Bioinformatics, 35, 2019, i218–i224
doi: 10.1093/bioinformatics/btz373
ISMB/ECCB 2019

OXFORD

Representation transfer for differentially private drug sensitivity prediction

Teppo Niinimäki[1], Mikko A. Heikkilä[2], Antti Honkela[2,3,4,*] and Samuel Kaski[1,*]

[1]Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Espoo 00076, Finland, [2]Department of Mathematics and Statistics, [3]Department of Public Health and [4]Department of Computer Science, Helsinki Institute for Information Technology HIIT, University of Helsinki, Helsinki 00014, Finland

*To whom correspondence should be addressed.

Abstract

Motivation: Human genomic datasets often contain sensitive information that limits use and sharing of the data. In particular, simple anonymization strategies fail to provide sufficient level of protection for genomic data, because the data are inherently identifiable. Differentially private machine learning can help by guaranteeing that the published results do not leak too much information about any individual data point. Recent research has reached promising results on differentially private drug sensitivity prediction using gene expression data. Differentially private learning with genomic data is challenging because it is more difficult to guarantee privacy in high dimensions. Dimensionality reduction can help, but if the dimension reduction mapping is learned from the data, then it needs to be differentially private too, which can carry a significant privacy cost. Furthermore, the selection of any hyperparameters (such as the target dimensionality) needs to also avoid leaking private information.

Results: We study an approach that uses a large public dataset of similar type to learn a compact representation for differentially private learning. We compare three representation learning methods: variational autoencoders, principal component analysis and random projection. We solve two machine learning tasks on gene expression of cancer cell lines: cancer type classification, and drug sensitivity prediction. The experiments demonstrate significant benefit from all representation learning methods with variational autoencoders providing the most accurate predictions most often. Our results significantly improve over previous state-of-the-art in accuracy of differentially private drug sensitivity prediction.

Availability and implementation: Code used in the experiments is available at https://github.com/DPBayes/dp-representation-transfer.

Contact: antti.honkela@helsinki.fi or samuel.kaski@aalto.fi

1 Introduction

Privacy-preserving machine learning has the potential to enable the research use of many sensitive datasets that would otherwise be out of reach for the community. This is especially the case for medical data, which almost always contain sensitive information traceable back to the data subjects. As an example, it has been shown that individuals can be identified from genomic data (Gymrek *et al.*, 2013) including mixtures from several individuals (Homer *et al.*, 2008). It is likely that functional genomics data such as gene expression data are also identifiable. Although different anonymization strategies (Li *et al.*, 2007; Machanavajjhala *et al.*, 2007; Sweeney, 2002) can protect the privacy of the data subjects to some degree, they do not have formal guarantees and can fail to provide sufficient protection in practice (Ganta *et al.*, 2008).

Differential privacy (DP) (Dwork *et al.*, 2006; Dwork and Roth, 2014) is a framework that guarantees strict bounds for the amount of leaked private information, even in the presence of arbitrary side information. The guarantees are obtained by adding specific forms of randomization to the computation process. In a machine learning context this usually means adding noise either directly to the input of the algorithm (*input perturbation*), to the output (*output perturbation*) or modifying the algorithm itself, for instance, by perturbing the optimization objective (*objective perturbation*).

The privacy guarantee is controlled by a 'privacy budget' parameter, usually denoted by $\epsilon > 0$; smaller ϵ means stricter guarantees,

and can be achieved by increasing the amount of noise. Formally, a randomized mechanism $\mathcal{M}$ is said to be *ϵ-differentially private*, if for all pairs of *neighboring* datasets X, X' differing (There are two slightly different definitions of neighboring datasets. In *bounded* case, the value of one sample is allowed to change. In *unbounded* case, the addition or removal of one sample is allowed. Unbounded ϵ-DP guarantee implies bounded 2ϵ-DP guarantee. This article uses the bounded case.) on a single sample and all measurable subsets S of possible outputs,

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

Intuitively, this means that changing one sample in the dataset can change the output distribution only by a factor e^{ϵ}.

As an extension, $\mathcal{M}$ is said to be *(ϵ, δ)-differentially private*, if

$$\Pr(\mathcal{M}(X) \in S) \leq e^{\epsilon}\Pr(\mathcal{M}(X') \in S) + \delta,$$

for all measurable S and all neighboring datasets X, X'. The condition with non-zero $\delta > 0$ is often easier to achieve than pure ϵ-DP.

In this article, we are interested in DP learning for drug sensitivity prediction using gene expression data. First proposed by Staunton *et al.* (2001), the drug sensitivity prediction problem has attracted significant attention recently, including from a DREAM challenge in 2012 (Costello *et al.*, 2014) that provided standardized evaluation metrics. The scale of the cytotoxicity assays needed has kept the sizes of the available datasets relatively small from a machine learning perspective. Honkela *et al.* (2018) were the first to apply DP learning to this problem. They needed to specifically limit the sensitivity of the learning and the dimensionality of the input data to make the learning feasible.

