Fundamentals of Learning

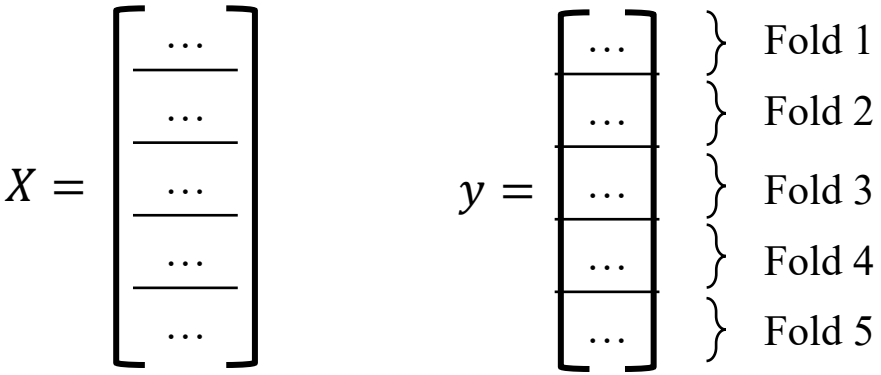
Kaiqi Zhao
The University of Auckland

Slides are partially based on the materials from the University of British Columbia

Cross-Validation

Cross-Validation (CV)

- How to mitigate the optimization bias problem?
- 5-fold cross-validation
  - Train on 80% of the data, validate on the other 20%
  - Repeat this 5 more times with different splits, and average the score

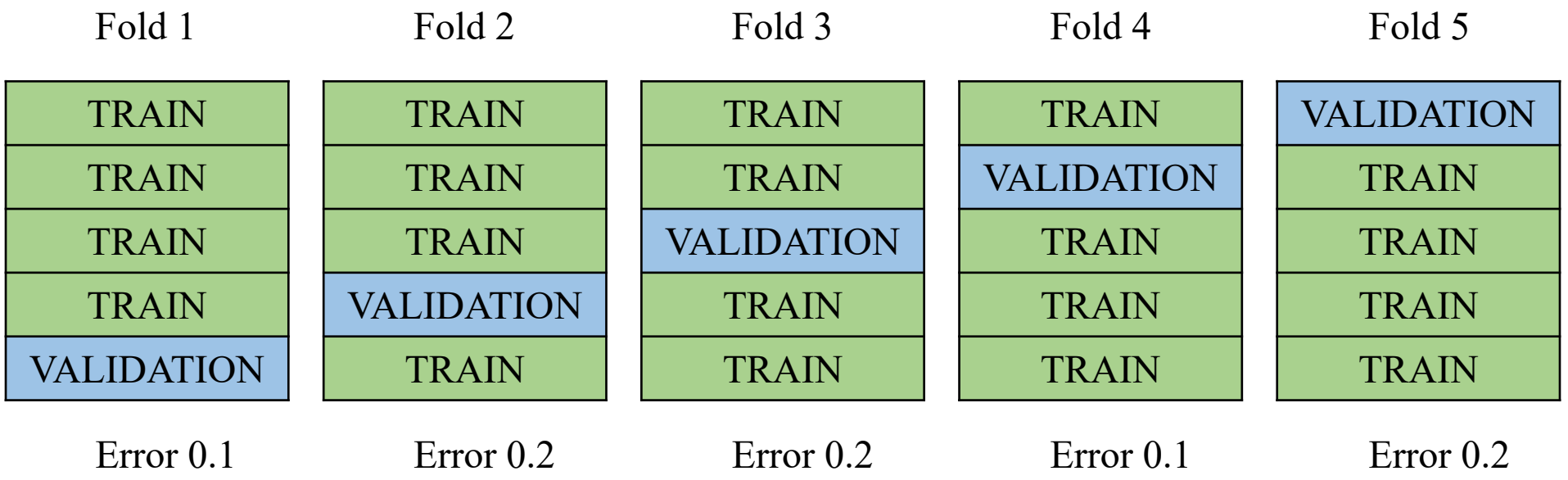


1. Train on folds 1, 2, 3, 4, compute error on fold 5
2. Train on folds 1, 2, 3, 5, compute error on fold 4
3. Train on folds 1, 2, 4, 5, compute error on fold 3

   . . .

