

Computational Learning Theory

Pardis Noorzad

Department of Computer Engineering and IT
Amirkabir University of Technology

Ordibehesht 1390

Introduction

- For the analysis of data structures and algorithms and their limits we have:
 - **Computational complexity theory**
 - and analysis of **time** and **space** complexity
 - e.g. Dijkstra's algorithm or Bellman-Ford?
- For the analysis of ML algorithms, there are other questions we need to answer:
 - **Computational learning theory**
 - **Statistical learning theory**

Computational learning theory

(Wikipedia)

- **Probably approximately correct learning** (PAC learning) --Leslie Valiant
 - inspired **boosting**
- **VC theory** --Vladimir Vapnik
 - led to **SVMs**
- **Bayesian inference** --Thomas Bayes
- **Algorithmic learning theory** --E. M. Gold
- **Online machine learning** --Nick Littlestone

Today

- PAC model of learning
 - sample complexity,
 - computational complexity
- VC dimension
 - hypothesis space complexity
- SRM
 - model estimation
- examples throughout ...

PAC learning

(L. Valiant, 1984)

- “Probably learning an approximately correct hypothesis”
- Problem setting:
 - learn an unknown target function c
$$c: X \rightarrow \{0,1\}, \quad c \in \mathcal{C}$$
 - given, training examples $\{x_i, c(x_i)\}$, of this target function
 - $x_i \in X$ i.i.d. according to an unknown but stationary distribution \mathcal{D}
 - given, a space of candidate hypothesis H

PAC learning: measures of error

- ‘**true error**’ of hypothesis f w.r.t. target concept c , and instance distribution \mathcal{D}

$$\text{error}_{\mathcal{D}}(f) = \Pr_{x \sim \mathcal{D}}[c(x) \neq f(x)]$$

– the probability that f will misclassify an instance drawn at random according to \mathcal{D}

- ‘**training error**’: $\text{error}_D(f)$
 - fraction of training samples misclassified by f
 - this is the error that can be observed by our learner L

PAC-learnability: definition

- For concept class C to be PAC-learnable by L
 - we will require that error be bounded by some constant ϵ
 - and it's probability of failure be bounded by some constant δ

Definition: Consider a concept class C defined over a set of instances X of length n and a learner L using hypothesis space H . 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 $size(c)$.

Sample complexity

- For finite hypothesis spaces $|H| < \infty$
 - for a consistent learner
 - for an agnostic learner
- For infinite hypothesis spaces
 - this is where VC dimension comes in
- Our analysis here is **worst-case**, because we demand that learner be general enough to learn any target concept $c \in \mathcal{C}$ regardless of the distribution of training samples \mathcal{D}

Sample complexity: consistent finite $|H|$

- First a reminder of **version space**:
 - the set of all hypotheses that correctly classify the **training samples**
- $$VS_{H,D} = \{f \in H \mid (\forall \langle x, c(x) \rangle \in D)(f(x) = c(x))\}$$
- A **consistent learner** outputs a hypothesis belonging to VS
 - To bound the number of samples needed by a consistent learner, it is enough to bound the number of samples needed to assure that VS contains no unacceptable f

Sample complexity: consistent finite $|H|$ (Haussler, 1988)

- The version space is **ϵ -exhausted** when all consistent hypotheses have true error less than ϵ

- So it bounds the probability that m training examples will fail to eliminate hypotheses with true error greater than ϵ

$$|H|e^{-\epsilon m}$$

Sample complexity: consistent finite $|H|$ (Haussler, 1988)

- Finally! we can determine the number of training samples required to reduce this probability below some desired δ

$$|H|e^{-\epsilon m} \leq \delta$$
$$m \geq \frac{1}{\epsilon} \left(\ln|H| + \ln(1/\delta) \right)$$

- this number of training samples is enough to assure that any consistent hypothesis will be probably $(1 - \delta)$ approximately (ϵ) correct

Sample complexity: inconsistent finite $|H|$

- What if H does not contain target concept c ?
- We want learner to output $f \in H$ that has minimum training error
- Define f_{best} as the hypothesis from H with minimum training error
- How many training samples are needed to ensure that

$$\text{error}_{\mathcal{D}}(f_{\text{best}}) \leq \epsilon + \text{error}_D(f_{\text{best}})$$

