CSC411 Assignment 2

Yue Guo

November 12, 2017

1 Gaussian

1.1 P(y)

$$
\begin{aligned}
P(\boldsymbol{x}|\mu,\sigma) &= \sum_{k=1}^{K} P(\boldsymbol{x}|y=k,\ mu,\sigma)P(y=k|\mu,\sigma) \\
&= \sum_{k=1}^{K} \alpha_k P(\boldsymbol{x}|y=k,\mu,\sigma) \\
&= \sum_{k=1}^{K} \alpha_k (\prod_{i=1}^{D} 2\pi\sigma_i)^{\frac{1}{2}} exp\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\}
\end{aligned}
$$

$$
\begin{aligned}
P(y|\boldsymbol{x},\ mu,\sigma) &= \frac{P(\boldsymbol{x},y=k|\mu,\sigma)}{P(\boldsymbol{x}|\mu,\sigma)} \\
&= \frac{P(\boldsymbol{x}|y=k,\ mu,\sigma)P(y=k|\mu,\sigma)}{P(\boldsymbol{x}|\mu,\sigma)} \\
&= \frac{\alpha_k(\prod_{i=1}^{D} 2\pi\sigma_i^2)^{-\frac{1}{2}} exp\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\}}{\sum_{k=1}^{K}\alpha_k(\prod_{i=1}^{D} 2\pi\sigma_i^2)^{-\frac{1}{2}} exp\{-\sum_{i=1}^{D}\frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\}}
\end{aligned}
$$

1.2 log likelihood

$$
\begin{aligned}
-logP(y^{(1)},x^{(1)},...,y^{(N)},x^{(N)}|\theta) &= -logP(y^{(1)},x^{(1)}|\theta) - logP(y^{(2)},x^{(2)}|\theta)... - log(y^{(N)},x^{(N)}|\theta) \\
&= \sum_{j=1}^{N} -logP(y^{(j)},x^{(j)}|\theta)
\end{aligned}
$$

N.B.: in this case, j is the index of the whole data set,
where as y(j) corresponds to the class that this sample
belongs to, and y(i) = k , k in (0, 9)

$$
\begin{aligned}
&= \sum_{j=1}^{N} -log[\alpha^{y(j)}(\prod_{i=1}^{D} 2\pi(\sigma_j^{y(i)})^2)^{-\frac{1}{2}} exp\{-\sum_{i=1}^{D}\frac{1}{(2\sigma_i^{y(j)})^2}(x_i^{(j)}-\mu_i^{y(j)})^2\}] \\
&= \sum_{j=1}^{N}\sum_{i=1}^{D}\frac{1}{2(\sigma_i^{y(j)})^2}(x_i^{(j)}-\mu_i^{y(j)})^2 + \sum_{j=1}^{N} -log[\alpha^{y(j)}(\prod_{i=1}^{D} 2\pi(\sigma_i^{y(j)})^2)^{-\frac{1}{2}}] \\
&= \sum_{j=1}^{N}\sum_{i=1}^{D}\frac{1}{2(\sigma_i^{y(j)})^2}(x_i^{(j)}-\mu_i^{y(j)})^2 + \sum_{j=1}^{N}\frac{1}{2}log[\alpha^{y(j)}(\prod_{i=1}^{D} 2\pi(\sigma_i^{y(j)})^2] \\
&= \sum_{j=1}^{N}\sum_{i=1}^{D}\frac{1}{2(\sigma_i^{y(j)})^2}(x_i^{(j)}-\mu_i^{y(j)})^2 + \sum_{j=1}^{N}\frac{1}{2}log(\alpha^{y(j)}) + \sum_{j=1}^{N}\sum_{i=1}^{D} log(2\pi(\sigma_i^{y(j)})^2)
\end{aligned}
$$

1.3 MLE of μ

Note: y(j) = k, j is the index in the whole data set, j in (0, N), y(j) indicates class k

$$
\begin{aligned}
\frac{\partial}{\partial \mu_{ki}} &= \sum_{j=1}^{N}\sum_{i=1}^{D} \frac{1}{2\sigma_i^2} 2(-x_i^{(j)} + \mu_{ki})\mathbb{1}(y(j) = k) \\
&= \sum_{j=1}^{N}\sum_{i=1}^{D} \frac{1}{\sigma_i^2} (-x_i^{(j)} + \mu_{ki})\mathbb{1}(y(j) = k)
\end{aligned}
$$

Setting $\frac{\partial}{\partial \mu_{ki}} = 0$, we have

$$
\begin{aligned}
\sum_{j=1}^{N}\sum_{i=1}^{D} \frac{1}{\sigma_i^2} (-x_i^{(j)} + \mu_{ki})\mathbb{1}(y(j) = k) = 0 \\
\sum_{j=1}^{N}\sum_{i=1}^{D} \frac{1}{\sigma_i^2} x_i \mathbb{1}(y(j) = k) &= \sum_{j=1}^{N}\sum_{i=1}^{D} \frac{1}{\sigma_i^2} \mu_{ki} \mathbb{1}(y(j) = k) \\
\mu_{ki} &= \frac{\sum_{j=1}^{N}\sum_{i=1}^{D} x_i^{(j)} \mathbb{1}(y(j) = k)}{\sum_{j=1}^{N}\sum_{i=1}^{D} \mathbb{1}(y(j) = k)}
\end{aligned}
$$

