

# Final Project Description Report

**Optimus Perceptron**    Teammates: Saurabh Daga, Karan Pradhan, Sibi Vijaykumar

## Algorithm 1

This algorithm is the one we have used for the submission in the leaderboard folder. It uses Linear SVM provided by the “Liblinear” package. Also, this algorithm separates the data with respect to cities as a preprocessing step. The file corresponding to this algorithm is “leaderboard\_algo.m”. It has other dependencies which are included in the submission. We have used a script called “addPathScript.m” which makes the dependencies for liblinear. However, all these scripts have been invoked internally in the “leaderboard\_algo.m”. Thus running “leaderboard\_algo.m” will train and predict and the predictions for the test set will be stored in a text file called “submit.txt”. Also, the submission includes “searchLiblinear.m” which is invoked in the “leaderboard\_algo.m” file, is used for cross validation by performing a full grid search on the linear SVM parameters. This algorithm is an example of the **discriminative method**.

## Algorithm 2

This algorithm uses linear SVM again on data separated with respect to cities but in this case, we have divided the city 4 data into 2 clusters. We have clustered the data points belonging only to the fourth city using Gaussian Mixture Models which is an example of a **Generative Method**. Running the file “generative\_algo.m” trains the data and stores the predictions in “submit\_generative.txt”. The script “addPathScript.m” is invoked the “generative\_algo.m” for building the Liblinear package. Also, “searchLiblinear.m” is used for cross validation by performing a full grid search on the linear SVM parameters.

## Algorithm 3

This algorithm uses **SVM based on a precomputed kernel**. However, we have used PCA for reducing the dimensionality of the data for speeding up SVM. We have used the exponential kernel which is a RBF very similar to the gaussian kernel. The file “svm\_custom\_kernel\_algo.m” is used for training and storing the predictions in “submit\_custom\_kernel.txt”. This file again invokes “addPathScript.m” for building the libsvm package. The model has been cross validated on the cost parameter C in SVM.

## Algorithm 4

This algorithm uses kernel regression which is an example of **instance based method**. Running “instance\_based\_algo.m” trains the data and stores the prediction in “submit\_instance\_based.txt”. The model for this method uses 1000 training data points and has K in the kernel regression to be set as 20 which we observed after 10 folds cross validation. Also, for achieving speed, we have implemented **semi-supervised dimensionality reduction** using PCA.<sup>1</sup>

---

<sup>1</sup>All the code files assume that data is in a directory one level hierarchial above and inside the “data” folder.