Measuring the Security of Blackbox Systems via Machine Learning

Part I - The quest for Machine Learning optimality

Giovanni Cherubin
@gchers

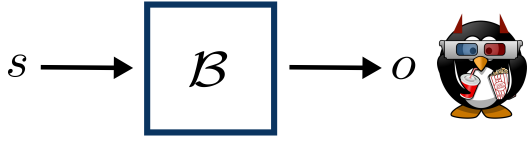
SPRING lab
EPFL

ForMal 2019

EPFL

Spoiler: Our final goal

Spoiler: Our final goal



Tutorial outline

Two parts ($\sim$ 2h, including 10' break):

1. Optimality in ML, asymptotically optimal rules, impossibility results
2. Applying these results for measuring the information leakage of generic systems

Outline

- Quick review of supervised learning
- Optimality: Bayes classifier
- Asymptotic properties of NN and k_n-NN, and universally consistent rules
- Impossibility results
- Features and convergence

Supervised learning

- "Nature" gives us n examples (object-label pairs) $(x_1, y_1), ..., (x_n, y_n)$, sampled independently from the same distribution μ on $\mathcal{X} \times \mathcal{Y}$, finite $\mathcal{Y}$.

Supervised learning

- "Nature" gives us n examples (object-label pairs) $(x_1, y_1), ..., (x_n, y_n)$, sampled independently from the same distribution μ on $\mathcal{X} \times \mathcal{Y}$, finite $\mathcal{Y}$.
- A new object x with label y arrives

Supervised learning

- "Nature" gives us n examples (object-label pairs) $(x_1, y_1), ..., (x_n, y_n)$, sampled independently from the same distribution μ on $\mathcal{X} \times \mathcal{Y}$, finite $\mathcal{Y}$.
- A new object x with label y arrives

Select a function $g(x)$ that minimizes some risk notion.

Supervised learning

- "Nature" gives us n examples (object-label pairs) $(x_1, y_1), ..., (x_n, y_n)$, sampled independently from the same distribution μ on $\mathcal{X} \times \mathcal{Y}$, finite $\mathcal{Y}$.
- A new object x with label y arrives

Select a function $g(x)$ that minimizes some risk notion.

Here we are mostly interested in the classification problem ($\approx$ finite $\mathcal{Y}$). Consider the 0-1 risk (misclassification error):

$$R^g = \mathbb{E}(I(g(x) \neq y)) = P(g(x) \neq y)$$

Supervised learning

- "Nature" gives us n examples (object-label pairs) $(x_1, y_1), ..., (x_n, y_n)$, sampled independently from the same distribution μ on $\mathcal{X} \times \mathcal{Y}$, finite $\mathcal{Y}$.
- A new object x with label y arrives

Select a function $g(x)$ that minimizes some risk notion.

Here we are mostly interested in the classification problem ($\approx$ finite $\mathcal{Y}$). Consider the 0-1 risk (misclassification error):

$$R^g = \mathbb{E}(I(g(x) \neq y)) = P(g(x) \neq y)$$

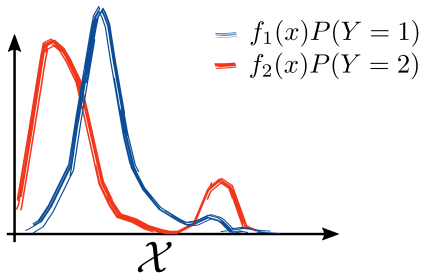
We call **learning rule** a sequence of classifiers $\{g_n, n \geq 1\}$.

The Bayes classifier

Let $\mathcal{Y} = \{1, 2\}$, $\mathcal{X} = \mathbb{R}$. Objects given a label y are generated according to pdf $f_y(x)$.

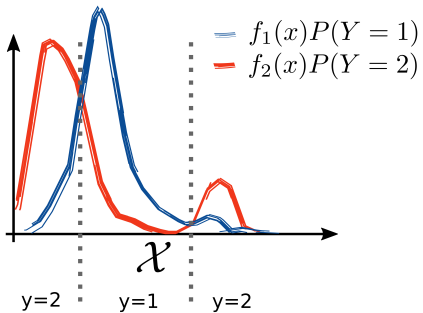
The Bayes classifier

Let $\mathcal{Y} = \{1, 2\}$, $\mathcal{X} = \mathbb{R}$. Objects given a label y are generated according to pdf $f_y(x)$.



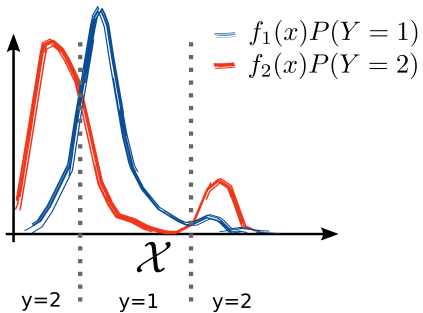
The Bayes classifier

Let $\mathcal{Y} = \{1, 2\}$, $\mathcal{X} = \mathbb{R}$. Objects given a label y are generated according to pdf $f_y(x)$.



