GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
LECTURE :ENSEMBLE METHODS

KEY CONCEPTS - ENSEMBLE METHODS

- Up until now we have been looking at single learning algorithms
- Train a model using a training set, optimize using a validation set, publish results on the test set, deploy…
- It's a single model

Ensembles methods represent an effective alternative since they tend to achieve better predictive accuracy by combining or chaining the results from different data samples, algorithm settings and types

KEY CONCEPTS - ENSEMBLE METHODS

Two Categories:

1. Averaging algorithms or Committees
    - predict by averaging the results of various parallel estimators
    - bagging, pasting, subspaces and patches
2. Boosting algorithms
    - predict by using a weighted average of sequential aggregated estimators

KEY CONCEPTS - ENSEMBLE METHODS

From the Sklearn documentation:

- Bagging - If samples are drawn with replacement
- Pasting - When random subsets of the dataset are drawn as random subsets of the samples
- Subspaces - When random subsets of the dataset are drawn as random subsets of the features
- Patches - subsets of both samples and features

AVERAGING ALGORITHMS OR COMMITTEES

KEY CONCEPTS - COMMITTEES

- The simplest way to construct a committee is to average the predictions of a set of individual models
- If you average, for example, a set of low bias models for predicting a sinusoid, then the result will be to average out any over-fitting. The over-fitting on each individual model is unique to that model, so combining models tends to cancel this out
- By reducing variance you obtain a more accurate model
- The problem is that often you only have a single training set - so how do you produce multiple models?

KEY CONCEPTS - COMMITTEES

- Bootstrapping is one solution
- Bootstrapping is the process of producing multiple data sets from a initial single data source
- If X is our initial data set containing M points, then we randomly draw M samples from X with replacement (after each point is drawn), to create a new data set X_B
- This process can be repeated, say L times, to generate L data sets
- Each dataset will, therefore, often contain duplicated points
- Each dataset will be of size M

KEY CONCEPTS - COMMITTEES - BAGGING

- After producing M bootstrapped data sets you train M predictive models
- For a new data point you classify/regress by taking the average of the output of all the models run against the new data point
- This is called ‘bootstrap aggregation’ or Bagging
- Bagging will reduce the overall error of your prediction (cf with that of a single model), but the reduction may be small

KEY CONCEPTS - ENSEMBLE METHODS - BAGGING

- Bagging with weak ensembles
- sklearn has a BaggingClassifier and a BaggingRegressor
- You decide on the algorithm and then plug it into the BaggingClassifier/ Regressor call
- You set a high number of estimators
- Good choices for the estimator are ones that train quickly
- Combined together they can do well, and may out-perform a more sophisticated single algorithm

KEY CONCEPTS - ENSEMBLE METHODS - PASTING

- Pasting is, again, building estimators by using a large number of small datasets generated from the original data
- Sampling without replacement (cf Bootstrapping)
- Useful when dealing with large datasets
- Leo Breiman (creator of the random forest algorithm)
- No specific sklearn routines - but easy enough to program

KEY CONCEPTS - ENSEMBLE METHODS - SUBSPACES AND PATCHES

- Subspaces refers to using subsets of feature space in order to build a model
- Patches refers to using both subsets of feature space and subsets of the training data in order to build a model
- This leads us to introduce the classification tree
- Examples include:
  - RandomForestClassifier, RandomForestRegressor
  - ExtraTreeClassifier, ExtraTreeRegressor

