Machine Learning

CSE6740/CS7641/ISYE6740, Fall 2012

Computational Learning Theory II

Le Song

Lecture 11, September 25, 2012

Based on Slides from Eric Xing, CMU

Reading: Chap. 7 T.M book
Last time: PAC and Agnostic Learning

- Finite H, assume target function $c \in H$

$$Pr(\exists h \in H, \text{ s.t. } (error_{\text{train}}(h) = 0) \land (error_{\text{true}}(h) > \epsilon) ) \leq |H|e^{-\epsilon m}$$

  - Suppose we want this to be at most $\delta$. Then $m$ examples suffice:

$$m \geq \frac{1}{\epsilon} (\ln |H| + \ln(1/\delta))$$

- Finite H, agnostic learning: perhaps $c$ not in H

$$P(\exists h \in H, |\epsilon(h) - \hat{\epsilon}(h)| > \gamma) = 2k \exp(-2\gamma^2 m)$$

  - $m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$

  - with probability at least $(1-\delta)$ the best training hypothesis $h$ satisfies

$$\epsilon(\hat{h}) \leq \left( \min_{h \in H} \epsilon(h) \right) + 2\sqrt{\frac{1}{m} \log \frac{2k}{\delta}}$$
Agnostic Case: Best Training Hypothesis

- We can also hold $m$ and $\delta$ fixed and solve for $\gamma$ in the previous equation, and show [again, convince yourself that this is right!] that with probability $1 - \delta$, we have that for all $h_i \in H$

$$|\hat{\epsilon}(h) - \epsilon(h)| \leq \sqrt{\frac{1}{m} \log \frac{2k}{\delta}}$$

- Define $\hat{h} = \arg\min_{h \in H} \hat{\epsilon}(h)$ to be the best hypothesis in training data, $h^* = \arg\min_{h \in H} \epsilon(h)$ to be the best possible hypothesis w.r.t. data distribution.
  - $\epsilon(\hat{h}) \leq \hat{\epsilon}(\hat{h}) + \gamma \leq \hat{\epsilon}(h^*) + \gamma \leq \epsilon(h^*) + 2\gamma$
  - If uniform convergence occurs, then the generalization error of $\hat{\epsilon}(h)$ is at most $2\gamma$ worse than the best possible hypothesis in H!
Agnostic Case: Best Training Hypothesis

**Theorem.** Let $|\mathcal{H}| = k$, and let any $m, \delta$ be fixed. Then with probability at least $1 - \delta$, we have that

$$
\varepsilon(\hat{h}) \leq \left( \min_{\hat{h} \in \mathcal{H}} \varepsilon(\hat{h}) \right) + 2 \sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}.
$$

**Corollary.** Let $|\mathcal{H}| = k$, and let any $\delta, \gamma$ be fixed. Then for $\varepsilon(\hat{h}) \leq \min_{\hat{h} \in \mathcal{H}} \varepsilon(\hat{h}) + 2\gamma$ to hold with probability at least $1 - \delta$, it suffices that

$$
m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}
$$

$$
= O \left( \frac{1}{\gamma^2} \log \frac{k}{\delta} \right),
$$
What if H is not finite?

- Can’t use our result for infinite H

- Need some other measure of complexity for H
  – Vapnik-Chervonenkis (VC) dimension!
What if $H$ is not finite?

- **Some Informal Derivation**
  - Suppose we have an $H$ that is parameterized by $d$ real numbers. Since we are using a computer to represent real numbers, and IEEE double-precision floating point (double's in C) uses 64 bits to represent a floating point number, this means that our learning algorithm, assuming we're using double-precision floating point, is parameterized by $64d$ bits.
  - $|H| = 2^{64d}$
  - $m \geq \frac{1}{2\gamma^2} \log \frac{2^{64d}}{\delta}$

- **Parameterization** (linear regression $\theta$; NB: $\mu, \sigma$)
How do we characterize “power”? 

