CSCI 567-Machine Learning Assignment-4

1 Neural Network

$$\mathcal{L}(x,\hat{x}) = \frac{1}{2}((x_1-\hat{x}_1)^2 + (x_2-\hat{x}_2)^2) + (x_3-\hat{x}_3)^2) \tag{1}$$

$$\mathcal{L}(y,\hat{y}) = \frac{1}{2}((y_1-\hat{y}_1)^2 + (y_2-\hat{y}_2)^2) \tag{2}$$

$$\frac{\partial \mathcal{L}(y,\hat{y})}{\partial v_{jk}} = \frac{\partial \mathcal{L}(y,\hat{y})}{\partial y}\frac{\partial y}{\partial v_{jk}} = (-y+\hat{y})z$$

$$\frac{\partial \mathcal{L}(y,\hat{y})}{\partial y} = \frac{1}{2}(-2y+2\hat{y}) = -y+\hat{y}$$

$$\frac{\partial y}{\partial v_{jk}} = \frac{\partial}{\partial v_{jk}}\sum_k v_{jk}z_k$$

$$\frac{\partial \mathcal{L}(y,\hat{y})}{\partial v_{jk}} = (-y_j+\hat{y}_j)z_k \tag{3}$$

$$\frac{\partial \mathcal{L}(y,\hat{y})}{\partial w_{ki}} = \frac{\partial \mathcal{L}(y,\hat{y})}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial P_1}\frac{\partial P_1}{\partial w_{ki}}$$

$$\frac{\partial \mathcal{L}(y,\hat{y})}{\partial y} = \frac{1}{2}(-2y+2\hat{y}) = -y+\hat{y}$$

$$\frac{\partial z}{\partial P_1} = z(1-z)$$

$$\frac{\partial y}{\partial z} = v$$

$$\frac{\partial P_1}{\partial w_{ki}} = \tilde{x}$$

$$\frac{\partial \mathcal{L}(y,\hat{y})}{\partial w_{ki}} = (-y+\hat{y})vz(1-z)\tilde{x} \tag{4}$$

$$\frac{\partial \mathcal{L}(x,\hat{x})}{\partial w_{ki}} = \frac{\partial \mathcal{L}(x,\hat{x})}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial w_{ki}}$$

$$\frac{\partial \mathcal{L}(x,\hat{x})}{\partial \hat{x}} = \frac{\partial}{\partial w_{ki}}w^T z = z + \frac{\partial z}{\partial w} = z + z(1-z)\tilde{x}$$

$$\frac{\partial z}{\partial w} = \frac{\partial z}{\partial P_1}\frac{\partial P_1}{\partial w} = z(1-z)\tilde{x}$$

$$\frac{\partial \mathcal{L}(x,\hat{x})}{\partial w_{ki}} = (-x+\hat{x})(z+z(1-z)\tilde{x}) \tag{5}$$

Backpropagation update for v_{jk} (where η_1 is the step length in steepest descent):

$$v_{jk}^{t+1} = v_{jk}^{t} - \eta_1(-y+\hat{y})z$$

Backpropagation update for w_{ki}:

$$w_{ki}^{t+1} = w_{ki}^{t} - \eta_2((-y+\hat{y})vz(1-z)\tilde{x}) - \eta_3((-x+\hat{x})(z+z(1-z)\tilde{x}))$$

2 Mixture Model and EM Algorithm

2.1 Log-Likelihood

X_i values are unknown for the last $n-r$ variables. Let's call them U_i for now. For rest $X_i = y_i$.

$$logP(x,\lambda) = \sum_{i=1}^{r}(log\lambda - \lambda X_i) + \sum_{i=r+1}^{n}(log\lambda - \lambda U_i) = \sum_{i=1}^{n} log\lambda - \lambda(\sum_{i=1}^{r} y_i + \sum_{i=r+1}^{n} U_i)$$

2.2 E-Step

We will use the expectation for the last $n-r$ variable. Exponential distribution has a memoryless property. We know that each U exponential random value did not happen until c_i. Than their expectation is:

$$E(U_i) = c_i + \frac{1}{\lambda}$$

2.3 M-Step

$$E[logP(x,\lambda)] = E[\sum_{i=1}^{n} log\lambda - \lambda(\sum_{i=1}^{r} y_i + \sum_{i=r+1}^{n} U_i)]$$

$$= n\log\lambda - \lambda\sum_{i=1}^{r} y_i - (n-r)\lambda(c_i + \frac{1}{\lambda}) = n\log\lambda - \lambda\sum_{i=1}^{r} y_i - n\lambda c_i - n + r\lambda c_i + r$$

$$\frac{\partial E[logP(x,\lambda)]}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{r} y_i - nc_i + rc_i = 0$$

$$\lambda = \frac{n}{\sum_{i=1}^{r} y_i + (n-r)c_i} \tag{6}$$

3 K-Means

4 Programming Questions

Method	Data Type	Training Acc	Test Acc
Batch Gradient	raw	0.8692	0.8817
Batch Gradient	standardized	0.9300	0.9261
Newton's Method	raw	0.8935	0.8922
Newton's Method	standardized	0.8935	0.8922
glmfit	raw	0.9357	0.9265
glmfit	standardized	0.9357	0.9265

