Machine Learning HW2

Sirui Tan
st2957

February 27, 2016

1 Problem 1 (multiclass logistic regression)

1.1 1

Since data $(x_1, y_1), \cdots, (x_n, y_n)$ are i.i.d distributed

$$\mathcal{L} = \ln \prod_{i=1}^{k} \prod_{t=1}^{n} \left(\frac{e^{x_t^T w_i}}{\sum_{j=1}^{k} e^{x_t^T w_j}} \right)^{\mathbf{1}(y_t = i)}$$

Note here the exponential part is non-zero only when $y_t = i$. Therefore the original relationship can be simplified as

$$\mathcal{L} = \ln \prod_{t=1}^{n} \left(\frac{e^{x_t^T w_{y_t}}}{\sum_{j=1}^{k} e^{x_t^T w_j}} \right)$$

Take logarithm into consideration

$$\mathcal{L} = \sum_{t=1}^{n} \left(x_t^T w_{y_t} - \ln \sum_{j=1}^{k} e^{x_t^T w_j} \right)$$

1.2 2

$$\Delta_{w_i} \mathcal{L} = \sum_{y_t = i} x_t - \sum_{t=1}^{n} \frac{x_t^T e^{x_t^T w_i}}{\sum_{j=1}^{k} e^{x_t^T w_j}}$$

$$\Delta_{w_i}^2 \mathcal{L} = -\sum_{t=1}^{n} \frac{(x_t^T)^2 e x_t^T w_i}{\sum_{j=1}^{k} e^{x_t^T w_j}}$$

2 Problem 2 (Gaussian kernels)

Take ϕ_t into the expression

$$k(u,v) = \int \frac{1}{(2\pi\nu)^{d/2}} e^{-\frac{\|u-t\|^2+\|v-t\|^2}{2\nu}} dt$$

Because

$$\begin{aligned}
\|u-t\|^2 + \|v-t\|^2 &= u^T u + t^T t - 2u^T t + v^T v + t^T t - 2v^T t \\
&= 2[\|t\|^2 - (u+v)^T + \frac{\|u\|^2}{2} + \frac{\|v\|^2}{2}] \\
&= 2[\|t - \frac{u+v}{2}\|^2 - \|\frac{u+v}{2}\|^2 + \frac{\|u\|^2}{2} + \frac{\|v\|^2}{2}]
\end{aligned}$$

Take into kernel expression yields

$$k(u,v) = \frac{1}{(2\pi\nu)^{d/2}} e^{-\frac{1}{\nu}(-\|\frac{u+v}{2}\|^2 + \frac{\|u\|^2}{2} + \frac{\|v\|^2}{2})} \int \frac{1}{(2\pi\nu)^{d/2}} e^{-\frac{1}{\nu}\|t - \frac{u+v}{2}\|^2} dt$$

We observe the integral part is a Gaussian integral, which equals to $\frac{1}{2^{\frac{d}{2}}}$. Also

$$-\|\frac{u+v}{2}\|^2 + \frac{\|u\|^2}{2} + \frac{\|v\|^2}{2} = \frac{\|u-v\|^2}{4}$$

Therefore

$$k(u,v) = \frac{1}{2^d (\pi\nu)^{d/2}} e^{-\frac{\|u-v\|^2}{4\nu}}$$

Clearly, let

$$\begin{aligned}
\alpha &= \frac{1}{2^d (\pi\nu)^{d/2}} \\
\beta &= 4\nu
\end{aligned}$$

and the mapping successfully reproduces Gaussian kernel.

3 Problem 3 (Classication)

3.1 Problem 3a

3.1.1 Implement KNN and show prediction accuracy

```
KNN with 1 neighbor(s) get accuracy 0.948
KNN with 2 neighbor(s) get accuracy 0.926
KNN with 3 neighbor(s) get accuracy 0.936
KNN with 4 neighbor(s) get accuracy 0.934
KNN with 5 neighbor(s) get accuracy 0.936
```

