IN4320 Machine Learning — Assignment 4

Francisca Mestre - 4930142

April 2019

1

The Linear Regression Classifier (LRC), when used in a supervised setting, finds the best hyperplane that divides the labeled data points, defined by w and w_o, and then uses this hyperplane as a decision boundary to classify new and unseen test data. In this assignment we consider to semi-supervised variations of the LRC.

The first semi-supervised variation of the LRC is based on the Transductive Support Vector Machine (TSVM) and it is built on the low-density separation assumption. This method is transductive as it directly finds labels for the unlabeled examples instead of computing a decision law. The TSVM-based variation optimizes the cost function

$$\frac{1}{l}\sum_{i=1}^{l}(\langle w, x_i\rangle + w_o - y_i)^2 + \frac{1}{u}\sum_{j=1}^{u}(\langle w, x_j\rangle + w_o - h_j)^2 \tag{1}$$

where l is the number of labeled examples, u is the number of unlabeled examples; x_i and y_i are the labeled examples and their labels, respectively; w and w_o are the parameters that define the hyperplane; x_j and h_j are, respectively, the unlabeled examples and their labels that the optimization computes, with $h_j \in \{-1, 1\}$. This method takes advantage of the unlabeled samples;, it finds the most dense regions of the feature space and finds a hyperplane that separates them best, while labeling the unlabeled examples such that the cost function value is minimal.

The second semi-supervised variation of the LRC was based on how mean and standard deviation are updated in Expectation Maximization (EM) methods, but instead using hyperplanes. Thus, the method finds a decision hyperplane based on the labeled data, by optimizing the cost function

$$\frac{1}{l}\sum_{i=1}^{l}(\langle w, x_i\rangle + w_o - y_i)^2. \tag{2}$$

Then, it uses the hyperplane defined by w and w_o found during training to classify one unlabeled example. With this added example and it's predicted label, the method re-trains (2). With the new found hyperplane, the algorithm predicts one new label, and so on, until labels are predicted for all unlabeled examples.

2

Figure 1 shows the averaged error rates obtained for the three algorithms. The blue line shows the results for the supervised setting, the red line is associated with the first semi-supervised algorithm and the yellow line links to the second semi-supervised algorithm. We can see that the supervised setting results in an error rate between 35%. Besides, the addition of unlabeled samples in the first semi-supervised algorithm does not improve the classification at all. As for the second algorithm, the addition of unlabeled data helps improve the classification task for a relatively small number of unlabeled samples only, i.e., for more than around 100 unlabeled samples the algorithm behaves the same way as the supervised one.

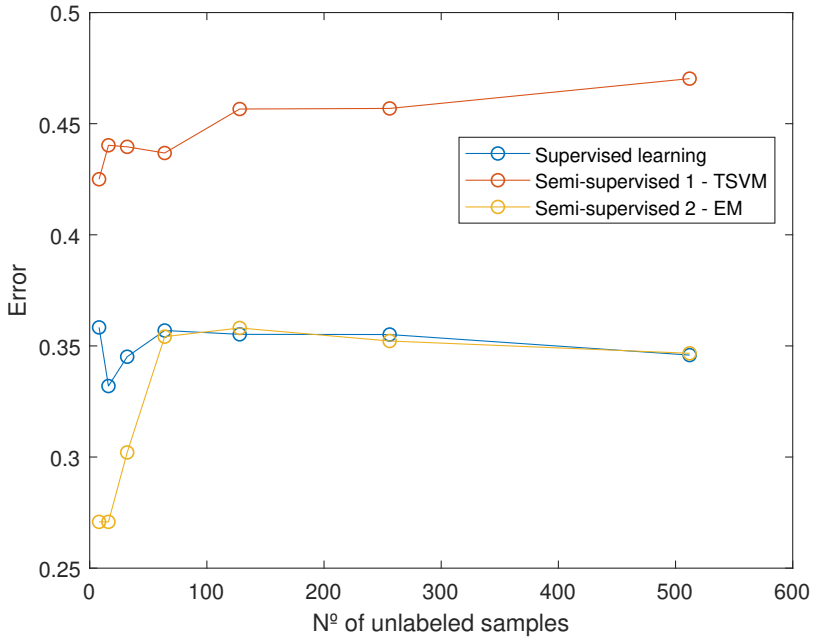

Figure 1: Error rates of the three algorithms applied to the two gaussians problem

3

In Figure 2 we show the expected square loss for each of the algorithms There are clearly no significant changes in between the error rates and the expected square loss, as we are using linear regression based algorithms.

