GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

LESSON 10: ENSEMBLE METHODS

I. ENSEMBLE TECHNIQUES
II. PROBLEMS IN CLASSIFICATION
III. BAGGING
IV. BOOSTING
V. REAL WORLD ENSEMBLES

QUESTIONS?

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.**

ENSEMBLE TECHNIQUES

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.

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 bc's 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 bc's must be* **accurate***: they must outperform random guessing*

2) *the bc's 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.

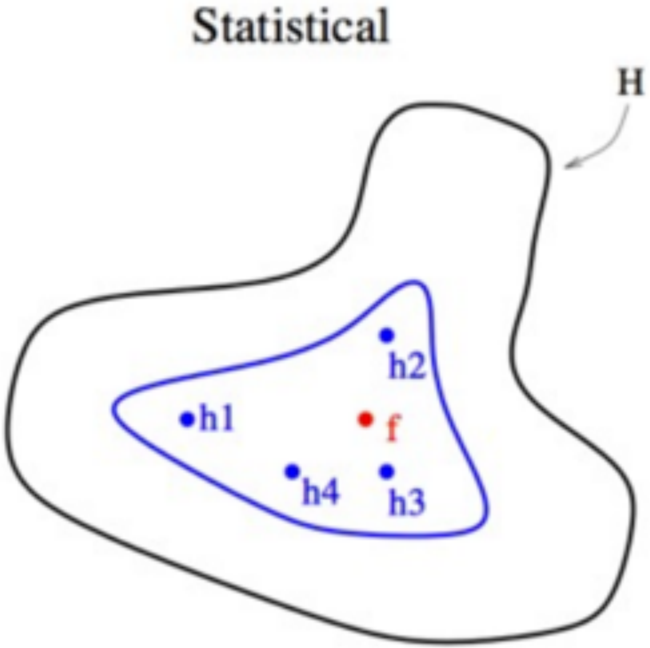
In any supervised learning task, our goal is to make predictions of the true classification function f by learning the classifier h.

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

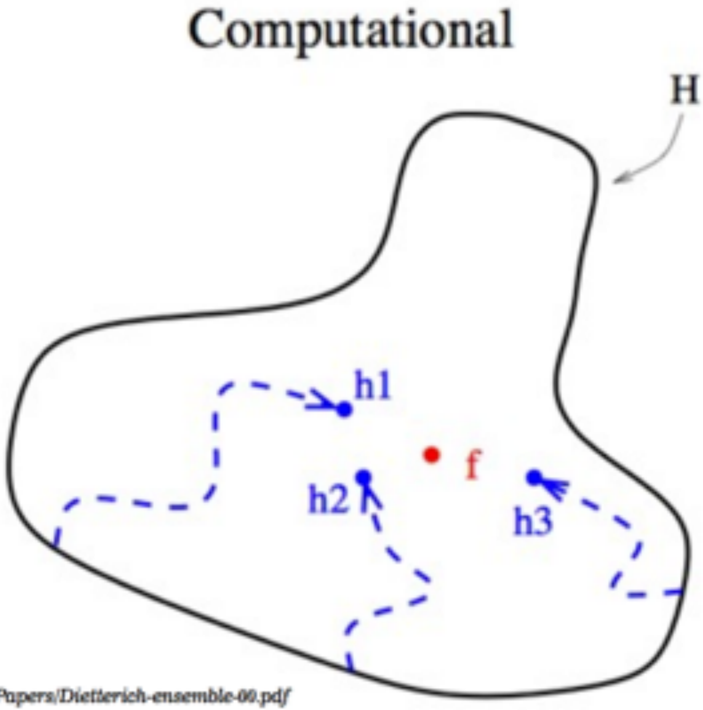
Even with sufficient training data, it may still be computationally difficult to find the best classifier h.

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.