Sample complexity: inconsistent finite $|H|$ (Hoeffding bounds, 1963)

- For a single hypothesis to have a misleading training error

$$\Pr[\text{error}_{\mathcal{D}}(f) > \epsilon + \text{error}_D(f)] \leq e^{-2m\epsilon^2}$$

- We want to assure that the best hypothesis has an error bounded this way

– so consider that any one of them could have a large error

$$\Pr[(\exists f \in H)\text{error}_{\mathcal{D}}(f) > \epsilon + \text{error}_D(f)] \leq |H|e^{-2m\epsilon^2}$$

Sample complexity: inconsistent finite $|H|$ (Hoeffding bounds, 1963)

- So if do the stuff we did before, call that probability δ , we can find a bound for the number of samples needed to hold δ to some desired value

$$m \geq \frac{1}{2\epsilon^2} \left(\ln|H| + \ln(1/\delta) \right)$$

Sample complexity: example

- C : conjunction of n boolean literals
- is C PAC-learnable?
- So $|H| = 3^n$ and

$$m \geq \frac{1}{\epsilon} \left(n \ln 3 + \ln(1/\delta) \right)$$

$$m = \frac{1}{.1} \left(10 \ln 3 + \ln(1/.05) \right) = 140$$

- m grows linearly in n , ϵ , and logarithmically in $1/\delta$
- So as long as L requires no more than polynomial computation per training sample, then total computation required will be polynomial as well.

Sample complexity: infinite $|H|$

- using $|H|$ leads to weak bounds
- and in case of $|H| = \infty$ we cannot apply it at all
- so we define a second measure of complexity called **VC dimension**
- in many cases it provides tighter bounds
- note (I will explain later):

$$VC(H) \leq \log_2 |H|$$

VC theory

(V. Vapnik and A. Chervonenkis, 1960-1990)

- **VC dimension**
- **Structural risk minimization**

VC dimension

(V. Vapnik and A. Chervonenkis, 1968, 1971)

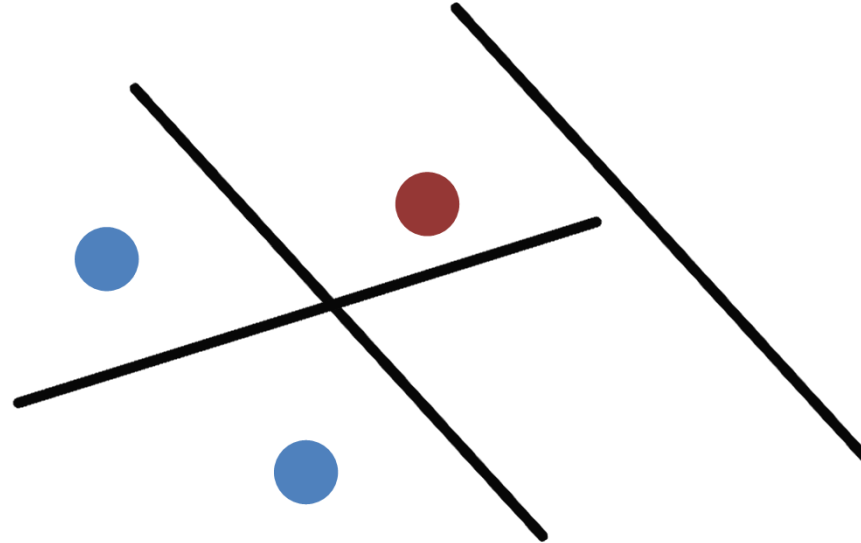
- First we'll define **shattering**
- Consider hypotheses for the two-class pattern recognition problem:

$$f(\mathbf{x}, \alpha) \in \{-1, 1\} \quad \forall \mathbf{x}, \alpha$$

- Now, if for a set of N points that can be labeled in all 2^N ways (either + or -),
 - a member of the set $\{f(\alpha)\}$ can be found which correctly assigns those labels...
 - we say, that set of points is **shattered** by $\{f(\alpha)\}$

