Kernel and Ensemble Methods: Narrative Document

Isabelle Kirby, Bridgette Bryant

1 SVM

1.1 Regression

1.1.1 Linear

Linear SVM utilizes a linear decision boundary, which is defined by a vector. The goal of the SVM linear algorithm is to find the best a linear hyperplane to split predictors. It has a hyperparameter cost, which controls how much we allow the variables to violate the decision boundary. On each side of the linear hyperplane, it attempts to maximize the margin, which is the space between the hyperplane and the closest support vector. A support vector is a data point on the edge of the margin. I think this SVM is very good for data which can be easily split into linear sections. The hyperplane can also utilize multiple dimensions.

1.1.2 Polynomial

Polynomial SVM utilizes a polynomial decision boundary, which is defined by a vector. The goal of the SVM polynomial algorithm is to find the best a polynomial hyperplane to split predictors. It has a hyperparameter cost, which controls how much we allow the variables to violate the decision boundary. On each side of the polynomial hyperplane, it attempts to maximize the margin, which is the space between the hyperplane and the closest support vector. A support vector is a data point on the edge of the margin. I think this SVM is very good for data which can be easily split into polynomial sections. The hyperplane can also utilize multiple dimensions.

1.1.3 Radial

Polynomial SVM utilizes a radial decision boundary (circle based), which is defined by a vector. The goal of the SVM radial algorithm is to find the best a radial hyperplane to split predictors. It has a hyperparameter cost, which controls how much we allow the variables to violate the decision boundary. On each side of the radial hyperplane, it attempts to maximize the margin, which is the space between the hyperplane and the closest support vector. A support vector is a data point on the edge of the margin. I think this SVM is very good for data which can be easily split into radial sections or data which is difficult to split using a linear or polynomial hyperplane. The hyperplane can also utilize multiple dimensions.

1.2 CLASSIFICATION

1.2.1 Linear

Linear SVM utilizes a linear decision boundary, which is defined by a vector. The goal of the SVM linear algorithm is to find the best a linear hyperplane to split the classes, can be used for multi-class classification. It has a hyperparameter cost, which controls how much we allow the variables to violate the decision boundary. On each side of the linear hyperplane, it attempts to maximize the margin, which is the space between the hyperplane and the closest support vector. A support vector is a data point on the edge of the margin. I think this SVM is very good for data which can be easily split into linear sections. The hyperplane can also utilize multiple dimensions.

1.2.2 Polynomial

Polynomial SVM utilizes a polynomial decision boundary, which is defined by a vector. The goal of the SVM polynomial algorithm is to find the best a polynomial hyperplane to split the classes, can be used for multi-class classification. It has a hyperparameter cost, which controls how much we allow the variables to violate the decision boundary. On each side of the polynomial hyperplane, it attempts to maximize the margin, which is the space between the hyperplane and the closest support vector. A support vector is a data point on the edge of the margin. I think this SVM is very good for data which can be easily split into polynomial sections. The hyperplane can also utilize multiple dimensions.

1.2.3 Radial

Polynomial SVM utilizes a radial decision boundary (circle based), which is defined by a vector. The goal of the SVM radial algorithm is to find the best a radial hyperplane to split the classes, can be used for multi-class classification. It has a hyperparameter cost, which controls how much we allow the variables to violate the decision boundary. On each side of the radial hyperplane, it attempts to maximize the margin, which is the space between the hyperplane and the closest support vector. A support vector is a data point on the edge of the margin. I think this SVM is very good for data which can be easily split into radial sections or data which is difficult to split using a linear or polynomial hyperplane. The hyperplane can also utilize multiple dimensions.

2 OTHER ENSEMBLES

2.1.1 Random Forest

Random forest works by repeatedly samples data to try and overcome variance in a dataset. Random forest keeps trees from choosing the same predictors to branch off, reducing repetition and allowing the program to find paths that might outperform trees using the strongest predictors repeatedly. Results are averages and the “majority vote” or the most common classification. Each tree uses a different data sample and different subsets.

2.1.2 XGBoost

The XGBoost runs nearly ten times faster than other boost methods, running faster than a Python version. The computational part of this package was written using C++ and takes advantage of multithreading on a computer. All data needs to be converted into integer values before running XGBoost. It really tends to shine with high nrounds.

2.1.3 fastAdaBoost

fastAdaBoost useds C++ code to run a lot faster than R base libraries. It uses Freund and Schapire's Adaboost.M1 algorithm and Zhu's SAMME algorithm. This algorithm trains a set of learners one by one, giving branches with the most error heavier weights so that the next learner can overcome that deficit in the next iteration and the correct branches are given smaller and smaller weights.

2.1.4 superLearner

The superLearner algorithm uses cross-validation to create the optimal weighted average of the models. It then uses these averages to make predictions about future data. Because it uses different models to make predictions, it is generally more accurate than more rudimentary forms of classification. To evaluate superLearner we validate it with a testing dataset that it hasn't seen before and judge its accuracy.