A Survey on Differentially Private Machine Learning

I. Differentially Private Supervised Learning

Supervised machine learning is that one can learn the mapping function from the training data to the respective known labels, and the goal is to train a model to predict accurate output(label) with the new input data. It is called classification problem if the label is a category, such as "blue" or "red" or "0" or "1". Alternatively, a regression problem is when the label is continuous like "weight". We will introduce supervised machine learning algorithms with differential privacy in this section.

A. Naive Bayes Model

In machine learning, Naive Bayes model is a simple but powerful classifier which predicts label Y with features in X. A Naive Bayesian model is particularly used for large datasets because there are no complicated iterative parameter estimates. The Naive Bayesian classifier is based on Bayes' theorem and the naive independence assumptions between features. Bayes theorem provides a method for calculating the posterior probability:

$$P\left(Y|X\right) = \frac{P\left(X|Y\right)P\left(Y\right)}{P\left(X\right)} \tag{1}$$

The conditional independence assumptions means that the features in X are independent of each other. On the premise of this hypothesis, we can reduce the parameter scale required to calculate the conditional probability. We can suppose X has n features, and then:

$$P\left(X|Y\right) = P\left(X_1|Y\right)P\left(X_2|Y\right)\cdots P\left(X_n|Y\right) \tag{2}$$

Given x, we can determine which category it belongs to by looking for the output with the highest posterior probability: $argmax_{y_k}P\left(y_k|x\right)$. In summary:

$$P\left(y_k|x\right) = \frac{P\left(x|y_k\right)P\left(y_k\right)}{\sum_k P\left(x|y_k\right)P\left(y_k\right)} \tag{3}$$

where the $P\left(y_k\right)$ in the molecule can be easily calculated based on the training set and $P\left(x|y_k\right) = \prod_{i=1}^{n} P\left(x_i|y_k\right)$. The naive Bayes classifier can ultimately be expressed as:

$$f(x) = argmaxP(y_x)\prod_{i=1}^{n} P\left(x_i|y_k\right) \tag{4}$$

[1] would like to apply the rigorous model of differential privacy to develop a Naive Bayes classifier which can protect privacy of users' data to the best extent possible. The basic idea of this paper is to derive the sensitivity of the classifier parameters and analyze them to know how to add Laplacian noise. This paper mentions that the calculation of conditional probabilities is different for nominal and numerical attributes, and then it discusses how to derive sensitivity separately in these two cases. In addition, it should be noted that in both cases, the sensitivity of the prior probability can be calculated in a similar way. After this, they add the Laplacian noise of the appropriate scale to the parameters such as the counts of categorical attributes. The processed parameters are then used in the standard Naive Bayes model so that the classifier can provide a strong privacy guarantee. This description can not insure the computed parameters are non-negative because the noise added can be negative. So they resample the Laplace distribution as many times to avoid this problem.

[2] mainly randomly divide the data to be trained and aggregate the intermediate results of the data in the partition. They propose a hypothesis that the variance of the posterior distribution of the data of a partition is proportional to the variance of the posterior distribution of the complete data set, thus finding the posterior probability. They prove that the average probability approximates the posterior probability of giving a complete data set. Finally, the difference private statistics published to analysts contain noise η to ensure differential privacy, where η is a draw from the Laplace distribution.

Previous work has proven that polynomial approximation is useful in differential privacy. [3] propose Bernstein functional mechanism to achieve the privatization of function value mapping and prove that this mechanism is applicable to explicitly and implicitly defined functions through theoretical analysis. They use an iterative Bernstein operator for polynomial approximation of the target function and polynomial coefficient perturbation to ensure privacy by eliminating the approximation coefficients.

In addition to adding noise in the original parameter domain, differential privacy can also be achieved by adding noise in the frequency domain. [4] study how to communicate the results of Bayesian inference to third parties under the premise of ensuring differential privacy. To this end, they implement two mechanisms for the probabilistic graphical model of Bayesian inference, including adding noise directly to the posterior parameters or their Fourier transforms. They mainly focuse on the Bayesian inference of PGMs (probabilistic graphical models), introduce a maximum-a-posteriori private mechanism and use this mechanism to preliminarily demonstrate that independent structures should affect privacy.