VC dimension: example

three points in \mathbb{R}^2 , shattered by oriented lines



- For our purposes, it is enough to find **one set of three points** that can be shattered
- It is not necessary to be able to shatter **every possible set of three points** in 2 dimensions



VC dimension: continued

- The **maximum number of points** that can be shattered by $H = \{f(\alpha)\}$ is called
 - **VC dimension of H**
 - and denoted as $VC(H)$
- So the VC dimension of the set of oriented lines in \mathbb{R}^2 is three (last example)
- $VC(H)$ is a measure of the **capacity** of the hypothesis class H
 - the higher the capacity, the higher the ability of the machine to **learn any training set without error**

VC dimension: intuition

- **High capacity:**
 - not a tree, b/c different from any tree I have seen (e.g. different number of leaves)
- **Low capacity:**
 - if it's **green** then it's a tree



Low VC dimension

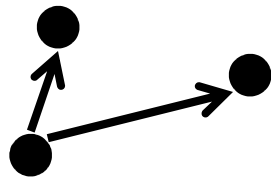
- VC dimension is pessimistic
- If using **oriented lines** as our hypothesis class
 - we can only learn datasets with **3 points!**
- This is b/c VC dimension is computed independent of distribution of instances
 - In reality, **near neighbors have same labels**
 - So no need to worry about **all possible labelings**
- There are a lot of datasets containing more points that are **learnable by a line!**

Infinite VC dimension

- Lookup table has infinite VC dimension
- But so does the nearest neighbor classifier
 - b/c any number of points, labeled arbitrarily (w/o overlaps), will be successfully learned
- But it performs well...
- So infinite capacity **does not guarantee** poor generalization performance

Examples of calculating VC dimension

- So we saw that the VC dimension of the set of oriented lines in \mathbb{R}^2 is 3
- Generally, the VC dimension of the set of oriented hyperplanes in \mathbb{R}^n is $n + 1$



Examples of calculating VC dimension: continued

- Let K be a positive kernel which corresponds to a minimal embedding space \mathcal{H} .
- Then the VC dimension of the corresponding support vector machine is $\dim(\mathcal{H}) + 1$.
- Proof...

VC dimension of SVMs with polynomial kernels

- e.g. $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^2$

- if \mathbf{x} and \mathbf{y} are 2-dimensional:

$$k(\mathbf{x}, \mathbf{y}) = (x_1y_1 + x_2y_2)^2 = x_1^2y_1^2 + 2x_1y_1x_2y_2 + x_2^2y_2^2$$

- the feature space is 3-dimensional

- and the VC dimension of an SVM with this kernel is $3 + 1 = 4$

- in general, for a space with dimension d , the dimension of the embedding space for homogeneous polynomial kernels is

$$\binom{d + p - 1}{p}$$

Sample complexity and the VC dimension

(Blumer et al., 1989)

$$m \geq \frac{1}{\epsilon} \left(4 \log_2 \left(\frac{2}{\delta} \right) + 8h \log_2 \left(\frac{13}{\epsilon} \right) \right)$$

- where $h = VC(H)$

Structural risk minimization

(V. Vapnik and A. Chervonenkis, 1974)

- A function that fits training data and minimizes VC dimension
 - will **generalize well**
- SRM provides a quantitative characterization of the **tradeoff** between
 - the complexity of the approximating functions,
 - and the quality of fitting the training data
- First, we will talk about a certain bound..

A bound

- on the generalization performance of a pattern recognition learning machine
 - from Burges, 1998

**true
mean
error/
actual
risk**

$$R(\alpha) = \int \frac{1}{2} |y - f(\mathbf{x}, \alpha)| dP(\mathbf{x}, y)$$

$$R(\alpha) = \int \frac{1}{2} |y - f(\mathbf{x}, \alpha)| p(\mathbf{x}, y) d\mathbf{x} dy$$

