Computational Learning Theory – Part 2

Reading:

- Mitchell chapter 7

Suggested exercises:

- 7.1, 7.2, 7.5, 7.7

Machine Learning 10-701

Tom M. Mitchell
Machine Learning Department
Carnegie Mellon University

March 22, 2010

Computational Learning Theory

What general laws constrain inductive learning?

We seek theory to relate:

- Probability of successful learning
- Number of training examples
- Complexity of hypothesis space
- Accuracy to which target function is approximated
- Manner in which training examples presented

What it means

[Haussler, 1988]: probability that the version space is not ε-exhausted after *m* training examples is at most $|H|e^{-\epsilon m}$

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

↑

Suppose we want this probability to be at most δ

1. How many training examples suffice?

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

2. If $error_{train}(h) = 0$ then with probability at least (1-δ):

$$error_{true}(h) \leq \frac{1}{m}(\ln|H| + \ln(1/\delta))$$

Agnostic Learning

Result we proved: probability, after *m* training examples, that *H* contains a hypothesis *h* with zero training error, but true error greater than ε is bounded

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

Agnostic case: don't know whether H contains a perfect hypothesis

$$\Pr[(\exists h \in H) s.t. (error_{true}(h) > \epsilon + error_{train}(h))] \leq |H|e^{-2\epsilon^2 m}$$

↑ overfitting

General Hoeffding Bounds

- When estimating the mean θ inside [a,b] from *m* examples

$$P(|\hat{\theta} - E[\hat{\theta}]| > \epsilon) \leq 2e^{\frac{-2m\epsilon^2}{(b-a)^2}}$$

- When estimating a probability θ is inside [0,1], so

$$P(|\hat{\theta} - E[\hat{\theta}]| > \epsilon) \leq 2e^{-2m\epsilon^2}$$

- And if we're interested in only one-sided error, then

$$P((E[\hat{\theta}] - \hat{\theta}) > \epsilon) \leq e^{-2m\epsilon^2}$$

PAC Learning

Consider a class C of possible target concepts defined over a set of instances X of length n, and a learner L using hypothesis space H.

Definition: C is **PAC-learnable** by L using H if for all $c \in C$, distributions $\mathcal{D}$ over X, ϵ such that $0 < \epsilon < 1/2$, and δ such that $0 < \delta < 1/2$,
learner L will with probability at least $(1 - \delta)$ output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, n and $size(c)$.

PAC Learning

Consider a class C of possible target concepts defined over a set of instances X of length n, and a learner L using hypothesis space H.

Definition: C is **PAC-learnable** by L using H if for all $c \in C$, distributions $\mathcal{D}$ over X, ϵ such that $0 < \epsilon < 1/2$, and δ such that $0 < \delta < 1/2$,

learner L will with probability at least $(1 - \delta)$ output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, n and $size(c)$.

Sufficient condition:

Holds if L requires only a polynomial number of training examples, and processing per example is polynomial

Sample Complexity based on VC dimension

How many randomly drawn examples suffice to ε-exhaust $VS_{H,D}$ with probability at least (1-δ)?

ie., to guarantee that any hypothesis that perfectly fits the training data is probably (1-δ) approximately (ε) correct

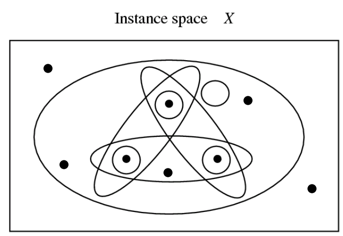
$$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon))$$

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

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

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 $VC(H) \equiv \infty$.



VC(H)=3

VC dimension: examples

What is VC dimension of lines in a plane?

- $H_2 = \{ ((w_0 + w_1x_1 + w_2x_2)>0 \rightarrow y=1) \}$

VC dimension: examples

What is VC dimension of

- $H_2 = \{ ((w_0 + w_1x_1 + w_2x_2)>0 \rightarrow y=1) \}$
  - $VC(H_2)=3$
- For H_n = linear separating hyperplanes in n dimensions, $VC(H_n)=n+1$

Can you give an upper bound on VC(H) in terms of |H|, for any hypothesis space H? (hint: yes)

More VC Dimension Examples to Think About

- Logistic regression over n continuous features
  - Over n boolean features?
- Linear SVM over n continuous features
- Decision trees defined over n boolean features

  F: <X_1, ... X_n> → Y
- Decision trees of depth 2 defined over n features
- How about 1-nearest neighbor?

Tightness of Bounds on Sample Complexity

How many examples *m* suffice to assure that any hypothesis that fits the training data perfectly is probably (1-δ) approximately (ε) correct?

$$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon))$$

How tight is this bound?

