William Jang

Personal Details

Email: wjang20@amherst.edu
Phone: 715-379-6650
Mailing Address:
16 Barrett Hill Drive,
AC # 1486 Keefe Campus Center,
Amherst, MA 01002
Degree: Bachelor of Arts in Mathematics and Computer Science
Expected Graduation: May 2020
Gender identity: Male
Country of citizenship: USA
URM? No

Funding

I currently have no external funding sources. In terms of estimating travel costs, I would need a bus ticket to and from New York from Amherst (100.00 full trip). Transportation within the city would be a combination of ride share and metro for around 30.00.

Research Advisor: Professor Scott Alfeld
School: Amherst College
Email: salfeld@amherst.edu
Phone: 4135425421
Dept: Computer Science

Do you have any work that you submitted and/or was accepted to AAAI-20? No

Keywords:
Machine Learning
Applications of AI
Information Extraction

Reconstructing Training Sets by Observing Sequential Updates

William Jang
Amherst College
220 South Pleasant Street
Amherst, Massachusetts 01002
wjang20@amherst.edu

Abstract

Machine learning methods are being used in an increasing number of settings where learners operate on confidential data. This underscores the need to investigate machine learning methods for vulnerabilities that may reveal information about the training set to a persistent attacker. We consider the task of reverse engineering a training set by watching how a learner responds to additional training data. Specifically, an adversary Alice observes a model learned by Bob on some original training set. Bob then collects more data, and retrains on the union of the original training set and the new data. Alice observes the new data and Bob's sequence of learned models with the aim of capturing information about the original training set. Previous work has addressed issues of data privacy, specifically in terms of theoretical limits to the amount of information leaked by publishing a model. Our contribution concerns the novel setting of when Alice observes a sequence of learned models (and the additional training data that induces this sequence), allowing her to perform a differencing attack. The successful completion of this line of work will yield a better understanding of the privacy guarantees of learners in real world settings where attacker and learner act in time.

Introduction

Using machine learning methods in practice introduces security vulnerabilities. An attacker may manipulate data so as to trick a learned model or a learner in process of training. Such is the study of adversarial learning (Lowd and Meek 2006; Vorobeychik and Kantarcioglu 2018; Joseph et al. 2019; Biggio and Roli 2018) In addition, in deploying a learned model, one may inadvertently reveal information about the training data used. The aim of privacy-preserving learning (Dwork et al. 2010) is to create learning methods with guaranteed limits on the amount of information revealed about the underlying data. Often in practice a learned model is deployed and then later (after additional training data has been gathered), a new model trained on the union of the old and new data is deployed. In this work we seek to quantify how much information about a training set can be gained by an attacker which observes not only the deployed model, but how that model evolves over time as new training data is introduced.

We consider the setting where a learner Bob uses a training set $D = (X, Y)$ to learn a model and an attacker Alice attempts to reverse engineer aspects of D. There is a rich collection of prior work in data privacy, in particular differential privacy (Dwork et al. 2006) which addresses this problem. In contrast to prior work, we model Alice as observing not only D, but also a sequence of new points and subsequently learned models. Formally, Bob learns θ_1 from D with learning algorithm L: $\theta_1 = L(D)$. He then gathers new data D' and learns a new model θ_2 from $D \cup D'$: $\theta_2 = L(D \cup D')$. Alice observes θ_1, D', and θ_2. This continues with Alice observing additional data sets, Bob training a model on the increasing set of points, and Alice observing his model. She attempts to reverse engineer some aspect of the original D (e.g., the mean of a particular feature, whether or not some specific instance is present in the training set, etc.). Our preliminary results show that this sequential observation process results in Alice having substantially more capability to reverse engineer the training set than if she had only observed the first model.

Methods

As an illustrative example, suppose Bob trains a linear regression model using ordinary least squares and Alice aims to reverse engineer the entire training set. That is, Bob learns a model θ_1 which satisfies the normal equations $(X^\top X)\theta_1 = X^\top Y$. We further assume that Alice simply observes, and has no control over the additional points added sequentially to the training process. Consider the toy example when the training set consists of a single point in two dimensions.

