GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

LECTURE 11: ENSEMBLE TECHNIQUES

LAST TIME:

- DECISION TREES
- DECISION TREES IN SCIKIT-LEARN

QUESTIONS?

I. ENSEMBLE TECHNIQUES
II. PROBLEMS IN CLASSIFICATION
III. BAGGING
IV. BOOSTING
V. RANDOM FORESTS

EXAMPLE – DECISION TREE

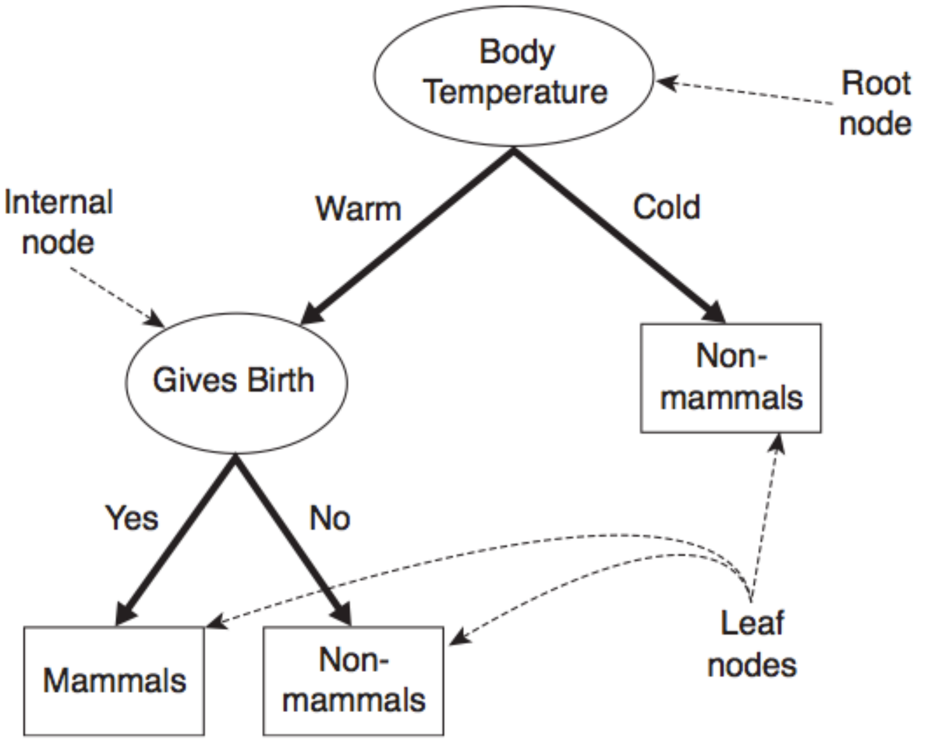


Figure 4.4. A decision tree for the mammal classification problem.

source: http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

NOTE

Internal nodes represent test conditions which partition the records at that node.

REVIEW: RANDOM FORESTS

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

1) *Select a sample of the original training set and build a tree as follows:*

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

1) *Select a sample of the original training set and build a tree as follows:*
    1) *Select m features out of the M available*
    2) *Pick the best split based on just those m variables*

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

1) *Select a sample of the original training set and build a tree as follows:*
    1) *Select m features out of the M available*
    2) *Pick the best split based on just those m variables*

2) *Repeat (1) for N iterations*

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

1) *Select a sample of the original training set and build a tree as follows:*
    1) *Select m features out of the M available*
    2) *Pick the best split based on just those m variables*

2) *Repeat (1) for N iterations*
3) *Predict based on majority vote of N trees*

I. ENSEMBLE TECHNIQUES

Q: What are ensemble techniques?

Q: What are ensemble techniques?

A: Methods of improving classification accuracy by aggregating predictions over several **base classifiers***.*

Q: What are ensemble techniques?

A: Methods of improving classification accuracy by aggregating predictions over several **base classifiers***.*

Ensembles are often much more accurate than the base classifiers that compose them.

