Kernel and Ensemble

Michael Stinnett
Spencer Gray
10/23/2022

How SVM works, and how SVM kernels work, your impression of the strengths and weaknesses of SVM?

SVM is short for Support Vector Machine. This algorithm divides data using a multidimensional hyperplane. Running alongside the hyperplane is a margin that indicates a multidimensional distance from the hyperplane. Several data points will define that distance particularly. Those data points are called the support vectors. Where if a data point is at least as far from the hyperplane as a support vector in that direction, then that data point belongs to a specific class. This creates the decision boundaries for the data. Where the margin from the hyperplane is no bigger or smaller than needed.

By default SVM assumes that the data can be separated linearly. Where the decision boundary created by the hyperplane is linear. However, SVM also has several other kernels including radial and polynomial. Depending how your data is shaped will inform you on the best kernel to choose.

The main strengths of SVM is its versatility. SVM works in a wide range of scenarios with several hyperparameters to tweak to give the user a wide array of options when using the algorithm. SVM works well in both classification and regression. The biggest weakness of SVM is its computational complexity. SVM has a rigorous mathematical background, thus putting the computer to the test. Not all machines can feasibly run some variations of SVM on larger sets of data.

How Random Forest works, and how the other 2 algorithms you used work compared to the simple decision tree. What are their strengths and weaknesses?

Random forest is actually called a forest because interestingly enough it contains multiple decision trees, each trained on a subset of the data. The branch condition in all of these trees is chosen randomly to fit their subset. A predetermined n number is chosen which determines how many trees are generated. When we run the model for classification

Michael Stinnett
Spencer Gray
10/23/2022

purposes, all the trees go through the test. Each tree one voting on what they think is the likely result, a majority vote is used to classify the data. Its benefits and drawbacks are similar to those of its decision tree components. In that it is highly interpretable and fast to train but may come at a cost for efficiency.

XGBoost also uses trees but in a slightly different manner. Unlike random forest where each tree had a subset, the entire set is used at once with one tree. However, this tree goes through multiple iterations slowly improving its accuracy. Its training time mostly depends on the number of parameters and number of rounds to train. In addition, its performance is much faster. Overall, XGBoost is a strong model, a good one for anyone who does not know where to start.

Aside from boosting and bagging is another method known as stacking. This is somewhat of a combination of the aforementioned models. It trains multiple models and combines them into a singular mode. Although a good idea in theory, it seems to flop in practice either on its own or its difficulty to implement.