Erasmus University Rotterdam

Erasmus School of Economics

Bachelor Thesis Econometrics and Operations Research

Stochastic Gradient Descent With Differentially Private Updates

Author:
Rayel Hardwarsing

Student ID number:
431003

Supervisor:
prof.dr. S.I. Birbil

Second assessor:
dr. F. Frasincar

July 8, 2018

Abstract

In recent decades, the amount of data available has grown immensely. A lot of this data may be private or sensitive. Privacy of of this data is very important, which is why algorithms that can operate on this data without violating privacy have become crucial. A framework for designing such algorithms is differential privacy. In this paper we propose differentially private versions of single-point and mini-batch stochastic gradient descent (SGD) and use these for optimizing the objective for logistic regression. We use several data sets of varying sizes to test the algorithms. We conclude that the performance of mini-batch differentially private SGD is very close to non-private SGD, in contrast to single-point differentially private SGD, which does not converge and has a high variance. This holds for both low and high dimensional problems. We also conclude that deciding on hyperparameters is not an easy choice. All the results mentioned before are obtained with doing a single pass through the data sets. We also test the effect of doing multiple passes through the data set for single-point differentially private SGD. This decreases the level of privacy and does not increase performance as much as mini-batching does.

1 Introduction

In this day and age, as data becomes easier to collect, the amount of data available grows. This amount of data available brings a lot of benefits. It helps with innovations beneficial to society, such as smart cars and health monitoring devices. However, a lot of this data may be private or sensitive, such as health information or financial records. Because this sort of personal data is easily abused, privacy of the data is very important. Even though we need to respect privacy, institutions want to use this data to learn more about their customers or users. They should be able to do this without violating the privacy of the users and customers. This is why efficient algorithms that can operate on this data without violating privacy have become crucial.

Differential privacy [6] is a framework for designing such algorithms. The basic idea of differential privacy is that algorithms and/or statistical queries produce the same results, regardless of an individual's information being in the database or not. By applying differential privacy, it is not possible to get information about a single individual in a data set based on the result of a query or algorithm.

If we want to use an algorithm that guarantees differential privacy, we need to approximate the algorithm. This approximation of course affects the performance of the algorithm. An example of this is that the amount of data needed increases as the required level of privacy increases. Since we need a lot of data to guarantee privacy, using a scalable algorithm is ideal. Stochastic gradient descent (SGD) algorithms are very popular right now for optimization problems as they are simple, scalable and their performance is asymptotically equal to other methods which are more computationally expensive [11, 13]. In this paper we look at both differentially private single-point SGD and mini-batch SGD and apply them to several data sets. This algorithm works for general optimization problems, but we consider logistic regression for classification tasks.

We find that differentially private mini-batch SGD has a much better performance than differentially private single-point SGD. Using mini-batching decreases the variance and is almost identical to non-private SGD. This result also holds for high dimensional problems. We also argue that it is difficult to choose general values for the parameters for differentially private SGD.

In the next section, we discuss some related literature, and how our research contributes to the existing literature. Thereafter, the methodology is explained in Section 3. In Section 4, we discuss the experiments and their results and in Section 5 overall conclusions of the paper are given.

2 Related literature

There has already been a lot of research done on differentially private optimization algorithms. Abadi et al. [2] developed algorithmic techniques which are able to train deep neural networks with non-convex objectives under a privacy budget. They test their algorithms on the MNIST [9] data set, and achieve an accuracy of 97%, while still guaranteeing a certain level of privacy.

Rajkumar et al. [10] use a differentially private algorithm based on a stochastic gradient descent based procedure for multiparty classification. Different parties are all a part of a data set. They learn a classifier on the entire data set, without revealing information about which data points are in which party.

Differential privacy for scalable stochastic gradient descent-based analytics have been researched by Wu et al. [16]. They propose an algorithm that successfully tackles the two main issues that come up when using differentially private stochastic gradient descent in a scalable anaytics framework: (1) low model accuracy due to noise addition and (2) runtime overhead of the private algorithms.

The work closest to our research is done by Song et al. [14]. They propose a differentially private version of stochastic gradient descent. The algorithm is tested on several data sets. They experiment with mini-batching and different hyperparameters. Their findings were that mini-batching improves the performance of the algorithm a lot and that choosing the appropriate hyperparameters is not very straightforward.

In this research we show a replication of the results of Song et al. [14]. Furthermore, as extensions, we perform the algorithms on different data sets of varying sizes, look at high dimensional problems, and try multiple passes (epochs) through the data sets.

3 Methodology

In this section we introduce differential privacy to get a basic understanding of this concept. After this we give the problem description for logistic regression, the problem which the algorithms are used for. Then the algorithms are presented. We start with non-private stochastic gradient descent, after which we introduce a differentially private version.

3.1 Differential privacy

Differential privacy is a recent framework to preserve privacy of the data of individuals. The basic idea behind differential privacy is that differential private algorithms will give the same results regardless of whether a particular individual is in the data set or not. This means that the participation of a particular individual in a data set does not influence the outcome of the algorithm. The way that differential privacy achieves this is by adding random noise to the results of the algorithm.

