Applied machine learning is challenging because the designing of a perfect learning system for a given problem is intractable.

There is no best training data or best algorithm for your problem, only the best that you can discover.

The application of machine learning is best thought of as search problem for the best mapping of inputs to outputs given the knowledge and resources available to you for a given project.

In this post, you will discover the conceptualization of applied machine learning as a search problem.

After reading this post, you will know:

- That applied machine learning is the problem of approximating an unknown underlying mapping function from inputs to outputs.
- That design decisions such as the choice of data and choice of algorithm narrow the scope of possible mapping functions that you may ultimately choose.
- That the conceptualization of machine learning as a search helps to rationalize the use of ensembles, the spot checking of algorithms and the understanding of what is happening when algorithms learn.

Let’s get started.

Overview

This post is divided into 5 parts; they are:

1. Problem of Function Approximation
2. Function Approximation as Search
3. Choice of Data
4. Choice of Algorithm
5. Implications of Machine Learning as Search

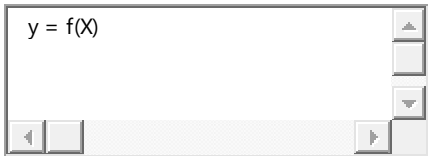
Problem of Function Approximation

Applied machine learning is the development of a learning system to address a specific learning problem.

The learning problem is characterized by observations comprised of input data and output data and some unknown but coherent relationship between the two.

The goal of the learning system is to learn a generalized mapping between input and output data such that skillful predictions can be made for new instances drawn from the domain where the output variable is unknown.

In statistical learning, a statistical perspective on machine learning, the problem is framed as the learning of a mapping function (*f*) given input data (*X*) and associated output data (*y*).

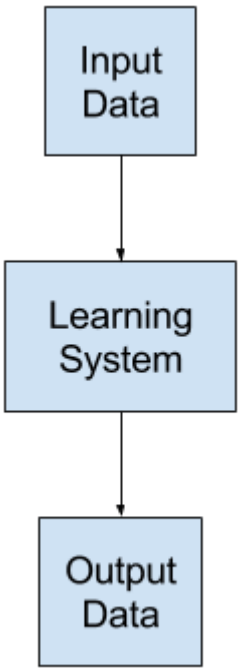



```
1  y = f(X)
```

We have a sample of *X* and *y* and do our best to come up with a function that approximates *f*, e.g. *fprime*, such that we can make predictions (*yhat*) given new examples (*Xhat*) in the future.


```
1  yhat = fprime(Xhat)
```

As such, applied machine learning can be thought of as the problem of function approximation.



Machine learning as the mapping from inputs to outputs

The learned mapping will be imperfect.

The problem of designing and developing a learning system is the problem of learning a useful approximate of the unknown underlying function that maps the input variables to the output variables.

We do not know the form of the function, because if we did, we would not need a learning system; we could specify the solution directly.

Because we do not know the true underlying function, we must approximate it, meaning we do not know and may never know how close of an approximation the learning system is to the true mapping.

Function Approximation as Search

We must search for an approximation of the true underlying function that is good enough for our purposes.

There are many sources of noise that introduce error into the learning process that can make the process more challenging and in turn result in a less useful mapping. For example:

