COMS4721 HW2

Jeff Hudson (jdh2182)

Friday, February 20, 2015

Problem 1 (multiclass logistic regression) - 15 points

Logistic regression with more than two classes can be done using the softmax function. For data $x \in \mathbb{R}^d$ and k classes (where class i has regression vector w_i) the class of x, denoted by y, follows the probability distribution

$$P(y|x, w_1, \ldots, w_k) = \prod_{i=1}^{k} \left(\frac{e^{x^T w_i}}{\sum_{j=1}^{k} e^{x^T w_j}} \right)^{\mathbb{1}(y=i)}$$

1. Write out the log likelihood $\mathcal{L}$ of data $(x_1, y_1), \ldots, (x_n, y_n)$ using an i.i.d. assumption.

Joint Likelihood

$$= \prod_{l=1}^{n} \prod_{i=1}^{k} \left(\frac{e^{x_l^T w_i}}{\sum_{j=1}^{k} e^{x_l^T w_j}} \right)^{\mathbb{1}(y_l=i)}$$

Log Likelihood

$$= \sum_{l=1}^{n} \sum_{i=1}^{k} \mathbb{1}(y_l = i)[\log e^{x_l^T w_i} - \log \sum_{j=1}^{k} e^{x_l^T w_j}]$$

$$= \sum_{l=1}^{n} \sum_{i=1}^{k} [x_l^T w_i - \log \sum_{j=1}^{k} e^{x_l^T w_j}] \mathbb{1}(y = i)$$

2. Calculate $\nabla_{w_i} \mathcal{L}$ and $\nabla^2_{w_i} \mathcal{L}$

$$\nabla_{w_i} \mathcal{L} = \sum_{l=1}^{n} \left[x_l^T - \frac{1}{\sum_{j=1}^{k} e^{x_l^T w_j}} \times e^{x_l^T w_i} \times x_l^T \right] \mathbb{1}(y_l = i)$$

$$= \sum_{l=1}^{n} x_l^T \left[1 - \frac{e^{x_l^T w_i}}{\sum_{j=1}^{k} e^{x_l^T w_j}} \right] \mathbb{1}(y_l = i)$$

Problem 2 (Gaussian kernels) - 15 points

We saw how we can construct a kernel between two points $u, v \in \mathbb{R}^d$ using the dot product (or integral) of their high-dimensional mappings $\phi(u)$ and $\phi(v)$. In the integral case,

$$k(u, v) = \int_{\mathcal{R}^d} \phi_t(u) \phi_t(v) dt$$

, where t is some parameter that is integrated out. Show that the mapping

$$\phi_t(u) = \frac{1}{(2\pi\beta')^{d/2}} \exp\left\{ -\frac{\|u - t\|^2}{2\beta'} \right\}$$

reproduces the Gaussian kernel $k(u, v) = \alpha \exp\left\{ -\frac{\|u-v\|^2}{\beta} \right\}$ for an appropriate setting of α and β.
Hint: This will be very difficult to do without using properties of multivariate Gaussians and their marginal distributions to draw some necessary conclusions. Try framing this as a probability question.

$$k(u,v) = \int_{\mathcal{R}^d} \phi_t(u)\phi_t(v)dt$$

$$= \int_{\mathcal{R}^d} \frac{1}{(2\pi\beta')^{d/2}} \exp\left\{-\frac{\|u-t\|^2}{2\beta'}\right\} \frac{1}{(2\pi\beta')^{d/2}} \exp\left\{-\frac{\|v-t\|^2}{2\beta'}\right\} dt$$

$$= \frac{1}{(2\pi\beta')^d} \int_{\mathcal{R}^d} \exp\left\{-\frac{\|u-t\|^2 + \|v-t\|^2}{2\beta'}\right\} dt$$

$$= \frac{1}{(2\pi\beta')^d} \int_{\mathcal{R}^d} \exp\left\{-\frac{\|u\|^2 - 2u^Tt + \|v\|^2 - 2v^Tt + 2\|t\|^2}{2\beta'}\right\} dt$$

$$= \ldots \{\|u\|^2 + \|v\|^2 - 2(u+v)^Tt + 2\|t\|^2\} \ldots$$

Once we separate out the terms that depend on t, we can normalize that as its own Gaussian which will integrate to 1, then all we'll be left with is the kernel function dependent on u and v.

