SSID: WWCode
Password: password

WOMEN WHO CODE MANILA

Machine Learning & AI Study Group

Twitter: @wwcodemanila
FB: fb.com/wwcodemanila

#WWCodeManila
#YourProgrammingLanguage
#StudyGroup

Issa Tingzon

Research Fellow

Philippine-California Advanced Research Institutes

OUR MISSION

Inspiring women to excel in technology careers.

WOMEN WHO CODE MANILA

OUR VISION

A world where women are representative as technical executives, founders, VCs, board members and software engineers.

WOMEN WHO CODE MANILA

STUDY GROUP

Study groups are events where women can come together and help each other learn and understand a specific programming language, technology, or anything related to coding or engineering.

GUIDELINES

- If you have a question, just **ask**
- If you have an idea, **share it**
- **Make friends** and learn from your study groupmates
- **Do not** promote your recruit or promote your business

New Member's Introduction

I am <name>

<your current profession>

<why did you join this study group?>

SHOW & TELL

(Handwritten Digit Recognition)

STUDY GROUPS

Study Group 1: Machine Learning Basics

Study Group 2: K-Nearest Neighbors

AGENDA

1. **Quick Review**: Machine Learning Basics
2. **New Topic**: K-Nearest Neighbor (KNN)
3. Exercise
4. Presentations

REVIEW

Intro to Machine Learning

1. Types of ML Algorithms:

- Supervised – KNOW about the data (Prediction, Classification)

- Unsupervised – DON'T KNOW about the data (Clustering, Association)

Intro to Machine Learning

2. Supervised Learning - Two main blocks:

- Training - take input **X** (*features*) and corresponding labels **y** (*classes*), outputs learned model **h**

- Prediction – take NEW DATA as input, use **h** to output corresponding predictions

Intro to Machine Learning

3. Testing a Machine Learning Algorithm:

- Splitting your dataset:
  - Training data (input to algorithm)
  - Test data (evaluation only)
  - General Splitting technique (80/20)

Intro to Machine Learning

4. Machine Learning Programming using Libraries:

1. *Import the learning algorithm* `(from sklearn…)`
2. *Instantiate the model* `(clf = SomeClassifier())`
3. *Learn the model* `(clf.fit(…))`
4. *Predict response* `(clf.predict)`
5. *Evaluate model* `(clf.score())`

Intro to Machine Learning

6. Our first machine learning project(s):

- Iris Plant Classification
- Handwritten Digit Recognition
- Bonus: Kaggle Submission

SUPERVISED LEARNING

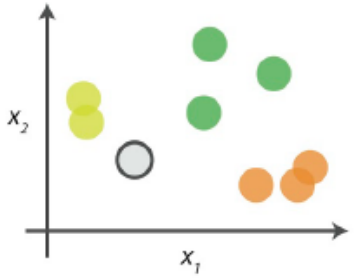
K-NEAREST NEIGHBOR

K-Nearest Neighbor (KNN)

KNN Task:

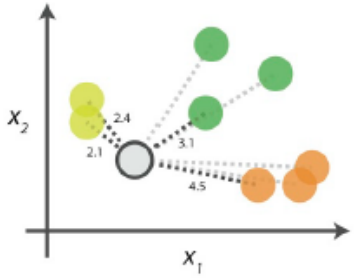
- Calculate the distance between points
- Find the nearest neighbour based on these pairwise distances
- Majority vote on a class label based on the nearest neighbor list

0. Look at the data



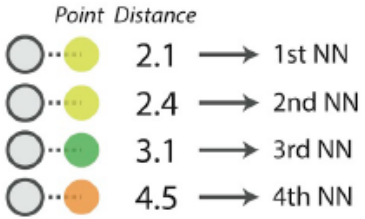
Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances



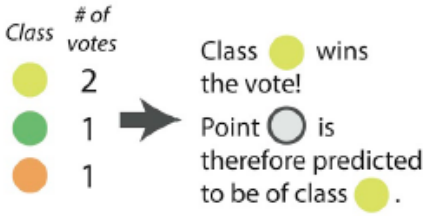
Start by calculating the distances between the grey point and all other points.

