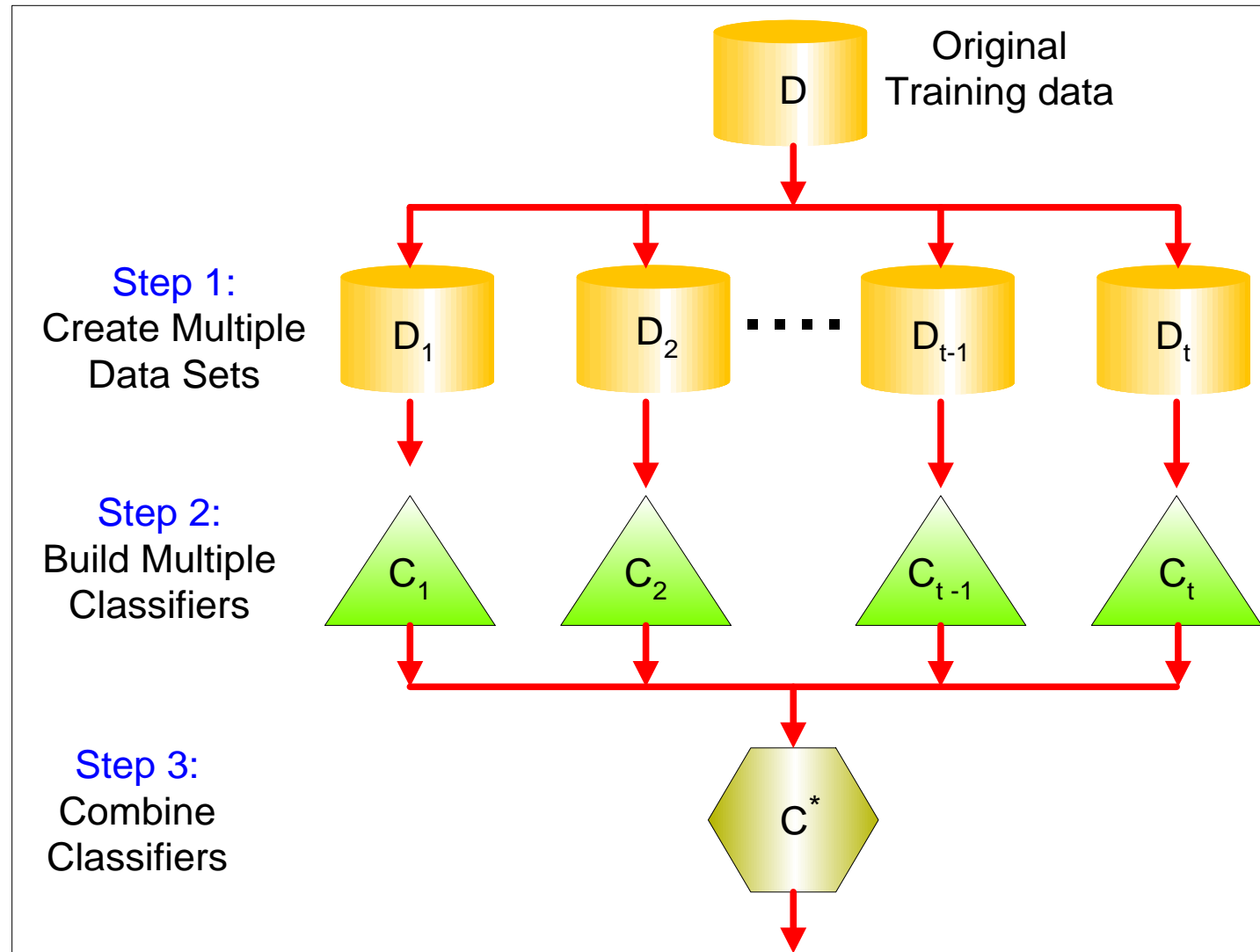


Ensemble Learning


















































Ensemble Learning

- So far we have focused on a single classifier.
- **Ensemble learning** → select a collection (ensemble) of classifiers and combine their predictions.
- Example : generate 100 different decision trees from the same or different training set and have them vote on the best classification for a new example.
- Key motivation: reduce the error rate. Hope is that it will become much more unlikely that the ensemble of classifiers will misclassify an example.

