Classification

Practical Issues

Huiping Cao

Outline

- Criteria to evaluate a classifier
- Underfitting and overfitting
- Model evaluation

Model Evaluation

- How to evaluate the performance of a model? Metrics for Performance Evaluation
- How to obtain reliable estimates? Methods for Performance Evaluation
- How to compare the relative performance among competing models? Methods for Model Comparison

Evaluating Classification Methods

- Accuracy
  - Classifier accuracy: predicting class label
- Speed
  - time to construct the model (training time)
  - time to use the model (classification/prediction time)
- Robustness: handling noise and missing values
- Scalability: efficiency in disk-resident databases
- Interpretability: understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

Evaluation

- Confusion matrix

		Predicted Class	
		Class=1	Class=0
Actual Class	Class=1	f_{11}	f_{10}
	Class=0	f_{01}	f_{00}

- Performance metric

$$Accuracy = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

$$Error\ rate = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Desirable classifier: high accuracy, low error rate

Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples $= 9990$
  - Number of Class 1 examples $= 10$
- If a model predicts everything to be class 0, accuracy is $9990/10000 = 99.9\%$
- Accuracy is misleading because model does not detect any class 1 example

Why more measures?

- Existence of data sets with imbalanced class distributions. E.g., deffective products and non-defective products.
- The accuracy measure, may not be well suited for evaluating models derived from imbalanced data sets.
- Analyzing imbalanced data sets, where the rare class is considered more interesting than the majority class.
- Binary classification, the rare class is denoted as the positive class.

Precision, Recall, F_1

- Confusion matrix

		Predicted Class	
		$+$	$-$
Actual Class	$+$	f_{11} (TP)	f_{10} (FN)
	$-$	f_{01} (FP)	f_{00} (TN)

- Performance metric

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

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

Desirable classifier: high precision, high recall

$F_1 = \frac{2rp}{r+p} = \frac{2\times TP}{2\times TP+FP+FN} = \frac{2}{\frac{1}{r}+\frac{1}{p}}$

Combined metrics

- Given n numbers $x_1, \cdots, x_n$
  - Arithmetic mean: $\frac{\sum_{i=1}^{n} x_i}{n}$
  - Harmonic mean: $\frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$
  - Geometric mean: $\sqrt[n]{x_1 \cdot \cdots \cdot x_n}$

Mean

- Given $a = 1,\ b = 5$
- Mean values
  - Arithmetic mean: 3
  - Geometric mean: $\sqrt{1 \times 5} = 2.236$
  - Harmonic mean: $\frac{2}{1+\frac{1}{5}} = \frac{10}{6} = 1.667$, closer to the smaller value between a and b

Confusion matrix

- Confusion matrix (or contingency table)

	Predicated class = Yes	Predicated class= No
Actual class = Yes	TP	FN
Actual class = No	FP	TN

- Alternative metrics
  - True positive rate: $TPR = \frac{TP}{TP+FN}$, also called recall or sensitivity
  - True negative rate: $TNR = \frac{TN}{TN+FP}$
  - False positive rate: $FPR = \frac{FP}{FP+TN} = 1 - SPC$
  - False negative rate: $FNR = \frac{FN}{TP+FN}$
  - Specificity $SPC = \frac{TN}{N} = \frac{TN}{FP+TN}$

ROC (Receiver Operating Characteristic)

- Developed for signal detection theory
- Characterize the trade-off between true positive rate (TPR) and false positive rate (FPR)
- ROC curve plots TPR (on the y-axis) against FPR (on the x-axis)
- Performance of each instance represented as a point on the ROC curve

ROC – Properties

- (TPR=0, FPR=0): every instance is predicted to be a negative class
- (TPR=1, FPR=1): every instance is predicted to be a positive class
- (TPR=1, FPR=0): the ideal model
- Diagonal line: random guessing
- A good classification model should be located as close as possible to the upper-left corner of the diagram.
- No model consistently outperforms the other
- M1 is better than M2 when FPR is smaller
- M2 is better than M1 when FPR is bigger

AUC (Area under the ROC Curve)