In abstract terms, our goal in this problem is DP learning of predictive models with high-dimensional input data, where both input and output variables need DP protection. This is a case where DP methods tend to run into trouble with moderate dataset sizes: the amount of noise that needs to be added usually increases quickly with the dimensionality, leading to output that is dominated by the noise. This warrants the use of dimensionality reducing methods with the aim of finding a good low-dimensional representation of the original data. However, unless one uses a random projection (RP) or some other 'dummy' method that does not depend on the data, finding a good representation can also leak private information. For this reason, the dimension reduction method itself would also need to be made differentially private, which can completely invalidate the noise magnitude savings obtained in any downstream task like prediction.

Different solutions have been proposed for various special cases: Kifer *et al.* (2012) solve sparse linear regression problems by using an ϵ-DP feature selection algorithm. Honkela *et al.* (2018) utilize external knowledge to select a relevant subset of features. Kasiviswanathan and Jin (2016) show theoretical results on using RPs to improve DP learning on high-dimensional problems. Differentially private versions of methods such as principal component analysis (PCA) (Chaudhuri *et al.*, 2012; Dwork *et al.*, 2014) or deep learning (Abadi *et al.*, 2016; Acs *et al.*, 2019) exist and could be used to learn a representation, but the noise cost can be impractically large for small but high-dimensional datasets.

We study a straightforward solution based on feature representation transfer, similar to self-taught learning of Raina *et al.* (2007). By using an additional non-sensitive dataset to learn the representation, we can apply more advanced representation learning methods. This approach has many advantages: we do not need labels for the additional dataset, although in our case we make use of labels for a different task; and only the main learning algorithm needs to be differentially private, while the representation can be learned using any non-DP method. Additionally, the public data can also be used for optimizing any hyperparameters for the representation learning. In this article, we consider PCA and variational autoencoders (VAEs).

Differentially private transfer learning was recently considered by Wang *et al.* (2019) in a hypothesis transfer setting, where models trained on several related source domains are used to improve learning in the desired target domain. This approach is only applicable to a case where we have labeled data from multiple related learning problems, which is not the case for drug sensitivity prediction.

Another related approach was considered by Papernot *et al.* (2017), who propose differentially private semi-supervised knowledge transfer that uses an ensemble of 'teacher' models trained on private data to label unlabeled public data, which is then used to train a 'student' model that will be released. The method is flexible in a sense that it can use any 'black-box' model as teachers and student. However, it is limited to classification tasks. Furthermore, it seems to require a large enough private dataset to train a sizeable ensemble of private teacher models in addition to a small public dataset. Papernot *et al.* (2017) note that a large ensemble is needed to compensate for the noise injected to ensure privacy. Their reported results use $n = 250$ teacher models and it seems unlikely that a significantly smaller number would lead to good results. Training so many independent models using the data available in the tasks we are interested in is clearly impossible.

Assuming there is labeled public data available, the importance weighting approach of Ji and Elkan (2013) can be used for efficient differentially private data publishing. Ji and Elkan (2013) report that the method can reach accurate results already with a small privacy budget, but their example has a much lower dimensionality than any genomic dataset and it is unclear how the method would scale to genomic data.

Yet another strategy is to learn a differentially private unsupervised generative model for the data (including the target variable for the prediction task of interest), use it to generate a synthetic version of the data, and use a non-DP algorithm for the actual learning task of interest. Several methods have been proposed for differentially private data sharing (Acs *et al.*, 2019; Xie *et al.*, 2018; Zhang *et al.*, 2017) that could be used for generative model learning and data generation. For the problem we are considering, however, this approach is problematic as it requires solving a more general and difficult learning task, good solution of which would typically require orders of magnitude more private data than a direct solution of the original prediction task.

The data needs of the alternative approaches to transfer learning in a DP context are summarized in Table 1.

The rest of this article is organized as follows: In Section 2, we formalize the problem setting and give an overview of our proposed approach. Section 3 gives more details on the implementation of different parts of the proposed approach. And finally, in Section 4, we conduct experiments with the approach on two different prediction tasks on genomic data.

2 Approach

We assume a setting where we have a private dataset containing a high-dimensional $n \times d$ feature matrix X_{priv} and an $n \times 1$ target vector Y_{priv}, where n is the number of samples and d is the number of features. The goal is to learn a differentially private predictor from X_{priv} to Y_{priv}. As learning to predict Y_{priv} from high-dimensional

Table 1. Overview of DP transfer learning approaches and their data needs

Approach	Public data	Private data
This article	A lot, unlabeled	Limited, labeled
Ji and Elkan (2013)	Moderate, labeled	Limited, labeled
Papernot *et al.* (2017)	Moderate, unlabeled	A lot, labeled
Wang *et al.* (2019)	Optional	Moderate, labeled

X_{priv} directly is typically not feasible, with moderate sample size and a reasonable privacy budget, we opt for using public data to learn a low-dimensional representation for X_{priv}. Therefore, we also assume a publicly available dataset of an $m \times d$ feature matrix X_{pub} and an $m \times 1$ auxiliary target vector Y'_{pub} for a related auxiliary prediction task. Although a representation can be learned with X_{pub} only, the availability of Y'_{pub} is useful for selecting the size of the representation and any other hyperparameters.