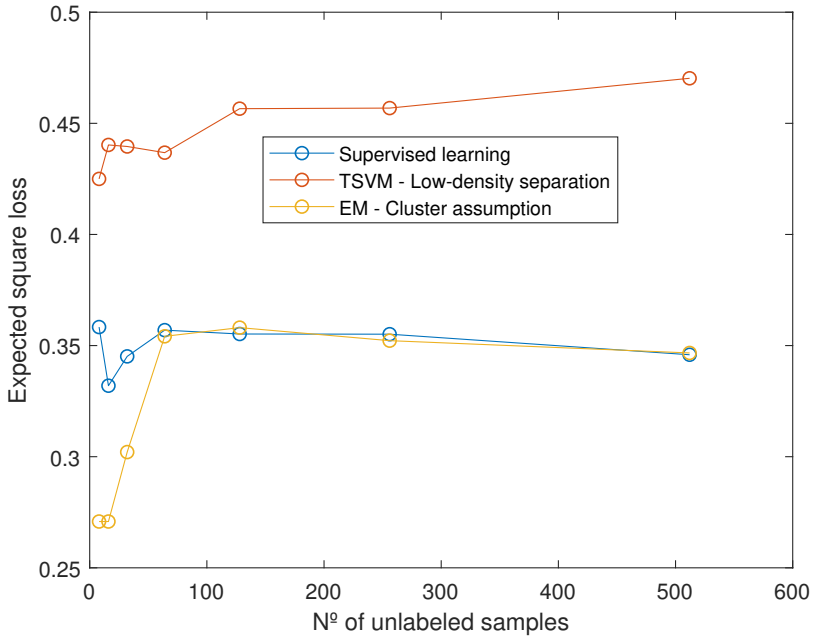

Figure 2: Square loss of the three algorithms applied to the two gaussians problem

4

Now we generate one distributions, to which we will call Task 1, in such a way that the first algorithm fails, originating large error rates, and the second algorithm performs better, relative to the supervised setting. The dataset used is a two-dimensional dataset and Figure 3 provides a good visualization of the set. The error rates obtained for the case with few labeled examples and many unlabeled examples (2 and 512, respectively) are summarized in Table 1.

	Task 1
Supervised setting	18.1%
Semi-supervised setting 1	25.6%
Semi-supervised setting 2	3.9%

Table 1: Error rates of the new classification task

We can note that indeed the second semi-supervised algorithm results in smaller error rates relative to the supervised setting, and the first semi-supervised algorithm performs worse than the supervised LRC. This is expected since most examples from class (+) (or blue) are concentrated around (0,0) so an example close to this point is more likely to be chosen. For class (-) (or red) examples close to (0,2.5) are more likely to be chosen. Then, for most of the time, the supervised algorithm will choose a decision boundary that will be somewhat of a horizontal line at, for example, 0.5.

The first semi-supervised algorithm will optimize both for the decision line and the labels of the unlabeled samples. This results in, more often than not, many of the unlabeled samples

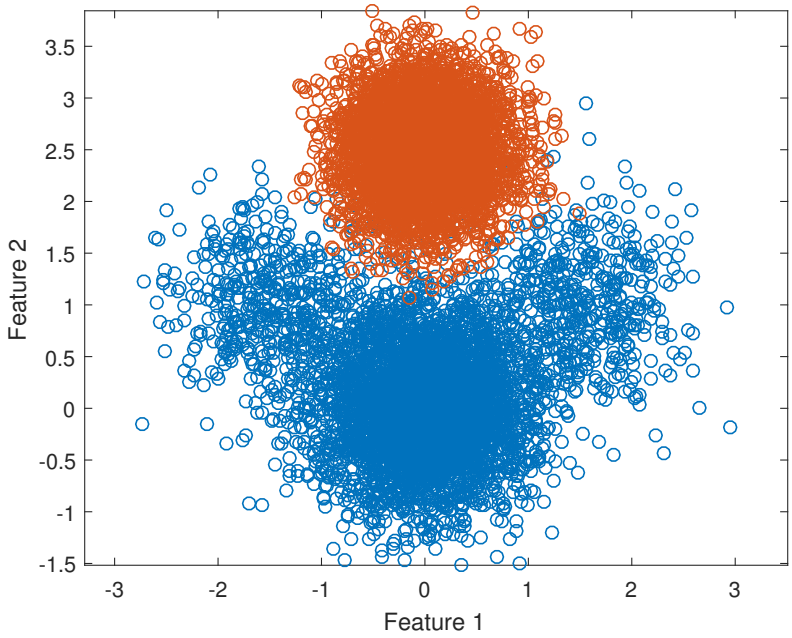

Figure 3: Distribution of dataset used in Task 1 - the blue dots are from class (+) and the red dots are from class (-)

of class (+) being assigned to class (-) as they might be closer to the big agglomerate of that distribution.

The second semi-supervised algorithm updates the decision line as a new unlabeled example is now seen. So for example consider that two labeled samples per class are chosen for training: for class (+) they will very likely be close to (0,0) and for class (-) will likely be close to (0,2.5) and a decision line will be generated. When a new example is seen by the algorithm the line will most likely be pushed up, finally resulting in better classification chances. Thus, on average over many experiments, the second semi-supervised algorithm yields better results.