Chapter 7

Ensemble Classifiers

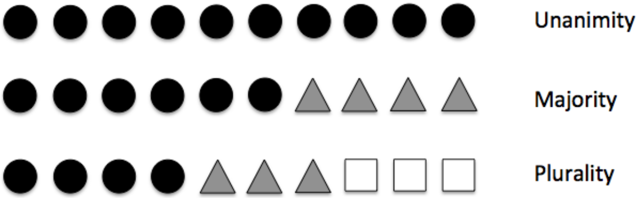
August 11, 2017

Learning with ensembles

- Our goal is to combined multiple classifiers
- Mixture of experts, e.g. 10 experts
- Predictions more accurate and robust
- Provide an intuition why this might work
- Simplest approach: majority voting

Majority voting

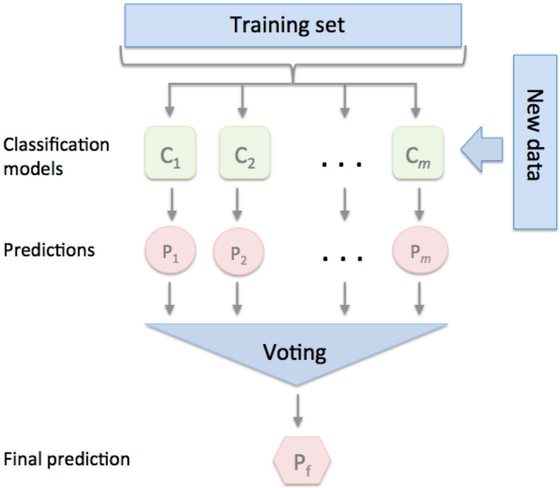
- Majority voting refers to binary setting
- Can easily generalize to multi-class: plurality voting
- Select class label that receives the most votes (mode)



Combining predictions: options

- Train m classifiers $C_1, \ldots, C_m$
- Build ensemble using different classification algorithms (e.g. SVM, logistic regression, etc.)
- Use the same algorithm but fit different subsets of the training set (e.g. random forest)

General approach



Combining predictions via majority voting

We have predictions of individual classifiers C_j and need to select the final class label $\hat{y}$

$$\hat{y} = mode\{C_1(\mathbf{x}), C_2(\mathbf{x}), \dots, C_m(\mathbf{x})\}$$

For example, in a binary classification task where $class_1 = -1$ and $class_2 = +1$, we can write the majority vote prediction as follows:

$$C(\mathbf{x}) = sign\left[\sum_{j}^{m} C_j(\mathbf{x})\right] = \begin{cases} 1 & \text{if } \sum_j C_j(\mathbf{x}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Intuition why ensembles can work better

Assume that all n base classifiers have the same error rate ϵ. We can expresss the probability of an error of an ensemble can be expressed as a probability mass function of a binomial distribution:

$$P(y \geq k) = \sum_{k}^{n} \binom{n}{k} \epsilon^k (1 - \epsilon)^{n-k} = \epsilon_{\text{ensemble}}$$

Here, $\binom{n}{k}$ is the binomial coefficient *n choose k*. In other words, we compute the probability that the prediction of the ensemble is wrong.

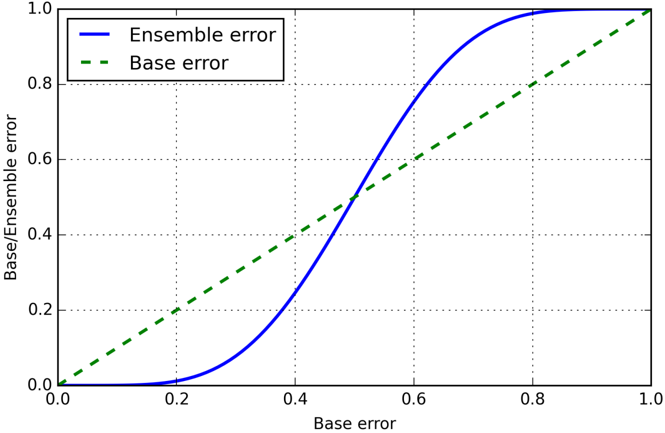
Example

Imagine we have 11 base classifiers ($n = 11$) with an error rate of 0.25 ($\epsilon = 0.25$):

$$P(y \geq k) = \sum_{k=6}^{11} \binom{11}{k} 0.25^k (1 - 0.25)^{11-k} = 0.034$$

So the error rate of the ensemble of $n = 11$ classifiers is much lower than the error rate of the individual classifiers.

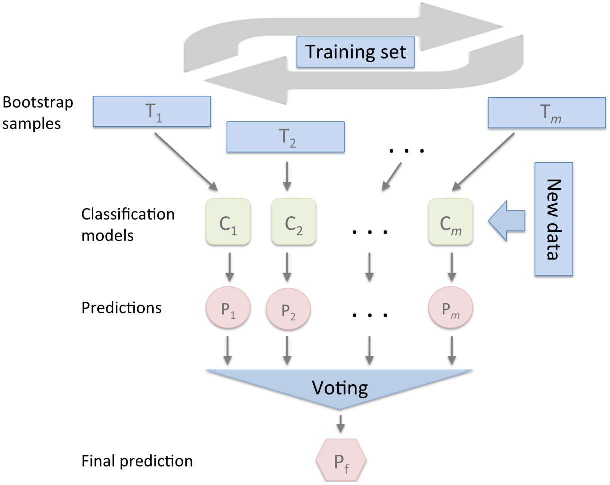
Same reasoning applied to a wider range of error rates



Boostrap aggregation (bagging)

- We used the entire training set for the majority vote classifier
- Here we draw **bootstrap samples**
- In statistics, **bootstrapping** is any test or metric that relies on **random sampling with replacement**.
- Hypothesis testing: bootstrapping often used as an alternative to statistical inference based on the assumption of a parametric model when that assumption is in doubt
- The basic idea of bootstrapping is that inference about a population from sample data, can be modelled by resampling with replacement the sample data and performing inference about a sample from resampled data.

Bagging


Training set
Bootstrap samples
T_1
T_2
T_m
Classification models
C_1
C_2
C_m
New data
Predictions
P_1
P_2
P_m
Voting
Final prediction
P_f

Boostrapping example

Sample indices	Bagging round 1	Bagging round 2	...
1	2	7	...
2	2	3	...
3	1	2	...
4	3	1	...
5	7	1	...
6	2	7	...
7	4	7	...

C_1 C_2 C_m

- Seven training examples
- Sample randomly with replacement
- Use each boostrap sample to train a classifier C_j
- C_j is typically a decision tree
- **Random Forests**: also use random feature subsets

Bagging in scikit-learn

- Instantiate a decision tree classifier
- Make a bagging classifier with decision trees
- Check that the accuracy is higher for the bagging classifier
- PML github