

MIT Splash

Fall 2014

Machine Learning and Audio Analysis with Python

Course Notes

Daryl Sew

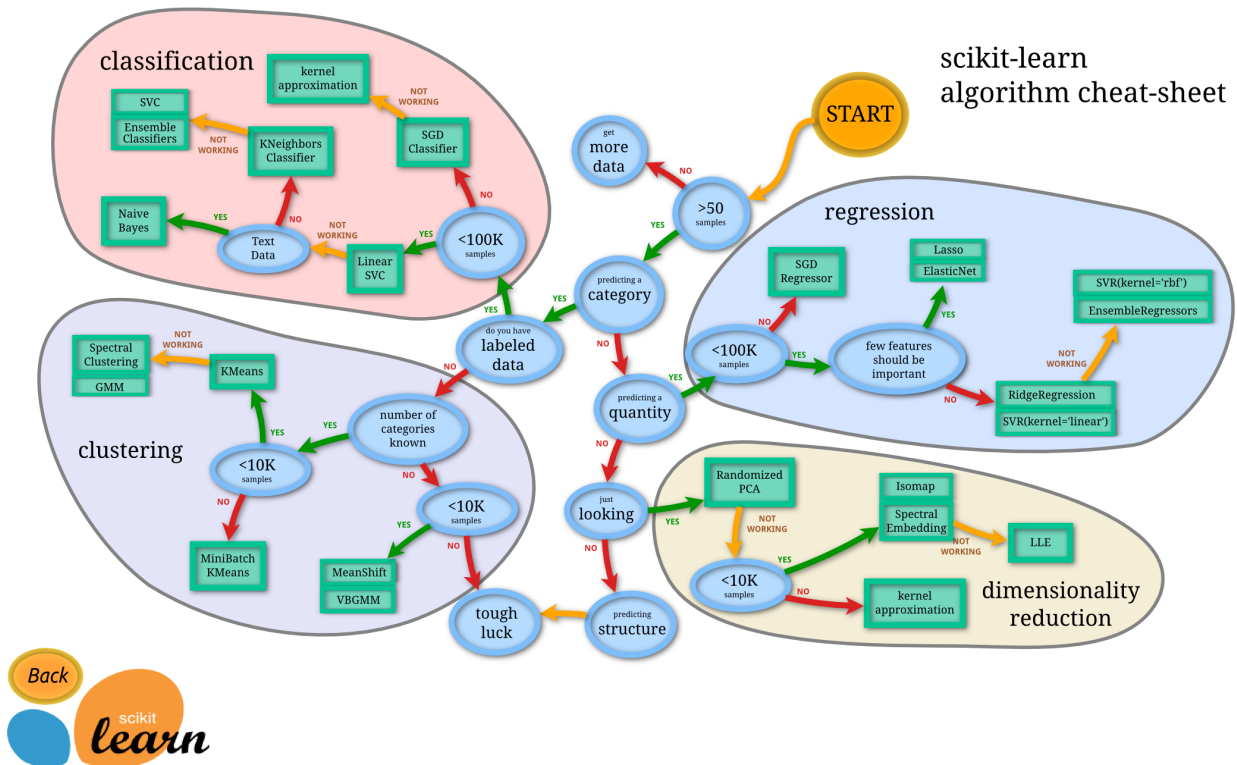


Figure 1: Flow chart for selecting the right class of algorithm for your problem from `scikit-learn`

Contents

1	Overview	2
2	What is Machine Learning?	2
2.1	A Brief, Vague History	2
2.2	Main Categories	3
3	Unsupervised Learning	3
3.1	K-means clustering	3
3.1.1	Theory	3
3.1.2	Applications	3
4	Supervised Learning	4
4.1	Regression	4
4.1.1	Theory	4
4.1.2	Applications	4
4.2	Support Vector Machines	4
4.2.1	Theory	4
4.2.2	Applications	4
5	Reinforcement Learning	4
5.1	Theory	4
5.1.1	Applications	4
6	Common Pitfalls (Overfitting)	4
6.1	Classifier Choice	4
6.2	Sampling Bias	5
6.3	Avoiding Overfitting	5
7	Algorithm Design Exercises	5
7.1	Pokemon	5
8	What is Audio?	5

