ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

Differential private relevance learning

Johannes Brinkrolf, Kolja Berger and Barbara Hammer*

CITEC center of excellence
Bielefeld University - Germany

Abstract. Digital information is collected daily in growing volumes. Mutual benefits drive the demand for the exchange and publication of data among parties. However, it is often unclear how to handle these data properly in the case that the data contains sensitive information. Differential privacy has become a powerful principle for privacy-preserving data analysis tasks in the last few years, since it entails a formal privacy guarantee for such settings. This is obtained by a separation of the utility of the database and the risk of an individual to lose his/her privacy. In this contribution, we introduce the Laplace mechanism and a stochastic gradient descent methodology which guarantee differential privacy [1]. Then, we show how these paradigms can be incorporated into two popular machine learning algorithm, namely GLVQ and GMLVQ. We demonstrate the results of privacy-preserving LVQ based on three benchmarks.

1 Introduction

The necessity to preserve a person's privacy in databases has been debated for more than twenty years [2]. While encryption can secure databases whenever private information is revealed to only the user him/herself, the setting becomes more problematic whenever important information of the database is offered to the public. This is the case if summary information or trends which have been inferred from the database are offered to the public, and it constitutes a key challenge if personal data are used to train a machine learning model, which is later rolled out to the public. While summary statistics or machine learning models deliver accumulated information and general models, it cannot be ruled out a priori that private information can be inferred from those, provided that the model is coupled with according auxiliary data.

In this context, the notion of *differential privacy (DP)* has been proposed as a formalism which provably limits the possibility to retrieve private information from published models [3]. Basically, DP formalizes the intuition that the amount of individual information which can be retrieved from such models is strictly limited per query. This way, it can formally guarantee essential properties such as immunity of the formalism to auxiliary information and privacy of individual information.

In this contribution, we propose an adaptation of Generalized Learning Vector Quantization (GLVQ) and its extension to relevance learning, the Generalized Matrix LVQ (GMLVQ). Since a prototype based classification mechanism is based on representatives within the vector space of input signals, it runs the risk of revealing sensitive information about data which have been used for training. Based on a formulation as cost optimization, we combine these methods with a differentially private stochastic gradient descent to prevent such issues [1]. We experimentally demonstrate the efficiency and effectiveness of the method.

*Funding by the CITEC center of excellence (EXC 277) is gratefully acknowledged.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

2 Differential Privacy

In the following, we briefly introduce the concept of differential privacy (DP). We shortly recapitulate the notion of DP as well as a few popular DP strategies.

Differential privacy [3, 4, 5] constitutes a strong standard for privacy guarantees for algorithms on aggregate databases. Informally, it requires that the output of a data analysis mechanism remains approximately the same if any sample in the input database is added or removed. This guarantees that a single entry cannot substantially affect the revealed outcome, hence it is impossible to retrieve sensitive individual information from the latter. Now, we define DP first and introduce specific differential private mechanisms later.

Definition (Differential Privacy [3])**.** *Assume $\varepsilon, \delta > 0$ are given. We are interested in the privacy of an operation $\mathcal{A}$ such as a machine learning algorithm, which maps a given set of training data D to a model or summary statistics revealed to the user. These outcomes might be subject to manipulation or attacks, which are unknown. To take this into account, the space of possible models is modeled as a probability space where measurable events can take place. A randomized function $\mathcal{A}$ gives* **(ε, δ)-differential privacy** *if and only if for all pairs of adjacent datasets D and D', and all events S*

$$\mathrm{P}[\mathcal{A}(D) \in S] \leq e^{\varepsilon} \cdot \mathrm{P}[\mathcal{A}(D') \in S] + \delta.$$

Here, P *refers to the probability induced by the algorithm $\mathcal{A}$. Thereby, two datasets D and D' are* **adjacent** *if and only if D can be obtained from D' by the deletion of one database entry (or vice versa).*

This notion of DP ensures the privacy of any single sample which can be used for training, because adding or removing any single sample results in e^{ε}-multiplicative-bounded changes in the probability distribution of the output of the algorithm only. DP is compositional in the sense that combining m multiple mechanisms $\mathcal{A}$ that satisfy DP for $\varepsilon_1, \ldots, \varepsilon_m$ results in a mechanism that satisfies ε-differential privacy for $\varepsilon = \sum_i \varepsilon_i$ [5]. We will call ε the privacy loss.