6. Take average of the 5 errors as approximation of test error

Cross-Validation (CV)



- CV error estimate for this hyper-parameter $mean(errors) = 0.16$

Cross-Validation (CV)

- You can take this idea further (k-fold cross-validation)
  - 10-fold cross-validation: train on 90% of data and validate on 10%
    - Repeat 10 times and average (test on fold 1, then fold 2,..., then fold 10)
  - Leave-one-out cross-validation: train on all but one training example
    - Repeat n times and average
- More folds provide better estimation of errors, but more expensive
  - To choose depth we compute the cross-validation score for each depth
- As before, if data is ordered then folds should be random splits
  - Randomize first, then split into fixed folds
- Usually used in classification: stratified cross-validation
  - This enforces that the class distribution in all folds is approximately the same as in the full data set

Revisit Post-Pruning of Decision Trees

- The Reduced Error Pruning (REP) needs to use a validation set. Can we use REP with cross-validation?
- No!
  - The pruning will be different for different folds!
  - After cross-validation, you will re-train a new tree on all training data – we don't have a validation set to post-prune this tree.
- Can we do post-pruning using training data instead of validation data?

Post-pruning: Cost Complexity Pruning

- Let $R(T)$ be a measure of cost for a decision tree T (e.g., accuracy/error or weighted entropy from all leaf nodes)

- **For simplicity, let's say $R(T)$ be the error**. If we use $R(T)$ of training data in REP:
  - REP tends to result in a deep (complex) tree, because deeper trees is more likely to be accurate on the training data!
  - But we don't want the model to be too complex!

- To use training data for pruning, we need a measure that considers the trade-off between cost (e.g., error) and complexity!

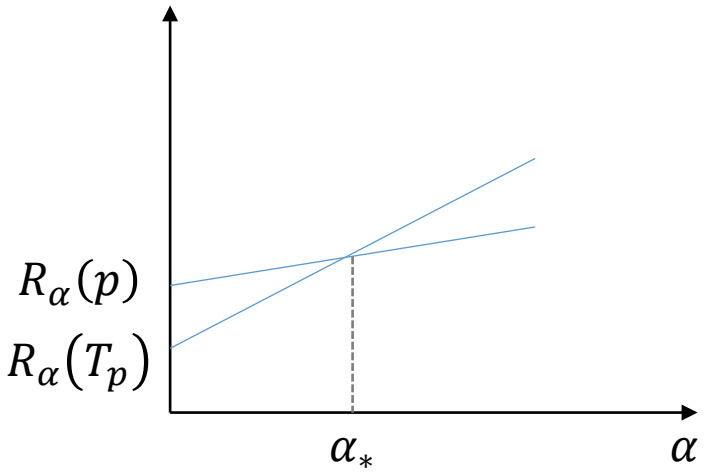
Post-pruning: Cost Complexity Pruning

- Cost-Complexity measure of (sub-)tree T:

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|,$$

where $\tilde{T}$ is the set of leave nodes of T.

  - For a leaf node p: $R_\alpha(p) = R(p) + \alpha$
  - For a sub-tree T_p rooted at p: $R_\alpha(T_p) = R(T_p) + \alpha|\tilde{T}_p|$



- When α is close to 0, $R_\alpha(p) > R_\alpha(T_p)$
- When α increase to some point α_*, $R_{\alpha_*}(p) = R_{\alpha_*}(T_p)$
- α_* is the smallest α that we can prune the subtree T_p
- Different non-leaf nodes have different α_*

