ML Big Picture

Learning Paradigms: What data is available and when? What form of prediction?
- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Problem Formulation: What is the structure of our output prediction?
- boolean: Binary Classification
- categorical: Multiclass Classification
- ordinal: Ordinal Classification
- real: Regression
- ordering: Ranking
- multiple discrete: Structured Prediction
- multiple continuous: (e.g. dynamical systems)
- both discrete & cont.: (e.g. mixed graphical models)

Theoretical Foundations: What principles guide learning?
- probabilistic
- information theoretic
- evolutionary search
- ML as optimization

Facets of Building ML Systems: How to build systems that are robust, efficient, adaptive, effective?
1. Data prep
2. Model selection
3. Training (optimization / search)
4. Hyperparameter tuning on validation data
5. (Blind) Assessment on test data

Big Ideas in ML: Which are the ideas driving development of the field?
- inductive bias
- generalization / overfitting
- bias-variance decomposition
- generative vs. discriminative
- deep nets, graphical models
- PAC learning
- distant rewards

Application Areas: Key challenges?
NLP, Speech, Computer Vision, Robotics, Medicine, Search
LEARNING THEORY
Questions For Today

1. Given a classifier with zero training error, what can we say about generalization error? (Sample Complexity, Realizable Case)

2. Given a classifier with low training error, what can we say about generalization error? (Sample Complexity, Agnostic Case)

3. Is there a theoretical justification for regularization to avoid overfitting? (Structural Risk Minimization)
PAC/SLT models for Supervised Learning

- **Data Source**
- **Distribution D on X**
- **Labeled Examples**
- **Expert / Oracle**

Learning Algorithm:

```
h : X → Y
```

Alg.outputs:

```
(x_1, c^*(x_1)), ..., (x_m, c^*(x_m))
```

Slide from Nina Balcan
Two Types of Error

True Error (aka. **expected risk**)

\[ R(h) = P_{x \sim p^*}(c^*(x) \neq h(x)) \]

Train Error (aka. **empirical risk**)

\[ \hat{R}(h) = P_{x \sim S}(c^*(x) \neq h(x)) \]

\[ = \frac{1}{N} \sum_{i=1}^{N} 1(c^*(x^{(i)}) \neq h(x^{(i)})) \]

\[ = \frac{1}{N} \sum_{i=1}^{N} 1(y^{(i)} \neq h(x^{(i)})) \]

where \( S = \{x^{(1)}, \ldots, x^{(N)}\}_{i=1}^{N} \) is the training data set, and \( x \sim S \) denotes that \( x \) is sampled from the empirical distribution.
PAC / SLT Model

1. Generate instances from unknown distribution \( p^* \)
   \[ x^{(i)} \sim p^*(x), \forall i \]  
   \hspace{100pt} (1)

2. Oracle labels each instance with unknown function \( c^* \)
   \[ y^{(i)} = c^*(x^{(i)}), \forall i \]  
   \hspace{100pt} (2)

3. Learning algorithm chooses hypothesis \( \hat{h} \in \mathcal{H} \) with low(est) training error, \( \hat{R}(\hat{h}) \)
   \[ \hat{h} = \arg \min_{\hat{h}} \hat{R}(\hat{h}) \]  
   \hspace{100pt} (3)

4. Goal: Choose an \( \hat{h} \) with low generalization error \( R(\hat{h}) \)
Three Hypotheses of Interest

The true function $c^*$ is the one we are trying to learn and that labeled the training data:

$$y^{(i)} = c^*(x^{(i)}), \forall i$$ \hspace{1cm} (1)

The expected risk minimizer has lowest true error:

$$\hat{h}^* = \arg\min_{h \in \mathcal{H}} R(h)$$ \hspace{1cm} (2)

The empirical risk minimizer has lowest training error:

$$\tilde{h} = \arg\min_{h \in \mathcal{H}} \tilde{R}(h)$$ \hspace{1cm} (3)
Probably Approximately Correct (PAC) Learning

Whiteboard:

– PAC Criterion
– Meaning of “Probably Approximately Correct”
– PAC Learnable
– Consistent Learner
– Sample Complexity
Generalization and Overfitting

Whiteboard:

– Realizable vs. Agnostic Cases
– Finite vs. Infinite Hypothesis Spaces
PAC Learning

The PAC criterion is that our learner produces a high accuracy learner with high probability:

\[ P(|R(h) - \hat{R}(h)| \leq \epsilon) \geq 1 - \delta \]

(1)

Suppose we have a learner that produces a hypothesis \( h \in \mathcal{H} \) given a sample of \( N \) training examples. The algorithm is called consistent if for every \( \epsilon \) and \( \delta \), there exists a positive number of training examples \( N \) such that for any distribution \( p^* \), we have that:

\[ P(|R(h) - \hat{R}(h)| > \epsilon) < \delta \]

(2)

The sample complexity is the minimum value of \( N \) for which this statement holds. If \( N \) is finite for some learning algorithm, then \( \mathcal{H} \) is said to be learnable. If \( N \) is a polynomial function of \( \frac{1}{\epsilon} \) and \( \frac{1}{\delta} \) for some learning algorithm, then \( \mathcal{H} \) is said to be PAC learnable.
Sample Complexity Results

Definition 0.1. The sample complexity of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about...

