Machine Learning

Lecture 9: Learning Theory

Feng Li

fli@sdu.edu.cn

https://funglee.github.io

School of Computer Science and Technology
Shandong University

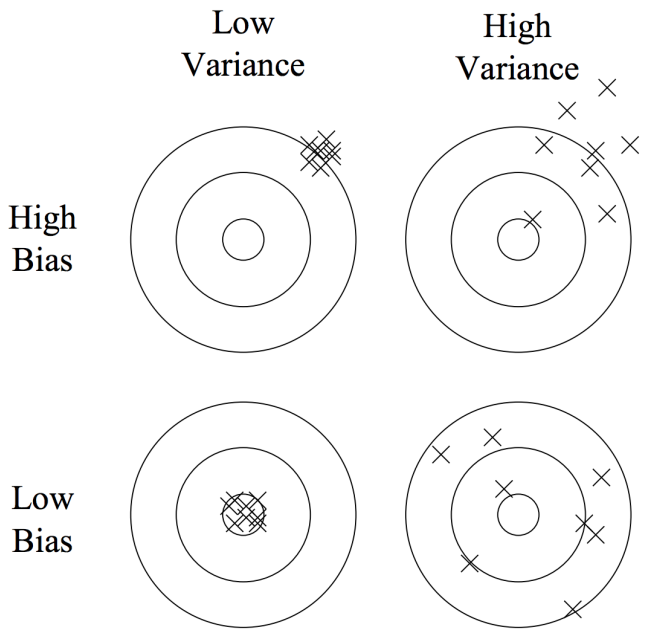
Fall 2018

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

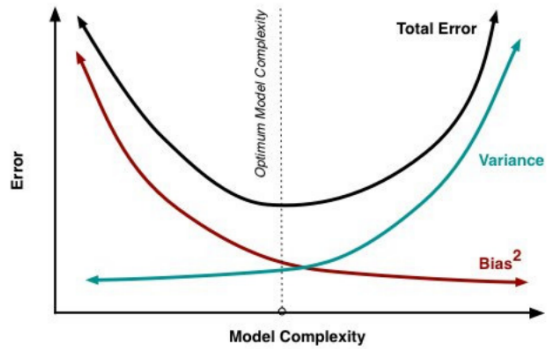
Bias vs Variance

- 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 Tradeoff

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

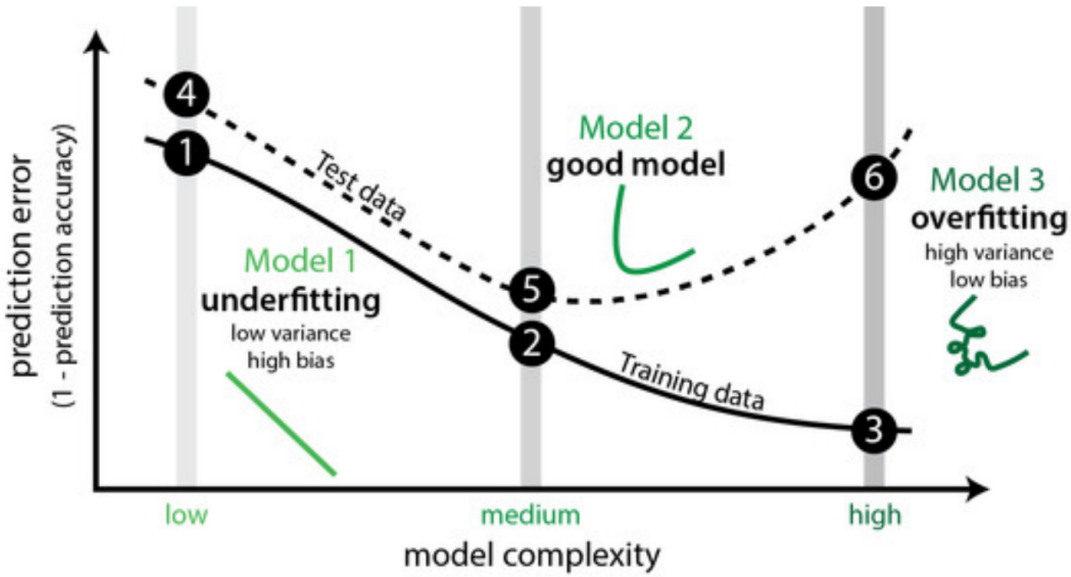


	Model				
	Bias	Variance	Complexity	Flexibility	Generalizability
Underfitting: you have an overly simple model	High	Low	Low	Low	High
Overfitting: your model is modelling the noise	Low	High	High	High	Low

- 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!)

High Bias or High Variance

- 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 Decomposition

- For some model $y = f(x) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$, given its estimate $\hat{f}$ learned by a "learner" using a finite training set, the following decomposition holds

$$E[(y - \hat{f}(x))^2] = Bias[\hat{f}(x)]^2 + Var[\hat{f}(x)] + \sigma^2$$

  - The above expectation is over all choices of training set
  - $Bias[\hat{f}(x)] = E[\hat{f}(x) - f(x)]$: Error due to wrong (perhaps too simple) model
  - $Var[\hat{f}(x)] = E[\hat{f}(x)^2] - E[\hat{f}(x)]^2$: Learner's sensitivity to choice of training set
  - The proof ($E[y] = f(x)$)

$$\begin{aligned}
E[(y - \hat{f})^2] &= E[y^2 + \hat{f}^2 - 2y\hat{f}] \\
&= E[y^2] + E[\hat{f}^2] - E[2y\hat{f}] \\
&= Var[y] + E[y]^2 + Var[\hat{f}] + E[\hat{f}]^2 - 2fE[\hat{f}] \\
&= Var[y] + Var[\hat{f}] + (f - E[\hat{f}])^2 \\
&= Var[y] + Var[\hat{f}] + E[f - \hat{f}]^2 \\
&= Var[y] + Var[\hat{f}] + Bias[\hat{f}]^2
\end{aligned}$$

Preliminaries