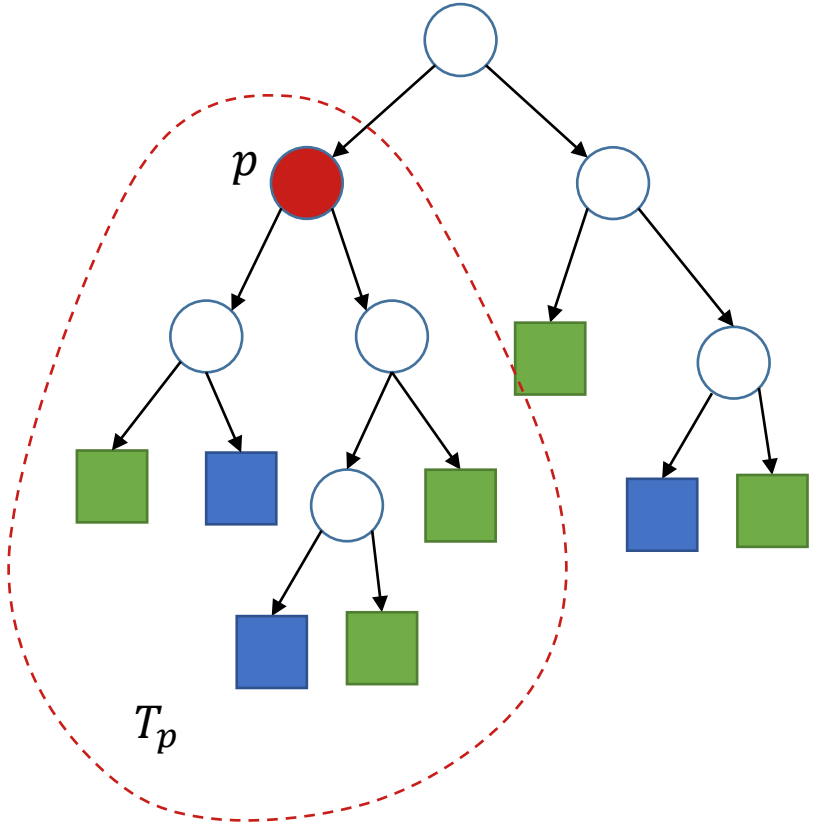
$$R(p) + \alpha_* = R(T_p) + \alpha_*|\tilde{T}_p|$$

$$\alpha_* = \frac{R(p) - R(T_p)}{|\tilde{T}_p| - 1}$$

Post-pruning: Cost Complexity Pruning

- Another perspective of $\alpha_* = \frac{R(p)-R(T_p)}{|\tilde{T}_p|-1}$:
  - If we prune the subtree T_p rooted at p
  - The error increases by $R(p) - R(T_p)$
  - # leaf nodes decreases by $|\tilde{T}_p| - 1$
  - α_* is the cost-complexity ratio
- Cost-Complexity Pruning (CCP):
  - Compute α_* for *all non-leaf nodes*, and prune the subtree rooted at the non-leaf node with the smallest α_*
  - Repeat on the pruned tree and stop until the smallest α_* is above a threshold

  Select the threshold using cross-validation!



Example codes: https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py

Classifier Evaluation

Confusion Matrix

- Given m classes, an entry, $CM_{i,j}$ in a confusion matrix indicates number of tuples in class i that were labeled by the classifier as class j

Predicted class

Actual class	Predicted as positive	Predicted as negative
Actual positive	**True positive (TP)**	**False negative (FN)**
Actual negative	**False positive (FP)**	**True negative (TN)**

- FP is often called type I error
- FN is called type II error

Classifier Evaluation Metrics: Accuracy, Precision and Recall

- **Accuracy**, or recognition rate
  - Percentage of test set tuples that are correctly classified

$$Accuracy = (TP + TN)/All$$

- **Precision**: Exactness: what % of tuples that the classifier labeled as positive are actually positive?

$$P = Precision = \frac{TP}{TP + FP}$$

- **Recall**: Completeness: what % of positive tuples did the classifier label as positive?

$$R = Recall = \frac{TP}{TP + FN}$$

Classifier Evaluation Metrics: F−measure

- ***F*-measure** (or ***F*-score**): harmonic mean of precision and recall

$$F_\beta = (1 + \beta^2)\frac{precison \cdot recall}{(\beta^2 \cdot precision) + recall}$$

