Learning Theory

Piyush Rai

CS5350/6350: Machine Learning

September 27, 2011

Why Learning Theory?

We want to have **theoretical guarantees** about our learning algorithms

- We have seen a number of learning algorithms so far
- How can we tell if our learning algorithm will do a good job?
  - Experimental results
  - Theoretical analysis
- Why theory?
  - I can only run so many experiments
  - Experiments rarely tell me what will go wrong
  - I want to be able to deploy my algorithm on Mars

"There is nothing more practical than a good theory" - Kurt Lewin

Hypothesis Class, Training Error, and Expected Error

- The hypothesis class $\mathcal{H}$ is a space of functions (assume it's finite for now)
- The learning algorithm learns a function (hypothesis) $h \in \mathcal{H}$
- Assume h is learned using a sample $\mathcal{D}$ of N i.i.d. training examples $(\mathbf{x}_n, y_n)_{n=1}^N$ drawn from $P(\mathbf{x}, y)$; (also denoted as $\mathcal{D} \sim P^N$)
- The 0-1 training error (also called the empirical error) of h

$$L_{\mathcal{D}}(h) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(h(\mathbf{x}_n) \neq y_n)$$

- The 0-1 expected error (also called the true error, or misclassification probability) of h

$$L_P(h) = \mathbb{E}_{(\mathbf{x},y)\sim P}[\mathbb{I}(h(\mathbf{x}) \neq y)]$$

- The expected error, in general, is much worse than the training error
  - We want to know how much worse it is..
  - .. without doing experiments (e.g., cross-validation) :-)

Zero Training Error

- Assume some $h \in \mathcal{H}$ with zero training error and true error $L_P(h) > \epsilon$
- Probability of h having zero error on any training example $\leq 1 - \epsilon$
- Probability of h having zero error on any training set $\mathcal{D}$ of N examples

$$P_{\mathcal{D}\sim P^N}(L_{\mathcal{D}}(h) = 0 \cap L_P(h) > \epsilon) \leq (1-\epsilon)^N$$

- Let's call $L_{\mathcal{D}}(h) = 0 \cap L_P(h) > \epsilon$ as "h is bad"
- Let's assume $\mathcal{H}$ has K such hypotheses $\{h_1, \ldots, h_K\}$
- Probability that **at least one** of these has zero training error

$$P_{\mathcal{D}\sim P^N}(\text{“}h_1 \text{ is bad”} \cup \ldots \cup \text{“}h_K \text{ is bad”}) \leq K(1-\epsilon)^N \quad \text{(using union bound)}$$

- Since $K \leq |\mathcal{H}|$, K can be replaced by the size of set $\mathcal{H}$

$$P_{\mathcal{D}\sim P^N}(\exists h : \text{“}h \text{ is bad”}) \leq |\mathcal{H}|(1-\epsilon)^N$$

Zero Training Error

- Using $(1-\epsilon) < e^{-\epsilon}$, we get:

$$P_{\mathcal{D}\sim P^N}(\exists h : \text{“}h \text{ is bad”}) \quad \leq \quad |\mathcal{H}|e^{-N\epsilon}$$

- Probability of h being bad decreases exponentially with N
- Number of examples needed to keep the failure probability $|\mathcal{H}|e^{-N\epsilon} \leq \delta$:

$$\boxed{N \geq \frac{1}{\epsilon}(\log|\mathcal{H}| + log\frac{1}{\delta})}$$

- This gives the sufficient number of examples for which the learned hypothesis will be probably (with probability $1-\delta$) and approximately (with error ϵ) correct (**PAC** Learning: **P**robably and **A**pproximately **C**orrect Learning)
- δ is the probability that the true error is $> \epsilon$. With probability $1-\delta$, given training sample size N, the true error is bounded by ϵ

$$\boxed{L_P(h) \leq \frac{\log|\mathcal{H}| + log\frac{1}{\delta}}{N}}$$

Non-Zero Training Error

- Given N random variables $z_1, \ldots, z_N$, the empirical mean
$$\bar{z} = \frac{1}{N}\sum_{n=1}^{N} z_n$$
- Let's assume the true mean is μ_z
- **Chernoff Bound** says:
$$P(|\mu_z - \bar{z}| \geq \epsilon) \leq e^{-2N\epsilon^2}$$
- Using the same result, for any single hypothesis $h \in \mathcal{H}$, we have:
$$P(L_P(h) - L_{\mathcal{D}}(h) \geq \epsilon) \leq e^{-2N\epsilon^2}$$
- Using the union bound, for **at least one** hypothesis $h \in \mathcal{H}$, we have:
$$P(\exists h : L_P(h) - L_{\mathcal{D}}(h) \geq \epsilon) \leq |\mathcal{H}|e^{-2N\epsilon^2}$$

Non-Zero Training Error

- Number of examples needed to keep the failure probability $|\mathcal{H}|e^{-2N\epsilon^2} \leq \delta$:

$$\boxed{N \geq \frac{1}{2\epsilon^2}(\log|\mathcal{H}| + log\frac{1}{\delta})}$$

- Number of examples grows as square of $1/\epsilon$ (note: $\epsilon < 1$)
  - In zero-error case, it grows linearly with $1/\epsilon$

  $\Rightarrow$ For given ϵ, δ, the non-zero training error case requires more examples

- δ is the probability that the difference between the expected error and the training error is $\epsilon \geq \sqrt{(\log|\mathcal{H}| + \log\frac{1}{\delta})/2N}$

