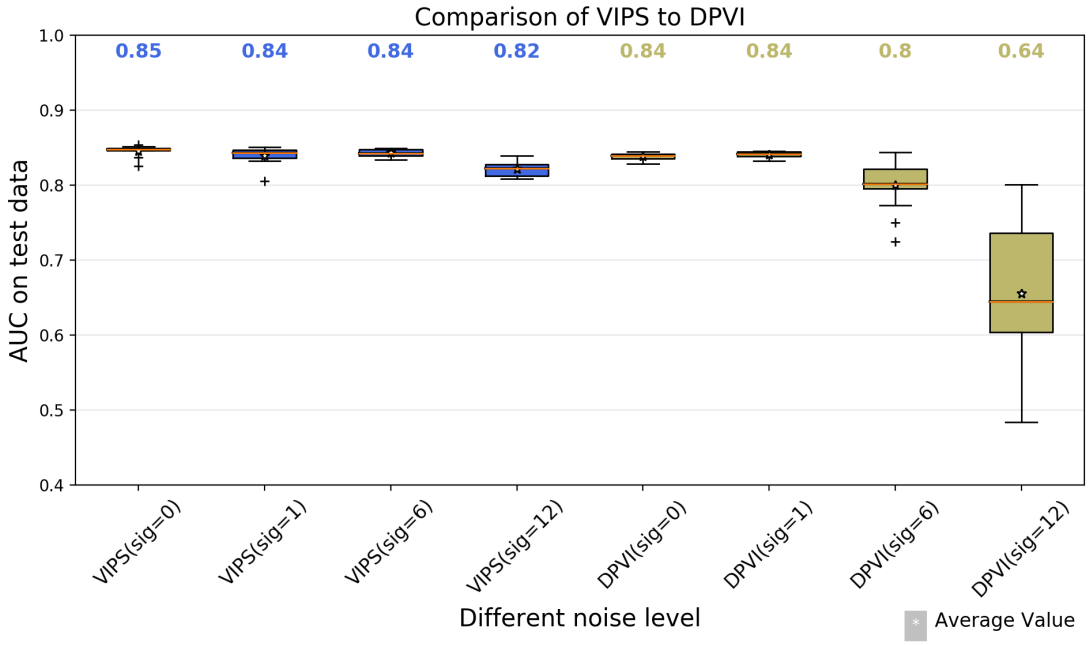

Figure 6: Adult data. Comparison between VIPS (left four blue boxes) and DPVI (right four yellow boxes) (Jälkö et al., 2017), with different levels of noise. The resulting ϵ values for a fixed value $\delta = 10^{-3}$ are $\epsilon_{tot} = \{0.8, 0.05, 0.025\}$ for $\sigma = \{1, 6, 12\}$, respectively. For $\sigma = 0$, the algorithm is equivalent to the non-private training under the logistic regression model. The numbers in bold fonts above each box are the average across 20 different runs with different initializations under the same noise setting. Our method seems to be more robust against the added noise for privacy in the high noise regime (i.e., when $\sigma = 12$).

Types	individual notations
Natural parameters	collectively as $\mathbf{b}$, $\mathbf{c}$, $\mathbf{W}$
Latent variables	collectively as $\mathbf{z}$
Pólya-Gamma auxiliary variables	$\boldsymbol{\xi}_n^{(0)}$ defined per datapoint, and $\boldsymbol{\xi}^{(1)}$
TPBN shrinkage priors for each element in $\mathbf{W}$	$\zeta_{j,k}, \xi_{j,k}, \phi_k, \omega$

Table 4: Summary of notation and their types for VIPS for sigmoid belief networks. Note that all of these quantities and corresponding distributions are learned during the training with the VIPS algorithm given in Algorithm 6.

variables and some parameters shown as Fig. 7. The notation specific to this section is summarised in Table 4. The complete-data likelihood for the nth observation and latent

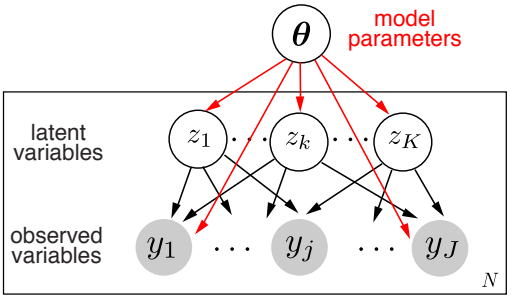

Figure 7: Schematic of the sigmoid belief network. A vector of binary observation $\mathbf{y} = [y_1, \cdots, y_J]$ is a function of a vector of binary latent variables $\mathbf{z} = [z_1, \cdots, z_K]$ and model parameters such that $p(y_j = 1|\mathbf{z}, \boldsymbol{\theta}) = \sigma(\mathbf{w}_j{}^\top \mathbf{z} + c_j)$. The latent variables are also binary such that $p(z_k = 1|\boldsymbol{\theta}) = \sigma(b_k)$, where $\boldsymbol{\theta} = \{\mathbf{w}, \mathbf{c}, \mathbf{b}\}$.

variables is given by

$$\begin{aligned} p(\mathcal{D}_n, \boldsymbol{l}_n|\boldsymbol{m}) &= p(\mathcal{D}_n|\boldsymbol{l}_n, \boldsymbol{m})p(\boldsymbol{l}_n|\boldsymbol{m}), \\ &= \prod_{j=1}^{J} \sigma(\mathbf{w}_j{}^\top \mathbf{z}_n + c_j) \prod_{k=1}^{K} \sigma(b_k), \qquad (60) \\ &= \prod_{j=1}^{J} \frac{[\exp(\mathbf{w}_j{}^\top \mathbf{z}_n + c_j)]^{y_{n,j}}}{1 + \exp(\mathbf{w}_j{}^\top \mathbf{z}_n + c_j)} \prod_{k=1}^{K} \frac{[\exp(b_k)]^{z_{n,k}}}{1 + \exp(b_k)}, \qquad (61) \end{aligned}$$

where we view the latent variables $\boldsymbol{l}_n = \mathbf{z}_n$ and model parameters $\boldsymbol{m} = \boldsymbol{\theta}$, and each datapoint $\mathcal{D}_n = \mathbf{y}_n$. Thanks to the Pólya-Gamma data augmentation strategy, we can rewrite the complete-data likelihood from

$$p(\mathbf{z}_n, \mathbf{y}_n|\boldsymbol{\theta}) = \prod_{j=1}^{J} \frac{[\exp(\psi_{n,j})]^{y_{n,j}}}{1 + \exp(\psi_{n,j})} \prod_{k=1}^{K} \frac{[\exp(b_k)]^{z_{n,k}}}{1 + \exp(b_k)}, \qquad (62)$$

where we denote $\psi_{n,j} = \mathbf{w}_j{}^\top \mathbf{z}_n + c_j$, to

$$p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\xi}_n^{(0)}, \boldsymbol{\xi}^{(1)}, \boldsymbol{\theta}) \propto \prod_{j=1}^{J} \exp((y_{n,j} - \tfrac{1}{2})\psi_{n,j}) \exp(-\tfrac{\xi_{n,j}^{(0)} \psi_{n,j}^2}{2}) \prod_{k=1}^{K} \exp((z_{n,k} - \tfrac{1}{2})b_k) \exp(-\tfrac{\xi_k^{(1)} b_k^2}{2})$$

where each element of vectors $\boldsymbol{\xi}_i^{(0)} \in \mathbb{R}^J$ and $\boldsymbol{\xi}^{(1)} \in \mathbb{R}^K$ is from PG$(1, 0)$.

