CS 4375.003 Intro to Machine Learning

Dr. Karen Mazidi

Students:

● Simon Kim - sxk190106

● Zuhayr Ali – zxa180002

a. how SVM works, and how SVM kernels work, your impression of the strengths and

weaknesses of SVM

how SVM works

SVM analyzes data using support vector and hyperplane. Hyperplane sets the boundary of the data not to be biased to either side, and it is ideal to set the hyperplane so that the distance value between hyperplane and support vector, margin, is maximum. An ideal svm can accurately classify new data by maximizing the margin.

how SVM kernels work

There is also a way to simply use hyperplane as a straight line like linear SVM, but this method is difficult to respond to various data. In order to classify datasets that are difficult to select linear hyperplanes, the original dimension can be changed using basis function to make linear hyperplanes easier to use. The kernel can not only save the effort of defining the basis function one by one, but can also drastically reduce the amount of computation. In this practice, in addition to linear kernel, polymorphic and radial kernel were used, and the characteristics are as follows.

The more dimension of the kernel increases, the more various hyperplanes can be created under polynomial kernel. For example, polynomial kernel uses (x,y) -> (xy20.5,x2,yˆ2) to calculate 2D data as 3D data.

Radial kernel enables classification by taking it one level higher for high-dimensional datasets that cannot be classified by a polymeric kernel.

Strength and weakness

There are no clear classification criteria such as the genre of movies, but it can be classified if there is data corresponding to the existing classified genre. SVMs are less likely to overfitting and are useful for classification and numerical prediction because of the low impact of error data. Also, in personal experience, I felt that SVM was very easy to use.

However, the disadvantage of SVM is that several combination tests are required. Various variations on the kernel and model are needed to find the optimal model, and as shown in this task, when there are many input data sets, this variation takes a long time, slowing down the learning speed. In addition, there is a problem that it is difficult to visually represent when learning about a large dimension. For example, visualization of three dimensions is possible, but it is impossible to analyze this data visually if there are more than four elements. That is, although it is possible to interpret complex forms of data, it is difficult to interpret intuitively.

b. how Random Forest works, how the other 2 algorithms you used work compared to the

simple decision tree, your impression of the strengths and weaknesses of these

ensemble techniques

how random forest works

Random forests work by generating a forest of de-correlated decision trees that at each split pick a predictor from a randomized subset of all predictors in the dataset. Each tree uses a distinct subset of data from any other tree. Then the resulting trees are ranked to determine the best model.

Strength and weakness of random forest

Random forest is a simple evolution of the decision tree relative to other ensemble methods. By sampling distinct subsets of both data and predictors, random forests can mitigate the shortcoming of decision trees that results from their greedy approach, discovering better models than those that result from picking the strongest predictor at every split. However the subsetting approach limits each tree in the forest in how they are able to predict and so can miss better yet models by the circumstance of the subsets.

how XGBoost works

XGBoost uses the concept of boosting. Boosting is the sequential generation of decision trees that are given weights corresponding to their predictions and mispredictions. The weights are used to inform later trees to pick what strategies work best. XGBoost prepares the data by converting the predictors and targets into matrices, and trees are refined to be represented as vectors of leaf scores. XGBoost works on mutliple iterations, and each iteration uses a new function that uses the weights from previous iterations to improve the accuracy and correlation of the model.

Strength and weakness of XGBoost

Based on how XGBoost performed on the “diamonds” dataset, it’s difficult to come up with weaknesses. XGBoost uses the weights principle of boosting and so learns to adjust its functions to produce strong positive weights. It can adapt where decision trees and random forests cannot. It is also well optimized and so produces a model in a very quick time for its level of accuracy. If there is something to say for weaknesses, it’s that the data needs extra processing to work in XGBoost and might also need dozens of iterations to produce an accurate model.

How light GBM works

Light GBM stands for light gradient boosting machine. Its design shares many similarities with XGBoost. Like XGBoost, light GBM uses the concept of boosting and prepares the data by converting to matrices. This time all elements in the matrices are numeric. It also performs multiple iterations to refine the model. However, light GBM uses different algorithms to optimize the speed and memory usage and has extra abilities, including the ability to leverage a GPU.

Strength and weakness of light GBM

Light GBM has some of the same strengths as XGBoost. It is also able to adapt through the weights principle, learning over iterations to improve its accuracy. This is again a leg up over decision trees and random forests. Like XGBoost it also produces models in quick time, and the ability to use a GPU in training gives the method a massive source of computational power restricted from other methods. However, light GBM also requires extra processing on the data, and the results of produced for the “diamonds” did not have the accuracy of XGBoost.

References

[1] “microsoft/LightGBM,” *GitHub*, Apr. 25, 2020. https://github.com/microsoft/LightGBM

‌