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

NOTE

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

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

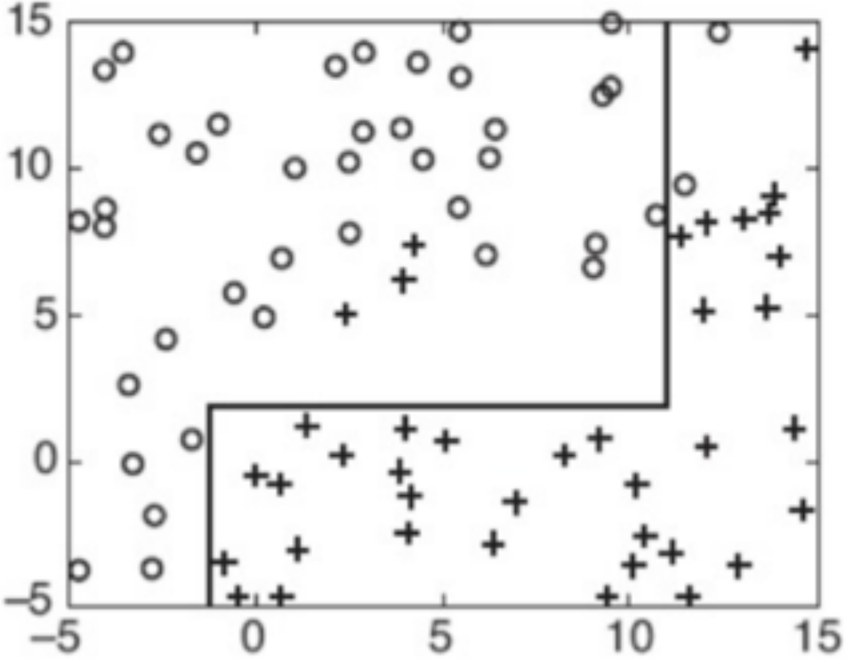
Sometimes f cannot be expressed in terms of our hypothesis at all.

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.



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?

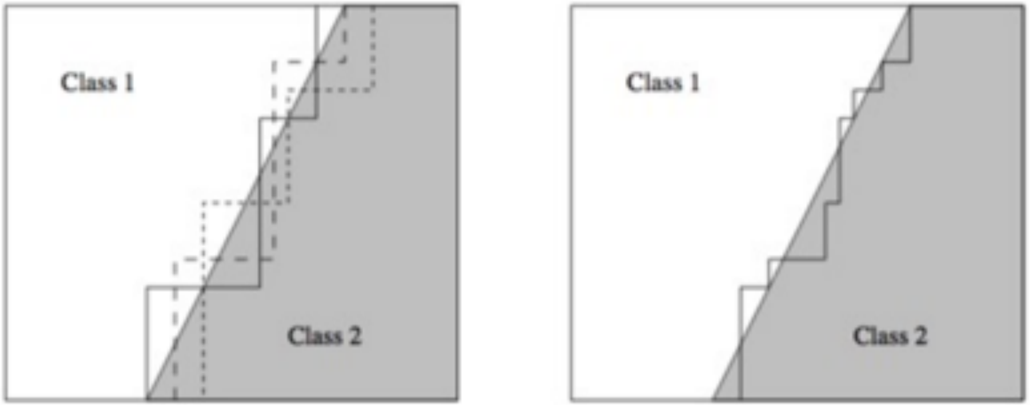
But what if f is a diagonal line?

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.

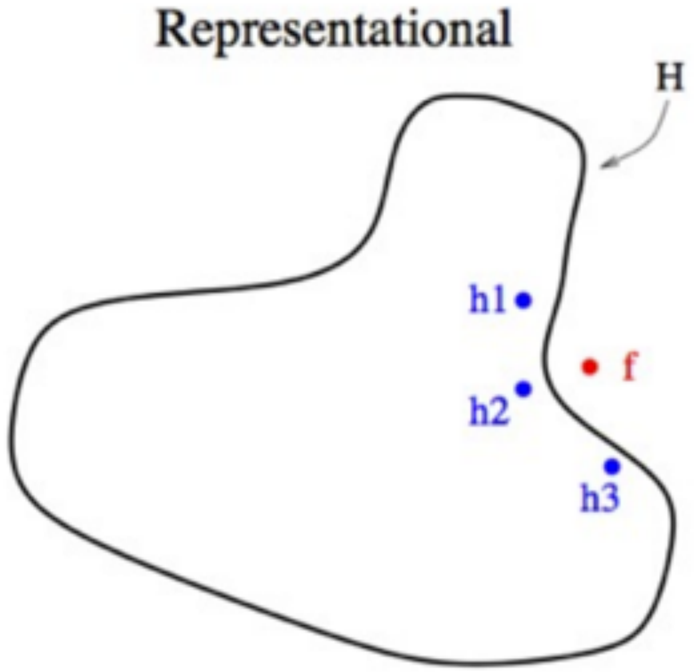
However, it may be still be possible to approximate f or even to expand the space of representable functions using ensemble methods.