Using the notations $\boldsymbol{\psi}_n \in \mathbb{R}^J$ where $\boldsymbol{\psi}_n = \mathbf{W}\mathbf{z}_n + \mathbf{c}$, $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_J]^\top \in \mathbb{R}^{J \times K}$, and $\mathbf{1}_J$ is a vector of J ones, we obtain

$$\begin{aligned} &p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\xi}_n^{(0)}, \boldsymbol{\xi}^{(1)}, \boldsymbol{\theta}) \\ &\propto \exp\left[(\mathbf{y}_n - \tfrac{1}{2}\mathbf{1}_J)^\top \boldsymbol{\psi}_n - \tfrac{1}{2}\boldsymbol{\psi}_n{}^\top \mathrm{diag}(\boldsymbol{\xi}_n^{(0)})\boldsymbol{\psi}_n + (\mathbf{z}_n - \tfrac{1}{2}\mathbf{1}_K)^\top \mathbf{b} - \tfrac{1}{2}\mathbf{b}^\top \mathrm{diag}(\boldsymbol{\xi}^{(1)})\mathbf{b}\right]. \qquad (63) \end{aligned}$$

The complete-data likelihood given the PG variables provides the exponential family form

$$\begin{aligned} &p(\mathbf{y}_n, \mathbf{z}_n|\boldsymbol{\xi}_n^{(0)}, \boldsymbol{\xi}^{(1)}, \boldsymbol{\theta}) \qquad (64) \\ &\propto \exp[(\mathbf{y}_n - \tfrac{1}{2}\mathbf{1}_J)^\top (W\mathbf{z}_n + \mathbf{c}) - \tfrac{1}{2}(W\mathbf{z}_n + \mathbf{c})^\top \mathrm{diag}(\boldsymbol{\xi}_n^{(0)})(W\mathbf{z}_n + \mathbf{c}) \\ &\qquad + (\mathbf{z}_n - \tfrac{1}{2}\mathbf{1}_K)^\top \mathbf{b} - \tfrac{1}{2}\mathbf{b}^\top \mathrm{diag}(\boldsymbol{\xi}^{(1)})\mathbf{b}], \qquad (65) \\ &\propto \exp[\mathbf{n}(\boldsymbol{\theta})^\top \mathbf{s}(\mathbf{y}_n, \mathbf{z}_n, \boldsymbol{\xi}_n^{(0)}, \boldsymbol{\xi}^{(1)})], \qquad (66) \end{aligned}$$

Algorithm 6 VIPS for sigmoid belief networks

Input: Data $\mathcal{D}$. Define $\rho_t = (\tau_0 + t)^{-\kappa}$, mini-batch size S, maximum iterations T, and σ
Output: Privatised expected natural parameters $\tilde{\bar{\mathbf{n}}}$ and expected sufficient statistics $\tilde{\bar{\mathbf{s}}}$
for $t = 1, \ldots, T$ **do**
Draw a minibatch of S datapoints, without replacement.
(1) E-step: Given expected natural parameters $\bar{\mathbf{n}}$, compute $q(\boldsymbol{\xi}_n^{(0)})$ for $n = 1, \ldots, S$.
Given $\bar{\mathbf{n}}$, compute $q(\boldsymbol{\xi}^{(1)})$ for $n = 1, \ldots, S$.
Given $\bar{\mathbf{n}}$, $q(\boldsymbol{\xi}_n^{(0)})$ and $q(\boldsymbol{\xi}^{(1)})$, compute $q(\mathbf{z}_n)$ for $n = 1, \ldots, S$.
Perturb $\bar{\mathbf{s}} = \frac{1}{S}\sum_{n=1}^{S} \langle \mathbf{s}(\mathcal{D}_n, \boldsymbol{\xi}_n^{(0)}, \boldsymbol{\xi}^{(1)}) \rangle_{q(\boldsymbol{\xi}_n^{(0)}) q(\boldsymbol{\xi}^{(1)}) q(\mathbf{z})}$ by Appendix Sec. 4, and output $\tilde{\bar{\mathbf{s}}}$.
(2) M-step: Given $\tilde{\bar{\mathbf{s}}}$, compute $q(\boldsymbol{m})$ by $\tilde{\boldsymbol{\nu}}^{(t)} = \boldsymbol{\nu} + N\tilde{\bar{\mathbf{s}}}$. Set $\tilde{\boldsymbol{\nu}}^{(t)} \hookleftarrow (1 - \rho_t)\tilde{\boldsymbol{\nu}}^{(t-1)} + \rho_t \tilde{\boldsymbol{\nu}}^{(t)}$. Using $\tilde{\boldsymbol{\nu}}^{(t)}$, update variational posteriors for hyper-priors by Appendix Sec. 5, and output $\tilde{\bar{\mathbf{n}}} = \langle \boldsymbol{m} \rangle_{q(\boldsymbol{m})}$.
end for
Compute the privacy loss $(\epsilon_{tot}, \delta_{tot})$ using the analytical moments account method (Wang et al., 2019).

where the natural parameters and sufficient statistics, found by collecting terms in the form of Eq. 14, are given by

$$\mathbf{n}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{b} \\ -\frac{1}{2}\mathrm{vec}(\mathbf{b}\mathbf{b}^\top) \\ \mathbf{c} \\ -\frac{1}{2}\mathrm{vec}(\mathbf{c}\mathbf{c}^\top) \\ -\frac{1}{2}\mathrm{vec}(\mathrm{diag}(\mathbf{c})\mathbf{W}) \\ \mathrm{vec}(\mathbf{W}) \\ -\frac{1}{2}\mathrm{vec}(\mathrm{vec}(\mathbf{W}^\top)\mathrm{vec}(\mathbf{W}^\top)^\top) \end{bmatrix}, \quad \mathbf{s}(\mathbf{y}_n, \mathbf{z}_n, \boldsymbol{\xi}_n^{(0)}, \boldsymbol{\xi}^{(1)}) = \begin{bmatrix} \mathbf{z}_n - \frac{1}{2}\mathbf{1}_K \\ \mathrm{vec}(\mathrm{diag}(\boldsymbol{\xi}^{(1)})) \\ \mathbf{y}_n - \frac{1}{2}\mathbf{1}_J \\ \mathrm{vec}(\mathrm{diag}(\boldsymbol{\xi}_n^{(0)})) \\ \mathrm{vec}(\boldsymbol{\xi}_n^{(0)}\mathbf{z}_n{}^\top) \\ \mathrm{vec}(\mathbf{z}_n(\mathbf{y}_n - \frac{1}{2}\mathbf{1}_J)^\top) \\ \mathrm{vec}(\mathrm{diag}(\boldsymbol{\xi}_n^{(0)}) \otimes (\mathbf{z}_n\mathbf{z}_n{}^\top)) \end{bmatrix}.$$

Now the PG variables form a set of sufficient statistics, which is separated from the model parameters. Similar to logistic regression, in the E-step, we compute the posterior over $\boldsymbol{\xi}$ and $\mathbf{z}$, and output perturbed expected sufficient statistics. The closed-form update of the posteriors over the PG variables is simply

$$q(\boldsymbol{\xi}_n^{(0)}) = \prod_{j=1}^{J} q(\boldsymbol{\xi}_{n,j}^{(0)}) = \prod_{j=1}^{J} \mathrm{PG}(1, \sqrt{\langle (\mathbf{w}_j{}^\top \mathbf{z}_n + c_j)^2 \rangle_{q(\boldsymbol{\theta}) q(\mathbf{z}_n)}}), \tag{67}$$

$$q(\boldsymbol{\xi}^{(1)}) = \prod_{k=1}^{K} q(\boldsymbol{\xi}_k^{(1)}) = \prod_{k=1}^{K} \mathrm{PG}(1, \sqrt{\langle b_k^2 \rangle_{q(\boldsymbol{\theta})}}). \tag{68}$$