Key Concepts - Ensemble Methods - Partitioning Feature Space

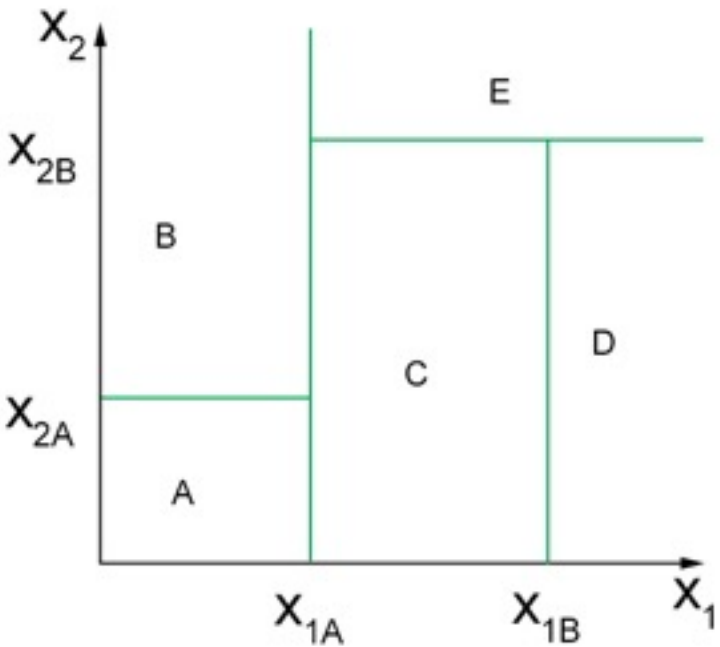
- There are various simple, but widely used models that work by partitioning the feature space into square/cuboid regions
- A simple model is then assigned to each region
- This can be viewed as a model combination method, where a single model is responsible for making predictions at any given point in input space

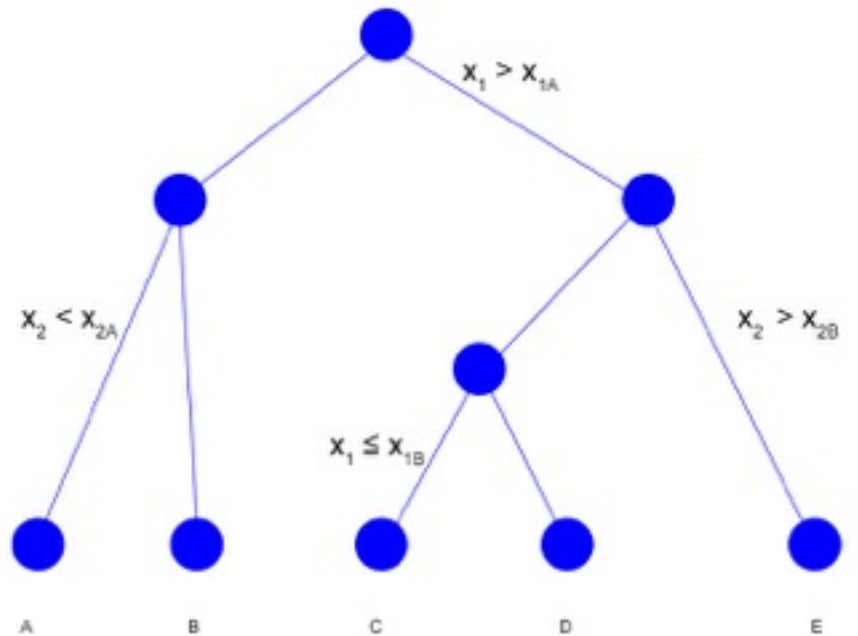
KEY CONCEPTS - ENSEMBLE METHODS - PARTITIONING FEATURE SPACE

- Given a new input, x, the process of selecting a model can be described by a sequential decision making process corresponding to the traversal of a binary tree

- Here we will focus on
  - CART - Classification and Regression Trees (Breiman)
  - ID3 - Iterative Dichotomiser 3, and C4.5 (Quinlan)

KEY CONCEPTS - DECISION TREES - PARTITIONING FEATURE SPACE





KEY CONCEPTS - DECISION TREES

- Within each region there is a separate model to predict the target variable
  - for regression we might assign each region to a number
  - for classification we would associate each region with a class

- Each region equates to the leaf of the tree

KEY CONCEPTS - DECISION TREES

- Key property of decision trees
  - The sequence of binary decisions can be interpretable by humans
  - In medical diagnosis, for example, you might have the following to predict a patient's disease:
    - is the temperature above 98 degree?
    - if 'yes', then is the blood pressure lower than 100 on 70?
    - etc…with the leaf being a specific diagnosis

