Model Selection and Assessment

CS4780/5780 – Machine Learning
Fall 2019

Nika Haghtalab & Thorsten Joachims
Cornell University

Reading: UML 11 (w/o 11.1)
https://machinelearningmastery.com/mcnemars-test-for-machine-learning/
https://en.wikipedia.org/wiki/McNemar%27s_test

Outline

- Model Selection
  - Controlling overfitting in decision trees
  - Train, validation, test
  - K-fold cross validation
- Evaluation
  - What is the true error of classification rule h?
  - Is rule h_1 more accurate than h_2?
  - Is learning algorithm A1 better than A2?

Sample & Generalization Error

Data generating distribution $P(X,Y)$ → Learning task.

Sample $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ drawn i.i.d. from $P(X,Y)$

$\Delta(a,b)$ is the 0/1-loss function. i.e.,

$$\Delta(a,b) = \begin{cases} 0 & if\ (a = b) \\ 1 & otherwise \end{cases}$$

Sample error of hypothesis h on sample S is

$$err_S(h) = \frac{1}{m}\sum_{i=1}^{m} \Delta(h(x_i), y_i)$$

Generalization error of hypothesis h on distribution $P(X,Y)$ is

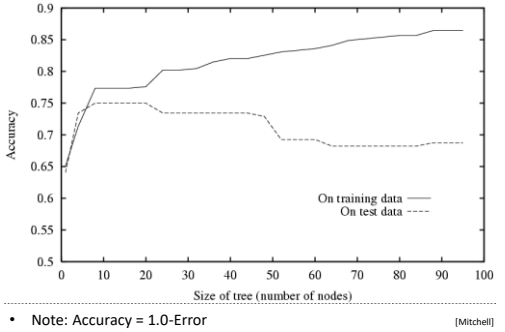
$$err_P(h) = \mathbb{E}_{(x,y)\sim P}[\Delta(h(x), y)]$$

Learning by Minimizing Training Error

- Goal:
  - Find $h \in H$ with small generalization error $err_P(h)$ over $P(X,Y)$.
- Strategy:
  - Pick $h \in H$ with small sample error $err_S(h)$ on training sample $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$.

→ Empirical Risk Minimization (ERM)

Overfitting



- Note: Accuracy = 1.0-Error

[Mitchell]

Occam's Razor for Decision Trees

"Pick the simplest tree that fits the data well."

- Restrict size of tree
  - Maximum number of nodes
  - Maximum depth
- Early Stopping: Introduce leaf when splitting no longer "reliable".
  - Minimum number of examples in node
  - Threshold on splitting criterion
- Post Pruning: Grow full tree, then simplify.
  - Reduced-error tree pruning
  - Rule post-pruning

Example: Text Classification

- Task: Learn rule that classifies Reuters Business News
  - Class +: "Corporate Acquisitions"
  - Class -: Other articles
  - 2000 training instances
- Representation:
  - Boolean attributes, indicating presence of a keyword in article
  - 9947 such keywords (more accurately, word "stems")

+ LAROCHE STARTS BID FOR NECO SHARES

Investor David F. La Roche of North Kingstown, R.I., said he is offering to purchase 170,000 common shares of NECO Enterprises Inc at 26 dlrs each. He said the successful completion of the offer, plus shares he already owns, would give him 50.5 pct of NECO's 962,016 common shares. La Roche said he may buy more, and possible all NECO shares. He said the offer and withdrawal rights will expire at 1630 EST/2130 gmt, March 30, 1987.

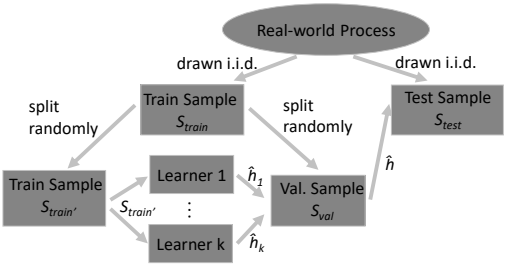
- SALANT CORP 1ST QTR FEB 28 NET

Oper shr profit seven cts vs loss 12 cts. Oper net profit 216,000 vs loss 401,000. Sales 21.4 mln vs 24.9 mln. NOTE: Current year net excludes 142,000 dlr tax credit. Company operating in Chapter 11 bankruptcy.