B. Linear Regression

Linear regression is commonly used for predictive analysis and attempts to fit a relationship between two variables with a linear equation where the model output is a continuous value. We can make an estimation function $f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$, where the x_1 and x_2 are the specific feature values and the parameter θ describes the influence of each feature. In the form of a vector, it is $f(x) = \theta^T X$. We generally choose the sum of squares as the loss function to evaluate whether θ is appropriate. There are many ways to adjust θ so that loss function takes the minimum, such as min square method or gradient descent method.

[5] study the relationship between differential privacy and stable learning theory and proved that output perturbation can get better privacy/utility trade-offs. They propose three methods of perturbation in this paper and apply the output perturbation which is one of the easy-to-implement mechanisms to the linear regression. For example, in Data-Independent Output Perturbation, they put $R = 1$ and pick the λ as $\sqrt{d/n\varepsilon_p}$ with the theorem mentioned in this paper. Their mechanisms are regularized versions of least squares optimization and they also describe in detail how to select parameters to regularize linear regression.

Regression involves solving an optimization problem, it is not easy to choose methods for ε-differentially private regression because we have difficulty in determining the minimum amount of the necessary noise. [6] propose a mechanism to perturb the objective function of the optimization problem to enforce privacy protect. Roughly speaking, they use FM(an extension of the Laplace mechanism) to inject noise directly into the parameter θ when implementing this mechanism on linear regression and then they optimize ω during the operation of the noise objective function. Finally the noise results of the perturbed optimization problem then achieves ε-differential privacy. This mechanism is also mentioned in [7], they add $Lap(2(d+1)2/\varepsilon)$ noise to each multi-item coefficient.

In [8], they focuse on the space restricted streaming algorithms and explore differential privacy on streamed data. Unlike traditional differential privacy mechanisms, they reversibly perturb the input instead of the output by

adding noise. They random sample of the rows or columns of the streamed matrix or generated a random sketch of the matrix. After that, they perturbe the input matrix and then multiply noise matrix(e.g.,the random Gaussian matrix).

In addition, it is mentioned in some literature that various diagnostic techniques are used to evaluate the model capabilities in order to modify the model according to actual needs. In releasing differentially private residual plots, it is important to determine the bounds on the predicted y and the residuals r. [9] propose a algorithm to estimate bounds on residuals using ϵ_1 of the total ϵ budget and compute distributions of residuals using the remaining budget $\epsilon_2 = \epsilon - \epsilon_1$ for linear regression.

C. Linear SVM

Support vector machine (SVM) is a two-class model and implements an original cutting plane algorithm. The basic idea of SVM learning is to solve the separation hyperplane that can correctly divide the training data set and have the largest geometric interval. This hyperplane can be represented by a classification function:

$$f(x) = w^T x + b \tag{5}$$

The parameter w and b can be computed by minimizing $\max(\frac{1}{2}||w||^2 - \Sigma_{i=1}^{N}\alpha_i(y_i(w \cdot x_i + b) - 1))$, where α_i is the Lagrange multiplier. After deriving we can get:

$$w^* = \Sigma_{i=1}^{N}\alpha_i^* y_i x_i \quad b^* = y_j - \Sigma_{i=1}^{N}\alpha_i^* y_i(x_i \cdot x_j) \tag{6}$$

Convex optimization is most commonly used for empirical risk minimization (ERM). The objective is to approach the expected risk with the minimum value of empirical risk. And Convex ERM is often used to fit the support vector machine model. [10] introduced a differentially private algorithm that can improve on the non-smooth loss functions for convex empirical risk minimization and use three techniques to implement this algorithm such as gradient descent, exponential sampling and localization. They use a simple output perturbation algorithm, first compute $\theta^* = argminL(\theta; D)$ and add noise according to the sensitivity of θ^*.

[11] considered the level of differential privacy guaranteed for statistical query and studied the trade-off between utility and risk in private SVM. They proposed two effective mechanisms, one for finite dimensional feature mapping and the other for potential infinite dimensional feature mapping. The first mechanism is as follows: after get the SVM's weight vector, they calculated the L_1-sensitivity of the weight vector and added Laplace noise with scale equal to sensitivity divided by β to show the β-differential privacy. In addition, they exploited the algorithmic stability of regularized ERM to calculate sensitivity. The second mechanism will be mentioned later.

