COMP90051

Workshop Week 02

About the Workshops

- 7 sessions in total
  - Tue 12:00-13:00 AH211
  - Tue 12:00-13:00 AH108 *
  - Tue 13:00-14:00 AH210
  - Tue 16:15-17:15 AH109
  - Tue 17:15-18:15 AH236 *
  - Tue 18:15-19:15 AH236 *
  - Fri 14:15-15:15 AH211

About Me (Yuan Li)

- 2008-2012, THU, EE
- 2013-2014, UoM, Master of IT
- 2015-present, UoM, PhD CS

- Working in the NLP group
- Supervisors: Trevor Cohn, Ben Rubinstein

- Contact: yuanl4@student.unimelb.edu.au

About the Workshops

- Two parts
  - Review the lecture, background knowledge, etc.

  - Run the ipython notebook files
    - Released on subject homepage
    - https://trevorcohn.github.io/comp90051-2017/workshops

    - Illustrate the ideas. Some “IMPLEMENT ME” to work on.

```
def neighbours(x, train_x, k):
    # IMPLEMENT ME to return the indices of
    # the k closest elements to x in train_x
```

Outline

- Review the lecture, background knowledge, etc.
  - Overfitting
  - Model evaluation (Metrics, Train/Test split)
- Setup the environment (to run the notebook)
  - We release workshop materials in this format
- Run the notebook files
  - Task 1: Overview of the k-NN classifier, overfitting
  - Task 2: Evaluation the classifier (metrics, train/test split)

Outline

- Review the lecture, background knowledge, etc.
  - Overfitting
  - Model evaluation (Metrics, Train/Test split)

- Setup the environment (to run the notebook)
  - We release workshop materials in this format

- Run the notebook files
  - Task 1: Overview of the k-NN classifier, overfitting
  - Task 2: Evaluation the classifier (metrics, train/test split)

Overfitting

- When do we realize overfitting?
- What kind of model is easy to overfit?
- How to reduce overfitting?