2. Find neighbours

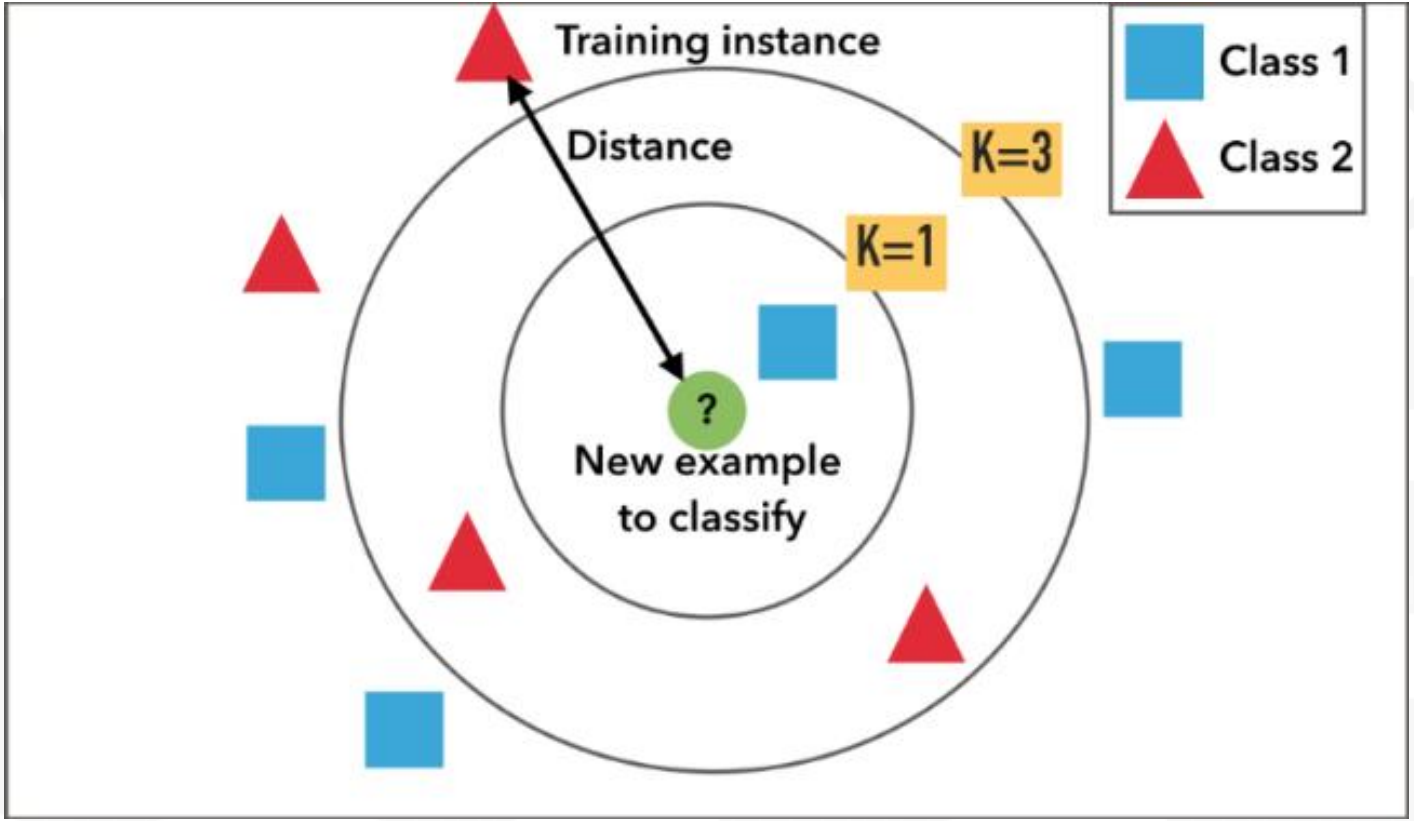


Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels



Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.



K should preferrably be odd to avoid ties.

K-Nearest Neighbor (KNN)

Distance functions:

- Euclidean Distance

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_n - x'_n)^2}$$

- Manhattan
- Chebyshev
- Hamming distance

K-Nearest Neighbor (KNN)

Train

- Do nothing. It's a lazy algorithm

Predict

- Compute the Euclidean distance between the "new observation" and all training data points
- Select the K nearest observations and perform a majority vote
- Assign the corresponding label to the observation

KNN Cheat Sheet

Importing the library:

```
from sklearn.neighbors import KNeighborsClassifier
```

Instantiating a model:

```
knn = KNeighborsClassifier(n_neighbors=3)
```

Fitting model to training set:

```
knn.fit(X_train, y_train)
```

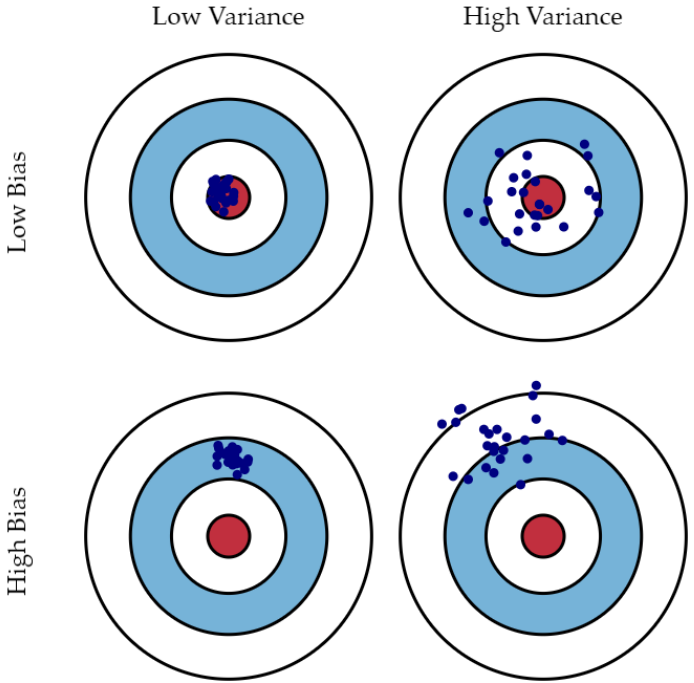
Predicting test set:

```
y_pred = knn.predict(X_test)
```