Now that we have introduced the basic idea of differential privacy, we will give a formal definition. We use the definition as given in Song et al. [14]: A (randomized) algorithm $\mathcal{A}$ whose outputs lie in a domain $\mathcal{S}$ is said to be $\alpha -$ differentially private if for all $\mathrm{S} \subseteq \mathcal{S}$, for all data sets $\mathcal{D}$ and $\mathcal{D}'$ that differ in the value of a single individual, it is the case that

$$\Pr(\mathcal{A}(\mathcal{D}) \in \mathrm{S}) \leq e^{\alpha} \Pr(\mathcal{A}(\mathcal{D}') \in \mathrm{S}) \tag{1}$$

which means that the log-likelihood ratio of the outputs of the algorithm under the data sets $\mathcal{D}$ and $\mathcal{D}'$ is smaller than α. The privacy parameter α specifies privacy risk. A lower α means higher privacy.

3.2 Problem description

We use the problem description similar to the problem description in Song et al. [14]. We use the algorithms to optimize the objective for logistic regression. In logistic regression the data are n labelled examples (x_1, y_1),...,(x_n, y_n), where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. We assume that the norm $\|\mathrm{x}_i\| \leq 1 \; \forall i$. The goal is to find a hyperplane through the origin that largely separates the examples labeled 1 from those labeled -1. The optimization problem is defined as follows:

$$w^* = \underset{w}{\operatorname{argmin}} \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} \ell(w, x_i, y_i) \tag{2}$$

where $\ell(w, x, y) = \log(1 + e^{-yw^\top x})$ is the logistic loss function and $w \in \mathbb{R}^d$ is the normal vector to the hyperplane separator. λ serves as a regularization parameter.

3.3 Stochastic gradient descent

To optimize the objective function we use stochastic gradient descent. We consider two versions of stochastic gradient descent, namely single-point SGD and mini-batch SGD. For single-point SGD, we start at an initial point w_0, after which we update w at step t as

$$w_{t+1} = w_t - \eta_t(\lambda w_t + \nabla \ell(w_t, x_t, y_t)) \tag{3}$$

where η_t is the learning rate at step t and $\nabla \ell(w_t, x_t, y_t)$ is the gradient for a single example, that is a single observation (x_t, y_t) in the data set. The gradient of our objective function can be computed as

$$\nabla \ell(w_t, x_t, y_t) = \frac{-y_t e^{-y_t w_t^\top x_t}}{1 + e^{-y_t w_t^\top x_t}} x_t = \frac{-y_t}{1 + e^{y_t w_t^\top x_t}} x_t \tag{4}$$

For mini-batch SGD, we update w at step t based on a randomly chosen batch of examples B_t of size b:

$$w_{t+1} = w_t - \eta_t(\lambda w_t + \frac{1}{b} \sum_{(x_i, y_i) \in B_t} \nabla \ell(w_t, x_i, y_i)) \tag{5}$$

3.4 Differentially private SGD

To produce a differentially private SGD algorithm, we add random noise to the updates. We now update w at step t for single-point differentially private SGD as:

$$w_{t+1} = w_t - \eta_t(\lambda w_t + \frac{1}{b} \sum_{(x_i, y_i) \in B_t} \nabla \ell(w_t, x_i, y_i) + Z_t), \tag{6}$$

where $Z_t \in \mathbb{R}^d$ is a random Laplace noise vector drawn independently from the Laplace$(\frac{2}{\alpha})$ distribution with the density

$$\rho(z) \propto e^{-(\alpha/2)\|z\|} \tag{7}$$

Recall that α is a privacy parameter. A lower α means a higher level of privacy.

Sampling from this distribution guarantees differential privacy, which is shown in Dwork et al. [7]. For mini-batch differentially private SGD we update w at step t as

$$w_{t+1} = w_t - \eta_t(\lambda w_t + \frac{1}{b} \sum_{(x_i, y_i) \in B_t} \nabla \ell(w_t, x_i, y_i) + \frac{1}{b} Z_t) \tag{8}$$

Note that in this case we sample only once for every batch, and not for every observation.

To sample the Laplace noise Z_t for differential private SGD as described before, we use the algorithm as described in Wu et al. [15]:

Algorithm 1 Laplace noise sampling algorithm

```
1. Sample a uniform vector $\mathbf{v} \in \mathbb{R}^d$ from the unit ball
2. Sample a magnitude $l \in \mathbb{R}$ from a Gamma distribution $\Gamma(d, \frac{2}{\alpha})$
3. Now calculate the noise as $\mathrm{Z}_t = l\mathbf{v}$
```

4 Simulation study

In this section we describe the data sets used to evaluate the performance of our algorithms. We also give some insight in how these data sets are preprocessed. After the introduction of the data sets, we discuss what experiments we do. Finally, we discuss the results and observations.

4.1 Data Sets and Preprocessing

We apply the algorithms in this research to several data sets. We consider two kinds of data sets, namely synthetic and real data sets. The synthetic data set consists of 10,000 samples drawn uniformly from a 5-dimensional sphere, and is linearly separable with margin 0.001.

The first real data set we use is the KDDCup 99 [12] data set, on which we perform a normal vs. malicious classification task for network connections. This data set contains $d = 34$ continuous features. We take a subsample of 50,000 observations for our experiments.

The second real data set is the MNIST data set which consists of 60,000 images of handwritten digits (0 to 9). For this data set, we perform a "1 vs all" classification task, where we do a binary classification between the digit 0 versus the other digits. The MNIST data set has $d = 784$ features, which represent the grey levels of all the pixels of the picture.

We also use the Banknote authentication data set [4], which consists of 1,327 images and $d = 5$ features of genuine and forged banknotes. The task is to determine whether a banknote is genuine or forged.