There are two general ensemble techniques:

There are two general ensemble techniques:

Averaging: *build several models independently and then to average their predictions.*

Examples*: Random Forests and Bagging.*

There are two general ensemble techniques:

***Boosting**: models are built sequentially to combine several weak models to produce a powerful ensemble.*

***Examples**: Ada Boost and Gradient Tree Boosting*

In order for an ensemble classifier to outperform a single base classifier, the following conditions must be met:

In order for an ensemble classifier to outperform a single base classifier, the following conditions must be met:

1) *the base classifiers must be* **accurate***: they must outperform random guessing*

In order for an ensemble classifier to outperform a single base classifier, the following conditions must be met:

1) *the base classifiers must be* **accurate***: they must outperform random guessing*

2) *the base classifiers must be* **diverse***: their misclassifications must occur on different training examples*

II. PROBLEMS IN CLASSIFICATION

In any supervised learning task, our goal is to make predictions of the true classification function `f` *by learning the classifier* `h`.

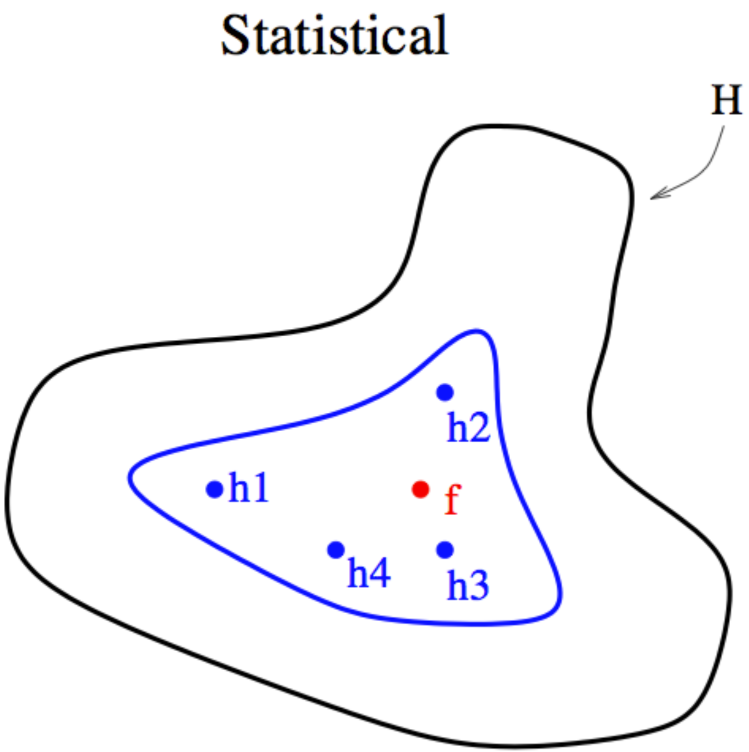
In any supervised learning task, our goal is to make predictions of the true classification function `f` *by learning the classifier* `h`.

There are three main problems that can prevent this:

- *statistical problem*
- *computational problem*
- *representational problem*

If the amount of training data available is small, the base classifier will have difficulty converging to h.

An ensemble classifier can mitigate this problem by “averaging out” base classifier predictions to improve convergence.



NOTE

The true function f is best approximated as an average of the base classifiers.