Realizable

Agnostic

Finite $|\mathcal{H}|$

We’ll start with the finite case...

Infinite $|\mathcal{H}|$
Sample Complexity Results

Definition 0.1. The sample complexity of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about...

\[
N \geq \frac{1}{\varepsilon} \left[ \log(|\mathcal{H}|) + \log\left(\frac{1}{\delta}\right) \right]
\]
labeled examples are sufficient so that with probability \((1 - \delta)\) all \(h \in \mathcal{H}\) with \(R(h) \geq \varepsilon\) have \(\hat{R}(h) > 0\).
Example: Conjunctions

*In-Class Quiz:*
Suppose $H = \text{class of conjunctions over } x \text{ in } \{0,1\}^M$
If $M = 10$, $\epsilon = 0.1$, $\delta = 0.01$, how many examples suffice?

| Finite $|H|$ | Realizable | Agnostic |
| --- | --- | --- |
| $N \geq \frac{1}{\epsilon} \left[ \log(|H|) + \log(\frac{1}{\delta}) \right]$ labeled examples are sufficient so that with probability $(1 - \delta)$ all $h \in H$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0.$ | | |

| Infinite $|H|$ | | |
| | | |
**Sample Complexity Results**

**Definition 0.1.** The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

**Four Cases we care about...**

| Finite $|\mathcal{H}|$ | Realizable | Agnostic |
|-----------------------|------------|----------|
| $N \geq \frac{1}{\varepsilon} \left[\log(|\mathcal{H}|) + \log\left(\frac{1}{\delta}\right)\right]$ labeled examples are sufficient so that with probability $(1 - \delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \varepsilon$ have $\hat{R}(h) > 0$. | $N \geq \frac{1}{2\varepsilon^2} \left[\log(|\mathcal{H}|) + \log\left(\frac{3}{\delta}\right)\right]$ labeled examples are sufficient so that with probability $(1 - \delta)$ for all $h \in \mathcal{H}$ we have that $|R(h) - \hat{R}(h)| < \varepsilon$. |
1. Bound is **inversely linear in epsilon** (e.g. halving the error requires double the examples)

2. Bound is **only logarithmic in |H|** (e.g. quadrupling the hypothesis space only requires double the examples)

---

1. Bound is **inversely quadratic in epsilon** (e.g. halving the error requires 4x the examples)

2. Bound is **only logarithmic in |H|** (i.e. same as Realizable case)

---

Finite $|\mathcal{H}|$

\[
N \geq \frac{1}{\epsilon} \left[ \log(|\mathcal{H}|) + \log\left(\frac{1}{\delta}\right) \right] \text{ labeled examples are sufficient so that with probability } (1 - \delta) \text{ all } h \in \mathcal{H} \text{ with } R(h) \geq \epsilon \text{ have } \hat{R}(h) > 0.
\]

---

Infinite $|\mathcal{H}|$

---

Realizable

Agnostic

\[
N \geq \frac{1}{2\epsilon^2} \left[ \log(|\mathcal{H}|) + \log\left(\frac{3}{\delta}\right) \right] \text{ labeled examples are sufficient so that with probability } (1 - \delta) \text{ for all } h \in \mathcal{H} \text{ we have that } |R(h) - \hat{R}(h)| < \epsilon.
\]
Generalization and Overfitting

Whiteboard:

– Sample Complexity Bounds (Agnostic Case)
– Corollary (Agnostic Case)
– Empirical Risk Minimization
– Structural Risk Minimization
– Motivation for Regularization
**Sample Complexity Results**

**Definition 0.1.** The *sample complexity* of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

**Four Cases we care about…**

<table>
<thead>
<tr>
<th></th>
<th>Realizable</th>
<th>Agnostic</th>
</tr>
</thead>
</table>
| Finite $|\mathcal{H}|$ | $N \geq \frac{1}{\varepsilon} \left[ \log(|\mathcal{H}|) + \log\left(\frac{1}{\delta}\right) \right]$ labeled examples are sufficient so that with probability $(1 - \delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) > 0$. | $N \geq \frac{1}{2\varepsilon^2} \left[ \log(|\mathcal{H}|) + \log\left(\frac{3}{\delta}\right) \right]$ labeled examples are sufficient so that $|R(h) - \hat{R}(h)| < \varepsilon$ with probability $(1 - \delta)$ for all $h \in \mathcal{H}$.

We need a new definition of “complexity” for a Hypothesis space for these results (see VC Dimension).
Learning Theory Objectives

You should be able to...

• Identify the properties of a learning setting and assumptions required to ensure low generalization error
• Distinguish true error, train error, test error
• Define PAC and explain what it means to be approximately correct and what occurs with high probability
• Apply sample complexity bounds to real-world learning examples
• Distinguish between a large sample and a finite sample analysis
• Theoretically motivate regularization