Introduction to Statistical Machine Learning
CSC/DSCC 265/465

Lecture 8: Supervised Learning – Part V

Cantay Caliskan

Notes and updates

Notes and updates

- Average for ***PS1*** is ***92.37***.
    - Names of graders posted on BlackBoard (under '*List of Graders*')

- Let me know if you are looking for additional resources

- If you have grading-related questions:
    - List of graders for ***PS1*** have been shared through an announcement
    - <u>Please</u>:
        - Contact your grader first
        - If more clarification is needed, contact our head TA (***Jawahar***)
        - No need to CC me for grading-related questions

- New deadline for the 3rd Problem Set is **Sunday, February 13, 11:59 PM**

Plan for today

- ***Bias and Variance***
- ***Cross-Validation***
- ***Multinomial Logistic Regression***

Plan for today

- ***Bias and Variance***
- ***Cross-Validation***
- ***Multinomial Logistic Regression***

Bias and Variance

UNIVERSITY of ROCHESTER

Bias and Variance

- <u>Goal</u>: Hit the target!

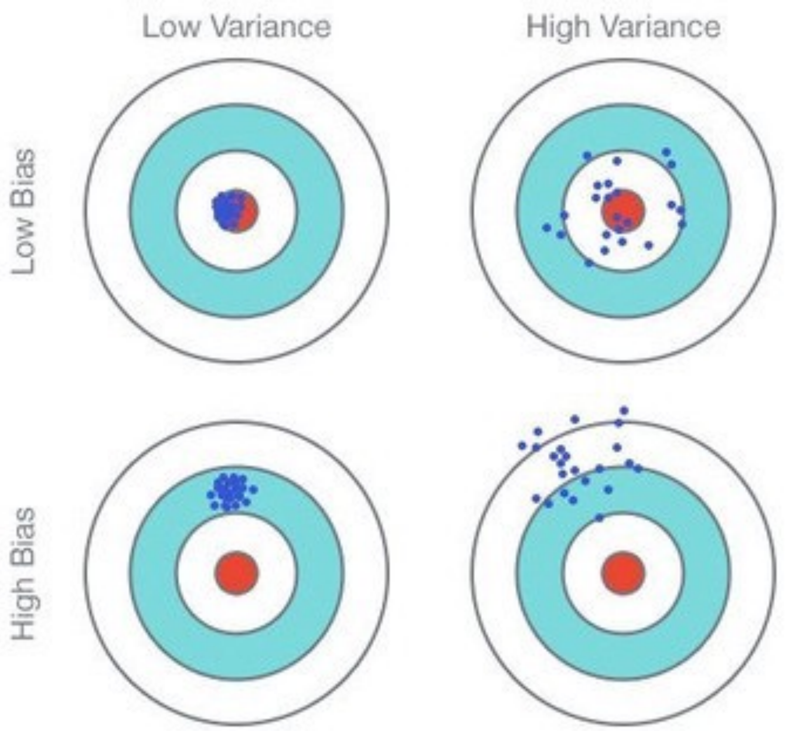


Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off

Bias and Variance

- Idea: Understanding how *bias* and *variance* happen can help us manage reducing them

- ***Error due to bias:*** Calculated as the difference between the expected prediction of our model and the correct value (=ground truth)

- ***Error due to variance:*** The variance shows how much the predictions for a given point vary between different realizations of the model

- Question: When is bias more likely? When is variance more likely to happen?

Bias and Variance

- There is a trade-off:

- ***Less complex*** models (=models with fewer parameters) have high bias and low variance
  - *Not accurate* but *generalizable*

- ***More complex*** models (=models with more parameters) have low bias and high variance
  - *Accurate* but *not generalizable*

- Question: Where is the optimal model?
- Answer: Somewhere in between

Bias and Variance: Which is worse?

- Most data scientists would try to lower bias at the expense of variance
  - Idea: If I am making the prediction sometimes correctly, this is better than getting it wrong all the time

- Answer: **Yes** and **No**
  - You can only understand the effects of bias and variance if you realize a model a lot of times
  - In fact: Most people realize a model only once
  - Also: Think about the application

- If you only realize the model once: Long run averages are irrelevant. Bias and variance are equally important.