1 Overview

Machine learning is a field of computer science that concerns writing programs that can make and improve predictions or behaviors based on data inputs. The applications of machine learning are very diverse - they range from self driving cars to spam filters to autocorrect algorithms and much more. Using scikit-learn, an open source machine learning library for Python, we'll cover reinforcement learning (the kind used to create artificial intelligence for games like chess), supervised learning (the kind used in handwriting recognition), and unsupervised learning (the kind eBay uses to group its products). We'll then cover audio analysis through Fourier transforms with numpy, an open source general purpose computational library for Python, and we'll use our newfound audio analysis and machine learning skills to write very basic speech recognition software. Applications of machine learning to the fields of multitouch gesture recognition and computer vision will also be discussed, drawing from my work at Tesla and research on self driving cars and autonomous submarines.

2 What is Machine Learning?

All machine learning algorithms aim to take observations from a system and produce a model of that system. The key words here are 'system' and 'model'. Systems include anything from stock markets to populations of organisms to the environment surrounding a robot, essentially anything imaginable for which you can record observations. Models are a set of mathematical rules that describe a system. Note that in the process of making observations about a system, there will almost always be measurement error - we refer to this as noise, and a good data scientist is able to apply machine learning algorithms in a way that extracts regularities from observations and models those rather than modeling noise.

2.1 A Brief, Vague History

The first computer programs were based on explicit instructions; if this, then that. This type of programming is well suited to many tasks, but by no means could these programs be considered 'intelligent' - they were deterministic and could only react to whatever situations their programmer had manually prepared them for. One of the first intelligent programs was written by Arthur Samuel at IBM in 1952 - he applied reinforcement learning to the game of checkers in order to create a model that ranked playing strategies. This program could now improve its performance each game (update its model favorably), and mimicked intelligence. Around the same time, Marvin Minsky (MIT Professor, started CSAIL, AI pioneer) and Seymour Papert (MIT Professor, inventor of NetLogo + Lego Mindstorms, learning pioneer) were working on something called 'perceptrons'. Perceptrons were a primitive form of machine learning classifier that are modeled after the way our brains work, with neurons. In our bodies, neurons are the mechanisms by which the different parts of our body communicate with each other. Neurons conduct an electrical signal called an action potential, and different neurons are connected by 'synapses'. At a synapse, next neuron is only triggered if a certain threshold is exceeded. In a perceptron, there are a number of inputs to a 'synapse', and when some function of the input values exceeds a threshold, the output is 'fired' to 1 from 0. A single perceptron can only model 'linearly separable' data, as the perceptron model functions as a dividing line between data. In the late 1960s, Minsky and Seymour wrote a book, *Perceptrons: An Introduction to Computational Geometry*, on perceptrons in which they discuss the impossibility of modeling the 'xor' function with this classifier, as xor is not 'linearly separable'. This book, along with a number of similarly unfortunate findings (DARPA cut general AI funding after

a switch to mission oriented funding, Speech Understanding Research at CMU failed to produce a satisfactory product for DARPA, UK killed AI program), resulted in something referred to as the 'AI Winter', where AI research stagnated until the 1990s. Then, new discoveries and funding revival within different fields of AI and countries led to a new explosion in research; multilayer perceptrons now known as neural networks were developed which could model nonlinearities in data, speech recognition became a commercial success, to name a few, and AI was combined with statistics to solve very practical problems, giving birth to the field now known as machine learning. Since then, machine learning has progressed to the point where knowledge of machine learning and statistical problem solving methods is helpful in any STEM field, and it's also in my opinion one of the coolest subfields of computer science.