The last real data set we use is the Occupancy detection data set [3]. This data is used for a binary classification on whether a room is occupied or not based on the temperature of the room, the humidity and some other environmental features. The data set consists of 8,143 observations and has $d = 6$ features. Note that the last two real data sets are significantly smaller than the MNIST and KDDCup data set. It will be interesting to see how the algorithms perform on these smaller sized data sets.

The data is stored in a matrix form, where each row represents an observation and each column represents a feature. We preprocess the real data sets by first normalizing each row, such that for each observation vector holds that $\|\mathrm{x}_i\| = 1$. To reduce the dimension of the data, we do a random projection. This random projection is only done on the KDDCup and MNIST data sets, since the other data sets already have a small number of features. For the KDDCup data set we use a dimension d = 9, and for MNIST we use d = 15. The random projection is done by just picking the number of features in the reduced dimension randomly from the original data.

4.2 Experiments

We run the algorithms for the batch sizes $b \in \{1, 2, 5, 10, 20, 50\}$. We use $\lambda = 0.0001$ and $\alpha = 1$ for our regularization and privacy parameter. We will look at the influence of using mini-batch differentially private SGD instead of single-point differentially private SGD. After this we discuss batch sizes and learning rates. We try $\eta_t = 1/\sqrt{t}$ and a larger learning rate $\eta_t = 10/\sqrt{t}$. A larger learning rate may converge faster, but it can skip the minimum of the objective value if it is too large. These are the experiments which are also done in Song et al. [14]. The next experiments we propose are extensions of their work.

For the runs mentioned above, we use a single pass over the data set. It is also of our interest to see if doing multiple passes through the data set will have an influence, especially for the smaller data sets. Note that by doing multiple passes over the data set, we will lose some privacy. We discuss this further in Section 4.3.

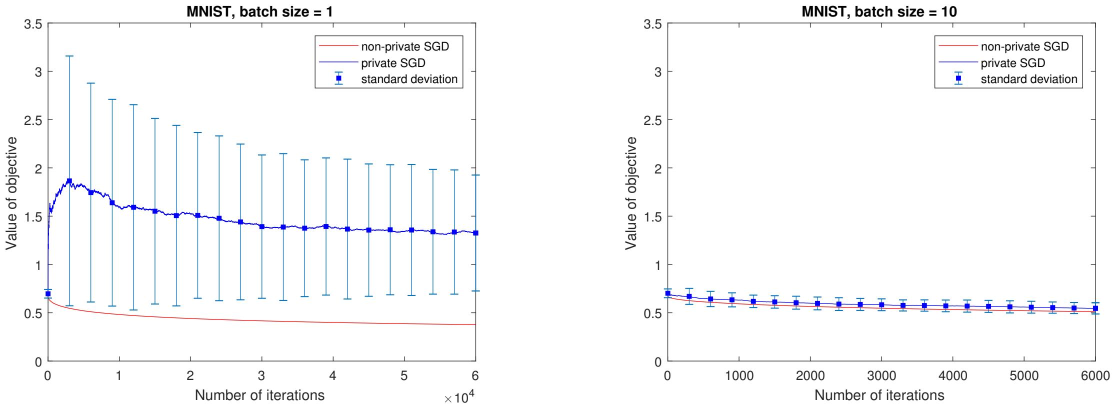
Last but not least we test the performance of the algorithms on a higher dimensional problem. For the before mentioned experiments, we do a random projection as described in the data preprocessing part to reduce the dimension of the data. But performance for higher dimensions is also important, so we will evaluate how well the algorithms work on the original KDDCup data set without dimension reduction, where the number of features is $d = 34$. Here we also look at the influence of using mini-batch differentially private SGD instead of single-point differentially private SGD.

In all the experiments, we take the average value of the objective value over 20 random permutations of the data with new samples for the noise for every permutation. In each permutation, the order of the observations in the data set is chosen randomly, but we still use every observation from the data set. The error bars in the plots are at one standard deviation of the objective, based on the outcomes of the 20 random permutations.

4.3 Results and observations

The results are discussed in this section. We first look at the influence of mini-batching, after which we show the results for the different batch sizes and learning rates. We also provide results for doing multiple passes, and show how this affects the level of privacy through the privacy parameter α. The last result we discuss is how the algorithms performs for a higher dimensional problem. The code for all the algorithms and experiments is written in MATLAB, and can be found on our GitHub page [1].

Effects of mini-batching. Figure 1 below shows the objective value versus the number of iterations for the MNIST data set for learning rate $\eta_t = 1/\sqrt{t}$. If we look at the graph for batch size $b = 1$ (single-point SGD), we observe that differentially private SGD performs much worse than non-private SGD. The objective value for differentially private SGD after 60,000 iterations is far from non-private SGD and it does not seem that differentially private SGD converges. This can be explained by the fact that for batch $b = 1$ noise is added for every observation. Another thing we notice is that the variance is very high. Now if we look at the graph for batch size $b = 10$, we see that differentially private SGD is almost identical to non-private SGD. It converges and the variance is a lot smaller. Adding noise less often seems to help the performance of differentially private SGD a lot.