We make the following informal assumptions about the relation of the public and the private data: (i) X_{priv} and X_{pub} contain the same set of features and are either draws from the same distribution or otherwise distributed similarly enough that using the same mapping to compute a representation is reasonable. (ii) Y'_{pub} may or may not be of the same type as Y_{priv}, but the prediction tasks should resemble each other enough that the prediction of Y'_{pub} can be used for optimizing the hyperparameters for the main task of predicting Y_{priv}.

We propose the following procedure:

1. Use the public data to learn a dimension-reducing representation mapping $f : \mathbb{R}^d \to \mathbb{R}^r$, where $r \ll d$, such that $f^{-1}(f(X_{\text{pub}})) \approx X_{\text{pub}}$.
2. Obtain a low-dimensional representation Z_{priv} of the private feature data by applying f to X_{priv}.
3. Learn a differentially private predictor g such that $g(Z_{\text{priv}}) \approx Y_{\text{priv}}$.
4. Publish $g \circ f$.

An overview of the learning process is shown in Figure 1.

It is easy to see that the proposed process has the same DP-guarantees as the learning algorithm of Step 3:

THEOREM 1. If Step 3 is (ϵ,δ)-DP w.r.t. Z_{priv} and Y_{priv}, then the whole process is also (ϵ,δ)-DP w.r.t X_{priv} and Y_{priv}.

PROOF. As the learning of f does not use private data, it does not leak any private information. Since each row of Z_{priv} depends only on the corresponding row of X_{priv}, (ϵ,δ)-guarantees w.r.t. Z_{priv} translate directly to guarantees w.r.t. X_{priv}.

In the following section, we give some methods that can be used to implement the DP predictor g and the representation mapping f. In addition, we describe a procedure for tuning the hyperparameters of f.

3 Materials and methods

3.1 Differentially private prediction

Later in Section 4, we will consider prediction tasks that are either real-valued regression or binary classification tasks. Linear regression will be applied to the former and logistic regression to the latter. For now, denote by X the feature matrix and by y the prediction target vector (either real-valued or binary $\{-1, 1\}$)

Logistic regression can be made differentially private with objective perturbation. The usual non-DP version of the problem can be solved by minimizing the regularized negative log-likelihood $n^{-1}\sum_{i=1}^{n} \log(1 + e^{y_i w^T x_i}) + \lambda w^T w$ with respect to the weight vector

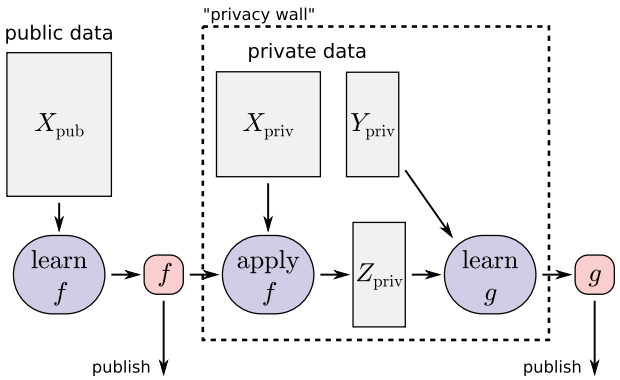

Fig. 1. The process of learning f and g. Since the learning of g is DP, the leakage of information outside of the 'privacy wall' is controlled

w, where x_i and y_i denote the ith sample in X and y, respectively and λ controls the strength of L_2 regularization. In a method presented by Chaudhuri and Monteleoni (2009), ϵ-DP privacy is obtained by adding a random bias term $b^T w/n$ (where b is a random vector drawn from a distribution with density proportional to $e^{-\epsilon\|b\|/2}$) to the optimization objective. The method requires that the samples in the input feature data are bounded into a 1-sphere.

Like DP logistic regression, also a DP linear regression algorithm can be obtained with an analogous objective perturbation method (Kifer *et al.*, 2012). However, since the underlying model belongs to the exponential family, there is also an alternative output-perturbation based ϵ-DP method that does not require iterative optimization: Compute the sufficient statistics ($X^T X$, $X^T y$ and $y^T y$) and add noise to them via the Laplace-mechanism (Foulds *et al.*, 2016). We use Bayesian linear regression with sufficient statistic perturbation and data clipping as described by Honkela *et al.* (2018).

3.2 Representation learning

RP (see e.g. Bingham and Mannila, 2001) projects the d-dimensional data to an r-dimensional subspace by multiplying it with a random $d \times r$ projection matrix. This transformation has been shown to preserve approximately the distances between data points (Johnson and Lindenstrauss, 1984), which is often a desired property for dimensionality reduction methods.

PCA finds an orthogonal linear transformation that converts the data to coordinates that are uncorrelated and whose variance decreases from first to last coordinate. When used for dimensionality reduction, only the first r coordinates are kept—these correspond to the r orthogonal directions in which the variance of the original data is the highest.

VAE (Kingma and Welling, 2014) learns a generative decoder model $p_\theta(x|z)$, where z is a latent representation of x, and an encoder model $q_\xi(z|x)$ that approximates the posterior distribution $p_\theta(z|x)$. Both p_θ and q_ξ are implemented as neural networks (typically MLPs) and optimized concurrently with variational inference.