The Bayes classifier

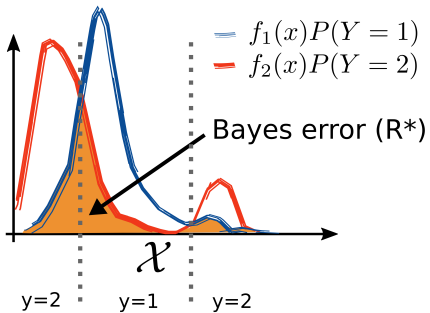
Let $\mathcal{Y} = \{1, 2\}$, $\mathcal{X} = \mathbb{R}$. Objects given a label y are generated according to pdf $f_y(x)$.



$$g^*(x) = \underset{y \in Y}{\operatorname{argmax}} P(y|x) = \underset{y \in Y}{\operatorname{argmax}} P(x|y)P(y)$$

The Bayes classifier

Let $\mathcal{Y} = \{1, 2\}$, $\mathcal{X} = \mathbb{R}$. Objects given a label y are generated according to pdf $f_y(x)$.

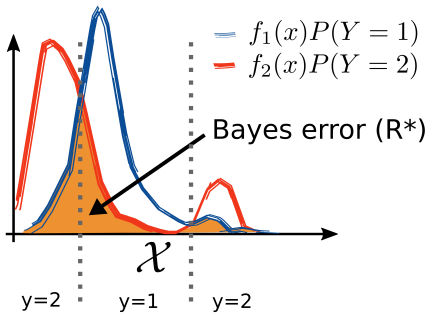


$$g^*(x) = \underset{y \in Y}{\operatorname{argmax}} P(y|x) = \underset{y \in Y}{\operatorname{argmax}} P(x|y)P(y)$$

The Bayes classifier

Remark: **Bayes classifier $\neq$ Naïve Bayes classifier**

Let $\mathcal{Y} = \{1, 2\}$, $\mathcal{X} = \mathbb{R}$. Objects given a label y are generated according to pdf $f_y(x)$.



$$g^*(x) = \underset{y \in Y}{\operatorname{argmax}} P(y|x) = \underset{y \in Y}{\operatorname{argmax}} P(x|y)P(y)$$

Bayes error
An example

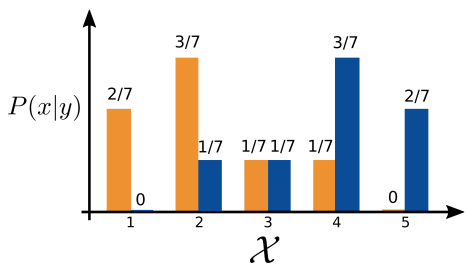
Consider finite space X. The Bayes risk is:

$$R^* = 1 - \sum_{x \in \mathcal{X}} \max P(x|y)P(y)$$

Bayes error
An example

Consider finite space X. The Bayes risk is:

$$R^* = 1 - \sum_{x \in \mathcal{X}} \max P(x|y)P(y)$$

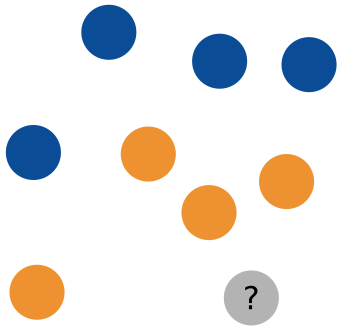


But...

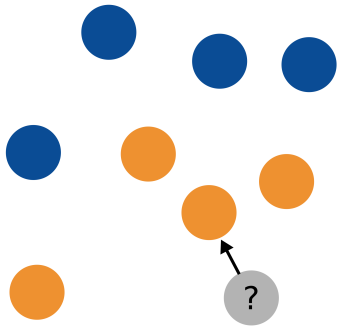
In practice, underlying distribution is unknown $\Longrightarrow$ we can't use Bayes classifier

Nearest neighbour

Nearest neighbour



Nearest neighbour



Nearest neighbour
Asymptotic convergence [CH'67]

Theorem

Let $|\mathcal{Y}| = 2$. Consider a separable metric space $\mathcal{X}$ with (arbitrary) metric d.
As $n \to \infty$, the error of the Nearest Neighbour (NN) classifier is at most twice the Bayes risk:

$$R^{NN} \leq 2R^*(1 - R^*)$$

($\mathcal{X}$ must be separable metric space, with (any) metric d. Minor continuity assumption on density functions omitted.)

Nearest neighbour

Asymptotic convergence - Proof idea

Consider training data $(x_1, y_1), ..., (x_n, y_n)$, and a new object to predict x.
Let x'_n be the closest training object to x.

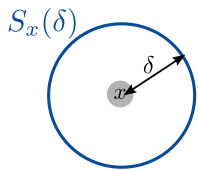
Two steps:

1. (Lemma) Show that $x'_n \to x$ as $n \to \infty$;
2. (Theorem) A posteriori distribution on x'_n converges to that of x. Some calculations for the bound.

Nearest neighbour

Asymptotic convergence - Lemma: $x'_n \to x$

Consider the δ-sphere centered in x:

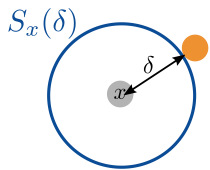


Pf. intuition: Work on x such that $\forall \delta > 0$ every sphere $S_x(\delta)$ has nonzero probability measure. (Indeed, the set of points without this property has probability 0.)