source: http://www.cs.iastate.edu/~jtian/cs573/Papers/Dietterich-ensemble-00.pdf

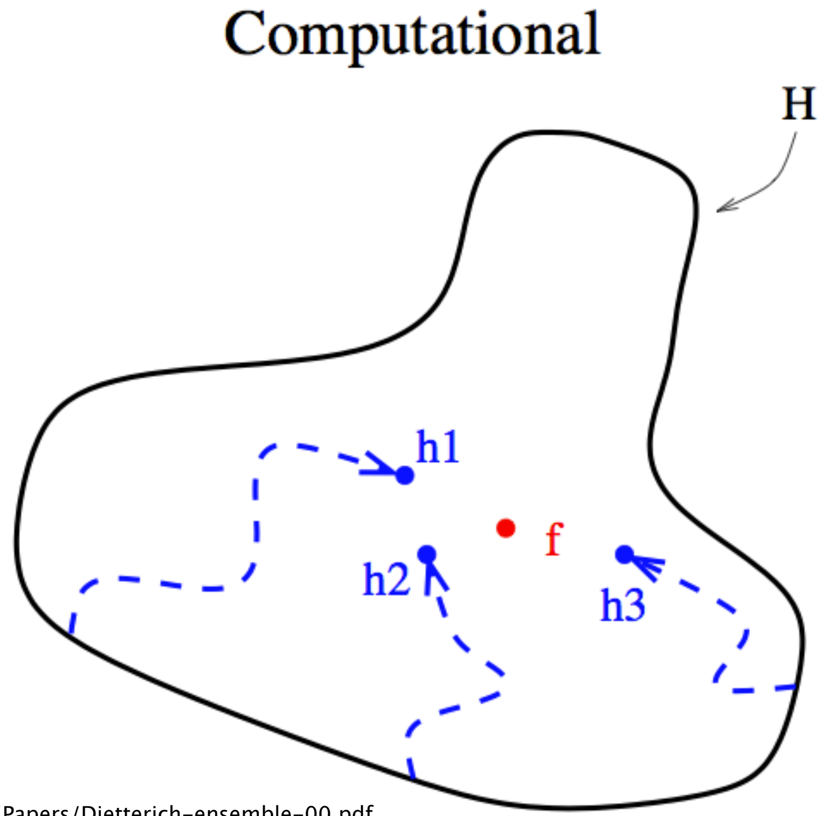
Even with sufficient training data, it may still be computationally difficult to find the best classifier h.

For example, if our base classifier is a decision tree, an exhaustive search of the hypothesis space of all possible classifiers is extremely complex (NP-complete).

Even with sufficient training data, it may still be computationally difficult to find the best classifier `h`.

For example, if our base classifier is a decision tree, an exhaustive search of the hypothesis space of all possible classifiers is extremely complex (NP-complete).

An ensemble composed of several BC's with different starting points can provide a better approximation to `f` *than any individual BC.*



NOTE

The true function f is often best approximated by using several starting points to explore the hypothesis space.