Bias and Variance

- Ways to deal with ***variance***
  - Running several similar models at once / in parallel
    - <u>Examples</u>: Ensemble methods, Bagging, Random Forest
  - Apply ***regularization***
- Ways to deal with ***bias***
  - Increasing complexity
  - Prefer overfitting over underfitting
  - Known as model selection

Plan for today

- *Bias and Variance*
- ***Cross-Validation***
- *Multinomial Logistic Regression*

UNIVERSITY of ROCHESTER

Cross-Validation

Cross-Validation

- Definition: Different model validation techniques for assessing how the results of a statistical analysis (model) will generalize to an independent data set
- Important: It is not a tool for estimating model coefficients

- Usually used in the context of ***prediction***

- Why is it helpful? What is the goal?
  - Helps us to evaluate the quality of the model
  - Helps us to select the model which will perform best on unseen data
  - Helps us to avoid ***overfitting*** and ***underfitting***
  - Helps us to have a model that is low on ***bias*** and ***variance***

Cross-Validation

- Idea:
  - You have a ***sample dataset***
    - This dataset represents the characteristics in the ***whole population***
  - This sample dataset may have some ”random” differences from the population data
  - Goal: To create an explanatory / predictive algorithm from the sample dataset
    - The algorithm will almost always perform less well when applied on the whole population
    - Why? Because of random error …
  - You do cross-validation -> To check the ***amount of error*** resulting from the random errors

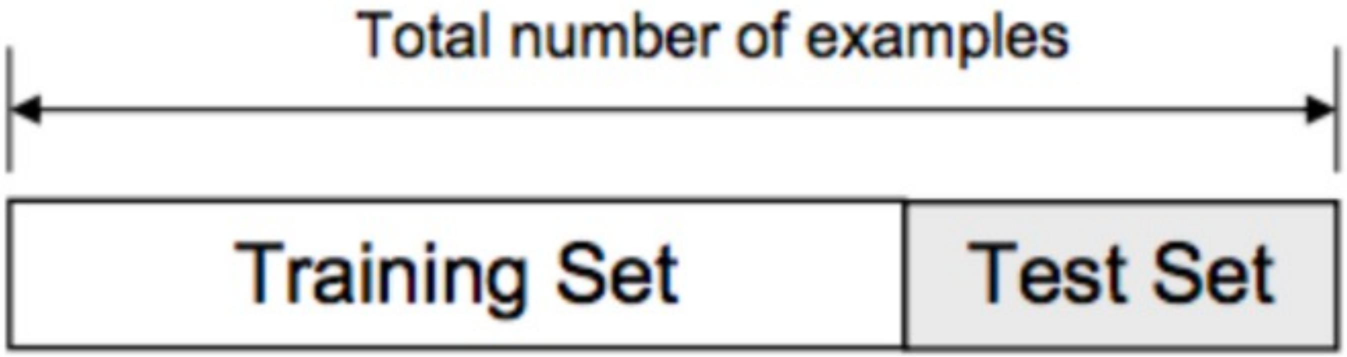
Cross-Validation Strategies

1. **Train / Test split**
2. **K–fold cross-validation**
3. **Leave one group out**
4. **Stratification**

Train / Test Split

- **Train / Test Split**
  - Number of groups: **2**

  - We split the data into two sets: Training and test data
  - Samples between train and test data sets do not overlap
  - Important not to have duplicate samples in our dataset



Train / Test Split

- **Train / Test Split** (or Holdout)
  - Disadvantage:
    - What if the train / test split isn’t random
      - This will result in ***overfitting***

- A good choice, if we have enough data

- Implementatation in **Python**:
  sklearn.model_selection.train_test_split

Pseudocode

- How should we go forward?