General Idea



Example: Weather Forecast

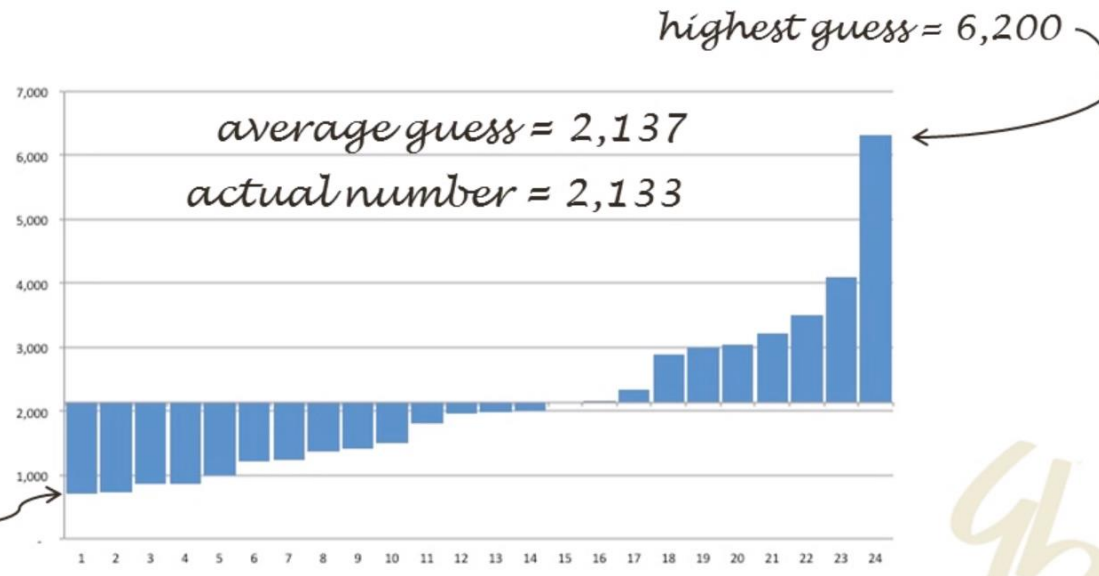
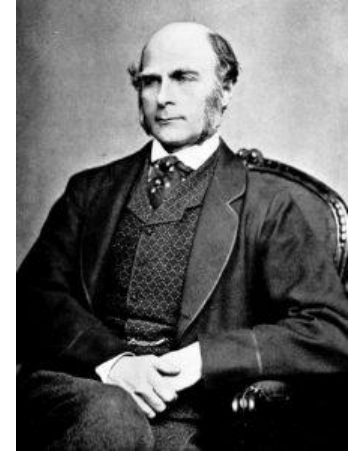
Reality							
1							
2							
3							
4							
5							
Combine							

Ensemble Learning

- When combining multiple classifiers
 - 1) independent classifiers(**diversity**)
 - 2) each classifier is more accurate than random guessing(**intelligence**)
- ***Diversity is more important than Intelligence!***
 - **Independent** random errors **cancel each other out**.
 - **Homogeneous** classifiers actually **amplify errors**, degrading the performance
- Diversity is the key in ensemble learning: It is not limited in machine learning. It is true in every group decision making in human as well

Wisdom of Crowds(Diversity)

- 1906 West of England Fat Stock and Poultry Exhibition experiment by Francis Galton
- 787 people guessed the weight of a steer.
- Their average guess of that weight was 1197 pounds. The actual weight of the steer was 1198 pounds.
- M&M jar estimation



Why Does It Work?

- Suppose there are 25 base classifiers
 - Each classifier has error rate, $\varepsilon = 0.35$
 - Assume classifiers are independent
 - Majority vote
 - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1 - \varepsilon)^{25-i} = 0.06$$

- Examples: Human ensembles are demonstrably better
 - How many jelly beans in the jar?
 - Who Wants to be a Millionaire: Audience vote.
- Remember: ensemble with similar classifiers is worse than single classifier.

Intuitions

- Majority vote
- Suppose we have 5 completely independent classifiers...
 - If accuracy is 70% for each
 - ◆ $(0.7^5) + 5(0.7^4)(0.3) + 10(0.7^3)(0.3^2)$
 - ◆ **83.7% majority vote accuracy**
 - 101 such classifiers
 - ◆ **99.9% majority vote accuracy**
- Note: Binomial Distribution: The probability of observing x heads in a sample of n independent coin tosses, where in each toss the probability of heads is p , is