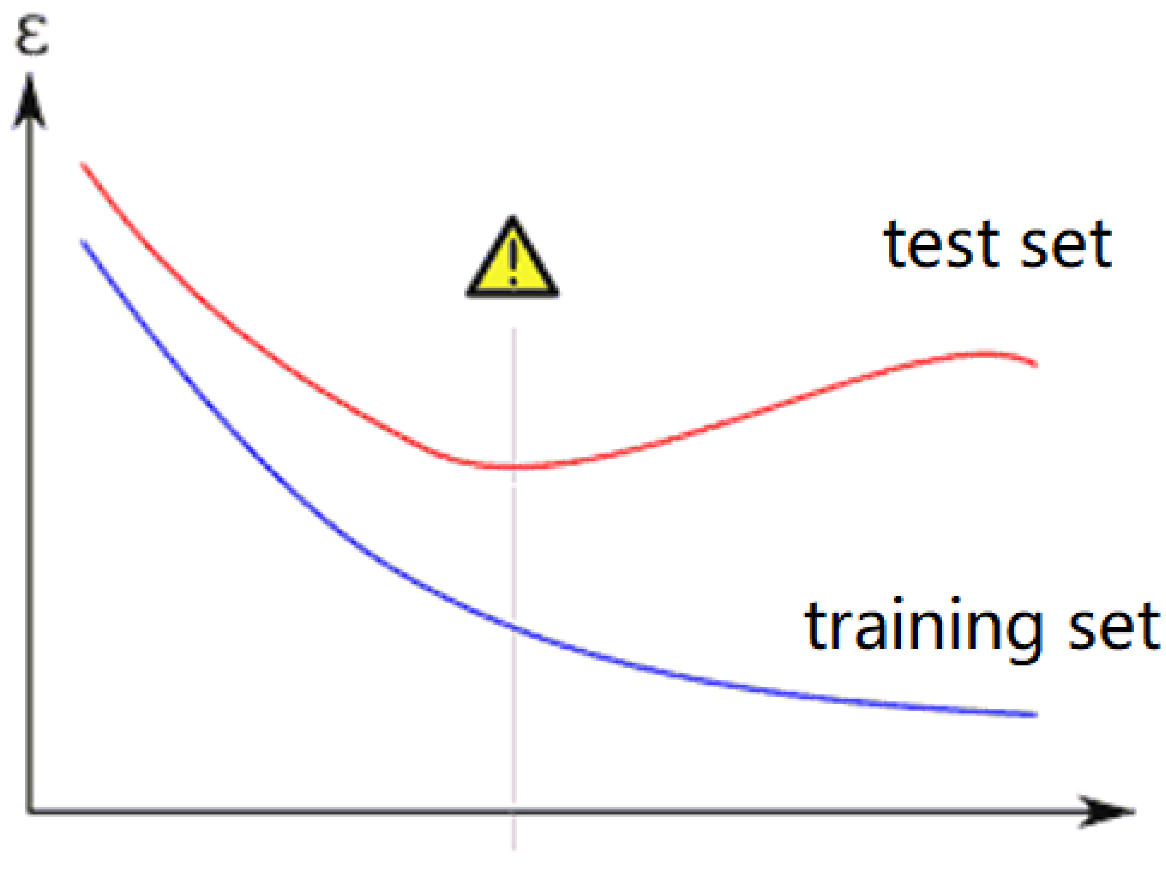
Overfitting

- When do we realize overfitting?
- What kind of model is easy to overfit?
- How to reduce overfitting?

- Analyze overfitting using plots
  - Train/Test plot
  - Bias/Variance plot
- Common beliefs on overfitting
- Ways to reduce overfitting

Overfitting – Train/Test plot

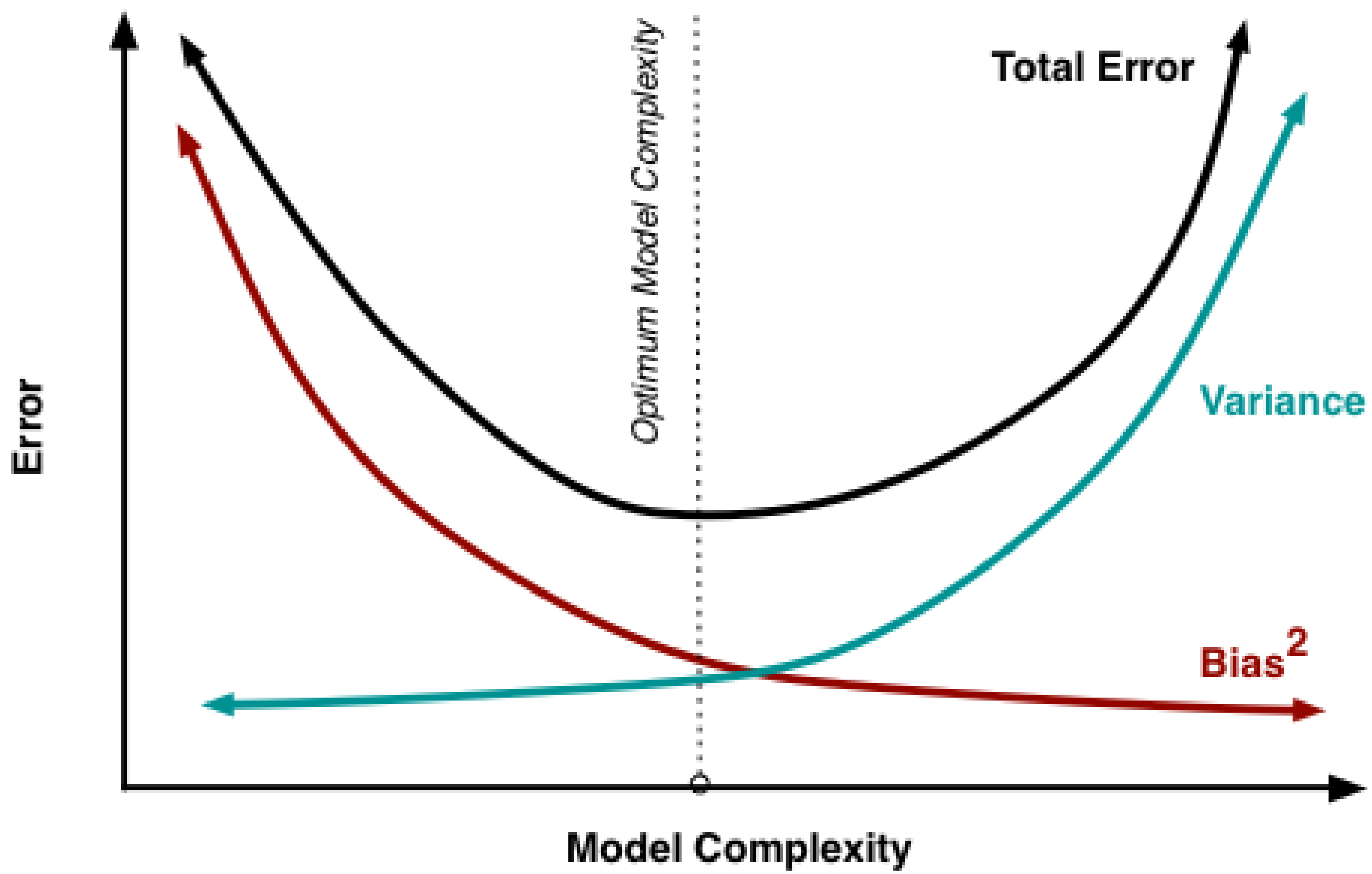
- Y-axis is error/loss, etc. The lower the better.
- X-axis could be
  - A parameter in the model
  - The number of iterations in the training algorithm.



