Challenges in Reliable Machine Learning

Kamalika Chaudhuri
University of California, San Diego
Textbook Machine Learning

Training and test data drawn iid from same distribution
High test accuracy means good classifier
Reality...
Reality...

Issues: Robustness, Sample selection bias, Privacy, ....
Issues: Robustness, Sample selection bias, Privacy,…. 

Our work: Foundations of Reliable Learning
Two Practical Issues

• Robustness

• Overfitting
Adversarial Examples

Small perturbation to legitimate inputs causing misclassification
Adversarial Examples

Can potentially lead to serious safety issues
Adversarial Examples: the SOTA

A large number of attacks
A few defenses
Not much understanding why they happen
Adversarial Examples: the SOTA

A large number of attacks
A few defenses
Not much understanding why they happen

Why do we have adversarial examples?
Why do we have adversarial examples?

1. Data distributions overlap

2. Too few samples

3. Bad algorithm
Background: Classification

Given: \( (x_i, y_i) \)

Vector of features \hspace{2cm} \text{Discrete Labels}

Find: Prediction rule in a class to predict \( y \) from \( x \)
The Statistical Learning Framework

Training and test data drawn from underlying distribution D
The Statistical Learning Framework

Training and test data drawn from underlying distribution $D$

**Goal:** Find classifier $f$ to maximize accuracy

$$\Pr_{(x,y) \sim D}(f(x) = y)$$
Robustness Measure

Classifier $f$ is robust with radius $s$ at $x$ if for distance $d$,

$$d(x, x') \leq s \implies f(x) = f(x')$$

Implies no adversarial examples around $x$
1. Adversarial accuracy = accuracy against adversarial examples at radius $s$

2. (Clean) Accuracy
Robustness in Neural Networks

NN computes function $f(x)$
Classifier outputs $\text{sign}(f(x))$

Robustness comes from Local Smoothness:
If $f$ is locally Lipschitz around $x$, and $f(x)$ is away from 0, then $f$ is robust at $x$
SOTA for Neural Networks

A large number of attacks

A few good defenses

All defenses show a robustness vs. accuracy tradeoff
SOTA for Neural Networks

A large number of attacks

A few good defenses

All defenses show a robustness vs. accuracy tradeoff

Is this tradeoff inevitable?
SOTA for Neural Networks

A large number of attacks

A few good defenses

All defenses show a robustness vs. accuracy tradeoff

Is this tradeoff inevitable?

Not in principle for real data
Definition: r-Separation

Distribution is $r$-separated if for any $(x, y)$ and $(x', y')$ drawn

\[ y \neq y' \implies d(x, x') \geq 2r \]
**Theorem [YRZSC20]** If distribution is \( r \)-separated, then there exists an \( f \) s.t. \( f \) is locally smooth and \( \text{sign}(f) \) has accuracy 1 and robustness radius \( r \).
# Real data is r-separated

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Separation (2r)</th>
<th>Typical s</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>0.74</td>
<td>0.1</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>0.21</td>
<td>0.03</td>
</tr>
<tr>
<td>SVHN*</td>
<td>0.09</td>
<td>0.03</td>
</tr>
<tr>
<td>ResImgnet*</td>
<td>0.18</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Separation = min distance of any two points in different classes

s = Typical robustness radius used in experiments
In principle, no robustness-accuracy tradeoff

In practice there is one

What accounts for this gap?
Empirical Study

4 standard image datasets
7 models
7 different training methods

Measure local smoothness, accuracy and adversarial accuracy

Question: Is local smoothness correlated with robustness?
## Typical Result - CIFAR 10

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Adversarial Accuracy</th>
<th>Lipschitz Constant</th>
<th>Gen Gap</th>
<th>Adv Gen Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>94</td>
<td>0</td>
<td>425</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>GR</td>
<td>81</td>
<td>21</td>
<td>28</td>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>LLR</td>
<td>91</td>
<td>22</td>
<td>94</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>RST</td>
<td>84</td>
<td>42</td>
<td>24</td>
<td>16</td>
<td>43</td>
</tr>
<tr>
<td>AT</td>
<td>83</td>
<td>44</td>
<td>26</td>
<td>16</td>
<td>50</td>
</tr>
<tr>
<td>Trades</td>
<td>86</td>
<td>47</td>
<td>22</td>
<td>14</td>
<td>46</td>
</tr>
</tbody>
</table>

Higher Lipschitz Constant means less smooth
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Adversarial Accuracy</th>
<th>Lipschitz Constant</th>
<th>Gen Gap</th>
<th>Adv Gen Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>94</td>
<td>0</td>
<td>425</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>GR</td>
<td>81</td>
<td>21</td>
<td>28</td>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>LLR</td>
<td>91</td>
<td>22</td>
<td>94</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>RST</td>
<td>84</td>
<td>42</td>
<td>24</td>
<td>16</td>
<td>43</td>
</tr>
<tr>
<td>AT</td>
<td>83</td>
<td>44</td>
<td>26</td>
<td>16</td>
<td>50</td>
</tr>
<tr>
<td>Trades</td>
<td>86</td>
<td>47</td>
<td>22</td>
<td>14</td>
<td>46</td>
</tr>
</tbody>
</table>

Higher Lipschitz Constant means less smooth.