- AUC can evaluate which model is better on average.
- AUC $= 1$: the model is perfect.
- AUC $= 0.5$: random guess

Construct a ROC curve

Instance	$P(+\|A)$	True class
1	0.95	$+$
2	0.93	$+$
3	0.87	$-$
4	0.86	$+$
5	0.85	$-$
6	0.84	$-$
7	0.76	$-$
8	0.53	$+$
9	0.43	$-$
10	0.25	$+$

- Use classifier that produces posterior probability for each test instance $P(+|A)$

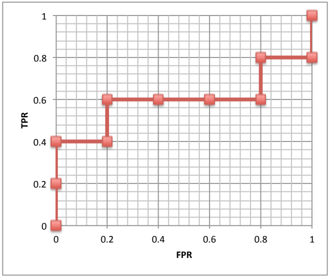
Construct a ROC curve (2)

class	+	−	+	−	−	−	+	−	+	+	
	0.25	0.43	0.53	0.76	0.84	0.85	0.86	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

- Sort the instances according to $P(+|A)$ in decreasing order
- Apply threshold at each unique value of $P(+|A)$
- Count the number of TP, FP, TN, FN at each threshold δ
  - Assign the selected record with $p \geq \delta$ to the positive class.
  - Assign those records with with $p < \delta$ as negative class
- TPR = TP/(TP+FN)
- FPR = FP/(FP+TN)

Construct a ROC curve (3)

class	+	−	+	−	−	−	+	−	+	+	
	0.25	0.43	0.53	0.76	0.84	0.85	0.86	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

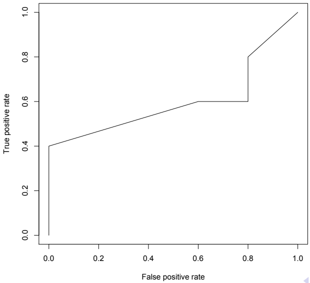


Construct a ROC curve (4) - Tie

- ROC curve when some probability values are duplicated? Use only distinct probabilities

class	+	−	+	−	+	−	−	−	+	+	
	0	0	0.7	0.76	0.85	0.85	0.85	0.85	0.95	0.95	1.00

boundary	0	0.7	0.76	0.85	0.95	1.00
TP	5	4	3	3	2	0
FP	5	4	4	3	0	0
TN	0	1	1	2	5	5
FN	0	1	2	2	3	5
TPR	1	0.8	3/5	3/5	0.4	0
FPR	1	0.8	3/4	3/5	0	0

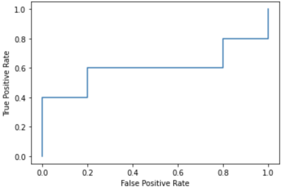


Construct a ROC curve - normal case

```
import numpy as np
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

pred = np.array([0.95,0.93,0.87,0.86,0.85,0.84,0.76,0.53,0.43,0.25])
labels = np.array([1,1,0,1,0,0,0,1,0,1])

fpr, tpr, _ = roc_curve(labels,pred)
print("fpr = ", fpr) # fpr =  [0.  0.  0.  0.2 0.2 0.8 0.8 1.  1. ]
print("tpr = ", tpr) # tpr =  [0.  0.2 0.4 0.4 0.6 0.6 0.8 0.8 1. ]
roc_auc = auc(fpr, tpr)
print("roc_auc=",  roc_auc) # roc_auc= 0.6000000000000001
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.plot(fpr, tpr)
```

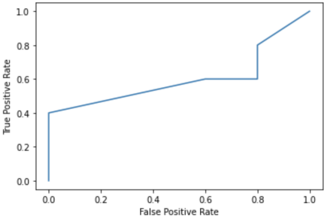


Construct a ROC curve - special case (ties)

```
1import numpy as np
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

pred = np.array([0.95,0.95,0.85,0.85,0.85,0.85,0.76,0.7,0,0])
labels = np.array([1,1,0,0,0,1,0,1,0,1])

fpr, tpr, _ = roc_curve(labels,pred)
roc_auc = auc(fpr, tpr)
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.plot(fpr, tpr)
```