NOTE

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

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

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

REVIEW

1. What are the three main problems in classification that ensemble methods can solve?

Q: How do you create an ensemble classifier?

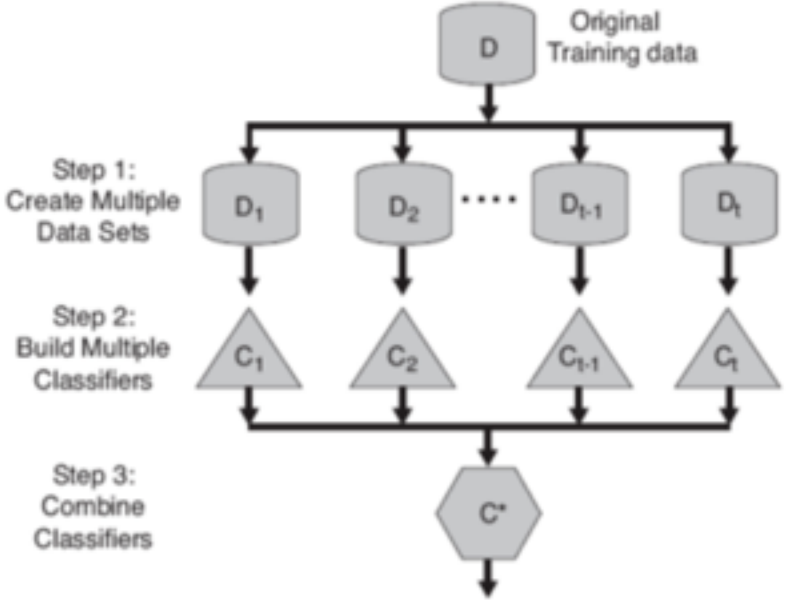


Figure 5.31. A logical view of the ensemble learning method.

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 distril**

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, 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, 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).

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.

Updating the sampling distribution and forming an ensemble prediction leads to a nonlinear combination of the base classifiers.

REVIEW

1. What is the difference between boosting and bagging?

REVIEW

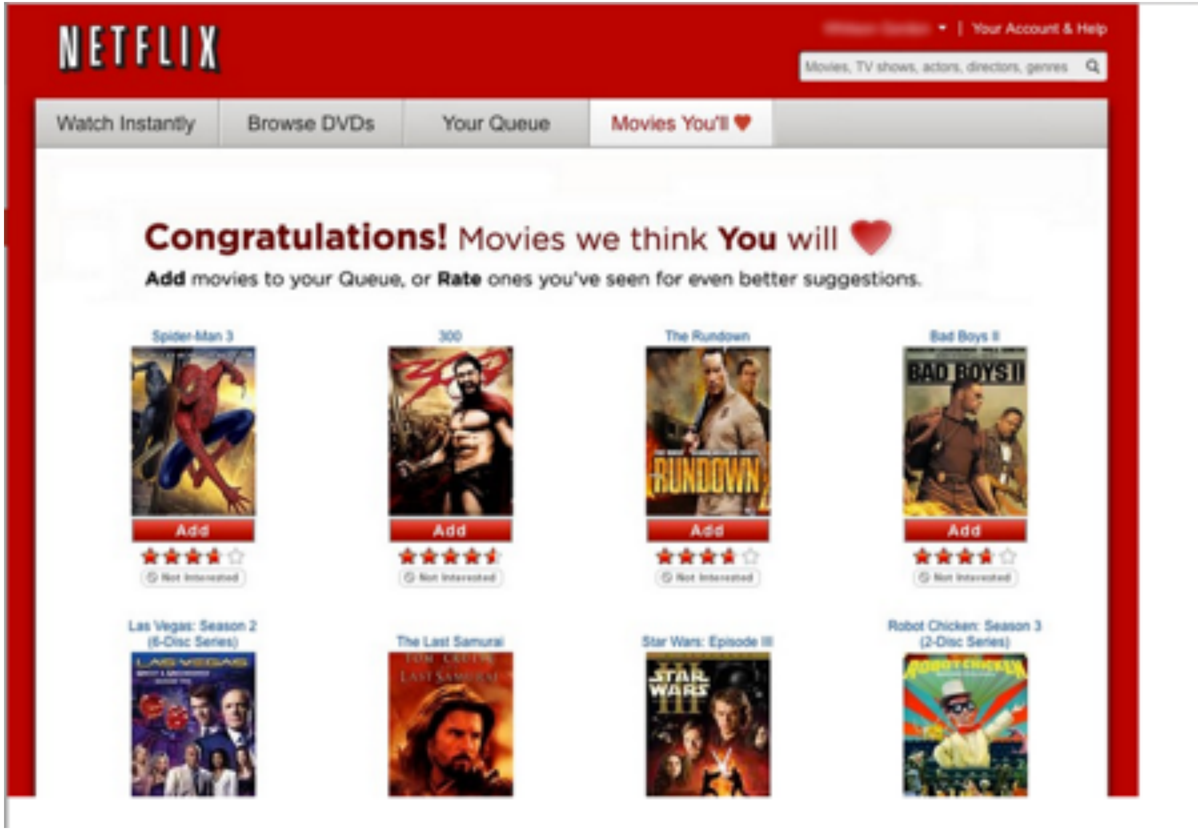
1. What are steps in creating a random forest? How is it similar to boosting and bagging?