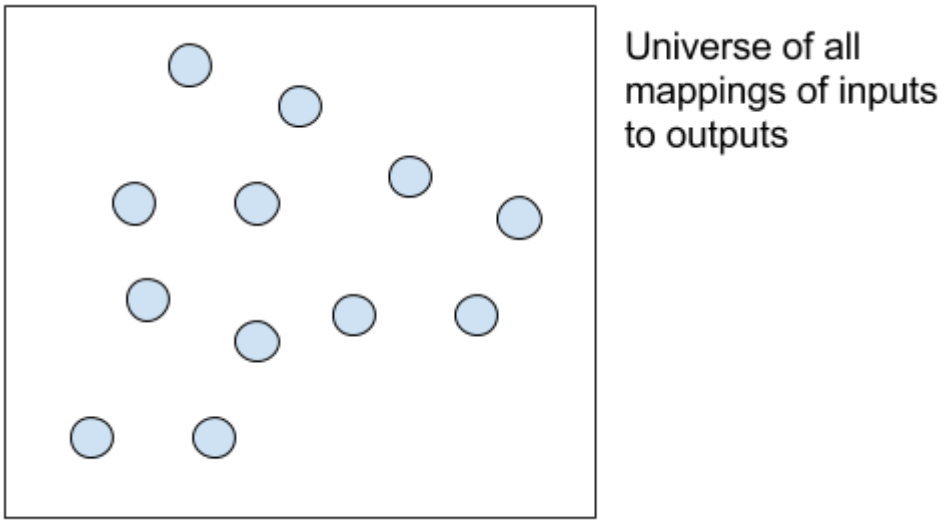
- The choice of the framing of the learning problem.
- The choice of the observations used to train the system.

- The choice of how the training data is prepared.
- The choice of the representational form for the predictive model.
- The choice of the learning algorithm to fit the model on the training data.
- The choice of the performance measure by which to evaluate predictive skill.

And so much more.

You can see that there are many decision points in the development of a learning system, and none of the answers are known beforehand.

You can think of all possible learning systems for a learning problem as a huge search space, where each decision point narrows the search.



Search space of all possible mapping functions from inputs to outputs

For example, if the learning problem was to predict the species of flowers, one of millions of possible learning systems could be narrowed down as follows:

- Choose to frame the problem as predicting a species class label, e.g. classification.
- Choose measurements of the flowers of a given species and their associated sub-species.
- Choose flowers in one specific nursery to measure in order to collect training data.
- Choose a decision tree model representation so that predictions can be explained to stakeholders.
- Choose the CART algorithm to fit the decision tree model.
- Choose classification accuracy to evaluate the skill of models.

And so on.

You can also see that there may be a natural hierarchy for many of the decisions involved in developing a learning system, each of which further narrows the space of possible learning systems that we could build.

This narrowing introduces a useful bias that intentionally selects one subset of possible learning systems over another with the goal of getting closer to a useful mapping that we can use in practice. This biasing applies both at the top level in the framing of the problem and at low levels, such as the choice of machine learning algorithm or algorithm configuration.

Choice of Data

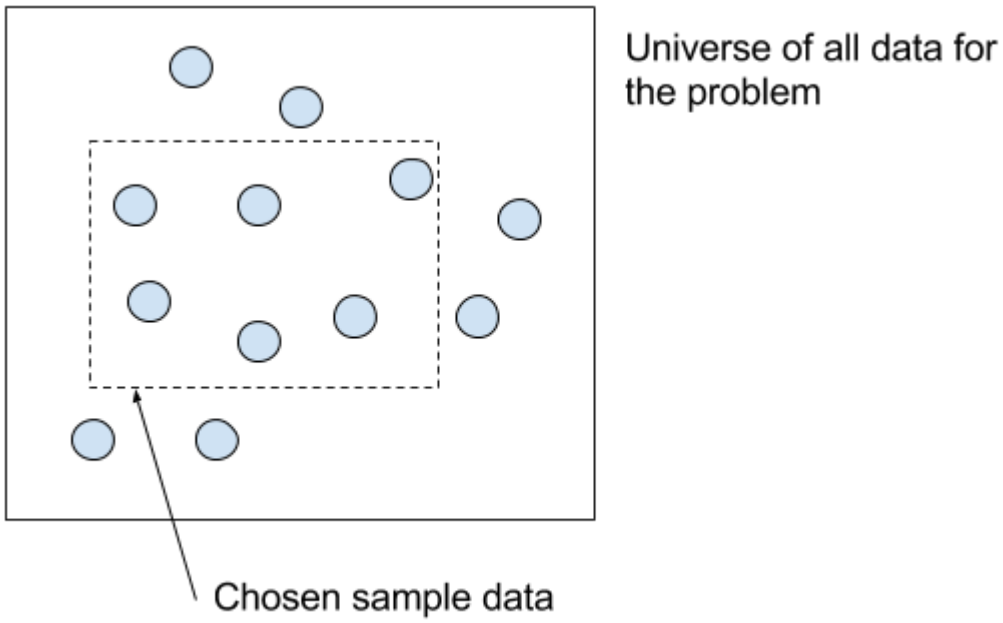
The chosen framing of the learning problem and the data used to train the system are a big point of leverage in the development of your learning system.

