HW5, Competition using SVM - Big Data

20170735
컴퓨터공학과
이승준

제가 강의를 듣고, 과제를 하며 깨달은 점에 대해 정리했습니다.

Support vector machine 알고리즘을 이용했다. 이 알고리즘에서 kernel function은 rbf, poly, 등등 많은 것들 중 선택할 수 있었는데, kernel function은 Weight 벡터들과 Data 벡터들을 higher dimension으로 projection 시키고, linearly separable 하게 만들고 거기서 내적한 것과 같은 효과를 낼 수 있게 해서 시간을 줄여주는 함수이다. 기본적으로 이것이 가능한 이유는, 높은 차원이든 낮은 차원이든 벡터의 내적값은 스칼라이기 때문이다. Kernel function을 통해, 우리는 실제로 projection, 내적 등을 하지 않아 시간을 줄이면서도 같은 효과를 낼 수 있다.
C 값은 support vector 들에 대한 weight, 즉 alpha에 대한 constraint 인데, C >= alpha >= 0 이다. 그리고 margin과 slack 사이의 trade-off variable 이다. 그래서, C를 높이는 것은, margin을 최대한 더 빡빡하게 줄여야 한다는 것이다. 이는 조금 더 hard boundary 를 요구하는 것이며, 그래서 linearly separable 한 데이터들에 적합하다. 즉, 조금 더 분류가 잘 되어있는 데이터들에 대해서는 C가 높으면 좋은 결과가 나올 것이다. 그런데 C를 낮게 설정하면, 반대로, 상대적으로 margin을 최소화하는 게 덜 빡빡하다. 즉, boundary 가 조금 부드러울 수 있는 것이다. 막 섞여 있는 데이터들에 대해서는 C값이 작은 것이 더 좋은 prediction을 보일 수 있을 것이다. 그리고 kaggle competition을 통한 파이썬 코드를 짜는 중에 C가 음수거나, 0이면 에러가 뜨는데, 이는 C >= alpha >= 0 이 조건이 있기 때문일 것이다. C가 음수가 되면 말이 안되고, 0이라면 alpha는 모든 데이터에 대해 무조건 0이 되어야 하기 때문에 의미가 없기 때문이다.

1. Binary Classification using SVM

먼저, 이용한 SVM model은 sklearn.svm.SVC 입니다.
HW4와 거의 비슷하게 전처리 했습니다. 각 Categorical string data feature 들에 대해서, y에 가장 큰 영향을 주는 것부터 오름차순으로 0 부터 assign 했습니다. 그리고, ‘balance’, ‘duration’ 등의 numerical data들은 seaborn library의 box plot 메서드를 이용해 outlier 들을 파악하고,
df2.loc[df2[‘balance']>=35000,'balance']=35000 과 같은 코드로 outlier들을 조정해줬습니다. 그리고, duration, campaign, previous 등등의 outlier들을 당겨주고, day는 y에 큰 영향이 없다고 생각해 column을 drop 했습니다. 그리고, standard scaler 를 이용해 스케일 했습니다.

model_selection.train_test_split 을 이용해 test 했습니다. 그 후,
clf=SVC(C=0.3,kernel='poly',cache_size=2048,gamma="auto", random_state=0, degree=4, tol=0.001, class_weight=‘balanced') 를 classifier parameter 로 이용했고, train_test_split 으로 테스트한 결과 0.5663801337153773 가 f1 score로 나왔습니다.

Public Score : 0.58423

2. Crime Category Classification using SVM

1번과 거의 비슷하게 preprocessing 했습니다. 그런데, 날짜와 시간을 컬럼을 나눴고, 년, 월, 일에 각각의 가중치를 둬서 Dates 컬럼을 만들었고, 시간과 분에 각각 가중치를 줘서 시간 컬럼을 만들었습니다. 그리고 ‘X’ 와 ‘Y’ 를 outlier 들을 당겨주었습니다. StandardScaler를 이용해 스케일 했습니다. 그리고,
model_selection.train_test_split 을 이용해 테스트했습니다.
clf=SVC(C=0.3,kernel='poly',cache_size=2048,gamma="auto", random_state=0, degree=4, tol=0.01, class_weight=‘balanced’) 를 이용했는데, max_iter를 3정도로 하고, parameter 들을 조절하면서 parameter 들을 결정했습니다.

Public Score : 2.60117