Lecture 9: Learning Theory

Feng Li

Shandong University

fli@sdu.edu.cn

January 5, 2022

Outline

Why Learning Theory?

- How can we tell if your learning algorithm will do a good job in future (test time)?
  - Experimental results
  - Theoretical analysis
- Why theory?
  - Can only run a limited number of experiments..
  - Experiments rarely tell us what will go wrong
- Using learning theory, we can make formal statements/give guarantees on
  - Expected performance ("generalization") of a learning algorithm on test data
  - Number of examples required to attain a certain level of test accuracy
  - Hardness of learning problems in general

Bias, Variance and Model Complexity

- Bias is a learner's tendency to consistently learn the same wrong thing
  - The bias is error from erroneous assumptions in the learning algorithm
  - High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting)
- Variance is the tendency to learn random things irrespective of the real signal
  - The variance is error from sensitivity to small fluctuations in the training set
  - High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting)

Bias, Variance and Model Complexity (Contd.)

- A target variable Y, a vector of inputs X and a prediction model $\hat{f}(X)$ which has been estimated from a training set $\mathcal{D}$
- The loss function for measuring errors between Y and $\hat{f}(X)$

$$L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2, & \text{squared error} \\ |Y - \hat{f}(X)|, & \text{absolute error} \end{cases}$$

Bias, Variance and Model Complexity (Contd.)

- Test error (or generalization error)

$$\mathrm{Err}_{\mathcal{D}} = \mathbb{E}[L(Y, \hat{f}(X)) \mid \mathcal{D}]$$

- Expected prediction (or test) error

$$\mathrm{Err} = \mathbb{E}[L(Y, \hat{f}(X))] = \mathbb{E}[\mathrm{Err}_{\mathcal{D}}]$$

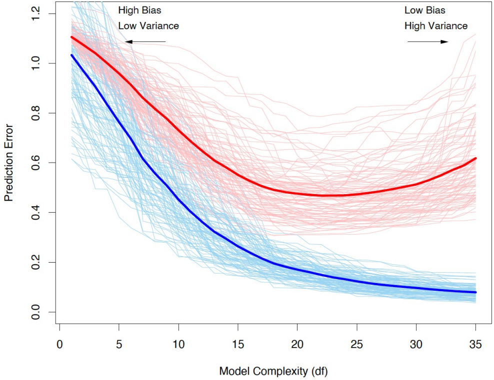
  which averages over everything that is random, including the randomness in the training set that produced $\hat{f}$

- Training error

$$\overline{\mathrm{err}} = \frac{1}{m} \sum_{i=1}^{m} L(y_i, \hat{f}(x_i))$$

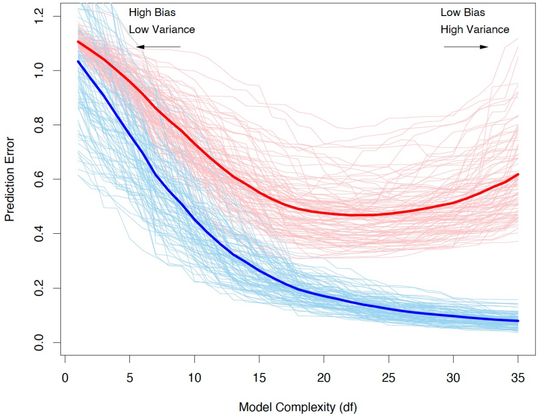
Bias, Variance and Model Complexity (Contd.)

Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error $\overline{\mathrm{Err}}$, while the light red curves show the conditional test error $\mathrm{Err}_{\mathcal{D}}$ for 100 training sets of size 50 each, as the model complexity is increased. The solid curves show the expected test error Err and the expected training error $\mathbb{E}[\overline{\mathrm{Err}}]$.

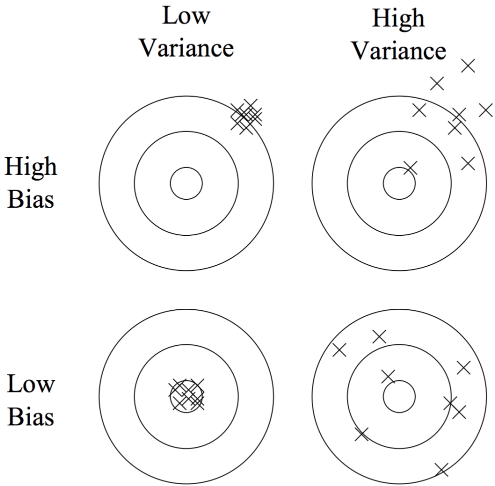


Bias, Variance and Model Complexity (Contd.)