We fix z to be low-dimensional, in which case the learned encoder q_ξ can be used for dimensionality reduction by setting $f(x) = \mathrm{E}_{z\sim q_\xi(\cdot|x)}[z]$. (As usual, define $q_\xi(\cdot|x)$ as a multivariate Gaussian distribution parametrized by mean $\mu_\xi(x)$ and diagonal covariance $\Sigma_\xi(x)$, in which case $f(x) = \mu_\xi(x)$.)

3.3 Optimization of hyperparameters

For selecting the dimension of the representation and any other hyperparameters of the representation-learning algorithm, we propose a combination of any parameter optimization approach (such

as Bayesian optimization, random search or grid search) and a cross-validation-like procedure for optimizing an auxiliary task of predicting Y'_{pub} from X_{pub}. As no private data are used, the parameter optimization phase does not consume any of the available privacy budget. In addition, if the auxiliary prediction task uses the same method as the main prediction task, then the hyperparameters could be optimized at the same time.

First the (public) data are divided into k disjoint subsets. Instead of using one of the subsets as 'validation' data and the rest as 'training' data as in cross-validation, we use one of the subsets to simulate the private data and the rest to simulate the public data. From now on, these are referred to as *pseudo-private* and *-public* sets. The proposed framework (from Section 2) is then applied to these, i.e. a representation mapping f is learned from the pseudo-public data, f is applied to the features of pseudo-private data, a predictor g is learned for the pseudo-private target variable and its accuracy is measured. As in k-fold cross-validation, this is repeated for all k possible selections of the pseudo-private subset. For measuring the accuracy of g, (actual) cross-validation can be used, i.e. the pseudo-private data can be further divided into different learning and validation sets.

To mimic the case in which the public and private data do not have exactly the same distribution, we also want the pseudo-public and -private data to be sufficiently different. This guides the optimizer towards selecting conservative hyperparameters that are more likely to work well on a wide range of different private datasets. If the auxiliary prediction task is classification and Y'_{pub} has multiple classes, the subset division can be based on the classes: Form each subset by selecting the samples from two (or more) classes. This strategy is based on the assumption that samples belonging to different classes have different distributions. Otherwise, for instance clustering (based on either X_{pub}, Y'_{pub} or both) could be used for finding a good subset division. An overview of the proposed hyperparameter optimization method is shown in Figure 2.

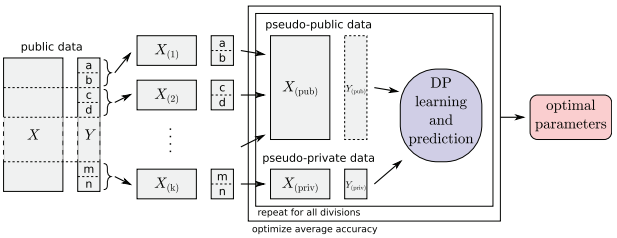

Fig. 2. The process of hyperparameter optimization. In this example, the auxiliary prediction task is assumed to be classification with multiple classes (denoted by a, b, . . ., n), which are partitioned into subsets that consist of two classes each. These are then used in a cross-validation-like hyperparameter optimization procedure

4 Results

We conducted experiments with two prediction tasks using cancer cell line gene expression data: cancer type classification and drug sensitivity prediction.

4.1 Representation learning for DP cancer type classification

We first demonstrate the method by classifying TCGA pan-cancer samples according to the annotated cancer type (e.g. lung squamous cell carcinoma) using RNA-seq gene expression data. In this task, we use the data from The Cancer Genome Atlas (TCGA) project (The TCGA authors, 2016) as both the private and public datasets. We use this example because it can be performed within the large TCGA dataset. Because most cancer type pairs are quite easy to identify, we focus on a number of most difficult pairs.

We used pre-processed TCGA pan-cancer RNA-seq data available at https://xenabrowser.net/datapages/. After further pre-processing (filtering out low-expression genes, applying RLE normalization) the dataset contains 10 534 samples, 14 796 genes and 33 distinct cancer types. We pick two cancer types as private data and the remaining cancer types form the public dataset. The main and auxiliary prediction tasks are therefore both cancer type classification tasks, but for distinct classes. For prediction, we use the differentially private logistic regression algorithm by Chaudhuri and Monteleoni (2009).

Although the split to private and public data could be done in multiple ways, the prediction task would be quite easy in many of those. Hence, we use the following procedure to produce several of these splits: (i) Consider all $\binom{33}{2}$ possible splits and run a non-DP version of the pipeline (as in Fig. 1) with PCA-based reduction to eight-dimensional space. (ii) Build a sequence of cancer type pairs by picking the pair that was the hardest to predict (i.e. has lowest classification accuracy), then from the remaining cancer types again the pair that was hardest, and so on. The result is a sequence of 16 pairs ordered by the prediction difficulty (see Table 2). (iii) Of these pairs, select the 6 hardest, as well as those 2 of the remaining pairs that had at least 200 samples in both classes.

The full testing pipeline, including the hyperparameter optimization phase, was then run separately for each of the eight selected pairs as a private dataset. In each case, the remaining 15 pairs form the $ks = 15$ subsets that were used for optimizing the hyperparameters.

