Measuring the Leakage of a Black-box using Machine Learning

Giovanni Cherubin
@gchers

Alan Turing Institute, London
6 July, 2018

“F-BLEAU: Practical Channel Leakage Estimation” (G. Cherubin, K. Chatzikokolakis, C. Palamidessi, 2018) [Under submission]

A black-box - Problem setting

- Frequentist approach
- Machine Learning approach
- Measuring the leakage: $\varepsilon$-security
- Working in a feature space: $(\varepsilon, \Phi)$-security
- Conclusions
A black-box
A black-box

- o sampled according to a density function $f_s$
A black-box

- \( o \) sampled according to a density function \( f_s \)
- \( s \) sampled according to set of priors \( \pi \)
A black-box

- $o$ sampled according to a density function $f_s$
- $s$ sampled according to set of priors $\pi$
- (Informal) questions:
A black-box

- \( o \) sampled according to a density function \( f_s \)
- \( s \) sampled according to set of priors \( \pi \)

(Informal) questions:
- how much does \( o \) leak about \( s \)?
A black-box

- **o** sampled according to a density function \( f_s \)
- **s** sampled according to set of priors \( \pi \)
- (Informal) questions:
  - how much does \( o \) leak about \( s \)?
  - how hard is it to predict \( s \) given \( o \)?
A black-box

S → B → O

Adversary
A black-box

Training phase:
A black-box

Training phase: \((s_1, o_1)\)
A black-box

Training phase: \((s_1, o_1), (s_2, o_2)\)
A black-box

Training phase: \((s_1, o_1), (s_2, o_2), \ldots\)
A black-box

Training phase: \((s_1, o_1), (s_2, o_2), \ldots, (s_n, o_n)\)
A black-box

Training phase: \((s_1, o_1), (s_2, o_2), \ldots, (s_n, o_n)\)

Attack phase:
A black-box

Training phase: \((s_1, o_1), (s_2, o_2), \ldots, (s_n, o_n)\)

Attack phase: \((s, o)\)
A black-box

Training phase: \((s_1, o_1), (s_2, o_2), \ldots, (s_n, o_n)\)

Attack phase: \((s, o)\)

\[ R = \Pr[s \neq \hat{s}] \]
Attacks
Website Fingerprinting

[²B+’01]
Attacks

Website Fingerprinting

[Reference: B+’01]
Attacks
Website Fingerprinting

[B+’01]
Attacks
Language in VoIP Traffic

[W+’07]

Blah, blah, …
Attacks
Language in VoIP Traffic

Blah, blah, …

Adversary
Attacks
Language in VoIP Traffic

W+’07

B

→

O

Adversary
Attacks

Language in VoIP Traffic

[W+’07]
Attacks
Membership Inference

\[ f(x) \text{ trained on private data: } (X_{\text{train}}, Y_{\text{train}}) \]
Attacks
Membership Inference  [S+’17]

\[ f(x) \text{ trained on private data: } (X_{\text{train}}, Y_{\text{train}}) \]
Adversary Attacks
Membership Inference [S+’17]

How confident are you on your prediction for “x”?

\[ f(x) \] trained on private data: \((X_{\text{train}}, Y_{\text{train}})\)
Adversary Attacks
Membership Inference [S+’17]

How confident are you on your prediction for “x”

\[ f(x) \text{ trained on private data: } (X_{\text{train}}, Y_{\text{train}}) \]
Adversary Attacks
Membership Inference [S+’17]

How confident are you on your prediction for “x”?

\[ S : \text{“x” in training data} \quad \rightarrow \quad f(x) \]

\[ f(x) \text{ trained on private data: } (X_{\text{train}}, Y_{\text{train}}) \]

96%

Adversary
A black-box

S → B → O

Adversary
A black-box
A black-box

Adversary
A black-box

Adversary

R*: Bayes Error
A black-box

\[ R \geq R^* \]

R*: Bayes Error

Adversary
What’s next
What’s next

Estimate $R^*$ (or bounds) given a dataset: $(s_1, o_1), \ldots, (s_n, o_n)$ with smallest (realistic) number of examples.
A black-box - Problem setting

Frequentist approach

Machine Learning approach

Measuring the leakage: \( \varepsilon \)-security

Working in a feature space: \((\varepsilon, \Phi)\)-security

Conclusions
Estimating R*
Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}
Estimating $R^*$
Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Estimating $R^*$

Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Estimating $R^*$

Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}
Estimating $R^*$

Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}
Estimating R*

Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}

<table>
<thead>
<tr>
<th>o = 0.3</th>
<th>S_1</th>
<th>S_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>o = 1.2</td>
<td>S_1</td>
<td>S_2</td>
</tr>
</tbody>
</table>
Estimating R*
Frequentist Approach

Suppose secrets can take one of 3 values: \(\{s_1, s_2, s_3\}\)