Figure 1: Accuracies reached with $k = 1, \cdots, 5$

3.1.2 Show three misclassified images for $k = 1, 3, 5$

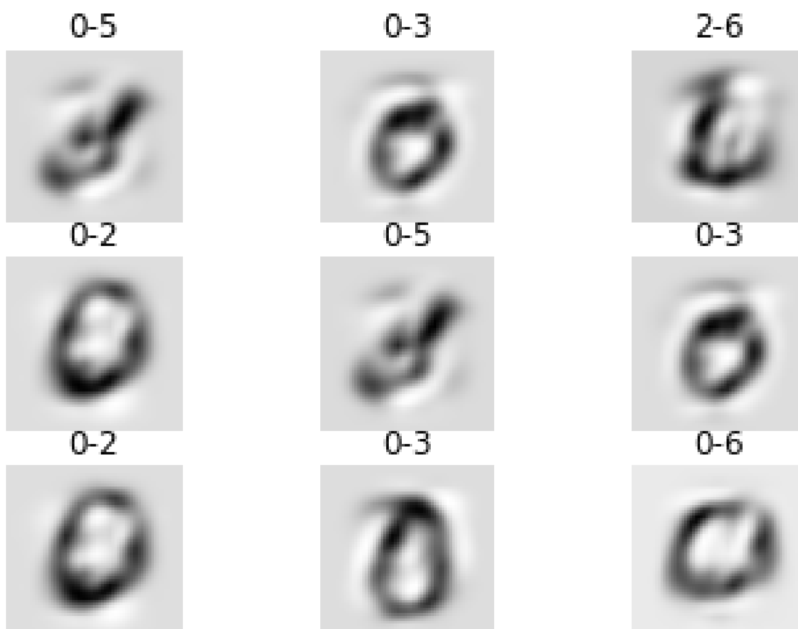


Figure 2: Misclassified images w.r.t $k = 1, 3, 5$ from top to bottom, three each, subtitles are in the format ⟨actual class⟩-⟨predicted class⟩

3.2 Problem 3b

3.2.1 Derive and show MLE for mean and covariance

Suppose the input X has dimension m. Considering the naive situation where dimensions of X are not correlated

$$\begin{aligned} P(X|y_j) &\sim N(\mu_j, \Sigma_j) \\ &\sim \prod_{k=1}^{m} N(\mu_{jk}, \sigma_{jk}) \end{aligned}$$

Therefore the likelihood can be derived as

$$L = \prod_{k=1}^{m} \left[\prod_{i=1}^{n} p(x_{ik}|y_j) \right]$$

where

$$p(x_{ik}|y_j) = N(\mu_{jk}, \sigma_{jk})$$

In order to maximize likelihood, the partial likelihood along each dimension should be maximized. Clearly

$$\begin{aligned} \mu_{jk} &= \frac{\sum_{i=1}^{n} x_{ik}}{n} \\ \sigma_{jk}^2 &= \frac{\sum_{i=1}^{n} (x_{ik} - \mu_{jk})^2}{n} \end{aligned}$$

On the other hand

$$p(y_j) = \frac{\sum_{i=1}^{n} \mathbf{1}(y_i = y_j)}{\sum_{i=1}^{n} 1}$$

3.2.2 Show confusion matrix and prediction accuracy

```
Confusion Matrix:
[[44  0  1  0  0  2  3  0  0  0]
 [ 0 48  1  0  0  1  0  0  0  0]
 [ 0  0 38  3  0  2  2  0  5  0]
 [ 1  0  1 38  0  4  0  0  5  1]
 [ 0  1  0  0 44  1  0  0  0  4]
 [ 1  1  0  3  4 40  1  0  0  0]
 [ 0  0  0  0  5  4 40  0  0  1]
 [ 0  1  2  0  1  0  0 44  1  1]
 [ 1  1  0  0  0  1  1  0 45  1]
 [ 0  0  1  0  2  0  0  0  0 47]]
NB with Gaussian likelihood get accuracy 0.856
```

Figure 3: Confusion matrix and its corresponding accuracy

3.2.3 Show mean of each Gaussian as an image

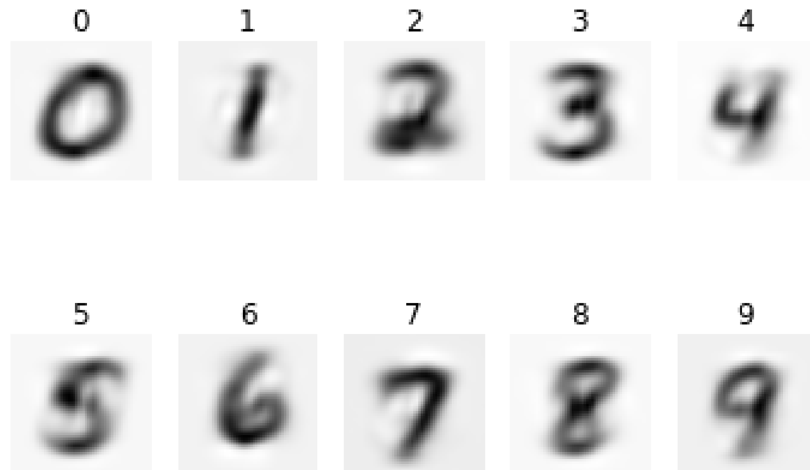


Figure 4: Mean images for each class

3.2.4 Show three misclassified images and their probability distributions


```
posterori for 0:
[  1.72688858e-05   1.26813457e-04   3.47352744e-04   1.68824274e-03
   2.08282831e-02   2.70606166e-02   1.44536812e-03   7.75761170e-01
   9.58252830e-04   1.71766631e-01]
posterori for 1:
[  5.39127161e-02   2.14607192e-13   4.83986068e-03   3.17534587e-01
   9.29565581e-03   9.93130255e-02   4.85417320e-01   2.75303235e-02
   4.77895638e-04   1.67861508e-03]
posterori for 2:
[  1.96855042e-04   6.01791438e-02   7.47515042e-01   2.53922787e-03
   6.63433212e-04   7.14861973e-04   6.52135235e-04   2.37699655e-04
   1.87261201e-01   4.04004777e-05]
```

Figure 5: Three misclassified images and their corresponding posterior probability distributions. Subtitles are in the format ⟨actual class⟩-⟨predicted class⟩