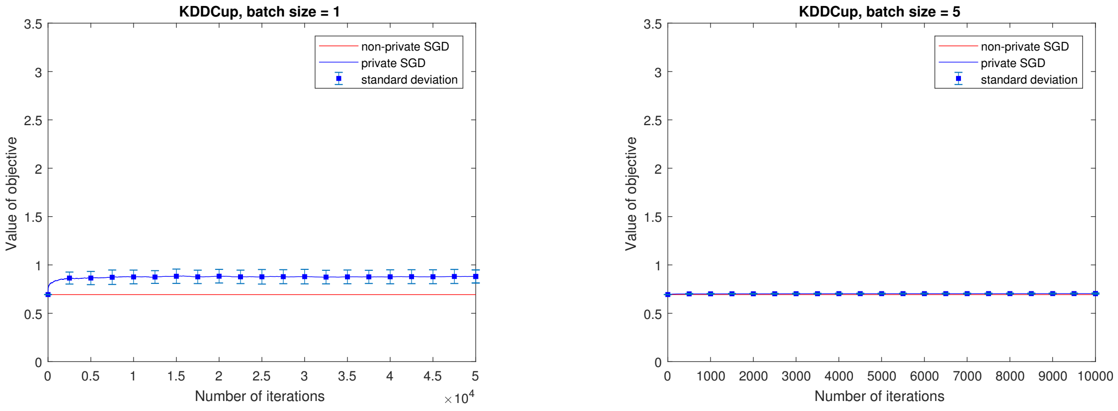


(a)

(b)

Figure 1: Objective value vs. number of iterations for differentially private and non-private SGD performed on the MNIST dataset. (a) is for batch size = 1, (b) is for batch size = 10.

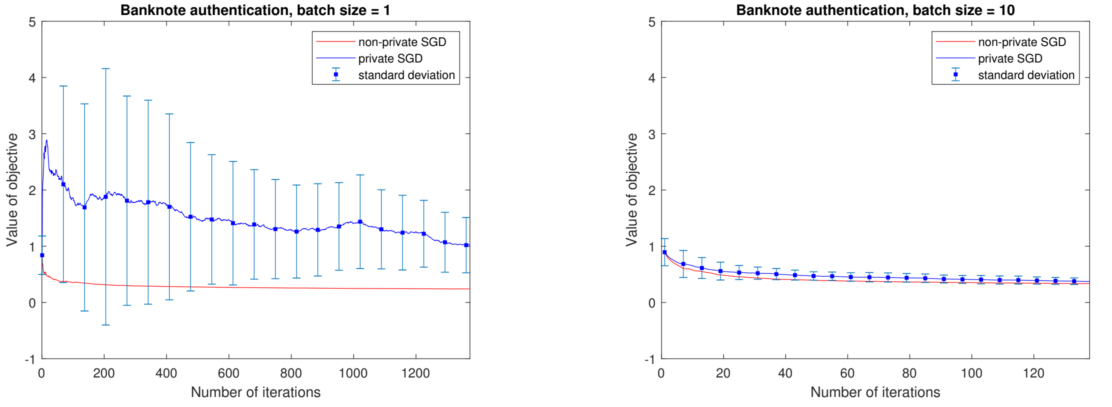
The results for the KDDCup data set, which can be seen in figure 2 below, show similar patterns. For this data set differentially private SGD is almost identical to non-private SGD when we increase the batch to $b = 5$. The only noticable difference from the results for the MNIST data set is that the variance for batch size $b = 1$ is smaller for this data set.



(a) (b)

Figure 2: Objective value vs. number of iterations for differentially private and non-private SGD performed on the KDDCup dataset. (a) is for batch size = 1, (b) is for batch size = 5.

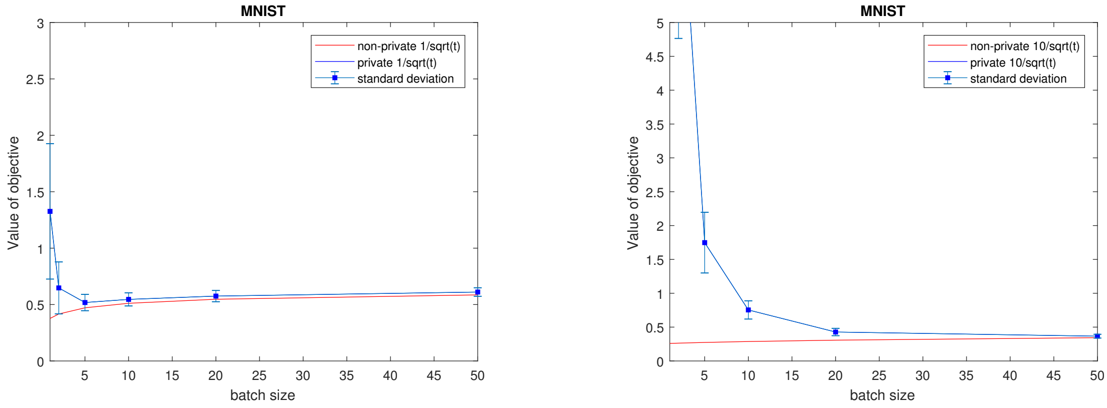
The MNIST and KDDCup are relatively large data sets. The Banknote authentication data set is significantly smaller. In figure 3 one can see that the results observed for the banknote authentication data set are in line with the results for the bigger data sets. Differentially private SGD is almost identical to non-private SGD with a smaller variance when we increase the batch size to $b = 10$. These results show that guaranteeing differential privacy is not very costly in the sense that the private and non-private SGD are almost identical when moderately increasing the batch size. Note that these results are based on the learning rate $\eta_t = 1/\sqrt{t}$ and a reduced dimension. These are some things that are discussed further in the research.



(a) (b)

Figure 3: Objective value vs. number of iterations for differentially private and non-private SGD performed on the Banknote authentication dataset. (a) is for batch size = 1, (b) is for batch size = 10.