- Different machines have different amounts of “power”.
- Tradeoff between:
  - More power: Can model more complex classifiers but might overfit.
  - Less power: Not going to overfit, but restricted in what it can model

\[
\epsilon(\hat{h}) \leq \left( \min_{h \in H} \epsilon(h) \right) + 2\sqrt{\frac{1}{m} \log \frac{2k}{\delta}}
\]

- How do we characterize the amount of power?
Shattering a Set of Instances

- **Definition:** Given a set \( S = \{ x^{(1)}, \ldots, x^{(d)} \} \) (no relation to the training set) of points \( x^{(i)} \in X \), we say that \( \mathcal{H} \) shatters \( S \) if \( \mathcal{H} \) can realize any labeling on \( S \).

  I.e., if for any set of labels \( \{ y^{(1)}, \ldots, y^{(d)} \} \), there exists some \( h \in \mathcal{H} \) so that \( h(x^{(i)}) = y^{(i)} \) for all \( i = 1, \ldots, d \).

- There are \( 2^d \) different ways to separate the sample in two sub-samples (a dichotomy)
Three Instances Shattered
The Vapnik-Chervonenkis Dimension

- **Definition:** The Vapnik-Chervonenkis dimension, \( VC(H) \), of hypothesis space \( H \) defined over instance space \( X \) is the size of the largest finite subset of \( X \) shattered by \( H \). If arbitrarily large finite sets of \( X \) can be shattered by \( H \), then \( VC(H) \equiv \infty \).
Consider $X = \mathbb{R}$, want to learn $c: X \rightarrow \{0, 1\}$

What is VC dimension of

- Open intervals:
  $H_1$: if $x > a$, then $y = 1$ else $y = 0$
  $VC_{H_1} = 2$

- Closed intervals:
  $H_2$: if $a < x < b$, then $y = 1$ else $y = 0$
  $VC_{H_2} = 3$
VC dimension: examples

Consider $X = \mathbb{R}^2$, want to learn $c: X \rightarrow \{0,1\}$

- What is VC dimension of lines in a plane?
  
  $H= \{ (wx+b)>0 \rightarrow y=1 \} \}$
  
  $VC(H)=3$
For any of the eight possible labelings of these points, we can find a linear classifier that obtains "zero training error" on them.

Moreover, it is possible to show that there is no set of 4 points that this hypothesis class can shatter.
• The VC dimension of H here is 3 even though there may be sets of size 3 that it cannot shatter.

• under the definition of the VC dimension, in order to prove that VC(H) is at least d, we need to show only that there's at least one set of size d that H can shatter.
**Theorem** Consider some set of \( m \) points in \( \mathbb{R}^n \). Choose any one of the points as origin. Then the \( m \) points can be shattered by oriented hyperplanes if and only if the position vectors of the remaining points are linearly independent.

- Obtain orthonormal basis \( v_1, \ldots, v_m \) from \( m \) points
- \( w = \sum_{i=1}^{m} y_i \ v_i \)

**Corollary**: The VC dimension of the set of oriented hyperplanes in \( \mathbb{R}^n \) is \( n+1 \).

Proof: we can always choose \( n + 1 \) points, and then choose one of the points as origin, such that the position vectors of the remaining \( n \) points are linearly independent, but can never choose \( n + 2 \) such points (since no \( n + 1 \) vectors in \( \mathbb{R}^n \) can be linearly independent).
The VC Dimension and the Number of Parameters

- The VC dimension thus gives concreteness to the notion of the capacity of a given set of $h$.
- Is it true that learning machines with many parameters would have high VC dimension, while learning machines with few parameters would have low VC dimension?