Alice knows the normal equation for linear regression: $A_1 B_1 = \theta_1$, where $A_1 = X_1^\top X_1$, $B_1 = X_1^\top y_1$, and θ_1 is the resulting model. She then observes an update to the training set, (x_2, y_2) which results in a new model, θ_2. Alice can set up the normal equations to find $A_2 B_2 = \theta_2$, where A_2 is the Gramian matrix of the training set with the additional point. Note that $A_2 B_2$ is a 2 x 2 matrix which means this equation of matrices yields 2 polynomial equations of degree 4. At this point, Alice has 2 equations and 3 unknowns, so the

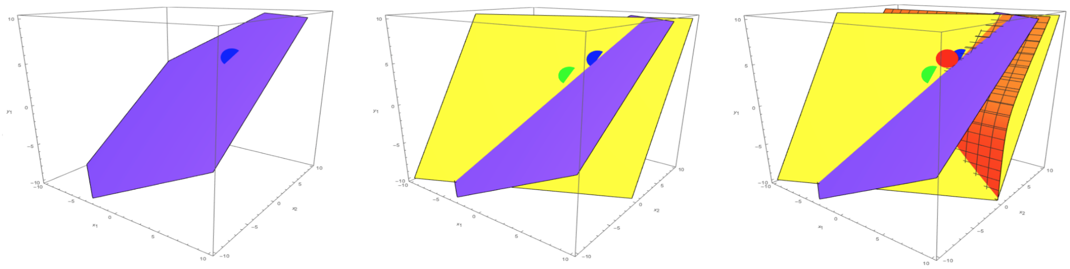

Figure 1: The first dot represents the initial training set, and the planes represent the information that Alice gets by observing each update. In the second graph, Alice knows the training set up to some equivalence class which is the line at the intersection of the two planes. In the third diagram, Alice has three planes that intersect at the original training set.

system of equations is still underdetermined. She has to observe a second point to solve the updated normal equation: $A_3B_3 = \theta_3$ for two more polynomial equations. Now that she has 4 equations and 3 unknowns she can solve for the original training set.

The number of updates Eve needs to observe increases as n and d increase. For instance, when we increase n to 2, Eve will now have 6 unknowns to solve for, requiring at least 6 equations. A general algorithm is as follows:

1. Given initial training set (X_1, y_1), number of samples n, dimensions of training set d, initial model θ_1, $n*(d+1)$ unknowns, and counter $i = 2$.
2. Initialize empty list of equations.
3. Append an update point (x_i, y_i) to (X_{i-1}, y_{i-1}) to get (X_i, y_i)
4. Observe new model θ_i.
5. Solve normal equations $(X_i^\top X_i)(X_i^\top y_i) = \theta_i$ for unknowns and add resulting equations to list of equations.
6. If system is still undefined, increment i and go back to 3.
7. Solve system of polynomial equations for (X_1, y_1).

Separately from exact analytically solving, we consider using a machine learner for Alice's task. Namely, given many examples (which can be generated synthetically) of how Bob's learned model changes given various D, D', we train a model to predict the original training set. Preliminary results using Artificial Neural Networks indicate that approximate inference is possible with this strategy.

Next Steps

Next steps include investigating the task of reverse engineering a training set from an information theoretic perspective. Namely, when Alive observes θ_1 there is an equivalence class of training sets that would have yielded that model. As Alice observes additional training points and the corresponding (updated) models, this equivalence class shrinks. In this way, the additional points and models are communicating information about the training set. A natural question we intend to explore is: how much information is communicated by each additional (set of) point(s) and model?

Separately, we intend to explore more sophisticated learners and attacker goals. For example, if a learned Artificial Neural Network (ANN) used for image classification in a unmanned aerial vehicle is captured by enemy forces, they may seek to find out whether or not a particular collection of images was used to train that ANN. Our work specifically considers the scenario where the enemy observes multiple learned models as they are updated over time with additional training.

Conclusion

We investigate the task of reverse engineering aspects of a training set by observing a series of models, each updated by the addition of training points. We approach this task along two trajectories: analytic computation and automated learning from data. Along the first trajectory we find that one can reverse engineer the training set used by a linear regression learning by solving a system of polynomial equations. The practicality of solving this system decreased rapidly with the size and dimension of the training set. Along the second trajectory, we deploy ANNs to predict the training set given a sequence of models and training points. Preliminary results show promise, but the architecture of the neural network has not yet been dialed in.

References

Biggio, B., and Roli, F. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* 84:317 – 331.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284. Springer.

Dwork, C.; Naor, M.; Pitassi, T.; and Rothblum, G. N. 2010. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, 715–724. ACM.

Joseph, A.; Nelson, B.; Rubinstein, B.; and Tygar, J. 2019. *Adversarial Machine Learning*. Cambridge University Press.

Lowd, D., and Meek, C. 2006. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 641–647. ACM.

Vorobeychik, Y., and Kantarcioglu, M. 2018. *Adversarial Machine Learning*. Morgan Claypool PUblishers.

