 doi:10.20944/preprints202209.0176.v1

Article

Differentially private block coordinate descent for linear regression on vertically partitioned data

Jins de Jong [1,†*], Bart Kamphorst [2,†] and Shannon Kroes [3,†]

[1] jins.dejong@tno.nl
[2] bart.kamphorst@tno.nl
[3] shannon.kroes@tno.nl
* Correspondence: jins.dejong@tno.nl
† TNO, Unit ICT, The Netherlands
Alphabetically ordered list of authors.

Abstract: We present a differentially private extension of the block coordinate descent based on objective perturbation. The algorithm iteratively performs linear regression in a federated setting on vertically partitioned data. In addition to a privacy guarantee, the algorithm also offers a utility guarantee; a tolerance parameter indicates how much the differentially private regression may deviate from an analysis without differential privacy. The algorithm's performance is compared with the standard block coordinate descent algorithm and the trade-off between utility and privacy is studied. The performance is studied using both artificial test data and the forest fires data set. We find that the algorithm is fast and able to generate practical predictions with single-digit privacy budgets, albeit with some accuracy loss.

Keywords: Differential privacy, Federated learning, Vertically partitioned data

1. Introduction

There are many circumstances, where organizations use each other's data to improve their performance[1–3]. However, sharing data is often undesirable or simply forbidden. In such situations privacy enhancing techniques can provide a solution. An example of this is federated learning, where the data is kept local, ensuring that no other party gets access to it. This is done for vertically partitioned data sets. In such data sets all contributing parties have different attributes on the same subjects. Block coordinate descent (BCD) is a promising and fast way to perform federated learning on vertically partitioned datasets. One of its strenghts is that it avoids computationally expensive cryptographic operations to secure the computations.

Data analysis consists of the computation on a data set. The results of this always leak some information about the underlying data set. Differential privacy was introduced by [4] to limit this leakage. Essentially, it adds uncertainty to the data analysis in such a way that similar data sets will likely lead to similar results. In this situation, although the training data remains at the owner, it is not protected sufficiently in all circumstances. The outcome of a joint cimputation may reveal whether a single subject is a member of the data set. This can in itself be sensitive information. To limit this possibility of information leakage we supplement BCD with differential privacy (DP). Since its introduction, differential privacy has seen some application, but not yet widespread adoption. One of the reasons for this, is that the DP parameters quantifying the privacy guarantees often cannot be made as small as hoped while preserving utility. The result is a noisy learning algorithm with reduced performance that theoretically could reveal information about data in the data set with considerable certainty.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

The motivation for this project is twofold. The first is to extend BCD with privacy guarantees to make it applicable to more use cases. The second motivation is demonstrating the practicality of DP in realistic use cases. To do so, we make some optimistic choices in our setup. This means that less noise has to be added and better performance is obtained. This clearly reduces the amount of protection DP offers. However, we believe that in this way we provide more meaningful privacy guarantees to correspond better to the data analyst's practice.

Our contributions

We introduce DP-BCD, a slightly reformulated version of the block coordinate descent algorithm [5] that has been made differentially private (DP) using objective perturbation. To makes these implementations as practical as possible we use local sensitivity parameters in a particularly small universe of possible data sets, instead of using global upper bounds on some large set of unseen data sets. Furthermore, before the analysis the parties agree on a loss scaling, fixing the amount of performance they are willing to sacrifice for more privacy. As a consequence, performance guarantees can be derived. Finally, we compare our algorithm with the standard BCD algorithm without differential privacy on both test data and the forest fires data set [6].

2. Materials and Methods

It is assumed throughout the paper that the parties have arranged that they all have disjunct data on the same N observations ordered in the same way. Furthermore, we consider problems with more observations than parameters m. In this setting the methods [7] for linear regression provide good differentially private linear regression algorithms.

2.1. Linear regression

We consider simple linear regression, which is the problem of finding $\boldsymbol{\beta}^*$ such that

$$\mathcal{L}(\boldsymbol{\beta}^*) = \min_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}) \quad , \tag{1}$$

where the loss $\mathcal{L}$ on the the data set $\boldsymbol{X}$ with labels $\boldsymbol{y}$ is given by

$$\mathcal{L}(\boldsymbol{\beta}) := \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \quad . \tag{2}$$

The optimal solution to this problem is found by deriving with respect to $\boldsymbol{\beta}$ and determining its root, yielding

$$\boldsymbol{\beta}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \quad . \tag{3}$$

2.2. Differential privacy

We begin with the standard definition of differential privacy and a localized variant [8–10]. An algorithm is (ε, δ)-DP, if it finds similar results for similar data sets with large probability $1 - \delta$. The similarity of the results is described by the privacy budget ε. In practice, this means that an attacker, who sees a certain result from the algorithm cannot decide which data set was used to generate the result. This implies records in the data set remain hidden from the attacker.

Definition 1. Differential privacy
A randomized mechanism $\mathcal{A}$ provides (ε, δ)-differential privacy, if for all pairs of data sets $x_1, x_2 \in \mathcal{X}$ at distance $1 = d(x_1, x_2)$ and for any outcome y

$$\mathbb{P}[\mathcal{A}(x_1) = y] \leq e^{\varepsilon}\,\mathbb{P}[\mathcal{A}(x_2) = y] + \delta \quad .$$

This definition provides guarantees that are unconditional on the knowledge or capabilities of the attacker. Furthermore, the parameters ε and δ can be bounded from above by a variety of composition laws. This allows the data owner to keep track of the maximum

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

amount of data leakage a data set has suffered.

Definition 2. Locally sensitive differential privacy
A randomized mechanism $\mathcal{A}$ provides (ε, δ)-locally sensitive differential privacy in the data set $x_1 \in \mathcal{X}$, if for all data sets $x_2 \in \mathcal{X}$ at distance $1 = d(x_1, x_2)$ and for any outcome y

$$\mathbb{P}[\mathcal{A}(x_1) = y] \leq e^{\varepsilon}\, \mathbb{P}[\mathcal{A}(x_2) = y] + \delta \quad .$$

2.2.1. Definition of a distance

Definition 1 makes it clear that some distance on the universe of data sets must be defined. Federated learning involves local data sets $\{\boldsymbol{X}^{(j)} | 1 \leq j \leq k\}$ that jointly form a federated data set $(\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(k)})$. It is preferable to use definitions that make sense both in the local and the federated context.

We use the intuitive definition here. Two data sets are at a distance 1, if the sets of subjects they have data on differ by one. It thus requires suppression of an entire row of the data set. This corresponds to having no information on someone and filling an entire row in $\boldsymbol{X}^{(i)}$ with zeros. This can be interpreted federatedly too. It means that all k parties remove their information about this subject from their local data. In this case, if all parties train ε-DP locally, this corresponds by simple composition, see Lemma 1, to $k\varepsilon$-DP in the federated setting.