The posterior over the latent variables is given by

$$q(\mathbf{z}) = \prod_{n=1}^{N}\prod_{k=1}^{K} q(z_{n,k}) = \text{Bern}(\sigma(d_{n,k})), \tag{69}$$

$$d_{n,k} = \langle b_k\rangle_{q(\boldsymbol{\theta})} + \langle \mathbf{w}_k{}^\top \mathbf{y}_n\rangle_{q(\boldsymbol{\theta})} - \tfrac{1}{2}\sum_{j=1}^{J}(\langle w_{j,k}\rangle_{q(\boldsymbol{\theta})} + \langle \xi^{(0)}_{n,j}\rangle_{q(\boldsymbol{\xi})}[2\langle \psi^{\setminus k}_{n,j} w_{j,k}\rangle_{q(\boldsymbol{\theta})q(\mathbf{z})} + \langle w^2_{j,k}\rangle_{q(\boldsymbol{\theta})}]), \tag{70}$$

$$\psi^{\setminus k}_{n,j} = \mathbf{w}_j{}^\top \mathbf{z}_n - w_{j,k} z_{n,k} + c_j. \tag{71}$$

Now, using $q(\mathbf{z})$ and $q(\boldsymbol{\xi})$, we compute the expected sufficient statistics,

$$\bar{\mathbf{s}}(\mathcal{D}) = \begin{bmatrix} \bar{\mathbf{s}}_1 = \frac{1}{N}\sum_{n=1}^{N}\mathbf{s}_1(\mathbf{y}_n) \\ \bar{\mathbf{s}}_2 = \frac{1}{N}\sum_{n=1}^{N}\mathbf{s}_2(\mathbf{y}_n) \\ \bar{\mathbf{s}}_3 = \frac{1}{N}\sum_{n=1}^{N}\mathbf{s}_3(\mathbf{y}_n) \\ \bar{\mathbf{s}}_4 = \frac{1}{N}\sum_{n=1}^{N}\mathbf{s}_4(\mathbf{y}_n) \\ \bar{\mathbf{s}}_5 = \frac{1}{N}\sum_{n=1}^{N}\mathbf{s}_5(\mathbf{y}_n) \\ \bar{\mathbf{s}}_6 = \frac{1}{N}\sum_{n=1}^{N}\mathbf{s}_6(\mathbf{y}_n) \\ \bar{\mathbf{s}}_7 = \frac{1}{N}\sum_{n=1}^{N}\mathbf{s}_7(\mathbf{y}_n) \end{bmatrix}, \quad \text{where} \quad \begin{bmatrix} \mathbf{s}_1(\mathbf{y}_n) \\ \mathbf{s}_2(\mathbf{y}_n) \\ \mathbf{s}_3(\mathbf{y}_n) \\ \mathbf{s}_4(\mathbf{y}_n) \\ \mathbf{s}_5(\mathbf{y}_n) \\ \mathbf{s}_6(\mathbf{y}_n) \\ \mathbf{s}_7(\mathbf{y}_n) \end{bmatrix} = \begin{bmatrix} \langle \mathbf{z}_n\rangle - \frac{1}{2}\mathbf{1}_K \\ \text{vec}(\text{diag}(\langle \boldsymbol{\xi}^{(1)}\rangle)) \\ \mathbf{y}_n - \frac{1}{2}\mathbf{1}_J \\ \text{vec}(\text{diag}(\langle \boldsymbol{\xi}^{(0)}_n\rangle)) \\ \text{vec}(\langle \boldsymbol{\xi}^{(0)}_n\rangle\langle \mathbf{z}_n\rangle^\top) \\ \text{vec}(\langle \mathbf{z}_n\rangle(\mathbf{y}_n - \frac{1}{2}\mathbf{1}_J)^\top) \\ \text{vec}(\text{diag}(\langle \boldsymbol{\xi}^{(0)}_n\rangle) \otimes (\langle \mathbf{z}_n\mathbf{z}_n{}^\top\rangle)) \end{bmatrix}.$$

Using the variational posterior distributions in the E-step, we perturb and output these sufficient statistics. See Appendix Sec. 4 for the sensitivities of each of these sufficient statistics. Note that when using the subsampled data per iteration, the sensitivity analysis has to be modified as now the query is evaluated on a smaller dataset. Hence, the $1/N$ factor has to be changed to $1/S$.

While any conjugate priors are acceptable for an analytic computation of the posterior update, following Gan et al. (2015), we put a Three Parameter Beta Normal (TPBN) prior for $\mathbf{W}$ where below we denote (j,k)th element of $\mathbf{W}$ by $W_{j,k}$,

$$W_{j,k} \sim \mathcal{N}(0, \zeta_{j,k}), \zeta_{j,k} \sim \text{Gam}(\tfrac{1}{2}, \xi_{j,k}), \xi_{j,k} \sim \text{Gam}(\tfrac{1}{2}, \phi_k), \phi_k \sim \text{Gam}(0.5, \omega), \omega \sim \text{Gam}(0.5, 1)$$

to induce sparsity, and isotropic normal priors for $\mathbf{b}$ and $\mathbf{c}$: $\mathbf{b} \sim \mathcal{N}(0, \nu_b I_K), \mathbf{c} \sim \mathcal{N}(0, \nu_c I_J)$, assuming these hyperparameters are set such that the prior is broad. The M-step updates for the variational posteriors for the parameters as well as for the hyper-parameters are given in Appendix Sec. 5. It is worth noting that these posterior updates for the hyper-parameters are one step away from the data, meaning that the posterior updates for the hyper-parameters are functions of variational posteriors that are already perturbed due to the perturbations in the expected natural parameters and expected sufficient statistics. Hence, we do not need any additional perturbation in the posteriors for the hyper-parameters.

Applying Moments Accountant's Composability to VIPS for SBNs Note that we cannot directly apply the moments accountant's composability theorem, because we compute multiple sufficient statistics given a mini-batch and the Gaussian noise added to the resulting sufficient statistics is *not independent*, while the composability theorem only

holds for independent noise addition given each mini-batch of data. To resolve this, we modify the sufficient statistic vector into a new vector with a modified sensitivity, such that we can apply the Gaussian mechanism only once given a freshly drawn mini-batch of data. This allows us to use the composability theorem, meaning that the log-moments of the privacy loss random variable are additive since the Gaussian noise added to the modified vector of sufficient statistics given each subsample of data is now independent.

Specifically, let us denote the sensitivities of each vector quantity by $C_1, \cdots, C_7$. Further, denote the moments accountant noise parameter by σ, and the subsampled data by $\mathcal{D}_\upsilon$ with sampling rate υ. We first scale down each sufficient statistic vector by its own sensitivity, i.e., $\mathbf{s}'_i = \bar{\mathbf{s}}_i / C_i$, so that the concatenated vector (denoted by $\mathbf{s}'$)'s sensitivity becomes simply $\sqrt{7}$. The main difference we make here is that instead of adding separate Gaussian draws per sufficient statistic vector based on their individual sensitivities, we add a single draw of Gaussian noise to the 7-dimensional concatenated sufficient statistic vector based on the sensitivity of the rescaled concatenated vector. The standard normal noise to the vectors is applied with scaled standard deviation, $\sqrt{7}\sigma$. We then scale up each chunk of the perturbed quantity by its own sensitivity to return it to its original scale, given as

$$\begin{bmatrix} \tilde{\mathbf{s}}_1(\mathcal{D}_\upsilon) \\ \tilde{\mathbf{s}}_2(\mathcal{D}_\upsilon) \\ \tilde{\mathbf{s}}_3(\mathcal{D}_\upsilon) \\ \tilde{\mathbf{s}}_4(\mathcal{D}_\upsilon) \\ \tilde{\mathbf{s}}_5(\mathcal{D}_\upsilon) \\ \tilde{\mathbf{s}}_6(\mathcal{D}_\upsilon) \\ \tilde{\mathbf{s}}_7(\mathcal{D}_\upsilon) \end{bmatrix} = \begin{bmatrix} C_1\tilde{\mathbf{s}}'_1 \\ C_2\tilde{\mathbf{s}}'_2 \\ C_3\tilde{\mathbf{s}}'_3 \\ C_4\tilde{\mathbf{s}}'_4 \\ C_5\tilde{\mathbf{s}}'_5 \\ C_6\tilde{\mathbf{s}}'_6 \\ C_7\tilde{\mathbf{s}}'_7 \end{bmatrix} \quad \text{where } \tilde{\mathbf{s}}'_{ij} = \mathbf{s}'_{ij} + \sqrt{7}\sigma\mathcal{N}(0, I), \tag{72}$$

where the ith chunk of the vector $\mathbf{s}$ is $\mathbf{s}'_i = \bar{\mathbf{s}}_i / C_i$, and where $\mathbf{s}'_{ij}$ indexes the jth entry of $\mathbf{s}'_i$, resulting in privatized sufficient statistics.