Model analysis tasks is to find the parameters of the model that best fit the dataset. The effectiveness of the model fitting algorithm and the amount of disturbance required to satisfy the privacy guarantee become the key to the quality of the tasks. [12] focused on PrivGene, a fitting solution based on genetic algorithms, to achieve higher overall quality of results with less perturbations. It is worth noting that PrivGene uses a new technique called the enhanced exponential mechanism which improve the exponential mechanism by using the special properties of model-fitting tasks to perform random perturbations. They use two parent vectors w^1 and w^2 as input and obtain two new vectors v^1 and v^2 with recombining the elements of the input, and then they add random noise to the elements of v. Then they applied PrivGene to model fitting tasks such as SVM classification and logistic regression and proved that it is superior to existing methods while satisfying differential privacy.

D. Logistic Regression

Logistic regression aims to learn a classification model from features and the most common used one in practice is the two classification. First, it use logistic function ($g(z) = \frac{1}{1+e^{-z}}$) to map the linear combination of features to (0,1). We construct the prediction function as: $h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$, where x is an n-dimensional vector, function g is a logistic function and $h_\theta(x)$ indicates the probability that the result is 1. From the above, when $\theta^T x \gg 0$, $h_\theta(x)$=1, otherwise $h_\theta(x)$=0. Then we construct the loss function and find the regression parameters θ to minimize the loss function. For a model consisting of a loss function, the model generalization ability may be poor due to excessive fitting of the training data. Overfitting problems can be solved by adding a regularization or penalty to the empirical risk.

[6] analyzes sensitivity and inserts noise on the objective functions. To solve logistic regression problem, they use the low-end part of Taylor expansion of the function and add noise to the parameters. However, their mechanism is not applied to the complicated objective function(e.g., Cox regression).

Objective perturbation technique can be considered to add a linear perturbation $< B, \theta >$ to the empirical loss, where B is a random vector drawn from a gamma distribution. [13] proposed a differential private algorithm based on convex empirical risk minimization (ERM) and aimed to find solutions with few non-zero coefficients. They significantly improve the existing objective perturbation algorithm for convex ERM problems with less noise and this makes the algorithm more accurate. They propose Gaussian distribution (instead of gamma) can be used to obtain the perturbation B.

The most important point to achieve end-to-end differential privacy is how to find an effective procedure for differentially private parameter tuning. Under certain stability conditions, [14] proved it is possible to implement efficient parameter tuning with differential privacy in a more general setup. It is worth noting that the training set size and the privacy budget are independent of the number of parameter values during their training.

[15] pay attention to release GWAS data in a way that protects privacy. To solve regression problems with convex penalty functions, they proposed an end-to-end differentially private method. They focused on penalized logistic regression with elastic-net regularization and extended the method for selecting the regularization parameters that is mentioned in [14]. Based on the results of [13], they use cross-validation to implement a differentially private procedure for penalized logistic regression.

E. Kernel SVM

In the dual problem of linear support vector machine learning, we replace the inner product with the kernel function to get the classification decision function of kernel SVM. The kernel function represents the inner product between two instances after a nonlinear transformation. $K(x, z)$, as a kernel function, means a mapping from input space to feature space $\phi(x)$. So there is $K(x, z) = \phi(x) \cdot \phi(z)$ for any (x, z) in input space. The resulting classification decision function can be expressed as: $f(x) = sign\left(\Sigma_{i=1}^{N} \alpha_i^* y_i K(x, x_i) + b^*\right)$. Here we introduce a commonly used kernel function - Gaussian kernel function: $K(x, z) = exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$.

Considering the characteristics of biomedical data, [16] developed a differential private support vector machine (SVM) model using existing public data and private data information. They computed the parameters in the RBF kernel function with public data and trained private classifers with linear SVM based on the private data. Finally, they inject noise into the parameters through the Laplace mechanism.

In [17], they considered how to implement differential privacy by accessing training features only through kernel functions. They accessed to each data just through the kernel ERM and constructed three simpler models

for kERM where they provided different algorithms and inject different Laplacian noise into the output of each algorithm to guarantee differential privacy to the training data.

F. Decision Tree Learning

Decision Tree is a basic classification and regression method and it is a tree structure that describes the classification of instances. When using the model for prediction, the judgment nodes are sequentially judged according to the input parameters, and finally the leaf nodes are predicted results. The core of decision tree learning is to construct a suitable decision tree by learning the data and selecting the judgment nodes. In the decision tree algorithm, we select the optimal features by Gini impurity or entropy and segment the dataset with the optimal features. We may have over-fitting in the decision tree we train, so we can manually set a threshold of information gain to achieve decision tree pruning. The decision tree mainly solves the classification problem (the result is discrete data). If the result is a number, the variance can be used instead of entropy or Gini impurity. Commonly used decision tree algorithms include ID3 algorithm, C4.5 algorithm and CART algorithm.