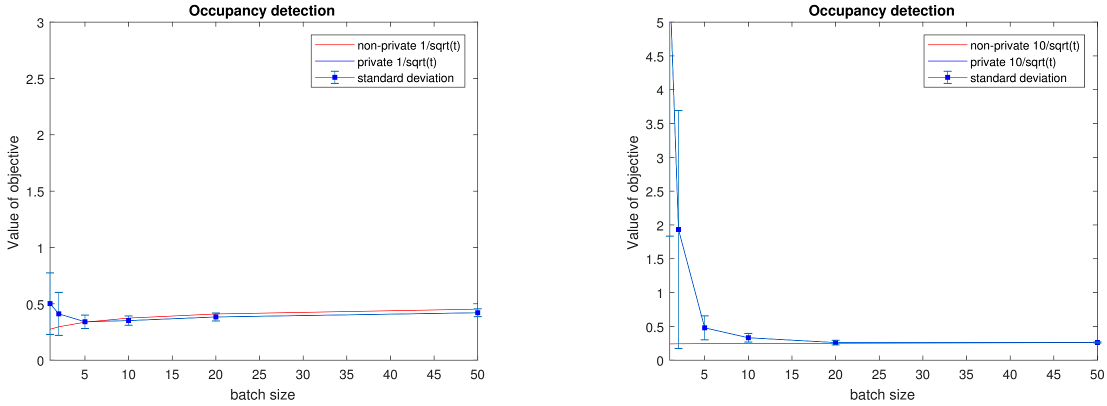
Choosing parameters. In the previous section we have seen that increasing the batch size improves perfomance. However, we do not know how different batch sizes for mini-batching influence the performance of the algorithms. Figures 4 and 5 show the batch size versus the objective value for the MNIST and Occupancy detection data sets. As observed earlier, increasing the batch size improves the performance of differentially private SGD, but increasing it too much will actually degrade the performance. For both data sets, we observe that with learning rate $\eta_t = 1/\sqrt{t}$, a batch size bigger than 5 degrades performance. The reason for this is that an increase in the batch size leads to a decrease in the number of iterations, which is costly when only making a single pass through the data sets. Figure 4 and 5 also show that for a bigger learning rate $\eta_t = 10/\sqrt{t}$, the algorithms degrade less.



(a) learning rate $\eta_t = 1/\sqrt{t}$

(b) learning rate $\eta_t = 10/\sqrt{t}$

Figure 4: Objective value vs. batch size for differentially private and non-private SGD performed on the MNIST data set.



(a) learning rate $\eta_t = 1/\sqrt{t}$

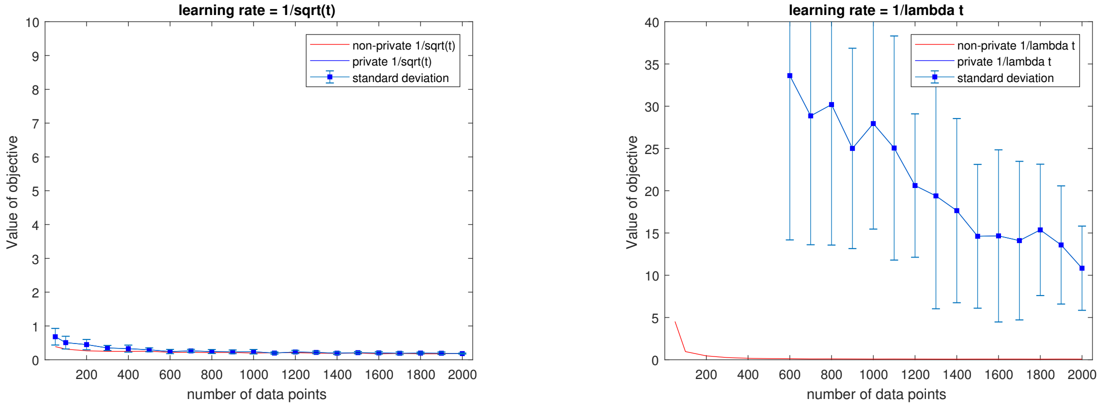
(b) learning rate $\eta_t = 10/\sqrt{t}$

Figure 5: Objective value vs. batch size for differentially private and non-private SGD performed on the Occupancy detection data set.

Now that we discussed the batch size, we have a look at choosing the appropriate learning rate. Rakhlin et al. [11] suggest that the learning rate $\eta_t = 1/\lambda t$ guarantees fast convergence when the objective function is strongly convex. However with the choice of $\lambda = 0.0001$, our objective function is not very strongly convex.

To display how important the choice of a good learning rate is for differentially private SGD, we perform the algorithms with two different learning rates $\eta_t = 1/\sqrt{t}$ and $\eta_t = 1/\lambda t$ on the synthetic data set. For this experiment we use batch size $b = 5$. Figure 6 shows the objective value versus the number of data points, where 2,000 data points means that we went through all the 10,000 observations (2,000 x (batch size = 5) = 10,000) and 200 data points means that we only used 1,000 observations. We observe that if we use the faster decreasing learning rate $\eta_t = 1/\lambda t$, the variance of differentially private SGD increases substantially and the objective value is still very large, even when we use all the observations.

Choosing a learning rate is often not an easy decision. For differentially private algorithms, this is even harder, because the random noise is also a factor to take into account.



(a) learning rate $\eta_t = 1/\sqrt{t}$

(b) learning rate $\eta_t = 1/\lambda t$

Figure 6: Objective value vs. number of data points for differentially private and non-private SGD performed on the synthetic data set with batch size $b = 5$.