4.0.1 Chosen 20 Features

The feature IDs chosen according to Mutual Information values:
$[52, 53, 56, 21, 55, 7, 16, 57, 25, 19, 24, 5, 27, 17, 3, 26, 23, 6, 11, 2]$

Using these 20 features on glmfit function: Training accuracy = 0.8914 and Testing Accuracy = 0.8926

4.1 Generative Model and Discriminative Model

4.1.1 MLE Parameters When Variances Are Independent

Component Proportions = ([0.2941,0.2679,0.4379])
μ Parameters = ([1.538,0.0354; -1.593,1.596; -1.486,-1.408])
Σ Parameters =

First Component	Second Component	Third Component
0.9988 0.0102	1.0197 -0.4525	1.0280 0.5110
0.0102 3.9734	-0.4525 1.0105	0.5110 1.1234

Testing Accuracy = 0.9007

4.1.2 MLE Parameters When Variances Are Equal

Component Proportions = ([0.293,0.275,0.430])
μ Parameters = ([1.549,0.0435;-1.759,-0.739;-1.385,0.0282])
Σ Parameters =
0.977 -0.004
-0.004 3.331
Testing Accuracy = 0.7600

4.1.3 Multinomial Logistic Model

Testing Accuracy = 0.8846

4.1.4 Plot and Analysis

We can see that the the best decision boundaries are given by Gaussian Mixture Model with different variance. And the worst one is given by Gaussian Mixture Model with same variance.

It is obvious that variances of three Gaussian models are different from each other. Even though green and blue colored data points have a similar variance, red colored data points' distribution has much larger variance. Since we made the same variance assumption in second model, it is reasonable to get not a very good decision boundary.

On the other hand, logistic regression has made a very good classification and draw a good decision boundary between classes even though it couldn't use the advantage of being nonlinear as different variance Gaussian Mixture Model did.

4.1.5 Subset Training Data

3 plots below shows the testing accuracy w.r.t. training data sizes for three models.

4.2 Practical Logistic Regression on Toy Data

4.2.1 Discretization

The best model is chosen according to the best cross-validation accuracy. Leave one out technique is used.

Data	Best Discretization	Training Acc	Test Acc	Heldout Acc
1	16^2	0.9950	0.9737	0.9775
2	16^2	0.9975	0.9949	0.9925
3	16^2	1	0.9950	1
4	4^2	0.9800	0.9539	0.9675

4.2.2 *l*2 norm regularization)

For Data1:

Type	$\lambda = 1$	$\lambda = 0.1$	$\lambda = 0.01$	$\lambda = 0.001$
Training Accuracy	0.9700	0.9750	0.9775	0.9775
Heldout Accuracy	0.9700	0.9700	0.9700	0.9700
Testing Accuracy	0.9559	0.9583	0.9611	0.9612

For Data2:

Type	$\lambda = 1$	$\lambda = 0.1$	$\lambda = 0.01$	$\lambda = 0.001$
Training Accuracy	0.9850	0.9850	0.9825	0.9825
Heldout Accuracy	0.9825	0.9825	0.9800	0.9800
Testing Accuracy	0.9833	0.9871	0.9915	0.9915

For Data3:

Type	$\lambda = 1$	$\lambda = 0.1$	$\lambda = 0.01$	$\lambda = 0.001$
Training Accuracy	0.9800	0.9850	0.9900	0.9950
Heldout Accuracy	0.9800	0.9800	0.9800	0.9800
Testing Accuracy	0.9741	0.9760	0.9779	0.9783

For Data4:

Type	$\lambda = 1$	$\lambda = 0.1$	$\lambda = 0.01$	$\lambda = 0.001$
Training Accuracy	0.9525	0.9800	0.9800	0.9825
Heldout Accuracy	0.9425	0.9675	0.9550	0.9625
Testing Accuracy	0.9425	0.9539	0.9520	0.9587

4.2.3 Visualization

For regularized terms; for Data2 λ chosen as 1 and for Data3 λ is chosen as 0.1.// For unregularized models for data2 and data3 are both 16^2 discretizations.