2.3. Composition mechanisms

The learning algorithm described in Section 4.1 consumes $\delta = 0$ and a privacy budget of ε for every learning phase iteration. Using either simple composition [8] or advanced composition [11] it is possible to determine the consumed privacy budget for an entire protocol run.

Lemma 1. *{Simple composition}*
Let $\mathcal{M}_i$ be an $(\varepsilon_i, \delta_i)$-differentially private algorithm. The combined algorithm

$$\mathcal{A}(x) = (\mathcal{M}_1(x), \ldots, \mathcal{M}_T(x))$$

is $(\sum_{i=1}^{T} \varepsilon_i, \sum_{i=1}^{T} \delta_i)$-differentially private.

Lemma 2. *{Advanced composition}*
For every $\varepsilon > 0$, $\delta \geq 0$, $\delta' > 0$ and $T \in \mathbb{N}$ the class of (ε, δ)-differentially private mechanisms is $(\varepsilon', T\delta + \delta')$-differentially private under T-fold adaptive composition, for

$$\varepsilon' = \varepsilon\sqrt{2T\log(1/\delta')} + T\varepsilon(e^{\varepsilon} - 1) \quad .$$

This means that the advanced composition yields better results, if

$$\sqrt{2T\log(1/\delta')} < T(2 - e^{\varepsilon}) \quad ,$$

which leads to

$$T\log(N) < T\log(1/\delta') < \frac{T^2}{2}(2 - e^{\varepsilon})^2 \quad , \tag{4}$$

since $\delta' < 1/N$. This means that advanced composition is only beneficial, if $\varepsilon < \log(2)$ and a protocol with many iterations and a small privacy budget per iteration is used. For a $T = 5$ iterations and $\varepsilon = 0.2$, the data set may consist of at most 4 data points for advanced composition to be the better choice. And since BCD does not function with a tiny privacy budget per round, this means that we will only use simple composition.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

2.4. Code

The code used in this project is available at *https://github.com/JDJ847879/dp-bcd*.

3. Block coordinate descent

3.1. Incremental block coordinate descent

The starting point is the block coordinate descent (BCD) algorithm introduced by [5]. It can be used to train generalized linear model in a federated setting. A simple modification of the original algorithm communicates the missing parts rather than the own predictions. This has the advantages that only a single party needs to know the true label. This is a common situation in joint learning problems. For this reason the *single label owner* variant Algorithm 1 of BCD will be used here.

Algorithm 1 Incremental 2-party block coordinate descent algorithm. The subscript a is for Alice and b for Bob.

Alice and Bob initiate $\boldsymbol{\beta}_a^{(0)} \leftarrow \mathbf{0}$ and $\boldsymbol{\beta}_b^{(0)} \leftarrow \mathbf{0}$, respectively
Alice initiates $\boldsymbol{v}_b \leftarrow \boldsymbol{y}$
$i \leftarrow 0$
while stopping criterion is not met **do**
    player Alice **do**
        $\tilde{\boldsymbol{\beta}}_a \leftarrow (\boldsymbol{X}_a^T \boldsymbol{X}_a)^{-1} \boldsymbol{X}_a^T \boldsymbol{v}_b$
        $\boldsymbol{\beta}_a \leftarrow \boldsymbol{\beta}_a + \tilde{\boldsymbol{\beta}}_a$
        $\boldsymbol{v}_a \leftarrow \boldsymbol{v}_b - \boldsymbol{X}_a \tilde{\boldsymbol{\beta}}_a$
        send $\boldsymbol{v}_a$ to Bob
    end player
    player Bob **do**
        $\tilde{\boldsymbol{\beta}}_b \leftarrow (\boldsymbol{X}_b^T \boldsymbol{X}_b)^{-1} \boldsymbol{X}_b^T \boldsymbol{v}_a$
        $\boldsymbol{\beta}_b \leftarrow \boldsymbol{\beta}_b + \tilde{\boldsymbol{\beta}}_b$
        $\boldsymbol{v}_b \leftarrow \boldsymbol{v}_a - \boldsymbol{X}_b \tilde{\boldsymbol{\beta}}_b$
        send $\boldsymbol{v}_b$ to Alice
    end player
    $i \leftarrow i + 1$
end while

3.2. Convergence

Since Algorithm 1 is iterative, an end point must be chosen. Typically, one would let the algorithm run until the result has converged, where the standard definition of convergence requires any single player to find a remainder $\boldsymbol{v}_{(t)}$ in iteration t that is sufficiently close to a remainder seen before,

$$\|\boldsymbol{v}_{(t)} - \boldsymbol{v}_{(s)}\| \leq \mathcal{B}_C \quad \text{, for } 1 \leq s < t \qquad . \tag{5}$$

This method demands the weights to converge. However, the optimal weights may depend heavily on a single data point. It is precisely this dependence that DP tries to cap. When adding noise in each round, the weights will absorb some of this noise , which could cause an increasing series of remainders, so that convergence might never occur. For these reasons (5) is not an ideal convergence definition.

At each iteration the loss $\mathcal{L}(\boldsymbol{\beta})$ is minimized. At iteration t a remainder $\boldsymbol{v}_{(t)} = \boldsymbol{v}_{(t-1)} - \boldsymbol{X}\boldsymbol{\beta}_{(t)}$ with minimum length is passed on to the next player. However, after a certain number of iterations the benefit of an additional round will become very small. One may define that convergence is reached when the length of the remainder

$$\|\boldsymbol{v}_{(t)}\|_2^2 \geq \|\boldsymbol{v}_{(t-1)}\|_2^2 - \mathcal{B}_C \tag{6}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

hardly decreases or even increases. The bound for this would be defined at the initialization of the training. This definition is not very sophisticated, but it has the added advantage that it is directly related to the loss function, which is the objective of the training algorithm. Furthermore, it is applicable in virtually all situations. For example, it will also work in the case of increasing remainders, which may occur in a differentially private algorithm.

Rather than using convergence as stopping criterion the experiments described here use a fixed number of $T = 5$ iterations. This makes the analysis of the algorithm and its performance simpler.

Data reconstruction

Block coordinate descent is an efficient federated learning algorithm, but can leak information about the used data set. In [5] it is explained that the attackers may reconstruct the used data set up to a rotation. From discussions with the authors of [5] we have learned that the data is better protected than by a rotation. The original data can be approximated within a quantifiable margin of error, depending on the amount of shared intermediate results. Earlier reconstruction attacks suggest that an external attacker with supplementary information might be able to mimic this approach even without access to the intermediate results. Although the design, feasability and success of such an attack are merely hypothetical, the fact is at this point we cannot say to what extend the approach in [5] protects the processed data. This is one of the reasons to study a differentially private version of BCD.