4.1.1 Methods

We compare three different representation learning methods: RP, PCA and VAE (Kingma and Welling, 2014). VAE was implemented with PyTorch (Paszke *et al.*, 2017) and uses one to three hidden layers with ReLU activation functions for both the encoder and the decoder. The learning phase uses the Adam optimizer (Kingma and Ba, 2015) and is given 1 h of GPU time with early stopping. The size of the representation (for RP, PCA and VAE) and other hyperparameters for VAE (the number of layers, layer sizes, learning rate) are optimized with GPyOpt (The GPyOpt authors, 2016). We also experimented with optimizing a much larger set of hyperparameters, 12 in total, but GPyOpt had difficulties in obtaining similar levels of performance.

For each of the eight test cases we ran the hyperparameter optimization phase once, giving it 5 days of time. Then with the best found hyperparameters we ran the final testing nine times with different random seeds, and report the mean prediction accuracy as well as the standard deviation of the mean. In measuring the prediction accuracy (both for hyperparameter optimization and for final testing) we use 10-fold cross-validation.

4.1.2 Results

Figure 3 shows the final prediction accuracy in the selected eight cases for $\epsilon = 1$. Although none of the methods fully dominates the others, VAE seems to get some edge, being clearly the best in about half of the cases and doing decent job in the rest of the cases too. The selected hyperparameters are listed in Table 3. Interestingly, VAE seems to always end up with lower dimensionality of the representation than the other two methods. This could be due to the fact that VAE allows non-linear transformations which can help to

Table 2. The list of cancer type pairs ordered in descending order by the difficulty of classification

Case	First cancer type	Second cancer type
1	lung squamous cell carcinoma	head and neck squamous cell carcinoma
2	bladder urothelial carcinoma	cervical and endocervical cancer
3	colon adenocarcinoma	rectum adenocarcinoma
4	stomach adenocarcinoma	esophageal carcinoma
5	kidney clear cell carcinoma	kidney papillary cell carcinoma
6	glioblastoma multiforme	sarcoma
	adrenocortical cancer	uveal melanoma
	testicular germ cell tumor	uterine carcinosarcoma
	lung adenocarcinoma	pancreatic adenocarcinoma
7	ovarian serous cystadenocarcinoma	uterine corpus endometrioid carcinoma
	brain lower grade glioma	pheochromocytoma and paraganglioma
	skin cutaneous melanoma	mesothelioma
	liver hepatocellular carcinoma	kidney chromophobe
8	breast invasive carcinoma	prostate adenocarcinoma
	acute myeloid leukemia	diffuse large B-cell lymphoma
	thyroid carcinoma	cholangiocarcinoma

Note: The pairs selected to be tested are numbered.

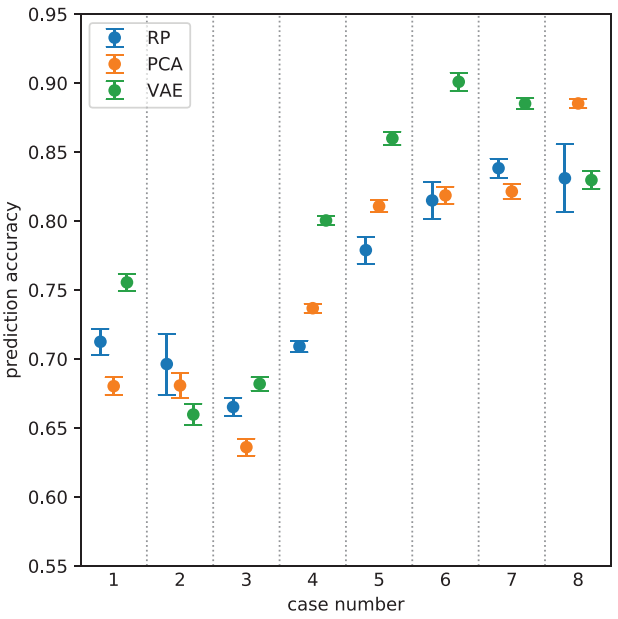

Fig. 3. Logistic regression prediction accuracy (the fraction of correctly classified samples) with $\epsilon = 1.0$ in eight cancer type classification tasks (see Table 2). Data: TCGA. Error bars show the SD of the mean accuracy over nine independent runs of the testing phase

compress the relevant information in the data into a smaller number of dimensions. On the other hand, it is not clear why RP also always chooses lower dimension than PCA.

The prediction accuracy as a function of ϵ in the Case 1 is shown in Figure 4 and the corresponding hyperparameters are shown in Table 4. As expected, larger ϵ results in better accuracy. There is some variability compared with Case 1 in Figure 3, which is mostly likely due to the results having been computed with different hyperparameters. Due to the high computational cost, variability due to hyperparameter adaptation is not included in the error bars.

The classification accuracies obtained under DP with $\epsilon = 1$ are significantly lower than using non-private logistic regression, which attains accuracies between 85 and 100% depending on the case. The reason here is probably that the datasets have few samples relative to their complexity, making DP classification at this level of DP difficult.