Training error is not a good estimate of the test error!

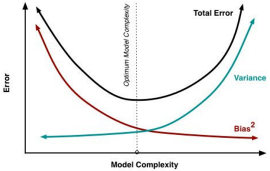


Bias, Variance and Model Complexity (Contd.)



Bias, Variance and Model Complexity (Contd.)

- Simple model have high bias and small variance, complex models have small bias and high variance

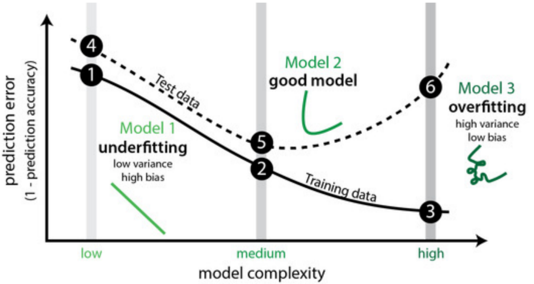


	Model				
	Bias	Variance	Complexity	Flexibility	Generalizability
Underfitting: you have an overly simple model	High	Low	Low	Low	High
Overfitting: your model is modelling the noise	Low	High	High	High	Low

- If you modified a model to reduce its bias (e.g., by increasing the model's complexity), you are likely to increase its variance, and vice-versa (if, however, both increase then you might be doing it wrong!)

Bias, Variance and Model Complexity (Contd.)

- The bad performance (low accuracy on test data) could be due to either high bias (underfitting) or high variance (overfitting)
- Looking at the training and test error can tell which of the two is the case



- High bias: Both training and test error are large
- High variance: Small training error, large test error (and huge gap)

Bias, Variance and Model Complexity (Contd.)

- **Model section**: Estimating the performance of different models in order to choose the best one
- **Model assessment**: Having chosen a final model, estimating its prediction error (generalization error) on new data.
- If we are in a data-rich situation, the best approach for both problems is to randomly divide the dataset into three parts: a *training* set, a *validation* set, and a *test* set.
  - The training set is used to fit the models
  - The validation set is used to estimate prediction error for model selection
  - The test set is used for assessment of the generalization error of the final chosen model
- A typical split might be 50% for training, and 25% each for validation and testing

Bias-Variance Decomposition

- For a model $Y = f(X) + \epsilon$ with $\mathbb{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma_\epsilon^2$

$$
\begin{aligned}
\mathrm{Err}(x_0) &= \mathbb{E}[(y_0 - \hat{f}(x_0))^2] \\
&= \mathbb{E}[y_0^2 - 2y_0\hat{f}(x_0) + \hat{f}^2(x_0)] \\
&= \mathbb{E}[y_0^2] + \mathbb{E}[\hat{f}^2(x_0)] - \mathbb{E}[2y_0\hat{f}(x_0)] \\
&= \mathrm{Var}[y_0] + \mathbb{E}^2[y_0] + \mathrm{Var}[\hat{f}(x_0)] + \mathbb{E}^2[\hat{f}(x_0)] - \mathbb{E}[2y_0\hat{f}(x_0)] \\
&= \mathrm{Var}[f(x_0) + \epsilon] + \mathbb{E}^2[f(x_0) + \epsilon] + \mathrm{Var}[\hat{f}(x_0)] \\
&\quad + \mathbb{E}^2[\hat{f}(x_0)] - \mathbb{E}[2(f(x_0) + \epsilon)\hat{f}(x_0)] \\
&= \sigma_\epsilon^2 + f^2(x_0) + \mathrm{Var}[\hat{f}(x_0)] + \mathbb{E}^2[\hat{f}(x_0)] - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] \\
&= \sigma_\epsilon^2 + (f(x_0) - E[\hat{f}(x_0)])^2 + \mathrm{Var}[\hat{f}(x_0)] \\
&= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}
\end{aligned}
$$

Bias-Variance Decomposition (Contd.)

