**Introduction**

**The first reason is statistical**.

A learning algorithm can be viewed as searching a space H of hypotheses to identify the best hypothesis in the space. The statistical problem arises when the amount of training data available is too small compared to the size of the hypothesis space.

**The second reason is computational.**

Many learning algorithms work by performing some form of local search that may get stuck in local optima. For example, neural network algorithms employ gradient descent to minimize an error function over the training data, and decision tree algorithms employ a greedy splitting rue to grow the decision tree. In case where there is enough training data (so that the statistical problem is absent), it may still very difficult computationally for the learning algorithm to find the best hypothesis.

**The third reason is representational.**

In the most applications of machine learning, the true function f cannot be represented by any of the hypotheses in H. By forming weighted sums of the hypotheses drawn from H, it may be possible to expand the space of representable functions.

Diagram

Description automatically generated

**Methods for Constructing Ensembles**

1. **Bayesian Voting**
2. **Manipulating the Training Examples**
3. **Manipulating the Input Feature**
4. **Manipulating the Output Targets**
5. **Injecting Randomness**

**Comparing Different Ensembles Methods**

**Conclusions**

**Ensembles are well-established as a method for obtaining highly accurate classifiers by combining less accurate ones. This paper has provided a brief survey of methods for constructing ensembles and reviewed the three fundamental reasons why ensemble methods are able to out-perform any single classifier within the ensemble. The paper has also provided some experimental results to elucidate one of the reasons why AdaBoost performs so well.**

**Links to explore:**

[**https://stats.stackexchange.com/questions/18891/bagging-boosting-and-stacking-in-machine-learning**](https://stats.stackexchange.com/questions/18891/bagging-boosting-and-stacking-in-machine-learning)

[**https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205**](https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205)

1. [Bagging](http://en.wikipedia.org/wiki/Bootstrap_aggregating) **(stands for Bootstrap Aggregating) is a way to decrease the variance of your prediction by generating additional data for training from your original dataset using** [**combinations with repetitions**](http://en.wikipedia.org/wiki/Combinations) **to produce** [**multisets**](http://en.wikipedia.org/wiki/Multiset) **of the same cardinality/size as your original data. By increasing the size of your training set you can't improve the model predictive force, but just decrease the variance, narrowly tuning the prediction to expected outcome.**
2. [Boosting](http://en.wikipedia.org/wiki/Boosting_(machine_learning)) **is a two-step approach, where one first uses subsets of the original data to produce a series of averagely performing models and then "boosts" their performance by combining them together using a particular cost function (=majority vote). Unlike bagging, in the** [**classical boosting**](http://www.cs.princeton.edu/courses/archive/spr08/cos424/readings/Schapire2003.pdf) **the subset creation is not random and depends upon the performance of the previous models: every new subsets contains the elements that were (likely to be) misclassified by previous models.**
3. [Stacking](http://en.wikipedia.org/wiki/Ensemble_learning#Stacking) **is like boosting you also apply several models to your original data. The difference here is, however, that you don't have just an empirical formula for your weight function, rather you introduce a meta-level and use another model/approach to estimate the input together with outputs of every model to estimate the weights or, in other words, to determine what models perform well and what badly given these input data.**

**To recap in short, Bagging and Boosting are normally used inside one algorithm, while Stacking is usually used to summarize several results from different algorithms.**

1. **Bagging: Bootstrap subsets of features and samples to get several predictions and average (or other ways) the results, for example, Random Forest, which eliminate variance and does not have overfitting issue.**
2. **Boosting: The difference from Bagging is that later model is trying to learn the error made by previous one, for example GBM and XGBoost, which eliminate the variance but have overfitting issue.**
3. **Stacking: Normally used in competitions, when one uses multiple algorithms to train on the same data set and average (max, min or other combinations) the result in order to get a higher accuracy of prediction.**