An infinite-VC function with just one parameter!

$$f(x, \alpha) \equiv \theta(\sin(\alpha x)), \quad x, \alpha \in R$$

where $\theta$ is an indicator function
An infinite-VC function with just one parameter

- You choose some number \( l \), and present me with the task of finding \( l \) points that can be shattered. I choose them to be

\[
x_i = 10^{-i} \quad i = 1, \ldots, l.
\]

- You specify any labels you like:

\[
y_1, y_2, \ldots, y_l, \quad y_i \in \{-1, 1\}
\]

- Then \( f(\alpha) \) gives this labeling if I choose \( \alpha \) to be

\[
\alpha = \pi \left(1 + \sum_{i=1}^{l} \frac{(1 - y_i)10^i}{2}\right)
\]

- Thus the VC dimension of this machine is infinite.
Sample Complexity from VC Dimension

- How many randomly drawn examples suffice to $\varepsilon$-exhaust $V_{S_{H,S}}$ with probability at least $(1 - \delta)$?

  ie., to guarantee that any hypothesis that perfectly fits the training data is probably $(1-\delta)$ approximately ($\varepsilon$) correct on testing data from the same distribution

\[ m \geq \frac{1}{\varepsilon} \left( 4 \log_2 \left( \frac{2}{\delta} \right) + 8 \text{VC}(H) \log_2 \left( \frac{13}{\varepsilon} \right) \right) \]

Compare to our earlier results based on $|H|$:  

\[ m \geq \frac{1}{2\varepsilon^2} \left( \ln |H| + \ln \left( \frac{1}{\delta} \right) \right) \]
Mistake Bounds

So far: how many examples needed to learn?
What about: how many mistakes before convergence?

Let's consider similar setting to PAC learning:

- Instances drawn at random from $X$ according to distribution $D$
- Learner must classify each instance before receiving correct classification from teacher
- Can we bound the number of mistakes learner makes before converging?
Statistical Learning Problem

- A model computes a function: \( h(X, w) \)

- Problem: minimize in \( w \) Risk Expectation

\[
R(w) = \int Q(z, w) dP(z)
\]

- \( w \): a parameter that specifies the chosen model
- \( z = (X, y) \) are possible values for attributes (variables)
- \( Q \) measures (quantifies) model error cost
- \( P(z) \) is the underlying probability law (unknown) for data \( z \)
Statistical Learning Problem (2)

- We get $L$ data from learning sample $(z_1, \ldots, z_L)$, and we suppose them iid sampled from law $P(z)$.
- To minimize $R(w)$, we start by minimizing Empirical Risk over this sample:

$$E(W) = \frac{1}{L} \sum_{i=1}^{L} Q(Z_i, W)$$

- We shall use such an approach for:
  - classification (eg. $Q$ can be a cost function based on cost for misclassified points)
  - regression (eg. $Q$ can be a cost of least squares type)
Central problem for Statistical Learning Theory:

What is the relation between Risk Expectation $R(W)$ and Empirical Risk $E(W)$?

How to define and measure a generalization capacity ("robustness") for a model?
Four Pillars for SLT

- Consistency (guarantees generalization)
  - Under what conditions will a model be consistent?

- Model convergence speed (a measure for generalization)
  - How does generalization capacity improve when sample size $L$ grows?

- Generalization capacity control
  - How to control in an efficient way model generalization starting with the only given information we have: our sample data?

- A strategy for good learning algorithms
  - Is there a strategy that guarantees, measures and controls our learning model generalization capacity?
Consistent training?

%error

number of training examples

%error

Test error

Training error

number of training examples

Test error

Training error
Q: Under which conditions will a learning model be consistent?

A: A model will be consistent if and only if the function $h$ that defines the model comes from a family of functions $H$ with finite VC dimension $d$.

A finite VC dimension $d$ not only guarantees a generalization capacity (consistency), but to pick $h$ in a family $H$ with finite VC dimension $d$ is the only way to build a model that generalizes.

Vapnik main theorem
Q : What is the nature of model error difference between learning data (sample) and test data, for a sample of finite size $m$?