You do not have access to all data: that is all pairs of inputs and outputs. If you did, you would not need a predictive model in order to make output predictions for new input observations.

You do have some historical input-output pairs. If you didn’t, you would not have any data with which to train a predictive model.

But maybe you have a lot of data and you need to select only some of it for training. Or maybe you have the freedom to generate data at will and are challenged by what and how much data to generate or collect.

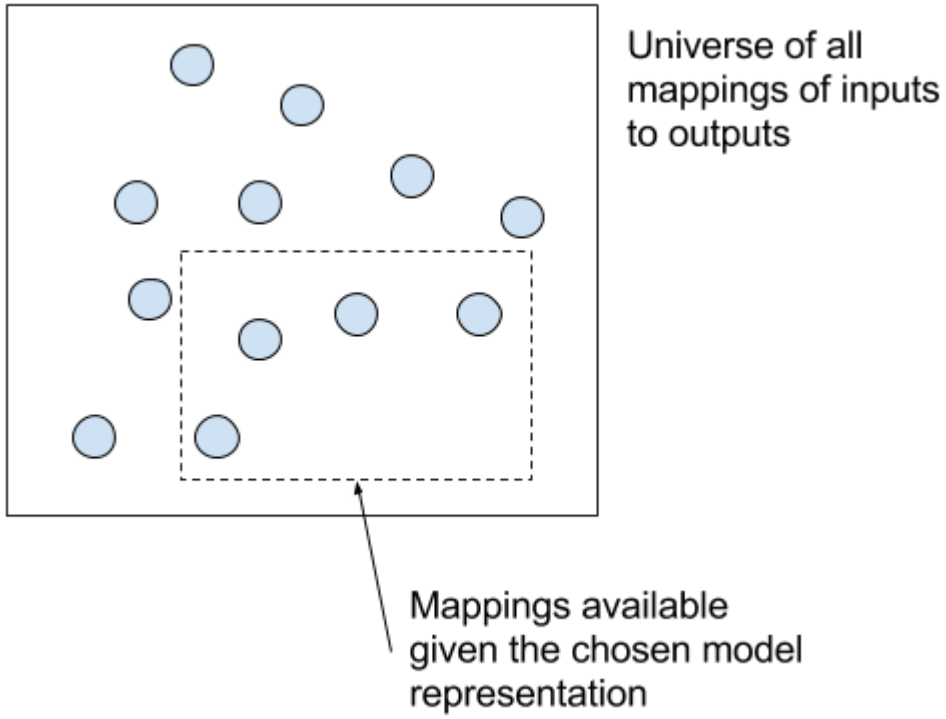
The data that you choose to model your learning system on must sufficiently capture the relationship between the input and output data for both the data that you have available and data that the model will be expected to make predictions on in the future.