https://en.wikipedia.org/wiki/Overfitting

Overfitting – Bias/Variance plot

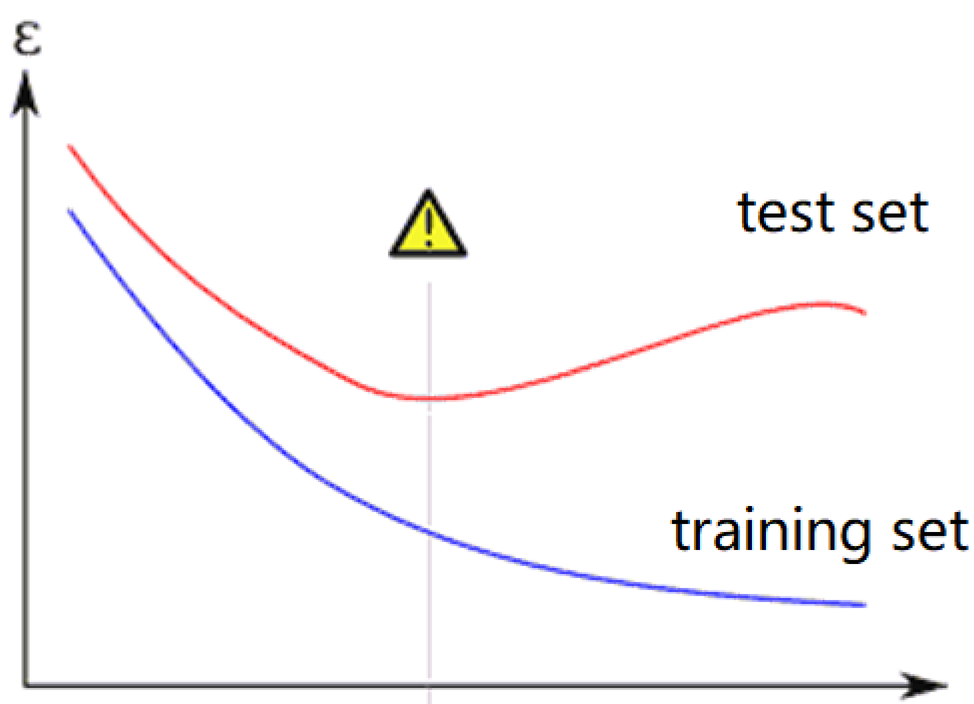
- Y-axis is the error. The lower the better.
- X-axis is the model complexity.