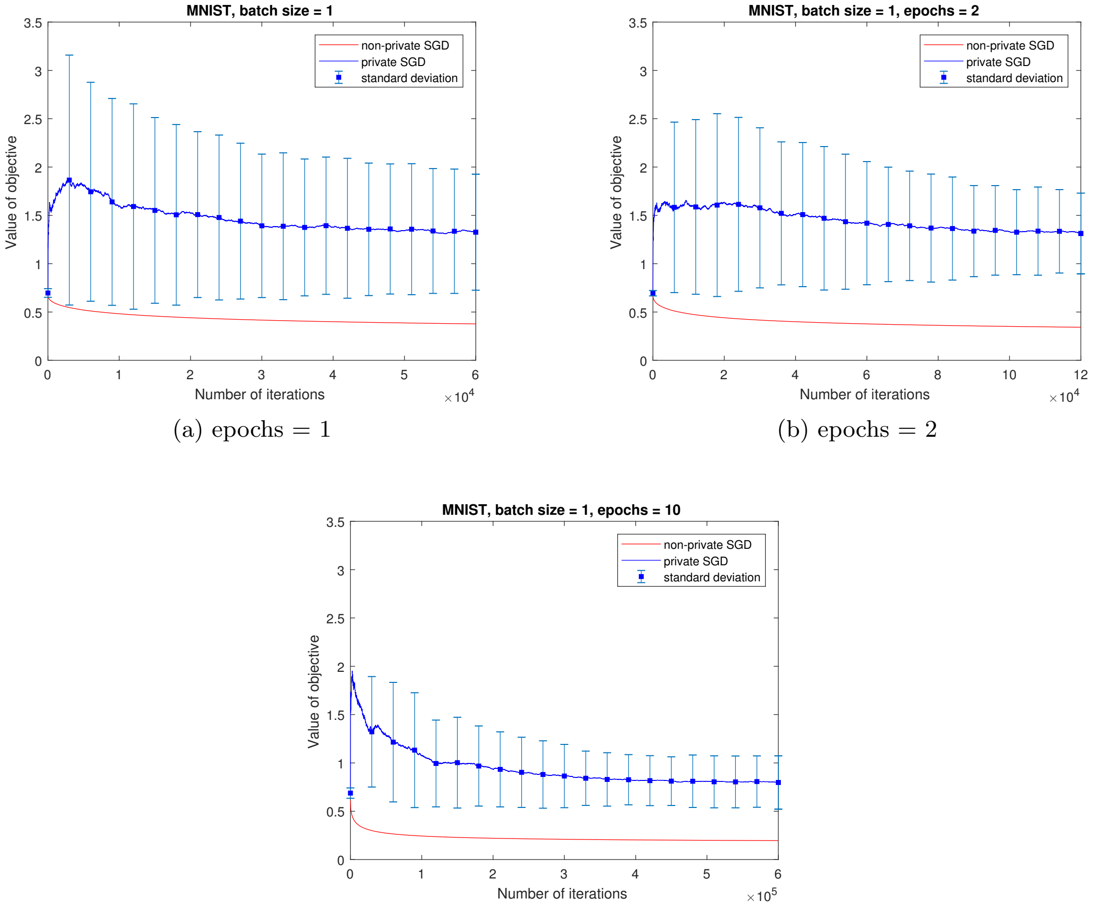
Multiple passes through the data set. In all the experiments before, we made a single pass through the data sets. We observed that in this case, for batch size $b = 1$, differentially private SGD performs poorly. In this experiment we make multiple passes (epochs) through the data set for batch size $b = 1$ and discuss if these results stay the same.

If we do multiple passes through the data set, we will lose some privacy. This follows from the following theorem, which is due to Dwork [5] and given in Kuzu et al. [8]:

Theorem 1 *Let $\mathcal{A}_i$ be ϵ_i-differentially private for $i = 1,...,T$. The sequence $(\mathcal{A}_1, ..., \mathcal{A}_T)$, whose output is the concatenation of the outputs of the individual algorithms, provides $\sum_{i=1}^{T} \epsilon_i$-differential privacy.*

In our case, this means that for a given number of epochs, the level of privacy when we use multiple passes is $\sum_{i=1}^{\text{epochs}} \alpha_i$, where each $\alpha_i = \alpha$, and α is the privacy parameter for differentially private SGD with a single pass through the data set. To make this more clear, we provide an example. Suppose we perform differentially private SGD with a single pass through the data set, with a privacy level of $\alpha = 0.01$. If we now want to do ten passes through the data set, the privacy level of the algorithm now changes to $\sum_{i=1}^{10} \alpha_i = 0.1$. Recall that a lower α means higher privacy.

Figure 7 shows the objective value versus the number of iterations for the MNIST data set for multiple epochs. We see that there is almost no difference between a single epoch and two epochs. However, for 10 epochs, there is a clear improvement in comparison with a single pass. The objective value is much closer to that of non-private SGD. But even though it is a lot better, it still has not converged.