7.1 Experiments with MNIST Data

We tested our VIPS algorithm for the SBN model on the binarized[17] MNIST digit dataset which contains 60,000 training images of ten handwritten digits (0 to 9), where each image consists of 28×28 pixels. For our experiment, we considered a one-hidden layer SBN with 50 or 100 hidden units, i.e. $K = \{50, 100\}$.[18] We varied the mini-batch size $S = \{400, 800, 1600, 3200\}$. For a fixed $\sigma = 1$, we obtained two different values of privacy loss due to different mini-batch sizes. We ran our algorithm until it processed the entire training data ($60,000$ data points). We tested the non-private version of VIPS as well as the private versions with strong and moments accountant compositions, where each algorithm was tested in 10 independent runs with different seed numbers.

As a performance measure, we calculated the pixel-wise prediction accuracy by first converting the pixels of reconstructed images into probabilities (between 0 and 1); then

17. The reason why they binarised the dataset is that the model, SBN, outputs binarised variables. So to match the data to model, they convert the pixel values to binary values.
18. We chose these numbers since when K is larger than 100, the variational lower bound on the test data, shown in Figure 2 of Gan et al. (2015), does not increase significantly.

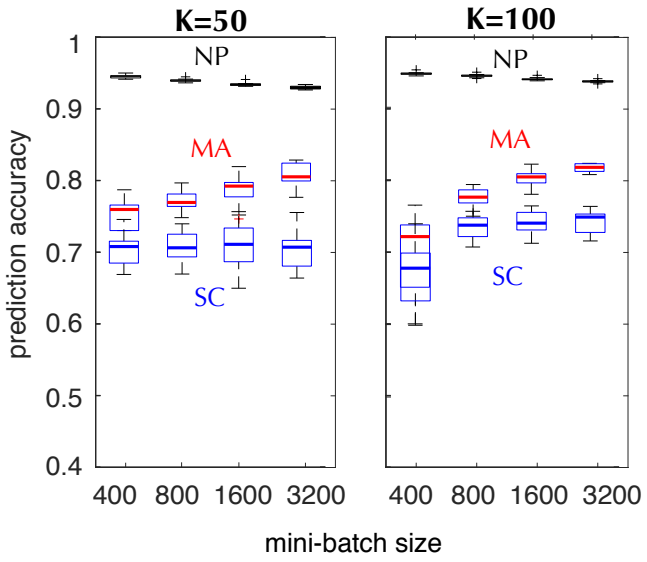

Figure 8: Prediction accuracy (binarised version of MNIST dataset). The non-private version (NP in black) achieves highest prediction accuracy under the SBN model, as expected. The private versions (moments accountant : MA (in red), and strong composition : SC (in blue)) with $\sigma = 1$ resulted in a total privacy loss of $\epsilon_{tot} = 1.34$ when the mini-batch size was $S = 400$, and $\epsilon_{tot} = 1.74$ for $S = 800$, $\epsilon_{tot} = 2.44$ for $S = 1600$, and $\epsilon_{tot} = 3.34$ for $S = 3200$. In all of these cases, we set $\delta_{total} = 10^{-4}$. We ran these algorithms until they processed the total training data, resulting in different numbers of iterations for each minibatch size. The private version using strong composition (blue) performed worse than when using moments accountant (red) regardless of the size of mini-batches, since the level of additive noise per iteration in strong composition is higher than that of the moments accountant.

converting the probabilities into binary variables; and averaging the squared distances[19] between the predicted binary variables and the test images. In each seed number, we selected 100 randomly selected test images from $10,000$ test datapoints. Fig. 8 shows the performance of each method in terms of prediction accuracy.

8. Discussion

We have developed a practical privacy-preserving VB algorithm which outputs accurate and privatized expected sufficient statistics and expected natural parameters. Our approach uses the moments accountant analysis combined with the privacy amplification effect due to subsampling of data, which significantly decrease the amount of additive noise for the same expected privacy guarantee compared to the standard analysis. Our methods show how to perform variational Bayesian inference in private settings, not only for the conjugate exponential family models but also for non-conjugate models with binomial likelihoods using

19. As the values in each pixel are binarised, the average squared distance between predictive image and test image is just average of a sum of 1's and 0's. This pixel-wise performance measure tells us that how many pixels are predicted correctly on average across the test points we tested.

the Polyá Gamma data augmentation. We illustrated the effectiveness of our algorithm on several real-world datasets.

The private VB algorithms for the Latent Dirichlet Allocation (LDA), Bayesian Logistic Regression (BLR), and Sigmoid Belief Network (SBN) models we discussed are just a few examples of a much broader class of models to which our private VB framework applies. Our positive empirical results with VB indicate that these ideas are also likely to be beneficial for privatizing many other iterative machine learning algorithms. In future work, we plan to apply this general framework to other inference methods for larger and more complicated models such as deep neural networks. More broadly, our vision is that *practical* privacy preserving machine learning algorithms will have a transformative impact on the practice of data science in many real-world applications.

Acknowledgments

M. Park and J. Foulds contributed equally. A significant amount of work of M. Park and M. Welling was supported by the QUVA lab at the University of Amsterdam. The later work of M. Park was supported by the Max Planck Society, as well as the Gibs Schüle Foundation and the Institutional Strategy of the University of Tübingen (ZUK63). K. Chaudhuri was partially supported by ONR under N00014-16-1-261, UC Lab Fees under LFR 18-548554 and a Google Faculty Fellowship. We also thank our anonymous reviewers for their useful comments.

Appendix

1. Proof of L2-Sensitivity of Expected Sufficient Statistics

$$
\begin{aligned}
\Delta M_l &= \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |M_l(\mathcal{D}) - M_l(\mathcal{D}')|, \\
&= \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |\tfrac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{l}_n)}\mathbf{s}_l(\mathcal{D}_n, \boldsymbol{l}_n) - \tfrac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{l}'_n)}\mathbf{s}_l(\mathcal{D}'_n, \boldsymbol{l}'_n)|, \\
&\leq \max_{\mathcal{D}_N, q(\boldsymbol{l}_N), \mathcal{D}'_N, q'(\boldsymbol{l}_N)} |\tfrac{1}{N}\mathbb{E}_{q(\boldsymbol{l}_N)}\mathbf{s}_l(\mathcal{D}_N, \boldsymbol{l}_N) - \tfrac{1}{N}\mathbb{E}_{q'(\boldsymbol{l}_N)}\mathbf{s}'_l(\mathcal{D}_N, \boldsymbol{l}_N)|, \\
&\leq \max_{\mathcal{D}_N, q(\boldsymbol{l}_N)} \tfrac{2}{N}|\mathbb{E}_{q(\boldsymbol{l}_N)}\mathbf{s}_l(\mathcal{D}_N, \boldsymbol{l}_N)|, \qquad (73)
\end{aligned}
$$

where the last line is due to the triangle inequality.