4.2 Representation learning for DP drug sensitivity prediction

Our main learning task is to predict the sensitivities of cancer cell lines to certain drugs. In this task we use data from the Genomics of Drug Sensitivity in Cancer (GDSC) project (Yang *et al.*, 2013) as private data. After pre-processing the data contains 985 samples, 11 714 genes and 265 drugs. The data are sparse in the sense that not all drugs have been tested on all samples. For prediction we use the differentially private Bayesian linear regression algorithm by Honkela *et al.* (2018). The DP linear regression is applied for each drug separately, using the full ϵ budget as if it was the only drug we are interested in. We then measure and report the average prediction accuracy over all drugs.

As public data we use the gene expression measurements from the TCGA data with cancer type classification as the auxiliary prediction task. The private and public datasets are unified by removing genes not appearing in both datasets. In addition, since the TCGA gene expression data are RNA-seq-based while GDSC data are based on microarrays, we apply quantile normalization to each gene in the TCGA data to make it match the distribution of the gene in the GDSC data. (Although this operation theoretically breaks the privacy guarantees, in practice we can avoid the issue by assuming that the expression distributions obtained with the microarray technology are public knowledge.) The non-private baseline uses GDSC directly, without unifying to TCGA.

Table 3. Representation dimensions (repr-dim) and other selected hyperparameters (log learning rate, the number of hidden layers, the size of hidden layers) for different cases on cancer type classification

Case	RP	PCA	VAE			
	repr-dim			log-lr	layers	layer-dim
1	7	10	5	−3.5	1	755
2	7	14	5	−4.8	1	1925
3	8	10	5	−5.3	2	2370
4	8	14	5	−5.3	2	1270
5	7	8	4	−4.3	1	260
6	8	10	5	−4.5	1	330
7	7	12	5	−3.7	2	1510
8	5	7	4	−3.8	1	88

4.2.1 Methods

In addition to RP, PCA and VAE, we also compare to DP feature selection by Sample and Aggregate framework (SAF) as presented by Kifer *et al.* (2012), as well as to using a set of 10 pre-selected genes that were used by Honkela *et al.* (2018) in the same prediction task. In the case of SAF half of the privacy budget is reserved for feature selection.

RP, PCA and VAE learning was performed in a similar manner as in the cancer type classification task. For selecting the size of the representation of SAF, we simply ran it with all possible sizes and select the best result (which is obviously unfair for the other methods and would yield a weaker privacy guarantee).

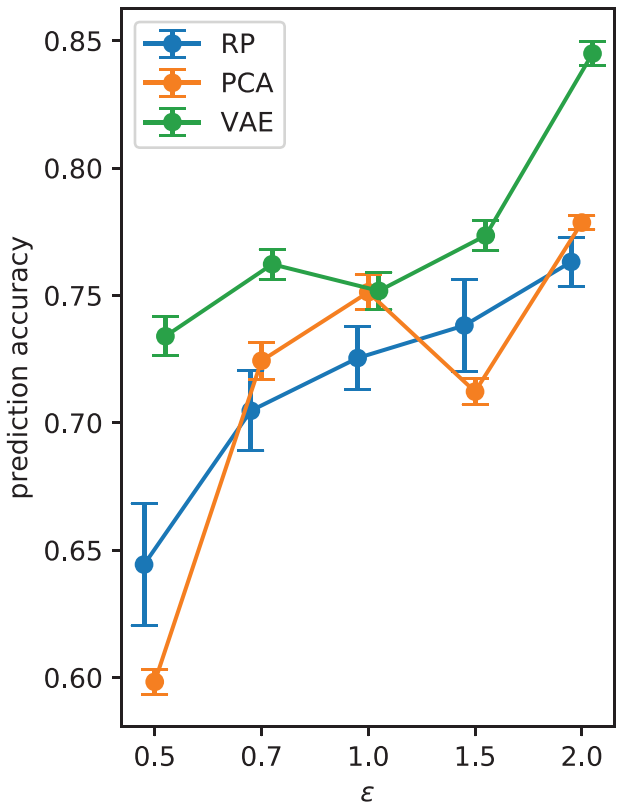

Fig. 4. Logistic regression classification accuracy in cancer type classification as a function of ϵ for Case 1. The error bars denote the SEM when repeating the DP learning but do not cover the uncertainty from hyperparameter selection.

Table 4. Selected hyperparameters for different values of ϵ in the Case 1 of cancer type classification

ϵ	RP	PCA	VAE			
	repr-dim			log-lr	layers	layer-dim
0.5	5	5	5	–4.6	1	725
0.7	6	14	5	–4.6	1	880
1	14	14	5	–4.1	1	395
1.5	9	9	5	–4.9	1	1570
2	10	11	10	–4.0	1	680

Note: See Table 3 for explanation of the columns.

4.2.2 Results

Figure 5 shows the average prediction performance, measured by Spearman's rank correlation. Here PCA and VAE are the best by some margin, both improving significantly over the results of Honkela *et al.* (2018) with 10 pre-selected genes. On the other hand, SAF is clearly the worst as the DP feature selection is essentially random due to small privacy budget, and since it leaves only half of the privacy budget for the main prediction task.