Nearest neighbour

Asymptotic convergence - Lemma: $x'_n \to x$

Consider the δ-sphere centered in x:

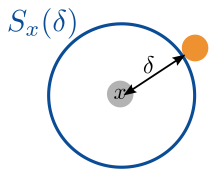


Pf. intuition: Work on x such that $\forall \delta > 0$ every sphere $S_x(\delta)$ has nonzero probability measure. (Indeed, the set of points without this property has probability 0.)

Nearest neighbour

Asymptotic convergence - Lemma: $x'_n \to x$

Consider the δ-sphere centered in x:



Pf. intuition: Work on x such that $\forall \delta > 0$ every sphere $S_x(\delta)$ has nonzero probability measure. (Indeed, the set of points without this property has probability 0.)
As $n \to \infty$, it becomes more and more unlikely that at least one training point is not in the δ-ball, for any chosen $\delta > 0$.

$$P(d(x, x'_n) > \delta) = P(x'_n \notin S_x(\delta)) = (1 - P(S_x(\delta))^n \to 0$$

Nearest neighbour

Asymptotic convergence - Main theorem

Let $\mathcal{X}$ separable metric space. Assume f_1 and f_2 are s.t. w.p 1, x either: i) continuity point of f_1 and f_2, or ii) point of nonzero probability measure.
Then:

$$R^* \leq R^{NN} \leq 2R^*(1 - R^*)$$

Nearest neighbour

Asymptotic convergence - Main theorem

Let $\mathcal{X}$ separable metric space. Assume f_1 and f_2 are s.t. w.p 1, x either: i) continuity point of f_1 and f_2, or ii) point of nonzero probability measure.
Then:

$$R^* \leq R^{NN} \leq 2R^*(1 - R^*)$$

Proof idea:

- $r(x, x'_n) = P(y \neq y'_n \mid x, x'_n) \to 2P(y = 1 \mid x)P(y = 2 \mid x)$

Nearest neighbour

Asymptotic convergence - Main theorem

Let $\mathcal{X}$ separable metric space. Assume f_1 and f_2 are s.t. w.p 1, x either: i) continuity point of f_1 and f_2, or ii) point of nonzero probability measure.
Then:

$$R^* \leq R^{NN} \leq 2R^*(1 - R^*)$$

Proof idea:

- $r(x, x'_n) = P(y \neq y'_n \mid x, x'_n) \to 2P(y = 1 \mid x)P(y = 2 \mid x)$
- $r(x) = 2r^*(x)(1 - r^*(x))$

Nearest neighbour

Asymptotic convergence - Main theorem

Let $\mathcal{X}$ separable metric space. Assume f_1 and f_2 are s.t. w.p 1, x either: i) continuity point of f_1 and f_2, or ii) point of nonzero probability measure.
Then:

$$R^* \leq R^{NN} \leq 2R^*(1 - R^*)$$

Proof idea:

- $r(x, x'_n) = P(y \neq y'_n \mid x, x'_n) \to 2P(y = 1 \mid x)P(y = 2 \mid x)$
- $r(x) = 2r^*(x)(1 - r^*(x))$
- Take expectation, $R^{NN} = \lim_{n\to\infty} \mathbb{E}(r(x, x'_n))$, conclude that $R^{NN} \leq 2R^*(1 - R^*)$

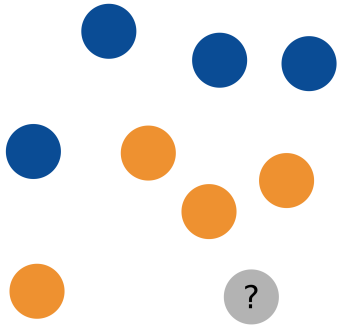
Nearest neighbour

Asymptotic convergence - General bound

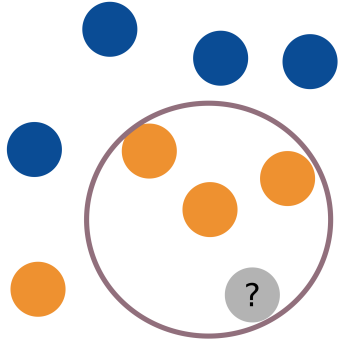
Let $L = |\mathcal{Y}|$. Then:

$$R^{NN} \leq R^* \left(2 - \frac{L}{L-1} R^*\right)$$

k-Nearest Neighbour



k-Nearest Neighbour



k-Nearest Neighbour

Asymptotic optimality [S'77]

Theorem

If we let $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$, then:

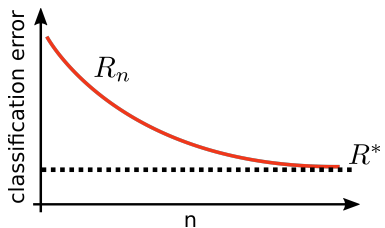
$$R^{k_n\text{-NN}} \to R^*$$

E.g.: $k_n = \log n$ or $k_n = \sqrt{n}$.

Universally consistent rules

Definition

A learning rule $g_n(x)$ is **universally consistent** if for every distribution μ on $\mathcal{X} \times \mathcal{Y}$ its expected error converges to R^* as $n \to \infty$.