Tightness of Bounds on Sample Complexity

How many examples *m* suffice to assure that any hypothesis that fits the training data perfectly is probably (1-δ) approximately (ε) correct?

$$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon))$$

How tight is this bound?

Lower bound on sample complexity (Ehrenfeucht et al., 1989):

Consider any class C of concepts such that VC(C) > 1, any learner L, any 0 < ε < 1/8, and any 0 < δ < 0.01. Then there exists a distribution $\mathcal{D}$ and a target concept in C, such that if L observes fewer examples than

$$\max\left[\frac{1}{\epsilon}\log(1/\delta), \frac{VC(C)-1}{32\epsilon}\right]$$

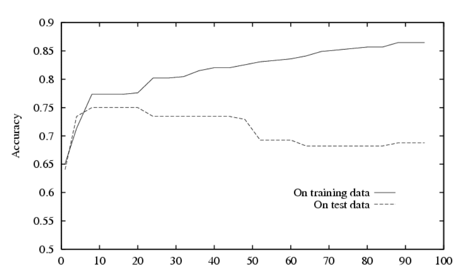
Then with probability at least δ, L outputs a hypothesis with $error_{\mathcal{D}}(h) > \epsilon$

Agnostic Learning: VC Bounds

[Schölkopf and Smola, 2002]

With probability at least (1-δ) every $h \in H$ satisfies

$$error_{true}(h) < error_{train}(h) + \sqrt{\frac{VC(H)(\ln\frac{2m}{VC(H)}+1)+\ln\frac{4}{\delta}}{m}}$$

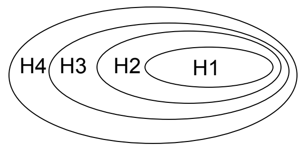


Structural Risk Minimization

[Vapnik]

Which hypothesis space should we choose?

- Bias / variance tradeoff



SRM: choose H to minimize bound on true error!

$$error_{true}(h) < error_{train}(h) + \sqrt{\frac{VC(H)(\ln \frac{2m}{VC(H)} + 1) + \ln \frac{4}{\delta}}{m}}$$

* unfortunately a somewhat loose bound...

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 $\mathcal{D}$
- Learner must classify each instance before receiving correct classification from teacher
- Can we bound the number of mistakes learner makes before converging?

Mistake Bounds: Find-S

Consider Find-S when H = conjunction of boolean literals

FIND-S:

- Initialize h to the most specific hypothesis $l_1 \wedge \neg l_1 \wedge l_2 \wedge \neg l_2 \ldots l_n \wedge \neg l_n$
- For each positive training instance x
  - Remove from h any literal that is not satisfied by x
- Output hypothesis h.

How many mistakes before converging to correct h?

Mistake Bounds: Halving Algorithm

1. Initialize VS ← H
2. For each training example,
   - remove from VS every hypothesis that misclassifies this example

Consider the Halving Algorithm:

- Learn concept using version space CANDIDATE-ELIMINATION algorithm
- Classify new instances by majority vote of version space members

How many mistakes before converging to correct h?

- ... in worst case?
- ... in best case?

Optimal Mistake Bounds

Let $M_A(C)$ be the max number of mistakes made by algorithm A to learn concepts in C. (maximum over all possible $c \in C$, and all possible training sequences)

$$M_A(C) \equiv \max_{c \in C} M_A(c)$$

Definition: Let C be an arbitrary non-empty concept class. The **optimal mistake bound** for C, denoted $Opt(C)$, is the minimum over all possible learning algorithms A of $M_A(C)$.

$$Opt(C) \equiv \min_{A \in learning\ algorithms} M_A(C)$$

$$VC(C) \leq Opt(C) \leq M_{Halving}(C) \leq log_2(|C|).$$

Weighted Majority Algorithm

a_i denotes the i^{th} prediction algorithm in the pool A of algorithms. w_i denotes the weight associated with a_i.

- For all i initialize $w_i \leftarrow 1$
- For each training example $\langle x, c(x) \rangle$
  - Initialize q_0 and q_1 to 0
  - For each prediction algorithm a_i
    - If $a_i(x) = 0$ then $q_0 \leftarrow q_0 + w_i$
      If $a_i(x) = 1$ then $q_1 \leftarrow q_1 + w_i$
  - If $q_1 > q_0$ then predict $c(x) = 1$
    If $q_0 > q_1$ then predict $c(x) = 0$
    If $q_1 = q_0$ then predict 0 or 1 at random for $c(x)$
  - For each prediction algorithm a_i in A do
    If $a_i(x) \neq c(x)$ then $w_i \leftarrow \beta w_i$

when β=0, equivalent to the Halving algorithm…

Weighted Majority

Even algorithms that learn or change over time…

[Relative mistake bound for WEIGHTED-MAJORITY] Let D be any sequence of training examples, let A be any set of n prediction algorithms, and let k be the minimum number of mistakes made by any algorithm in A for the training sequence D. Then the number of mistakes over D made by the WEIGHTED-MAJORITY algorithm using $\beta = \frac{1}{2}$ is at most

$$2.4(k + \log_2 n)$$

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
- Mistake bounds
- Conference on Learning Theory: http://www.learningtheory.org

