Why Privacy Matters

Threat Models for Non-private ML

David Madras

May 23, 2019

African Institute for Mathematical Sciences

Rwanda, Kigali

Making Privacy Concrete

What is a Threat Model?

- Wikipedia: "**Threat modeling** is a process by which potential threats can be identified … all from a **hypothetical attacker's point of view**."

- Best way to talk about the **security** of our model is to specify a threat model – how might we be **vulnerable**?

- In this talk, I'll try to convince you **privacy is a real *security* threat** by presenting a concrete threat model

Let's Talk About *Model Inversion!*

- A trained ML model with parameters **w** is released to the public
  - **W** = training_procedure(X)
  - Training data X is hidden

- Can we *recover* some of X just through access to **w**?
  - X' = training_procedure^{-1}(**w**) <--- notational abuse
  - That would be bad

- Intersection of security and privacy

What Model Inversion Looks Like

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

Two Examples We'll Discuss

- "The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets", Carlini et al., 2018

- "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures", Fredrikson et al., 2015

Example 1. The Secret Sharer (Carlini et al.)

- Step 1: Find some training text containing sensitive information (e.g. credit card numbers)
  - "My credit card number is 3141-9265-3587-4001" $\in$ Training text
- Step 2: Train your language model without really thinking too hard
  - State-of-the-art log-likelihood!
- Step 3: Profit … for hackers
  - Sounds bad

Extracting Secrets

- “My credit card number is X” ∈ Training text
- Hacker is given black-box access to the model
- Prompt the model with ‘My credit card number is’ and generate!
  - **$$$$$$$$$$**

- Note: This isn’t exactly how they do it in the paper
  - Many annoying details in implementation

This Attack Kind of Works

		Number of Unique Phrases				
		1	10	50	100	500
# Insertions	1	80%	11%	2%	1%	0.1%
	2	100%	38%	18%	16%	1%
	5	100%	100%	100%	100%	98%
	10	100%	100%	100%	100%	100%

Table 4: Expected percentage of phrases that are uniquely extractable. Each inserted secret has the same format.

This Attack Kind of Works (Part II)

User	Secret Type	Exposure	Extracted?
A	CCN	52	✓
B	SSN	13	
C	SSN	16	
	SSN	10	
	SSN	22	
D	SSN	32	✓
F	SSN	13	
G	CCN	36	
	CCN	29	
	CCN	48	✓

Table 5: Summary of results on the Enron email dataset. Three secrets are extractable in under an hour; all are heavily memorized.

How to Defend?

- Maybe regularization?
  - Memorization relates to generalization
  - Authors try weight decay, dropout, and quantization – none work
  - The problem seems distinct from overfitting

- Maybe sanitization?
  - This makes sense: if you know what the secret looks like, just remove it before training
  - But you may not know all possible secret formats – this is heuristic

How to Defend?

- Differential Privacy!
  - Each token in the training text = “a record in the database”

	Optimizer	ε	Testing Loss	Estimated Exposure
With DP	RMSProp	0.65	1.69	1.1
	RMSProp	1.21	1.59	2.3
	RMSProp	5.26	1.41	1.8
	RMSProp	89	1.34	2.1
	RMSProp	2×10^8	1.32	3.2
	RMSProp	1×10^9	1.26	2.8
	SGD	∞	2.11	3.6
No DP	SGD	N/A	1.86	9.5
	RMSProp	N/A	1.17	31.0

Example 2: Targeted Model Inversion in Classifiers (Fredrikson et al.)

- Step 1. Train classifier parameters **w** on some secret dataset X
- Step 2. Release **w** to the public (white-box)
- Step 3. A hacker can recover parts of your training set by targeting specific individuals
  - That would be bad

Attacking a CNN

- Target: specific output (e.g. person's identity, sensitive feature)
- Start with some random input vector
- Use gradient descent in *input space* to maximize model's confidence in the target prediction

Attacking a Decision Tree using Auxiliary Information

- Given a trained decision tree where we know person X was in the the training set
- Assume we know $x_2 \ldots x_d$ for X, and want to find the value of x_1 (sensitive)

The following estimator characterizes the probability that $\mathbf{x}_1 = v$ given that $\mathbf{x}$ traverses one of the paths $s_1, \ldots, s_m$ and $\mathbf{x}_K = \mathbf{v}_K$:

$$\begin{aligned}
&\Pr\left[\, \mathbf{x}_1 = v \mid (s_1 \vee \cdots \vee s_m) \wedge \mathbf{x}_K = \mathbf{v}_K \,\right] \\
&\quad \propto \sum_{i=1}^{m} \frac{p_i \phi_i(v) \cdot \Pr\left[\, \mathbf{x}_K = \mathbf{v}_K \,\right] \cdot \Pr\left[\, \mathbf{x}_1 = v \,\right]}{\sum_{j=1}^{m} p_j \phi_j(v)} \\
&\quad \propto \frac{1}{\sum_{j=1}^{m} p_j \phi_j(v)} \sum_{1 \le i \le m} p_i \phi_i(v) \cdot \Pr\left[\, \mathbf{x}_1 = v \,\right] \qquad (1)
\end{aligned}$$

Decision Tree Experiments

- Trying to uncover the values of sensitive answers like:

risk-taking behaviors [17]. To support the analysis, FiveThirtyEight commissioned a survey of 553 individuals from SurveyMonkey, which collected responses to questions such as: "Do you ever smoke cigarettes?", "Have you ever cheated on your significant other?", and of course, "How do you like your steak prepared?". Demographic characteristics such as

and 11 variables, including basic demographic information and responses to questions such as, "How happy are you in your marriage?" and "Have you watched X-rated movies in the last year?" We discarded rows that did not contain re-

Decision Tree Results

	FiveThirtyEight			GSS		
algorithm	**acc.**	**prec.**	**rec.**	**acc.**	**prec.**	**rec.**
whitebox	86.4	100.0	21.1	80.3	100.0	0.7
blackbox	85.8	85.7	21.1	80.0	38.8	1.0
random	50.0	50.0	50.0	50.0	50.0	50.0
baseline	82.9	0.0	0.0	82.0	0.0	0.0
ideal	99.8	100.0	98.6	80.3	61.5	2.3

Figure 4: MI results for for BigML models. All numbers shown are percentages.

Defending Against Model Inversion

- Decision trees: split on sensitive features lower down
- CNNs: no concrete suggestions
  - But this paper came out before DP-SGD
- I think differential privacy would protect against both these attacks
- As always, consider tradeoffs with dataset size and accuracy

Conclusion

- If your software works, great! ☺
- If your software works but can be hacked –
  - Then your software doesn't work! ☹
- Hopefully, this presentation convinced you that privacy is a realistic security issue by providing a concrete threat model
- Not everyone needs to think about privacy all the time
  - But some people need to think about it some of the time
  - Or bad things will happen! ☺ ☺ ☺

The End