Lecture 4 Whiteboard Notes – Ensemble 1:

**Overview**

1. Single Classifiers (K-NN, Decision Tree, Naïve Bayes)
   * Can compute Accuracies and generalization errors
2. Ensemble – Combination of Classifiers
   * Expect that performance is better than random guessing
   * Expect that each classifier is diverse
     + Different types of classifiers in ensemble is called “heterogeneous ensembles”
     + Same types of classifiers in ensemble is called “homogeneous ensembles”
       - classifiers would typically have different datasets and different parameters
   * Combining outputs would:
     + maximize accuracy
     + minimize generalization errors
   * Independent Classifiers
     + Bagging
     + RF
     + Random Subspaces
   * Dependent Classifiers
     + Boosting
     + Stacking

**Intuition**

Why combining outputs gives us better performance?

Example: 5 completely independent classifiers

|  |  |
| --- | --- |
| Separate Decision Trees | Accuracies |
| DT1 | 70% |
| DT2 | 70% |
| … | … |
| DT5 | 70% |

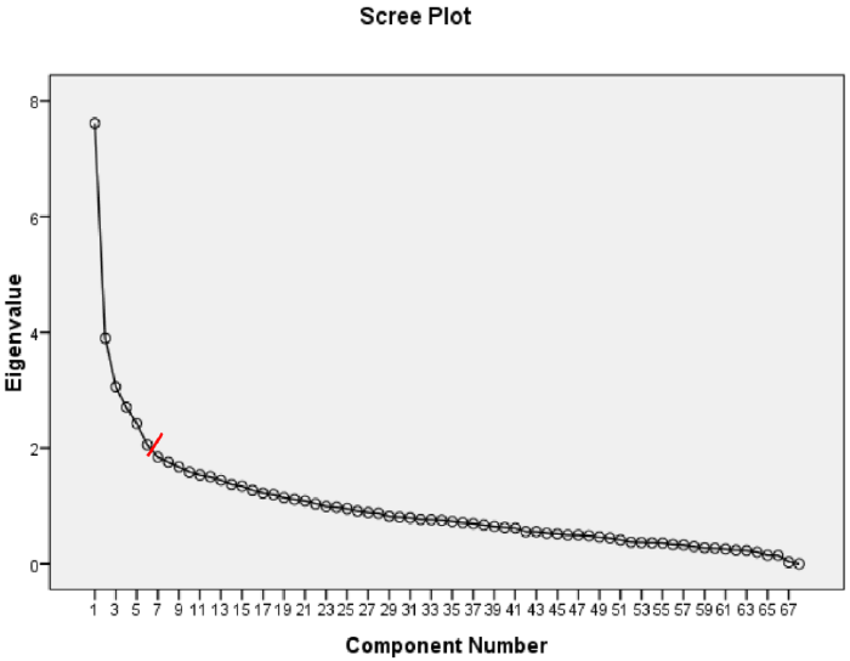
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | DT1 | DT2 | DT3 | DT4 | DT5 | **Majority Vote** |
| Instance1 | 1 | 1 | 1 | 0 | 0 | 1 |
| Instance2 | 1 | 0 | 0 | 1 | 1 | 1 |

Majority Vote Accuracy = + …..

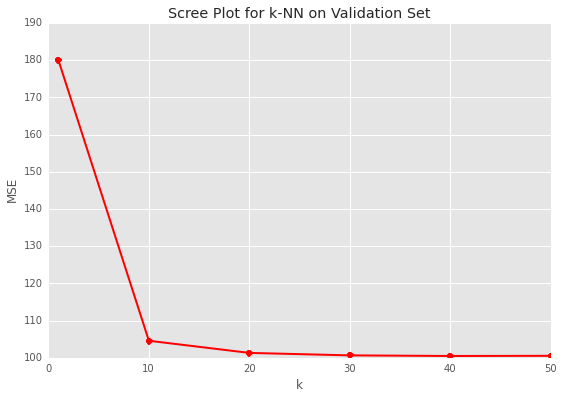
* 10 is calculated by:
* taken from Binomial Distribution
  + In the example above,
* Imagine if we have 100 classifiers instead of 5
  + More classifiers 🡪 higher accuracy

How many classifiers to choose?

* Look for “knee” in a scree plot
* Example1 : PCA

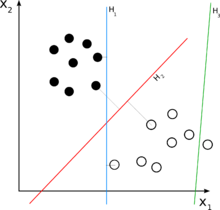


* Example2: K-NN



**Why do Ensembles Work?**

1. Statistical Problems – “Too many possibility of classifiers”
   * Hypothesis space is too large for the amount of available data
   * Hypothesis means classifier because you do not know whether that model is the best model



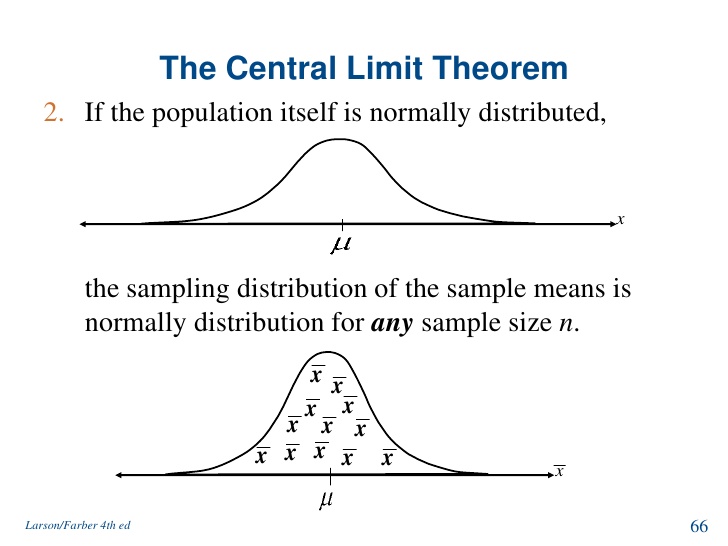
* + Imagine how many lines (single linear classifiers) can separate the black and white dots
    - Too Many 🡪 “Underfitting”