Text Classification Example: "Corporate Acquisitions" Results

- Unpruned Tree (ID3 Algorithm):
  - Size: 437 nodes Training Error: 0.0% Test Error: 11.0%
- Early Stopping Tree (ID3 Algorithm):
  - Size: 299 nodes Training Error: 2.6% Test Error: 9.8%
- Reduced-Error Tree Pruning (C4.5 Algorithm):
  - Size: 167 nodes Training Error: 4.0% Test Error: 10.8%
- Rule Post-Pruning (C4.5 Algorithm):
  - Size: 164 tests Training Error: 3.1% Test Error: 10.3%
  - Examples of rules
    - IF vs = 1 THEN - [99.4%]
    - IF vs = 0 & export = 0 & takeover = 1 THEN + [93.6%]

Model Selection: Validation Sample



- **Training:** Run learning algorithm k times (e.g. different parameters).
- **Validation Error:** Errors $err_{S_{val}}(\hat{h}_i)$ is an estimates of $err_P(\hat{h}_i)$ for each $\hat{h}_i$.
- **Selection**: Use $\hat{h}_i$ with min $err_{S_{val}}(\hat{h}_i)$ for prediction on test examples.

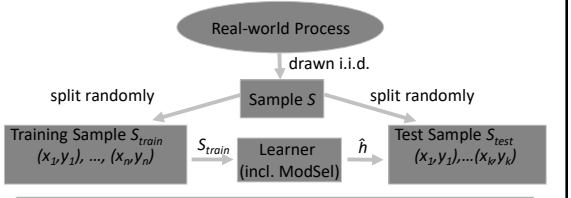
Text Classification Example: "Corporate Acquisitions" Results

- Unpruned Tree (ID3 Algorithm):
  - Size: 437 nodes Training Error: 0.0% Val Error: 11.0%
- Early Stopping Tree (ID3 Algorithm):
  - Size: 299 nodes Training Error: 2.6% Val Error: 9.8%
- Reduced-Error Tree Pruning (C4.5 Algorithm):
  - Size: 167 nodes Training Error: 4.0% Val Error: 10.8%
- Rule Post-Pruning (C4.5 Algorithm):
  - Size: 164 tests Training Error: 3.1% Val Error: 10.3%
  - Examples of rules
    - IF vs = 1 THEN - [99.4%]
    - IF vs = 0 & export = 0 & takeover = 1 THEN + [93.6%]

Model Selection: k-fold Cross Validation

- Given
  - Train examples S
  - Learning Algorithm A_p with parameter p
- Compute
  - Randomly partition S into k equally sized subsets $S_1, \ldots, S_k$
  - For each value of p:
    - For i from 1 to k
      - Train $A_p(\{S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_k\})$ and get $\hat{h}_i$.
      - Apply $\hat{h}_i$ to S_i and compute $err_{S_i}(\hat{h}_i)$.
    - $err_{CV}(A_p) = \sum_i err_{S_i}(\hat{h}_i)$
- Selection
  - Pick parameter p^* that minimizes $err_{CV}(A_p)$
  - Train $A_p(S)$ on full sample S to get final $\hat{h}$

Evaluating Learned Hypotheses

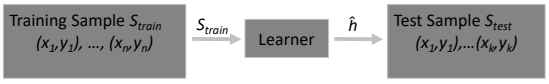


- Goal: Find h with small prediction error $err_P(h)$ over $P(X,Y)$.
- Question: How good is $err_P(\hat{h})$ of $\hat{h}$ found on training sample S_{train}.

- **Training Error:** Error $err_{S_{train}}(\hat{h})$ on training sample.
- **Test Error:** Error $err_{S_{test}}(\hat{h})$ is an estimate of $err_P(\hat{h})$.

What is the Generalization Error of a Hypothesis?

- Given
  - Sample of labeled instances S
  - Learning Algorithm A
- Setup
  - Partition S randomly into S_{train} (70%) and S_{test} (30%)
  - Train learning algorithm A on S_{train}, result is $\hat{h}$.
  - Apply $\hat{h}$ to S_{test} and compare predictions against true labels.
- Test
  - Error on test sample $Err_{S_{test}}(\hat{h})$ is estimate of true error $Err_P(\hat{h})$.
  - Compute confidence interval.