Choice of training data from the universe of all data for a problem

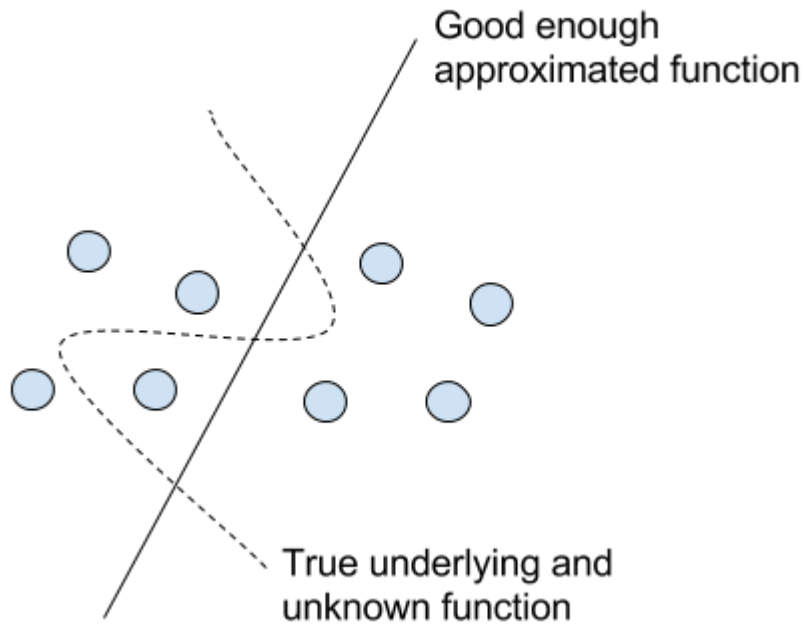
Choice of Algorithm

You must choose the representation of the model and the algorithm used to fit the model on the training data. This, again, is another big point of leverage on the development of your learning system.



Choice of algorithm from the universe of all algorithms for a problem

Often this decision is simplified to the selection of an algorithm, although it is common for the project stakeholders to impose constraints on the project, such as the model being able to explain predictions which in turn imposes constraints on the form of the final model representation and in turn on the scope of mappings that you can search.



Effect of choosing an approximate mapping from inputs to outputs

Implications of Machine Learning as Search

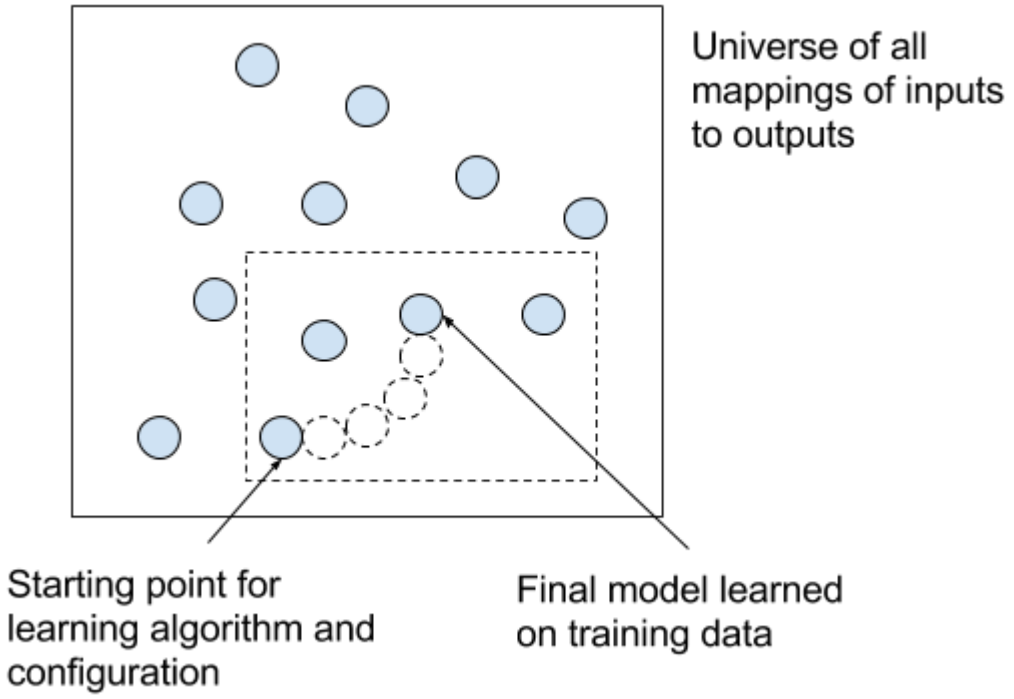
This conceptualization of developing learning systems as a search problem helps to make clear many related concerns in applied machine learning.