- For a model $Y = f(X) + \epsilon$ with $\mathbb{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma_\epsilon^2$

$$
\begin{aligned}
\mathrm{Err}(x_0) &= \mathbb{E}[(Y - \hat{f}(x_0))^2] \\
&= \sigma_\epsilon^2 + (f(x_0) - E[\hat{f}(x_0)])^2 + \mathrm{Var}[\hat{f}(x_0)] \\
&= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}
\end{aligned}
$$

  - The first term is the variance of the target around its true mean $f(x_0)$, and cannot be avoided no matter how well we estimate $f(x_0)$, unless $\sigma_\epsilon^2 = 0$
  - The second term is the squared bias, i.e., the amount by which the average of our estimate differs from the true mean
  - The last term is the variance, i.e., the expected squared deviation of $\hat{f}(x_0)$ around its mean

Bias-Variance Decomposition (Contd.)

- For k-nearest neighbor regression,

$$
\begin{aligned}
\mathrm{Err}(x_0) &= \mathbb{E}[(Y - \hat{f}_k(x_0))^2] \\
&= \sigma_\epsilon^2 + \left[f(x_0) - \frac{1}{k} \sum_{\ell=1}^{k} f(x_{(\ell)}) \right]^2 + \frac{\sigma_\epsilon^2}{k}
\end{aligned}
$$

Bias-Variance Decomposition (Contd.)

- For linear regression model $Y = X\theta + \epsilon$,

$$
\begin{aligned}
\text{Bias}(x_0) &= f(x_0) - \mathbb{E}[\hat{f}(x_0)] \\
&= x_0^T\theta - \mathbb{E}[x_0^T\hat{\theta}] \\
&= x_0^T\theta - \mathbb{E}[x_0^T(X^TX)^{-1}X^TY] \\
&= x_0^T\theta - \mathbb{E}[x_0^T(X^TX)^{-1}X^T(X\theta + \epsilon)] \\
&= x_0^T\theta - \mathbb{E}[x_0^T(X^TX)^{-1}X^TX\theta + x_0^T(X^TX)^{-1}X^T\epsilon] \\
&= x_0^T\theta - \mathbb{E}[x_0^T\theta + x_0^T(X^TX)^{-1}X^T\epsilon] \\
&= \mathbb{E}[x_0^T\theta - x_0^T\theta + x_0^T(X^TX)^{-1}X^T\epsilon] \\
&= 0
\end{aligned}
$$

Bias-Variance Decomposition (Contd.)

- For linear regression model $Y = X\theta + \epsilon$,

$$
\begin{aligned}
\mathrm{Var}(\hat{f}(x_0)) &= \mathbb{E}[(f(x_0) - \mathbb{E}[\hat{f}(x_0))^2] \\
&= \mathbb{E}[(x_0^T(X^TX)^{-1}X^TY - X_0^T\theta)^2] \\
&= \mathbb{E}[(x_0^T(X^TX)^{-1}X^T(X\theta+\epsilon) - X_0^T\theta)^2] \\
&= \mathbb{E}[(x_0^T(X^TX)^{-1}X^T\epsilon)^2] \\
&= \mathbb{E}[(x_0^T(X^TX)^{-1}X^T\epsilon)(x_0^T(X^TX)^{-1}X^T\epsilon)^T] \\
&= \mathbb{E}[x_0^T(X^TX)^{-1}X^T\epsilon\epsilon^T(x_0^T(X^TX)^{-1}X^T)^T] \\
&= x_0^T(X^TX)^{-1}X^T\mathbb{E}[\epsilon\epsilon^T](x_0^T(X^TX)^{-1}X^T)^T \\
&= x_0^T(X^TX)^{-1}X^T\sigma_\epsilon^2 I(x_0^T(X^TX)^{-1}X^T)^T \\
&= \sigma_\epsilon^2 x_0^T(X^TX)^{-1}X^T(x_0^T(X^TX)^{-1}X^T)^T \\
&= \sigma_\epsilon^2 x_0^T(X^TX)^{-1}x_0 \\
&\approx \sigma_\epsilon^2\frac{n}{m}
\end{aligned}
$$