Impossibility results

No convergence rate results [A+]

We only have asymptotic results. But can we determine **how** R_n converges to R^*?

Impossibility results

No convergence rate results [A+]

We only have asymptotic results. But can we determine **how** R_n converges to R^*?

Nope

Impossibility results

No convergence rate results [A+]

We only have asymptotic results. But can we determine **how** R_n converges to R^*?

Nope In the general case, for every rule $g_n(x)$ we can find a distribution for which $g_n(x)$ converges arbitrarily slowly.

Impossibility results
No Free Lunch (NFL)

In the finite sample, is there a rule $g_n^+(x)$ which is “often” better than the others?

Impossibility results
No Free Lunch (NFL)

In the finite sample, is there a rule $g_n^+(x)$ which is “often” better than the others?

Nope

Impossibility results
No Free Lunch (NFL)

In the finite sample, is there a rule $g_n^+(x)$ which is "often" better than the others?

Nope For every two classifiers A and B there exists exactly as many distributions for which A is better than B and vice-versa.

Impossibility results
No Free Lunch (NFL)

In the finite sample, is there a rule $g_n^+(x)$ which is "often" better than the others?

Nope For every two classifiers A and B there exists exactly as many distributions for which A is better than B and vice-versa.

Takeaway Try as many learning algorithms as you can.*

Improving convergence

Features!

x : “anything” physically measurable

y : will the car make it to the other side?

Improving convergence

Features!

x : "anything" physically measurable

y : will the car make it to the other side?

Features:

- speed
- acceleration
- bridge status...

Features

Basic results:

- features can improve the finite sample convergence

Features

Basic results:

- features can improve the finite sample convergence
- BUT they cannot improve the asymptotic behaviour (trivially, the Bayes risk based on feature information cannot be smaller than the Bayes risk based on full information).

Summary

- Bayes classifier is optimal, but we generally cannot use it
- NN classifier offers a good approximation (bound)
- Universally Consistent rules (e.g., k_n-NN) approximate the Bayes classifier asymptotically
- There's no free lunch (well, we're having one at 12:00 today, but free lunches are rare)
- Features

What's next We'll use all this beautiful theory to solve real! world! problems!
Spoiler: `https://github.com/gchers/fbleau`

Bibliography

L. Devroye, L. Györfi, G. Lugosi
A probabilistic theory of pattern recognition. Springer, 2013.

C. Stone
Consistent nonparametric regression. JSTOR, 1977

T. Cover, P. Hart,
Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 1967.

D. H. Wolpert,
The supervised learning no-free-lunch theorems. Springer, 2002.

Measuring the Security of Blackbox Systems via Machine Learning

Part II - Measuring security with ML

Giovanni Cherubin
@gchers

SPRING lab
EPFL

ForMal 2019

EPFL

Based on

- G. Cherubin, *Bayes, not naïve: Security bounds on website fingerprinting defenses*, PETS 2017.
- G. Cherubin, K. Chatzikokolakis, C. Palamidessi, *F-BLEAU: Fast Black-box Leakage Estimation*, IEEE S&P 2019.

`https://github.com/gchers/fbleau`

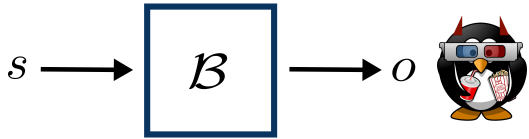
`https://giocher.com/projects/bayes`

Outline

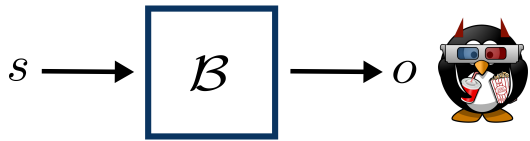
- A black-box
- Application examples
- Current estimation paradigm (frequentist approach)
- Analogy between ML and Black-box security
- Feature-based security notions
- Infinite secret space

A black-box

A black-box

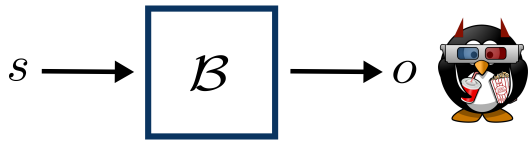


A black-box



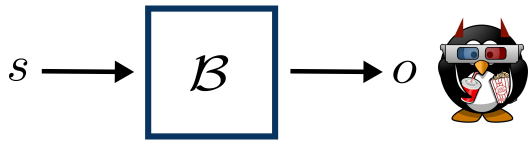
- o sampled according to density function f_s

A black-box



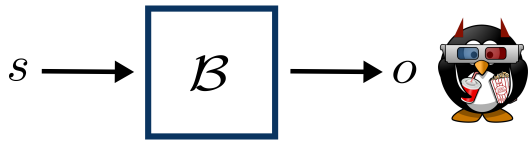
- o sampled according to density function f_s
- s sampled according to priors π

A black-box



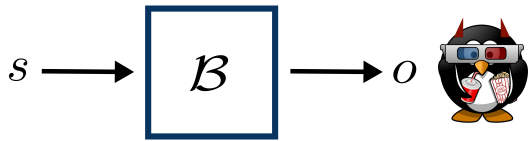
- o sampled according to density function f_s
- s sampled according to priors π
- Informal questions:

A black-box



- o sampled according to density function f_s
- s sampled according to priors π
- Informal questions:
  - how much does o leak about s

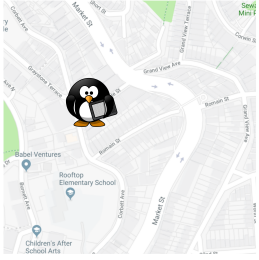
A black-box



- o sampled according to density function f_s
- s sampled according to priors π
- Informal questions:
  - how much does o leak about s
  - how hard is it to predict s given o.