4. Results

4.1. Objective perturbation

Information from the data set may leak in an analysis, because the attacker knows what computation was performed. This allows him to exclude certain data points from the data set, include other specific points or deduce relations that the data set fulfills. In objective perturbation [12,13] it is the loss function that is perturbed preventing the attacker from knowing what computation was performed.

The algorithm presented here consists of two phases. In the first phase all parties train a linear model on their local data set. The labels they use for this are the parts missing from the joint prediction. In the second phase the linear models are put together to form a linear model in the federated setting. This linear model can then be published. There are two potential groups of attacker possible in this setting. During the first phase it is the group of all other participants. At publication it is the outside world that receives the jointly trained model. This means that we must shield our data from both.

In this study we use local diferential privacy (LSDP), as defined in Definition 2. This means that only deviations at distance 1 of a party's own data set are considered. This means that the composition laws are not uniform over the entire universe $\mathcal{X}$ of data sets. Besides that, we use a small universe of possible data sets $\mathcal{X}$. It consists only of the actual data set and all data sets obtained by removing one record. We do not include possible data sets with one record more than our data set.

One may argue that using LSDP instead of DP to reduce the amount of noise needed, while lowering the privacy budget and increasing the noise cancel each other. This is not the case. In the transition the privacy guarantee is shifted from absent data points with a high privacy budget to the actual data with a low privacy budget. The privacy budget is the explicit security guarantee that (L)DP offers and as such is what potential users look at.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

The ambition is to minimize the following k-party loss function in both an iterative and a federated manner

$$\mathcal{L} = \left(\boldsymbol{y} - \sum_{i=1}^{k} \boldsymbol{X}^{(i)}\boldsymbol{\beta}^{(i)}\right)^2 + \sum_{i=1}^{k} \left(\boldsymbol{\beta}^{(i)}\right)^T \left(\boldsymbol{X}^{(i)}\right)^T \boldsymbol{b}^{(i)} \quad . \tag{7}$$

This is the k-party form of (2). A ridge regression term is omitted to perform a cleaner comparison to the original BCD algorithm. However, nothing prevents such a term. Furthermore, each party's loss function is perturbed by the dot product of the prediction and a secret vector $\boldsymbol{b}^{(i)}$ known only by party i.

If the vectors $\boldsymbol{b}^{(i)}$ would be sampled from a normal distribution such as (8), the perturbation term would have the added benefit that the local and federated perturbation term are of the same form. This would provide a similar perturbation term in the federated and local objective function. As explained in Remark 1, a different distribution is used.

Remark 1. *The vector $\boldsymbol{b}$ could be sampled from a normal distribution with density*

$$p_{naive}(\boldsymbol{b}) = \left(\frac{\varepsilon}{2\pi\xi^2}\right)^{N/2} \exp[-\frac{\varepsilon\|\boldsymbol{b}\|_2^2}{2\xi^2}] \quad . \tag{8}$$

It is clear that the direction of the vector is uniformally sampled from the surface of the N-dimensional sphere. For its length we want to solve for R

$$\frac{2}{\Gamma(N/2)} \int_0^{R/(\sqrt{2}\sigma)} \mathrm{d}r\, r^{N-1} e^{-r^2} = p \qquad , with\ p \in (0,1) \quad ,$$

which transforms into

$$\frac{1}{\Gamma(N/2)} \int_0^{R^2/(2\sigma^2)} \mathrm{d}t\, t^{(N/2)-1} e^{-t} = p$$

and is solved by the inverse lower incomplete gamma function. However, this is problematic. The high dimension pushes the the vector outwards, so that the noise vectors tend to get bigger with increasing number of observations. This leads to noise vectors overwhelming the data and a remainder that is larger than the input label.

As explained before, only the first party needs to know the labels. Afterwards, during iteration t party j obtaines

$$\boldsymbol{v}_t^{(j)} = y - \sum_{i=1}^{k} \boldsymbol{X}^{(i)}\boldsymbol{\beta}_t^{(i)} \quad , \text{ where } \quad \boldsymbol{\beta}_t^{(i)} = \begin{cases} \sum_{s=1}^{t} \boldsymbol{\beta}_{(s)}^{(i)} & , \text{for } i < j \\ \sum_{s=1}^{t-1} \boldsymbol{\beta}_{(s)}^{(i)} & , \text{for } i \geq j \end{cases}$$

of the label that is not yet explained by party $j-1$. From now on we will suppress the sub- and superscripts when possible. The local solution is given by

$$0 = \boldsymbol{X}^T(\boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^* - \boldsymbol{b}) \qquad \Rightarrow \qquad \boldsymbol{\beta}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\boldsymbol{v} - \boldsymbol{b}) \quad . \tag{9}$$

For each party we write that $\boldsymbol{X} \in M_{N\times m}$, so there are N observations of m attributes in this party's data. It follows from our data assumption that $N > m$.

There are two algorithms in use in the protocol. The first is used during the learning phase to communicate the missing part of the labels. It is given by

$$\mathcal{A}_l(\boldsymbol{X}) = \boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^* \quad . \tag{10}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

The second is used in the revealing phase and is defined by

$$\mathcal{A}_r(\boldsymbol{X}) = \sum_{t=1}^{T} \boldsymbol{\beta}^*_{(t)} \quad , \tag{11}$$

where $\boldsymbol{\beta}^*$ is in both cases defined in (9). In the special case of unperturbed learning, i.e. $\boldsymbol{b} = 0$ we call this solution $\boldsymbol{\beta}^\star$.

We start with the privacy of the learning algorithm $\mathcal{A}_l$. We sample $\boldsymbol{b} = l \cdot \boldsymbol{s}$ with $\boldsymbol{s} \in S^{N-1}$ uniformly and l with density $2\sqrt{\frac{\varepsilon}{2\pi\xi^2}} \exp[-\frac{\varepsilon l^2}{2\xi^2}]$, so that

$$p_{\xi,\varepsilon}(\boldsymbol{b}) = \frac{\Gamma(\frac{N}{2})}{\pi^{\frac{N}{2}}} \sqrt{\frac{\varepsilon}{2\pi\xi^2}} \exp[-\frac{\varepsilon \|\boldsymbol{b}\|_2^2}{2\xi^2}] \tag{12}$$

So the length of the perturbation vector is normally distributed and its direction is uniformally distributed. This ensures that the length of the perturbation vector is independent of the number of observations. The parameter ξ is the largest allowed value of $\|\boldsymbol{v}_{out}\|$ for a succesful protocol run.