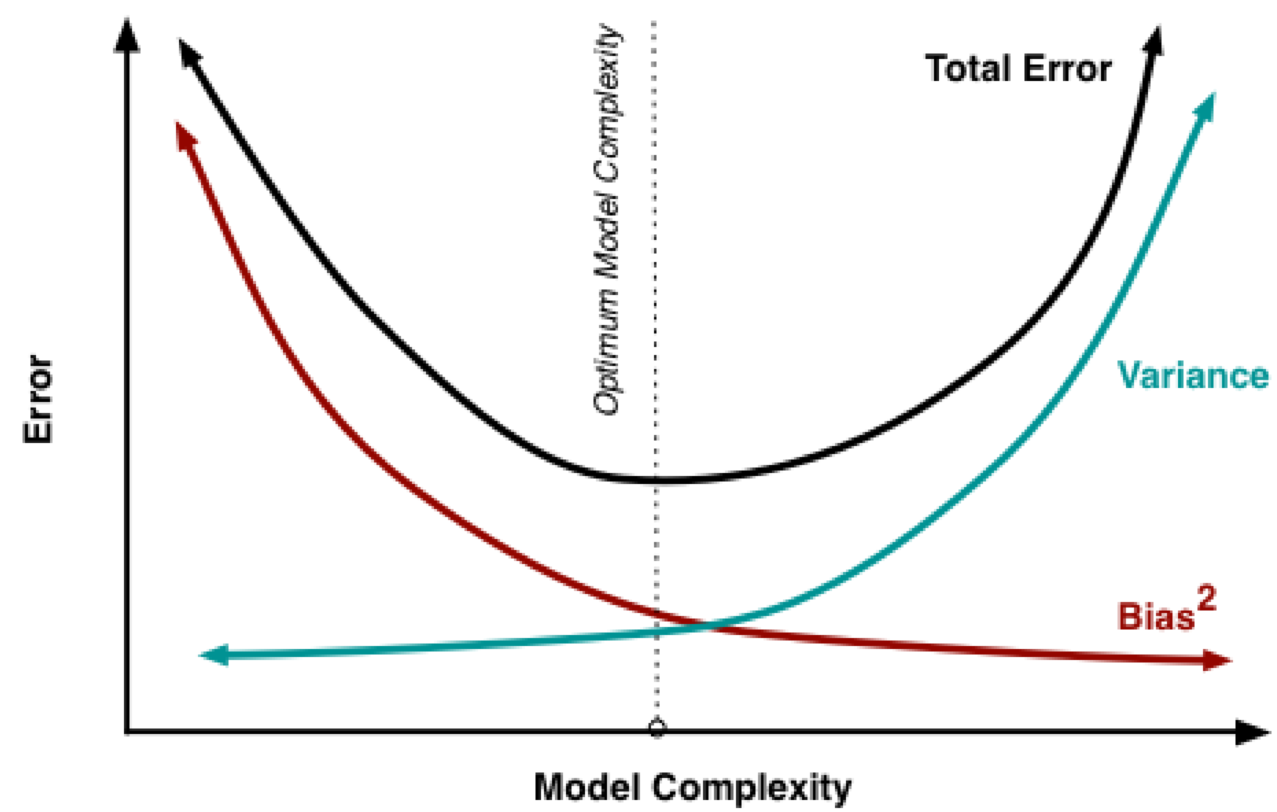


http://scott.fortmann-roe.com/docs/BiasVariance.html

Comparison

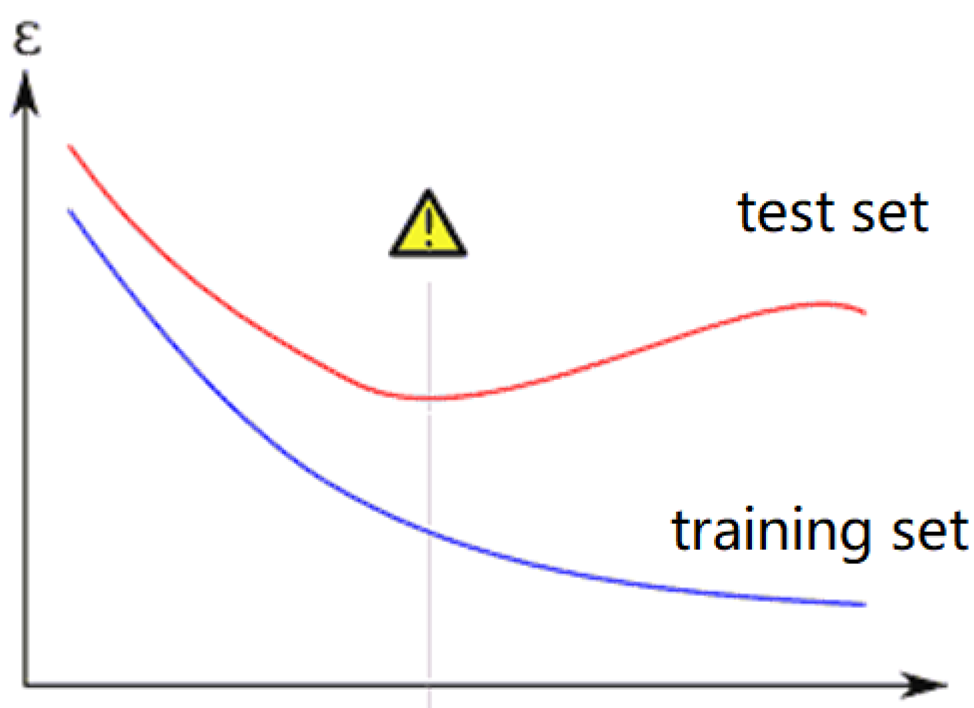
- The train/test plot is more practical.
- The bias/variance plot is more theoretical.
- The total error is in theory.
- The error on the test set is an approximation to it.