A : This difference is no greater than a limit that only depends on the ratio between VC dimension $d$ of model functions family $H$, and sample size $m$, ie $d/m$

This statement is a new theorem that belongs to Kolmogorov-Smirnov way for results, ie theorems that do not depend on data’s underlying probability law.
Agnostic Learning: VC Bounds

- **Theorem:** Let $H$ be given, and let $d = VC(H)$. Then with probability at least $1 - \delta$, we have that for all $h \in H$,

$$|\hat{\epsilon}(h) - \epsilon(h)| \leq O\left(\sqrt{\frac{d}{m} \log \frac{m}{d}} - \frac{1}{m} \log \delta \right)$$

or

$$\epsilon(h) \leq \hat{\epsilon}(h) + O\left(\sqrt{\frac{d}{m} \log \frac{m}{d}} - \frac{1}{m} \log \delta \right)$$

recall that in finite $H$ case, we have:

$$|\hat{\epsilon}(h) - \epsilon(h)| \leq \sqrt{\frac{1}{m} \log 2k} - \frac{1}{m} \log \delta$$
Model convergence speed

% error

Confidence Interval

Test data error

Learning sample error

Sample size L
How to control model generalization capacity

Risk Expectation = Empirical Risk + Confidence Interval

- To minimize Empirical Risk alone will not always give a good generalization capacity: one will want to minimize the sum of Empirical Risk and Confidence Interval

- What is important is not Vapnik limit numerical value, most often too large to be of any practical use, it is the fact that this limit is a non decreasing function of model family function “richness”
Empirical Risk Minimization

- With probability $1 - \delta$, the following inequality is true:

$$\int (y - f(x, w^0))^2 dP(x, y) < \frac{1}{m} \sum_{i=1}^{m} (y_i - f(x_i, w^0))^2 + \sqrt{\frac{d(\ln(2m/d) + 1) - \ln \delta}{m}}$$

- where $w^0$ is the parameter $w$ value that minimizes Empirical Risk:

$$E(W) = \frac{1}{m} \sum_{i=1}^{m} (y_i - f(x_i, w))^2$$
Minimizing The Bound by Minimizing $d$

- Given some selection of learning machines whose empirical risk is zero, one wants to choose that learning machine whose associated set of functions has minimal VC dimension.

- By doing this we can attain an upper bound on the actual risk. This does not prevent a particular machine with the same value for empirical risk, and whose function set has higher VC dimension, from having better performance.

- What is the VC of a kNN? ($\infty$)
Structural Risk Minimization

- Which hypothesis space should we choose?
- Bias / variance tradeoff
- SRM: choose $H$ to minimize bound on true error!

$$
\epsilon(h) \leq \hat{\epsilon}(h) + O\left(\sqrt{\frac{d}{m} \log \frac{m}{d}} - \frac{1}{m} \log \delta\right)
$$

unfortunately a somewhat loose bound...
SRM strategy (1)

- With probability $1 - \delta$,

$$
\epsilon(h) \leq \hat{\epsilon}(h) + O\left(\sqrt{\frac{d}{m} \log \frac{m}{d}} - \frac{1}{m} \log \delta\right)
$$

- When $m/d$ is small (d too large), second term of equation becomes large

- SRM basic idea for strategy is to minimize simultaneously both terms standing on the right of above majoring equation for $\epsilon(h)$

- To do this, one has to make $d$ a controlled parameter
SRM strategy (2)

- Let us consider a sequence $H_1 < H_2 < \ldots < H_n$ of model family functions, with respective growing VC dimensions
  $$d_1 < d_2 < \ldots < d_n$$

- For each family $H_i$ of our sequence, the inequality
  $$\epsilon(h) \leq \hat{\epsilon}(h) + O\left(\sqrt{\frac{d}{m} \log \frac{m}{d}} - \frac{1}{m} \log \delta\right)$$

  is valid
  - That is, for each subset, we must be able either to compute $d$, or to get a bound on $d$ itself.