Binomial Distribution

- The probability of observing x heads in a sample of n independent coin tosses, where in each toss the probability of heads is p, is

$$P(X = x|p,n) = \frac{n!}{x!(n-x)!}p^x(1-p)^{n-x}$$

Text Classification Example: Results

- Data
  - Training Sample: 2000 examples
  - Test Sample: 600 examples
- Unpruned Tree:
  - Size: 437 nodes Training Error: 0.0% Test Error: 11.0%
- Early Stopping Tree:
  - Size: 299 nodes Training Error: 2.6% Test Error: 9.8%
- Post-Pruned Tree:
  - Size: 167 nodes Training Error: 4.0% Test Error: 10.8%
- Rule Post-Pruning:
  - Size: 164 tests Training Error: 3.1% Test Error: 10.3%

Confidence Intervals

- [Hoeffding] For the zero/one loss and any hypothesis h, with probability $1-\delta$ over the choice of test samples S of size m, it holds that

$$err_P(h) \in \left[err_S(h) - \sqrt{\frac{\log(2/\delta)}{2m}}, err_S(h) + \sqrt{\frac{\log(2/\delta)}{2m}} \right]$$

Text Classification Example: Results

- Data
  - Training Sample: 2000 examples
  - Test Sample: 600 examples
- Unpruned Tree:
  - Size: 437 nodes Training Error: 0.0% Test Error: 11.0%
- Early Stopping Tree:
  - Size: 299 nodes Training Error: 2.6% Test Error: 9.8%
- Post-Pruned Tree:
  - Size: 167 nodes Training Error: 4.0% Test Error: 10.8%
- Rule Post-Pruning:
  - Size: 164 tests Training Error: 3.1% Test Error: 10.3%

Is Rule h_1 More Accurate than h_2?

- Given
  - Sample of labeled instances *S*
  - Learning Algorithms A_1 and A_2
- Setup
  - Partition *S* randomly into S_{train} (70%) and S_{test} (30%)
  - Train learning algorithms A_1 *and* A_2 on S_{train}, result are $\hat{h}_1$ and $\hat{h}_2$.
  - Apply $\hat{h}_1$ and $\hat{h}_2$ to S_{test} and compute $Err_{S_{test}}(\hat{h}_1)$ and $Err_{S_{test}}(\hat{h}_2)$.
- Test
  - Decide, if $Err_P(\hat{h}_1) \neq Err_P(\hat{h}_2)$?
  - Null Hypothesis: $Err_{S_{test}}(\hat{h}_1)$ and $Err_{S_{test}}(\hat{h}_2)$ come from binomial distributions with same *p*.

    → Binomial Sign Test (McNemar's Test)

Is Learning Algorithm A_1 better than A_2?

- Given
  - k samples $S_1 \ldots S_k$ of labeled instances, all i.i.d. from P(X,Y).
  - Learning Algorithms A_1 and A_2
- Setup
  - For i from 1 to k
    - Partition S_i randomly into S_{train} (70%) and S_{test} (30%)
    - Train learning algorithms A_1 and A_2 on S_{train}, result are $\hat{h}_1$ and $\hat{h}_2$.
    - Apply $\hat{h}_1$ and $\hat{h}_2$ to S_{test} and compute $Err_{S_{test}}(\hat{h}_1)$ and $Err_{S_{test}}(\hat{h}_2)$.
- Test
  - Decide, if $E_S(Err_P(A_1(S_{train}))) \neq E_S(Err_P(A_2(S_{train})))$?
  - Null Hypothesis: $Err_{S_{test}}(A_1(S_{train}))$ and $Err_{S_{test}}(A_2(S_{train}))$ come from same distribution over samples S.
    → Binomial Sign Test or Wilcoxon Signed-Rank Test

Approximation via K-fold Cross Validation

- Given
  - Sample of labeled instances S
  - Learning Algorithms A_1 and A_2
- Compute
  - Randomly partition S into k equally sized subsets $S_1 \ldots S_k$
  - For i from 1 to k
    - Train A_1 and A_2 on $S_1 \ldots S_{i-1}\, S_{i+1} \ldots S_k$ and get $\hat{h}_1$ and $\hat{h}_2$.
    - Apply $\hat{h}_1$ and $\hat{h}_2$ to S_i and compute $Err_{S_i}(\hat{h}_1)$ and $Err_{S_i}(\hat{h}_2)$.
- Estimate
  - Average $Err_{S_i}(\hat{h}_1)$ is estimate of $E_S(Err_P(A_1(S_{train})))$
  - Average $Err_{S_i}(\hat{h}_2)$ is estimate of $E_S(Err_P(A_2(S_{train})))$
  - Count how often $Err_{S_i}(\hat{h}_1) > Err_{S_i}(\hat{h}_2)$ and $Err_{S_i}(\hat{h}_1) < Err_{S_i}(\hat{h}_2)$