<table>
<thead>
<tr>
<th>Prob</th>
<th>Secret 1</th>
<th>Secret 2</th>
<th>Secret 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>S_1</td>
<td>S_2</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>S_1</td>
<td></td>
<td>S_3</td>
</tr>
<tr>
<td>1.2</td>
<td>S_1</td>
<td>S_2</td>
<td>S_3</td>
</tr>
</tbody>
</table>
Estimating $R^*$
Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}
Estimating $R^*$
Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}

<table>
<thead>
<tr>
<th>$o = 0.3$</th>
<th>$S_1$</th>
<th>$S_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o = 1.2$</td>
<td>$S_1$</td>
<td>$S_2$</td>
</tr>
<tr>
<td>$o = 0.7$</td>
<td>$S_3$</td>
<td></td>
</tr>
</tbody>
</table>

Problems:
Estimating $R^*$
Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}

<table>
<thead>
<tr>
<th>$o = 0.3$</th>
<th>S_1</th>
<th>S_2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$o = 1.2$</td>
<td>S_1</td>
<td>S_2</td>
<td>S_3</td>
</tr>
<tr>
<td>$o = 0.7$</td>
<td>S_3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Problems:

- If space $S$ or $O$ large (or infinite) needs too many examples
Estimating $R^*$
Frequentist Approach

Suppose secrets can take one of 3 values: \{s_1, s_2, s_3\}

<table>
<thead>
<tr>
<th>$o = 0.3$</th>
<th>S_1</th>
<th>S_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o = 1.2$</td>
<td>S_1</td>
<td>S_2</td>
</tr>
<tr>
<td>$o = 0.7$</td>
<td>S_3</td>
<td></td>
</tr>
</tbody>
</table>

Problems:

- If space $S$ or $O$ large (or infinite) needs too many examples
- Strongly affected by noise in data
A black-box - Problem setting

Frequentist approach

**Machine Learning approach**

Measuring the leakage: $\varepsilon$-security

Working in a feature space: $(\varepsilon, \Phi)$-security

Conclusions
Machine Learning (Classification)
Machine Learning
(Classification)

Classification problem:

Training data:  \((s_1, o_1), \ldots, (s_n, o_n)\)
Machine Learning
(Classification)

Classification problem:

Training data: \((s_1, o_1), \ldots, (s_n, o_n)\)

Test object: \((s, o)\)
Machine Learning
(Classification)

Classification problem:

Training data: \((s_1, o_1), ..., (s_n, o_n)\)

Test object: \((s, o)\)

Learning rule \(A\): selects a classifier \(f: O \rightarrow S\) that minimises:
Machine Learning (Classification)

Classification problem:

Training data: \((s_1, o_1), \ldots, (s_n, o_n)\)

Test object: \((s, o)\)

Learning rule \(A\): selects a classifier \(f : O \rightarrow S\) that minimises:

\[ R_f = \Pr[s \neq f(o)] = \Pr[s \neq \hat{s}] \]
Impossibility Results

No convergence rates nor free lunches
Impossibility Results

No convergence rates nor free lunches

**Theorem (no convergence rate)** If $O = \mathbb{R}$, no Bayes error estimate can guarantee to converge with a certain rate (w.r.t. size of training data $n$).
Impossibility Results [A+’99, W’02]
No convergence rates nor free lunches

**Theorem (no convergence rate)**  If $O = \mathbb{R}$, no Bayes error estimate can guarantee to converge with a certain rate (w.r.t. size of training data $n$).

**Theorem (~NFL)**  No rule is “optimal” among all the possible learning problems (distributions).
Estimates

NN Bound

[CH’67]
Estimates
NN Bound
Estimates
NN Bound

[CH’67]
Estimates

NN Bound

\[ R^* \leq R \]
Estimates
NN Bound

\[ \frac{L - 1}{L} \left( 1 - \sqrt{1 - \frac{L}{L-1} R^{NN}} \right) \leq R^* \leq R \]
Estimates

$k_n$-NN

[S’77]
Estimates
kn-NN
Estimates

$k_n$-NN

[S’77]
Estimates

**Theorem** If $k_n/n \to 0$ and $k_n \to \infty$ as $n \to \infty$, then:

$R_{k_n-\text{NN}}$ converges to $R^*$

[S’77]
Theorem  If $k_n / n \to 0$ and $k_n \to \infty$ as $n \to \infty$, then:

$R_{k_n-NN}$ converges to $R^*$

$R \geq R_{k_n-NN} \approx R^*$

[Estimates: $k_n$-NN]
Theorem If \( k_n/n \to 0 \) and \( k_n \to \infty \) as \( n \to \infty \), then:

\[
R^{k_n-NN} \quad \text{converges to } R^*
\]

Remark: any Universally Consistent rule (e.g., SVM) gives this guarantee as the size of the training data \( n \to \infty \).
Frequentist VS NN-based

\[ P(o \mid s) = \lambda \exp \left( -\nu \mid g(s) - o \mid \right) \]
Frequentist VS NN-based

\[ P(o \mid s) = \lambda \exp \left( -\nu \mid g(s) - o \mid \right) \]
Frequentist VS NN-based

\[ P(o \mid s) = \lambda \exp (-\nu \mid g(s) - o \mid) \]
Frequentist VS NN-based

\[
P(o \mid s) = \lambda \exp (-\nu \mid g(s) - o \mid)
\]
- A black-box - Problem setting
- Frequentist approach
- Machine Learning approach