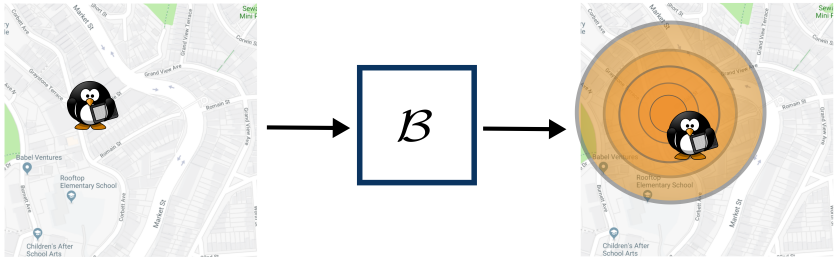
Application Examples

Location privacy



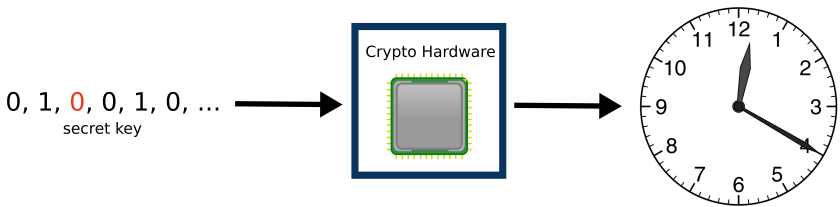
Application Examples

Location privacy



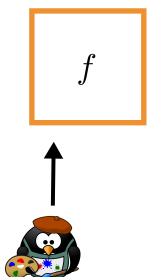
Application Examples

Side channels in crypto primitives' implementation [B+'08]



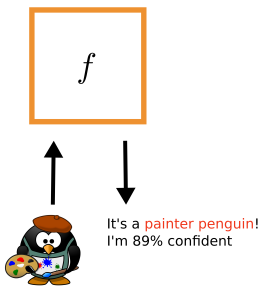
Application Examples
Membership Inference [S+'17]

Consider a classifier $f : \mathcal{X} \mapsto [0,1]^L$ trained on data (X, Y).



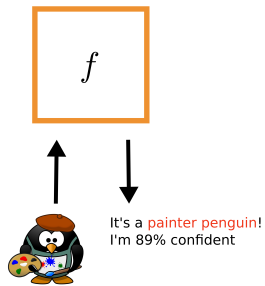
Application Examples
Membership Inference [S+'17]

Consider a classifier $f : \mathcal{X} \mapsto [0,1]^L$ trained on data (X, Y).



Application Examples
Membership Inference [S+'17]

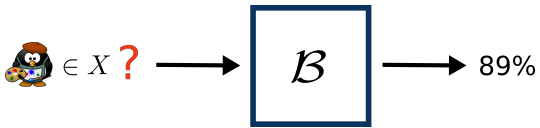
Consider a classifier $f : \mathcal{X} \mapsto [0,1]^L$ trained on data (X, Y).



 wants to guess if the object $x =$  was in the training data (X, Y).

Application Examples

Membership Inference in the Black-box setting



Bayes risk and security measures

$$Pr\left(\text{  }(o) \neq s\right)$$

Bayes risk and security measures

$$R^* = \min_{\text{  }} \{ Pr\left(\text{  }(o) \neq s \right) \}$$

Bayes risk and security measures

Bayes risk R^* is the probability of error of the **optimal** adversary:

$$R^* = \min_{\text{  }} \{ Pr\left(\text{  }(o) \neq s \right) \}$$

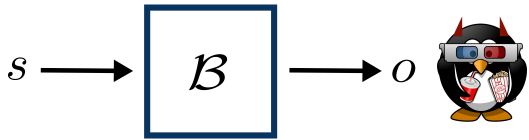
Bayes risk and security measures

Bayes risk R^* is the probability of error of the **optimal** adversary:

$$R^* = \min_{\text{  }} \{ Pr\left(\text{  }(o) \neq s \right) \}$$

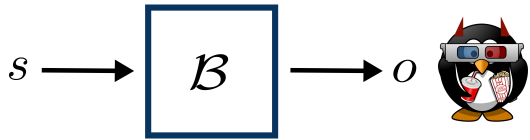
On its basis we can compute **leakage measures** (e.g., Min-entropy, multiplicative/additive leakage). We write $\lambda(R^*)$ for a generic leakage measure.