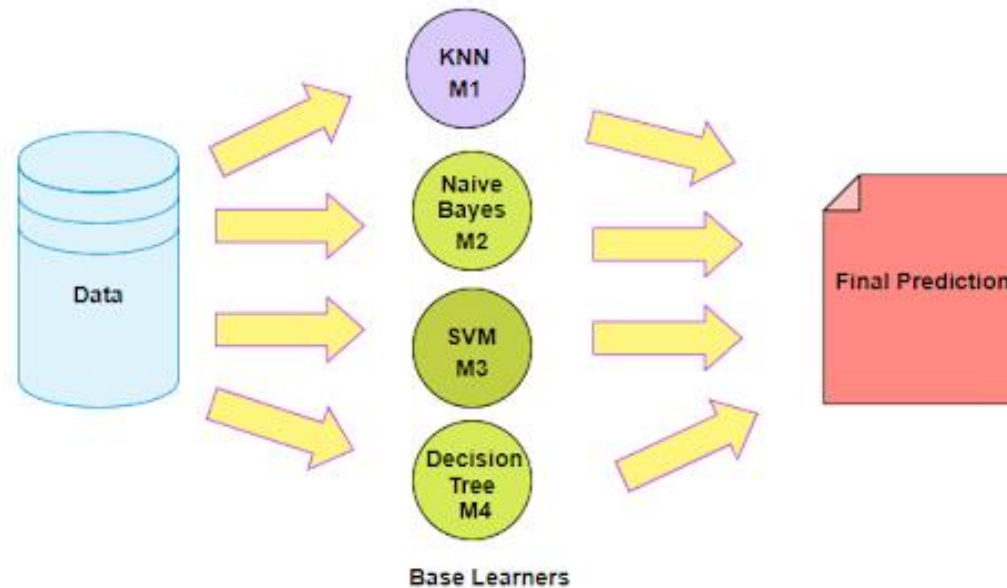
$$P(X = x|p, n) = \frac{n!}{x!(n-x)!} p^x (1 - p)^{n-x}$$

Learning Ensembles

- Two different ways for constructing ensembles
 - 1) using different training data or
 - 2) different learning algorithms.
- Combine decisions of multiple classifiers, e.g. using majority voting or weighted voting.

Different Learners

- Different learning algorithms
- Algorithms with different choice for parameters