Classification errors

- Training errors: the number of misclassification errors on training records
- Generalization errors: the expected error of the model on previously unseen records

Overfitting and Underfitting

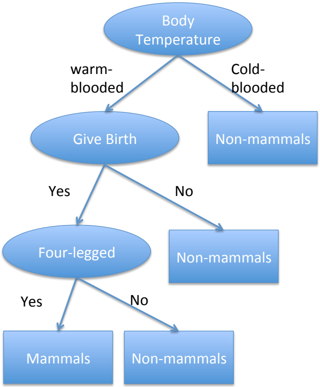
- Model overfitting: A model fits the training data too well, but has poorer generalization error than a model with a higher training error.
- Model underfitting: a model has not learned the true structure of the data. Has high training error and generalization error. **Underfitting**: when a model is too simple, both training and test errors are large

Overfitting Reason 1 – Noise –Example

Training set:

Name	Body Temperature	Give Birth	Four-legged	Hibernates	Class
porcupine	warm-blooded	yes	yes	yes	mammals
cat	warm-blooded	yes	yes	no	mammals
bat	warm-blooded	yes	no	yes	non-mammals*
whale	warm-blooded	yes	no	no	non-mammals*
salamander	cold-blooded	no	yes	yes	non-mammals
komodo	cold-blooded	no	yes	no	non-mammals
python	cold-blooded	no	no	yes	non-mammals
salmon	cold-blooded	no	no	no	non-mammals
eagle	warm-blooded	no	no	no	non-mammals
guppy	cold-blooded	yes	no	no	non-mammals

Overfitting Reason 1 – Noise –Example



Training error: 0.0

Overfitting Reason 1 – Noise –Example

Testing set:

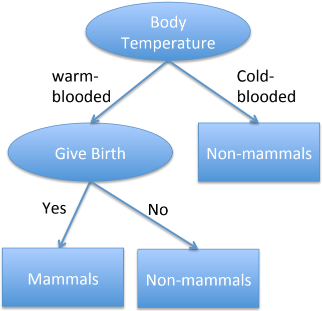
Name	Body Temperature	Give Birth	Four-legged	Hibernates	Class
human	warm-blooded	yes	no	no	?
pigeon	warm-blooded	no	no	no	?
elephant	warm-blooded	yes	yes	no	?
leopard shark	cold-blooded	yes	no	no	?
turtle	cold-blooded	no	yes	no	?
penguin	cold-blooded	no	no	no	?
eel	cold-blooded	no	no	no	?
dolphin	warm-blooded	yes	no	no	?
spiny anteater	warm-blooded	no	yes	yes	?
gila monster	cold-blooded	no	yes	yes	?

Overfitting Reason 1 – Noise –Example

Test error: 30%

Name	Body Temperature	Give Birth	Four-legged	Hibernates	Class
human	warm-blooded	yes	no	no	non-mammals
pigeon	warm-blooded	no	no	no	non-mammals
elephant	warm-blooded	yes	yes	no	mammals
leopard shark	cold-blooded	yes	no	no	non-mammals
turtle	cold-blooded	no	yes	no	non-mammals
penguin	cold-blooded	no	no	no	non-mammals
eel	cold-blooded	no	no	no	non-mammals
dolphin	warm-blooded	yes	no	no	non-mammals
spiny anteater	warm-blooded	no	yes	yes	non-mammals
gila monster	cold-blooded	no	yes	yes	non-mammals