- With probability $1 - \delta$, given training sample size N:

$$\boxed{L_P(h) \leq L_{\mathcal{D}}(h) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{2N}}}$$

- The difference worsens as the size of $\mathcal{H}$ (grows as square-root of $\log|\mathcal{H}|$)
- Size is also a measure of the complexity of the hypothesis class

Infinite Sized Hypothesis Spaces

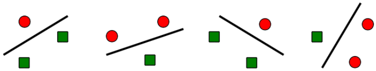
- For the finite sized hypothesis class $\mathcal{H}$

$$L_P(h) \leq L_{\mathcal{D}}(h) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{2N}}$$

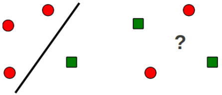
- What happens when the hypothesis class size $|\mathcal{H}|$ is infinite?
  - Example: the set of all linear classifiers
- The above bound doesn't apply (it just becomes trivial)
- We need some other way of measuring the size of $\mathcal{H}$
  - One way: use the complexity $\mathcal{H}$ as a measure of its size
  - .. enters the Vapnik-Chervonenkis dimension (VC dimension)
  - VC dimension: a measure of the complexity of a hypothesis class

Shattering

- **Definition:** A set of points is shattered by a hypothesis class $\mathcal{H}$ if for **all possible binary labelings** of the points, there exists some $h \in \mathcal{H}$ that can represent the corresponding labeling function
- Consider 3 points (in any positions) in 2D and some possible labelings



- In 2D, 3 points can always be shattered by linear separators
  - .. no matter how they are positioned
- Now how about 4 points in 2D?



- For some labelings of 4 points in 2D, a linear separator doesn't exist
- The hypothesis class of linear separator can shatter maximum 3 points in 2D

VC Dimension: The Shattering Game

The concept of shattering is used to define the VC dimension of hypothesis classes

Consider the following shattering game between us and an adversary

To show that a hypothesis class $\mathcal{H}$ has a VC dimension d (in some input space)

- We choose d points positioned however we want
- Adversary labels these d points
- We choose a hypothesis $h \in \mathcal{H}$ that separates the points

The VC dimension of $\mathcal{H}$, *in that input space*, is the maximum d we can choose so that we always succeed in the game

In the previous slide, we just (informally) showed that the VC dimension of linear classifiers in $\mathbb{R}^2$ is . . . 3

VC Dimension: Some Examples

What about the VC dimension of linear classifiers in $\mathbb{R}^3$?

$VC = 4$ seems like a reasonable guess!

What about the VC dimension of linear classifiers in $\mathbb{R}^D$?

$VC = D + 1$ would be our next guess (and that's right!)

Recall: a linear classifier in $\mathbb{R}^D$ is defined by D parameters (one per feature)

For linear classifiers, high $D \Rightarrow$ high VC dimension $\Rightarrow$ high complexity

Note: VC dimension isn't always the number of parameters of the classifier

What about the VC dimension of 1-nearest neighbors?

Infinite. Why?

What about the VC dimension of SVM with RBF kernel?

Infinite. Why?

Using VC Dimension in Generalization Bounds

Recall the PAC based Generalization Bound

$$\text{ExpectedLoss}(h) \leq \text{TrainingLoss}(h) + \sqrt{\frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{2N}}$$

For hypothesis classes with infinite size ($|\mathcal{H}| = \infty$), but VC dimension d:

$$\boxed{\text{ExpectedLoss}(h) \leq \text{TrainingLoss}(h) + \sqrt{\frac{d(\log\frac{2N}{d} + 1) + \log\frac{4}{\delta}}{2N}}}$$

For **linear classifiers**, what does it imply?

Having fewer features is better (since it means smaller VC dimension)

VC Dimension of Support Vector Machines

Recall: VC dimension of an SVM with RBF kernel is infinite. Is it a bad thing?

Not really. SVM's large margin property ensures good generalization

Theorem (Vapnik, 1982):

- Given N data points in $\mathbb{R}^D$: $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with $||\mathbf{x}_n|| \leq R$
- Define $\mathcal{H}_\gamma$: set of classifiers in $\mathbb{R}^D$ having margin γ on $\mathbf{X}$

The VC dimension of $\mathcal{H}_\gamma$ is bounded by:

$$\boxed{VC(\mathcal{H}_\gamma) \leq \min\left\{D, \left\lceil \frac{4R^2}{\gamma^2} \right\rceil\right\}}$$

Generalization bound for the SVM:

$$\boxed{\text{ExpectedLoss}(h) \leq \text{TrainingLoss}(h) + \sqrt{\frac{VC(\mathcal{H}_\gamma)(\log \frac{2N}{VC(\mathcal{H}_\gamma)} + 1) + \log \frac{4}{\delta}}{2N}}}$$

Large $\gamma \Rightarrow$ small VC dim. $\Rightarrow$ small complexity of $\mathcal{H}_\gamma \Rightarrow$ good generalization

Things to Remember..

- We care about the expected error, not the training error
- For finite sized hypothesis spaces $\log |\mathcal{H}|$ is a measure of complexity
- Difference between expected error and training error grows as $\log |\mathcal{H}|$
- Standard PAC bounds only apply to finite hypothesis classes
- VC dimension is a measure of complexity of **infinite sized hypothesis classes**
- Generalization error (as measured by the difference between expected error and training error) now scales in terms of VC dimension (large VC dimension $\Rightarrow$ poor generalization)
  - .. unless we have large margins
    $\Rightarrow$ Large margins imply small VC dimension