5 Discussion

Our results clearly demonstrate that representation learning with public data can significantly improve the accuracy of differentially private learning, compared with using a set of pre-selected dimensions or doing differentially private feature selection. Whether it is

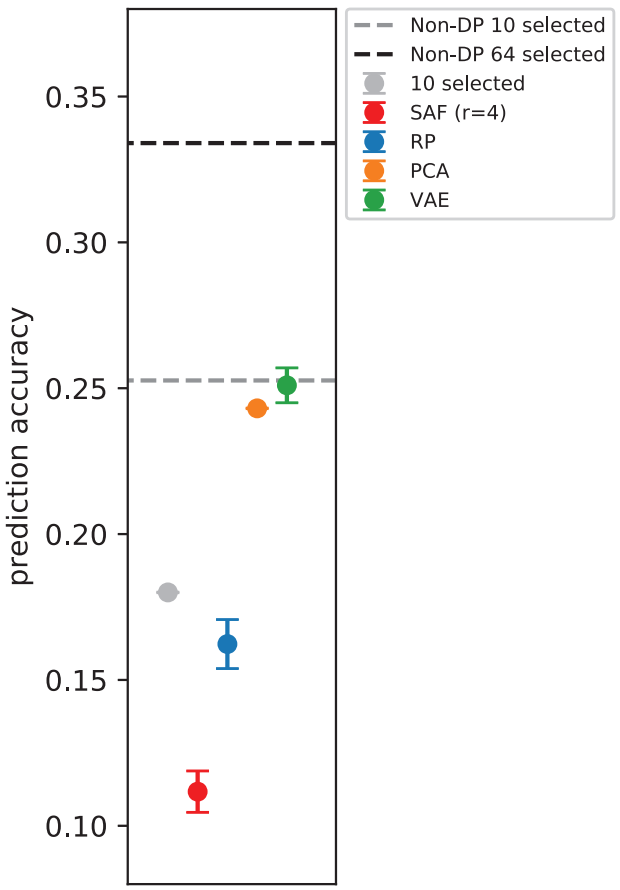

Fig. 5. The accuracy of drug sensitivity prediction (Spearman's rank correlation coefficient between the measured ranking of the cell lines and the ranking predicted by the models) with differentially private linear regression ($\epsilon = 1.0$) on the GDSC data. The dashed lines mark corresponding non-private results. The results for '10 selected genes' represent the previous state-of-the-art DP method of Honkela *et al.* (2018)

beneficial to use more advanced representation learning methods such as VAEs instead of simple methods such as PCA or RPs depends on the task. On some tasks that certainly seems to be the case.

In our current approach, the representation is learned in an unsupervised manner and the auxiliary supervised task is only used for hyperparameter selection. A natural question that we leave for further work is whether representation learning would also benefit from having an integral auxiliary prediction task that would be learned concurrently with the representation. The optimization target would in that case be a combination of unsupervised reconstruction error and supervised prediction error. This approach would require an auxiliary target variable, as is the case in this work with hyperparameter optimization.

In general, we believe DP learning can be important in opening genomic and other biomedical datasets to broader use. This can significantly advance open science and open data, and lead to more accurate models for precision medicine. So far, the accuracy of DP learning in most practical applications is not comparable to realistic non-private alternatives. Our work makes an important contribution toward making DP learning practical.

One big open question is how the choice of X_{pub} and Y'_{pub} will affect the results. If there is not enough variation in X_{pub} and the learned representation relevant for the final prediction task, it is

possible that important information may be lost. Examples of this can be seen in our experiments where RP that does not use the public data is occasionally more accurate than one of the representation learning methods, even though it is overall the least accurate method. Similarly, one needs to be careful to make sure that Y'_{pub} is sufficiently informative on the hyperparameter selection. For example, if the prediction task for Y'_{pub} is of very different level of difficulty than for Y_{priv}, it may lead to selection of highly suboptimal hyperparameters. If this becomes a problem, the selection of the hyperparameters can be performed on private data similarly as one would optimize hyperparameters of the DP learning, possibly at extra privacy cost.

In this work, the representation learning was not performed under DP. This is a clear limitation if the other dataset also needs privacy protection. This can in theory be addressed easily, by simply training the representation model under DP, but this will likely have an impact on the accuracy of the final model. Ultimately we believe that a clever combination of private and non-private data such as in our article can lead to the best results.

Acknowledgements

We thank the reviewers of an earlier workshop version of this article for helpful comments.

Funding

This work has been supported by the Academy of Finland [Finnish Center for Artificial Intelligence FCAI and grant numbers 292334, 294238, 303815, 303816 and 313124].

Conflict of Interest: none declared.

References

Abadi,M. *et al*. (2016) Deep learning with differential privacy. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS 2016)*, pp. 308–318. ACM, New York, NY, USA.

Acs,G. *et al*. (2019) Differentially private mixture of generative neural networks. *IEEE Trans. Knowl. Data Eng.*, **31**, 1109–1121.

Bingham,E. and Mannila,H. (2001) Random projection in dimensionality reduction: applications to image and text data. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001)*, pp. 245–250. ACM, New York, NY, USA.