Overfitting Reason 1 – Noise –Example



Training error: 20%

Overfitting Reason 1 – Noise –Example

Test error: 10%

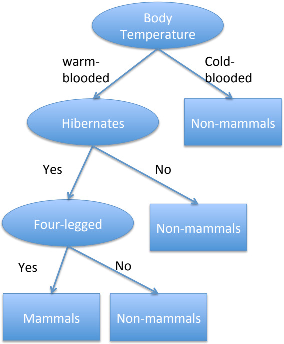
Name	Body Temperature	Give Birth	Four-legged	Hibernates	Class
human	warm-blooded	yes	no	no	mammals
pigeon	warm-blooded	no	no	no	non-mammals
elephant	warm-blooded	yes	yes	no	mammals
leopard shark	cold-blooded	yes	no	no	non-mammals
turtle	cold-blooded	no	yes	no	non-mammals
penguin	cold-blooded	no	no	no	non-mammals
eel	cold-blooded	no	no	no	non-mammals
dolphin	warm-blooded	yes	no	no	mammals
spiny anteater	warm-blooded	no	yes	yes	non-mammals
gila monster	cold-blooded	no	yes	yes	non-mammals

Overfitting Reason 2 – Insufficient Examples

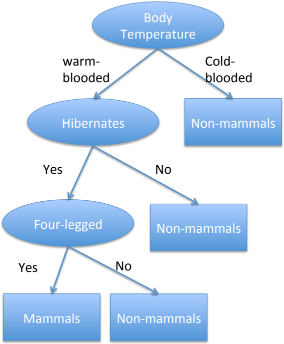
Training set:

Name	Body Temperature	Give Birth	Four-legged	Hibernates	Class
salamander	cold-blooded	no	yes	yes	non-mammals
guppy	cold-blooded	yes	no	no	non-mammals
eagle	warm-blooded	no	no	no	non-mammals
poorwill	warm-blooded	no	no	yes	non-mammals
platypus	warm-blooded	no	yes	yes	mammals

Overfitting Reason 2 – Insufficient Examples



Overfitting Reason 2 – Insufficient Examples



Name	Body Temperature	Give Birth	Four-legged	Hibernates	Class
human	warm-blooded	yes	no	no	non-mammals
elephant	warm-blooded	yes	yes	no	non-mammals
dolphin	warm-blooded	yes	no	no	non-mammals

Notes on Overfitting

- What is the primary reason for overfitting? a subject of debate
- Generally agreed: the complexity of a model has an impact on model overfitting
- E.g., Overfitting results in decision trees that are more complex than necessary
- Need new ways for estimating generalization errors

Estimating Generalization Errors

- Training errors: error on training $(\sum_{i=1}^{n} e(t_i))$
- Generalization errors: error on testing $(\sum_{i=1}^{m} e'(t_i))$
- Methods for estimating generalization errors:
  - Optimistic approach: $e'(t) = e(t)$
  - Reduced error pruning (REP):
    - Uses validation data set to estimate generalization error
  - Incorporating model complexity.
    - The ideal complexity is that of a model that produces the lowest generalization error.
    - The problem: the learning algorithm has no knowledge of the test data in building the model.

Incorporating model complexity – Occam's Razor

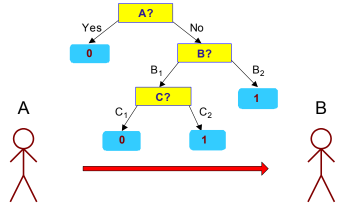
- Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
- For complex models, there is a greater chance that it was fitted accidentally by errors in data
- Therefore, one should include **model complexity** when evaluating a model

Incorporating model complexity - Pessimistic approach

- For each leaf node: $e'(t) = (e(t) + 0.5)$
- Total errors: $e'(T) = e(T) + N \times 0.5$ (N: number of leaf nodes)
- For a tree with 30 leaf nodes and 10 errors on training (out of 1000 instances):
  - Training error $= 10/1000 = 1\%$
  - Generalization error $= (10 + 30\times\ 0.5)/1000 = 2.5\%$

Incorporating model complexity - Minimum Description Length (MDL)

X	y
X_1	1
X_2	0
X_3	0
X_4	1
...	...
X_n	1



X	y
X_1	?
X_2	?
X_3	?
X_4	?
...	...
X_n	?

- Cost is the number of bits needed for encoding.
- A DT algorithm aims at obtaining the smallest decision tree that can capture the relations, that is, the DT that requires the MDL.
- Search for the least costly model.

  $$\text{Cost}(\text{Model},\text{Data}) = \text{Cost}(\text{Data}|\text{Model}) + \text{Cost}(\text{Model})$$

  - Cost(Data|Model): the cost of encoding the misclassification errors.
  - Cost(Model): uses node encoding (number of children) plus splitting condition encoding.