- **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, Training and Generalization Error

- 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 iid from some probability distribution $\mathcal{D}$
- The learning algorithm, given training data, learns a hypothesis $h \in \mathcal{H}$
- The training error (or empirical risk, empirical error) is

$$\hat{\varepsilon}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{h(x^{(i)}) \neq y^{(i)}\}$$

  i.e., the fraction of the misclassified training examples
- The generalization is

$$\varepsilon(h) = P_{(x,y)\sim\mathcal{D}}(h(x) \neq y)$$

  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) is a principle in statistical learning theory which defines a family of learning algorithms and is used to give theoretical bounds on the performance of learning algorithms

$$\hat{\theta} = \arg\min_{\theta} \hat{\varepsilon}(h_\theta)$$

  where $h_\theta(x) = \mathbf{1}\{\theta^T x \geq 0\}$
- In another word, we aim at seeking the optimal hypothesis $\hat{h} = h_{\hat{\theta}}$
- ERM can also be thought of a minimization over the class

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{\varepsilon}(h)$$

Finite $\mathcal{H}$

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

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 $\varepsilon(h_i)$ is the expected value of Z (and Z_j)
- The training error $\hat{\varepsilon}(h_i)$ can be written as

$$\hat{\varepsilon}(h_i) = \frac{1}{m} \sum_{j=1}^{m} Z_j$$

- $\hat{\varepsilon}(h_i)$ is exactly the mean of the m random variables Z_j's that are drawn iid from a Bernoulli distribution with mean $\varepsilon(h_i)$
- By applying Hoeffding inequality, we have

$$P(|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma) \leq 2 \exp(-2\gamma^2 m)$$