Less smooth, more accurate.

More smooth, less accurate.
## Typical Result - CIFAR 10

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Adversarial Accuracy</th>
<th>Lipschitz Constant</th>
<th>Gen Gap</th>
<th>Adv Gen Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>94</td>
<td>0</td>
<td>425</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>GR</td>
<td>81</td>
<td>21</td>
<td>28</td>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>LLR</td>
<td>91</td>
<td>22</td>
<td>94</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>RST</td>
<td>84</td>
<td>42</td>
<td>24</td>
<td>16</td>
<td>43</td>
</tr>
<tr>
<td>AT</td>
<td>83</td>
<td>44</td>
<td>26</td>
<td>16</td>
<td>50</td>
</tr>
<tr>
<td>Trades</td>
<td>86</td>
<td>47</td>
<td>22</td>
<td>14</td>
<td>46</td>
</tr>
</tbody>
</table>

Higher Lipschitz Constant means less smooth.
Empirical Study

Question: Does dropout help generalization?
# Typical Result - CIFAR 10 - Dropout

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Adversarial Accuracy</th>
<th>Lipschitz Constant</th>
<th>Gen Gap</th>
<th>Adv Gen Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td>94</td>
<td>0</td>
<td>384 (425)</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>RST</td>
<td>85</td>
<td>43</td>
<td>31</td>
<td>14</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>(84)</td>
<td>(42)</td>
<td>(24)</td>
<td>(16)</td>
<td>(43)</td>
</tr>
<tr>
<td>AT</td>
<td>85</td>
<td>40</td>
<td>34</td>
<td>15</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>(84)</td>
<td>(44)</td>
<td>(26)</td>
<td>(16)</td>
<td>(50)</td>
</tr>
<tr>
<td>Trades</td>
<td>86</td>
<td>49</td>
<td>14</td>
<td>13</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>(86)</td>
<td>(47)</td>
<td>(22)</td>
<td>(14)</td>
<td>(46)</td>
</tr>
</tbody>
</table>
Observations

Smoother methods (TRADES, RST, AT) are more robust, less smooth methods (Nat, GR, LLR) less

Smoother methods have a large generalization gap

Dropout reduces the generalization gap (a little)
Conclusion: for Neural Networks

Real data is r-separated

Robust and accurate neural networks exist

Current algorithms do not find them, possibly because they do not generalize well enough

Need better methods for adversarial generalization
Why do we have adversarial examples?

1. Data distributions overlap

2. Too few samples

3. Bad algorithm
Non-Parametric Methods

k-Nearest Neighbors

Decision Trees

Others: Random Forests, Kernel classifiers
What is known about Nonparametric Methods?
The Bayes Optimal Classifier

Classifier with highest accuracy on data distribution

Reachable only in the infinite sample limit
[Stone77] As training data size grows, the accuracy of many non-parametric methods converges to accuracy of the Bayes Optimal
What about Robustness?
What is the limit of robust classification?

Bayes optimal undefined outside distribution

Bayes optimal
What is the limit of robust classification?

Bayes optimal undefined outside distribution

--- Bayes optimal

--- r-optimal

\[ r\text{-optimal} = \text{classifier that maximizes robust accuracy} \]

\[ f^*_r(x) = \begin{cases} 
\text{Predict } + & \text{if } d(x, X^+) \leq r \\
- & \text{otherwise}
\end{cases} \]

[Yang, Rashtchian, Wang, Chaudhuri, 20]
Convergence

Theorem: Conditions when non-parametrics converge to r-optimal

Convergence limit:
- **r-optimal**: Nearest neighbor, Kernel classifier
- **Bayes-optimal but not r-optimal**: Histogram, Decision tree

[Bhattacharjee, Chaudhuri, 20]
What happens with Decision Trees?

Data r-separated

Decision tree accurate but not robust
What about Robustness?

Robustness depends on training algorithm
Conclusion

• Propose the r-optimal — a large sample limit for robust classification

• Show that nearest neighbors and kernels converge to r-optimal for separated data, but not decision trees

• For non-parametrics, robustness also depends on the training algorithm
Two Practical Issues

• Robustness

• Overfitting
Deep Generative Models …

are highly successful
But do they overfit?