2.2 Main Categories

Unsupervised learning is used to find hidden structure in unlabeled data. Finding structure can answer a lot of questions about a dataset, as well as guide further exploration. An example structure is a set of groups of data points that are considered by some distance metric to be very close together. Supervised learning is used to infer a function that describes a set of labeled data such that labels can be attached to newly collected, unlabeled data. Reinforcement learning is used to find the optimal set of actions to take in a dynamic environment.

3 Unsupervised Learning

3.1 K-means clustering

3.1.1 Theory

1. Gather a set of points in n dimensions.
2. Select k points at random from this data set; these will be the 'k means' (or centroids).
3. Assign all points to the 'cluster' containing the nearest centroid.
4. Update the centroids of the clusters to reflect the new point memberships.
5. Repeat 3-4 until an update results in no change (this is referred to as convergence, a la sequences and series).

3.1.2 Applications

One particularly important application of unsupervised learning is constructing gene families. DNA is central to our existence; it contains all the information our cells (and ourselves) require to replicate. This information is stored physically in the famous double helix (usually wound up into histones) and can be extracted and represented holistically with strings (‘‘ATGTCTATGAACCATC’’).

Another example relates to some research I've been doing with self driving cars. To give you a little background, let's watch a short video. . . . -spiel on sensors in the car

-spiel on sensor fusion

-spiel on localization

-spiel on calibration procedure, applying ransac

4 Supervised Learning

4.1 Regression

Regression is useful for creating a continuous model of a system based on inputs and outputs.

4.1.1 Theory

The derivation for linear regression is a minimization of the squared error term between the output function and the input dataset.

4.1.2 Applications

4.2 Support Vector Machines

SVMs are useful for creating a discrete model of a system based on inputs and outputs.

4.2.1 Theory

It's all about finding the best set of dividing hyperplanes for our data...

4.2.2 Applications

SVMs are probably my favorite classifier. -tesla spiel

-gesture recognition

-audio recognition (save for later)

5 Reinforcement Learning

5.1 Theory

There's actually not much more general reinforcement learning theory than the agent state action model here. However, each

5.1.1 Applications

box2d.js auv proj (link to RISE based control)

6 Common Pitfalls (Overfitting)

Most common pitfalls within machine learning lead to overfitting, a condition where a model describes random error, noise, or quirks peculiar to the model inputs, rather than describing the system it is meant to.

6.1 Classifier Choice

The choice of classifier has a significant effect on whether data is overfit or not. For example, fitting the following set of (clearly linear) data with a polynomial function results in severe overfitting: But applying a linear regression works just fine. Likewise, many machine learning algorithms apply

something called a 'kernel' to the data they process. The SVM described earlier applied a 'linear' kernel; for very nonlinear data (i.e. orientations), a linear SVM could overfit/badly fit its data.

6.2 Sampling Bias

6.3 Avoiding Overfitting

William of Ockham, Franciscan friar, theologian and philosopher, devised a principle hundreds of years ago now known as Occam's razor: In the absence of certainty, among competing hypotheses, the one with the fewest assumptions (i.e. the simplest one) should be selected. Occam's razor guides the application of machine learning classifiers in that adhering to it tends to minimize overfitting, as simpler classifiers have less room for overfitting. There also exist 'cross validation' techniques by which training data and test data are split up so as to ensure the model will extrapolate to data not contained in the input set.

7 Algorithm Design Exercises

Let's see if we can apply these techniques to a novel problem. Note that, as with many problems within data science/machine learning, there isn't necessarily a 'best' way to solve the following problem, as many different methods can arrive at models that are equally good.

7.1 Pokemon

You've been tasked with building an AI for Pokemon from scratch. At a high level, how would you go about doing this? You have unlimited time and resources. Let's say you're not starting from scratch, and you've been provided with data about every Pokemon Showdown match ever. How does this change your approach?

8 What is Audio?