2. Proof of L2-Sensitivity for LDA

$$
\begin{aligned}
\Delta\bar{\mathbf{s}} &= \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} \sqrt{\sum_k \sum_v (\bar{\mathbf{s}}_k^v(\mathcal{D}) - \bar{\mathbf{s}}_k^v(\mathcal{D}'))^2}, \\
&= \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} \sqrt{\sum_k \sum_v \left(\frac{1}{S} \sum_n \sum_{d=1}^{S} \boldsymbol{\phi}_{dn}^k \mathbf{w}_{dn}^v - \frac{1}{S} \sum_n \sum_{d=1}^{S} \boldsymbol{\phi}'^k_{dn} \mathbf{w}'^v_{dn} \right)^2}, \\
&= \max_{\boldsymbol{\phi}_{Sn}^k, \mathbf{w}_{Sn}^v, \boldsymbol{\phi}'^k_{Sn} \mathbf{w}'^v_{Sn}} \sqrt{\sum_k \sum_v \left(\frac{1}{S} \sum_n \boldsymbol{\phi}_{Sn}^k \mathbf{w}_{Sn}^v - \frac{1}{S} \sum_n \boldsymbol{\phi}'^k_{Sn} \mathbf{w}'^v_{Sn} \right)^2}, \\
&\leq \max_{\boldsymbol{\phi}_{Sn}^k, \mathbf{w}_{Sn}^v} \sqrt{\sum_k \sum_v \left(\frac{1}{S} \sum_n \boldsymbol{\phi}_{Sn}^k \mathbf{w}_{Sn}^v \right)^2}, \text{ since } 0 \leq \boldsymbol{\phi}_{dn}^k \leq 1, \, \mathbf{w}_{dn}^v \in \{0,1\}, \\
&\qquad \text{and so the worst case per } k,v \text{ entry is where one of the terms is 0,} \\
&\leq \max_{\boldsymbol{\phi}_{Sn}^k, \mathbf{w}_{Sn}^v} \frac{1}{S} \sum_n (\sum_k \boldsymbol{\phi}_{Sn}^k)(\sum_v \mathbf{w}_{Sn}^v) \leq \frac{N}{S}, \qquad (74)
\end{aligned}
$$

since $\sum_k \boldsymbol{\phi}_{Sn}^k = 1$, and $\sum_v \mathbf{w}_{Sn}^v = 1$.

3. Variational Lower Bound with Auxiliary Variables

The variational lower bound (per-datapoint for simplicity) given by

$$
\begin{aligned}
\mathcal{L}_n(q(\boldsymbol{m}), q(\boldsymbol{l}_n)) &= \int q(\boldsymbol{m}) q(\boldsymbol{l}_n) \log \frac{p(\mathcal{D}_n|\boldsymbol{l}_n, \boldsymbol{m}) p(\boldsymbol{m}) p(\boldsymbol{l}_n)}{q(\boldsymbol{m}) q(\boldsymbol{l}_n)} d\boldsymbol{l}_n d\boldsymbol{m}, \qquad (75) \\
&= \int q(\boldsymbol{m}) q(\boldsymbol{l}_n) \log p(\mathcal{D}_n|\boldsymbol{l}_n, \boldsymbol{m}) d\boldsymbol{l}_n d\boldsymbol{m} \\
&\quad - \mathrm{D}_{KL}(q(\boldsymbol{m})||p(\boldsymbol{m})) - \mathrm{D}_{KL}(q(\boldsymbol{l}_n)||p(\boldsymbol{l}_n)),
\end{aligned}
$$

where the first term can be re-written using Eq. 35,

$$
\begin{aligned}
\int q(\boldsymbol{m}) q(\boldsymbol{l}_n) \log p(\mathcal{D}_n|\boldsymbol{l}_n, \boldsymbol{m}) d\boldsymbol{l}_n d\boldsymbol{m} &= -b \log 2 + (y_n - \tfrac{b}{2}) \langle \boldsymbol{l}_n \rangle_{q(\boldsymbol{l}_n)}^\top \langle \boldsymbol{m} \rangle_{q(\boldsymbol{m})} \\
&\quad + \int q(\boldsymbol{m}) q(\boldsymbol{l}_n) \log \int_0^\infty \exp(-\tfrac{1}{2} \xi_n \boldsymbol{l}_n^\top \boldsymbol{m} \boldsymbol{m}^\top \boldsymbol{l}_n) p(\xi_n) d\xi_n.
\end{aligned}
$$

We rewrite the third term using $q(\xi_n)$, the variational posterior distribution for ξ_n,

$$
\begin{aligned}
&\int q(\boldsymbol{m}) q(\boldsymbol{l}_n) \log \int_0^\infty \exp(-\tfrac{1}{2} \xi_n \boldsymbol{l}_n^\top \boldsymbol{m} \boldsymbol{m}^\top \boldsymbol{l}_n) p(\xi_n) d\xi_n \\
&= \int q(\boldsymbol{m}) q(\boldsymbol{l}_n) \log \int_0^\infty q(\xi_n) \exp(-\tfrac{1}{2} \xi_n \boldsymbol{l}_n^\top \boldsymbol{m} \boldsymbol{m}^\top \boldsymbol{l}_n) \frac{p(\xi_n)}{q(\xi_n)} d\xi_n, \\
&\geq \int q(\boldsymbol{m}) q(\boldsymbol{l}_n) \int_0^\infty q(\xi_n) \left[(-\tfrac{1}{2} \xi_n \boldsymbol{l}_n^\top \boldsymbol{m} \boldsymbol{m}^\top \boldsymbol{l}_n) + \log \frac{p(\xi_n)}{q(\xi_n)} \right] d\xi_n, \\
&= -\tfrac{1}{2} \langle \xi_n \rangle_{q(\xi_n)} \langle \boldsymbol{l}_n^\top \boldsymbol{m} \boldsymbol{m}^\top \boldsymbol{l}_n \rangle_{q(\boldsymbol{l}_n) q(\boldsymbol{m})} - \mathrm{D}_{KL}(q(\xi_n)||p(\xi_n)), \qquad (76)
\end{aligned}
$$

which gives us a lower bound on the log likelihood,

$$\begin{aligned}\mathcal{L}_n(q(\boldsymbol{m}), q(\boldsymbol{l}_n)) \geq & \mathcal{L}_n(q(\boldsymbol{m}), q(\boldsymbol{l}_n), q(\xi_n)) \\ := & -b\log 2 + (y_n - \tfrac{b}{2})\langle \boldsymbol{l}_n \rangle_{q(\boldsymbol{l}_n)}{}^\top \langle \boldsymbol{m} \rangle_{q(\boldsymbol{m})} \\ & - \tfrac{1}{2}\langle \xi_n \rangle_{q(\xi_n)} \langle \boldsymbol{l}_n{}^\top \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{l}_n \rangle_{q(\boldsymbol{l}_n)q(\boldsymbol{m})} \\ & - \mathrm{D}_{KL}(q(\xi_n)||p(\xi_n)) - \mathrm{D}_{KL}(q(\boldsymbol{m})||p(\boldsymbol{m})) - \mathrm{D}_{KL}(q(\boldsymbol{l}_n)||p(\boldsymbol{l}_n)),\end{aligned} \tag{77}$$

which implies that

$$\begin{aligned}&\int q(\boldsymbol{m})q(\boldsymbol{l}_n)q(\xi_n)\log p(\mathcal{D}_n|\boldsymbol{l}_n, \xi_n, \boldsymbol{m}) \\ &= -b\log 2 + (y_n - \tfrac{b}{2})\langle \boldsymbol{l}_n \rangle_{q(\boldsymbol{l}_n)}{}^\top \langle \boldsymbol{m} \rangle_{q(\boldsymbol{m})} - \tfrac{1}{2}\langle \xi_n \rangle_{q(\xi_n)} \langle \boldsymbol{l}_n{}^\top \boldsymbol{m}\boldsymbol{m}^\top \boldsymbol{l}_n \rangle_{q(\boldsymbol{l}_n)q(\boldsymbol{m})}.\end{aligned} \tag{78}$$

4. Perturbing Expected Sufficient Statistics in SBNs

Using the variational posterior distributions in the E-step, we perturb and output each sufficient statistic as follows. Note that the $1/N$ factor has to be changed to $1/S$ when using the subsampled data per iteration.