- **The union bound**
  Assume $A_1, A_2, \cdots, A_k$ be k different events (that may not be independent),

$$p(A_1 \bigcup A_2 \cdots \bigcup A_k) \leq p(A_1) + \cdots + p(A_k)$$

- **Hoeffding inequality (Chernoff bound)**
  Let $Z_1, \cdots, Z_m$ be m independent and identically distributed (iid) random variables drawn from a Bernoulli(ϕ) distribution (i.e., $p(Z_i = 1) = \phi$ and $p(Z_i = 0) = 1 - \phi$). Let $\hat{\phi} = \frac{1}{m} \sum_{i=1}^{m} Z_i$ be the mean of these random variables, and let any $\gamma > 0$ be fixed. Then

$$p(|\phi - \hat{\phi}| > \gamma) \leq 2\exp(-2\gamma^2 m)$$

Hypothesis Class

- A hypothesis class $\mathcal{H}$: a set of all classifiers considered by a learning algorithm
- A training set $S = \{(x^{(i)}, y^{(i)})\}_{i=1,\cdots,m}$ with $y^{(i)} \in \{0, 1\}$ are drawn i.i.d. from some probability distribution $\mathcal{D}$
- The learning algorithm, given training data, learns a hypothesis $h \in \mathcal{H}$

Training and Generalization Error

- The training error (or empirical risk, empirical error) is

$$\overline{\mathsf{Err}}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{h(x^{(i)}) \neq y^{(i)}\}$$

  i.e., the fraction of the misclassified training examples

- The generalization is

$$\mathsf{Err}_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}(h(x) \neq y)$$

  i.e., the probability that, if we now draw a new example (x, y) from the distribution $\mathcal{D}$, h will misclassify it

Empirical Risk Minimization

- Empirical Risk Minimization (ERM)
  - Consider a linear classification $h_\theta(x) = \mathbf{1}(\theta^T x \geq 0)$
  - Minimize the training error

$$\theta^* = \arg\min_{\theta} \overline{\text{Err}}(h_\theta)$$

  - Optimal hypothesis $h^* = h_{\theta^*}$
- ERM can also be thought of a minimization over the class

$$h^* = \arg\min_{h \in \mathcal{H}} \overline{\text{Err}}(h)$$

Finite $\mathcal{H}$

- A finite hypothesis class $\mathcal{H} = \{h_1, \cdots, h_k\}$
- $h^* \in \mathcal{H}$ denotes the optimal hypothesis function with the training error minimized by ERM
- Does there exist a guarantee on the generalization error of $\hat{h}$?
  - $\text{Err}_{\mathcal{D}}(h)$ is a reliable estimate of $\text{Err}(h)$ for $\forall h$
  - This implies an upper-bound on the generalization error of h^*

Finite $\mathcal{H}$ (Contd.)

- Assume $(x, y) \sim \mathcal{D}$
- For $h_i \in \mathcal{H}$, define Bernoulli random variables

$$
\begin{aligned}
Z &= \mathbf{1}(h_i(x) \neq y) \\
Z_j &= \mathbf{1}\{h_i(x^{(j)}) \neq y^{(j)}\}
\end{aligned}
$$

- The generalization error

$$
\mathrm{Err}_{\mathcal{D}}(h_i) = \mathbb{E}[Z] = \mathbb{E}[Z_j]
$$

- The training error

$$
\overline{\mathrm{Err}}(h_i) = \frac{1}{m} \sum_{j=1}^{m} Z_j
$$

Finite $\mathcal{H}$ (Contd.)

- Assume $(x, y) \sim \mathcal{D}$
- For $h_i \in \mathcal{H}$, define Bernoulli random variables

$$Z = \mathbf{1}(h_i(x) \neq y)$$
$$Z_j = \mathbf{1}\{h_i(x^{(j)}) \neq y^{(j)}\}$$

- The generalization error $\mathrm{Err}_{\mathcal{D}}(h_i) = \mathbb{E}[Z] = \mathbb{E}[Z_j]$
- The training error $\overline{\mathrm{Err}}(h_i) = \frac{1}{m} \sum_{j=1}^{m} Z_j$
- By applying Hoeffding inequality, we have

$$P(|\overline{\mathrm{Err}}(h_i) - \mathrm{Err}_{\mathcal{D}}(h_i)| > \gamma) \leq 2\exp(-2\gamma^2 m)$$

- For a particular h_i, training error will be close to generalization error with high probability, assuming m is large

Finite $\mathcal{H}$ (Contd.)