Minimum Description Length (MDL) - several words more

- Given a body of data D and a representation language L, one seeks the shortest possible representation of D in L.
- Many different forms of learning can be characterized in terms of the MDL principle:
  - General rules: The assertion of a general rule eliminates the need to specify each instance.
    E.g., given a table of animals and their features, the rule "All crows are black" means that the "color" field can be omitted for each crow in the table.
  - Numerical rules. E.g., data consisting of the pairs "X=1.0, Y=3.0; X=1.2, Y=3.2; X=2.7, Y=4.7; X=5.9, Y=7.9"' can be replaced by the rule "Y = X+2.0"
  - There are other rules.

Minimum Description Length (MDL) - several words more

- Explanation of overfitting.
- The MDL theory gives an elegant explanation of why too rich representational schemes tend to overfit
  - When the encoding of the classifier itself is longer than the original data, or almost as long, then nothing is gained in terms of description length.
  - You can exactly fit a decision tree to data, if there is a separate leaf for each datum, but again no gain.

How to Address Overfitting – Pre-Pruning (Early Stopping Rule)

- Stop the algorithm before it becomes a fully-grown tree
- Typical stopping conditions for a node:
  - Stop if all instances belong to the same class
  - Stop if all the attribute values are the same
- More restrictive conditions:
  - Stop if the number of instances is less than some user-specified threshold
  - Stop if expanding the current node does not improve impurity measures or estimated generalization error. (Threshold)

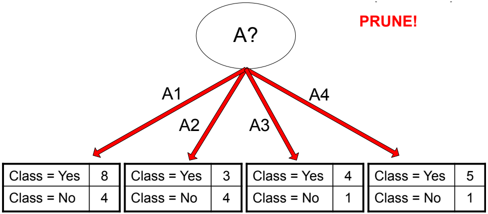
How to Address Overfitting – Post-pruning

- Grow decision tree to its entirety
- Trim the nodes of the decision tree in a bottom-up fashion
- Replace a subtree by
  - a leaf node: class label is determined from majority class of instances in the sub-tree (subtree replacement)
  - most frequently used branch of the subtree (subtree raising)
- Termination: no further improvement on generalization errors

Example of Post-Pruning

Class=Yes	20
Class=No	10
Error = 10/30	

- Training Error (Before splitting) $= 10/30$
- Pessimistic error $= (10 + 0.5)/30 = 10.5/30$
- Training Error (After splitting) $= 9/30$
- Pessimistic error (After splitting) $= (9 + 4 \times 0.5)/30 = 11/30$



Model Evaluation

- How to evaluate the performance of a model? Metrics for Performance Evaluation
- How to obtain reliable estimates? Methods for Performance Evaluation
- How to compare the relative performance among competing models? Methods for Model Comparison

Holdout

- Reserve 2/3 (half) for training and 1/3 (half) for testing
- Limitations
  - Fewer for training
  - Highly depend on the composition of training and the test sets
  - Training set is not independent of the test set

Random Subsampling

- Repeated holdout k times
- Overall accuracy:

$$acc_{sub} = \frac{\sum_{i=1}^{k} acc_i}{k}$$

- Limitations
  - Still it does not use all the original data for training.
  - No control over the number of times each record is used for testing and training.

Cross validation

- Each record is used the same number of times for training and exactly once for testing.
- General k-fold cross-validation
  - Partition data into k disjoint equal-sized subsets
  - k-fold: train on k-1 partitions, test on the remaining one
- Special case: Leave-one-out: $k = N$, the size of the data set
- Limitations
  - Computationally expensive
  - High variance in estimated performance metric

The 0.632 Bootstrap (S.S.)