source: http://www.cs.iastate.edu/~jtian/cs573/Papers/Dietterich-ensemble-00.pdf

Sometimes f *cannot be expressed in terms of our hypothesis at all.*

Sometimes f *cannot be expressed in terms of our hypothesis at all.*

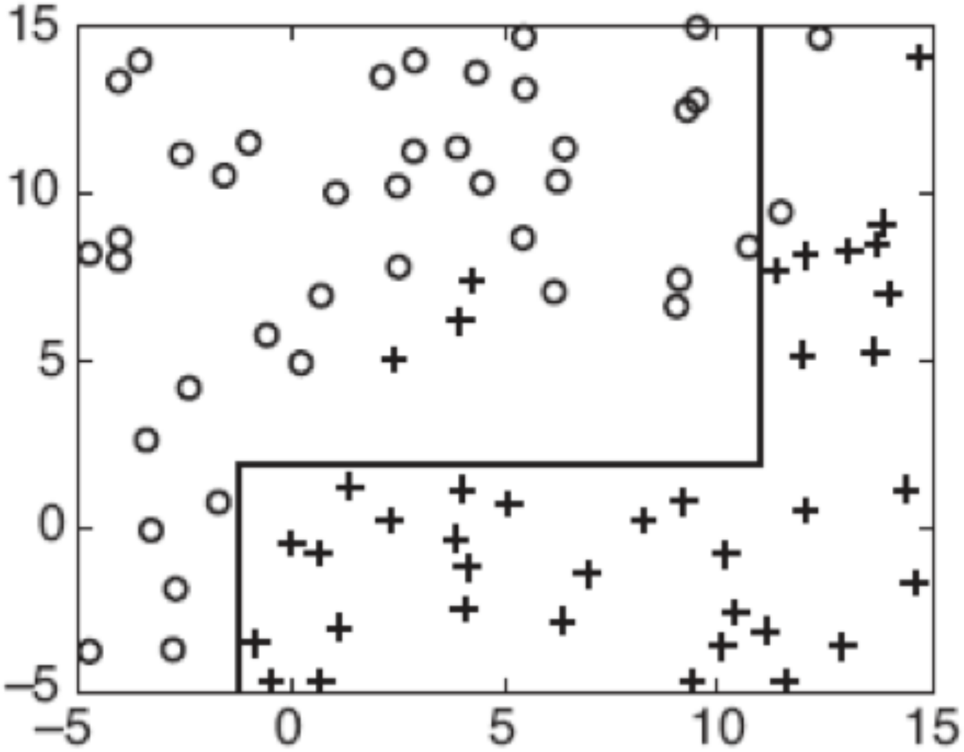
To illustrate this, suppose we use a decision tree as our base classifier.

Sometimes f *cannot be expressed in terms of our hypothesis at all.*

To illustrate this, suppose we use a decision tree as our base classifier.

A decision tree works by forming a rectilinear partition of the feature space.

THE REPRESENTATIONAL PROBLEM – 2D DECISION TREE



NOTE

What is a rectilinear decision boundary?

One whose segments are orthogonal to the x & y axes.

But what if `f` *is a diagonal line?*

But what if `f` is a diagonal line?

Then it cannot be represented by finitely many rectilinear segments, and therefore the true decision boundary cannot be obtained by a decision tree classifier.

But what if `f` *is a diagonal line?*

Then it cannot be represented by finitely many rectilinear segments, and therefore the true decision boundary cannot be obtained by a decision tree classifier.

However, it may be still be possible to approximate `f` *or even to expand the space of representable functions using ensemble methods.*

