Privacy Preserving Machine Learning: Related Work

Bargav Jayaraman

December 6, 2015

Privacy Preserving Machine Learning (PPML) has been extensively studied in literature and the related works can be broadly classified into three categories based on the scenario they target.

PP learning of ML algorithm on centralized data

A practical scenario of PPML is where only one central party has the entire data on which the ML algorithm has to be learned. Agrawal and Ramakrishnan [1] proposed the first method to learn a Decision Tree classifier on a database without revealing any information about individual records. They consider public model - private data setting where the algorithm and its parameters are public whereas the data, over which the algorithm is trained, is kept private. The privacy of individual records in data is achieved through data perturbation. This notion of privacy was later identified as differential privacy by Dwork [2]. Though the data is private in the scenario considered by Agrawal and Ramakrishnan, where algorithm is trained on the data and only the learned algorithm is revealed, some orthogonal scenarios consider publishing the perturbed data thereby leading to public data setting with respect to the perturbed data. Chaudhuri and Monteleoni [3] proposed learning of Logistic Regression with differential privacy in which they consider the same setting of public model - private data as above. However, a major distinction is that they perturb the learning algorithm rather than the data itself (unlike the method of Agrawal and Ramakrishnan) thereby not revealing information about individual data points upon which the algorithm is learned. Chaudhuri et al. [4] further generalized this solution for learning Support Vector Machine (SVM) under the same settings. Zhang et al. [5] gave a differential privacy solution to learning Logistic and Linear Regression models by perturbing the objective function similar to Chaudhuri and Monteleoni under the public model - private data setting. Rubinstein et al. [6] came up with differentially private learning of SVM under public model - private data setting, where differential privacy is guaranteed by introducing weight regularization in objective loss function. Jain et al. [7] proposed a mechanism of training Deep Belief Networks with differential privacy using drop-out (unlike the above methods of objective perturbation which rely on regularization for differential privacy). Like all the above differential privacy mechanisms, Jain et al. also consider public model - private data setting.

All these works on differential privacy assume semi-honest model for learning the ML algorithms. Although differential privacy guarantees statistical privacy rather than the usual information theoretic privacy, it is practically important as it maintains the anonymity of individual records. This might be an interesting topic to explore.

PP learning of ML algorithm on distributed data

Various works have been proposed for learning a common ML algorithm across private data of multiple parties, which utilize secure multi-party computation.
(MPC)\footnote{Secure Multi-Party Computation itself has a rich literature which is not included in this survey} protocols. Lindell and Pinkas \cite{Lindell2003} proposed the first such model in which two parties want to jointly compute a Decision Tree on their private data, i.e., data is horizontally\footnote{In horizontal partitioning, each party has a disjoint set of records} partitioned among both the parties. They consider the public model - private data setting, where the data of each party is private and only the final trained model is revealed to both the parties. Du et al. \cite{Du2009} proposed learning of Linear Regression model for multivariate statistical analysis under the same public model - private data setting but they consider that the data is vertically\footnote{In vertical partitioning, each party has all records but only a sub-set of data attributes of the records} partitioned among the two parties. Vaidya and Clifton \cite{Vaidya2009} came up with Association Rule mining on boolean and categorical attributes of vertically distributed data under the same settings as above. Also, they point out to some possible future research which include quantitative association rule mining on non-categorical data and the non-trivial case of extending to multi-party settings which considers collusion. In \cite{Vaidya2010}, Vaidya et al. propose learning and application of Naive Bayes classifier in 2PC and MPC settings (under semi-honest settings) on both vertically and horizontally partitioned data. Jagannathan et al. \cite{Jagannathan2010} proposed collaborative k-clustering algorithm on horizontally distributed data under the public model - private data settings, however, in their model intermediate clustering information is not revealed and only the final cluster centres are revealed to the parties making the model more private. Zhong et al. \cite{Zhong2015} consider a different scenario of distributed data where the data is horizontally partitioned among the data owners such that each owner has one row of data and the data miner mines the frequency over all the rows using homomorphic encryption. The work preserves the confidentiality of honest data owners even in the presence of some corrupted data owners collaborating with data miner. The frequency is further utilized to train a Naive Bayes classifier under semi-honest settings. This work, in a way, shares the goal of differential privacy since it preserves the confidentiality of individual records. Again, this work falls into the public model - private data setting but is more practical since it incurs only single round of interaction between the data miner and each data owner. Yang and Wright \cite{Yang2010} proposed a 2PC model where both parties collaboratively train a Bayesian Network on their vertically partitioned data. Yu et al. \cite{Yu2011} achieve learning of SVM over vertically partitioned data. Laur et al. \cite{Laur2011} consider a restricted vertically partitioned data such that server has all the features and client has all the classification labels. Pathak et al. \cite{Pathak2012} propose an MPC model with differential privacy where individual parties train their local classifiers and a curator aggregates all the classifiers along with noise for differential privacy. Recently Shokri and Shmatikov \cite{Shokri2015} have come up with the first implementation of collaborative Deep Learning on private data of multiple parties. All these methods consider semi-honest settings where the parties do not deviate from the protocol but may try to infer the private data of other parties. Semi-honest setting seems practically sufficient for collaborative learning of ML algorithm, since all parties wish to correctly learn the algorithm.