There are several approaches which satisfy ε-differential privacy, including the *Laplace Mechanism* [3]. The latter deals with algorithms or functions $f : \mathcal{D} \mapsto \mathbb{R}^k$ from the domain of all datasets to vectorial outputs. It adds symmetric and scaled noise to each dimension of the output. The magnitude of the required noise depends on the so-called *sensitivity* of f. It refers to the maximum difference between the outputs of two adjacent datasets. Formally, the sensitivity of f is defined as

$$\Delta f = \max_{\text{adj } D,\, D'} \parallel f(D) - f(D') \parallel_1$$

measured in L_1 norm. Given a function f the Laplace mechanism is defined as

$$\mathcal{A}_f(D) = f(D) + (Y_1, \ldots, Y_k)^T$$

for a given database D, where Y_i are i.i.d. random variables drawn from the Laplace distribution $\mathrm{Lap}\,(\Delta f/\varepsilon)$. This distribution is defined by the probability density function $\mathrm{P}[\mathrm{Lap}(\beta) = x] = \frac{1}{2\beta} e^{-|x|/\beta}$. It can be shown that the resulting

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

mechanism $\mathcal{A}_f$ is $(\varepsilon, 0)$-differential private. The Laplace mechanism constitutes a very convenient way to turn a given database query into a differentially private one. However, it has only limited applicability if f is given by a learning algorithm since its sensitivity might be complicated to bound. Therefore, more methods which directly rely on typical machine learning mechanisms have been proposed. A very popular one adds differential privacy to gradient techniques.

Differential Private Stochastic Gradient Descent is introduced by Abadi et al. [1]. Essentially, the mechanism assumes that an objective loss function $\mathcal{L}(\theta)$ with parameters θ is given which is optimized to reveal the model parameters θ. The proposed formalism computes the gradient $\nabla_\theta \mathcal{L}(\theta, \mathbf{x}_i)$ of the loss function for each sample in a random subset of size L which is taken from the training set of size N with sample probability $q = L/N$. Then, each gradient is clipped whenever its L_2 norm is greater than a threshold C. Adding Gaussian noise drawn from a normal distribution $\mathcal{N}(0, \sigma^2 C^2)$ for each dimension for a specific σ guarantees DP. The results are averaged and a noisy gradient descent according to these directions is taken.

This algorithm reflects mini-batch optimization techniques as are popular for the optimization of non-convex cost functions in machine learning. It has been shown that the resulting algorithm is (ε, δ)-differential private for any $\delta > 0$, provided $\sigma \in \Omega(q\sqrt{T \log(1/\delta)})/\varepsilon$, where T is the number of steps.

3 Differential Private G(M)LVQ

In the following we describe how we change the training of the G(M)LVQ models. We propose to use the Laplace mechanism for the initialization and the introduced gradient descent to optimize the cost function of our GLVQ and GMLVQ model. The result will be a novel version of GLVQ and GMLVQ which fulfills the requirements of differential privacy. Note, that we need to guarantee the differential privacy of all operations, including the prototype initialization and gradient update.

We are interested in classification scenarios in $\mathcal{D} \subset \mathbb{R}^d$ with k classes which are enumerated as $\{1, \ldots, k\}$. Prototype-based classifiers are defined as follows: labeled prototypes $\mathbf{w}_1, \ldots, \mathbf{w}_w$ are specified such that a good classification and representation of the data is achieved. A new sample $\mathbf{x}$ is classified by the winner takes all scheme. Standard GLVQ uses the squared Euclidean metric $d(\mathbf{x}, \mathbf{w}_j) = (\mathbf{x} - \mathbf{w}_j)^T(\mathbf{x} - \mathbf{w}_j)$ and GMLVQ learns a semi-positive definite matrix $\Lambda = \Omega^T\Omega$ and uses the squared distance function $d_\Lambda(\mathbf{x}, \mathbf{w}_j) = (\mathbf{x} - \mathbf{w}_j)^T \Lambda (\mathbf{x} - \mathbf{w}_j)$ [6]. The cost function

$$E = \sum_i \Phi\left(\frac{d^+(\mathbf{x}_i) - d^-(\mathbf{x}_i)}{d^+(\mathbf{x}_i) + d^-(\mathbf{x}_i)}\right)$$