- Let A_i denote the event that $|\overline{\text{Err}}(h_i) - \text{Err}_{\mathcal{D}}(h_i)| > \gamma$, then

$$\mathbb{P}(A_i) \leq 2\exp(-2\gamma^2 m)$$

- By using the union bound, we have

$$\begin{aligned} & \mathbb{P}(|\overline{\text{Err}}(h_i) - \text{Err}_{\mathcal{D}}(h_i)| > \gamma) \\ = \; & \mathbb{P}(A_1 \bigcup \cdots \bigcup A_k) \leq \sum_{i=1}^{k} P(A_i) \\ \leq \; & \sum_{i=1}^{k} 2\exp(-2\gamma^2 m) = 2k\exp(-2\gamma^2 m) \end{aligned}$$

Finite $\mathcal{H}$ (Contd.)

- Then, we have the following result

$$
\begin{aligned}
& \mathbb{P}(\neg\exists h \in \mathcal{H} : |\overline{\mathrm{Err}}(h) - \mathrm{Err}_{\mathcal{D}}(h)| > \gamma) \\
= \; & \mathbb{P}(\forall h \in \mathcal{H} : |\overline{\mathrm{Err}}(h) - \mathrm{Err}_{\mathcal{D}}(h)| > \gamma) \\
\geq \; & 1 - 2k\exp(-2\gamma^2 m)
\end{aligned}
$$

- With probability at least $1 - 2k\exp(-2\gamma^2 m)$, we have

$$|\overline{\mathrm{Err}}(h) - \mathrm{Err}_{\mathcal{D}}(h)| \leq \gamma$$

for $\forall h \in \mathcal{H}$

Finite $\mathcal{H}$ (Contd.)

Given γ and $\delta > 0$, how large should m be such that we can guarantee

$$|\overline{\mathrm{Err}}(h) - \mathrm{Err}_{\mathcal{D}}(h)| \leq \gamma$$

with probability $\geq 1 - \delta$?

- Solution

$$1 - 2k\exp(-2\gamma^2 m) \geq 1 - \delta \Rightarrow m \geq \frac{1}{2\gamma^2}\log\frac{2k}{\delta}$$

- The training set size m that a certain method or algorithm requires in order to achieve a certain level of performance is so-called the algorithm's sample complexity
- The number of training examples needed to make this guarantee is only logarithmic in the number of hypotheses in $\mathcal{H}$ (i.e., k)

Finite $\mathcal{H}$ (Contd.)

- Fixing m and δ, solving for γ gives

$$1 - 2k\exp(-2\gamma^2 m) \geq 1 - \delta \Rightarrow |\overline{\text{Err}}(h) - \text{Err}_{\mathcal{D}}(h)| \leq \sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

Given m and $\delta > 0$, with probability at least $1 - \delta$,

$$|\overline{\text{Err}}(h) - \text{Err}_{\mathcal{D}}(h)| \leq \sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

Finite $\mathcal{H}$ (Contd.)

- Assume $\hat{h} = \arg\min_{h \in \mathcal{H}} \mathrm{Err}_{\mathcal{D}}(h)$

$$\begin{aligned} \mathrm{Err}_{\mathcal{D}}(h^*) &\leq \overline{\mathrm{Err}}(h^*) + \gamma \\ &\leq \overline{\mathrm{Err}}(\hat{h}) + \gamma \\ &\leq \mathrm{Err}_{\mathcal{D}}(\hat{h}) + 2\gamma \end{aligned}$$

- If uniform convergence occurs, then the generalization error of h^* is at most 2γ worse than the best possible hypothesis in $\mathcal{H}$

Finite $\mathcal{H}$ (Contd.)

Theorem

Let $|\mathcal{H}| = k$ and let any m and δ be fixed. With probability at least $1 - \delta$, we have

$$\mathrm{Err}_{\mathcal{D}}(h^*) \leq \left(\min_{h \in \mathcal{H}} \mathrm{Err}_{\mathcal{D}}(h) \right) + 2\sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}$$

- If we take a larger hypothesis set $\mathcal{H}'$ such that $\mathcal{H} \subseteq \mathcal{H}'$
  - the first term is decreased (the bias is decreased)
  - the second term is increased (the variance is increased)

Finite $\mathcal{H}$ (Contd.)