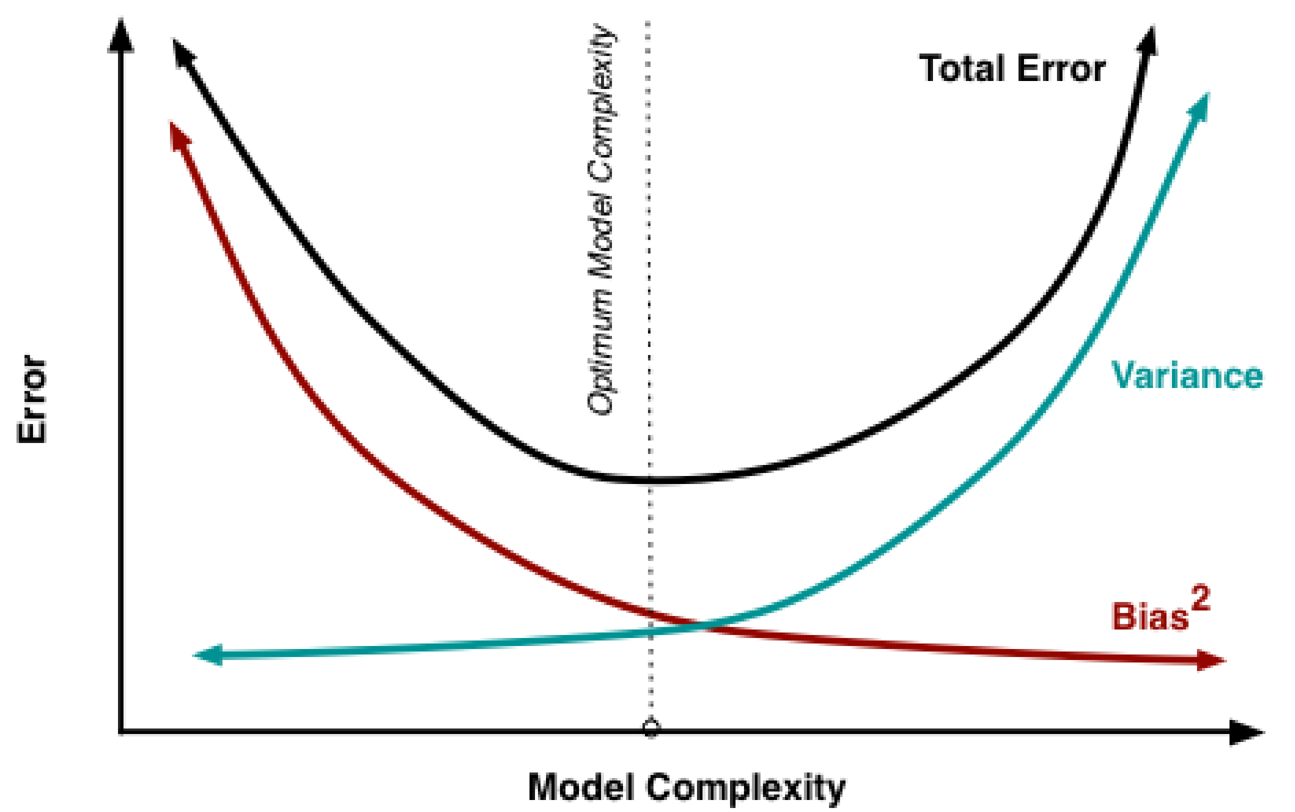




Comparison

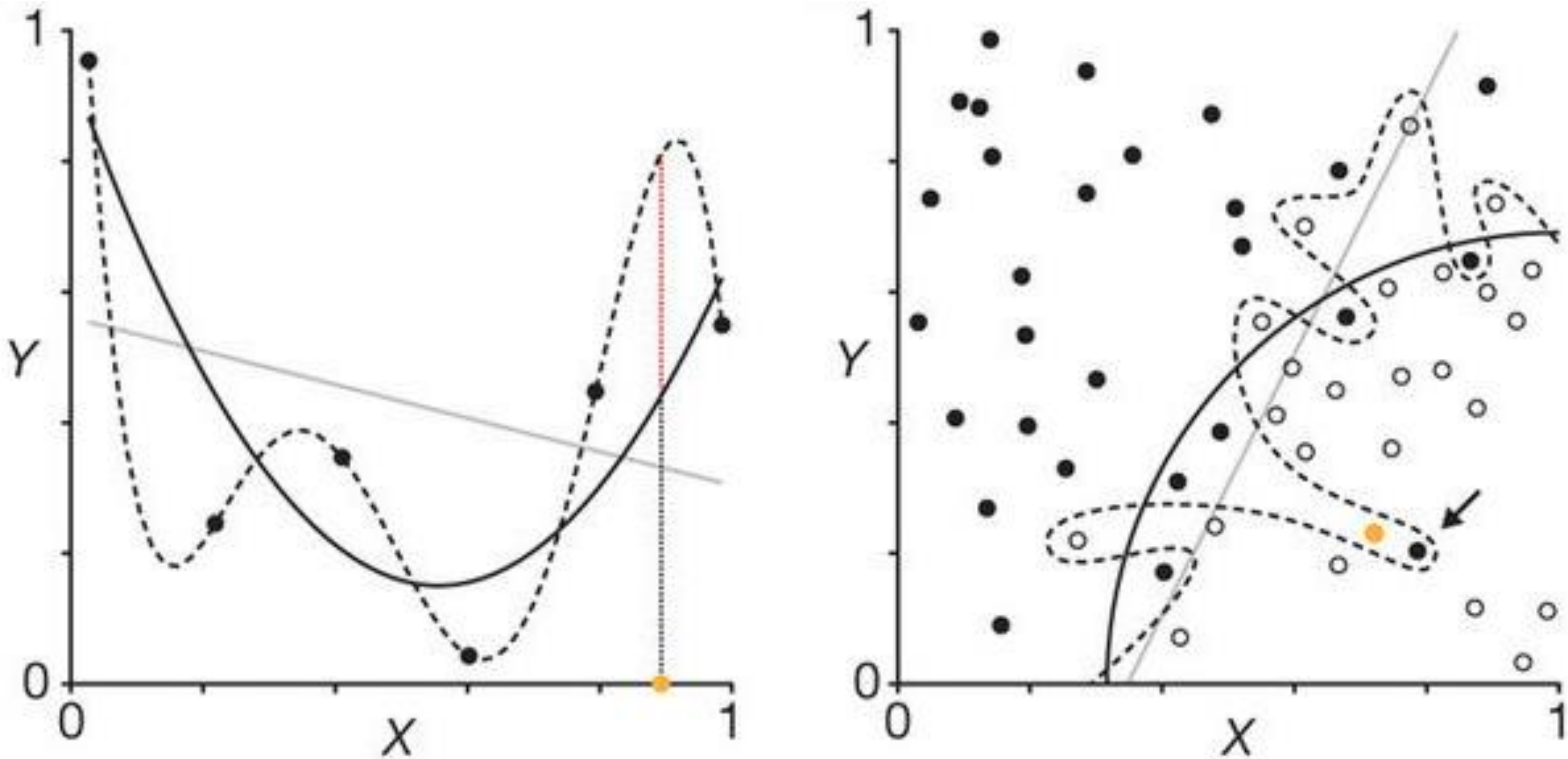
- The common thing is
- model complexity ↑, total error/test error first ↓, then ↑





Beliefs on Overfitting

- Note: beliefs are not always correct.
- Beliefs on the smoothness



https://www.nature.com/nmeth/journal/v13/n9/fig_tab/nmeth.3968_F1.html

Beliefs on Overfitting

- Note: beliefs are not always correct.
- Beliefs on the model complexity
  - Simpler model has lower risk of overfitting.
- Beliefs on the number of parameters
  - Should not exceed the number of examples.
- Beliefs on the sparseness of learned parameters
  - The more sparse, the less likely to overfit.
  - Lead to L1/L2 regularization.