THE REPRESENTATIONAL PROBLEM – APPROXIMATION

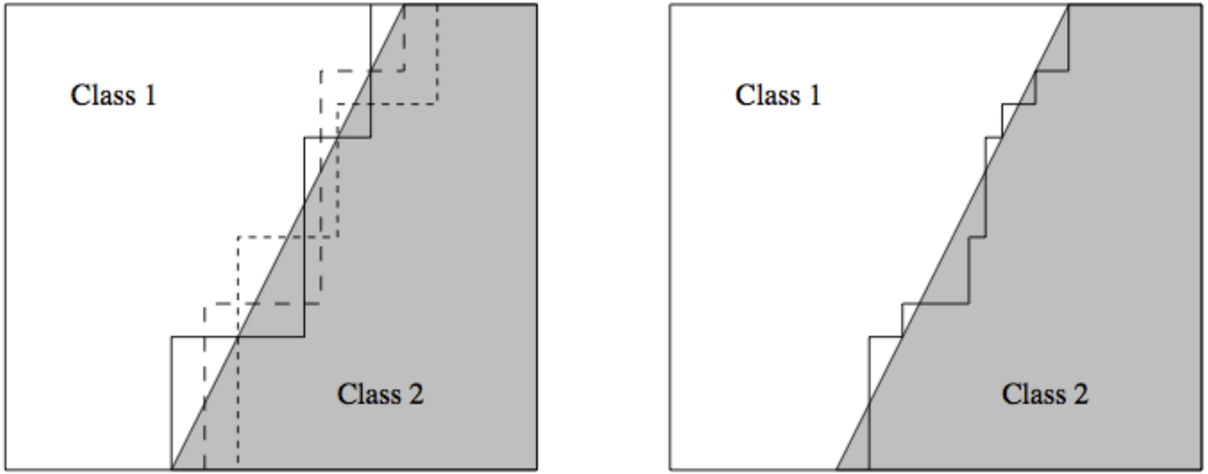
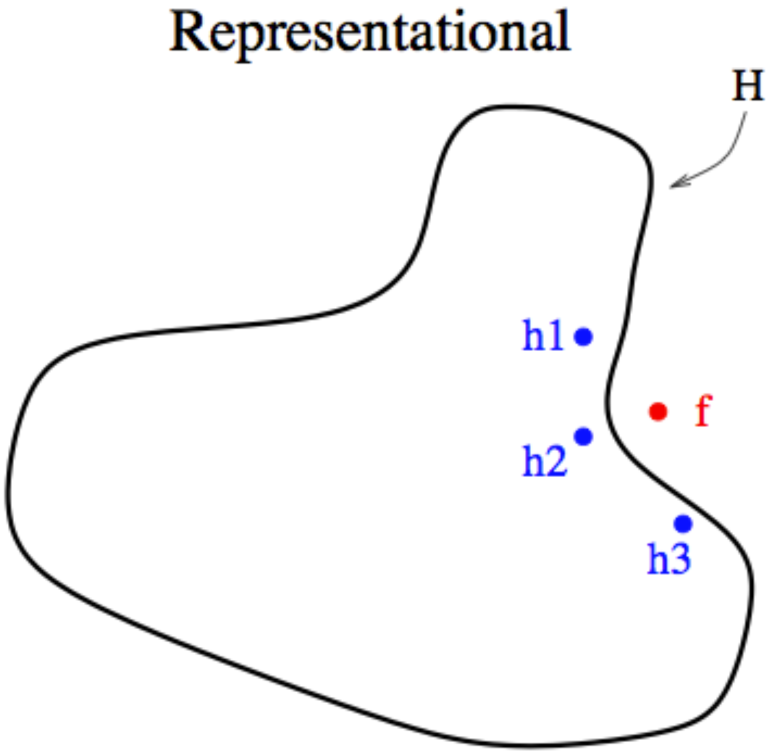


Fig. 4. The left figure shows the true diagonal decision boundary and three staircase approximations to it (of the kind that are created by decision tree algorithms). The right figure shows the voted decision boundary, which is a much better approximation to the diagonal boundary.

NOTE

An ensemble of decision trees can approximate a diagonal decision boundary.



NOTE

Ensemble classifiers can be effective even if the true decision boundary lies outside the hypothesis space.

Q: How do you create an ensemble classifier using averaging methods?