KEY CONCEPTS - DECISION TREES

- The problem of developing classification trees lies in determining their structure
  - which input variable is chosen at each node to form the split criterion
  - what is the threshold value for that variable
  - assigning class/value to leaf-nodes

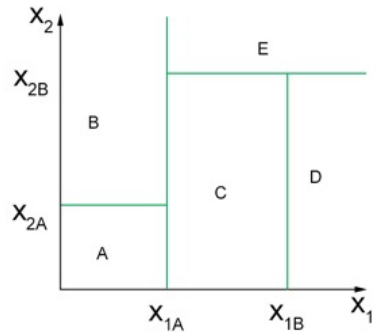
- In terms of regression the optimal tree is the one having the smallest residual sum-of-squared error

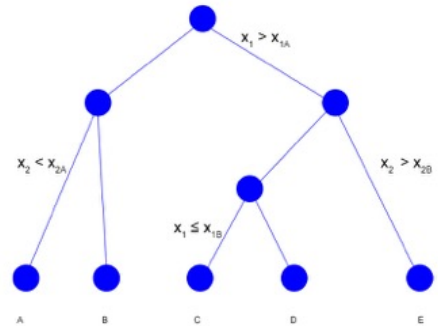
- The problem is that to minimize the sum-of-squared error is usually computationally impossible given the number of possible tree structures
- Instead we use a ‘greedy’ optimization
- Start at the root node and grow the tree one node at a time
- At each node there is the choice of input variables and the value of the threshold, remembering each split results in a set of (possibly temporary) leaf nodes

KEY CONCEPTS - DECISION TREES

- At each level we can do an exhaustive search of the feature space to minimize a residual sum-of-squared error
- To determine when to stop growing the tree is more difficult
- The best strategy is to grow a ‘large’ tree based upon the number of data points associated with the leaves, then prune it back
- Pruning is based upon a criterion that balances residual sum-of-squared error and ‘model’ complexity

KEY CONCEPTS - DECISION TREES





Denote the starting tree for pruning by T_0

Define $T \subset T_0$ to be a subtree

Let the leaf nodes be indexed by $\tau = 1, \ldots, |T|$

There are T leaf nodes

Leaf node τ represents a region of input space R_τ having N_τ data points

KEY CONCEPTS - DECISION TREES

For Regression the optimal prediction for region R_τ is given by:

$$y_\tau = \frac{1}{N_\tau} \sum_{x_n \in R_\tau} t_n$$

i.e. it is just the average of all the target values within that region of feature space

Residual Sum-of-Squares is:

$$Q_\tau(T) = \sum_{x_n \in R_\tau} (t_n - y_\tau)^2$$

KEY CONCEPTS - DECISION TREES

The pruning criterion is then given by:

$$C(T) = \sum_{r=1}^{|T|} Q_r(T) + \lambda |T|$$

λ is the regularization parameter that determines the trade-off between the overall residual sum-of-squares error and the complexity of the model as measured by the number of $|T|$ leaf nodes

As always choose λ by cross validation

KEY CONCEPTS - DECISION TREES

For classification

The process is similar but the sum-of-squares is replaced by a more appropriate measure of perfomance

if $p_{\tau k}$ is the proportion of data points in region R_τ assigned to class k, where $k = 1, \ldots, K$, then there are 2 commonly used choices:

1. Negative cross-entropy

$$Q_\tau(T) = \sum_{k=1}^{K} p_{\tau k} \log p_{\tau k}$$

and

2. Gini index

$$Q_\tau(T) = \sum_{k=1}^{K} p_{\tau k} (1 - p_{\tau k})$$

KEY CONCEPTS - DECISION TREES

- These encourage the formation of regions in which a high proportion of data points belong to a single class
- They are better measures than classification accuracy as they are more sensitive to the node probabilities
- They are also differentiable and thus more suited for gradient descent based optimization

SIDEBAR - ENTROPY

KEY CONCEPTS - DECISION TREES - ENTROPY

- The determination of what features to use in order to make a node with a decision tree requires the measurement of information content

- We want to split using a feature that contains as much information content on how to separate our classes as possible

- The ideal would be a feature that completely separates a dataset containing 2 classes into the 2 classes!