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

- For a particular h_i, training error will be close to generalization error with high probability, assuming m is large
- Is it true for $\forall h \in \mathcal{H}$?
  - Let A_i denote the event that $|\varepsilon(h_i) - \hat{\varepsilon}(h_i)| > \gamma$, then $P(A_i) \leq 2\exp(-2\gamma^2 m)$
  - By using the union bound, we have

$$\begin{aligned} P(\exists h \in \mathcal{H} : |\varepsilon(h) - \hat{\varepsilon}(h)| > \gamma) &= P(A_1 \bigcup \cdots \bigcup A_k) \\ &\leq \sum_{i=1}^{k} P(A_i) \\ &\leq \sum_{i=1}^{k} 2\exp(-2\gamma^2 m) \\ &= 2k\exp(-2\gamma^2 m) \end{aligned}$$

  - Then, we have the following uniform convergence result

$$\begin{aligned} & P(\neg\exists h \in \mathcal{H} : |\varepsilon(h) - \hat{\varepsilon}(h)| > \gamma) \\ = \; & P(\forall h \in \mathcal{H} : |\varepsilon(h) - \hat{\varepsilon}(h)| \leq \gamma) \\ \geq \; & 1 - 2k\exp(-2\gamma^2 m) \end{aligned}$$

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

- Question: Given γ and $\delta > 0$, how large must m be such that we can guarantee that with probability $\geq 1 - \delta$, training error will be within γ of generalization error?
- When

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

  with probability at least $1 - \delta$, we have $|\varepsilon(h) - \hat{\varepsilon}(h)| \leq \gamma$ for all $h \in \mathcal{H}$
- 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

$$1 - 2k\exp(-2\gamma^2 m) \geq 1 - \delta \Rightarrow |\varepsilon(h) - \hat{\varepsilon}(h)| \leq \sqrt{\frac{1}{2m}\log\frac{2k}{\delta}}$$

- Assume $h^* = \arg\min_{h\in\mathcal{H}} \varepsilon(h)$ denotes the best possible hypothesis in $\mathcal{H}$

$$\begin{aligned} \varepsilon(\hat{h}) &\leq \hat{\varepsilon}(\hat{h}) + \gamma \\ &\leq \hat{\varepsilon}(h^*) + \gamma \\ &\leq \varepsilon(h^*) + 2\gamma \end{aligned}$$

- If uniform convergence occurs, then the generalization error of $\hat{h}$ 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. Then with probability at least $1 - \delta$, we have that

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

- If we take a large $\mathcal{H}$
  - the first term is decreased (the bias is decreased)
  - the second term is increased (the variance is increased)
- Corollary: Let $\mathcal{H} = k$, and let any δ, γ be fixed. Then for $\varepsilon(\hat{h}) \leq \min_{h \in \mathcal{H}} \varepsilon(h) + 2\gamma$ to hold with probability at least $1 - \delta$, it suffices that

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

Infinite $\mathcal{H}$

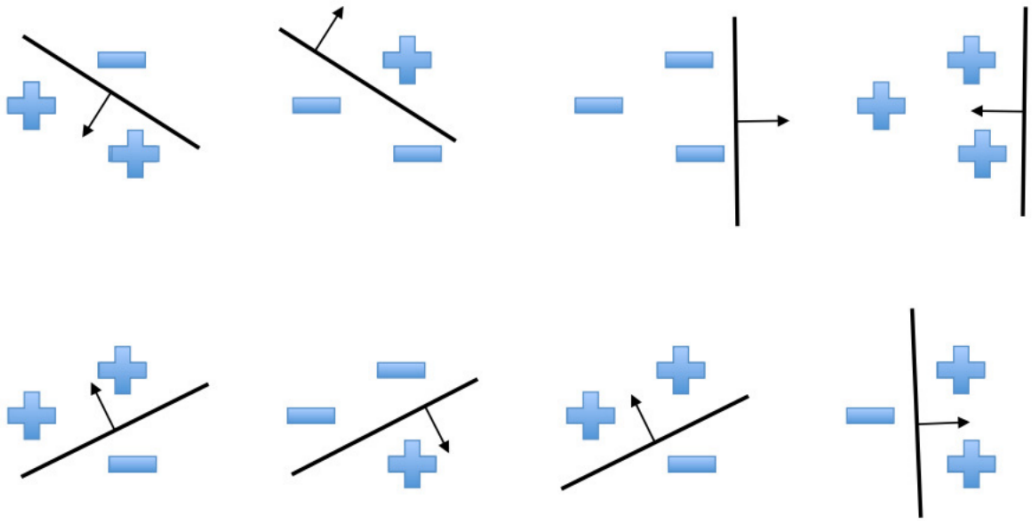
- For the finite sized hypothesis class $\mathcal{H}$