3.3 Problem 3c

3.3.1 Train softmax and show evolution of $\mathcal{L}$ with iterations

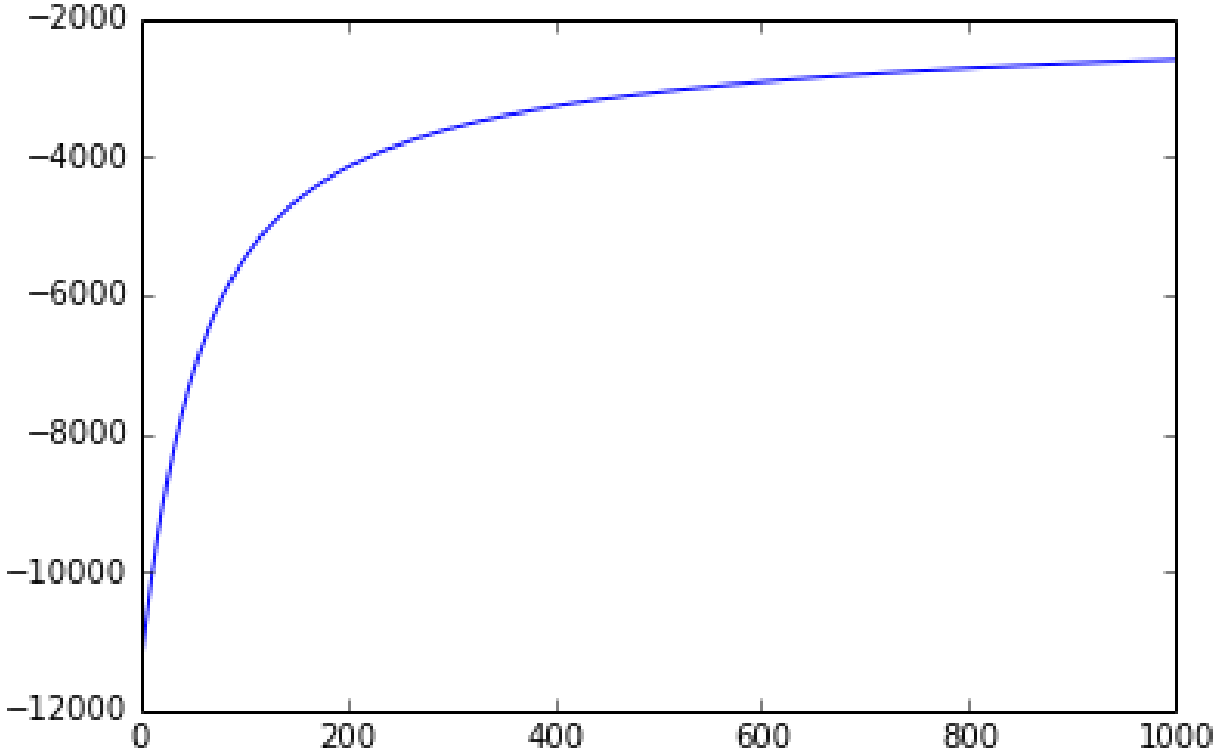


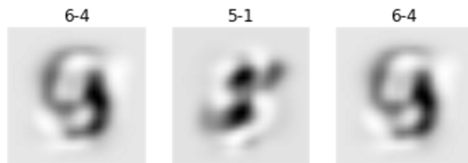
Figure 6: The evolution of log-likelihood $\mathcal{L}$ with the increase of iterations

3.3.2 Show confusion matrix and prediction accuracy

```
Confusion Matrix:
[[46  0  1  1  0  0  2  0  0  0]
 [ 0 49  0  0  0  0  0  0  1  0]
 [ 0  0 38  2  1  0  4  0  5  0]
 [ 1  0  2 39  0  2  0  1  5  0]
 [ 0  0  1  0 42  1  0  0  1  5]
 [ 1  1  0  4  2 39  1  0  0  2]
 [ 0  0  1  0  4  3 42  0  0  0]
 [ 0  0  3  0  1  0  0 44  1  1]
 [ 0  0  0  0  0  2  1  0 46  1]
 [ 0  1  1  0  3  0  0  1  0 44]]
Home-made softmax classifier get accuracy 0.858
```

Figure 7: Confusion matrix and its corresponding accuracy

3.3.3 Show three misclassified images and their probability distributions




```
posterori for 0:
[ 0.00829482  0.01107798  0.06291947  0.07534041  0.42174891  0.02790394
  0.1844372   0.09517093  0.02394314  0.08916322]
posterori for 1:
[ 0.00358437  0.68413307  0.09222992  0.00752431  0.01550136  0.01650325
  0.01249437  0.04375087  0.10962164  0.01465684]
posterori for 2:
[ 0.00829482  0.01107798  0.06291947  0.07534041  0.42174891  0.02790394
  0.1844372   0.09517093  0.02394314  0.08916322]
```

Figure 8: Three misclassified images and their corresponding posterior probability distributions. Subtitles are in the format ⟨actual class⟩-⟨predicted class⟩