- Suppose we have a variable x, how much information does x contain? or how much information is received when we find out the value of x?
- Consider information content as the ‘degree of surprise’ on finding out about the value of the variable x
- If you are told about that an event that has happened is very unlikely then the message contains more information than if you are told that the event was very likely
  - sunrise vs eclipse of the sun

- Information content is therefore, related to the probability distribution of x
- Ideally it would be helpful to express information as a monotonic function of a variable's probability distribution, i.e. so we have h(x)

$$h(x) = -log_2[p(x)]$$

- The negative sign ensure information is 0, or +ve
- If p(x) is low, then information content is high

KEY CONCEPTS - DECISION TREES - ENTROPY

```
x = 0.00010, h(x) = 13.29
x = 0.11119, h(x) =  3.17
x = 0.22228, h(x) =  2.17
x = 0.33337, h(x) =  1.58
x = 0.44446, h(x) =  1.17
x = 0.55554, h(x) =  0.85
x = 0.66663, h(x) =  0.59
x = 0.77772, h(x) =  0.36
x = 0.88881, h(x) =  0.17
x = 0.99990, h(x) =  0.00
```

- Now, if a sender wishes to transmit the value of x to a receiver, what is the average amount of information they will transmit?
- This is calculated by taking the expectation of h(x)
- Expectation is just a probabilistically weighted average

$$H(x) = -\sum_{x} p(x) log_2[p(x)]$$

- H(x) is called the Entropy of the variable x

KEY CONCEPTS - ENTROPY - EXAMPLE

- Suppose x has 8 possible values, all of which are equally likely
- The entropy will determine the information content and convey the number of bits required to send x
- p(x) = 1/8 (the probability of each value)
- $H(x) = -8 * 1/8 * \log_2[1/8] = 3$ bits!!
- H has units called bits!!
- Claude Shannon, 1948 (American mathematician, 'Father of Information Theory')

KEY CONCEPTS - DECISION TREES

- In classification tasks the leaf nodes of the decision tree represent the classes
- In regression tasks the leaf nodes of the decision tree may be averaged to produce the estimate for the response variable
- Once a decision tree has been trained (i.e. constructed) making a prediction for a test instance requires only following a path from root to leaf, based upon the test instance feature values

KEY CONCEPTS - DECISION TREES

- In practice the human interpretability of a tree model is not such a strength as you would imagine.

- A particular tree structure is very sensitive to the details of the data set, such that even a small change in the training set can result in very different splits

- The complexity and size of resulting trees can make interpretation difficult

KEY CONCEPTS - DECISION TREES

- Other problems include the fact that splitting is axis aligned, whereas this may not be the optimal way to construct a decision boundary

- In addition the splits are hard, so that each region of input space is associated with one, and only one, leaf node. This is a particular problem in regression where we are typically aiming for smooth functions

KEY CONCEPTS - DECISION TREES

- Sklearn uses an optimized version of the CART algorithm
- Critical hyper-parameters:
  - max_features: The number of sampled features that are present at every split. The lower the number, the higher the bias
  - min_samples_leaf: This will affect the depth of the tree. Large numbers increase the bias of the model (and therefore decrease the variance). Large numbers mean trees of smaller depth
  - bootstrap: This boolean parameter allows for bootstrapping

KEY CONCEPTS - DECISION TREES

- RandomForests and ExtraTrees are parallel algorithms
  - n_jobs can be set to maximize multi-processing
- Classification is done by using the most voted class (i.e. a majority vote)
- Regression is done by using an average of the values