- SRM then consists of finding that subset of functions which minimizes the bound on the actual risk.
SRM strategy (3)

SRM: find $i$ such that expected risk $\epsilon(h)$ becomes minimum, for a specific $d^*=d_i$, relating to a specific family $H_i$ of our sequence; build model using $h$ from $H_i$. 

![Graph showing the relationship between risk, model complexity, and best model selection.](image-url)
Putting SRM into action: linear models case (1)

- There are many SRM-based strategies to build models:

- In the case of linear models

  \[ y = \langle w | x \rangle + b, \]

  one wants to make \( ||w|| \) a controlled parameter: let us call \( H_C \) the linear model function family satisfying the constraint:

  \[ ||w|| < C \]

Vapnik Major theorem:

When \( C \) decreases, \( d(H_C) \) decreases

\[ ||x|| < R \]
Putting SRM into action: linear models case (2)

- To control $||w||$, one can envision two routes to model:
  
  - **Regularization/Ridge Regression**, *ie min. over $w$ and $b$*
    
    $$RG(w,b) = S\{(y_i<w|x_i> - b)^2 | i=1,..,L\} + \lambda ||w||^2$$
  
  - **Support Vector Machines (SVM)**, *ie solve directly an optimization problem (hereunder: classif. SVM, separable data)*
    
    *Minimize $||w||^2$,*
    
    *with $(y_i= +/-1)$*
    
    *and $y_i(<w|x_i> + b) >=1$ for all $i=1,..,L$*
The VC Dimension of SVMs

- An SVM finds a linear separator in a Hilbert space, where the original data $x$ can be mapped to via a transformation $\phi(x)$.

- Recall that the kernel trick used by SVM alleviates the need to find explicit expression of $\phi(.)$ to compute the transformation.
The Kernel Trick

- Recall the SVM optimization problem
  \[
  \max_{\alpha} \quad J(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i^T x_j)
  \]

  s.t. \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, k

  \[
  \sum_{i=1}^{m} \alpha_i y_i = 0.
  \]

- The data points only appear as inner product

- As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly

- Define the kernel function \( K \) by
  \[
  K(x_i, x_j) = \phi(x_i)^T \phi(x_j)
  \]
Mercer’s Condition

- For which kernels does there exist a pair \( \{ \mathcal{H}; \phi(.) \} \) with the valid geometric properties (e.g., nonnegative dot-product) for a transformation satisfied, and for which does there not?

- **Mercer’s Condition for Kernels**
  - There exists a mapping \( \phi(.) \) and an expansion

\[
K(x, y) = \sum_i \phi_i(x)\phi_i(y)
\]

iff for any \( g(x) \) such that

\[
\int g(x)^2 dx \text{ is finite}
\]

then

\[
\int K(x, y)g(x)g(y)dx dy \geq 0
\]
The VC Dimension of SVMs

- We will call any kernel that satisfies Mercer’s condition a positive kernel, and the corresponding space $H$ the embedding space.

- We will also call any embedding space with minimal dimension for a given kernel a “minimal embedding space”.

- **Theorem:** Let $K$ be a positive kernel which corresponds to a minimal embedding space $H$. Then the VC dimension of the corresponding support vector machine (where the error penalty $C$ is allowed to take all values) is $\dim(H) + 1$. 
It is striking that the two curves have minima in the same place: thus in this case, the VC bound, although loose, seems to be nevertheless predictive.
What You Should Know

- Sample complexity varies with the learning setting
  - Learner actively queries trainer
  - Examples provided at random

- Within the PAC learning setting, we can bound the probability that learner will output hypothesis with given error
  - For ANY consistent learner (case where c in H)
  - For ANY "best fit" hypothesis (agnostic learning, where perhaps c not in H)

- VC dimension as measure of complexity of H
- Quantitative bounds characterizing bias/variance in choice of H
  - but the bounds are quite loose...

- Mistake bounds in learning