- For perturbing $\bar{\mathbf{s}}_1$, we perturb $A = \frac{1}{N}\sum_{n=1}^N \langle \mathbf{z}_n \rangle$ where the sensitivity is given by

$$\begin{aligned}\Delta A &= \max_{|\mathcal{D}-\mathcal{D}'|=1} |A(\mathcal{D}) - A(\mathcal{D}')|_2, \\ &\leq \max_{\mathbf{y}_n, q(\mathbf{z}_n)} \frac{1}{N}\sqrt{\sum_{k=1}^K (\sigma(d_{n,k}))^2} \leq \frac{\sqrt{K}}{N},\end{aligned} \tag{79}$$

  due to Eq. 22 and the fact that $\mathbf{z}_n$ is a vector of Bernoulli random variables (length K) and the mean of each element is $\sigma(d_{n,k})$ as given in Eq. 69.

- For perturbing $\tilde{\bar{\mathbf{s}}}_2$, we perturb $B = \langle \boldsymbol{\xi}^{(1)} \rangle$, i.e. the part before we apply the diag and vec operations. The sensitivity of B is given by $\Delta B \leq \frac{\sqrt{K}}{4}$ as shown in Fig. 4.

- For perturbing $\bar{\mathbf{s}}_3$, we perturb $C = \frac{1}{N}\sum_{n=1}^N \mathbf{y}_n$, where the sensitivity is given by

$$\Delta C = \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |C(\mathcal{D}) - C(\mathcal{D}')|_2 = \max_{\mathbf{y}_n} \tfrac{1}{N}|\mathbf{y}_n|_2 \leq \tfrac{\sqrt{J}}{N}, \tag{80}$$

  since $\mathbf{y}_n$ is a binary vector of length J. Note that when performing the batch optimisation, we perturb $\bar{\mathbf{s}}_3$ only once, since this quantity remains the same across iterations. However, when performing the stochastic optimisation, we perturb $\bar{\mathbf{s}}_3$ in every iteration, since the new mini-batch of data is selected in every iteration.

- For perturbing $\tilde{\bar{\mathbf{s}}}_4$, we perturb $D = \frac{1}{N}\sum_{n=1}^N \langle \boldsymbol{\xi}_n^{(0)} \rangle$, which is once again the part before taking diag and vec operations. Due to Eq. 22, the sensitivity is given by

$$\Delta D = \max_{\substack{\mathcal{D},\mathcal{D}' \\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |D(\mathcal{D}) - D(\mathcal{D}')|_2 \leq \frac{\sqrt{J}}{4N}. \tag{81}$$

- For $\tilde{\bar{\mathbf{s}}}_5$, we perturb $E = \frac{1}{N}\sum_{n=1}^{N}\langle \boldsymbol{\xi}_n^{(0)}\rangle\langle \mathbf{z}_n\rangle^\top$. From Eq. 22, the sensitivity is given by

$$\begin{aligned}\Delta E &= \max_{\substack{\mathcal{D},\mathcal{D}'\\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |E(\mathcal{D}) - E(\mathcal{D}')|_2, \\ &\leq \max_{\mathbf{y}_n, q(\mathbf{z}_n), q(\xi_n)} \frac{1}{N}\sqrt{\sum_{k=1}^{K}\sum_{j=1}^{J}(\langle \boldsymbol{\xi}_{n,j}^{(0)}\rangle\langle \mathbf{z}_{n,k}\rangle)^2} \leq \frac{\sqrt{JK}}{4N}. \end{aligned} \tag{82}$$

- For $\tilde{\bar{\mathbf{s}}}_6$, we can use the noisy $\tilde{\bar{\mathbf{s}}}_1$ for the second term, but perturb only the first term $F = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\langle \mathbf{z}_n\rangle^\top$. Due to Eq. 22, the sensitivity is given by

$$\begin{aligned}\Delta F &= \max_{\substack{\mathcal{D},\mathcal{D}'\\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |F(\mathcal{D}) - F(\mathcal{D}')|_2, \\ &\leq \max_{\mathbf{y}_n, q(\mathbf{z}_n)} \frac{1}{N}\sqrt{\sum_{k=1}^{K}\sum_{j=1}^{J}(\mathbf{y}_{n,j}\langle \mathbf{z}_{n,k}\rangle)^2} \leq \frac{\sqrt{JK}}{N}. \end{aligned} \tag{83}$$

- For $\bar{\mathbf{s}}_7$, we define a matrix G, which is a collection of J matrices where each matrix is $G_j = \frac{1}{N}\sum_{n=1}^{N}\langle \boldsymbol{\xi}_{n,j}^{(0)}\rangle\langle \mathbf{z}_n{\mathbf{z}_n}^\top\rangle$, where $G_j \in \mathbb{R}^{K\times K}$.

  Using Eq. 22, the sensitivity of G_j is given by

$$\begin{aligned}\Delta G_j &= \max_{\substack{\mathcal{D},\mathcal{D}'\\ \mathrm{d}(\mathcal{D},\mathcal{D}')\leq 1}} |G_j(\mathcal{D}) - G_j(\mathcal{D}')|_2, \\ &\leq \max_{\mathbf{y}_n} \tfrac{1}{N}\sqrt{\sum_{k=1}^{K}\sum_{k'=1}^{K}(\langle \boldsymbol{\xi}_{n,j}^{(0)}\rangle\langle \mathbf{z}_{n,k}\mathbf{z}_{n,k}\rangle)^2} \leq \tfrac{K}{4N}, \end{aligned} \tag{84}$$

  which gives us the sensitivity of $\Delta G \leq \frac{\sqrt{J}K}{4N}$.

5. M-step Updates for SBNs

The M-step updates are given below (taken from Gan et al., 2015):

- $q(W) = \prod_{j=1}^{J} q(\mathbf{w}_j)$, and $q(\mathbf{w}_j) = \mathcal{N}(\mu_j, \Sigma_j)$, where

$$\begin{aligned}\Sigma_j &= \left[\sum_{n=1}^{N}\langle \xi_{n,j}^{(0)}\rangle_{q(\boldsymbol{\xi})}\langle \mathbf{z}_n{\mathbf{z}_n}^\top\rangle + \mathrm{diag}(\langle \boldsymbol{\zeta}_j^{-1}\rangle_{q(\boldsymbol{\zeta})})\right]^{-1} \\ \boldsymbol{\mu}_j &= \Sigma_j\left[\sum_{n=1}^{N}(\mathbf{y}_{n,j} - \tfrac{1}{2} - \langle c_j\rangle_{q(\boldsymbol{\theta})}\langle \xi_{n,j}^{(0)}\rangle_{q(\boldsymbol{\xi})})\langle \mathbf{z}_n\rangle_{q(\mathbf{z})}\right].\end{aligned}$$

  where we replace $\frac{1}{N}\sum_{n=1}^{N}\langle \xi_{n,j}^{(0)}\rangle_{q(\boldsymbol{\xi})}\langle \mathbf{z}_n{\mathbf{z}_n}^\top\rangle$, $\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_{n,j}\langle \mathbf{z}_n\rangle_{q(\mathbf{z})}$, $\frac{1}{N}\sum_{n=1}^{N}\langle \mathbf{z}_n\rangle_{q(\mathbf{z})}$, and $\frac{1}{N}\sum_{n=1}^{N}\langle \xi_{n,j}^{(0)}\rangle_{q(\boldsymbol{\xi})}\langle \mathbf{z}_n\rangle_{q(\mathbf{z})}$, with perturbed expected sufficient statistics.