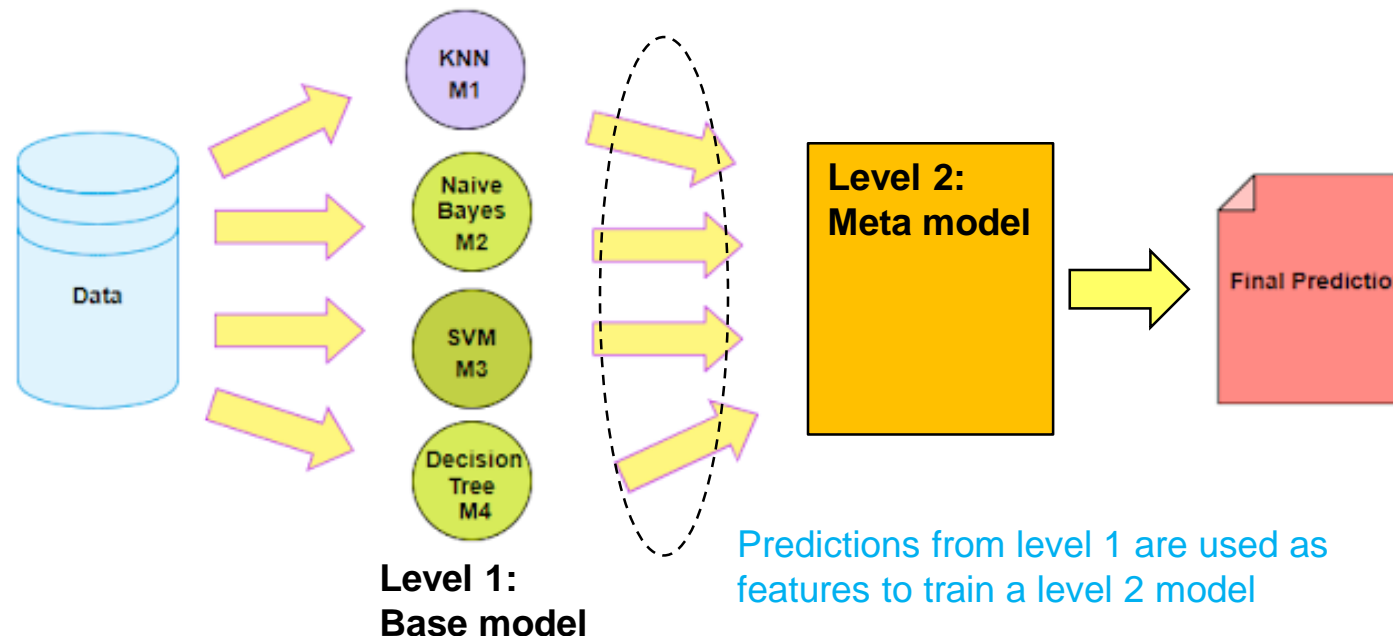


Homogenous Ensembles: Different Training Data

- Use a single, arbitrary learning algorithm but manipulate training data to make it learn multiple models.
 - $\text{Data1} \neq \text{Data2} \neq \dots \neq \text{Data } m$
 - $\text{Learner1} = \text{Learner2} = \dots = \text{Learner } m$
- Different methods for changing training data:
 - Bagging: Resample training data
 - Boosting: Reweight training data

Stacking Ensemble Methods

- “Stacking”: Use predictions of multiple models as “features” to train a new model
- Variation of Different Learners method in ensemble learning
- Base-Models (Level-0 Models): Models that fit the training data and predict out-of-sample data.
- Meta-Model (Level-1 Model): Model that fits on the prediction from base-models and learns how to best combine the predictions.



Stacking Ensemble Methods

- The meta-model is often a simple model such as Linear Regression for regression and Logistic Regression for classification.
- The base model's predictions are usually strongly correlated, as they are all trying to predict the same relationship.
- One reason why more complex meta-models are often not chosen is because there is a much higher chance that the meta-model may overfit to the predictions from the base models.
- For some applications, Ridge Regression works much better than Linear Regression.
- Ridge Regression comes with regularization parameters and hence is able to deal with the correlation between each base model's predictions much better than Linear Regression.

Stacking Ensemble Methods

Advantages:

- A more robust predictive performance than just regular individual models or average ensembles.

Disadvantages:

- Added complexity; that is, the final model becomes much harder to explain.

Note:

- The improvement stacking together models is only the most effective while using none or low correlated base models.
- An ensemble of diverse models means more diversity for the stacking model to optimize and reach better performance.

Examples of Ensemble Methods

- How to generate an ensemble of classifiers?
- Basic question is how to generate an ensemble of classifiers with high independency/diversity
- We cover the following methods in this chapter
 - Bagging
 - Boosting (Gradient Boosting)
 - Random Forests

Bagging

Bagging – Bootstrap Aggregating

- Given a set S of samples
- Generate a bootstrap sample T from S . Cases in S may not appear in T or may appear more than once (sample with replacement)
- Repeat this sampling procedure, getting a sequence of M independent training sets
- A corresponding sequence of classifiers $C_1 .. C_M$ is constructed for each of these training sets, by using the same classification algorithm
- To classify an unknown sample X , let each classifier predict or vote
- The Bagged Classifier C^* counts the votes and assigns X to the class with the “most” votes

Bagging

- Given a standard training set S of size n
- For $i = 1 \dots M$
 - Draw a sample of size $n^* < n$ from S uniformly and with replacement
 - Learn classifier C_i
- Each classifier $C_1 \dots C_M$ make predictions
- Final prediction C^* is the most votes from $C_1 \dots C_M$

Bagging Reduces Variance

- Remember the Bias / Variance decomposition:

$$\underbrace{E_{\mathbf{x},y,D} \left[(h_D(\mathbf{x}) - y)^2 \right]}_{\text{Expected Test Error}} = \underbrace{E_{\mathbf{x},D} \left[(h_D(\mathbf{x}) - \bar{h}(\mathbf{x}))^2 \right]}_{\text{Variance}} + \underbrace{E_{\mathbf{x},y} \left[(\bar{y}(\mathbf{x}) - y)^2 \right]}_{\text{Noise}} + \underbrace{E_{\mathbf{x}} \left[(\bar{h}(\mathbf{x}) - \bar{y}(\mathbf{x}))^2 \right]}_{\text{Bias}^2}$$

- Our goal is to reduce the variance term: $E_{\mathbf{x},D} \left[(h_D(\mathbf{x}) - \bar{h}(\mathbf{x}))^2 \right]$.
- For this, we want $h_D \rightarrow \bar{h}$
- The law of large numbers says (roughly) for i.i.d. random variables x_i with mean \bar{x} , we have,

$$\frac{1}{m} \sum_{i=1}^m x_i \rightarrow \bar{x} \text{ as } m \rightarrow \infty$$