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

- 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



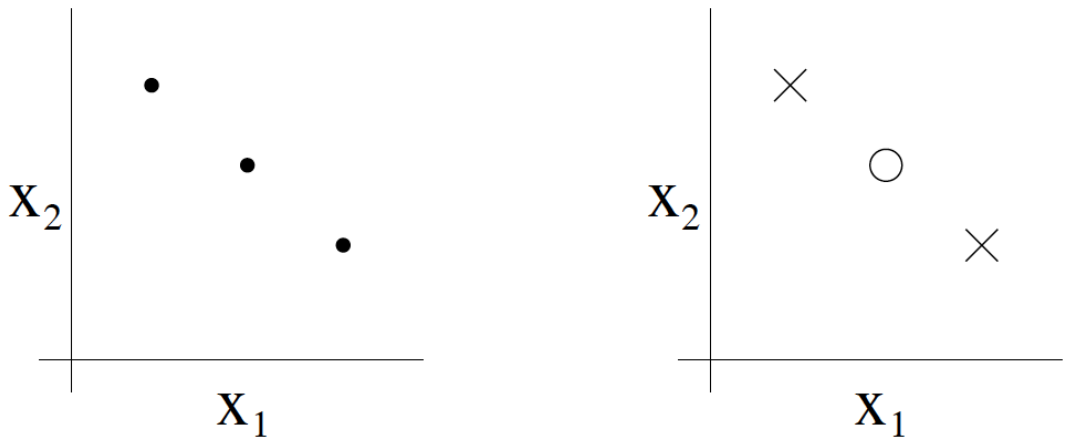
- Figure above: 3 points in 2D (locations fixed, only labeling varies), $\mathcal{H}$: set of linear classifier

Vapnik-Chervonenkis Dimension

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

Vapnik-Chervonenkis Dimension

- Even when $VC(\mathcal{H}) = 3$, 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 $VC(\mathcal{H})$ is at least d, we need to show only that there's at least one set of size d that $\mathcal{H}$ can shatter.

Vapnik-Chervonenkis Dimension

- 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

Vapnik-Chervonenkis Dimension

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

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

  and thus

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

- Recall for finite hypothesis space

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

  - $VC(H)$ is like a substitute for $k = |H|$

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 hyperparameter 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.)
- Note: Usually considered in supervised learning contexts but unsupervised learning too faces this issue (e.g., how many clusters when doing clustering)

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%). Here, S_{cv} is 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 $\hat{\varepsilon}_{S_{cv}}(h_i)$ on the hold-out cross validation set (Recall, $\hat{\varepsilon}_{S_{cv}}(h)$ denotes the empirical error of h on the set of examples in S_{cv})
- Option: After selecting $M^* \in \mathcal{M}$ such that $h^* = \arg\min_i \hat{\varepsilon}_{S_{cv}}(h_i)$, 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 of m/k training examples each. Lets call these subsets $S_1, \cdots, S_k$.
- 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 $\hat{\varepsilon}_{S_j}(h_{ij})$.
  - The estimated generalization error of model M_i is then calculated as the average of the $\hat{\varepsilon}_{S_j}(h_{ij})$'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. The resulting hypothesis is then output as our final answer.

Feature Selection

- n features result in 2^n possible feature subsets
- The question is which one is optimal (i.e., the most relevant features to the learning problem)
- 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

- 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