The standard deviation of the length of the perturbation vector is given by $\xi/\sqrt{\varepsilon}$, where ε is the privacy budget for the round and

$$\xi = \gamma \|\boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^\star\|_2 \quad .$$

The parameter $\gamma > 1$ gives the maximally allowable deterioration in performance compared to the unperturbed case.

The probability that two databases $\boldsymbol{X}_1, \boldsymbol{X}_2$ of full rank at a distance 1 of each other yield the same output vector $\boldsymbol{v}_{out} = \boldsymbol{v}_{in} - \boldsymbol{X}_1\boldsymbol{\beta}^*_1 = \boldsymbol{v}_{in} - \boldsymbol{X}_2\boldsymbol{\beta}^*_2$ is, according to (9), given by

$$\begin{aligned}
\frac{\mathbb{P}[\mathcal{A}_l(\boldsymbol{X}_1) = \boldsymbol{v}_{out}]}{\mathbb{P}[\mathcal{A}_l(\boldsymbol{X}_2) = \boldsymbol{v}_{out}]} &= \frac{\mathbb{P}[0 = \boldsymbol{X}_1^T(\boldsymbol{v}_{out} - \boldsymbol{b}_1)]}{\mathbb{P}[0 = \boldsymbol{X}_2^T(\boldsymbol{v}_{out} - \boldsymbol{b}_2)]} = \frac{\mathbb{P}[\boldsymbol{b}_1 \in \ker(\boldsymbol{X}_1^T) + \boldsymbol{v}_{out}]}{\mathbb{P}[\boldsymbol{b}_2 \in \ker(\boldsymbol{X}_2^T) + \boldsymbol{v}_{out}]} \\
&= \frac{\mathbb{P}[\boldsymbol{b}_1 \in \ker(\boldsymbol{X}_1^T) + \boldsymbol{v}_{1,\perp_1}]}{\mathbb{P}[\boldsymbol{b}_2 \in \ker(\boldsymbol{X}_2^T) + \boldsymbol{v}_{2,\perp_2}]} \\
&= \frac{\exp[-\frac{\varepsilon}{2\xi^2}\|\boldsymbol{v}_1\|_2^2]}{\exp[-\frac{\varepsilon}{2\xi^2}\|\boldsymbol{v}_2\|_2^2]} \leq e^{\varepsilon}
\end{aligned} \tag{13}$$

Here we have decomposed $\boldsymbol{v}_{out} = \boldsymbol{v}_{1,\ker_1} + \boldsymbol{v}_{1,\perp_1} = \boldsymbol{v}_{2,\ker_2} + \boldsymbol{v}_{2,\perp_2}$ into parts inside the kernel and perpendicular to it. Note that the decomposition for $\boldsymbol{X}_1$ is different from that for $\boldsymbol{X}_2$. For the probabilities it suffices that

$$\exp[-\alpha(\boldsymbol{v}_\perp + \sum_j \lambda_j \boldsymbol{w}_j)^2] = \exp[-\alpha \boldsymbol{v}_\perp^2] \cdot \prod_j \exp[-\alpha\lambda_j^2] \quad ,$$

where $\{w_j\}$ is an orthonormal basis for the kernel. Note that the parts inside the kernel can only stem from $\boldsymbol{v}_{in}$. Since both matrices are of full rank, their kernels have the same dimensions and selecting a vector out of them is equally likely. For the perpendicular parts a standard argument can be used. The final inequality follows from

$$\|\boldsymbol{v}_{i,\perp}\|_2^2 \leq \|\boldsymbol{v}_{out}\|_2^2 = \|\boldsymbol{v}_{in} - \boldsymbol{X}_i\boldsymbol{\beta}^*_i\|_2^2 \leq \xi^2 \quad .$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

For the revealing phase a very similar argument works. Instead of the missing labels it are now the weights that are communicated. The privacy loss for revealing a single $\boldsymbol{\beta}^*_{(t)}$ is computed by

$$\begin{aligned}\frac{\mathbb{P}[\mathcal{A}_r(\boldsymbol{X}_1) = \boldsymbol{\beta}^*_{(t)}]}{\mathbb{P}[\mathcal{A}_r(\boldsymbol{X}_2) = \boldsymbol{\beta}^*_{(t)}]} &= \frac{\mathbb{P}[0 = \boldsymbol{X}_1^T(\boldsymbol{v}_{in,(t)} - \boldsymbol{X}_1\boldsymbol{\beta}^*_{(t)} - \boldsymbol{b}_{1,(t)})]}{\mathbb{P}[0 = \boldsymbol{X}_2^T(\boldsymbol{v}_{in,(t)} - \boldsymbol{X}_2\boldsymbol{\beta}^*_{(t)} - \boldsymbol{b}_{2,(t)})]} \\ &= \frac{\mathbb{P}[\boldsymbol{b}_{1,(t)} \in \ker(\boldsymbol{X}_1^T) + (\boldsymbol{v}_{in,(t)} - \boldsymbol{X}_1\boldsymbol{\beta}^*_{(t)})_{\perp_1}]}{\mathbb{P}[\boldsymbol{b}_{2,(t)} \in \ker(\boldsymbol{X}_2^T) + (\boldsymbol{v}_{in,(t)} - \boldsymbol{X}_2\boldsymbol{\beta}^*_{(t)})_{\perp_2}]} \\ &\leq \frac{\exp[-\frac{\varepsilon}{2\xi^2_{(t)}}\|(\boldsymbol{v}_{in,(t)} - \boldsymbol{X}_1\boldsymbol{\beta}^*_{(t)})_{\perp_1}\|_2^2]}{\exp[-\frac{\varepsilon}{2\xi^2_{(t)}}\|(\boldsymbol{v}_{in,(t)} - \boldsymbol{X}_2\boldsymbol{\beta}^*_{(t)})_{\perp_2}\|_2^2]} \leq e^{\varepsilon} \quad . \end{aligned} \tag{14}$$

This shows that revealing the weights $\sum_{t=1}^T \boldsymbol{\beta}^*_{(t)}$ consumes at most a privacy budget of $T\varepsilon$. For the final inequality we demand that for every iteration t

$$\|(\boldsymbol{v}_{in,(t)} - \boldsymbol{X}_i\boldsymbol{\beta}^*_{(t)})_{\perp_i}\|_2^2 \leq \|\boldsymbol{v}_{in,(t)} - \boldsymbol{X}_i\boldsymbol{\beta}^*_{(t)}\|_2^2 \leq \tilde{\xi}^2_{(t)} \quad .$$

The variance parameter is given by

$$\xi = \gamma\|\boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^\star\|_2 \quad ,$$

where $\boldsymbol{\beta}^\star$ is the solution of the unperturbed loss function, i.e. $\boldsymbol{b} = 0$, so that $0 = \boldsymbol{X}^T(\boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^\star)$. This implies that in each iteration of the protocol the loss scaling parameter must satisfy

$$\gamma \geq \sup_{\substack{|\boldsymbol{X}-\tilde{\boldsymbol{X}}|=1 \\ m=\mathrm{rk}(\boldsymbol{X})=\mathrm{rk}(\tilde{\boldsymbol{X}})}} \left\{\frac{\|\boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^\star\|_2}{\|\boldsymbol{v} - \tilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}}^\star\|_2}, \frac{\|\boldsymbol{v} - \tilde{\boldsymbol{X}}\tilde{\boldsymbol{\beta}}^\star\|_2}{\|\boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^\star\|_2}\right\} \quad . \tag{15}$$

So, γ represents the cost per round of adding differential privacy to the learning algorithm. It is the muliplier of the loss with respect to the unperturbed case, where $\boldsymbol{b} = \boldsymbol{0}$.