\section*{PP application of learned ML algorithm on third-party data}

This scenario is most relevant to my proposed protocol. Brickell and Shmatikov \cite{Brickell2010} propose a method of learning a Decision Tree on remote server’s dataset and also perform classification on third party data using Garbled Circuits (GCs). Though they use free-XOR implementation, the GC bandwidth could be further reduced using half-gates implementation of Zahur et al. \cite{Zahur2010}. Pathak et al. \cite{Pathak2011, Pathak2012} propose a 2PC scenario where Alice has data and Bob has ML algorithm, where Alice sends encrypted data and Bob uses homomorphic encryption for evaluating ML algorithm over Alice’s encrypted data. Barni et al. \cite{Barni2011} also propose a 2PC client-server model in which client sends the encrypted data to server and server applies ML algorithm using homomorphic encryption and GCs (free-XOR implementation). They imple-
ment ML algorithms like Decision Trees and 2-layer Neural Networks. Bos et al. [24] use homomorphic encryption over encrypted medical data for Logistic Regression. Xie et al. [25] consider the cloud scenario for applying neural networks on encrypted data and propose some interesting modification to the activation functions of neural networks. They explained their method theoretically and have left practical implementation for future work. Graepel et al. [26] and Bost et al. [27] also consider the scenario where client wants to evaluate server’s ML algorithm on his/her private data. They implement various ML algorithms like Naive Bayes, SVM, Decision Trees Perceptrons, Fisher Linear Discriminant classifier, etc.

All the above methods fall under semi-honest setting which is not sufficient for privacy preserving application of ML algorithms. Though Graepel et al. [26] argue that client can verify the correctness by sending known queries. The above scenario of classification involves the evaluation of ML algorithm on algorithm owner side and thus assumes that data owner knows the purpose of the ML algorithm (i.e., what is the pattern the ML algorithm identifies, for instance). On the other hand, my scenario considers the opposite, i.e., evaluation of ML algorithm occurs at data owner side and keeping the algorithm’s purpose anonymous from data owner (Though the algorithm structure is known to data owner, but the parameters are unknown).

Conclusion

Most of the related works on PPML are focused on learning of ML algorithm. The next natural step would be to apply the ‘learned’ algorithm on third-party data without revealing the ‘intent’ of the algorithm to the third-party. The existing works so far consider semi-honest setting and hence there is no guarantee that the ML algorithm is being correctly evaluated. Work has to be done on achieving the above functionality under malicious setting. Recently, the deep learning algorithms are gaining much acclaim in the machine learning community and are being readily utilized in natural language processing and computer vision tasks. Hence it would be beneficial if such algorithms are applied in a privacy preserving way, thereby motivating further research in this problem space.

References


[25] Pengtao Xie, Misha Bilenko, Tom Finley, Ran Gilad-Bachrach, Kristin Lauter, and Michael