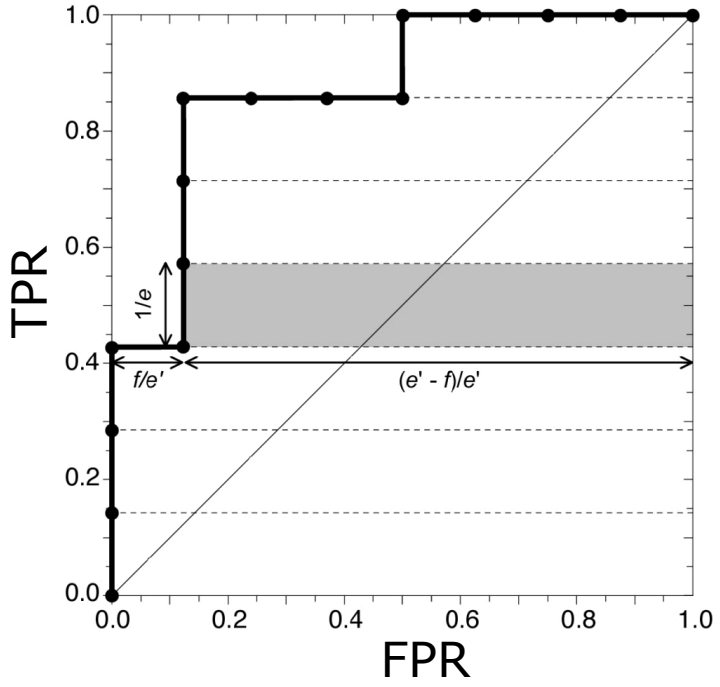
  - In general, it is the weighted measure of precision & recall
  - β is a weight − assign β times as much weight to recall as to precision
  - Most commonly used: F_1-measure with $\beta = 1$

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

ROC Curves

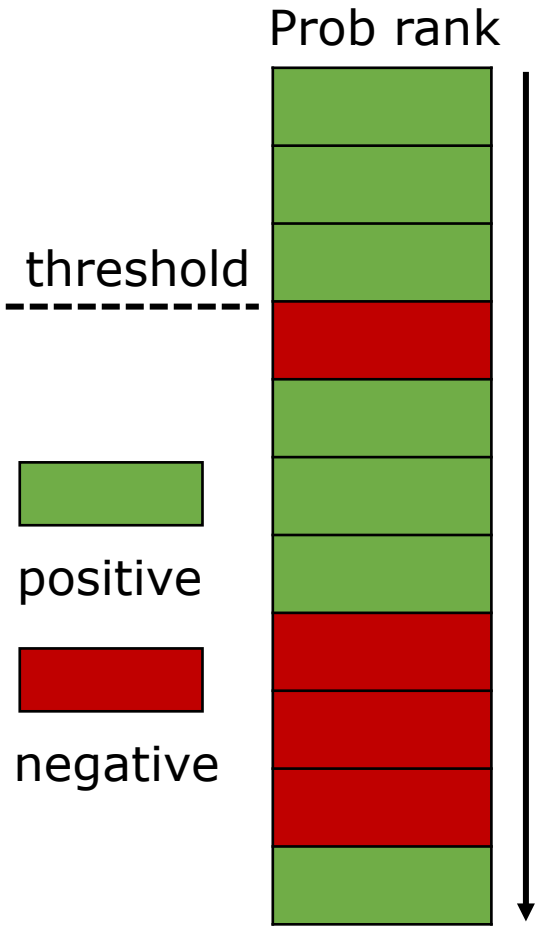
- Some classifiers return the probability or scores instead of class label
  - E.g., 90% to be positive
  - You can set a threshold to determine the class label

- **Receiver Operating Characteristic (ROC)** curve shows the trade-off between true positive rate (TPR) and false positive rate (FPR) by varying the threshold.
  - True positive ratio (TPR): $\frac{TP}{TP+FN}$
  - False positive ratio (FPR): $\frac{FP}{FP+TN}$