Reduce Overfitting

- More training data.
- Limit the model to have the right complexity.
  - Limit the number of parameters
  - Regularizations
- Average many different models.
  - For example, random forest
- Bayesian approaches.
  - Add prior belief to the model

http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec9.pdf

Outline

- Review the lecture, background knowledge, etc.
  - Overfitting
  - Model evaluation (Metrics, Train/Test split)
- Setup the environment (to run the notebook)
  - We release workshop materials in this format
- Run the notebook files
  - Task 1: Overview of the k-NN classifier, overfitting
  - Task 2: Evaluation the classifier (metrics, train/test split)

Evaluation Metrics

- Given two arrays:
  - y_true = [1, 0, 0, 1, 1, 0, 0]
  - y_pred = [0, 1, 0, 1, 0, 0, 0]

- Result in a table

Truth	Prediction	Count	Name
1	1		
1	0		
0	1		
0	0		

Evaluation Metrics

- Given two arrays:
  - y_true = [1, 0, 0, 1, 1, 0, 0]
  - y_pred = [0, 1, 0, 1, 0, 0, 0]

- Result in a table

Truth	Prediction	Count	Name
1	1	1	
1	0	2	
0	1	1	
0	0	3	

Evaluation Metrics

- Accuracy = (1+3) / (1+2+1+3) = 4/7
- Precision = TP/(TP+FP) = 1/(1+1) = 1/2
- Recall = TP/(TP+FN) = 1/(1+2) = 1/3, is also called TPR
- FPR = FP/(FP+TN) = 1/(1+3) = 1/4

Truth	Prediction	Count	Name
1	1	1	TP
1	0	2	FN
0	1	1	FP
0	0	3	TN

True/False Positive/Negative

- Positive/Negative -> the prediction is positive/negative
- True/False -> the prediction is correct/wrong

- False Positive -> the prediction is positive but is wrong

Summary

- https://en.wikipedia.org/wiki/Confusion_matrix

		True condition		
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$
Predicted condition	Predicted condition positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	

Train/Test Split

- If there is no test set, we may create one.
  - To evaluate our model, or diagnose overfitting, etc.
- By splitting the whole dataset into train and test.

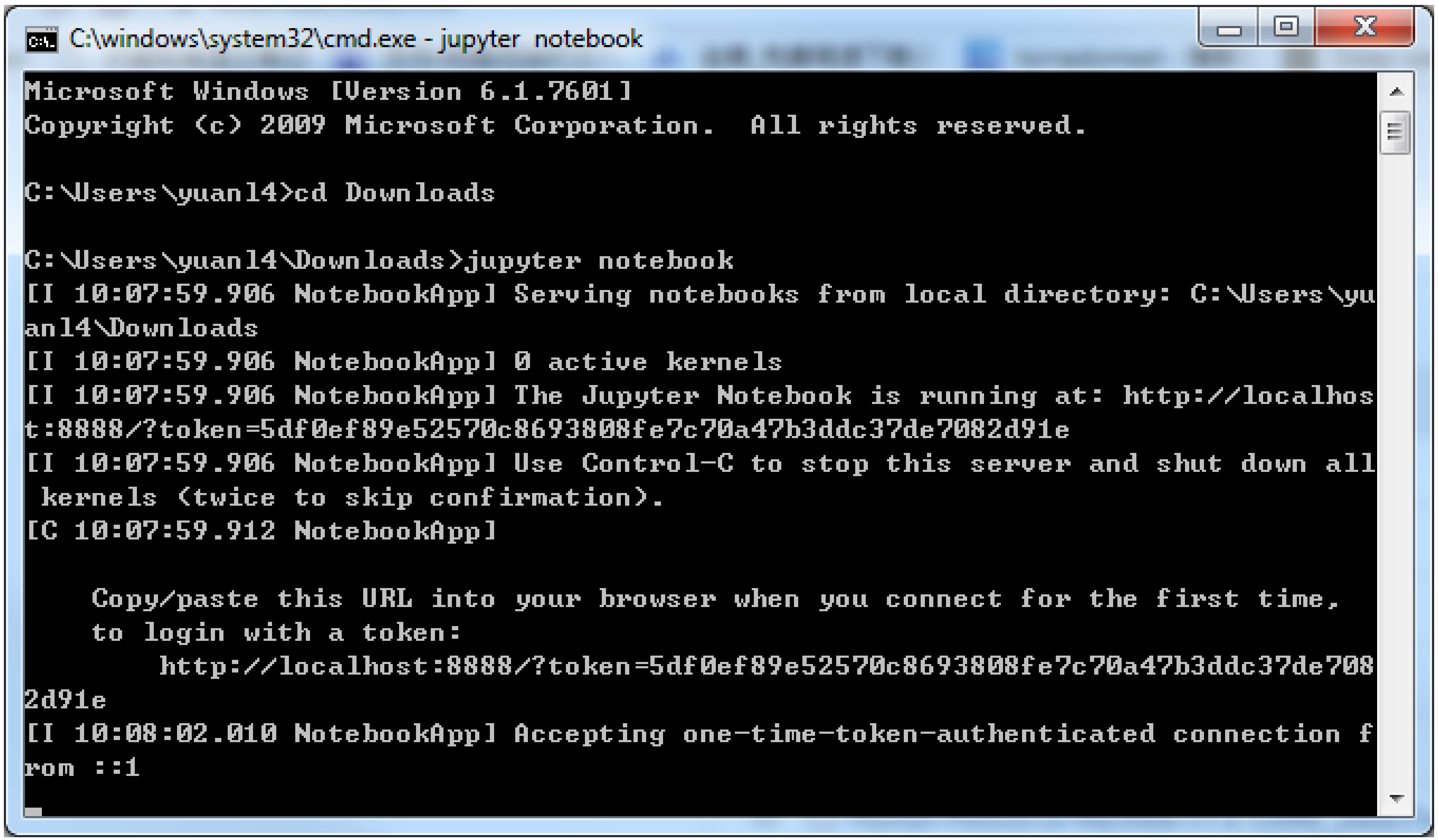
- Three ways
  - Split once at the beginning.
    - The test set is also called development set or validation set
  - Split k times to create k-fold -> cross validation
  - Leave one out -> an extreme case
    - where k = N, N is the number of examples in total