Problem 3 (Classification) - 70 points

In this problem you will implement three classifiers and run them on the MNIST Handwritten Digits data set posted on Courseworks and the class website. Do not do preprocessing to the data other than what is indicated at the end of the README below. The three classifiers must be implemented by you to receive full credit. Information about the data is given at the end of the assignment.

All three sub-problems ask for you to show your results in a 10×10 "confusion matrix" (call it C). This can be done as follows: For each of the 500 predictions you make from the test set, let y_t be the true class label and y_p be the predicted class label using your algorithm. Update $C(y_t, y_p) \leftarrow C(y_t, y_p) + 1$ for each prediction. At the end, C should sum to 500 and each row should sum to 50. (C can then be normalized, but leave it unnormalized for this assignment.)

Problem 3a (15 points) :

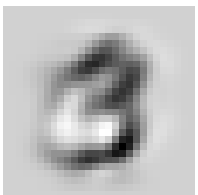
- Implement the k-NN classifier for $k = 1, 2, 3, 4, 5$.
- For each k calculate the confusion matrix and show the trace of this matrix divided by 500. This is the prediction accuracy. You don't need to show the confusion matrix.

```
##  k= Accuracy
##   1    0.948
##   2    0.914
##   3    0.944
##   4    0.940
##   5    0.946
```

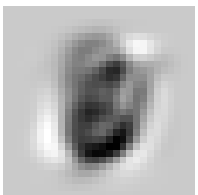
- For $k = 1$, 3, 5, show three misclassified examples and indicate the true class and the predicted class for each one (see the README below).

k=1

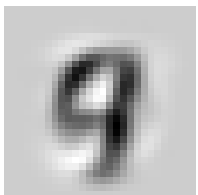
True Class: 3
Predicted Class: 5



True Class: 6
Predicted Class: 2

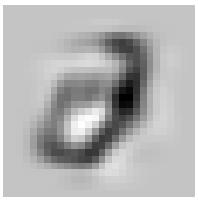


True Class: 9
Predicted Class: 4

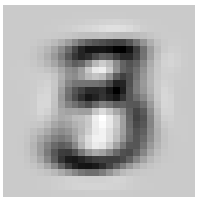


k=3

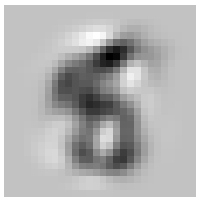
True Class: 2
Predicted Class: 3



True Class: 3
Predicted Class: 8

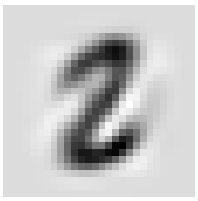


True Class: 8
Predicted Class: 5

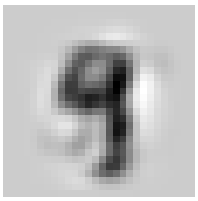


k=5

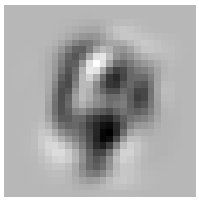
True Class: 2
Predicted Class: 8



True Class: 4
Predicted Class: 9



True Class: 6
Predicted Class: 4



Problem 3b (25 points) :

