ML Assignment 3

Tarush Goyal — 180050110

September 26, 2020

1 Feature Design For Perceptron

I have achieved 99.5% test accuracies with the following features:

1) Area of the shape : $\frac{\text{number of black points}}{1000}$

2) Number of straight line points : $(\frac{\text{number of points with 1 white neighbour}}{100})^2$

3) Number of diagonal line points : $(\frac{\text{number of points with 2 white neighbour}}{100})^2$

4) Number of stray points : $(\frac{\text{number of points with 3 white neighbour}}{100})^2$

IDEA: The idea is that the shapes of one class are almost **similar** and hence have a fixed linear relation between perimeter and area. However, It is very difficult to calculate perimeter as it is a weighted sum of the 3 types of points i have presented in 2), 3) and 4). So I have kept those 3 points as separate features. They have been **squared** to provide a **dimensional uniformity** with respect to **Area**. Also the **normalising factors** are arbitrary so that all features are in same range

2 Logistic Regression

2.1

(a)

$$P(Y=1|\mathbf{w},\phi(x)) = \frac{exp(\mathbf{w}_1^T\phi(\mathbf{x}))}{exp(\mathbf{w}_1^T\phi(\mathbf{x})) + exp(\mathbf{w}_2^T\phi(\mathbf{x}))} \tag{1}$$

$$P(Y=2|\mathbf{w},\phi(x)) = \frac{exp(\mathbf{w}_2^T\phi(\mathbf{x}))}{exp(\mathbf{w}_1^T\phi(\mathbf{x})) + exp(\mathbf{w}_2^T\phi(\mathbf{x}))} \tag{2}$$

Let $w' = w_1 - w_2$, Then (1) and (2) can be written as :

$$P(Y=1|\mathbf{w},\phi(x)) = \frac{1}{1+exp(\textbf{-w'}^T\phi(\mathbf{x}))} = \sigma(\textbf{-w'}^T\phi(\mathbf{x})) \tag{3}$$

$$P(Y=2|\mathbf{w},\phi(x)) = \frac{exp(\textbf{-w'}^T\phi(\mathbf{x}))}{1+exp(\textbf{-w'}^T\phi(\mathbf{x}))} = 1-\sigma(\textbf{-w'}^T\phi(\mathbf{x})) \tag{4}$$

Now,

$$E(\mathbf{W}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_k^{(i)} log(P(Y=k|\mathbf{w}_k,\phi(\mathbf{x}^{(i)})))$$

$$= -\frac{1}{N}\sum_{i=1}^{N} y_1^{(i)} log(P(Y=1|\mathbf{w}_1, \phi(\mathbf{x}^{(i)}))) + y_2^{(i)} log(P(Y=2|\mathbf{w}_2, \phi(\mathbf{x}^{(i)})))$$

(substituted K as 2)

$$= -\frac{1}{N}\sum_{i=1}^{N} y^{(i)} log(P(Y=1|\mathbf{w}_1, \phi(\mathbf{x}^{(i)}))) + (1-y^{(i)}) log(P(Y=2|\mathbf{w}_2, \phi(\mathbf{x}^{(i)})))$$

(Since $y_1^{(i)} = 1 - y_2^{(i)}$ and $y_2^{(i)} = 1 - y_1^{(i)}$, we can just write $y_1^{(i)}$ as $y^{(i)}$ and $y_2^{(i)}$ as $(1 - y^{(i)})$)

$$= -\frac{1}{N}\sum_{i=1}^{N} y^{(i)} log(\sigma_{\mathbf{w}'}(\mathbf{x}^{(i)})) + (1-y^{(i)}) log(1 - \sigma_{\mathbf{w}'}(\mathbf{x}^{(i)}))$$

(As proved in (3) and (4))

Thus we reach the same form as in the slides

(b) I will use a different approach for calculating the gradient. Let p_k represent the vector for probability of kth class for all samples. Let z_k represent the kth row of Z (given in question) and let y_k represent vector for actual labels of kth class for all samples

$$\frac{\partial E(w)}{\partial W}$$

$$= \frac{\partial E(w)}{\partial Z}\frac{\partial Z}{\partial W}$$

$$= \phi(X)^T \left(\frac{\partial E(w)}{\partial z_1} \frac{\partial E(w)}{\partial z_2} \dots \frac{\partial E(w)}{\partial z_K} \right)$$

$$\frac{\partial E(w)}{\partial z_i}$$

$$= -\sum_k y_k \frac{\partial log(p_k)}{\partial o_i}$$

$$= -\sum_k y_k \frac{1}{p_k}\frac{\partial p_k}{\partial o_i}$$

$$= -y_i(1-p_i) - \sum_{k \neq i} y_k \frac{1}{p_k}(-p_k p_i)$$

$$\begin{pmatrix} \text{if i=k: } \frac{\partial p_k}{\partial o_i} = p_i(1-p_i) \\ \text{else: } \frac{\partial p_k}{\partial o_i} = -p_k p_i \end{pmatrix}$$

$$= p_i(\sum_k y_k) - y_i$$

$$= p_i - y_i$$

$$\therefore \frac{\partial E(w)}{\partial W}$$

$$= \phi(X)^T \begin{pmatrix} p_1 - y_1 & p_2 - y_2 & \dots & p_K - y_K \end{pmatrix}$$

$= \phi(X)^T(P - Y)$
where P is n x C matrix with P_{ik} = $Pr(Y = k|\mathbf{w}, \phi(x_i))$

2.2

(b) Test Accuracy I achieved : 0.871
Test Accuracy M would achieve : 0.842
Since even a trivial model could achieve such high accuracies, "accuracy" is not a good evaluation metric as it does not take into account the fact that number of **positive labels are very less.**

(c) F1 I achieved : 0.368
F1 M would achieve : 0
Yes, F1 is a good evaluation metric, as it penalises both low precision (**mislabeling as positive**) and low recall (**missing out positive labels**). So F1 takes into account the fact that positive labels are very less in the test data.

(e) Logistic Regression Test Accuracy: 0.844
Perceptron Test Accuracy : 0.791
Logistic Regression performs better than Perceptron.
This is because Perceptron tries to find **linear plains** to separate features and is thus not very efficient for non separable features as it cannot catch **outliers**. So points which deviate even slightly from their distribution can very negatively effect the results of perceptron.
On the other hand, Logistic regression attempts to find the **probability** of a data point belonging to a particular class and the probability values are improved in every iteration making it much more **generalised and flexible**. Outliers mildly effect performance as only probabilities are taken into account and not the actual separation.