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*

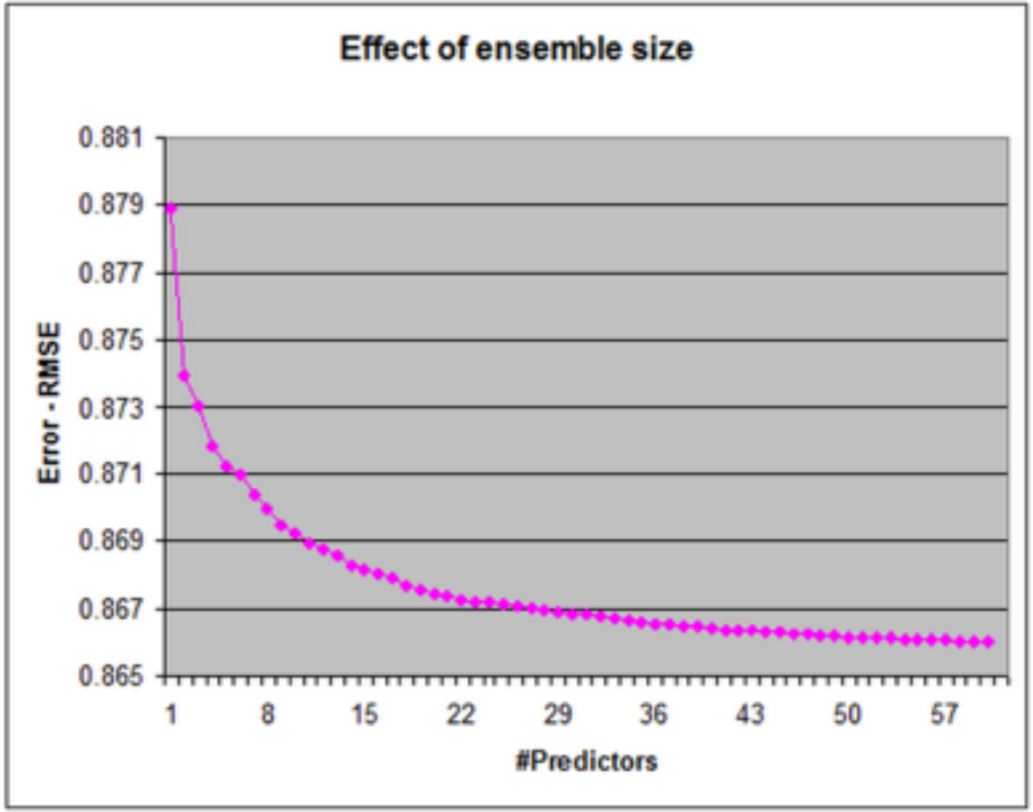
2) *Repeat (1) for N iterations*
3) *Predict based on majority vote of N trees*