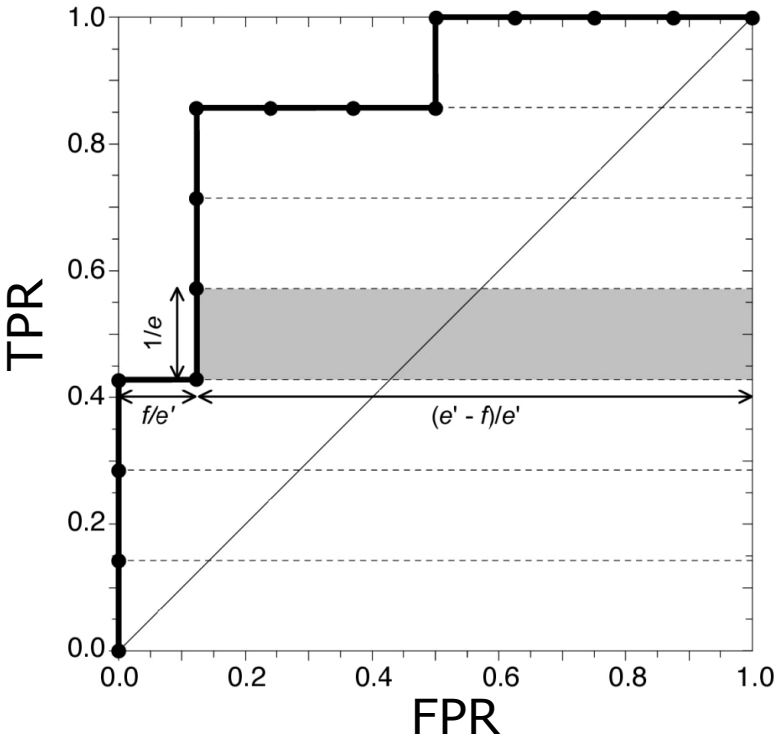


Area Under Curve (AUC)

- The area under the ROC curve (AUC: Area Under Curve, also AUROC) is a measure of the accuracy of the model



#	TPR	FPR	Area increment
1	1/7	0/8	1/7 * 1 = 1/7
2	2/7	0/8	1/7 * 1 = 1/7
3	3/7	0/8	1/7 * 1 = 1/7
4	3/7	1/8	0
5	4/7	1/8	1/7 * (1-1/8) = 1/8
6	5/7	1/8	1/7 * (1-1/8) = 1/8
7	6/7	1/8	1/7 * (1-1/8) = 1/8
8	6/7	2/8	0
9	6/7	3/8	0
10	6/7	4/8	0
11	7/7	4/8	1/7 * (1-4/8) = 1/14



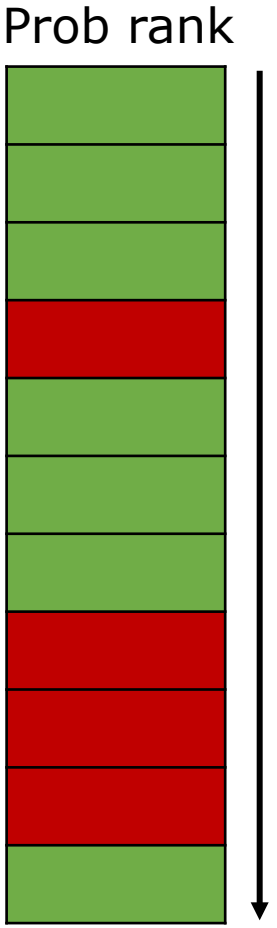
$$AUC = \frac{7}{8}$$

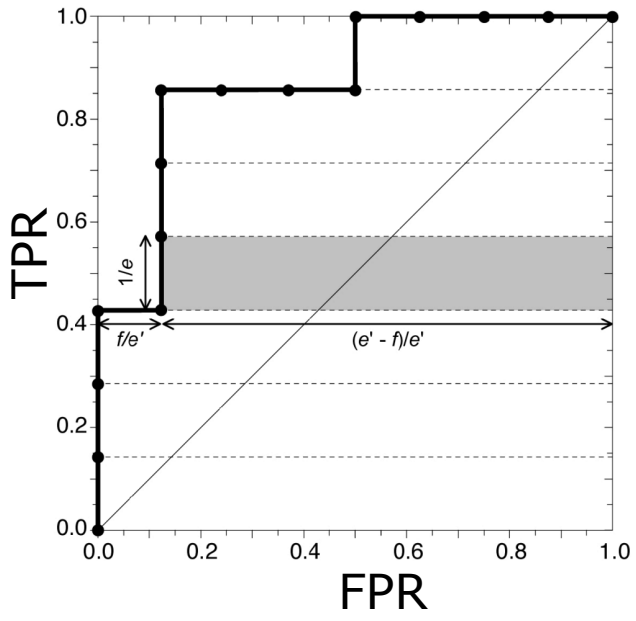
Area Under Curve (AUC)

- Another view of AUC:
  - How many positive examples are ranked higher than the negative examples

$$AUC(u) = \frac{\sum_{j \in P_u^-} \sum_{j' \in P_u^+} \mathbf{1}[f(j) < f(j')]}{|P_u^-| \cdot |P_u^+|}$$

  - $f(\cdot)$ is the model that outputs the probability of an example being positive
  - P_u^+ - the set of positive examples
  - P_u^- - the set of negative examples