KEY CONCEPTS - ENSEMBLE METHODS - DECISION TREES

- These algorithms also utilize the ensemble, i.e training multiple trees
- The final prediction is an average of the output of the ensemble, which, as we have already seen increases the accuracy of the prediction, and helps to control variance
- n_estimators sets the number of trees to train

KEY CONCEPTS - DECISION TREES - EXAMPLE

Table 4.1. The vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

KEY CONCEPTS - DECISION TREES - EXAMPLE

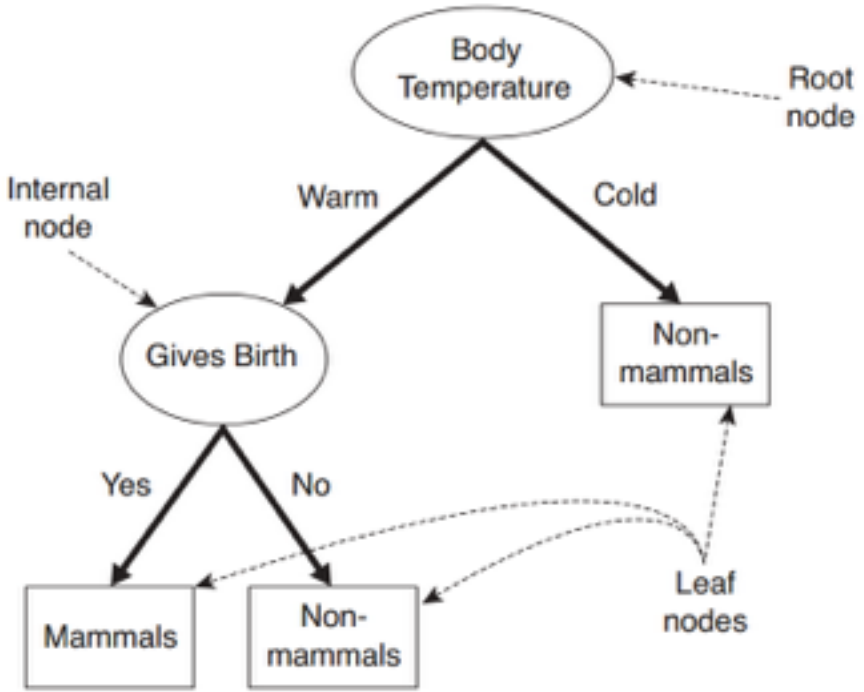


Figure 4.4. A decision tree for the mammal classification problem.

BOOSTING

Boosting is a powerful technique for combining ‘base’ classifiers to produce a form of ‘committee’ whose performance can be significantly better than any of the base classifiers themselves

- Adaptive Boosting or Ada Boost is probably the most widely used boosting algorithm
- Boosting can give good results even if the base classifiers are only slightly better than random!
- The base classifiers are sometimes called ‘weak learners’

KEY CONCEPTS - ENSEMBLE METHODS - BOOSTING

- The principal difference between boosting and other averaging algorithms (such as bagging) is that the base classifiers are trained in sequence and each classifier is trained using a weighted form of the data set in which the weighting coefficient associated with each data point depends upon the performance of the previous classifier

- Points that are misclassified by one base classifier and more heavily weighted when the subsequent classifier is trained

KEY CONCEPTS - ENSEMBLE METHODS - BOOSTING

Once all the classifiers have been trained classification occurs by using a combined weighted majority voting scheme