$$R_{emp}(\alpha) = \frac{1}{2N} \sum_{i=1}^N |y_i - f(\mathbf{x}_i, \alpha)|$$

**empirical
risk**

A bound: continued

$$R(\alpha) \leq R_{emp}(\alpha)$$

$$+ \sqrt{\left(\frac{h(\log(2N/h) + 1) - \log(\eta/4)}{N} \right)}$$

**VC
confidence**

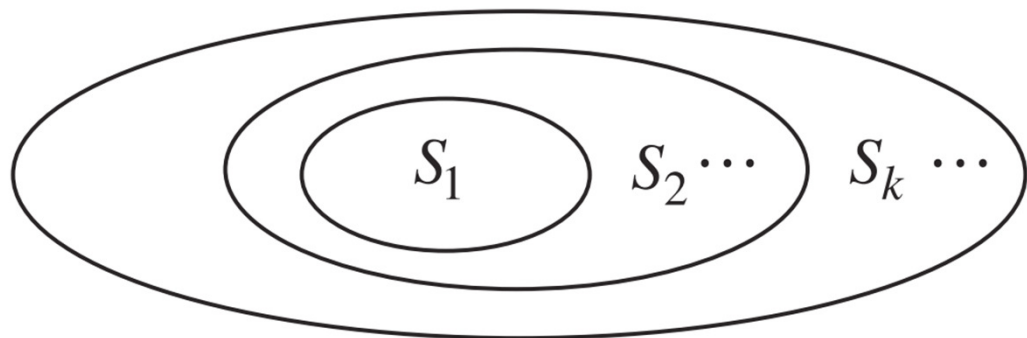
risk bound

SRM: continued

- To minimize true error (actual risk), both empirical risk and VC confidence term should be minimized
- The VC confidence term depends on the chosen **class of functions**
- Whereas empirical risk and actual risk depend on the **one particular function** chosen for training

SRM: continued

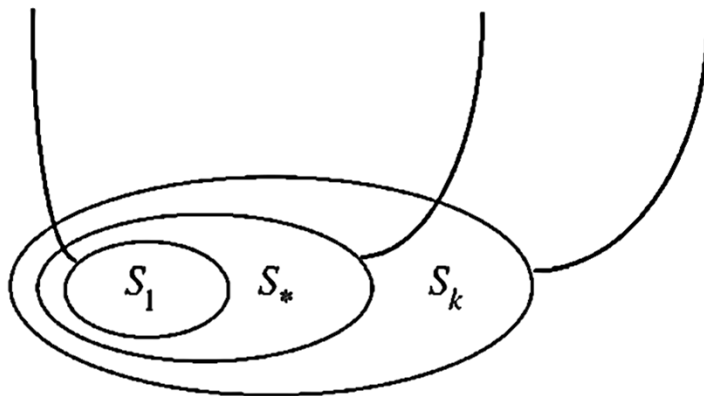
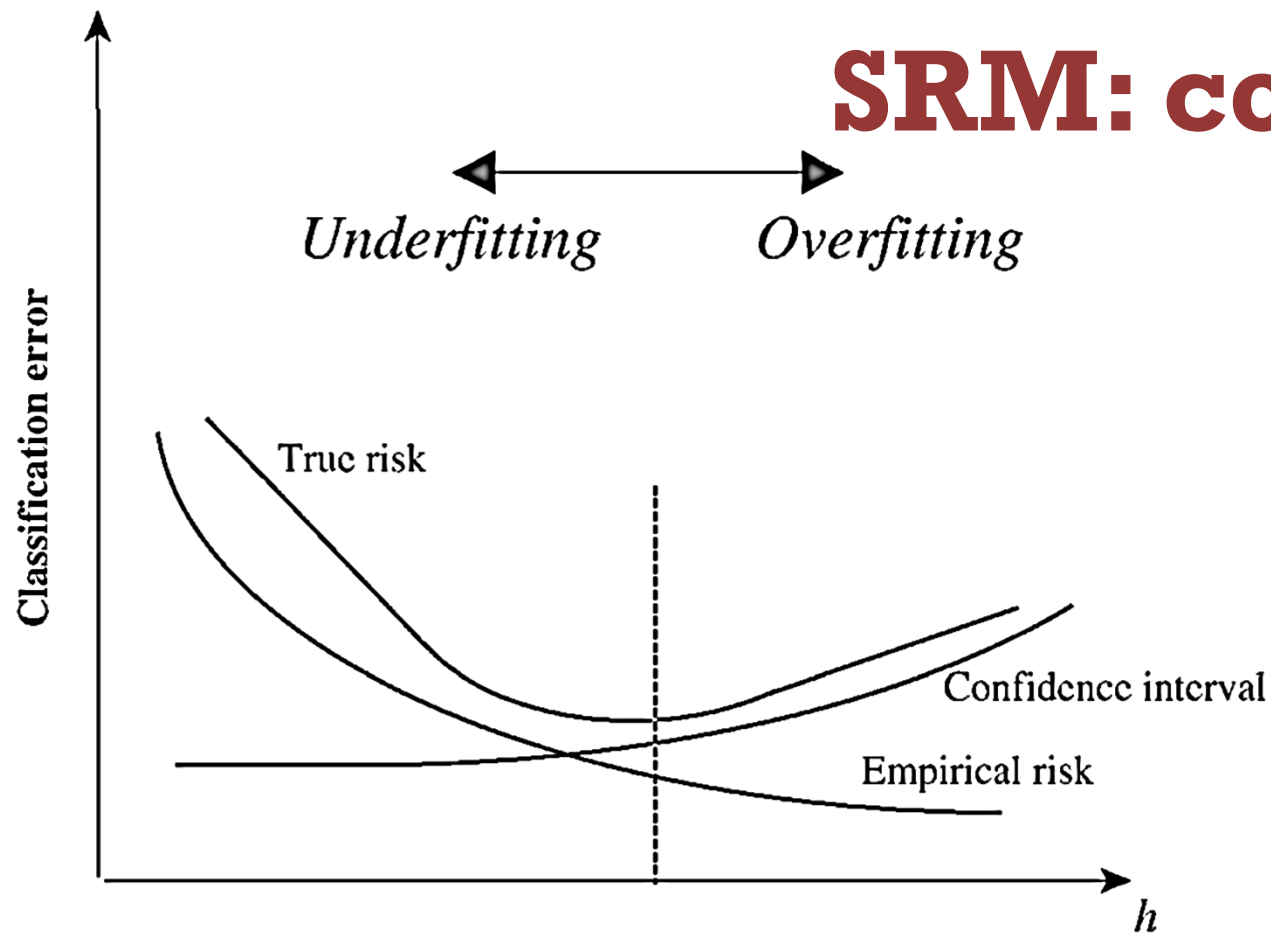
- The VC dimension h doesn't vary smoothly since it is an integer
 - Therefore the entire class of functions is structured into nested subsets (ordered by $h, h < \infty$)
- $$h_1 \leq h_2 \leq \dots \leq h_k \leq \dots$$