1. Computational Problem – “Limitations of ML Algorithms”
   * E.g. Decision Tree
     + No matter how I change the parameters, it does not give me the best
     + At the end, that’s the best a decision tree can do
2. Representational Problem – “Data is not well represented”
   * Dataset1 🡪 Accuracy1
   * Dataset2 🡪 Accuracy2
   * Assume Dataset 1 and Dataset 2 are taken from the same population
   * Accuracy1 is very different than Accuracy2
   * Dataset1 and Dataset 2 are not good representations of the population
   * Best only on its respective dataset, i.e. “Overfitting”

**Bias-Variance Decomposition**

Expect samples taken from population

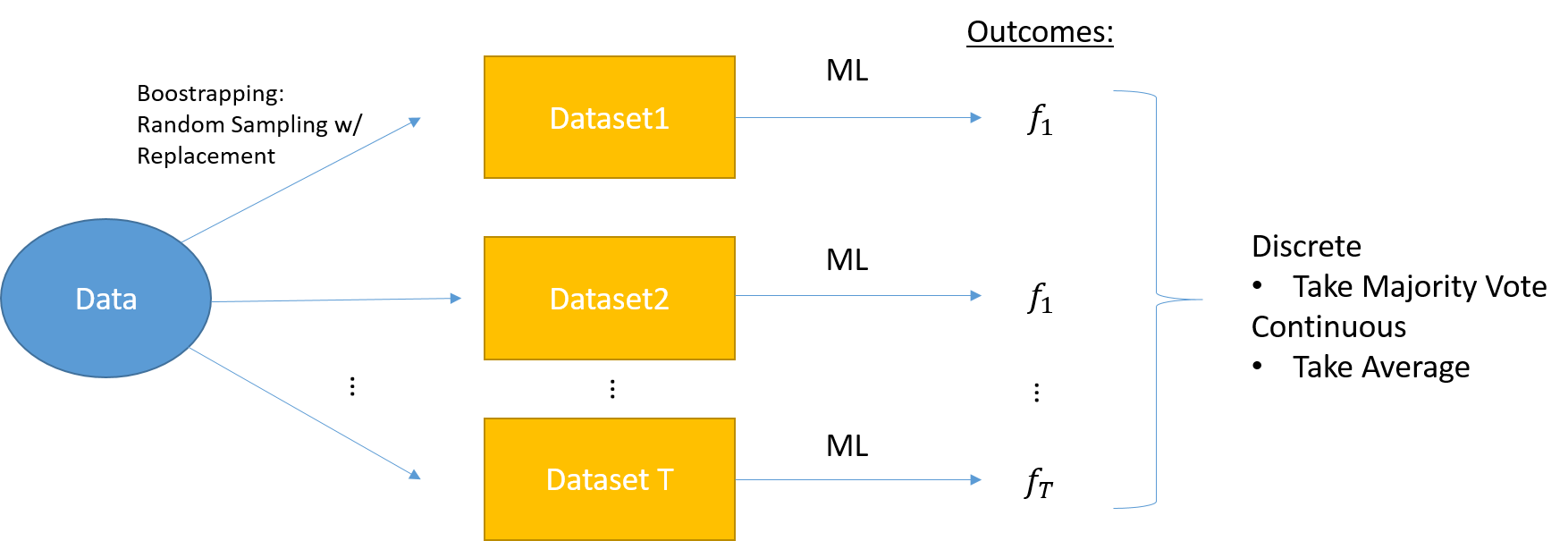
* Unbaised: When samples is plotted as a distribution, and the distribution should represent a normal distribution



* Want variance of samples to be small!
* Total Expected Error =
  + Goal: Improve bias problem in algorithmic level and improve variance problem in data level

**Independently Constructing Ensembles**

Bagging

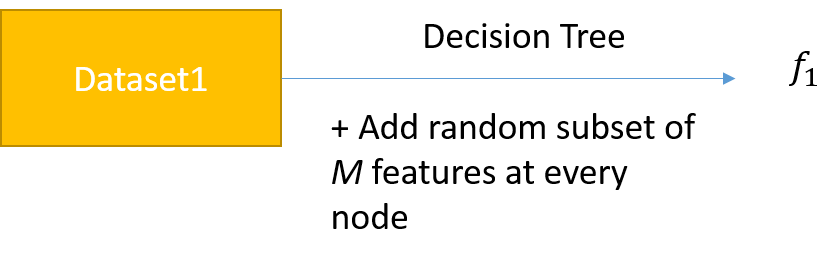


* Independently constructed, homogeneous
* might not be as good in terms of diversity
* useful when ML are unstable (Representational Problem)
* more related to variance

Randomized Injection

* Example: Decision Tree
  + At each node, choose *M* random features to create next level

Random Forest (Uses Randomized Injection)



* Will be diverted because features used to produce will be different than the ones in
* Possibly lose accuracy because not all features of used

Rotational Forest

* Accurate and Diverse
* Use PCA, then apply ML in PCA space (features are rotated)