- Belong to one special sampling strategy (sampling with replacement)
- Format training "set" by sampling (with replacement) N times from a dataset of N instances
  - strictly speaking, not actually a set
  - a set cannot, by definition, contain duplicates
- It is very likely that:
  - some instances in the training set will be repeated
  - some of the original instances will not have been picked
- Unpicked instances are put in test set

Related work

- Kohavi compared *random subsampling, bootstrapping, and k-fold cross-validation*.
  The best is ten-fold stratified cross-validation.

- Ron Kohavi: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In IJCAI 1995, 1137-1145.

- B. Efron and R. Tibshirani: Cross-validation and the Bootstrap: Estimating the Error Rate of a prediction Rule. Technical report, Stanford University, 1995.
  This includes theoretical and empirical comparison.

Model Evaluation

- How to evaluate the performance of a model? Metrics for Performance Evaluation
- How to obtain reliable estimates? Methods for Performance Evaluation
- How to compare the relative performance among competing models? Methods for Model Comparison

Test of Significance

- Given two models:
  - Model M1: accuracy = 85%, tested on 30 instances
  - Model M2: accuracy = 75%, tested on 5000 instances
- Can we say M1 is better than M2?
  - How much confidence can we place on accuracy of M1 and M2?

Confidence Interval for Accuracy

- Derive confidence intervals by modeling the classification task as a binomial experiment.
- Prediction can be regarded as a Bernoulli trial
  - A Bernoulli trial has 2 possible outcomes: correct or wrong
  - The probability of success p in each trial is constant
  - Collection of Bernoulli trials has a Binomial distribution:
    - $x \sim Bin(N, p)$ where x: number of correct predictions
    - mean $N \cdot p$, variance $N \cdot p \cdot (1 - p)$

Confidence Interval

- A confidence interval gives an estimated range of values which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data.
- Confidence level: if independent samples are taken repeatedly from the same population, and a confidence interval calculated for each sample, then a certain percentage (confidence level) of the intervals will include the unknown population parameter.
  Confidence intervals are usually calculated so that this percentage is 95%, but we can produce 90%, 99%, 99.9% (or whatever) confidence intervals for the unknown parameter.
- The width of the confidence interval gives us some idea about how uncertain we are about the unknown parameter.
  A very wide interval may indicate that more data should be collected before anything very definite can be said about the parameter.

Confidence Interval for Accuracy (cont.)

- Given a test set that contains N records
- Let X be the number of records correctly predicted by a model
- Let p be the true accuracy of the model
- Empirical accurace $acc = \frac{X}{N}$ has a binomial distribution
- According to the central limit theorem (CLT), for large test set ($N > 30$), acc has a normal distribution with mean p and variance $p(1-p)/N$ (i.e., $\mathcal{N}(p, \frac{p(1-p)}{N})$).

$$P(Z_{\alpha/2} \leq \frac{acc - p}{\sqrt{p(1-p)/N}} \leq Z_{1-\alpha/2}) = 1 - \alpha$$

  - $Z = \frac{acc-p}{\sqrt{p(1-p)/N}}$ is a standard normal distribution (mean 0, variance 1, i.e., $\mathcal{N}(0,1)$) from $\mathcal{N}(p, \frac{p(1-p)}{N})$
  - $Z_{\alpha/2}$ and $Z_{1-\alpha/2}$: upper and lower bounds from $\mathcal{N}(0,1)$ at confidence level $1-\alpha$.
  - $-Z_{\alpha/2} = Z_{1-\alpha/2}$ for $\mathcal{N}(0,1)$

Confidence Interval for Accuracy – Example

- Let $1-\alpha = 0.95$ (95% confidence)
- From probability table, $Z_{\alpha/2} = 1.96$

$1-\alpha$	0.99	0.98	0.95	0.9	0.8	0.7	0.5
$Z_{\alpha/2}$	2.58	2.33	1.96	1.65	1.28	1.04	0.67

- Given the observed accuracy *acc* and the size of observations N, what is the confidence interval for its true accuracy p at a 95% (1-α) confidence level?
  - Get $Z_{\alpha/2}$ from probability table using 1-α
  - Put $Z_{\alpha/2}$, N, *acc* to the formula to solve p

Confidence Interval for Accuracy (cont.)