To demand that that observations should generate a full rank matrix is a minor demand. If it were not the case, a certain attribute could be predicted perfectly by the other attributes. Hence, it could be removed from the database to generate a full rank matrix again. Furthermore, it is not necessary for the proof to work with full rank matrices. They should only be of equal rank.

The complete algorithm DP-BCD is shown in Algorithm 2.

4.1.1. Utility bound

This parameter $\gamma > 1$ may be used in another way. It is directly related to the utility loss and can be given a maximum value to bound the utility loss upfront. It remains to be checked by the participants during the protocol whether this goal, (15), is met at every iteration. If not, the protocol will be aborted by the participants, because a model with sufficient utility cannot be trained. At every single iteration the sum of squared errors, which is the unperturbed loss, is bounded by

$$\|\boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^*\|_2^2 \leq \gamma^2\|\boldsymbol{v} - \boldsymbol{X}\boldsymbol{\beta}^\star\|_2^2 \quad .$$

This means that in a protocol run with k parties and T iterations the sum of squared errors is at most a factor γ^{2kT} larger than in the unperturbed case. If we denote with f_* the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

Algorithm 2 Differentially private 2-party block coordinate descent algorithm.

$\varepsilon' > 0$, $T \in \mathbb{N}$ and $\gamma > 1$
Alice and Bob initiate $\boldsymbol{\beta}_a \leftarrow \mathbf{0}$ and $\boldsymbol{\beta}_b \leftarrow \mathbf{0}$, respectively.
Alice initiates $\boldsymbol{v}_b \leftarrow \boldsymbol{y}$
for $t \in \{1, \ldots, T\}$ **do**
    player Alice **do**
        $\xi_a = \gamma \|\boldsymbol{v}_b - \boldsymbol{X}_a \boldsymbol{\beta}_a^\star\|_2$
        $\boldsymbol{b}_a \sim p_{\xi_a, \varepsilon'/(2T)}$
        $\tilde{\boldsymbol{\beta}}_a \leftarrow (\boldsymbol{X}_a^T \boldsymbol{X}_a)^{-1} \boldsymbol{X}_a^T (\boldsymbol{v}_b - \boldsymbol{b}_a)$
        $\boldsymbol{\beta}_a \leftarrow \boldsymbol{\beta}_a + \tilde{\boldsymbol{\beta}}_a$
        $\boldsymbol{v}_a \leftarrow \boldsymbol{v}_b - \boldsymbol{X}_a \tilde{\boldsymbol{\beta}}_a$
        if $\|\boldsymbol{v}_a\|_2 \leq \xi_a$ **then**
            send $\boldsymbol{v}_a$ to Bob
        else
            abort
        end if
    end player
    player Bob **do**
        $\xi_b = \gamma \|\boldsymbol{v}_a - \boldsymbol{X}_b \boldsymbol{\beta}_b^\star\|_2$
        $\boldsymbol{b}_b \sim p_{\xi_b, \varepsilon'/(2T)}$
        $\tilde{\boldsymbol{\beta}}_b \leftarrow (\boldsymbol{X}_b^T \boldsymbol{X}_b)^{-1} \boldsymbol{X}_b^T (\boldsymbol{v}_a - \boldsymbol{b}_b)$
        $\boldsymbol{\beta}_b \leftarrow \boldsymbol{\beta}_b + \tilde{\boldsymbol{\beta}}_b$
        $\boldsymbol{v}_b \leftarrow \boldsymbol{v}_a - \boldsymbol{X}_b \tilde{\boldsymbol{\beta}}_b$
        if $\|\boldsymbol{v}_b\|_2 \leq \xi_b$ **then**
            send $\boldsymbol{v}_b$ to Alice
        else
            abort
        end if
    end player
end for
Alice sends $\boldsymbol{\beta}_a$ to Bob.
Bob sends $\boldsymbol{\beta}_b$ to Alice.

differentially private predictions and with $\boldsymbol{f}_\star$ those without DP, then we see that the utility measure

$$R^2 = 1 - \frac{\|\boldsymbol{y} - \boldsymbol{f}_*\|_2^2}{Var_{\boldsymbol{y}}} \geq 1 - \gamma^{2kt} \frac{\|\boldsymbol{y} - \boldsymbol{f}_\star\|_2^2}{Var_{\boldsymbol{y}}} = 1 - \gamma^{2kt}(1 - R_\star^2) \quad . \tag{16}$$

This shows that we obtain a utility guarantee along with the privacy guarantee. The additional utility loss is bounded by parameters that can be set before the start of the protocol.

This proves the following theorem.

Theorem 1. *The linear regression of $\boldsymbol{y}$, held by Alice, against the data $(\boldsymbol{X}_a, \boldsymbol{X}_b)$ can be approximated by Algorithm 2, provided that $\mathrm{rk}(\boldsymbol{X}_a) = m_a$ and $\mathrm{rk}(\boldsymbol{X}_b) = m_b$ are of full column rank and contain N data points, where $N > m_a$ and $N > m_b$. For $T \in \mathbb{N}$, $\varepsilon' > 0$ and $\gamma > 1$ it is an ε'-differentially private algorithm. Furthermore, the utility is bounded from below by*

$$R^2 \geq 1 - \gamma^{2kt}(1 - R_\star^2) \quad ,$$

where $R_\star^2$ is the utility of the block coordinate algorithm without differential privacy (Algorithm 1 with $\lambda = 0$).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

4.2. Experiments

In order to quantify the performance of DP-BCD, a simulation was performed with synthetic data. We used standard normally distributed data and normally distributed β parameters (μ=2, σ=1.5). In the *baseline scenario* there are 9 predictors, with a correlation of 0.3, N=1000, R^2=0.3, ε=1, and γ=1.2 with 2 parties. Because preliminary analyses indicated that five iterations was a favourable cut-off in the trade-off between privacy and noise-accumulation we used this number of iterations.