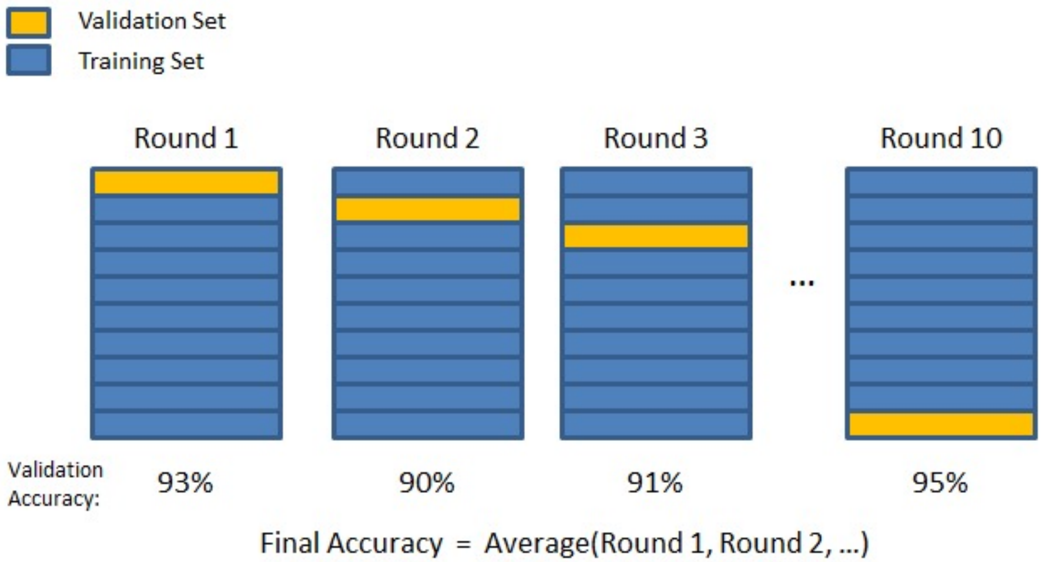
1) **Split** the data into training test and validation datasets
2) Find a set of **parameters** that you think would work well with the model
3) Run the model by using a **cost function**
4) Check the cost value obtained from the **test dataset**
5) Report the cost value

K-fold cross-validation (#groups = k)

- Removing a part of the dataset for validation means:
  - Risk of underestimating the amount of ***overfitting***
- Reducing the size of the training data means:
  - Risk of overestimating the amount of ***underfitting***
- <u>Solution</u>: K-fold cross-validation
  - Provides ample data for training the model
  - Leaves a lot of data for validation
- <u>Idea</u>: repeated holdout, and average scores after **K** different holdouts

K-fold cross-validation (#groups = k)

- A good choice when we have:
  - Low amount of data / small data set
  - When the choice of optimal parameters are greatly different between folds



  - As a general rule, we choose **k=5** or **k=10** …
  - Implementation in **Python**: sklearn.model_selection.KFold

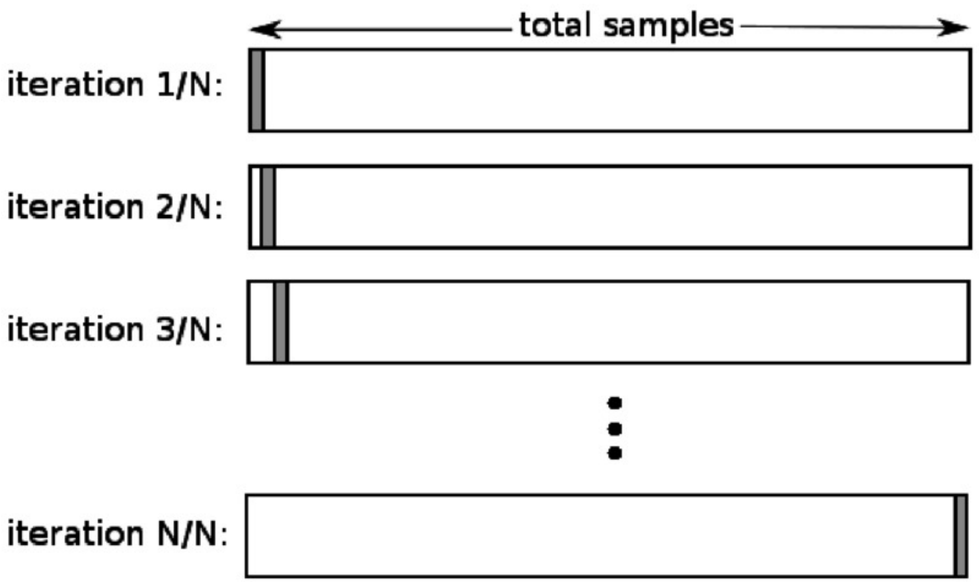
Pseudocode (when we have #k folds)

- What is difference here?