- Confidence interval for p:

$$\frac{2 \times N \times acc + Z_{\alpha/2}^2 \pm Z_{\alpha/2}\sqrt{Z_{\alpha/2}^2 + 4N \cdot acc - 4N \cdot acc^2}}{2(N + Z_{\alpha/2}^2)}$$

- Details: Condition for $1 - \alpha$:

$$\begin{aligned}
&Z_{\alpha/2} \leq \frac{acc - p}{\sqrt{p(1-p)/N}} \leq Z_{1-\alpha/2} \\
\Longrightarrow &\frac{(acc - p)^2}{\frac{p(1-p)}{N}} \leq Z_{\alpha/2}^2 \\
\Longrightarrow &N \cdot (acc - p)^2 \leq p \cdot (1 - p) \cdot Z_{\alpha/2}^2 \\
\Longrightarrow &N \cdot acc^2 - 2N \cdot p \cdot acc + N \cdot p^2 \leq p \cdot Z_{\alpha/2}^2 - p^2 \cdot Z_{\alpha/2}^2 \\
\Longrightarrow &(N + Z_{\alpha/2}^2) \cdot p^2 - (2N \cdot acc + Z_{\alpha/2}^2) \cdot p + N \cdot acc^2 \leq 0
\end{aligned}$$

Confidence Interval for Accuracy (cont.)

Solve:

$$(N + Z^2_{\alpha/2}) \cdot p^2 - (2N \cdot acc + Z^2_{\alpha/2}) \cdot p + N \cdot acc^2 \leq 0$$

Two roots for $ax^2 + bx + c = 0$ are $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.
Thus,

$$\begin{aligned}
b^2 - 4ac = & \quad (2N \cdot acc + Z^2_{\alpha/2})^2 - 4 \cdot (N + Z^2_{\alpha/2}) \cdot (N \cdot acc^2) \\
= & \quad 4 \cdot N^2 \cdot acc^2 + 4 \cdot N \cdot acc \cdot Z^2_{\alpha/2} + Z^4_{\alpha/2} - 4 \cdot N^2 \cdot acc^2 - 4 \cdot N \cdot acc^2 \cdot Z^2_{\alpha/2} \\
= & \quad 4 \cdot N \cdot acc \cdot Z^2_{\alpha/2} + Z^4_{\alpha/2} - 4 \cdot N \cdot acc^2 \cdot Z^2_{\alpha/2} \\
= & \quad Z^2_{\alpha/2} \cdot (4 \cdot N \cdot acc + Z^2_{\alpha/2} - 4 \cdot N \cdot acc^2)
\end{aligned}$$

$$\begin{aligned}
p = & \quad \frac{(2N \cdot acc + Z^2_{\alpha/2}) \pm \sqrt{b^2 - 4ac}}{2(N + Z^2_{\alpha/2})} \\
= & \quad \frac{(2N \cdot acc + Z^2_{\alpha/2}) \pm Z_{\alpha/2} \cdot \sqrt{4 \cdot N \cdot acc + Z^2_{\alpha/2} - 4 \cdot N \cdot acc^2}}{2(N + Z^2_{\alpha/2})}
\end{aligned}$$

Confidence Interval for Accuracy – Example

Consider a model that produces an accuracy of 80% when evaluated on 100 test instances:

- Let $1-\alpha = 0.95$ (95% confidence),from probability table, $Z_{\alpha/2} = 1.96$

$1-\alpha$	0.99	0.98	0.95	0.9	0.8	0.7	0.5
$Z_{\alpha/2}$	2.58	2.33	1.96	1.65	1.28	1.04	0.67

- $N = 100$, $acc = 0.8$
- Put $Z_{\alpha/2}$, N, acc to the formula of p
- Confidence intervals for different N (71.1%, 86.7%)?

N	50	100	500	1000	5000
p(lower)	0.670	0.711	0.763	0.774	0.789
p(upper)	0.888	0.866	0.833	0.842	0.811

- The confidence interval is tighter when N increases.

References

- Chapter 3: Introduction to Data Mining (2nd Edition) by Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar
- ROC: https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html