is introduced by Sato and Yamada [7], where Φ is a monotonic increasing function, e.g., the logistic one, $d_+(\mathbf{x}_i)$ the squared distance of $\mathbf{x}_i$ to the closest prototype of the correct class and $d^-(\mathbf{x}_i)$ the closest squared distance to another prototype of a different class than $\mathbf{x}_i$. Training takes place based on a given training set, by initializing the prototypes within the class centers and minimizing the cost term E with respect to the prototypes and, for GMLVQ, also the metric parameters afterwards. In the following we use the identity $\Phi(\mathbf{x}) = \mathbf{x}$.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Initialization: For simplicity, we assume that we use one prototype per class. In standard G(M)LVQ, we initialize each prototype by the class centers. These can be calculated based on the sum of all samples of each class and the number of class members. These operations can be enhanced to DP versions based on the Laplace mechanism as follows: We want to compute $\mathbf{w}_j = 1/N_j \sum_{i:c(\mathbf{x}_i)=j} \mathbf{x}_i$ for each class j. The cardinalities of the classes are given by the function $f : \mathcal{D} \mapsto \mathbb{N}^k$, $f(D) = (N_1, N_2, \ldots, N_k)$. This function has a sensitivity $\Delta f = 1$ because adding or removing one data point in the dataset changes only the output of one N_i by one. In the literature, these functions are also known as histogram queries [5]. The sum of all points in each class is given by the function $g : \mathcal{D} \mapsto \mathbb{R}^{k \cdot d}$, $g(D) = \left(\sum_{i:c(\mathbf{x}_i)=1} \mathbf{x}_i, \ldots, \sum_{i:c(\mathbf{x}_i)=k} \mathbf{x}_i\right)$. Without loss of generality, we assume that the samples are normalized such that $\mathcal{D} \subset [-1, 1]^d$. Then the sensitivity of the function is $\Delta g = d$. One adjacent dataset can change the output at least by one in each dimension in the L_1 norm because the classes are disjoint sets.

For a given privacy loss ε_1 we obtain all N_i and all sums with the Laplace Mechanism in a differentially private way. If we use the noise scales $\beta_f = 2/\varepsilon_1$ for the function f and $\beta_g = 2d/\varepsilon_1$ for g we achieve a ε_1-differential private mechanism altogether due to standard arguments for composition. Note, that the noise does not depend on the number of samples in the dataset. Hence, it has a smaller impact on big ones and a higher if it is getting smaller.

Gradient descent: For the gradient descent, we rely on the algorithm as described in chapter 2 by Abadi et al. [1]. Let L be the batch size, C the gradient norm bound, $q = L/N$ the sample probability for one sample, E the number of epochs and $T = E/q$ the runs of the gradient descent and the number of updates. For GLVQ we just have the gradients of the prototypes which we have to clip. In the case of GMLVQ, the parameters of the projection matrix Ω are also clipped together with the parameters for the prototypes in the L_2 norm. For a given ε_2 and δ we can calculate the noise scale by $\sigma = 2q\sqrt{T \log(1/\delta)}/\varepsilon_2$. Hence, the total privacy loss of the whole training is $\varepsilon = \varepsilon_1 + \varepsilon_2$ and we obtain an (ε, δ)-differential private algorithm.

4 Experiments

We test our approach with three real world datasets, MNIST [8], Motion Tracking [9] and Image Segmentation [10]. The first has 70.000 instances with pictures of handwritten digits. The second recorded accelerator data by a mobile phone to classify motions and has 10.299 samples. The last one has real valued image descriptors of 2.310 small landscapes image patches. For the benchmark tests, we repeat a 5-fold cross validation five times. The total privacy loss is split into $\varepsilon_1 = 0.2\varepsilon$ for the initialization step and $\varepsilon_2 = 0.8\varepsilon$ for the parameter optimization. The other parameters are $\delta = 10^{-5}$, $q = 0.01$, $C = 0.5$ and $E = 50$. We compare our approach with non-private versions of GLVQ and GMLVQ. There, the optimum is found by a standard stochastic gradient descent (SGD) and by the BFGS algorithm, a quasi-Newton method for solving nonlinear optimization problems [11]. It represents the minimum error which we can reach based on

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

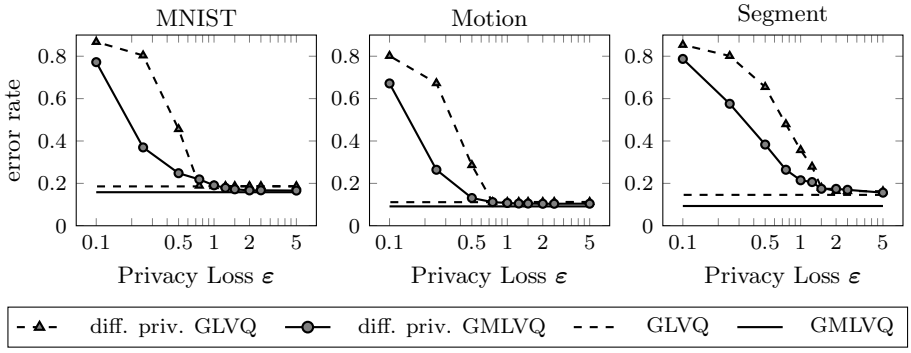

Fig. 1: GLVQ and GMLVQ test error rates for our approach and the non-private version with SGD optimization on the datasets. Only the privacy loss varies and all others are fixed ($C = 0.5$, $q = 0.01$ and $E = 50$).

GLVQ and GMLVQ respectively. For MNIST we use one fold for training and four as the test set. For Image Segmentation and Motion Tracking, we swap the training and test sets due to the smaller set sizes.