1) **Split** the data into training and test datasets
2) Find a set of **parameters** that you think would work well with the model
3) Place the parameters in a loop structure
4) Place the folds in a nested loop structure. Create training and validation datasets from each fold
5) Fit the model in each smaller **training dataset**
6) Report the **cost** for the test dataset for **each fold**
7) Report the **average cost** at the end

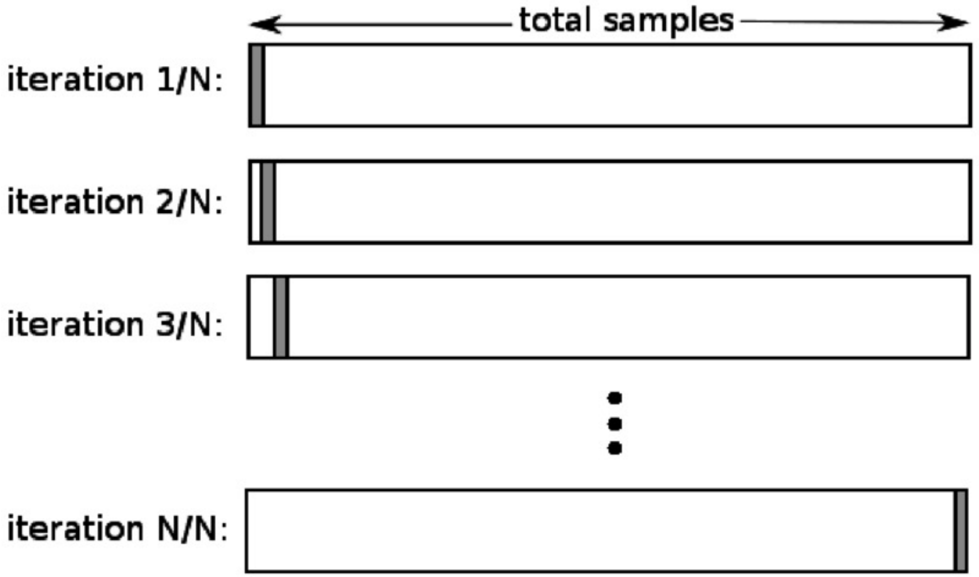
Leave one out (#groups = len(train)) - LOOCV

- It is a special case of *K-fold* when *K* is equal to the number of samples in our data set
  - We will iterate through every sample in our dataset each time each time using k-1 object as train samples and 1 object as test set



Leave one out (#groups = len(train)) - LOOCV

- Useful, when:
  - We have too little data
  - And the model is fast enough to retrain
- Implementation in **Python**:
  - sklearn.model_selection.LeaveOneOut



Stratification / Stratified Cross-Validation

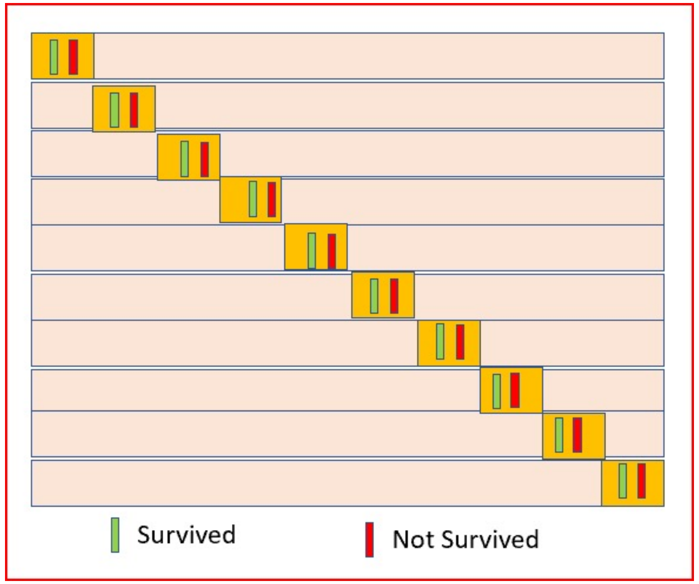
- Also called ***"Stratified k-fold Cross-Validation"***