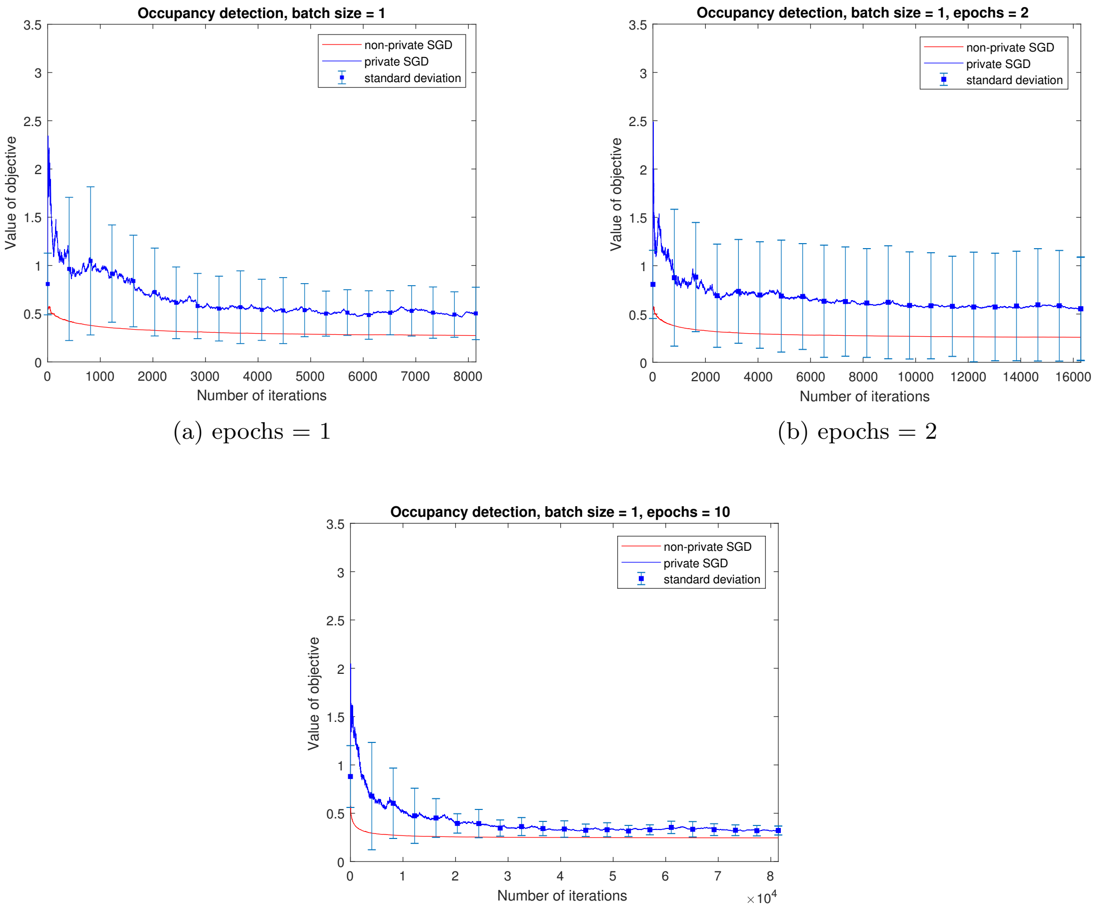
(a) epochs = 1

(b) epochs = 2

(c) epochs = 10

Figure 7: Objective value vs. number of iterations for multiple passes (epochs) through the MNIST data set with batch size = 1

The results for multiple passes through the Occupancy detection data set, shown in figure 8, show a similar pattern. Making two passes through the entire data set does not have a lot influence. When we make 10 passes, we see a huge improvement, but differentially private SGD is still not equal to non-private SGD. The fact that the improvement is better for the Occupancy detection data set is bigger than for the MNIST data set can be explained by the difference in the number of features between the data sets. In this experiment, we used the reduced dimension $d = 15$ for the MNIST data set, while the dimension of the Occupancy detection data set is only 4. The noise vector for the MNIST data set will therefore contain larger values, which results in a bigger disturbance in the algorithm.