**Measuring the leakage: \( \varepsilon \)-security**

- Working in a feature space: \((\varepsilon, \Phi)\)-security

- Conclusions
A leakage measure

Problem  An error estimate \( R^* \) alone does not convey information about the setting. Random guessing \( R^G \):
A leakage measure

**Problem** An error estimate $R^*$ alone does not convey information about the setting. Random guessing $R^G$:

$$P(s_1) = P(s_2) = 0.5$$
A leakage measure

Problem  An error estimate $R^*$ alone does not convey information about the setting. Random guessing $R^G$:

$$P(s_1) = P(s_2) = 0.5$$

$$R^G = 1/2$$
Problem  An error estimate $R^*$ alone does not convey information about the setting. Random guessing $R^G$:

$$P(s_1) = P(s_2) = 0.5$$

$$P(s_1) > P(s_2)$$

$$R^G = 1/2$$
A leakage measure

**Problem** An error estimate $R^*$ alone does not convey information about the setting. Random guessing $R^G$:

- $P(s_1) = P(s_2) = 0.5$
- $P(s_1) > P(s_2)$
- $R^G = 1/2$
- $R^G < 1/2$
A leakage measure

**Problem** An error estimate $R^*$ alone does not convey information about the setting. Random guessing $R^G$:

$$P(s_1) = P(s_2) = 0.5$$

$$P(s_1) > P(s_2)$$

$$R^G = 1/2$$

$$R^G < 1/2$$

Define leakage measure ("$\epsilon$-security"):

$$\epsilon = \frac{\hat{R}^*}{R^G}$$
Properties of $\varepsilon$
Properties of $\varepsilon$

- $\varepsilon \in [0,1]$
Properties of $\varepsilon$

- $\varepsilon \in [0, 1]$

- $\varepsilon = 1$ is perfect security: must random guess
Properties of $\varepsilon$

- $\varepsilon \in [0,1]$
- $\varepsilon = 1$ is perfect security: must random guess
- Corresponds to $1 - \text{Adv}$
Properties of $\varepsilon$

- $\varepsilon \in [0,1]$
- $\varepsilon = 1$ is perfect security: must random guess
- Corresponds to $1 - \text{Adv}$
- “Miracle theorem” [B+’09]: $\varepsilon_{\text{Uni}} \leq \varepsilon_{\pi}$
A black-box - Problem setting

Frequentist approach

Machine Learning approach

Measuring the leakage: $\varepsilon$-security

Working in a feature space: $(\varepsilon, \Phi)$-security

Conclusions
Learning in Feature Space
$(\varepsilon, \Phi)$-security
Learning in Feature Space

$(\varepsilon, \Phi)$-security

Because of NFL, convergence in finite sample may be too slow
Learning in Feature Space

$(\varepsilon, \Phi)$-security

Because of NFL, convergence in finite sample may be too slow
Learning in Feature Space

$(\varepsilon, \Phi)$-security

Because of NFL, convergence in finite sample may be too slow

Work in feature space $O' = \Phi(O)$:
Learning in Feature Space

$(\varepsilon, \Phi)$-security

Because of NFL, convergence in finite sample may be too slow

Work in feature space $O' = \Phi(O)$:

$\Phi(\overrightarrow{\text{[ ] [ ] [ ] [ ]}}) = (\text{total time, avg packet size, } \ldots)$
Learning in Feature Space
Convergence for WF
A black-box - Problem setting

Frequentist approach

Machine Learning approach

Measuring the leakage: $\varepsilon$-security

Working in a feature space: $(\varepsilon, \Phi)$-security

Conclusions
TL;DL

- **Black-box formulation**: arbitrary systems
- Asymptotic results, **NFL**
- White-box approaches **not always possible**
- **NN approaches**, perform well
- $\varepsilon$: interpretability (advantage w.r.t. random guessing)
- $(\varepsilon, \Phi)$-security: reliance on features (sometimes)
References

• 1967, T. M. Cover and P. E. Hart. “Nearest neighbor pattern classification.”
• 1977, C. J. Stone. “Consistent nonparametric regression.”
• 2002, Wolpert, David H. "The supervised learning no-free-lunch theorems."
• 2007, C. V. Wright, L. Ballard, F. Monrose, G. M. Masson “Language identification of encrypted Voip traffic: Alejandra y Roberto or Alice and Bob?.”
• 2009, C. Braun, K. Chatzikokolakis, and C. Palamidessi. “Quantitative notions of leakage for one-try attacks.”

More at:  https://giocher.com/pages/bayes.html
Measuring the Leakage of a Black-box using Machine Learning

https://giocher.com/pages/bayes.html

Giovanni Cherubin
@gchers

Alan Turing Institute, London
6 July, 2018