Outline

- Review the lecture, background knowledge, etc.
  - Overfitting
  - Model evaluation (Metrics, Train/Test split)
- Setup the environment (to run the notebook)
  - We release workshop materials in this format
- Run the notebook files
  - Task 1: Overview of the k-NN classifier, overfitting
  - Task 2: Evaluation the classifier (metrics, train/test split)

To launch jupyter on the lab computer.

- Open a command line prompt
- “cd” to your working directory
- Type “jupyter notebook”



C:\windows\system32\cmd.exe - jupyter notebook

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\yuanl4>cd Downloads

C:\Users\yuanl4\Downloads>jupyter notebook
[I 10:07:59.906 NotebookApp] Serving notebooks from local directory: C:\Users\yu
anl4\Downloads
[I 10:07:59.906 NotebookApp] 0 active kernels
[I 10:07:59.906 NotebookApp] The Jupyter Notebook is running at: http://localhos
t:8888/?token=5df0ef89e52570c8693808fe7c70a47b3ddc37de7082d91e
[I 10:07:59.906 NotebookApp] Use Control-C to stop this server and shut down all
 kernels (twice to skip confirmation).
[C 10:07:59.912 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=5df0ef89e52570c8693808fe7c70a47b3ddc37de708
2d91e
[I 10:08:02.010 NotebookApp] Accepting one-time-token-authenticated connection f
rom ::1
```

jupyter is installed, but is not in PATH

- Windows users (now in `C:\Users\yuanl4\Downloads`)

```
C:\Users\yuanl4\Downloads>where python
C:\Program Files\Python35\python.exe

C:\Users\yuanl4\Downloads>
"C:\Program Files\Python35\Scripts\jupyter.exe"
notebook
```

- Linux/Mac users (now in `~/comp90051-2017`)

```
yuanl4@slug:~/comp90051-2017$ which python3
/home/yuanl4/python35env/bin/python3

yuanl4@slug:~/comp90051-2017$
/home/yuanl4/python35env/bin/jupyter notebook
```

jupyter is running, but no browser opened

```
C:\Users\yuanl4\Downloads>jupyter notebook

[I 15:50:13.236 NotebookApp] Serving notebooks
from local directory: C:\Users\yuanl4\Downloads

[I 15:50:13.236 NotebookApp] 0 active kernels

[I 15:50:13.236 NotebookApp] The Jupyter Notebook
is running at:
http://localhost:8888/?token=8a45ae92166791fbe4868
f6575ca958bf6ff3c300df3ab1c

[I 15:50:13.236 NotebookApp] Use Control-C to stop
this server and shut down all kernels (twice to
skip confirmation).

...
```

Outline

- Review the lecture, background knowledge, etc.
  - Overfitting
  - Model evaluation (Metrics, Train/Test split)

- Setup the environment (to run the notebook)
  - We release workshop materials in this format

- Run the notebook files
  - Task 1: Overview of the k-NN classifier, overfitting
  - Task 2: Evaluation the classifier (metrics, train/test split)