[18] studied using decision trees to build private classifiers which can satisfy differential privacy. They constructed privacy-preserving ID3 decision trees with low-level differentially private sum queries, but this model can not provide high privacy and high accuracy at the same time. Then They used a random decision tree to generate a classifier to solve this problem. Note that they added $Lap\left(\frac{1}{\epsilon}\right)$ noise to the component of V and then they released the resulting noisy vector. In addition, in order to periodically attach new data to the existing database, they proposed a differential privacy algorithm.

[19] studied to construct a sufficient number of random decision trees in which any given node is chosen uniformly at random from all the features to eventually form a random forest. They add perturbation noise to the counters in leaves rather than the inner nodes to meet the requirements of differential privacy. They proved that majority voting, threshold averaging and probabilistic averaging are good differentially private classifiers.

REFERENCES

[1] J. Vaidya, B. Shafiq, A. Basu, and Y. Hong, "Differentially private naive bayes classification," in *Ieee/wic/acm International Joint Conferences on Web Intelligence*, 2013, pp. 571–576. I-A

[2] G. Amitai, "Bayesian inference via partitioning under differential privacy," 2018. I-A

[3] F. Ald胰and B. I. P. Rubinstein, "The bernstein mechanism: Function release under differential privacy," *Computer Science*, 2015. I-A

[4] Z. Zhang, B. I. P. Rubinstein, and C. Dimitrakakis, "On the differential privacy of bayesian inference," pp. 2365–2371, 2015. I-A

[5] X. Wu, M. Fredrikson, W. Wu, S. Jha, and J. F. Naughton, "Revisiting differentially private regression: Lessons from learning theory and their consequences," *Computer Science*, 2015. I-B

[6] Z. Zhang, Z. Zhang, Y. Yang, Y. Yang, and M. Winslett, "Functional mechanism: regression analysis under differential privacy," *Proceedings of the Vldb Endowment*, vol. 5, no. 11, pp. 1364–1375, 2012. I-B, I-D

[7] B. N. Wang, X. J. Fang, and D. O. Computer, "Based on differential privacy of linear regression analysis," *Computer Knowledge & Technology*, 2016. I-B

[8] J. Upadhyay, "Differentially private linear algebra in the streaming model," *Eprint Arxiv*, 2017. I-B

[9] C. Yan, A. Machanavajjhala, J. P. Reiter, and A. F. Barrientos, "Differentially private regression diagnostics," in *IEEE International Conference on Data Mining*, 2017, pp. 81–90. I-B

[10] R. Bassily and A. Thakurta, "Differentially private empirical risk minimization: Efficient algorithms and tight error bounds," *Computer Science*, pp. 464–473, 2014. I-C

[11] B. I. P. Rubinstein, P. L. Bartlett, H. Ling, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for svm learning," *Eprint Arxiv*, vol. 4, no. 1, 2009. I-C

[12] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett, "Privgene:differentially private model fitting using genetic algorithms," in *ACM SIGMOD International Conference on Management of Data*, 2013, pp. 665–676. I-C

[13] D. Kifer, A. Smith, and A. Thakurta, "Private convex empirical risk minimization and high-dimensional regression," *Journal of Machine Learning Research*, vol. 1, 2013. I-D

[14] K. Chaudhuri and S. Vinterbo, "A stability-based validation procedure for differentially private machine learning," in *International Conference on Neural Information Processing Systems*, 2013, pp. 2652–2660. I-D

[15] F. Yu, M. Rybar, C. Uhler, and S. E. Fienberg, "Differentially-private logistic regression for detecting multiple-snp association in gwas databases," vol. 8744, pp. 170–184, 2014. I-D

[16] H. Li, L. Xiong, L. Ohnomachado, and X. Jiang, "Privacy preserving rbf kernel support vector machine," *Biomed Res Int*, vol. 2014, no. 1, p. 827371, 2014. I-E

[17] P. Jain and A. Thakurta, "Differentially private learning with kernels," in *International Conference on Machine Learning*, 2014, pp. 118–126. I-E

[18] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, *A Practical Differentially Private Random Decision Tree Classifier*. IIIA-CSIC, 2012. I-F

[19] M. Bojarski, A. Choromanska, K. Choromanski, and Y. Lecun, "Differentially- and non-differentially-private random decision trees," *Eprint Arxiv*, 2014. I-F