What does “overfitting” mean?
An Example

Generative model reproduces training data

Privacy concern in healthcare applications
The Setting and Classical View
The Setting

Training Data → Generative Model → Generated Data → Evaluation Score → Test Data

Training and test data drawn i.i.d from same distribution
Classical View of Overfitting

Likelihood = \( p(\text{data} \mid \text{model}) \)

Likelihood vs. model complexity
Classical View of Overfitting

Overfitting means

\[ p(\text{training data} \mid \text{model}) \gg p(\text{test data} \mid \text{model}) \]
Need to Estimate from Samples

Deep generative models:

- Intractable likelihood (e.g., VAEs)
- No likelihood, only samples (e.g., GANs)

10s - 100s of millions parameters, overfitting may be a concern
Talk Outline

• Two formal notions of Overfitting
Overfitting: Two Formal Notions

• Over-Representation

• Data-Copying
Over-Representation

Training data $T$  
Generated data $Q$

Test data $P$

Over-Representation in a region $C$ if

$\Pr_{x \sim Q} [x \text{ in } C] \gg \Pr_{x \sim P} [x \text{ in } C]$
Data-Copying

Training data $T$  
Generated data $Q$

Test data $P$

Distance metric $d$

Data-copying in a region $C$ if

$$\Pr_{x \sim Q, z \sim P}[d(x, T) < d(z, T) \mid x, z \text{ in } C]$$
Over-Representation is not Data Copying!

Overrepresentation
No data-copying

Data-copying
No overrepresentation
Existing Hypothesis Tests

- Inception Score (Salimans et al 2016)
- Frechet Inception Distance (Heusal et al 2017)
- Binning based Evaluation (Weiss et al 2018)
- Kernel MMD (Gretton et al 2016)
- Precision and Recall (Sajjadi et al 2018)

Almost all test over-representation!
Fail to detect exact data-copying
Talk Outline

• Two formal notions of Overfitting

• A Simple Test of Data-Copying
Data-Copying: A First Test

Training data $T$

Generated data $Q$

Test data $P$

Metric $d$

Test if

$$\frac{1}{|P|} \sum_{i \in P} d(x_i, T) - \frac{1}{|Q|} \sum_{i \in Q} d(x_i, T) \geq 0$$

Avg dist btw **test** and closest training pt

Avg dist btw **generated** and closest training pt

**Challenge:** High variance from outliers
Solution: Use Comparisons

Training data $T$  Generated data $Q$
Test data $P$  Metric $d$

Test if

$$\frac{1}{|P| \cdot |Q|} \sum_{i \in P, j \in Q} 1(d(x_i, T) \geq d(x_j, T)) \geq \frac{1}{2}$$

Is $P$ farther on average to $T$ than $Q$?

Mann-Whitney on distances between $P$ and $T$ and $Q$ and $T$
Talk Outline

• Two formal notions of Overfitting
• A Simple Test of Data-Copying
• Addressing Heterogeneity
Problem: Heterogeneity

Global tests will fail for heterogeneous models

How to account for heterogeneity?
Algorithmic Tool: Binning

Divide space into cells $c$
Algorithmic Tool: Binning

Divide space into cells $c$
Do test in each cell
Algorithmic Tool: Binning

Divide space into cells $c$
Do test in each cell
Report weighted average of one-sided test results
Talk Outline

- Two formal notions of Overfitting
- A Simple Test of Data-Copying
- Addressing Heterogeneity
- Evaluation
How well does our test detect data-copying?
Case Study 1: VAE

• **Dataset:** MNIST

• Latent dimension from 1 (underfit) to 100 (overfit)

• Metric d is the 64 dimensional latent space of an autoencoder trained with a VGGnet perceptual loss [Zhang et al, 2018]
Case Study 1: VAE

Baselines:

NN (Lopez-Paz, Oquab, 2016)

Generalization Gap

Our Test $C_T$
Case Study 2: Big GAN

- **Dataset:** Imagenet, Classes - coffee, bubble, schooner
- Generated sample complexity measured by truncation threshold
- Metric d is 64 dimensional PCA of Inception space
Case Study 2: Big GAN

Baselines:

NN (Lopez-Paz and Oquab, 2016)
Our test $C_T$
Case Study 2: Big GAN

Data-copied cell: \( Z_U = -1.46 \)

Not data-copied cell: \( Z_U = +1.40 \)
Conclusion

• We present a new three-sample test for data-copying

• Formalize two notions of overfitting
References

• “A Closer Look at Robustness vs. Accuracy”, Yaoyuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov and Kamalika Chaudhuri, NeuRIPS 2020


• “When are Non-Parametric Methods Robust?”, Robi Bhattacharjee and Kamalika Chaudhuri, ICML 2020

• “A Three Sample Test for Detecting Data-Copying in Generative Models”, Casey Meehan, Kamalika Chaudhuri and Sanjoy Dasgupta, AISTATS 2020
Thanks

Cyrus Rashtchian
Yaoyuan Yang
Yizhen Wang
Hongyang Zhang
Robi Bhattacharjee
Casey Meehan
Sanjoy Dasgupta
Ruslan Salakhutdinov