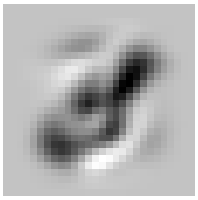
- Implement the Bayes classifier using multivariate Gaussian distributions as the generative distribution for the data in each class.
- Derive the maximum likelihood estimate for the 10-dimensional distribution on classes and the Gaussian parameters for a particular class j that you will need for this problem.
- Show the confusion matrix in a table. As in Problem 3a, indicate the prediction accuracy by summing along the diagonal and dividing by 500.

```
## $`Confusion Matrix`
##    0  1  2  3  4  5  6  7  8  9
## 0 48  0  0  1  0  1  0  0  0  0
## 1  0 49  0  0  0  0  0  0  1  0
## 2  0  0 48  0  1  0  1  0  0  0
## 3  0  0  1 47  0  0  0  0  2  0
## 4  0  0  0  0 48  0  0  0  1  1
## 5  0  0  0  1  0 45  2  0  1  1
## 6  0  0  0  0  1  5 43  0  0  1
## 7  0  0  2  0  2  0  0 46  0  0
## 8  0  0  1  0  0  1  0  0 47  1
## 9  1  0  0  0  2  0  0  0  0 47
##
## $`Prediction Accuracy`
## [1] 0.936
```

- Show the mean of each Gaussian as an image using the provided Q matrix (see the README).

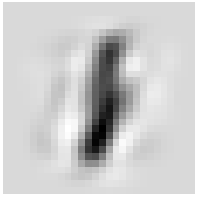
- Show three misclassified examples and show the probability distribution on the 10 digits learned by the Bayes classifier for each one.

True Class: 0
Predicted Class: 3



```
##  Digit Probability
##      0    0.000121
##      1    0.000000
##      2    0.017657
##      3    0.907672
##      4    0.000001
##      5    0.000111
##      6    0.000215
##      7    0.000000
##      8    0.074222
##      9    0.000000
```

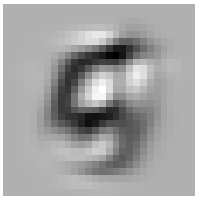
True Class: 1
Predicted Class: 8



```
##  Digit Probability
##      0    0.000000
##      1    0.000025
```

```
##      2    0.000023
##      3    0.000000
##      4    0.000173
##      5    0.000000
##      6    0.000000
##      7    0.000000
##      8    0.999780
##      9    0.000000
```

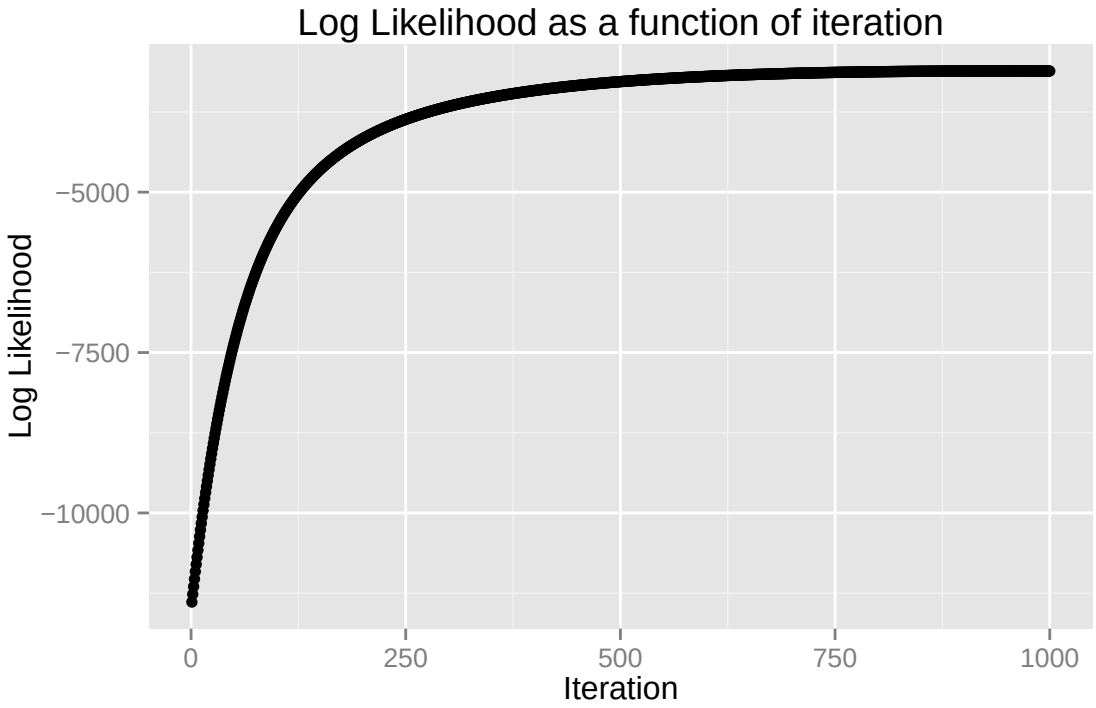
True Class: 5
Predicted Class: 9



```
##  Digit Probability
##      0    0.000000
##      1    0.000000
##      2    0.000038
##      3    0.000003
##      4    0.000072
##      5    0.123511
##      6    0.000000
##      7    0.000425
##      8    0.000213
##      9    0.875739
```