With this baseline scenario, each of the following factors were varied separately: the sample size N (100, 250, 1000, 5,000, 10,000), the correlation between predictors (0.1, 0.3, 0.5), R^2 (0.1, 0.3, 0.8), ε (0.1, 0.3, 0.5, 0.8, 1.0, 1.5, 2.5, and 10), and γ (1.15, 1.25, 1.5, 1.8, 2, 2.5, 3). The γ values were chosen such that the desired privacy level could be achieved.[1] Each of the variations was repeated 500 times, with the exception of the sample size experiment which had 100 repetitions per variation. At every iteration, a different data set ($\mathbf{X}$ and y) was generated. In experiments where the privacy parameters ε and γ were varied, different β parameters were generated at each iteration as well.

We evaluated utility by considering two outcomes: R^2 and the β estimates. These outcomes were also generated with the centralized setting and when using BCD without differential privacy. Because the results for these two algorithms were practically identical, we only compare it to the centralized results. For several scenarios, we computed the average absolute proportional distance (AAPD) with these β estimates. For r repetitions of a scenario with $||\boldsymbol{\beta}||$ predictors, the corresponding AAPD is defined as:

$$AAPD = \frac{\sum_{i=1}^{r} |\boldsymbol{\beta}^* - \boldsymbol{\beta}^\star|_1}{r \cdot ||\boldsymbol{\beta}||}.$$

4.2.1. Results

Impact of privacy parameters

The impact of γ and ε on the β and R^2 estimates is non-linear. As expected, γ has a stronger impact on R^2 than ε. From Figure 1b it can be seen that although the bound for R^2 decreases significantly with γ, for these synthetic data, the *expected* decrease was not nearly as steep its bound. For example, for γ=3, the average R^2 is approximately -2.5 on average, whereas it is bounded by -2.44*10^9. Although the results are in line with Theorem 1, the bound can be almost meaningless for large values of γ, and perhaps a tighter bound or expected deviation can be derived.

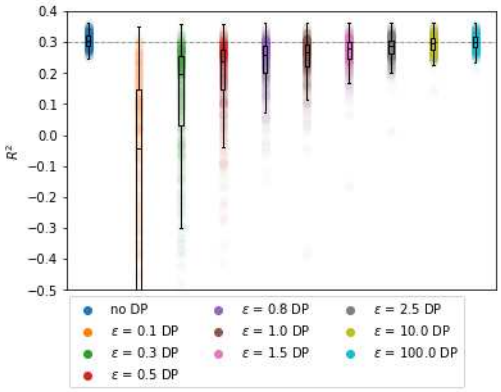
The β estimates grow closer to the BCD results as ε increases, which is in line with the expectation. Table 1 shows that for ε=1, the β estimates deviate 47% from the centralized β parameters on average. For γ=1.15 (the lowest tested value), the β estimates deviate 295% from the centralized setting, but note that this is for $\varepsilon = 1$, see Table 2. For higher values of ε the estimates are closer to the centralized β parameters, though still differing by up to 47%. As a reference, the average and median deviation after five iterations for BCD without DP is zero.

Impact of R^2

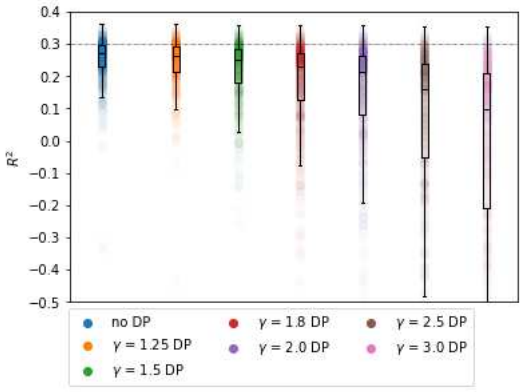
As R^2 increases in the data-generating model, more predictive power is preserved with DP-BCD as well, see Figure 2. The precision and bias with which the β parameters can be estimated is also significantly impacted by R^2 in the data generating model, see

[1] For low values of γ the algorithm may terminate (see Algorithm 1, because γ is too low. This could lead to an unbalanced comparison between scenarios where the γ is sufficiently high and those where the algorithm could not carry out all iterations for each repetition.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1



(a) R^2 after DP-BCD as a function of ε.



(b) R^2 after DP-BCD as a function of γ.

Figure 1. R^2 of DP-BCD in articifial test data.

ε	Mean	Median
0.2	1.50	10.09
1.0	0.67	4.37
2.0	0.47	3.08
3.0	0.38	2.51
5.0	0.30	1.94
10.0	0.21	1.37
20.0	0.15	0.97

Table 1. Mean and median proportional absolute error of β estimates compared to centralized setting, over ε after five iterations for $\gamma = 1.2$.

γ	Mean	Median
1.15	2.95	0.45
1.25	3.21	0.49
1.50	3.86	0.59
1.80	4.64	0.71
2.00	5.17	0.79
2.50	6.51	0.98
3.00	7.87	1.18

Table 2. Mean and median proportional absolute error of β estimates compared to centralized setting, over γ after five iterations for ε=1.

Table 3.

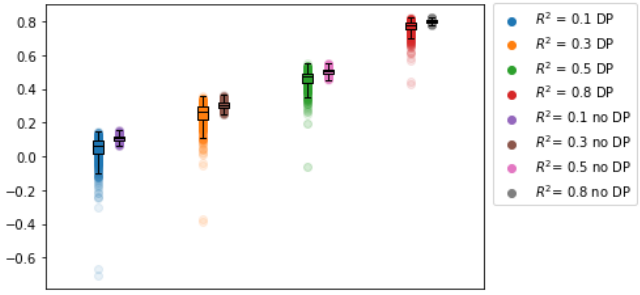


Figure 2. R^2 after DP-BCD as a function of R^2 in the data-generating model

R^2	Mean	Median
0.1	4.72	0.85
0.3	3.08	0.47
0.5	1.88	0.33
0.8	1.02	0.21

Figure 3. Mean and median proportional absolute error over R^2

Impact of correlation

As expected, the average β error increases with the correlation between predictors. This can be seen in the wider sampling distribution in Table 5. This is to be expected for an implementation of DP, since more noise must be added to hide the outliers in the data. For very high correlations, the average β parameters differ as well, which means that the estimates are biased. The R^2, however, remained unaffected by this parameter, though it is lower than with the BCD algorithm.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

As studied by [5], strongly correlated data require more iterations for accurate parameter estimation. In fact, for highly correlated data with over 25 variables, hundreds of iterations can be required for convergence of the weights. In a differential privacy setting, this may consume vast privacy budgets or yield poor results due to noise accumulation.