- Reminder: ***Stratification*** OR ***stratified sampling***
  - Process of dividing members of population into homogenous subgroups before sampling

- Idea: We re-arrange the data in a way that each fold has a good representation of the whole dataset
  - Stratification forces each fold to have at least some number instances (let's call this ***m***) of each class
  - Ensures that one class of data is not overrepresented especially when the target variable is unbalanced

Example: Stratification

- In a binary classification problem, we want to make sure that each fold has enough number of observations from each class
- Example: Titanic Survivors (!)
  - Survived or not survived
- In a potential ***k-fold*** stratification, each fold needs to have passengers of both sorts



Time series cross-validation

- Splitting time series data randomly is a bad idea:
  - Because of the time dimension

- <u>Solution</u>: Use each group of data created at time ***t*** as the training set for each group of data created at time ***t+1***

Time series cross-validation

- We start using smaller training datasets, and make them grow as we go forward …
- This ensures that we consider the time series aspect of the data for prediction

Which kind of cross validation?

	Downside	Upside
Test-set	may give unreliable estimate of future performance	cheap
Leave-one-out	expensive	doesn't waste data
10-fold	wastes 10% of the data,10 times more expensive than test set	only wastes 10%, only 10 times more expensive instead of **n** times
3-fold	wastes more data than 10-fold, more expensive than test set	slightly better than test-set
N-fold	Identical to Leave-one-out	

Training, Test, Validation Split Ratio

- The <u>split ratio</u> depends on a few factors:
  - The total number of samples in your data
  - The actual model you are training

- Usual practice:
  - Split by **70% by 30%** or **80% by 20%**
  - The <u>larger set</u> is the training set
  - And then (you may) choose a validation set from the training set

Evaluating the model performance

- **Mean Squared Error (MSE)**
- **Root Mean Squared Error (RMSE)**
- **Mean Absolute Percentage Error (MAPE)**

- These are all cost functions ☺

Evaluating the accuracy

- We need to understand how well the models give the correct classification
  - And then measure the value of prediction

- This brings us to ***confusion matrix*** …

- Confusion matrix visualizes the performance of an algorithm by showing true positives, false positives, true negatives, and false negatives

Confusion matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

- <u>Goal (usually)</u>: Look for a balance between sensitivity and specificity

Confusion matrix

Confusion Matrix		Target			
		1	0		
Logit	1	261	64	Positive Predictive Value	0.803076923
	0	81	485	Negative Predictive Value	0.856890459
		Sensitivity	Specificity	Accuracy=	
		0.763157895	0.88342441	0.837261504	

Titanic Survivors

	Predicted Labels										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
True Labels	**0**	987	1	2	0	0	0	2	0	7	1
	1	0	977	7	2	3	2	0	2	6	1
	2	2	3	976	4	4	0	1	4	6	0
	3	0	1	18	951	0	14	0	3	9	4
	4	0	1	2	0	979	0	2	0	3	13
	5	3	0	3	9	5	968	2	0	5	5
	6	1	3	2	0	0	7	982	0	5	0
	7	3	4	3	0	13	0	0	969	0	8
	8	2	6	4	7	3	5	2	3	966	2
	9	1	1	2	6	12	2	0	8	5	963

MNIST Hand-written Digits

- Harder to be accurate with higher number of classes
- Goal: Always look for a balance in misclassifications

Even better: F1

- If we are looking for “some kind of” balance between sensitivity and specifity:
  - We can use the ***F1 score***
  - Also called F-score or F-measure

- <u>Formally</u>: Harmonic mean of ***precision*** and ***recall***

$$F1 = \frac{2\,TP}{2\,TP + FP + FN}$$

Reminders

- Please work on the problem set
- Read ***Chapter 12***