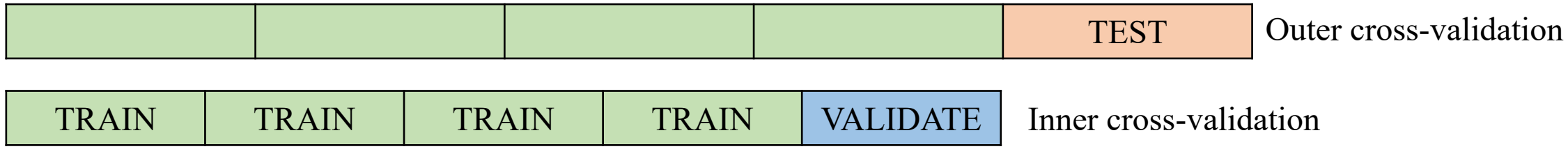


$P_u^+ = 7, P_u^- = 8$

Comparing Classifiers

Comparing Classifiers –Which one is better?

- Suppose we have 2 classifiers, M_1 and M_2, which one is better?
- Use K-fold cross-validation to obtain the mean error (or accuracy, AUROC, ...)
  - If you need to tune hyperparameters for M_1 and M_2, you might need a nested cross-validation.



- These mean error rates are just estimates of error on the true population of future data cases
- What if the difference between the 2 error rates is just by chance?
  - Use a test of statistical significance

Estimating Confidence Intervals: Null Hypothesis

- Perform 10-fold cross-validation
- Use hypothesis testing to determine whether the two error rates of the two models have the same mean (not by chance)
- Null Hypothesis: M_1 & M_2 are the same
- If we can reject null hypothesis, then
  - We conclude that the difference between M_1 & M_2 is ***statistically significant***
  - Choose model with lower error rate
- Use t-test (or Student's t-test)

Estimating Confidence Intervals: t-test

- If only 1 test set available: paired t-test
  - For i-th round of 10-fold cross-validation, the same test partition is used to obtain errors $e(M_1)_i$ and $e(M_2)_i$
  - Let $d_i = e(M_1)_i - e(M_2)_i$ be the difference of the errors
  - Average over 10 rounds to get $\bar{d}$.
  - t-test computes t-value with $k - 1$ degrees of freedom:

$$t = \frac{\bar{d}}{\sqrt{S_d/k}}$$

where S_d is the sample variance of the difference of M_1 and M_2.

$$S_d = \frac{1}{k-1}\sum_{i=1}^{k}(d_i - \bar{d})^2$$

$e(M_1)$	$e(M_2)$	d	$(d-\bar{d})^2$
0.09	0.11	-0.02	0.000001
0.12	0.1	0.02	0.001681
0.07	0.12	-0.05	0.000841
0.12	0.09	0.03	0.002601
0.08	0.14	-0.06	0.001521
0.1	0.13	-0.03	0.000081
0.08	0.12	-0.04	0.000361
0.08	0.11	-0.03	0.000081
0.1	0.13	-0.03	0.000081
0.09	0.09	0	0.000441

$\bar{d} = -0.021$,

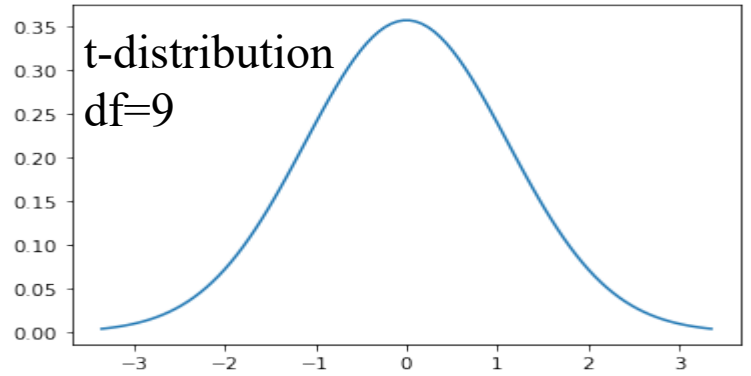
Estimating Confidence Intervals: Statistical Significance

- Are M_1 & M_2 significantly different?

