Machine Learning- CS 60050

Assignment 5:
Support Vector Machines

Avinab Saha, 15EC10071

24th October, 2018

Methodology:

The data is loaded from the data file and split into training set and test sets in 70:30 ratio.

We repeat the experiment twice, once without normalizing the data, once by normalizing the data.

Details of Packages used:

1. **Numpy** - For handling Matrix related operations
2. **Sklearn**
   a. **from sklearn.model_selection import train_test_split**

      This function is used for splitting the entire dataset into train and test subsets.

   b. **from sklearn.preprocessing import normalize**

      This function is used for normalizing all the features of the dataset to prevent the influence a particular feature in the training process.

   c. **from sklearn.svm import SVC**

      This function is used to import the built in SVM function of the Sklearn library. We need to different values of hyperparameter ‘C’ for Linear, Quadratic and RBF Kernels.

Experiment A: Without Normalization

Linear Kernel:

Value of ‘C’	Training Set Accuracy	Test Set Accuracy
0.0001	75.52%	76.53%
0.0005	89.90%	92.68%
0.0010	95.99%	96.81%
0.0015	98.85%	98.55%
0.0020	99.59%	99.56%
0.0025	99.84%	99.85%
0.0030	99.90%	100%
0.0035	99.60%	100%
0.0040	99.96%	100%
0.0043	100%	100%

Quadratic Kernel:

Value of ‘C’	Training Set Accuracy	Test Set Accuracy
0.001	95.49%	95.36%
0.005	97.70%%	97.53%
0.010	98.07%	99.05%
0.015	98.63%	99.20%
0.020	98.72%	99.27%
0.025	98.85%	99.34%
0.030	98.97%	99.42%
0.035	99.09%	99.42%
0.045	99.19%	99.42%
0.050	99.31%	99.42%

RBF Kernel:

Value of ‘C’	Training Set Accuracy	Test Set Accuracy
0.1	77.29%	68.35%
0.2	83.69%	78.27%
0.5	92.79%	83.05%
1.0	96.77%	86.38%
1.5	98.10%	87.61%
2.0	98.69%	88.19%
3.0	99.56%	88.77%
4.0	99.84%	85.03%
5.0	99.93%	88.84%
6.0	100%	89.21%
7.0	100%	89.28%
8.0	100%	89.35%
9.0	100%	89.13%
10.0	100%	89.28%
>10.5	100%	89.42%

Experiment B: With Normalization

Importance of Normalization of features before training

The answer to this depends on what similarity/distance function you plan to use (in SVMs). If it's simple Euclidean distance, then if you don't normalize your data you are unwittingly giving some features more importance than others.

For example, if your first dimension ranges from 0-10, and second dimension from 0-1, a difference of 1 in the first dimension (just a tenth of the range) contributes as much in the distance computation as two wildly different values in the second dimension (0 and 1). So by doing this, you're exaggerating small differences in the first dimension.

Linear Kernel:

Value of 'C'	Training Set Accuracy	Test Set Accuracy
0.0001	61.11%	59.37%
0.0002	61.11%	59.37%
0.0003	61.11%	59.37%
0.0004	61.11%	59.37%
0.0005	61.11%	59.37%
0.0006	61.11%	59.37%
0.0007	61.11%	59.37%
0.0008	99.75%	99.85%
0.0009	100%	100%
>0.001	100%	100%

Quadratic Kernel:

Value of ‘C’	Training Set Accuracy	Test Set Accuracy
0.1-1.9	61.11%	59.37%
2.0	61.14%	59.37%
2.1	61.30%	59.52%
2.2	62.42%	60.75%
2.3	68.91%	68.93%
2.4	80.90%	81.89%
2.5	90.21%	91.52%
2.6	95.99%	97.32%
2.7	100%	100%

RBF Kernel:

Value of ‘C’	Training Set Accuracy	Test Set Accuracy
0.0010-0.0210	61.11%	59.37%
0.0220	61.24%	59.44%
0.0222	61.35%	59.52%
0.0225	63.54%	61.98%
0.0227	69.81%	69.44%
0.0228	74.59%	75.74%
0.0229	79.19%	80.73%
0.0230	84.00%	85.03%
0.0231	88.88%	90.65%
0.0232	93.10%	94.35%
0.0233	95.59%	97.32%
0.0234	98.01%	98.91%
0.0236	99.78%	100%
>0.024	100%	100%