Corollary

Let $|\mathcal{H}| = k$ *and let any* δ, γ *be fixed. For*

$$\mathrm{Err}_{\mathcal{D}}(h^*) \leq \min_{h \in \mathcal{H}} \mathrm{Err}_{\mathcal{D}}(h) + 2\gamma$$

to hold with probability at least $1 - \delta$, *it suffices that*

$$\begin{aligned} m &\geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta} \\ &= O\left(\frac{1}{\gamma^2} \log \frac{k}{\delta}\right) \end{aligned}$$

Infinite $\mathcal{H}$

- What happens when the hypothesis class size $|\mathcal{H}|$ is infinite?
  - Example: The set of all linear classifiers
- The above bound does not 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
  - Vapnik-Chervonenkis dimension (VC dimension)
  - VC dimension: a measure of the complexity of a hypothesis class

Shattering

- A set of points (in a given configuration) is shattered by a hypothesis class $\mathcal{H}$, if, no mater how the points are labeled, there exists some $h \in \mathcal{H}$ that can separate the points

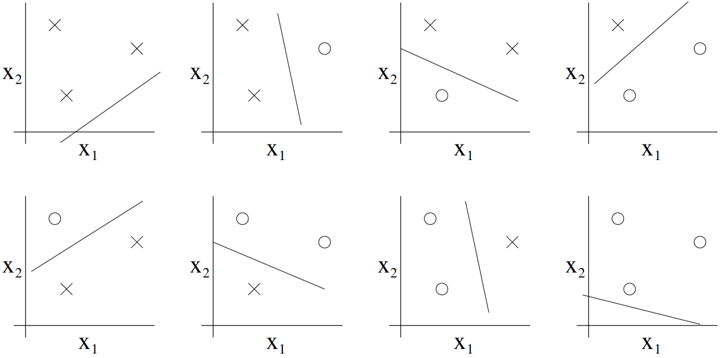


Figure: 3 points in 2D (locations fixed, only labeling varies), $\mathcal{H}$: set of linear classifier

Vapnik-Chervonenkis (VC) Dimension

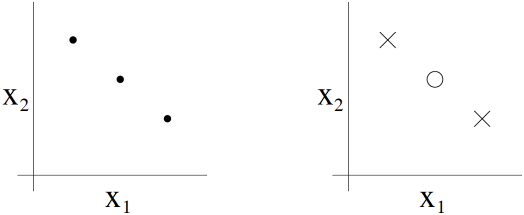
Definition (VC Dimension)

Given a hypothesis class $\mathcal{H}$, we then define its Vapnik-Chervonenkis dimension, VC($\mathcal{H}$), to be the size of the largest set that is shattered by $\mathcal{H}$

- Consider the following shattering game between us and an adversary
  - We choose d points in an input space, positioned however we want
  - Adversary labels these d points
  - We define a hypothesis $h \in \mathcal{H}$ that separates the points
  - Note: Shattering just one configuration of d points is enough to win
- 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

VC Dimension (Contd.)

- Even when $\text{VC}(\mathcal{H}) = 3$, there exist sets of size 3 that cannot be classified correctly



- In order words, under the definition of the VC dimension, in order to prove that $\text{VC}(\mathcal{H})$ is at least d, we need to show only that there's at least one set of size d that $\mathcal{H}$ can shatter.

VC Dimension (Contd.)

- A measure of the "power" or the "complexity" of the hypothesis space
  - Higher VC dimension implies a more "expressive" hypothesis space)
- Shattering: A set of N points is shattered if there exists a hypothesis that is consistent with every classification of the N points
- VC Dimension: The maximum number of data points that can be "shattered"
- If VC Dimension = d, then:
  - There exists a set of d points that can be shattered
  - There does not exist a set of $d+1$ points that can be shattered

VC Dimension (Contd.)

Theorem

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

$$|\mathrm{Err}_{\mathcal{D}}(h) - \overline{\mathrm{Err}}(h)| \leq O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}}\right)$$

and thus

$$\mathrm{Err}_{\mathcal{D}}(h^*) \leq \overline{\mathrm{Err}}(\hat{h}) + O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}}\right)$$

VC Dimension (Contd.)