K-Nearest Neighbor (KNN)

Choosing K:

- *Small K* – low bias, high variance
- *High K* – high bias, low variance
- *Optimum K*: **Cross validation**



HYPERPARAMETER TUNING

CROSS VALIDATION

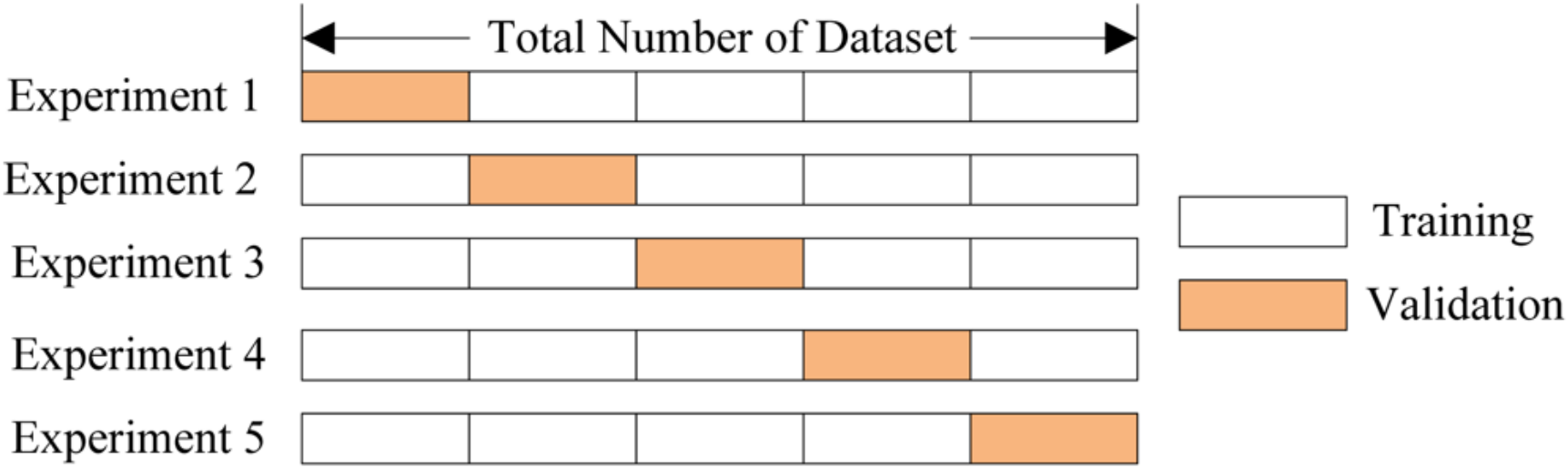
Choosing the right K

Suppose we carry out repeated measurements of the test error for different values of K.

Inadvertently, what we are doing is using the test set as a training set!

Essentially, we can't use the test set for hyperparameter tuning.

k-fold Cross Validation



The average over the *k* rounds (experiments) is called the cross validation score.

Cross Validation Cheat Sheet

Importing the library:

```
from sklearn.model_selection import cross_val_score
```

Cross Validation:

```
scores = cross_val_score(knn, X_train, y_train, cv=10,
scoring='accuracy')
```

Note: `cross_val_score` returns a list of k scores

Getting the average score: `scores.mean()`

Hyperparameter Tuning

1. Perform cross validation for different values of K (e.g. K = 1, 3, 5, 7, ..., 25)

2. Choose the K that returns the lowest misclassification error (or highest classification accuracy).

Partner/Group/Individual Exercise:

KNN FOR IRIS PLANT CLASSIFICATION

Tip: You can use `samples/iris_script.py` as a guideline

Assignment

Use KNN for Handwritten Digit Recognition

References:

WWCodeLondon Slides

https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

T.I.L.

SHARE IT!
In front!

On Twitter: @wwcodemanila
Or FB: fb.com/wwcodemanila

Don’t forget to tag WWCodeManila so we can retweet or share it.

https://goo.gl/YzSqcS

Feedback Form

Please don't rate the event on meetup.
Not helpful. It is best to just tell your concerns via the feedback form. We are a building a community not a Yelp restaurant.

THANK YOU :)