- $q(\mathbf{b}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{b}, \Sigma_\mathbf{b})$, where

$$\Sigma_\mathbf{b} = \left[\frac{1}{\nu_b} I + N\mathrm{diag}(\langle \boldsymbol{\xi}^{(1)} \rangle)\right]^{-1}, \tag{85}$$

$$\boldsymbol{\mu}_\mathbf{b} = \Sigma_\mathbf{b} \left[\sum_{n=1}^{N} (\langle \mathbf{z}_n \rangle - \frac{1}{2}\mathbf{1}_K)\right], \tag{86}$$

  where we replace $\langle \boldsymbol{\xi}^{(1)} \rangle$ and $\frac{1}{N}\sum_{n=1}^{N} \langle \mathbf{z}_n \rangle_{q(\mathbf{z})}$ with perturbed expected sufficient statistics.

- $q(\mathbf{c}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{c}, \Sigma_\mathbf{c})$, where

$$\Sigma_\mathbf{c} = \left[\frac{1}{\nu_c} I + \mathrm{diag}(\sum_{n=1}^{N} \langle \boldsymbol{\xi}_n^{(0)} \rangle)\right]^{-1}, \tag{87}$$

$$\boldsymbol{\mu}_\mathbf{c} = \Sigma_\mathbf{c} \left[\sum_{n=1}^{N} ({\mathbf{y}_n}^\top - \frac{1}{2}{\mathbf{1}_J}^\top - \frac{1}{2}\mathrm{diag}(\langle \boldsymbol{\xi}_n^{(0)} \rangle \langle \mathbf{z}_n \rangle^\top \langle W \rangle^\top))\right], \tag{88}$$

  where we replace $\frac{1}{N}\sum_{n=1}^{N} \langle \boldsymbol{\xi}_n^{(0)} \rangle$, $\frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_n$, and $\frac{1}{N}\sum_{n=1}^{N} \langle \boldsymbol{\xi}_n^{(0)} \rangle \langle \mathbf{z}_n \rangle^\top$ with perturbed expected sufficient statistics.

- TPBN shrinkage priors:

$$\begin{aligned}
q(\zeta_{j,k}) &= \mathcal{GIG}(0, 2\langle \xi_{j,k} \rangle_{q(\xi)}, \langle w_{j,k}^2 \rangle_{q(\boldsymbol{\theta})}), \\
q(\xi_{j,k}) &= \mathrm{Gam}(1, \langle \zeta_{j,k} \rangle_{q(\zeta)}), \\
q(\phi_k) &= \mathrm{Gam}(\tfrac{J}{2} + \tfrac{1}{2}, \langle \omega \rangle_{q(\omega)} + \sum_{j=1}^{J} \langle \xi_{j,k} \rangle_{q(\xi)}), \\
q(\omega) &= \mathrm{Gam}(\tfrac{K}{2} + \tfrac{1}{2}, 1 + \sum_{k=1}^{K} \langle \phi_k \rangle_{q(\phi)}).
\end{aligned}$$

  When updating these TPBN shrinkage priors, we first calculate $q(\zeta_{j,k})$ using a data-independent initial value for $2\langle \xi_{j,k} \rangle_{q(\xi)}$ and perturb $\langle w_{j,k}^2 \rangle_{q(\boldsymbol{\theta})}$ (since $q(c)$ is perturbed). Using this perturbed $q(\zeta_{j,k})$, we calculate $q(\xi_{j,k})$. Then, using $q(\xi_{j,k})$, we calculate $q(\phi_k)$ with some data-independent initial value for $\langle \omega \rangle_{q(\omega)}$. Then, finally, we update $q(\omega)$ using $q(\phi_k)$. In this way, these TPBN shrinkage priors are perturbed.

6. The EM Algorithm and its Relationship to VBEM

In contrast to VBEM, which computes an approximate posterior distribution, EM finds a point estimate of model parameters $\boldsymbol{m}$. A derivation of EM from a variational perspective, due to Neal & Hinton (1998), shows that VBEM generalizes EM (Beal, 2003). To derive EM from this perspective, we begin by identifying a lower bound $\mathcal{L}(q, \boldsymbol{m})$ on the log-likelihood

using another Jensen's inequality argument, which will serve as a proxy objective function. For any parameters $\boldsymbol{m}$ and auxiliary distribution over the latent variables $q(\boldsymbol{l})$,

$$\log p(\mathcal{D};\boldsymbol{m}) = \log\Big(\int d\boldsymbol{l}\; p(\boldsymbol{l},\mathcal{D};\boldsymbol{m})\frac{q(\boldsymbol{l})}{q(\boldsymbol{l})}\Big) = \log\Big(\mathbb{E}_q\Big[\frac{p(\boldsymbol{l},\mathcal{D};\boldsymbol{m})}{q(\boldsymbol{l})}\Big]\Big)$$
$$\geq \mathbb{E}_q\Big[\log p(\boldsymbol{l},\mathcal{D};\boldsymbol{m}) - \log q(\boldsymbol{l})\Big] = \mathbb{E}_q\Big[\log p(\boldsymbol{l},\mathcal{D};\boldsymbol{m})\Big] + H(q) \triangleq \mathcal{L}(q,\boldsymbol{m}). \quad (89)$$

Note that $\mathcal{L}(q,\boldsymbol{m})$ has a very similar definition to the ELBO from VB (Eq. 7). Indeed, we observe that for an instance of VBEM where $q(\boldsymbol{m})$ is restricted to being a Dirac delta function at $\boldsymbol{m}$, $q(\boldsymbol{m}) = \delta(\boldsymbol{m}-\boldsymbol{m}^*)$, and with no prior on $\boldsymbol{m}$, we have $\mathcal{L}(q(\boldsymbol{l}),\boldsymbol{m}) = \mathcal{L}(q(\boldsymbol{l},\boldsymbol{m}))$ (Beal, 2003). The EM algorithm can be derived as a coordinate ascent algorithm which maximises $\mathcal{L}(q,\boldsymbol{m})$ by alternatingly optimizing q (the E-step) and $\boldsymbol{m}$ (the M-step) (Neal & Hinton, 1998). At iteration i, with current parameters $\boldsymbol{m}^{(i-1)}$, the updates are:

$$q^{(i)} = \arg\max_q \mathcal{L}(q,\boldsymbol{m}^{(i-1)}) = p(\boldsymbol{l}|\mathcal{D};\boldsymbol{m}^{(i-1)}) \qquad (\textbf{E-step})$$
$$\boldsymbol{m}^{(i)} = \arg\max_{\boldsymbol{m}} \mathcal{L}(q^{(i)},\boldsymbol{m}) \qquad (\textbf{M-step}). \quad (90)$$

In the above, $\arg\max_q \mathcal{L}(q,\boldsymbol{m}^{(i-1)}) = p(\boldsymbol{l}|\mathcal{D};\boldsymbol{m}^{(i-1)})$ since this maximisation is equivalent to minimising $D_{KL}(q(\boldsymbol{l})\|p(\boldsymbol{l}|\mathcal{D};\boldsymbol{m}^{(i-1)}))$. To see this, we rewrite $\mathcal{L}(q,\boldsymbol{m})$ as

$$\begin{aligned}\mathcal{L}(q,\boldsymbol{m}) &= \mathbb{E}_q\Big[\log p(\boldsymbol{l},\mathcal{D};\boldsymbol{m})\Big] - \mathbb{E}_q[\log q(\boldsymbol{l})] \\ &= \log p(\mathcal{D};\boldsymbol{m}) + \mathbb{E}_q\Big[\log p(\boldsymbol{l}|\mathcal{D},\boldsymbol{m})\Big] - \mathbb{E}_q[\log q(\boldsymbol{l})] \\ &= \log p(\mathcal{D};\boldsymbol{m}) - D_{KL}(q(\boldsymbol{l})\|p(\boldsymbol{l}|\boldsymbol{m},\mathcal{D}))\;. \end{aligned} \quad (91)$$

