

Coding Assignment 2

Due: 2015/11/11 23:59:59

- Given a dataset with 1000 instances, your assignment is to implement a parametric classifier. We'll test your classifier with another dataset and grade your program based on its performances (precision, recall) and your report.
- Specification
 - Follow the OOP (Object-Oriented Programming) paradigm in Matlab.
 - Implement a class **ParametricClassifier** with:
 - * A static method: **train(X, y) → ParametricClassifier**
 - $\mathbf{X} \in R^{N \times d}$ is a matrix of training instances, where N is the number of instances and d is the number of features.
 - $\mathbf{y} \in \{-1, +1\}^N$ is the label vector.
 - * A instance method: **predict(X) → y**
 - $\mathbf{X} \in R^{M \times d}$ is a matrix of testing instances, where M is the number of instances and d is the number of features.
 - Returns $\mathbf{y} \in \{-1, +1\}^M$, the predicted labels.
 - You are free to use any data preprocessing techniques.
 - You may use any model selection and ensembling techniques to find the model with desired performance.
- Submission Requirements:
 - A report (REPORT.*) describing:
 - * How you implement the classifier.
 - * How you tune your model.
 - * Anything worth mentioning.
 - The code:
 - * **ParametricClassifier.m**
 - * You have to follow the specs (as described above) strictly.
 - Put your work under package **model.classifier** (i.e. under directory **+model/+classifier**).
 - Follow the submission guide (see **README.md** for details).
- Hint:
 - You may try to plot the data with less dimensions.
- High Performance Rewards:
 - You'll get extra credits if the performance of your classifier is top 20% of the class (this will contribute to 10% of your final grade).