Locally aware ensemble of Linear Classifiers

Archit Rathore & Yash Gangrade 16th November 2017

1 Introduction

We explore ways to combine multiple linear classifiers based on spatial information. Essentially, we want to devise methods that creates an ensemble of multiple linear classifiers, each of which is good at classifying points in its local vicinity and the final prediction for any point is a combination of prediction by each of the local classifiers weighted by the distance of the point from classifier

2 Literature Review

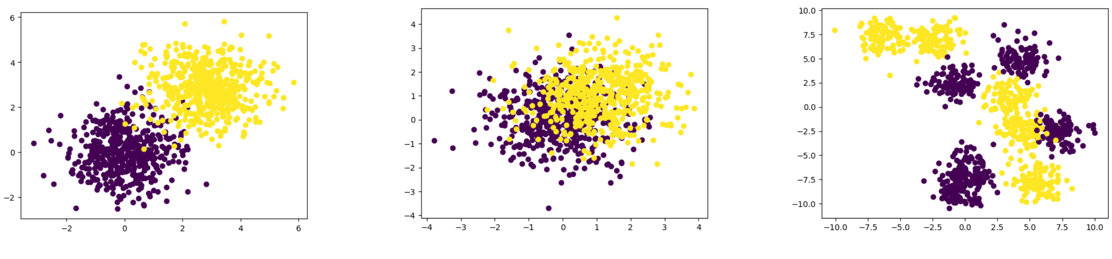
We studied and researched a few papers related to the project. In (Dietterich, 2000)[1], different ensemble methods for constructing a set of classifiers and using voted prediction to classify a new point. Paper by (Duin, 2002)[2] presents a discussion on the use of combination of linear classifiers and how they should be trained combined to produce an optimal result in the case where a single classifier gives a suboptimal result. Paper by (Erdogen, 2010) explores the number of methods of combining the outputs of a set of linear classifiers. One of the effective method of combination is using a linearly weighted combination rule [3]. (Ponti Jr., 2011)[4] discusses different approaches to make an ensemble of classifiers, effectiveness of each method and how it can be improved. Paper by (Tumer, 1996)[5] provides an analytical approach for quantifying the improvements achieved in the classification results by using a combination of linear classifiers.

3 Work done

3.1 Data Generation

We consider three type of spatial data for now.

- **Almost Linearly Separable Data**: Two isotropic normal distributions $\mathcal{N}_1, \mathcal{N}_2$ at a distance of 2σ ($|\mu_1 - \mu_2| = 2\sigma$) have nearly will overlap in nearly 5% of data (95% of points in a normal distribution are contained in a radius of 2σ). We use this fact to generate nearly seperable data.
- **Badly Non Linearly Separable Data**: Following the same procedure as above, this time we are keeping a low separation between the centers of the clusters so that there is sufficient overlap between the clusters to make it badly non linearly separable.
- **Locally Linearly Separable Data**: In this case, the data is nearly linearly separable locally but not as a whole. This is achieved by having points generated by k gaussian distributions with random centres and random labels.



(a) (b) (c)

Figure 1: Different types of Data. (a) Almost Linearly Separable (b) Badly Linearly Inseparable (c)Locally Linearly Separable

3.2 Creating local classifiers and ensembling them

We use Gaussian Mixture Model (GMM) to perform the clustering of the data points and probabilisticaly assign each point to cluster centers. For an observation x and a Gaussian Mixture Model with k clusters denote the cluster probability as $P_x = \{p_1(x), p_2(x), ..., p_k(x)\}$. Then we train k perceptrons where the update rule is $w_k = w_k + p_k(x) \cdot w_k \cdot x \cdot y$. Now, whenever a new observation x' comes in, we pass it through GMM and get the probability distribution. To predict the label of x', we find it's prediction through individual classifiers and then find a single value by incorporating the probability distribution (acts like weights). We have,

$$pred(x) = sign\left(\sum_{i=1}^{k} p_i(x) \cdot (w_i \cdot x)\right)$$

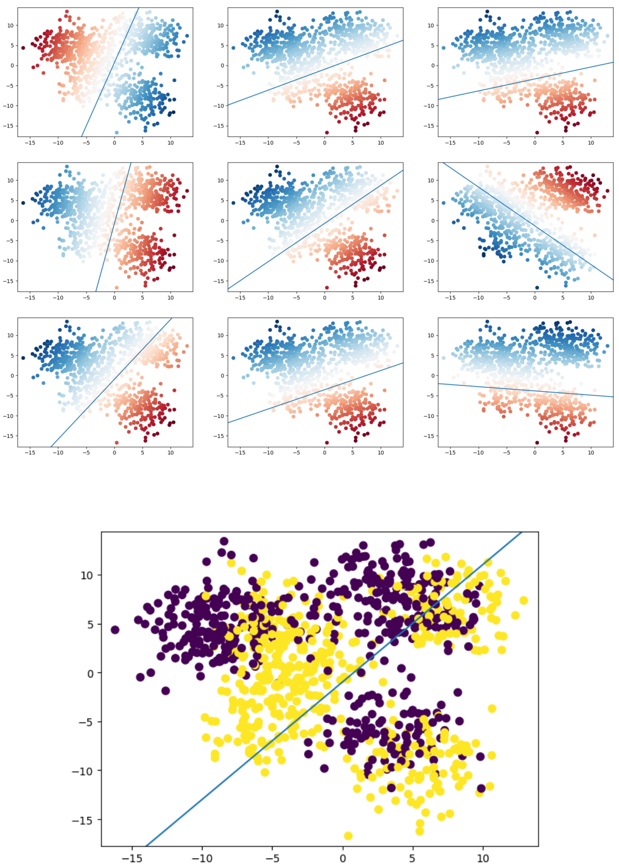


Figure 2: (top) Multiple local classifiers (bottom) Single global classifier

4 Preliminary Results

Type of data (10000 points)	Accuracy of Global Linear Classifier	Accuracy of Ensemble of Local Classifiers
Locally Linearly Separable	58.93%	**78.69%**
Almost Linearly Separable	95.52%	**95.98%**
Badly Linearly Inseperable	49.93%	**59.30%**

5 Plan for rest of the semester

- Analyze cases of failure in such ensembles - what is the robustness of this method under various noise, seperation etc.
- Replace GMM weighing with other methods such as locality sensitive hashing (LSH) during training phase.
- Use SVM, logistic regression etc as linear classifiers instead of perceptron
- Pose learning weights of each classifier for test points as a learning problem itself - use learning to learn to the weights for each point. (Maybe try out some non-linear combinations of linear classification)

6 References

[1] Thomas G Dietterich et al. “Ensemble methods in machine learning”. In: *Multiple classifier systems* 1857 (2000), pp. 1–15.

[2] R. P. W. Duin. “The combining classifier: to train or not to train?” In: *Object recognition supported by user interaction for service robots*. Vol. 2. 2002, 765–770 vol.2. DOI: 10.1109/ICPR.2002.1048415.

[3] Hakan Erdogan and Mehmet Umut Sen. “A unifying framework for learning the linear combiners for classifier ensembles”. In: *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE. 2010, pp. 2985–2988.

[4] Moacir P Ponti Jr. “Combining classifiers: from the creation of ensembles to the decision fusion”. In: *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2011 24th SIBGRAPI Conference on*. IEEE. 2011, pp. 1–10.

[5] Kagan Tumer and Joydeep Ghosh. “Analysis of decision boundaries in linearly combined neural classifiers”. In: *Pattern Recognition* 29.2 (1996), pp. 341–348.