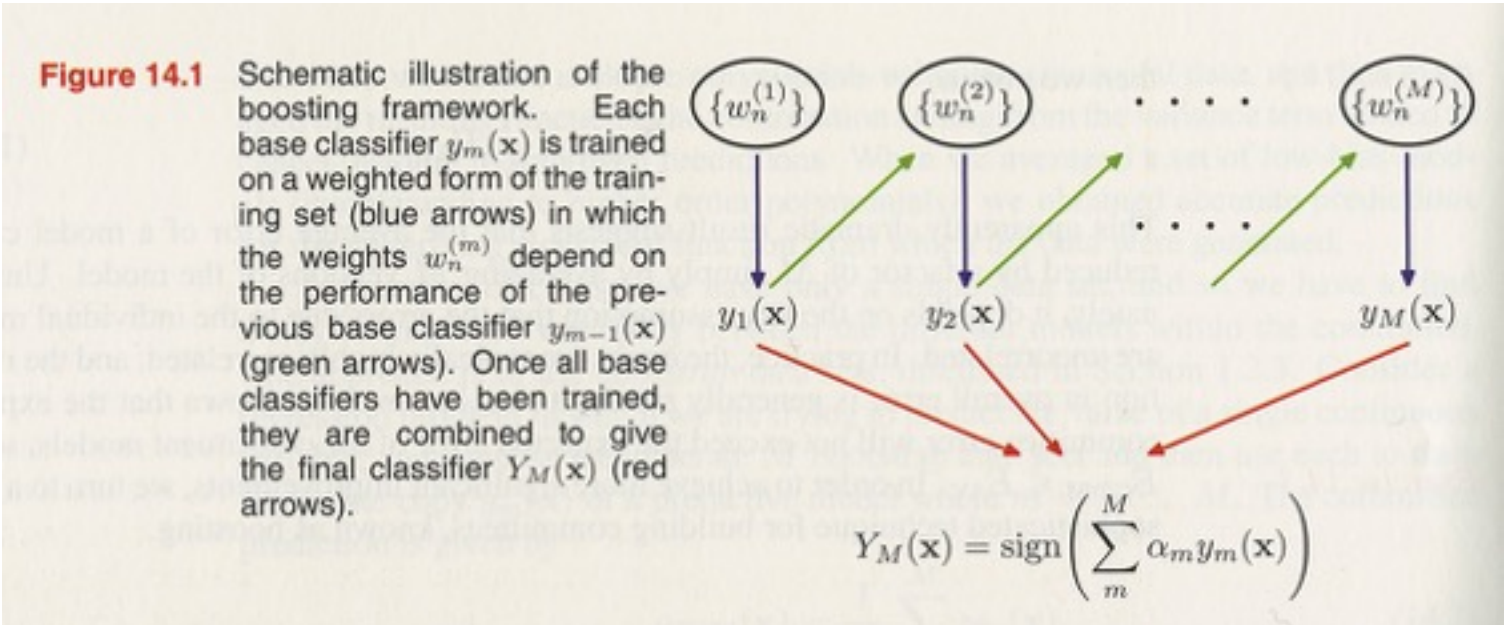


Figure 14.1 Schematic illustration of the boosting framework. Each base classifier $y_m(\mathbf{x})$ is trained on a weighted form of the training set (blue arrows) in which the weights $w_n^{(m)}$ depend on the performance of the previous base classifier $y_{m-1}(\mathbf{x})$ (green arrows). Once all base classifiers have been trained, they are combined to give the final classifier $Y_M(\mathbf{x})$ (red arrows).

from Pattern Recognition and Machine Learning, Christopher M. Bishop

KEY CONCEPTS - BOOSTING ALGORITHMS - ADA BOOST

AdaBoost

- fits a sequence of weak learners, on re-weighted versions of the data
- initially the weighting coefficients are equal
- with each iteration the weighting coefficients are increased for those data points that are misclassified
- weighting goes unchanged for data points correctly classified
- each of the classifiers has its error recorded
- the final classification is made by weighting the classifiers themselves, such that greater weight is given to the more accurate classifiers