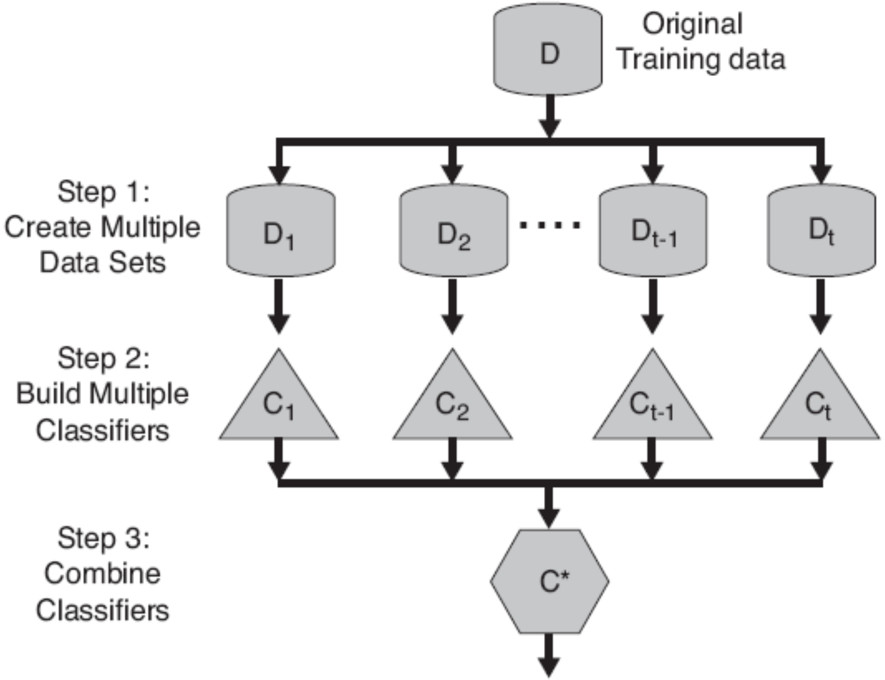


Figure 5.31. A logical view of the ensemble learning method.

Q: How do you generate several base classifiers?

Q: How do you generate several base classifiers?

A: There are several ways to do this:

- *manipulating the training set*
- *manipulating the output labels*
- *manipulating the learning algorithm itself*

We will talk about a few examples of each of these.

III. BAGGING

Bagging *(bootstrap aggregating) is a method that involves manipulating the training set by* **resampling***.*

Bagging *(bootstrap aggregating) is a method that involves manipulating the training set by* **resampling***.*

We learn k *base classifiers on* k *different samples of training data.*

These samples are independently created by resampling the training data using uniform weights (eg, a uniform **sampling distribution***).*

Bagging *(bootstrap aggregating) is a method that involves manipulating the training set by* **resampling**.

We learn k *base classifiers on* k *different samples of training data.*

These samples are independently created by resampling th

data using uniform weights (eg, a uniform **sampling distri**

NOTE

Each training sample is the same size as the original training set.

***Bagging** (bootstrap aggregating) is a method that involves manipulating the training set by **resampling**.*

We learn k *base classifiers on* k *different samples of training data.*

These samples are independently created by resampling th data using uniform weights (eg, a uniform **sampling distril**

NOTE

Resampling means that some training records may appear in a sample more than once, or even not at all.

Bagging *(bootstrap aggregating) is a method that involves manipulating the training set by* **resampling***.*

We learn k *base classifiers on* k *different samples of training data.*

These samples are independently created by resampling the training data using uniform weights (eg, a uniform **sampling distribution***).*

The final prediction is made by taking a majority vote across bc's.

Bagging reduces the variance in our generalization error by aggregating multiple base classifiers together (provided they satisfy our earlier requirements).

Bagging reduces the variance in our generalization error by aggregating multiple base classifiers together (provided they satisfy our earlier requirements).

If the base classifier is stable (to minor changes in training data), then the ensemble error is primarily due to bc bias, and bagging may not be effective.

Bagging reduces the variance in our generalization error by aggregating multiple base classifiers together (provided they satisfy our earlier requirements).

If the base classifier is stable (to minor changes in training data), then the ensemble error is primarily due to bc bias, and bagging may not be effective.

Since each sample of training data is equally likely, bagging is not very susceptible to overfitting with noisy data.

IV. BOOSTING

Boosting is an iterative procedure that adaptively changes the sampling distribution of training records at each iteration.

Boosting is an iterative procedure that adaptively changes the sampling distribution of training records at each iteration.