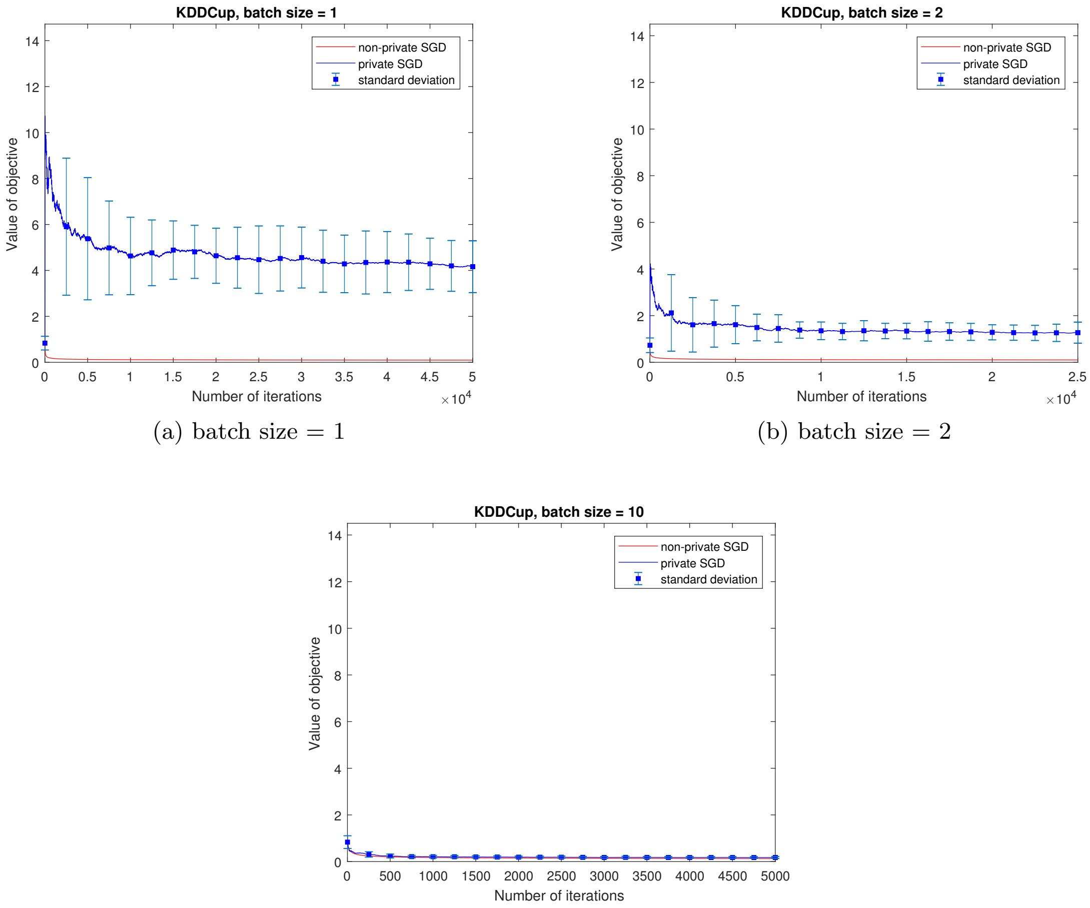
(a) epochs = 1

(b) epochs = 2

(c) epochs = 10

Figure 8: Objective value vs. number of iterations for multiple passes (epochs) through the Occupation detection data set with batch size = 1

Higher dimensional problems. Until now, all the experiments we have done are with relatively low dimensional problems. Recall that we did a random projection to reduce the dimension for the MNIST and KDDCup data sets. In this experiment we perform the algorithms on a higher dimensional data set, which is the KDDCup data set without dimension reduction, which has a dimension of $d = 34$. Figure 9 shows the objective value versus the number of iterations for different batch sizes. We observe that the results for this higher dimensional problem are in line with the results for lower dimensional problems. We see that for batch size $b = 1$, differentially private SGD does not converge and has a high variance. However, when we look at the results for batch size $b = 10$, we see that there is barely any difference between differentially private SGD and non-private SGD. This shows that even for a higher dimensional problem, where the noise vector has much larger values, mini-batching is still very effective.



(a) batch size = 1

(b) batch size = 2

(c) batch size = 10

Figure 9: Objective value vs. number of iterations for the KDDCup data set with dimension $d = 34$

5 Conclusion

In this research, a differentially private SGD algorithm is presented. We introduce single-point differentially private SGD and mini-batch differentially private SGD. The algorithms are tested on logistic regression. We found that single-point differentially private SGD does not perform well. However, mini-batching improves the performance of the algorithm significantly. For a moderate increase in batch size, there is almost no difference between non-private SGD and differentially private SGD. We also looked at how to choose the batch size and learning rate for the algorithms. We concluded that increasing the batch size too much will degrade performance, and that choosing a learning rate is a difficult choice for differentially private SGD, where we also have to take the noise addition into account.

For all the before mentioned results we only did one pass through the entire data set. The next experiment we did is doing multiple passes through the data sets for single-point differentially private SGD. Doing multiple passes results in a lower level of privacy. We found that doing multiple passes helps, but not enough to the point where differentially private SGD is equal to non-private SGD. Another thing we concluded is that the improvement when doing multiple passes was bigger for the smaller data set than for the bigger data set. The last experiment we performed is testing the algorithm on a high dimensional problem. We concluded that even for a higher dimensional problem, where the noise vector contains much larger values, mini-batching is still very effective.

A direction for future work would be to research how the value for the privacy parameter α

influences the results of the experiments and to investigate if and how much more data we need if we demand a higher level of privacy. In this research, we focused on the objective function value when discussing the results. It would also be interesting to look at prediction accuracy when evaluating the performance of the differentially private algorithm.

References

[1] MATLAB code: `https://github.com/RayelH/Differentially-Private-SGD`.

[2] M. Abadi, A. Chu, I. Goodfellow, H. Brendan McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. *arXiv:1607.00133*, pages 1 – 14, 2016.

[3] Luis M. Candanedo and Véronique Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. *Energy and Buildings*, 112:28 – 39, 2016.

[4] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. `http://archive.ics.uci.edu/ml`, Last accessed: 04-04-2013.

[5] Cynthia Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, pages 1–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[7] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC '10, pages 715–724, New York, NY, USA, 2010. ACM.

[8] N. Kuru, S. Yildirim, and S. Birbil. A differentially private stochastic gradient descent algorithm with smoothing. *Privacy in Machine Learning and Artificial Intelligence FAIM 2018 Workshop, Stockholm, July 15*, 2018.

[9] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[10] Arun Rajkumar and Shivani Agarwal. A differentially private stochastic gradient descent algorithm for multiparty classification. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 933–941, La Palma, Canary Islands, 21–23 Apr 2012. PMLR.

[11] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1571–1578, USA, 2012. Omnipress.

[12] Saharon Rosset and Aron Inger. Kdd-cup 99: Knowledge discovery in a charitable organization's donor database. *SIGKDD Explor. Newsl.*, 1(2):85–90, January 2000.

[13] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming*, 127(1):3–30, Mar 2011.

[14] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248, 2013.

[15] Xi Wu, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey F. Naughton. Differentially private stochastic gradient descent for in-RDBMS analytics. *CoRR*, abs/1606.04722, 2016.

[16] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 1307–1322, New York, NY, USA, 2017. ACM.