- Recall for finite hypothesis space

$$\mathrm{Err}_{\mathcal{D}}(h^*) \leq \left(\min_{h \in \mathcal{H}} \mathrm{Err}_{\mathcal{D}}(h) \right) + 2\sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}$$

- $\mathrm{VC}(H)$ is like a substitute for $k = |\mathcal{H}|$

Select The Right Model

- Given a set of models $M = \{M_1, M_2, ..., M_R\}$, choose the model that is expected to do the best on the test data
- M may consist of:
  - Same learning model with different complexities or hyperparameters
    - Nonlinear Regression: Polynomials with different degrees
    - K-Nearest Neighbors: Different choices of K
    - Decision Trees: Different choices of the number of levels/leaves
    - SVM: Different choices of the misclassification penalty parameter C
    - Regularized Models: Different choices of the regularization parameter
    - Kernel based Methods: Different choices of kernels
    - ... and almost any learning problem
  - Different learning models (e.g., SVM, KNN, DT, etc.)

Hold-Out Cross Validation (Simple Cross Validation)

- Given a training set S, do the following
  - Randomly split S into S_{train} (say, 70% of the data) and S_{cv} (the remaining 30% called the hold-out cross validation set)
  - Train each model M_i on S_{train} only, to get some hypothesis h_i.
  - Select and output the hypothesis h_i that had the smallest error $\overline{\text{Err}}_{S_{cv}}(h_i)$ on the hold-out cross validation set
- Option: After selecting $M^* \in \mathcal{M}$ such that $h^* = \arg\min_i \overline{\text{Err}}_{S_{cv}}(h_i)$, retrain M^* on the entire training set S
- Weakness: It seems we are trying to select the best model based on only part of the training set

k-Fold Cross Validation

- Randomly split S into k disjoint subsets $S_1, \cdots, S_k$, each of which involves m/k training examples
- For each model M_i, we evaluate it as follows:
  - For $j = 1, \cdots, k$, train the model M_i on $S_1 \bigcup \cdots \bigcup S_{j-1} \bigcup S_{j+1} \bigcup \cdots \bigcup S_k$ (i.e., train on all the data except S_j) to get some hypothesis h_{ij}, and then test the hypothesis h_{ij} on S_j, to get $\widehat{\mathrm{Err}}_{S_j}(h_{ij})$.
  - The estimated generalization error of model M_i is then calculated as the average of the $\widehat{\mathrm{Err}}_{S_j}(h_{ij})$'s (averaged over j).
- Pick the model M_i with the lowest estimated generalization error, and retrain that model on the entire training set S

Feature Selection

- Given n features resulting in 2^n possible feature subsets, which one is the optimal?
- Forward search:
  - Initialize $\mathcal{F} = \emptyset$
  - Until $|\mathcal{F}| = \epsilon$ *or* $|\mathcal{F}| = n$, repeat
    - (a) For $i = 1, \cdots, n$, if $i \notin \mathcal{F}$, let $\mathcal{F}_i = \mathcal{F} \bigcup \{i\}$, and use cross validation to evaluate $\mathcal{F}_i$
    - (b) Set $\mathcal{F}$ to be the best feature subset found in (a)
- Backward search: Start with $\mathcal{F} = \{1, \cdots, n\}$, and repeatedly deletes features one at a time until $|\mathcal{F}| = \epsilon$
- The above two methods are so-called wrapper model, which is a procedure that "wraps" around your learning algorithm
- Wrapper feature selection algorithms usually have considerable computational cost
  - $O(n^2)$ calls to the learning algorithm

Filter Feature Selection (Contd.)

- Heuristic but computationally efficient
- Basic idea: Compute a score $S(i)$ to measure how informative each feature x_i is about the class labels y; then, select the k features with the largest scores $S(i)$
- Mutual information $MI(x_i, y)$ between x_i and y

$$MI(x_i, y) = \sum_{x_i \in \{0,1\}} \sum_{y \in \{0,1\}} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)}$$

  with $p(x_i, y)$, $p(x_i)$ and $p(y)$ estimated according their empirical distributions on the training set
- How to choose a right k?
  - Use cross validation

Thanks!

Q & A