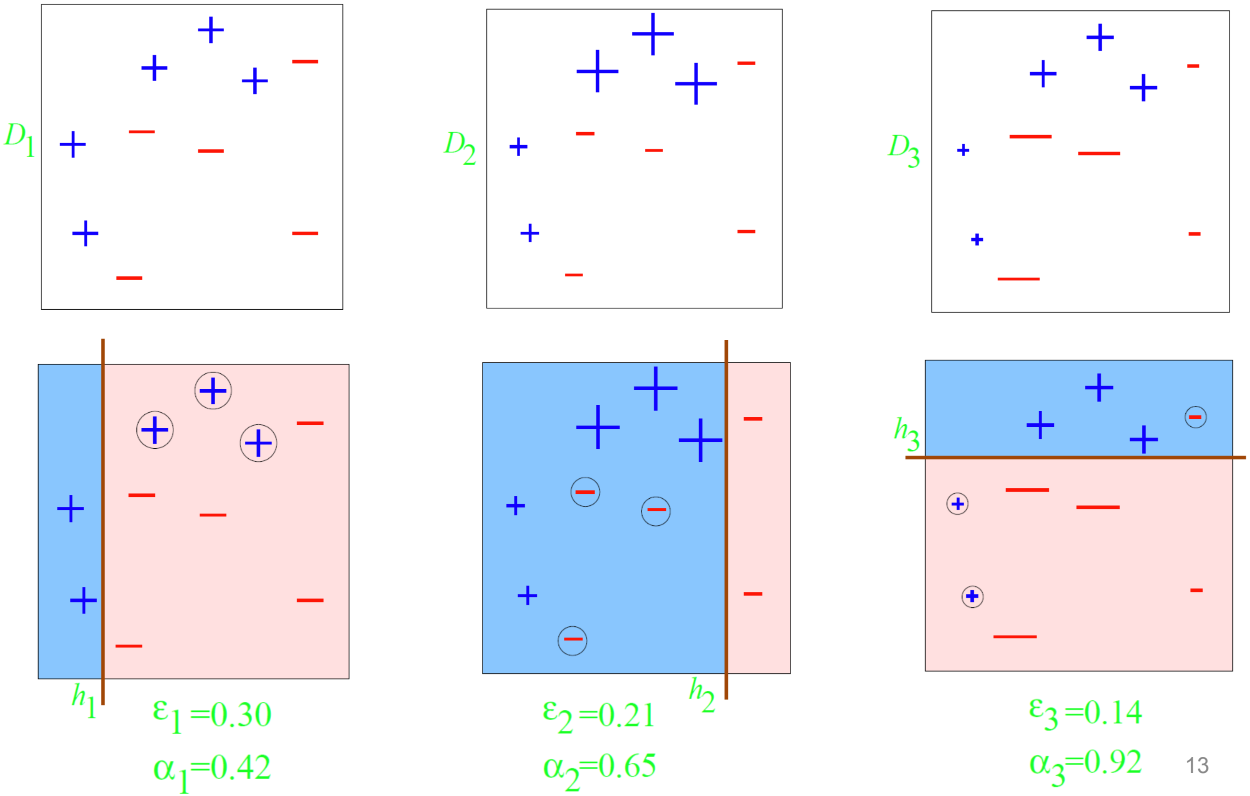
The first iteration uses uniform weights (like bagging). In subsequent iterations, the weights are adjusted to emphasize records that were misclassified in previous iterations.

Boosting is an iterative procedure that adaptively changes the sampling distribution of training records at each iteration.

