Assignment 2

CSE 343/543 : Machine Learning Due: 11:59PM, Sept. 25,2017

Note: Please complete programming as well as theory component. Submissions for predictions need to be submitted on Kaggle.

Note: Using any built-in function other than sklearn's *.fit()* **and** *t-sne* **is not allowed (for parts other than the Kaggle competition). You are to write your own functions, even for** *.predict()***.**

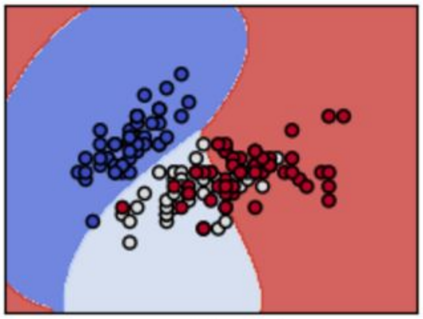
Submission (on Backpack) : Code + theory.pdf (a legible copy of scanned answers to theory questions) + report.pdf (a report explaining all your codes, plots and approaches)

Programming [75 marks + 25 marks bonus]

Note that ***1/3rd*** *of the programming marks are on the basis of ranking in the Kaggle competition.*

Exploring data sets and kernels:

1. **You are given five different 2-dimensional datasets. Some datasets are noisy, unbalanced, etc. Explore the datasets, plot them and write your observations and findings of the datasets.**
2. **For each dataset, write a kernel (if required) to make them linearly separable. Plot the datasets with decision boundaries corresponding to those kernels; something like this:**



   Explain the choice of kernels.
3. **Use outlier removal techniques to remove outliers in the datasets. Plot the outlier-removed datasets also.**

SVM

1. **Implement Soft margin SVM with linear kernel. Use the built in** *sklearn's* **binary classifier. Use the binary classifiers to implement a multi-class classifier for M classes. Test this on the above five datasets and the datasets of the previous assignment. For each dataset, do an analysis on the performance of the model, the choice of parameters and preprocessing.**

2. **Repeat the above part using an RBF kernel instead of a linear one.**

   Note: For converting the binary classifier to multiclass, you need to use *One vs Rest* **Classifier and** *One-vs-One* **Classifier You need to implement them yourselves and write an analysis (based on running time and performance metric, accuracy in this case) on the two techniques. You can only use the** *.fit()* **function of the** *sklearn's* **SVM module. You would have to write all the other functions including predict function (overall predict for the M-class classifier, not predict of the individual SVMs) yourself.**

   The analysis part is very important. Thus, it is important to make a good report with the findings, numbers and plots.

   In the analyses, mention the following :
   - **What choice of hyperparameters is useful for which dataset and why? How did you choose the hyperparameters?**
   - **Comparison of the SVM models with the models in the previous assignment and mention in which cases which models should be preferred. Which evaluation metric would you use to compare?**
   - **Plot the support vectors and the margin separating hyperplane. (For** *n* **dimensional datasets, reduce it to 2D via t-SNE or any other technique)**
   - **The plots should be easy to interpret and should make sense.**
   - **Plot confusion matrices and ROC curves.**

3. **You are given a dataset (uploaded on Kaggle). You are required to use and train any of the SVM based classifiers that have been covered in class so far on that dataset, and then make a submission (along with your name and roll number). Use any implementation of SVM (linear/nonlinear) you want to, with any set of parameters.**
   - **You would need to preprocess the data to filter out outliers, irrelevant or redundant indices, etc.**
   - **You may need to balance the data by upsampling/downsampling during training.**
   - **You can use any technique for extracting features.**
   - **You are allowed to use external libraries.**
   - **There are marks for accuracy (on the basis of Kaggle rankings) , so you will need to perform data preprocessing and find good hyperparameters.**

- Make a report on the dataset, preprocessing and the techniques used

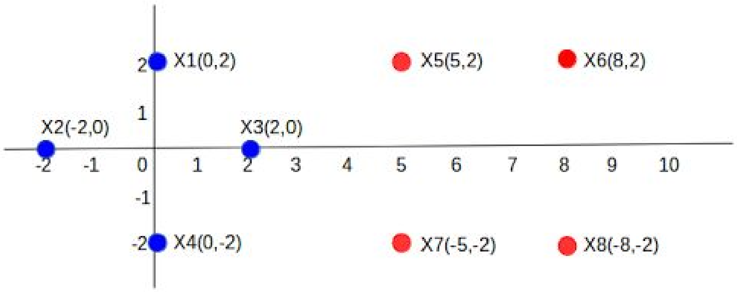
4. Bonus [25 marks]: You are required to implement kernelized PCA (KPCA) using an RBF Kernel with k-nearest neighbor (kNN) classification. Owing to the simplicity of the technique, you are required to implement KPCA and k-nearest neighbor classification yourself. You would need to follow the steps below:
   a. Use 150 randomly selected samples per class from the the 5 2-D training datasets given in this assignment.
   b. Construct the kernel matrix, which in this case will be N × N, center it, and perform KPCA , where N is the number of train examples in the split.
   c. Project the validation data onto the PCs in the kernelized feature space, followed by kNN classification with k = 3 , 5 , 10. Vary k and plot and report the results.
   d. Use a five-fold cross-validation scheme with grid search to estimate the γ parameter for an RBF kernel. Use the classification accuracy to select the best value of γ .
   e. Write an analysis (in terms of accuracy and computational effort) on how kernelized PCA works and for what kind of datasets it works the best.

Theory Questions [25 marks]

1. Consider the RBF kernel: it can map the given data into a higher dimensional space, with possibly infinite dimensions. There are two ways to look at it: one way is that every data points gets its own dimension, which would lead to overfitting. However, in reality, this generally does not happen. Why?

2. Show that, irrespective of the dimensionality of the data space, a data set consisting of just two data points, one from each class, is sufficient to determine the location of the maximum-margin hyperplane. You are expected to give a mathematical proof.

3. Consider the hard-margin SVM. Find the maximum margin for the following case. How does the size of maximum margin changes if we remove *X7* from the training dataset.



Red Points (class-1), Blue Points (class-2)

4. **Can you model the XOR operator using an SVM? Justify.**