WILLIAM JANG

715-379-6650 // wjang20@amherst.edu
https://github.com/billyjang

EDUCATION

Amherst College // Amherst, MA — Expected graduation May 2020
Bachelor of Arts in Mathematics and Computer Science — Overall GPA: 3.76/4.0, Major GPA: 3.83/4.0

PUBLICATIONS

Enabling Multiuser Access Control on Smart Home Devices — October 2017
William Jang, Adil Chhabra, Aarathi Prasad
In Proceedings in Internet of Things Security and Privacy Workshop 2017 at ACM CCS 2017

TECHNICAL STRENGTHS

Computer Languages/Frameworks — Java, Python, R, C, SQL, x86, Bash, Swift, HTML/CSS/JS
Tools — Android Studio, XCode, LaTeX, Git, Docker, MS Azure, Google Compute Engine

EXPERIENCE

Machine Learning Thesis Student // Amherst College // Amherst, MA — Aug 2019 - Present

- Design and run experiments using **NumPy, Pandas, Keras, TensorFlow, Google Compute Engine, and Scikit-Learn** examining online gradient descent of various learners in the field of adversarial learning and machine teaching with Dr. Scott Alfeld.
- Explore how various mathematical properties of learners can be exploited by observing online learning.

Civic Data Science NSF REU // Georgia Tech // Atlanta, GA — May 2019 - Sep 2019

- **Merged 8 siloed datasets** containing over 14 million records for an open government data initiative project with the City of Albany, Esri, and Dr. Omar Asensio.
- Visualized spatial housing data using ArcGIS and QGIS.
- **Pulled data using Google Maps and Attom Data API and preprocessed data to allow for merging**.
- Ran various casual inference techniques such as propensity score matching and a genetic algorithm using Python and R to find reference groups.
- Hosted the merged database on MS Azure and sent code and documentation to the City of Albany and Esri for further development and analysis.

Security of Smart Things NSF REU // Florida International University // Miami, FL — May 2018 - Nov 2018

- Implemented a Wearable-Assisted Continuous-Authentication (WACA) framework on the **Apple Watch using Swift, C, Flask, and Xcode** under the direction of Dr. Selcuk Uluagac.
- Collected data, extracted features from the accelerometer and gyroscope, utilized homomorphic encryption, and calculated distance measures.
- Evaluated partially homomorphic encryption schemes within the WACA framework on memory and CPU usage.

Teaching Assistant and Grader // Amherst College // Amherst, MA — January 2017 - Present

- Provided in-lab help, office hours, and lectures for **Introduction to Computer Science I and II, Data Structures, and Artificial Intelligence** courses.
- Automated grading process for myself and future TAs for these classes through shell and Python scripts, Java unit testing and documentation.

PROJECTS

Center for International Student Engagement Database // Amherst, MA — September 2019 - Present

- Design and develop a web app and database for a class project using **HTML, CSS, Bootstrap, JQuery, JS, Flask, Heroku, PostgreSQL** that helps staff at the Center for International Student Engagement keep track of visas and work authorizations for international students.
- Send automated email reminders based upon custom SQL queries using Google Suite API.

Electronic Club Interactive Light Display // Amherst, MA — August 2018 - Present

- **Led team of coders to create an interactive light display** in our science center. The light display uses time of flight distance to present patterns on the lights which are operated by a microcontroller.
- Developed code for an interactive wave motion display that converts digital input from an ultrasonic sensor and microphone to an analog output that controls two wave generators.

Amherst Hackathon Organizer // Amherst, MA — September 2017 - April 2018

- Co-organized the **first Hackathon at Amherst by acquiring $15,000.00 in funding from various sources**, reaching out to schools across the nation, and creating a website.

I'm interested in attending the UC and AAAI because I recognize the importance of collaboration and presence in the research community. Being able to attend the UC and meet other undergraduates, learn what they are interested in and what they've been working on is valuable to me. The past three summers I've been fortunate enough to work on different research projects, and each experience has cultivated positive, strong connections with my peers that I maintain. For instance, this past summer I attended Machine Learning in Science and Engineering (MLSE) and also the Women in Data Science Workshop (WDSW) at Georgia Tech. I got to talk to a lot of presenters and ask them about their motivations for their research, what they were interested in pursuing next, and where they thought the future of their field was headed. It was really conducive to spurring on my own thoughts for potential projects. I also just got to spend a couple days surrounded by bright, kind people.

I remember presenting at Internet of Things Security and Privacy at ACM CCS 2017. I was really nervous, ended up speaking too closely to the mic, and apparently my voice came out really garbled and hard to understand. But after the talk, a lot of people came up to me and asked me about my research and assured me that it wasn't that hard to understand me. I later found out that one of my future research advisors took my paper from that presentation as inspiration for a larger project. I think it's a really unique opportunity to be able to have that kind of voice and influence as an undergrad, which is why I think attending and presenting at the UC would be very special. It'd allow me to get feedback on my own research, share my ideas, find potential collaborators, or just brainstorm together. I think the implications and potential next steps for my research is something that people would be interested in hearing about and thinking about themselves.

I would also meet my peers who may also be interested in going to grad school for AI/ML. I've been interested in becoming a professor for the majority of my life. Ultimately, I believe I would have a positive impact on my peers at the UC at AAAI, and that attending and participating would be a huge opportunity to make connections and learn from my peers.