

CSE471: Statistical Methods in AI -- Spring 2016

Assignment 2: Linear Discriminant Functions

DUE: Before 12midnight on 01 Feb 2016

INSTRUCTIONS:

- i. You may do the assignment in Matlab/Octave, R, Python, C/C++ or Java.
- ii. You need to upload pdf files in the Course Portal. One file should contain your answers, results and analysis. A separate file should contain code you have written and its sample output.
- iii. At the top-right of the first page of your submission, include the assignment number, your name and roll number.
- iv. **IMPORTANT:** Make sure that the assignment that you submit is your own work. *Do not copy any part from any source* including your friends, seniors or the internet. Any breach of this rule could result in serious actions including an **F grade** in the course.
- v. Your grade will depend on the correctness of answers and output. In addition, due consideration will be given to the clarity and details of your answers and the legibility and structure of your code.

Preamble:

The idea of this assignment is to explore the material presented as part of Chapter 5 on constructing Linear Discriminant Functions based on various approaches such as Perceptron criterion function and Mean Squared Error.

Theory Question:

1. Prove that the **single-sample perceptron algorithm will always converge to a solution**, if one exists (List out all the assumptions, make each step explicit and mathematically precise, but not be verbose! Do not copy directly from the textbook).

Practical Exercises:

The data set to be used for the exercises given below is the following sample set comprising a two-class problem.

$$\omega_1 = [(1; 6); (7; 2); (8; 9); (9; 9); (4; 8); (8; 5)]$$

$$\omega_2 = [(2; 1); (3; 3); (2; 4); (7; 1); (1; 3); (5; 2)]$$

Implement the following algorithms:

2. **Single-sample perceptron**
3. **Single-sample perceptron with margin**
4. **Relaxation algorithm with margin**
5. **Widrow-Hoff or Least Mean Squared (LMS) Rule**

- A. In each case, plot the data points in a graph (e.g. Circle: class- ω_1 and Cross: class- ω_2) and also show the weight vector \mathbf{a} learnt from all of the above algorithms in the same graph (labeling clearly to distinguish different solutions).
- B. Create a test set comprising three more data samples (\mathbf{y}_i) for each class and test your implementation by computing for each of the test samples the output (class label) predicted by the respective algorithm. Create a comparison table listing test set accuracies of each of the above algorithms.
- C. Run each of the above algorithms for various initial values of the weight vector, and comment on the dependence of convergence time (run-time) on initialization.
- D. Similarly explore the effect of adding different margins on the final solution as well as on the convergence (run-) time for algorithms (3) and (4).

LMS VS Perceptron

6. Remember the discussion on LMS and Perceptron solutions being different in some cases – can you experiment and construct a training set that makes the solution of LMS rule differ from those of the other. Of course, if the dataset given above already yields different solutions, can you adjust the dataset so that the solutions align!
7. Modify the dataset such that it becomes linearly non-separable (clearly list the changes in the report). Now run algorithms (4) and (5) with suitable stopping criteria. Plot the data points in a graph (e.g. Circle: class- ω_1 and Cross: class- ω_2) and also show the weight vector \mathbf{a} learnt from these two algorithms in the same graph (labeling clearly to distinguish different solutions) and comment on the nature of the solution found in each case.

The report should clearly detail the formulation of the problem (construction of augmented features, weight vector notation, etc.) and specifically list the test set that you created and the results thereof.

As part of the submission include the code for each of the algorithms along with a small report that explains the algorithms, implementation details, the results and their analysis.