- Apply this to classifiers: Assume we have m training sets D_1, D_2, \dots, D_m
- Train a classifier on each one and average result:

$$\hat{h} = \frac{1}{m} \sum_{i=1}^m h_{D_i} \rightarrow \bar{h} \quad \text{as } m \rightarrow \infty$$

Advantages of Bagging

- Easy to implement
 - Reduces variance, so has a strong beneficial effect on high variance classifiers.
 - As the prediction is an average of many classifiers, you obtain a mean score *and variance*.
 - Latter can be interpreted as the uncertainty of the prediction.
 - For example, imagine the prediction of a house price is \$300,000. If a buyer wants to decide how much to offer, it would be very valuable to know if this prediction has standard deviation $\pm \$10,000$ or $\pm \$50,000$.
-
- Bagging provides an unbiased estimate of the test error, which we refer to as the *out-of-bag error*. The idea is that each training point was not picked and all the data sets .
 - If we average the classifiers of all such data sets, we obtain a classifier (with a slightly smaller) that was not trained on ever and it is therefore equivalent to a test sample.
 - If we compute the error of all these classifiers, we obtain an estimate of the true test error.
 - The beauty is that we can do this without reducing the training set. We just run bagging as it is intended and obtain this so called out-of-bag error for free.

Bagging

- Sampling with replacement

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bootstrap sample
- Each sample has probability $1 - (1 - 1/n)^n$ of being selected (63% if n is infinite) : $(1 - 1/e)$

* **Decision Stump**: single level decision tree

If $X_1 \leq v_{11}$ [&& $X_2 \leq v_{21}$...] then c_1 else c_2

Illustrating Bagging

Original data

X	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Y	1	1	1	-1	-1	-1	-1	1	1	1

Bagging Round

Round 1

X	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
Y	1	1	1	1	-1	-1	-1	-1	1	1

Decision stump

$X \leq 0.35 \rightarrow y=1$
 $X > 0.35 \rightarrow y=-1$

Round 2

X	0.1	0.2	0.3	0.4	0.5	0.8	0.9	1	1	1
Y	1	1	1	-1	-1	1	1	1	1	1

$X \leq 0.05 \rightarrow y=-1$
 $X > 0.05 \rightarrow y=1$

Round 3

X	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9
Y	1	1	1	-1	-1	-1	-1	-1	1	1

$X \leq 0.35 \rightarrow y=1$
 $X > 0.35 \rightarrow y=-1$

Round 4

X	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9
Y	1	1	1	-1	-1	-1	-1	-1	1	1

$X \leq 0.3 \rightarrow y=1$
 $X > 0.3 \rightarrow y=-1$

Illustrating Bagging

Decision stump

Round 5

X	0.1	0.1	0.2	0.5	0.6	0.6	0.6	1	1	1
Y	1	1	1	-1	-1	-1	-1	1	1	1

$X \leq 0.35 \rightarrow y=1$
 $X > 0.35 \rightarrow y=-1$

Round 6

X	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1
Y	1	-1	-1	-1	-1	-1	-1	1	1	1

$X \leq 0.75 \rightarrow y=-1$
 $X > 0.75 \rightarrow y=1$

Round 7

X	0.1	0.4	0.4	0.6	0.7	0.8	0.9	0.9	0.9	1
Y	1	-1	-1	-1	-1	1	1	1	1	1

$X \leq 0.75 \rightarrow y=-1$
 $X > 0.75 \rightarrow y=1$

Round 8

X	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1
Y	1	-1	-1	-1	-1	-1	-1	1	1	1

$X \leq 0.75 \rightarrow y=-1$
 $X > 0.75 \rightarrow y=1$

Round 9

X	0.1	0.3	0.4	0.4	0.6	0.7	0.7	0.8	1	1
Y	1	-1	-1	-1	-1	-1	-1	1	1	1

$X \leq 0.75 \rightarrow y=-1$
 $X > 0.75 \rightarrow y=1$

Round 10

X	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9
Y	1	1	1	1	1	1	1	1	1	1

$X \leq 0.05 \rightarrow y=1$
 $X > 0.05 \rightarrow y=-1$

Illustrating Bagging

Round	X=0.1	X=0.2	X=0.3	X=0.4	X=0.5	X=0.6	X=0.7	X=0.8	X=0.9	X=1.0
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
Sum	2	2	2	-6	-6	-6	-6	2	2	2
Sign	1	1	1	-1	-1	-1	-1	1	1	1
True Class	1	1	1	-1	-1	-1	-1	1	1	1