Chaudhuri,K. and Monteleoni,C. (2009) Privacy-preserving logistic regression. In: Koller,D. *et al*. (eds) *Advances in Neural Information Processing Systems*. Curran Associates, Inc., NY, USA, Vol. 21, pp. 289–296.

Chaudhuri,K. *et al*. (2012) Near-optimal differentially private principal components. In: Pereira,F. *et al*. (eds) *Advances in Neural Information Processing Systems*. Curran Associates, Inc., NY, USA, Vol. **25**, pp. 989–997.

Costello,J.C. *et al*. (2014) A community effort to assess and improve drug sensitivity prediction algorithms. *Nat. Biotechnol.*, **32**, 1202–1212.

Dwork,C. and Roth,A. (2013) The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, **9**, 211–407.

Dwork,C. *et al*. (2006) Calibrating noise to sensitivity in private data analysis. In: Halevi,S. and Rabin,T. (eds) *Theory of Cryptography (TCC 2006)*, Springer, Berlin, Heidelberg, pp. 265–284.

Dwork,C. *et al*. (2014) Analyze Gauss: optimal bounds for privacy-preserving principal component analysis. In: *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, ACM, New York, NY, USA, pp. 11–20.

Foulds,J. *et al*. (2016) On the theory and practice of privacy-preserving Bayesian data analysis. In: Alexander,I. and Dominik,J. (eds) *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence (UAI 2016)*, pp. 192–201. AUAI Press, Arlington, Virginia, US.

Ganta,S.R. *et al*. (2008) Composition attacks and auxiliary information in data privacy. In: *Proceedings 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, ACM, New York, NY, USA, pp. 265–273.

Gymrek,M. *et al*. (2013) Identifying personal genomes by surname inference. *Science*, **339**, 321–324.

Homer,N. *et al*. (2008) Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genet.*, **4**, 1–9.

Honkela,A. *et al*. (2018) Efficient differentially private learning improves drug sensitivity prediction. *Biol. Direct.*, **13**, 1.

Ji,Z. and Elkan,C. (2013) Differential privacy based on importance weighting. *Mach. Learn.*, **93**, 163–183.

Johnson,W.B. and Lindenstrauss,J. (1984) Extensions of Lipschitz mappings into a Hilbert space. In: *Conference in Modern Analysis and Probability (New Haven, Conn., 1982), Volume 26 of Contemporary Mathematics. American Mathematical Society*, Providence, RI, pp. 189–206.

Kasiviswanathan,S.P. and Jin,H. (2016) Efficient private empirical risk minimization for high-dimensional learning. In: Maria,F.B. and Kilian,Q.W. (eds) *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, PMLR, New York, New York, USA, pp. 488–497.

Kifer,D. *et al*. (2012) Private convex empirical risk minimization and high-dimensional regression. In: Shie,M. *et al*. (eds) *Proceedings of the 25th Annual Conference on Learning Theory (COLT 2012)*, pp. 25.1–25.40. PMLR, New York, New York, USA.

Kingma,D. and Ba,J. (2015) Adam: A method for stochastic optimization. In: *Proceedings of the 3rd International Conference for Learning Representations (ICLR 2015)*.

Kingma,D. and Welling,M. (2014) Auto-encoding variational Bayes. In: Proceedings of the 2nd International Conference on Learning Representation (ICLR 2014).

Li,N. *et al*. (2007) t-closeness: Privacy beyond k-anonymity and l-diversity. In: *IEEE 23rd International Conference on Data Engineering (ICDE 2007)*, pp. 106–115.

Machanavajjhala,A. *et al*. (2007) l-diversity: privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, **1**.

Papernot,N. *et al*. (2017) Semi-supervised knowledge transfer for deep learning from private training data. In: *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.

Paszke,A. *et al*. (2017) Automatic differentiation in PyTorch. In: *NIPS 2017 Workshop*.

Raina,R. *et al*. (2007) Self-taught learning. In: *Proceedings of the* 24th *International Conference* on Machine Learning (ICML 2007). ACM Press, New York, NY, USA.

Staunton,J.E. *et al*. (2001) Chemosensitivity prediction by transcriptional profiling. *Proc. Natl. Acad. Sci. USA*, **98**, 10787–10792.

Sweeney,L. (2002) k-anonymity: a model for protecting privacy. *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, **10**, 557–570.

The GPyOpt authors. (2016) *GPyOpt: A Bayesian Optimization Framework in Python*. http://github.com/SheffieldML/GPyOpt (28 January 2019, date last accessed).

The TCGA authors. (2016) *Data Published by TCGA Research Network*. http://cancergenome.nih.gov/ (4 November 2018, date last accessed).

Wang,Y. *et al*. (2019) Differentially private hypothesis transfer learning. In: *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2018), Volume 11052 of Lecture Notes in Computer Science*, Springer International Publishing, pp. 811–826.

Xie,L. *et al*. (2018) Differentially private generative adversarial network. arXiv: 1802.06739 [cs.LG].

Yang,W. *et al*. (2013) Genomics of drug sensitivity in cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res.*, **41**, D955–D961.

Zhang,J. *et al*. (2017) PrivBayes: private data release via Bayesian networks. *ACM Trans. Database Syst.*, **42**, 1–41.