Black-box estimates



We wish to measure leakage based on queries we make to the system:

Black-box estimates



We wish to measure leakage based on queries we make to the system:

$$(s_1, o_1), (s_2, o_2), \ldots, (s_n, o_n)$$

Desiderata

Leakage estimate should:

- Given enough queries, converge to the true value (**asymptotic optimality**)

Desiderata

Leakage estimate should:

- Given enough queries, converge to the true value (**asymptotic optimality**)
- Require few queries (i.e., it should **converge quickly**)

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(s, 0.4), (s, 0.7), (s, 1.2), (s, 0.4), (s, 0.4)$$

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(s, 0.4), (s, 0.7), (s, 1.2), (s, 0.4), (s, 0.4)$$

$\text{Freq}\left(0.4\right) =$

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(\underline{s, 0.4}), (s, 0.7), (s, 1.2), (\underline{s, 0.4}), (\underline{s, 0.4})$$

$\text{Freq}\left(0.4\right) =$

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(s, 0.4), (s, 0.7), (s, 1.2), (s, 0.4), (s, 0.4)$$

$\text{Freq}\left(0.4\right)$ = Most frequent among $\{s, s, s\}$

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(s, 0.4), (s, 0.7), (s, 1.2), (s, 0.4), (s, 0.4)$$

$\text{Freq}\left(0.4\right)$ = Most frequent among $\{s, s, s\} = s$

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(s, 0.4), (s, 0.7), (s, 1.2), (s, 0.4), (s, 0.4)$$

$\text{Freq}\left(0.4\right) =$ Most frequent among $\{s, s, s\} = s$

$\text{Freq}\left(0.5\right) =$

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(\underline{s, 0.4}), (s, 0.7), (s, 1.2), (\underline{s, 0.4}), (\underline{s, 0.4})$$

$\text{Freq}\left(0.4\right) =$ Most frequent among $\{s, s, s\} = s$

$\text{Freq}\left(0.5\right) =$ Random guessing (e.g., most frequent label overall)

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(\underline{s, 0.4}), (s, 0.7), (s, 1.2), (\underline{s, 0.4}), (\underline{s, 0.4})$$

$\text{Freq}\left(0.4\right)$ = Most frequent among $\{s, s, s\} = s$

$\text{Freq}\left(0.5\right)$ = Random guessing (e.g., most frequent label overall)

- Does not work for **continuous** output space

Frequentist paradigm [C+'10]

(E.g., `leakiEst`, `LeakWatch`)

Suppose adversary observed data:

$$(\underline{s, 0.4}), (s, 0.7), (s, 1.2), (\underline{s, 0.4}), (\underline{s, 0.4})$$

$\text{Freq}\left(0.4\right)$ = Most frequent among $\{s, s, s\} = s$

$\text{Freq}\left(0.5\right)$ = Random guessing (e.g., most frequent label overall)

- Does not work for **continuous** output space
- Does not scale to **large** systems (needs at least one example per output value)

Black-box security and Machine Learning

An equivalence [C'17, C+'19]

Idea! We can use ML techniques (e.g., UC rules/bounds) for estimating the security of a system.

Estimation tools

Method	Guarantee	Space $\mathcal{X}$	Assumptions
Frequentist [C+'10]	$\to R^*$	finite	
k_n-NN	$\to R^*$	infinite	
NN Bound: $\frac{\lvert\mathcal{S}\rvert-1}{\lvert\mathcal{S}\rvert}\left(1-\sqrt{1-\frac{\lvert\mathcal{S}\rvert}{\lvert\mathcal{S}\rvert-1}R^{NN}}\right)$	$\leq R^*$	infinite	$(d,\mathcal{X})$ separable
NN	$\to R^*$	finite	
SVM-RBF	$\to R^*$	infinite	$\mathcal{X}$ compact subset of $\mathbb{R}^m$

(Under certain parameter choices.)

Bayes Error estimates in Practice

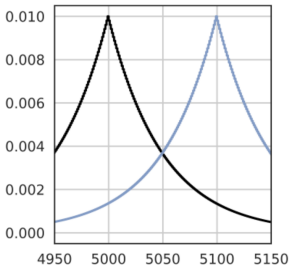
UC rules give guarantees: i) for $n \to \infty$ and ii) w.r.t. their expected error.

In practice, we only have access to a finite number of examples.
Options:

- training + held-out sets
- k-fold CV

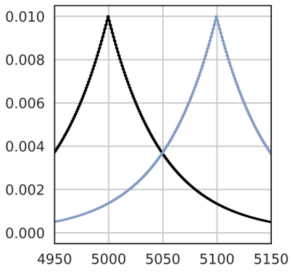
Results

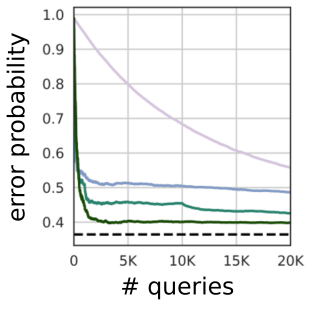
Geometric

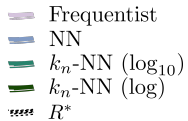


Results

Geometric

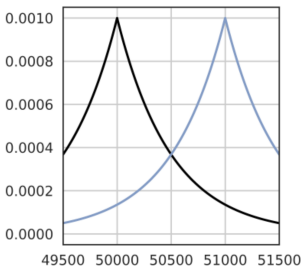






Results

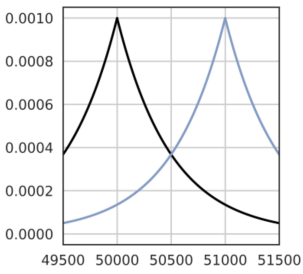
Geometric

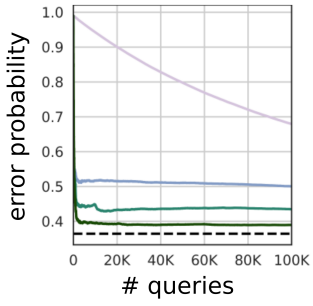


Frequentist
NN
k_n-NN ($\log_{10}$)
k_n-NN (log)
R^*

Results

Geometric



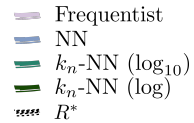


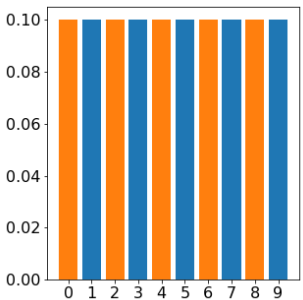
Results

Do NN methods ever fail? The “Spiky” channel

Results

Do NN methods ever fail? The “Spiky” channel

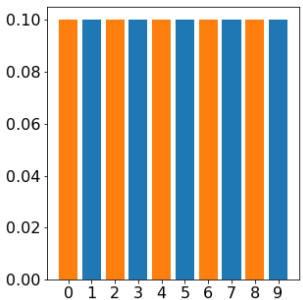


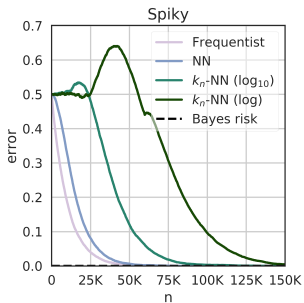


2 secrets, 10K observable values

Results

Do NN methods ever fail? The “Spiky” channel





2 secrets, 10K observable values

Results
Location privacy

Defense mechanisms:

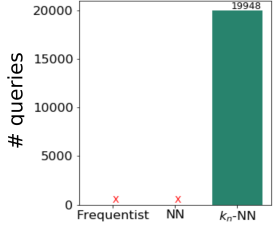
- **Geometric**
- Laplacian
- Blahut-Arimoto [O+'17]

Gowalla dataset: users' location data, $100K$ examples.

Results
Location privacy

Defense mechanisms:

- **Geometric**
- Laplacian
- Blahut-Arimoto [O+'17]



Gowalla dataset: users' location data, $100K$ examples.

Results
Location privacy

Defense mechanisms:

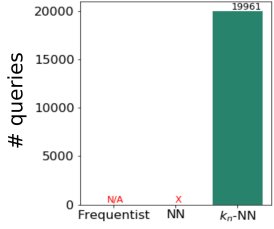
- Geometric
- **Laplacian**
- Blahut-Arimoto [O+'17]

Gowalla dataset: users' location data, $100K$ examples.

Results
Location privacy

Defense mechanisms:

- Geometric
- **Laplacian**
- Blahut-Arimoto [O+'17]

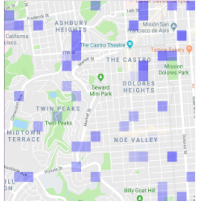


Gowalla dataset: users' location data, $100K$ examples.

Results
Location privacy

Defense mechanisms:

- Geometric
- Laplacian
- **Blahut-Arimoto [O+'17]**

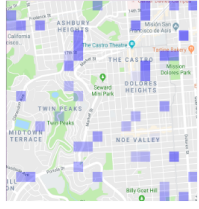


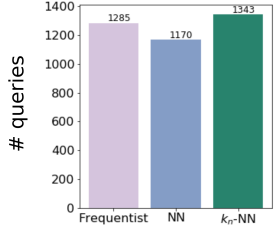
Gowalla dataset: users' location data, $100K$ examples.

Results
Location privacy

Defense mechanisms:

- Geometric
- Laplacian
- **Blahut-Arimoto [O+'17]**





Gowalla dataset: users' location data, $100K$ examples.

Results
Summary

- Scale to large systems & excel when there's a metric on the output

Results
Summary

- Scale to large systems & excel when there's a metric on the output
- When no metric: equivalent to Frequentist

Results
Summary

- Scale to large systems & excel when there's a metric on the output
- When no metric: equivalent to Frequentist
- However, may converge slowly for maliciously crafted systems

Results
Summary

- Scale to large systems & excel when there's a metric on the output
- When no metric: equivalent to Frequentist
- However, may converge slowly for maliciously crafted systems

fbleau: `https://github.com/gchers/fbleau`

Features and convergence

Recall that features may allow us to obtain faster convergence.

Features and convergence

Recall that features may allow us to obtain faster convergence.

We can define feature-dependent security measures. That is, estimate the Bayes risk in the feature space, R^*_Φ, and define security with $(\lambda(R^*_\Phi), \Phi)$.

Features and convergence

Recall that features may allow us to obtain faster convergence.