KEY CONCEPTS - BOOSTING ALGORITHMS - ADA BOOST

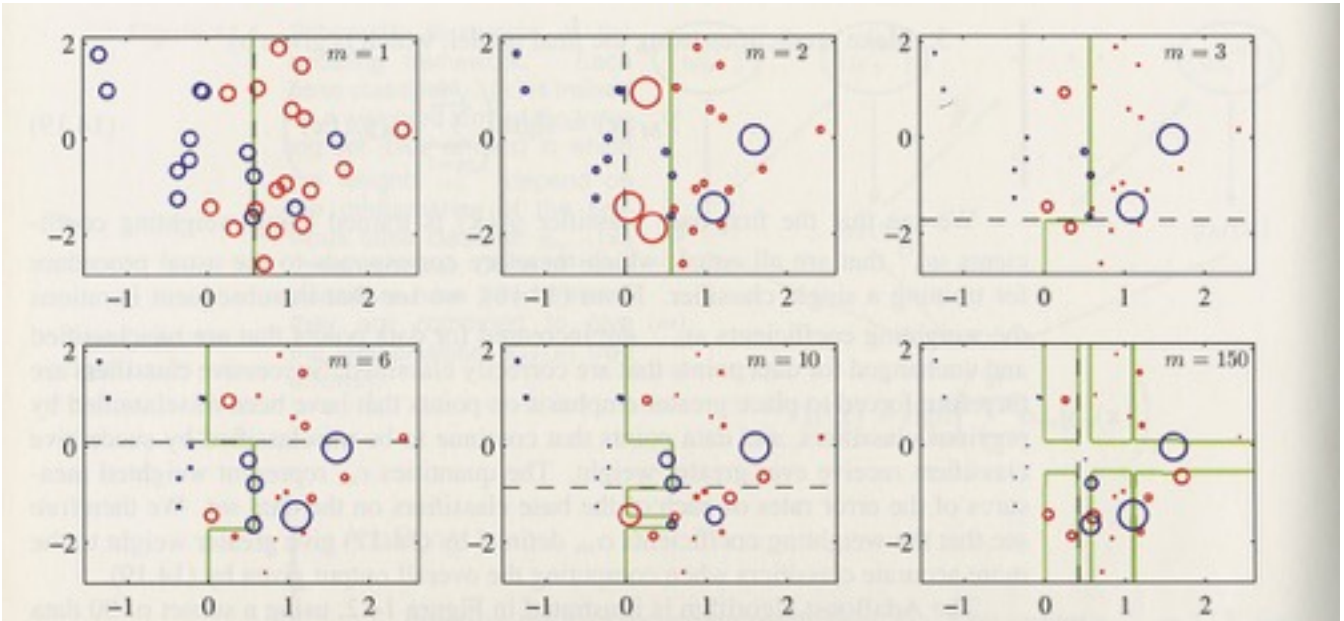


Figure 14.2 Illustration of boosting in which the base learners consist of simple thresholds applied to one or other of the axes. Each figure shows the number m of base learners trained so far, along with the decision boundary of the most recent base learner (dashed black line) and the combined decision boundary of the ensemble (solid green line). Each data point is depicted by a circle whose radius indicates the weight assigned to that data point when training the most recently added base learner. Thus, for instance, we see that points that are misclassified by the $m = 1$ base learner are given greater weight when training the $m = 2$ base learner.

from Pattern Recognition and Machine Learning, Christopher M. Bishop

KEY CONCEPTS - BOOSTING ALGORITHMS

Gradient Tree Boosting

Like AdaBoost it is based upon gradient descent

Characterised by major variance, more sensibility to noise in data.

Both these problems can be attenuated by using subsampling

Also characterised by increased computational cost due to non-parallel operations

GradientBoostClassifier(), GradientBoostRegression()

KEY CONCEPTS - BOOSTING ALGORITHMS

Hyper-parameters:

n_estimators: The more the better, but too few - too many affects bias - variance.

max_depth: of the tree. Trees of high depth increase variance and are more complex

subsample: can be used to reduce variance

learning_rate: standard gradient descent parameter

min_samples_leaf: again affects shape of the tree and the variance of the final model

SKLEARN ALGORITHM SUMMARY

KEY CONCEPTS - SKLEARN ALGORITHMS

- AdaBoost - classifier and regressor
- Bagging - classifier and regressor
- ExtraTrees - classifier and regressor
- GradientBoosting - classifier and regressor
- RandomForest - classifier and regressor

- sklearn also supports decision trees (no ensemble) for classification and regression