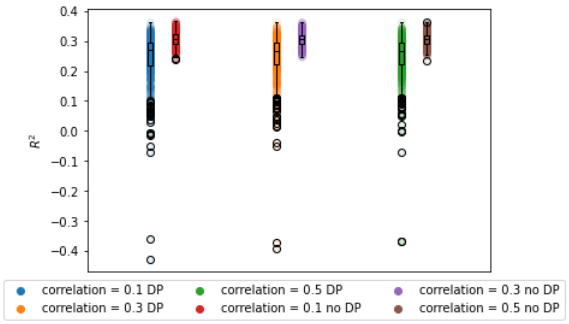


Figure 4. R^2 after DP-BCD as a function of the correlation between predictors in the data-generating model

correlation	Mean	Median
0.1	0.56	0.27
0.3	1.06	0.45
0.5	3.11	0.69

Figure 5. Mean and median proportional absolute error over the correlation between predictors.

Impact of sample size

As can be seen from Table 7, the β error steadily decreases with the sample size. Furthermore, the R^2 distribution grows closer to the centralized results.

4.3. Evaluation with forest fires data

We run experiments with a forest fire data set [6], which was used by [5]. In Figure 8, we create a plot similar to Figure 5 of [5] using the same data and parties. We plot the average coefficients and the 2.5^{th} and 97.5^{th} percentiles for $\gamma = 1.2$, $\varepsilon = 2$ and 20, for 5 iterations and repeated this 1000 times. We also plot the parameter estimates for the centralized analysis, which [5] showed to be almost identical to BCD with 450 iterations.

For a relatively small privacy budget of ε=2, the average coefficients are similar to those from the centralized setting. For ε=20, the distributions are narrower, which is in line with the synthetic data results. The closeness of the sampling distributions to the centralized setting is likely affected by the low correlations between the predictors. The absolute average and median are 0.08 and 0.05, respectively.

The R^2 values are quite low, due to the fact that the centralized R^2 is only .07. Because R^2 values for DP-BCD are generally lower than BCD, all median R^2 values are negative for the forest fires data. The y-axis in Figure 9 is cut off at -.5, because negative R^2 values

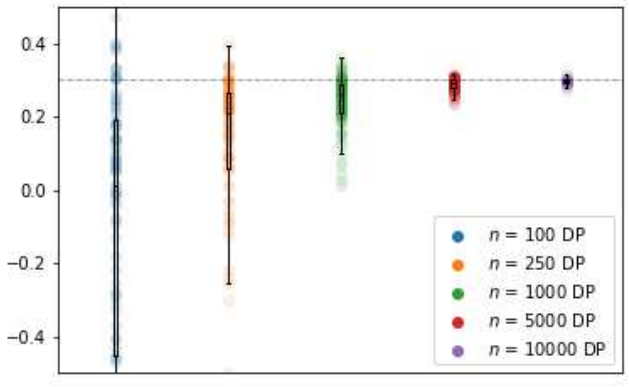


Figure 6. AAPD after DP-BCD as a function of N

N	Mean	Median
100	7.68	1.46
250	3.82	0.95
1,000	2.30	0.47
5,000	0.85	0.22
10,000	0.59	0.15

Figure 7. Mean and median proportional absolute error over N

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 September 2022 doi:10.20944/preprints202209.0176.v1

are not informative, but for $\varepsilon = 1.0$ and $\varepsilon = 2.0$ the median R^2 values are -4.07 and -0.94, respectively. Thus, for a centralized model that already has low predictive power, adding differential privacy generally results in a complete loss of predictive power.

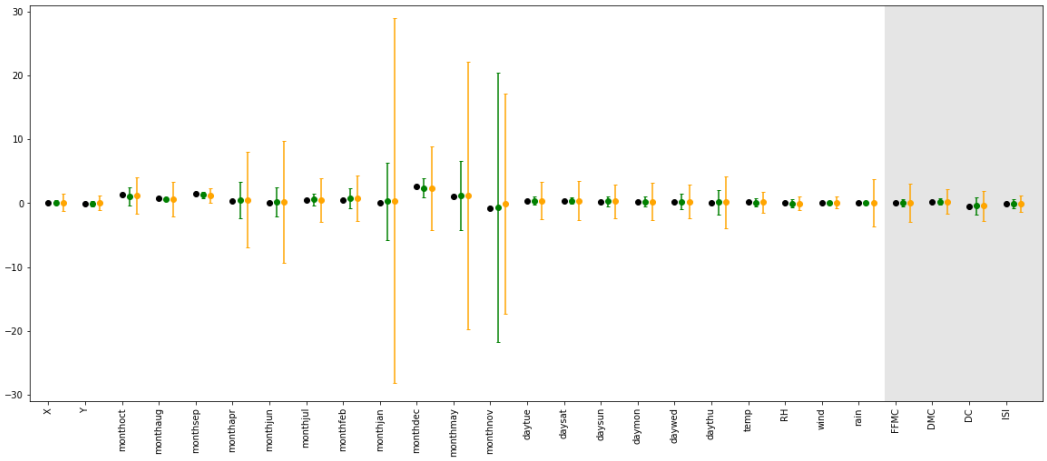


Figure 8. Centralized parameter estimates (black) for forest fire analysis, with average coefficients and 95% confidence intervals for ε =2 (orange) and ε=20 (green), for γ=1.2, 1000 repetitions. Parties are separated with background shading.

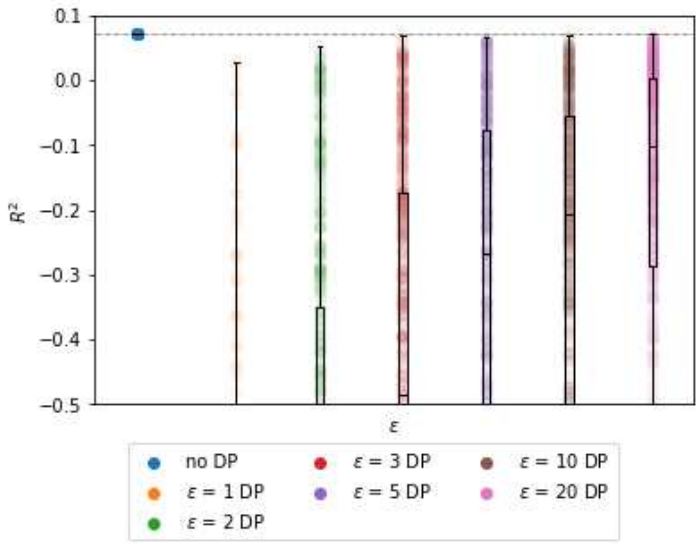


Figure 9. R^2 over ε for the forest fires data set (5 iterations, 100 repetitions, and $\gamma = 1.2$)

5. Discussion

In this work, we presented a differentially private extension of the BCD algorithm proposed by [5]. We show that in scenarios where privacy concerns or regulations limit collaborative opportunities, DP-BCD can be used to enable multi-party collaboration, with both privacy and utility bounds. If one party wants to predict their labels with (sensitive) predictors from another party's data base, we have shown that the proposed approach can be used to construct a predictive model.