SRM: continued

- For a given data set, **optimal model estimation** with SRM consists of the following steps:
 1. select an element of a structure (model selection)
 2. estimate the model from this element (statistical parameter estimation)

SRM: continued



SRM chooses the subset S_i for which minimizing the empirical risk yields the best bound on the true risk.

Support vector machine

(Vapnik, 1982)

- Does SVM implement the SRM principle?
- We have shown that the VC dimension of a nonlinear SVM is $\dim(\mathcal{H}) + 1$, where $\dim(\mathcal{H})$ is the dimension of space \mathcal{H}
- and equal to $\binom{d_L + p - 1}{p}$ and ∞ for a p -degree polynomial and RBF kernel, respectively
 - So SVM can have very high VC dimension
- But, it is possible to prove that SVM training actually minimizes the VC dimension and empirical error at the same time.

Support vector machine

(Vapnik, 1995)

- Given m data points in \mathbb{R}^n , $\|x_i\| \leq R$
- H_γ : set of linear classifiers in \mathbb{R}^n with margin γ on X
- Then

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

- This means that hypothesis spaces with large margin have small VC dimension
- (Burges, 1998) claimed that this bound is for a gap-tolerant classifier but not the SVM classifier...

What we said today

- PAC bounds
 - only apply to finite hypothesis spaces
- VC dimension is a measure of complexity
 - can be applied for infinite hypothesis spaces
- Training neural networks uses only ERM, whereas SVMs use SRM
- Large margins imply small VC dimension
😊

References

1. Machine Learning by T. M. Mitchell
2. A Tutorial on Support Vector Machines for Pattern Recognition by C. J. C. Burges
3. <http://www.svms.org/>



Thanks!
Questions?