From a VBEM perspective, the M-step equivalently finds the Dirac delta $q(\boldsymbol{m})$ which maximises $\mathcal{L}(q(\boldsymbol{l},\boldsymbol{m}))$. Although we have derived EM as optimizing the variational lower bound $\mathcal{L}(q,\boldsymbol{m})$ in each E- and M-step, it can also be shown that it monotonically increases the log-likelihood $\log p(\mathcal{D};\boldsymbol{m})$ in each iteration. The E-step selects $q^{(i)} = p(\boldsymbol{l}|\mathcal{D};\boldsymbol{m}^{(i-1)})$ which sets the KL-divergence in Eq. 91 to 0 at $\boldsymbol{m}^{(i)}$, at which point the bound is tight. This means that $\log p(\mathcal{D};\boldsymbol{m}^{(i-1)})$ is achievable in the bound, so $\max_{\boldsymbol{m}} \mathcal{L}(q^{(i)},\boldsymbol{m}) \geq \log p(\mathcal{D};\boldsymbol{m}^{(i-1)})$. As $\mathcal{L}(q,\boldsymbol{m})$ lower bounds the log likelihood at $\boldsymbol{m}$, this in turn guarantees that the log-likelihood is non-decreasing over the previous iteration. We can thus also interpret the E-step as computing a lower bound on the log-likelihood, and the M-step as maximizing this lower bound, i.e. an instance of the Minorise Maximise (MM) algorithm. EM is sometimes written in terms of the *expected complete data log-likelihood* $Q(\boldsymbol{m};\boldsymbol{m}^{(i-1)})$,

$$Q(\boldsymbol{m};\boldsymbol{m}^{(i-1)}) = \mathbb{E}_{p(\boldsymbol{l}|\mathcal{D};\boldsymbol{m}^{(i-1)})}[\log p(\boldsymbol{l},\mathcal{D};\boldsymbol{m})] \qquad (\textbf{E-step})$$
$$\boldsymbol{m}^{(i)} = \arg\max_{\boldsymbol{m}} Q(\boldsymbol{m};\boldsymbol{m}^{(i-1)}) \qquad (\textbf{M-step}). \quad (92)$$

This is equivalent to Eq. 90: $Q(\boldsymbol{m};\boldsymbol{m}^{(i-1)}) + H(p(\boldsymbol{l}|\mathcal{D};\boldsymbol{m}^{(i-1)})) = \mathcal{L}(p(\boldsymbol{l}|\mathcal{D};\boldsymbol{m}^{(i-1)}),\boldsymbol{m})$.

References

Abadi, M., Chu, A., Goodfellow, I., Brendan McMahan, H., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016*

ACM SIGSAC Conference on Computer and Communications Security (pp. 308–318). ACM.

Barthe, G., Farina, G. P., Gaboardi, M., Arias, E. J. G., Gordon, A., Hsu, J., & Strub, P. (2016). Differentially private Bayesian programming. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, (pp. 68–79). ACM.

Bassily, R., Smith, A. D., & Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, (pp. 464–473).

Beal, M. J. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Unit, University College London.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859–877.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Bun, M. & Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, (pp. 635–658). Springer.

Chaudhuri, K., Monteleoni, C., & Sarwate, A. D. (2011). Differentially private empirical risk minimization. *Journal of Machine Learning Research*, *12*, 1069–1109.

Daries, J. P., Reich, J., Waldo, J., Young, E. M., Whittinghill, J., Ho, A. D., Seaton, D. T., & Chuang, I. (2014). Privacy, anonymity, and big data in the social sciences. *Communications of the ACM*, *57*(9), 56–63.

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

Dimitrakakis, C., Nelson, B., Mitrokotsa, A., & Rubinstein, B. I. (2014). Robust and private Bayesian inference. In *Algorithmic Learning Theory (ALT)*, (pp. 291–305). Springer.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., & Naor, M. (2006). Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, (pp. 486–503). Springer.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, (pp. 265–284). Springer.

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

Dwork, C., Rothblum, G. N., & Vadhan, S. (2010). Boosting and differential privacy. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, (pp. 51–60). IEEE.

Dwork, C., Talwar, K., Thakurta, A., & Zhang, L. (2014). Analyze Gauss: optimal bounds for privacy-preserving principal component analysis. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, (pp. 11–20).

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

Gan, Z., Henao, R., Carlson, D. E., & Carin, L. (2015). Learning deep sigmoid belief networks with data augmentation. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent Dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems 23* (pp. 856–864). Curran Associates, Inc.

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, *14*(1), 1303–1347.

Husmeier, D., Dybowski, R., & Roberts, S. (2006). *Probabilistic modeling in bioinformatics and medical informatics*. Springer Science & Business Media.

Jälkö, J., Dikmen, O., & Honkela, A. (2017). Differentially Private Variational Inference for Non-conjugate Models. In *Uncertainty in Artificial Intelligence 2017, Proceedings of the 33rd Conference (UAI)*.

Jelinek, F., Mercer, R. L., Bahl, L. R., & Baker, J. K. (1977). Perplexity–a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, *62*(S1), S63–S63.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, *37*(2), 183–233.

Kairouz, P., Oh, S., & Viswanath, P. (2017). The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, *63*(6), 4037–4049.

Kifer, D., Smith, A., Thakurta, A., Mannor, S., Srebro, N., & Williamson, R. C. (2012). Private convex empirical risk minimization and high-dimensional regression. In *Proceedings of COLT*, (pp. 94–103).

Letham, B., Rudin, C., McCormick, T. H., & Madigan, D. (2014). Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. Department of Statistics Technical Report tr608, University of Washington.

McSherry, F. & Talwar, K. (2007). Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, (pp. 94–103).

Neal, R. M. (1992). Connectionist learning of belief networks. *Artif. Intell.*, *56*(1), 71–113.

Neal, R. M. & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 355–368). Kluwer Academic Publishers.

Nissim, K., Raskhodnikova, S., & Smith, A. D. (2007). Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, (pp. 75–84).

Park, M., Foulds, J. R., Chaudhuri, K., & Welling, M. (2017). DP-EM: Differentially private expectation maximization. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., & Koller, D. (2013). Tuned models of peer assessment in MOOCs. In *Proceedings of the 6th International Conference on Educational Data Mining*, (pp. 153–160).

Polson, N. G., Scott, J. G., & Windle, J. (2013). Bayesian inference for logistic models using Polya-gamma latent variables. *Journal of the American Statistical Association*, *108*(504), 1339–1349.

Rogers, R. M., Roth, A., Ullman, J., & Vadhan, S. (2016). Privacy odometers and filters: Pay-as-you-go composition. In *Advances in Neural Information Processing Systems*, (pp. 1921–1929).

Sarwate, A. D. & Chaudhuri, K. (2013). Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE Signal Process. Mag.*, *30*(5), 86–94.

Song, S., Chaudhuri, K., & Sarwate, A. (2013). Stochastic gradient descent with differentially private updates. In *Proceedings of GlobalSIP*.

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

Wang, Y.-X., Balle, B., & Kasiviswanathan, S. P. (2019). Subsampled Renyi differential privacy and analytical moments accountant. In Chaudhuri, K. & Sugiyama, M. (Eds.), *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019*, volume 89 of *Proceedings of Machine Learning Research*, (pp. 1226–1235).

Wang, Y.-X., Fienberg, S., & Smola, A. (2015). Privacy for free: Posterior sampling and stochastic gradient Monte Carlo. In Bach, F. & Blei, D. (Eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, (pp. 2493–2502)., Lille, France.

Wang, Y.-X., Lei, J., & Fienberg, S. E. (2016). Learning with differential privacy: Stability, learnability and the sufficiency and necessity of ERM principle. *Journal of Machine Learning Research*, *17*(183), 1–40.

Welling, M. & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, (pp. 681–688).

Wu, X., Kumar, A., Chaudhuri, K., Jha, S., & Naughton, J. F. (2016). Differentially private stochastic gradient descent for in-RDBMS analytics. *CoRR*, *abs/1606.04722*.

Zhang, Z., Rubinstein, B., & Dimitrakakis, C. (2016). On the differential privacy of Bayesian inference. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*.