The simulations show that the obtained weights obtained with DP-BCD are similar to BCD, also for correlated data. Nonetheless, the predictive power is lower, expecially for problems with low R^2. Nonetheless, the median R^2 values observed are similar to the BCD, albeit with larger deviations and some outliers. The predicitve power is considerably lower for conservative values of ε, high values of γ or small sample sizes, provided

 doi:10.20944/preprints202209.0176.v1

that γ is chosen big enough not to abort.

Both γ and ε have an exponential impact on the predictive power. Therefore, the γ level must be set as low as possible, as there is no benefit to having a high γ when ε is at the suitable level. With respect to ε, the algorithm retained predictive power even for high privacy values (e.g. ε=2.5). Though not incorporated in the simulation, the number of parties is also expected to impact R^2, as it makes the BCD procedure more challenging and has a significant impact on the utility bound.

Unbiased estimation of β parameters is a more challenging task than retaining predictive power. With the current procedure, it is not feasible with the amount of noise required. Particularly for highly correlated variables, the number of iterations may exceed the "turning point" where the increased precision as a result of iterations is overshadowed by accumulated noise. For large sample sizes and large values of ε it is possible to obtain β parameters similar to the BCD procedure. This was also visualized in the forest fire analysis, where $\varepsilon = 20$ led to parameter estimates closer to the centralized and BCD setting (though not identical).

5.1. Possible improvements of the algorithm

A fixed number of iterations has been used in the experiments. In this way it a clearer presentation of the performance of DP-BCD can be given. However, using a convergence criterion, as described in Paragraph 3.2, makes it possible to explicitly decide each round whether the improved utility is worth the consumed privacy budget. In this way, algorithms with better performance in terms of privacy budget and utility can be constructed.

Although a bound for R^2 is derived in the present work, this bound may be made much tighter. The lower bound in this form deteriorates quickly with an increasing number of parties and iterations. Furthermore, if users wish to conduct inference regarding the effect of specific variables on the outcome variable, standard errors and confidence intervals should be corrected for the added noise.

Also the analysis of the privacy budget in (14) is not very tight. Using more sophisticated methods it should be possible to given more accurate bounds on the privacy budget consumed with publication.

6. Conclusions

In this article we have presented an extension of the block coordinate descent algorithm with differential privacy with a single label owner. Our implementation is based on objective perturbation and local sensitivity. In this way, we are able to generate models with both comparable explanatory power as BCD and single digit privacy budgets. Furthermore, the setup allows for a theoretical utility bound that gives a lower bound for the R^2 of the differentially private version in terms of that of the original algorithm.

Experiments indicate that DP-BCD performs particularly well in settings where the data has a high R^2, meaning that the data contains a lot of explanatory power. Furthermore, the low number of iterations used benefits data sets with little correlation. For the forest fires data set we find that the obtained weights are very similar, although the R^2 is much lower, which is also due to the low R^2 of the solution in the centralized setting.

Author Contributions: Jins de Jong: conceptualization, methodology, formal analysis, writing—original draft preparation.
Bart Kamphorst: conceptualization, software, validation, writing—review and editing.
Shannon Kroes: investigation, writing—original draft preparation, visualization.
All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank Ásta Magnúsdóttir and Savvina Daniil for their contributions to the project.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAPD	Average absolute proportional distance
BCD	Block coordinate descent
DP	Differential privacy
DP-BCD	Differentially private block coordinate descent
LSDP	Locally sensitive differential privacy

References

1. Veugen, T. Secure Multi-party Computation and Its Applications. In Proceedings of the Innovations for Community Services. Springer International Publishing, 2022, pp. 3–5.
2. Veugen, T.; Kamphorst, B.; van de L'Isle, N.; van Egmond, M.B. Privacy-Preserving Coupling of Vertically-Partitioned Databases and Subsequent Training with Gradient Descent. In Proceedings of the Cyber Security Cryptography and Machine Learning. Springer International Publishing, 2021, pp. 38–51.
3. Sangers, A.; van Heesch, M.; Attema, T.; Veugen, T.; Wiggerman, M.; Veldsink, J.; Bloemen, O.; Worm, D. Secure Multiparty PageRank Algorithm for Collaborative Fraud Detection. In Proceedings of the Financial Cryptography and Data Security. Springer International Publishing, 2019, pp. 605–623.
4. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. *Journal of Privacy and Confidentiality* **2017**, *7*, 17–51.
5. van Kesteren, E.J.; Sun, C.; Oberski, D.L.; Dumontier, M.; Ippel, L. Privacy-Preserving Generalized Linear Models using Distributed Block Coordinate Descent, 2019, [arXiv:cs.LG/1911.03183].
6. Cortez, P.; Morais, A.d.J.R. A data mining approach to predict forest fires using meteorological data, 2007.
7. Kifer, D.; Smith, A.; Thakurta, A.; Mannor, S.; Srebro, N.; Williamson, R. Private Convex Empirical Risk Minimization and High-dimensional Regression. *Proceedings of the 25th Annual Conference on Learning Theory (COLT '12)* **2012**, *Vol. 23*.
8. Dwork, C.; Roth, A. *The Algorithmic Foundations of Differential Privacy*; Foundations and trends in theoretical computer science, Now Publishers, 2014.
9. Farias, V.A.E.; Brito, F.T.; Flynn, C.; Machado, J.C.; Majumdar, S.; Srivastava, D. Local Dampening: Differential Privacy for Non-Numeric Queries via Local Sensitivity. *Proc. VLDB Endow.* **2020**, *14*, 521–533.
10. Nissim, K.; Raskhodnikova, S.; Smith, A. Smooth Sensitivity and Sampling in Private Data Analysis. In Proceedings of the Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing; Association for Computing Machinery: New York, NY, USA, 2007; STOC '07, p. 75–84.
11. Dwork, C.; Rothblum, G.N.; Vadhan, S. Boosting and Differential Privacy. In Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 51–60.
12. Chaudhuri, K.; Monteleoni, C. Privacy-preserving logistic regression. In Proceedings of the NIPS, 2008.
13. Neel, S.; Roth, A.; Vietri, G.; Wu, Z.S. Oracle Efficient Private Non-Convex Optimization. In Proceedings of the Proceedings of the 37th International Conference on Machine Learning. JMLR.org, 2020, ICML'20.