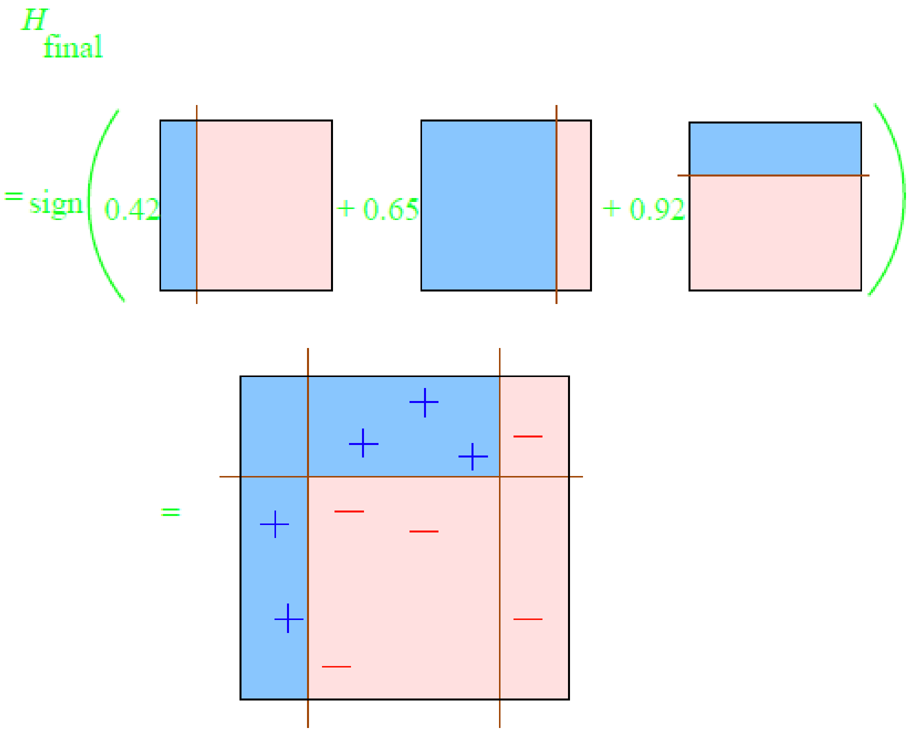
The first iteration uses uniform weights (like bagging). In subsequent iterations, the weights are adjusted to emphasize records that were misclassified in previous iterations.

The final prediction is constructed by a weighted vote (where the weights for a bc depends on its training error).

BOOSTING WITH STUMPS



http://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture10.pdf



http://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture10.pdf

1) *Start with a training set where each item has equal weight*
2) *Build a classifier - G*

1) *Start with a training set where each item has equal weight*
2) *Build a classifier – G*
3) *Compute the error e(G) of the classifier (% incorrect)*

1) *Start with a training set where each item has equal weight*
2) *Build a classifier - G*
3) *Compute the error e(G) of the classifier (% incorrect)*
4) *Build a new training set where the incorrect instance are weighted (repeated) by the error e(G) of the classifier*

1) *Start with a training set where each item has equal weight*
2) *Build a classifier - G*
3) *Compute the error e(G) of the classifier (% incorrect)*
4) *Build a new training set where the incorrect instance are weighted (repeated) by the error e(G) of the classifier*
5) *Repeat*

1) *Start with a training set where each item has equal weight*
2) *Build a classifier - G*
3) *Compute the error e(G) of the classifier (% incorrect)*
4) *Build a new training set where the incorrect instance are weighted (repeated) by the error e(G) of the classifier*
5) *Repeat*
6) *Predict based on majority vote (or vote based on accuracy – inverse error)*

Like in bagging, sampling is done with replacement, and as a result some records may not appear in a given training sample.

Like in bagging, sampling is done with replacement, and as a result some records may not appear in a given training sample.

These omitted records will likely be misclassified, and given greater weight in subsequent iterations once the sampling distribution is updated.

Like in bagging, sampling is done with replacement, and as a result some records may not appear in a given training sample.

These omitted records will likely be misclassified, and given greater weight in subsequent iterations once the sampling distribution is updated.

So even if a record is left out at one stage, it will be emphasized later.

V. RANDOM FORESTS

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

1) *Select a sample of the original training set and build a tree as follows:*

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

1) *Select a sample of the original training set and build a tree as follows:*
    1) *Select m features out of the M available*
    2) *Pick the best split based on just those m variables*

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

1) *Select a sample of the original training set and build a tree as follows:*
    1) *Select m features out of the M available*
    2) *Pick the best split based on just those m variables*

2) *Repeat (1) for N iterations*

A random forest is an ensemble of decision trees where each base classifier is grown using a random effect.

1) *Select a sample of the original training set and build a tree as follows:*
    1) *Select m features out of the M available*
    2) *Pick the best split based on just those m variables*
2) *Repeat (1) for N iterations*
3) *Predict based on majority vote of N trees*

EX: ENSEMBLE METHODS IN SCIKIT-LEARN