We can define feature-dependent security measures. That is, estimate the Bayes risk in the feature space, R^*_Φ, and define security with $(\lambda(R^*_\Phi), \Phi)$.

- Advantage: allows tackling even more larger/more complex systems

Features and convergence

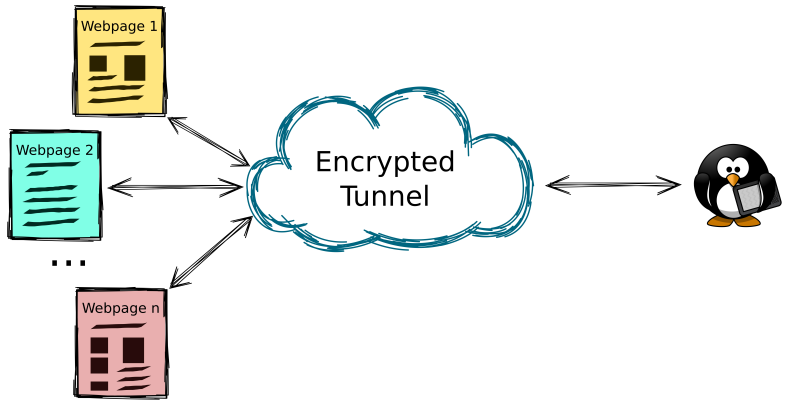
Recall that features may allow us to obtain faster convergence.

We can define feature-dependent security measures. That is, estimate the Bayes risk in the feature space, R^*_Φ, and define security with $(\lambda(R^*_\Phi), \Phi)$.

- Advantage: allows tackling even more larger/more complex systems
- But, weaker security definition.

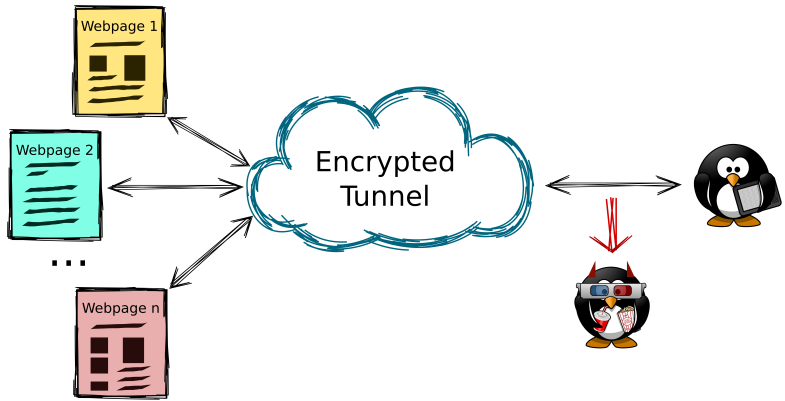
Application Example

Website Fingerprinting [C'17]



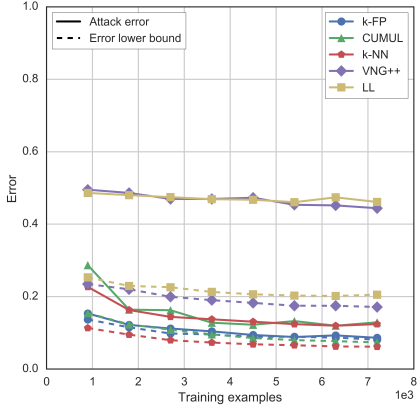
Application Example

Website Fingerprinting [C'17]



Website Fingerprinting
Results

Results on undefended Tor traffic



Extensions
Infinite secret spaces

A natural extension is to consider continuous secrets s.

Extensions
Infinite secret spaces

A natural extension is to consider continuous secrets s.

- 0-1 risk notion does not apply. Use for example $\ell(s, \hat{s}) = (s - \hat{s})^2$

Extensions
Infinite secret spaces

A natural extension is to consider continuous secrets s.

- 0-1 risk notion does not apply. Use for example $\ell(s, \hat{s}) = (s - \hat{s})^2$
- Use an UC regressor; for example, the k_n-NN regressor is UC under some conditions.

Summary

- With Black-box security we can model several attacks: location privacy, side channels, traffic analysis, …
- Frequentist paradigm doesn't scale & not applicable to continuous
- We can use ML results to scale to large/complex systems
- If needed, we can use features for scaling to even more complex problems (albeit achieving weaker security notions)

`https://github.com/gchers/fbleau`

Bibliography
Black-box security


K. Chatzikokolakis, T. Chothia, A. Guha
Statistical measurement of information leakage, 2010.


G. Cherubin
Bayes, not naïve: Security bounds on website fingerprinting defenses, 2017.


G. Cherubin, K. Chatzikokolakis, C. Palamidessi
F-BLEAU: Fast Black-box Leakage Estimation, IEEE S&P 2019.

Bibliography
Applications


T. Chothia, V. Smirnov
A traceability attack against e-passports, 2010.


M. Backes, B. Köpf
Formally bounding the side-channel leakage in unknown-message attacks, 2008.


S. Oya, C. Troncoso, F. Pérez-González
Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms, 2017.


R. Shokri, M. Stronati, C. Song, V. Shmatikov
Membership inference attacks against machine learning models, IEEE S&P 2017.