This section looks at a few.

Algorithms that Learn Iteratively

The algorithm used to learn the mapping will impose further constraints, and it, along with the chosen algorithm configuration, will control how the space of possible candidate mappings is navigated as the model is fit (e.g. for machine learning algorithms that learn iteratively).

Here, we can see that the act of learning from training data by a machine learning algorithm is in effect navigating the space of possible mappings for the learning system, hopefully moving from poor mappings to better mappings (e.g. hill climbing).



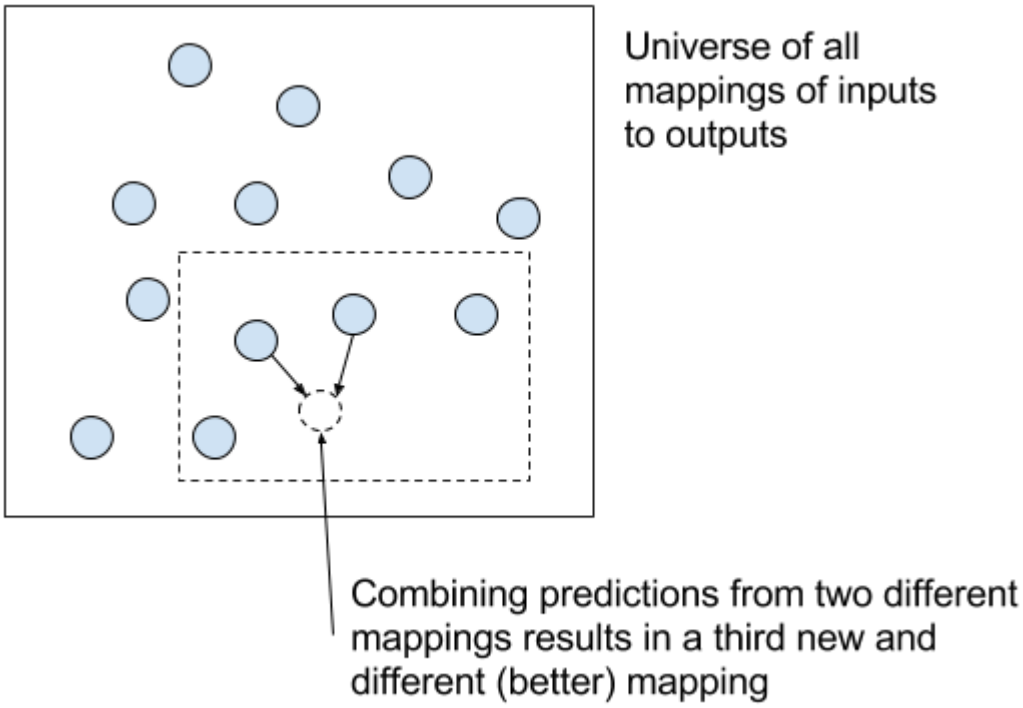
Effect of a learning algorithm iteratively training on data

This provides a conceptual rationale for the role of optimization algorithms in the heart of machine learning algorithms to get the most out of the model representation for the specific training data.

Rationale for Ensembles

We can also see that different model representations will occupy quite different locations in the space of all possible function mappings, and in turn have quite different behavior when making predictions (e.g. uncorrelated prediction errors).

This provides a conceptual rationale for the role of ensemble methods that combine the predictions from different but skillful predictive models.



Interpretation of combining predictions from multiple final models

Rationale for Spot Checking

Different algorithms with different representations may start in different positions in the space of possible function mappings, and will navigate the space differently.

If the constrained space that these algorithms are navigating is well specified by an appropriating framing and good data, then most algorithms will likely discover good and similar mapping functions.

We can also see how a good framing and careful selection of training data can open up a space of candidate mappings that may be found by a suite of modern powerful machine learning algorithms.

This provides rationale for spot checking a suite of algorithms on a given machine learning problem and doubling down on the one that shows the most promise, or selecting the most parsimonious solution (e.g. Occam's razor).