

# ELL784/AIP701: Assignment 2

Last updated: 22-09-2022, 12:09pm

## Deadlines

1. Deadline for final submission of assignment report and all code: **12 October 2022, 11:59 PM.**

## Instructions

1. While you are free to discuss all aspects of the assignment with your classmates or others, your code (for those registered for AIP701) and your results and report must be entirely your own. We will be checking for any copying, and this will be treated as plagiarism and dealt with accordingly. In case of any doubts in this regard, please ask us.
2. **For AIP701: You are free to use any platform to write and test your code. However, if you are using Python, please make sure you submit .py files for evaluation during the viva.**
3. All students should prepare a report, compiling all your results and your interpretation of them, along with your overall conclusions. In particular, you should attempt to answer all of the questions posed in the respective parts of the assignment below. Any graphs or other visualisations should also be included therein.
4. A single tar/zip file containing the source code (if applicable), report, and any other relevant files has to be submitted via Moodle. The zip/tar file should be structured as per the below guidelines:
  - Upon deflating all submission files should be under a directory with the student's entry number. E.g., if a student's entry number is 20XXCSXX999 then the tarball/zip submission should be named 20XXCSXX999.zip or equivalent, and upon deflating **all contained files** should be under a directory named ./20XXCSXX999 only.
5. The schedule for demos/vivas will be announced by your respective TAs, in advance. If for any reason you cannot attend in your scheduled slot, you must arrange for an alternative slot with your TA well in advance. Last-minute requests for rescheduling will normally not be accepted.

## Weightage

*For ELL784:* This assignment is worth 5 marks or 5% of your overall course grade; a part-wise break-up is provided below.

*For AIP701:* The coding part of this assignment will additionally be worth 5 marks, which will amount to 25% of your grade for that course.

## Objective

To experiment with the use of SVMs for both binary and multiclass classification problems, and understand the effects of varying various hyperparameters therein.

## Part 1 (ELL784: 3 marks; AIP701: 5 marks)

### Data

We provide a personalised input file that contains 3,000 labeled data points, with 25 features each. You should download your file from [https://web.iitd.ac.in/~sumeet/A2/<EntryID>/feat\\_25.csv](https://web.iitd.ac.in/~sumeet/A2/<EntryID>/feat_25.csv) (for example, [https://web.iitd.ac.in/~sumeet/A2/2017CRZ8638/feat\\_25.csv](https://web.iitd.ac.in/~sumeet/A2/2017CRZ8638/feat_25.csv)). This file contains 3,000 rows, with each row corresponding to a data point. Each row has 26 comma-separated values; the first 25 are the values of the features, and the last is the class label for that data point (there are 10 classes, denoted by the labels 0 to 9). Each data point is actually a low-dimensional representation of an image.

### Methodology

Your task is to try and learn an SVM classifier for these images, using just the given features, and thereby also to assess the usefulness of the different features. Here is how you should proceed:

1. Familiarise yourself with a standard SVM library. The recommended one is [LIBSVM](#), which is available for MATLAB, Python, and many other languages. Figure out how you can set various hyperparameters, such as the value of  $C$  for the soft-margin SVM, the choice of kernel function, and the kernel parameters (if any). You may wish to play with a simple toy data set to get a feel for using the library, before you move on to the actual data for this assignment.

**(AIP701 only)** As an alternative to LIBSVM or other standard library, we would like you to attempt to train an SVM directly using a convex optimisation package. We recommend [CVX](#). You don't need to do this for all hyperparameter settings; you can first tune the hyperparameters using LIBSVM etc., and then just use the tuned values to train using CVX. But you should train at least one linear and non-linear SVM each for binary classification in this fashion, and report how the results and support vectors obtained compare to those using a standard SVM library. In order to use CVX to train an SVM, you will need to figure out how to take the SVM dual problem and express it in a form which can be fed into the convex optimisation package. You should be able to clearly explain how you did so, and also submit whatever wrapper code you write to take the given data and convert it into the form where it can be fed into CVX; as well as the code which takes the CVX output and uses it to construct the actual classifier to be run on test data.

2. *Binary classification*: Choose just 2 out of the 10 classes in your data, and train an SVM. Leave some data aside for validation, or ideally, use cross-validation. Study the effects of changing the different hyperparameter values, including the type of kernel function being used. How do they affect the accuracy? Can you distinguish cases of overfitting, underfitting, and good fitting? Also try using only the first 10 features, instead of all 25, and compare the results in the two cases. Repeat this exercise for at least two more pairs of classes out of the 10 given to you. Do you consistently get the best results for the same hyperparameter settings, or does it vary a lot depending on which pair of classes you're looking at? Also, try visualising your data using just the first 2 features. Does this help you get some sense of which pairs of classes are harder to distinguish between, or which specific data points tend to be misclassified?
3. *Multiclass classification*: Now we would like to train a classifier for all 10 classes (AIP701 students: here you can just use a standard SVM library, no need to use the convex optimisation package). Figure out what method(s) your chosen library uses for multiclass classification. For instance, LIBSVM uses *one-versus-one* (see Bishop p. 183). Based on your choice, build a classifier for all the classes, and evaluate using validation or cross-validation. Again, study the effects of changing the various hyperparameters and kernel function. Try and finetune them to obtain the best possible performance. How do these tuned values compare to what you obtained in the binary classification setting? If there are major differences, what might be the reason? Also, try training the multiclass classifier using only the first 10 features, instead of all 25. How does this affect your results? What does this tell you about the usefulness of the different features?

## Part 2 (ELL784: 2 marks)

Similar to the previous leaderboard exercise in Assignment 1, we provide an extended version of the data set for the leaderboard exercise in Assignment 2. This data set has 8,000 instances in the same type of 25-dimensional feature space (compared to the 3,000 instances in Part 1), and you are required to train a multiclass SVM model for predicting the labels on a target set of 2,000 instances (whose labels are hidden from you).

The files provided are the following:

1. <https://web.iitd.ac.in/~sumeet/A2/train.csv>, a training set of 8,000 samples of 25 features (similar to Part 1), followed by the target label for each sample. Hence, the training set has dimensionality  $8000 \times 26$ .
2. <https://web.iitd.ac.in/~sumeet/A2/test.csv>, a target set of 2,000 samples, on which you will have to predict the labels using your multiclass SVM model. The target set has dimensionality  $2000 \times 25$ .

Similar to the previous assignment, you are expected to experiment with hyperparameter tuning, model and feature selection and cross-validation to get an optimal accuracy score, which will be evaluated on the leaderboard (link to be shared separately via Piazza).