  Method 1:

  1. Compute t value ($t = -2.271$ in the previous example).
  2. Select significance level (e.g. sig = 5%)
  3. Consult table for t-distribution: Find t value corresponding to $k - 1$ degrees of freedom (here, 9), note that t distribution is symmetric, we find 0.05 from the row "two-tails". In this case it is 2.262.
  4. If $t > 2.262$ or $t < -2.262$, then t value lies in rejection region:
     - Reject null hypothesis (i.e., error rates of M_1 & M_2 are the same)
     - Conclude: statistically significant difference between M_1 & M_2

     Otherwise, conclude that any difference is chance



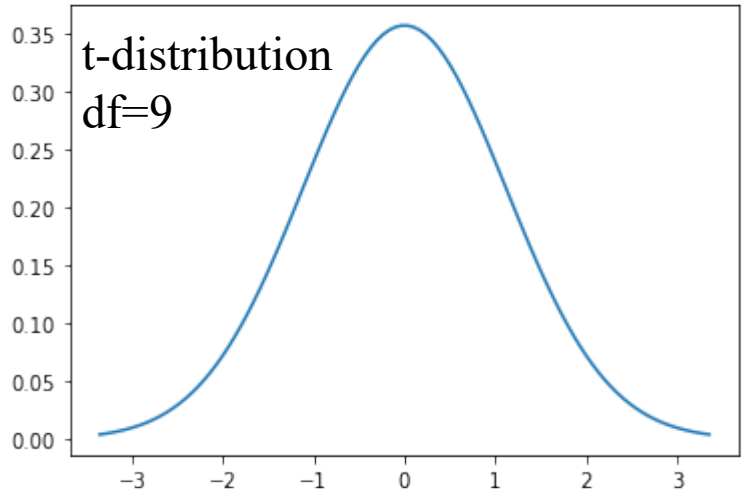
	P			
one-tail	0.1	0.05	0.025	0.01
two-tails	0.2	0.1	0.05	0.02
DF				
1	3.078	6.314	12.706	31.821
2	1.886	2.92	4.303	6.965
3	1.638	2.353	3.182	4.541
4	1.533	2.132	2.776	3.747
5	1.476	2.015	2.571	3.365
6	1.44	1.943	2.447	3.143
7	1.415	1.895	2.365	2.998
8	1.397	1.86	2.306	2.896
9	1.383	1.833	2.262	2.821
10	1.372	1.812	2.228	2.764

Estimating Confidence Intervals: Statistical Significance

- Are M_1 & M_2 significantly different?

  Method 2:

  1. Compute t value ($t = -2.271$ in the previous example).
  2. Compute the p −value based on the t-value.
     - The p-value is the probability of events that are equivalent or rarer.
     - In the example, the p-value is 0.049
  3. The p-value < 0.05, which means the chance of M_1 and M_2 having the same error rates is less than 5%. So, we can reject the null hypothesis.



Python package for paired t-test:
https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_rel.html

Multiple Comparisons

- Beyond paired comparison: Nemenyi test, Friedman test, Critical Distance plots
- Janez Demšar: "Statistical Comparisons of Classifiers over Multiple Data Sets"

The Best Machine Learning Model

The Best Machine Learning Model

- Decision trees are not always most accurate on test error
- What is the best machine learning model?
- An alternative measure of performance is the generalization error
  - Average error over all x_i vectors that are not seen in the training set
  - How well we expect to do for a completely unseen feature vector
- No free lunch theorem
  - There is no best model achieving the best generalization error for every problem
  - If model A generalizes better to new data than model B on one dataset, there is another dataset where model B works better
- This question is like asking which is best among rock, paper, and scissors

The Best Machine Learning Model

- Implications of the lack of a best model
  - We need to learn about and try out multiple models
- So which ones to study?
  - We'll usually motivate each method by a specific application
  - But we're focusing on models that have been effective in many applications
- Caveat of no free lunch (NFL) theorem
  - The world is very structured
  - Some datasets are more likely than others
  - Model A really could be better than model B on every real dataset in practice
- Machine learning research
  - Large focus on models that are useful across many applications.

Summary

- Training error vs. testing error
  - What we care about in machine learning is the testing error
- Golden rule of machine learning
  - The test data cannot influence training the model in any way
- Independent and identically distributed (IID)
  - One assumption that makes learning possible
- Fundamental trade-off:
  - Trade-off between getting low training error and having training error approximate test error

Summary

- Validation set:
  - We can save part of our training data to approximate test error
- Hyperparameters
  - Parameters that control model complexity, typically set with a validation set
- Cross-validation: allows better use of data to estimate test error
- No free lunch theorem: there is no best ML model