V. REAL WORLD ENSEMBLES



NETFLIX
Your Account & Help
Movies, TV shows, actors, directors, genres
Watch Instantly
Browse DVDs
Your Queue
Movies You'll
Congratulations! Movies we think You will
Add movies to your Queue, or Rate ones you've seen for even better suggestions.
Spider-Man 3
300
The Rundown
Bad Boys II
Add
Add
Add
Add
Las Vegas: Season 2 (6-Disc Series)
The Last Samurai
Star Wars: Episode III
Robot Chicken: Season 3 (2-Disc Series)

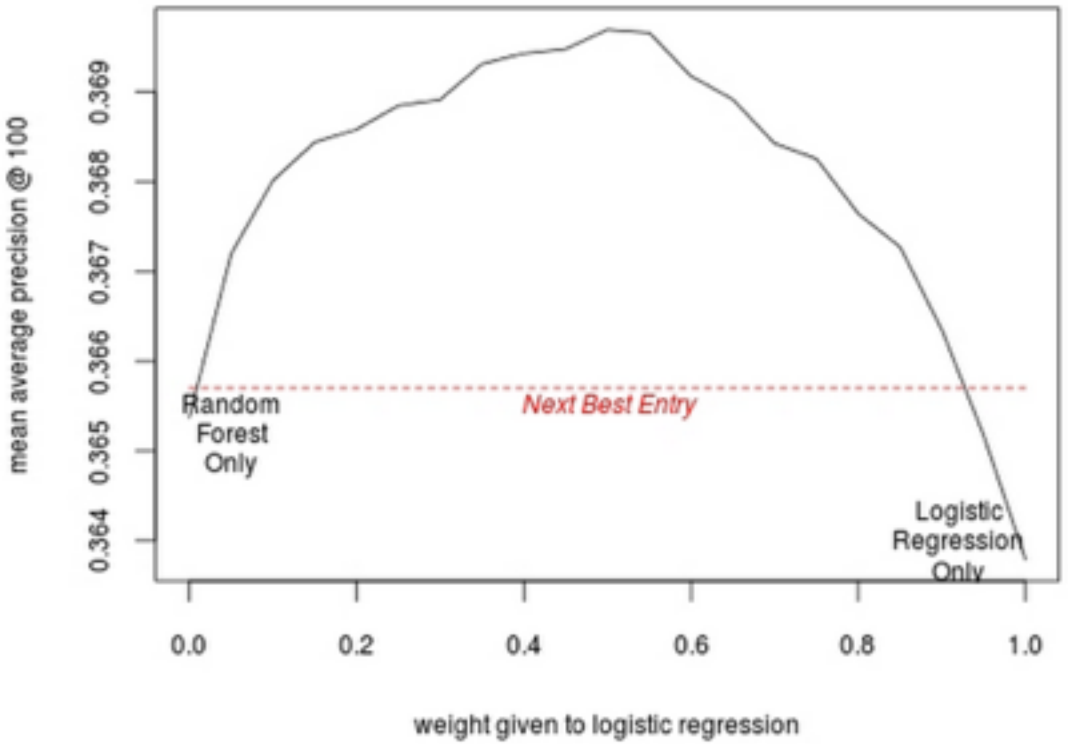
REAL WORLD ENSEMBLES



Quote from BellKor (competition winners):

“However, we would like to stress that it is not necessary to have such a large number of models to do well. The plot below shows RMSE as a function of the number of methods used. One can achieve our winning score (RMSE=0.8712) with less than 50 methods, using the best 3 methods can yield RMSE < 0.8800, which would land in the top 10. Even just using our single best method puts us on the leaderboard with an RMSE of 0.8890. **The lesson here is that having lots of models is useful for the incremental results needed to win competitions, but practically, excellent systems can be built with just a few well-selected models.”**

REAL WORLD ENSEMBLES



REVIEW

I. What are the practical considerations of ensemble methods?

LAB