Problem 3c (30 points) :

- Implement the multiclass logistic regression classifier you derived in Problem 1. You only need to use $\nabla_w \mathcal{L}$ to satisfy the requirements of this problem. In this case, you might want try a stepsize on the order of $\rho = 0.1/5000$.
- For each cycle through $w_0, \ldots, w_9$, calculate $\mathcal{L}$ (see Problem 1) and plot as a function of iteration. Run your algorithm for 1000 iterations.



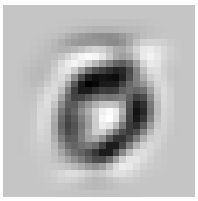
- Show the confusion matrix in a table. Indicate the prediction accuracy by summing along the diagonal and dividing by 500.

```
## $`Confusion Matrix`
##    0  1  2  3  4  5  6  7  8  9
## 0 43  0  1  0  0  5  1  0  0  0
## 1  0 40  0  0  0  2  0  0  8  0
## 2  1  0 36  3  0  0  3  0  7  0
## 3  1  0  1 38  0  3  0  0  7  0
## 4  0  0  2  0 40  1  0  0  2  5
## 5  0  1  0  6  2 38  0  0  1  2
## 6  0  0  1  0  8  4 35  0  2  0
## 7  0  0  2  0  1  0  0 42  4  1
## 8  0  0  0  0  0  3  0  0 46  1
## 9  0  0  1  0  2  1  0  0  1 45
```

```
## $`Prediction Accuracy`
## [1] 0.806
```

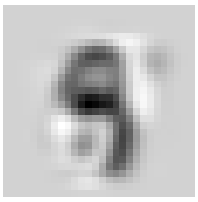
- Show three misclassified examples and show the probability distribution on the 10 digits learned by the softmax function for each one.

True Class: 0
Predicted Class: 6



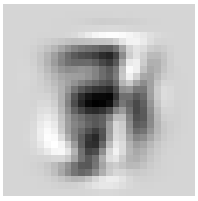
```
##  Digit Probability
##      0    0.104853
##      1    0.077484
##      2    0.097599
##      3    0.109839
##      4    0.098370
##      5    0.111870
##      6    0.112824
##      7    0.090013
##      8    0.101200
##      9    0.095948
```

True Class: 4
Predicted Class: 5



```
##  Digit Probability
##      0    0.075164
##      1    0.084157
##      2    0.091551
##      3    0.110817
##      4    0.107071
##      5    0.117754
##      6    0.095402
##      7    0.092379
##      8    0.113595
##      9    0.112109
```

True Class: 7
Predicted Class: 8



```
##  Digit Probability
##      0    0.081339
```

```
##       1    0.088542
##       2    0.099882
##       3    0.106931
##       4    0.099647
##       5    0.109767
##       6    0.101668
##       7    0.092985
##       8    0.114005
##       9    0.105236
```