Extra slides

Training

Input: a labeled training set $\langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\rangle$
number of epochs T

Output: a list of weighted perceptrons $\langle(\mathbf{v}_1, c_1), \ldots, (\mathbf{v}_k, c_k)\rangle$

- Initialize: $k := 0$, $\mathbf{v}_1 := \mathbf{0}$, $c_1 := 0$.
- Repeat T times:
  - For $i = 1, \ldots, m$:
    - * Compute prediction: $\hat{y} := \text{sign}(\mathbf{v}_k \cdot \mathbf{x}_i)$
    - * If $\hat{y} = y$ then $c_k := c_k + 1$.
      else $\mathbf{v}_{k+1} := \mathbf{v}_k + y_i \mathbf{x}_i$;
      $c_{k+1} := 1$;
      $k := k + 1$.

Voted Perceptron

[Freund & Shapire, 1999]

Prediction

Given: the list of weighted perceptrons: $\langle(\mathbf{v}_1, c_1), \ldots, (\mathbf{v}_k, c_k)\rangle$
an unlabeled instance: $\mathbf{x}$

compute a predicted label $\hat{y}$ as follows:

$$s = \sum_{i=1}^{k} c_i \, \text{sign}(\mathbf{v}_i \cdot \mathbf{x}); \qquad \hat{y} = \text{sign}(s) \; .$$

* here y is +1 or -1

Voted Perceptron

[Freund & Shapire, 1999]

Training

Input: a labeled training set $\langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\rangle$
number of epochs T

Output: a list of weighted perceptrons $\langle(\mathbf{v}_1, c_1), \ldots, (\mathbf{v}_k, c_k)\rangle$

- Initialize: $k := 0$, $\mathbf{v}_1 := \mathbf{0}$, $c_1 := 0$.
- Repeat T times:
  - For $i = 1, \ldots, m$:
    - * Compute prediction: $\hat{y} := \text{sign}(\mathbf{v}_k \cdot \mathbf{x}_i)$
    - * If $\hat{y} = y$ then $c_k := c_k + 1$.
      else $\mathbf{v}_{k+1} := \mathbf{v}_k + y_i \mathbf{x}_i$;
      $c_{k+1} := 1$;
      $k := k + 1$.

* y is +1 or -1

Mistake Bounds for Voted Perceptron

When data is linearly separable:

THEOREM 1 (BLOCK, NOVIKOFF) *Let* $\langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\rangle$ *be a sequence of labeled examples with* $||\mathbf{x}_i|| \leq R$. *Suppose that there exists a vector* $\mathbf{u}$ *such that* $||\mathbf{u}|| = 1$ *and* $y_i(\mathbf{u} \cdot \mathbf{x}_i) \geq \gamma$ *for all examples in the sequence. Then the number of mistakes made by the online perceptron algorithm on this sequence is at most* $(R/\gamma)^2$.

Mistake Bounds for Voted Perceptron

When data is linearly separable:

THEOREM 1 (BLOCK, NOVIKOFF) *Let* $\langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\rangle$ *be a sequence of labeled examples with* $||\mathbf{x}_i|| \leq R$. *Suppose that there exists a vector* $\mathbf{u}$ *such that* $||\mathbf{u}|| = 1$ *and* $y_i(\mathbf{u} \cdot \mathbf{x}_i) \geq \gamma$ *for all examples in the sequence. Then the number of mistakes made by the online perceptron algorithm on this sequence is at most* $(R/\gamma)^2$.

When not linearly separable:

THEOREM 2 *Let* $\langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\rangle$ *be a sequence of labeled examples with* $||\mathbf{x}_i|| \leq R$. *Let* $\mathbf{u}$ *be any vector with* $||\mathbf{u}|| = 1$ *and let* $\gamma > 0$. *Define the deviation of each example as*

$$d_i = \max\{0, \gamma - y_i(\mathbf{u} \cdot \mathbf{x}_i)\} ,$$

and define $D = \sqrt{\sum_{i=1}^{m} d_i^2}$. *Then the number of mistakes of the online perceptron algorithm on this sequence is bounded by*

$$\left(\frac{R + D}{\gamma}\right)^2 .$$