In Fig. 1 results for different privacy loss for all datasets are shown. The dashed lines are the results for GLVQ and the solid for GMLVQ. One can see that the privacy loss effects the classification strongly and the curves fall sharply at a certain point. For the datasets the points vary between $\varepsilon = 0.75$ for Motion Tracking and $\varepsilon = 2.5$ for Segmentation. This is due to the higher effects of the noise on smaller datasets or smaller lot sizes. The choice of the other hyperparameters has a strong influence in critical regions for epsilon only, being rather robust for epsilon larger 1. As an example, for Motion data we test ten different values for $C \in [0.05, 2]$ and $E \in [10, 100]$ and 20 for $q \in [0.0005, 0.1]$. In this contribution we report the standard deviation of the test errors for three different ε. In the critical region where $\varepsilon = 0.75$ the standard deviations are 0.1275, 0.0361 and 0.0899 for C, E and q, respectively. If $\varepsilon = 1$ they are 0.0412, 0.0081 and 0.037 and if $\varepsilon = 2.5$ they are 0.0062, 0.006 and 0.0062. The impact of C and q are higher than for E due to the variances of the noise in the gradient descent ($\sigma^2 \sim C^2(q\sqrt{E/q}/\varepsilon)^2$).

In Tab. 1 the means and the standard deviations of the error rates for all

dataset	DP $\varepsilon = 0.75$	DP $\varepsilon = 1.5$	DP $\varepsilon = 2.5$	non priv. SGD	non priv. BFGS
MNIST	0.1893 (0.0042) 0.2188 (0.0162)	0.1871 (0.0020) 0.1721 (0.0067)	0.1871 (0.0020) 0.1673 (0.0033)	0.1857 (0.0022) 0.1583 (0.0031)	0.1853 (0.0018) 0.1484 (0.0021)
Motion	0.1121 (0.0061) 0.1116 (0.0074)	0.1123 (0.0063) 0.1048 (0.0080)	0.1121 (0.0058) 0.1038 (0.0057)	0.1112 (0.0062) 0.0914 (0.0068)	0.1111 (0.0062) 0.0897 (0.0066)
Segment	0.4793 (0.0779) 0.2642 (0.0432)	0.1792 (0.0152) 0.1745 (0.0205)	0.1635 (0.0124) 0.1696 (0.0233)	0.1458 (0.0133) 0.0932 (0.0108)	0.1458 (0.0132) 0.0870 (0.0109)

Table 1: Mean and std. dev. in brackets for test error rates. As a baseline, the results of a non-private training with SGD and a BFGS optimizer are given. The first rows for each dataset are results for GLVQ the second for GMLVQ.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

three benchmark sets are listed. For GLVQ we often get trained models which are as good as the non-private ones. For GMLVQ the BFGS optimization finds better parameters than SGD. Here, the private versions get more trouble due to the noise in the relevance matrix. Even small changes in the values of the matrix can cause a worse classification. To test the matrix sensitivity, we add normal distributed random numbers with variance $\sigma = 0.025$ on each element of the relevance matrix. We observe an increase of 0.0277 ± 0.001 (from 0.1484 to 0.1761) of the error rate for the original GMLVQ approach and the MNIST dataset. For the Motion dataset the error increases by 0.0234 ± 0.0085 using the same settings.

5 Conclusions

We have introduced an approach to obtain a differential private version of GLVQ and GMLVQ. We changed the initialization step and used a differential private SGD for optimization. In the results, we showed that for the real-world dataset MNIST a privacy loss $\varepsilon > 1.5$ suffices to achieve a differential private model that is as good as the non-private versions. For smaller datasets, a bigger ε is needed because the noise has a bigger impact. These promising results open the way towards LVQ variants which can publicly be released, e.g., in the medical domain albeit it has been trained based on sensitive data.

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] J. Richard Dowell. An overview of privacy and security requirements for data bases. In *Proceedings of the 15th Annual Southeast Regional Conference*, ACM-SE 15, pages 528–536, New York, NY, USA, 1977. ACM.

[3] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[4] Cynthia Dwork. A firm foundation for private data analysis. *Commun. ACM*, 54(1):86–95, 2011.

[5] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[6] Petra Schneider, Michael Biehl, and Barbara Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561, 2009.

[7] Atsushi Sato and Keiji Yamada. Generalized learning vector quantization. In David S. Touretzky, Michael Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 423–429. MIT Press, 1995.

[8] 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.

[9] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *21st European Symposium on Artificial Neural Networks, Bruges, Belgium*, 2013.

[10] M. Lichman. UCI machine learning repository, 2013.

[11] R. Fletcher. *Practical Methods of Optimization; (2Nd Ed.)*. Wiley-Interscience, New York, NY, USA, 1987.