1.4 MLE of σ

Note: y(j) = k, j is the index in the whole data set, j in (0, N), y(j) indicates class k

$$
\begin{aligned}
\frac{\partial}{\partial \sigma_i^2} &= \sum_{j=1}^{N}\sum_{i=1}^{D} -\frac{1}{2}(x_i^{y(j)} - \mu_{ki})^2 \frac{1}{\sigma_i^4} + \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{D} \frac{2\pi}{2\pi\sigma_i^2} \\
&= \sum_{j=1}^{N}\sum_{i=1}^{D} -\frac{1}{2}(x_i^{y(j)} - \mu_{ki})^2 \frac{1}{\sigma_i^4} + \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{D} \frac{1}{\sigma_i^2} \\
&= \sum_{j=1}^{N}\sum_{i=1}^{D} \frac{1}{2}[-(x_i^{y(j)} - \mu_{ki})^2 \frac{1}{\sigma_i^4} + \frac{1}{\sigma_i^2}]\mathbb{1}(y(j) = k)
\end{aligned}
$$

Setting $\frac{\partial}{\partial \sigma_i^2}$ to zero, we have

$$
\begin{aligned}
\sum_{j=1}^{N}\sum_{i=1}^{D} \mathbb{1}(y(j) = k)(x_i - \mu_{ki})^2 \frac{1}{\sigma_i^4} &= \sum_{j=1}^{N}\sum_{i=1}^{D} \mathbb{1}(y(j) = k)\frac{1}{\sigma_i^2} \\
\sum_{j=1}^{N}\sum_{i=1}^{D} \mathbb{1}(y(j) = k)(x_i - \mu_{ki})^2 &= \sum_{j=1}^{N}\sum_{i=1}^{D} \mathbb{1}(y(j) = k)\sigma_i^2 \\
\sigma_i^2 &= \frac{\sum_{j=1}^{N}\sum_{i=1}^{D} \mathbb{1}(y(j) = k)(x_i - \mu_{ki})^2}{\sum_{j=1}^{N}\sum_{i=1}^{D} \mathbb{1}(y(j) = k)}
\end{aligned}
$$

2 Handwritten digits

2.1 KNN

1. Plot of digit means

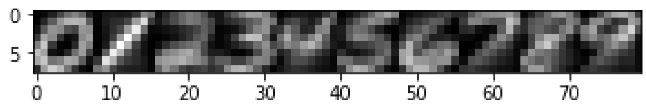

Figure 1: Plot of means in each digit class

2.1.1 KNN Euclidean

(a) K = 1, test accuracy = 0.96875, train accuracy = 1.0

(b) K = 15, test accuract = 0.9585 , train accuracy = 0.959428571429

2.1.2 Break ties

In my program, I have the array label_count[0 , 10] in function def query_knn(self, test_point, k): that stores the number of neighbours with the digit label corresponding to the index of label_count array. I find the maximum number in label_count, and the index of the max is the majority vote. If there are multiple, the function label_count.argmax() will pick the index that comes first. This is justified in class, we can break ties randomly or assign to the first maximum vote.

2.1.3 K = 1 to 15

K	Training Accuracy	Test Accuracy
1	1.0	0.96875
2	0.982571428571	0.96175
3	0.983428571429	0.96975
4	0.978	0.9665
5	0.977571428571	0.96775
6	0.974285714286	0.9645
7	0.973714285714	0.96325
8	0.970571428571	0.9615
9	0.969285714286	0.9605
10	0.967571428571	0.961
11	0.965428571429	0.9595
12	0.963142857143	0.95825
13	0.962428571429	0.95775
14	0.960142857143	0.95775
15	0.959428571429	0.9585

K	Average Fold
1	0.964428571429
2	0.957571428571
3	0.963428571429
4	0.961
5	0.960857142857
6	0.959
7	0.957857142857
8	0.957428571429
9	0.955571428571
10	0.952857142857
11	0.952285714286
12	0.951428571429
13	0.950428571429
14	0.95
15	0.948571428571

2.2 Conditional Gaussian

2.2.1 Plot of log

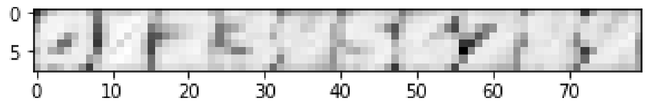

Figure 2: log of the diagonal elements of each covariance matrix

2.2.2 average log likelihood

Train: 40.0734191088
Test: 35.1502166328

2.2.3 Accuracy

Train 0.981285714286
Test 0.95925

2.3 Naives Bayes

2.3.1 Plot of η

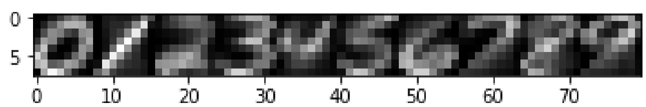

Figure 3: Plot of eta

2.3.2 Plot of new samples

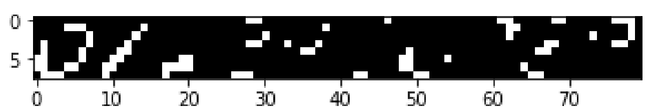

Figure 4: Plot of new samples

2.3.3 average log likelihood

training average log likelihood -30.7898021047
testing average log likelihood -30.7369008592

2.3.4 accuracy

training accuracy 0.774142857143
testing accuracy 0.76425

2.4 summary

The test accuracy of KNN is 96.875%, test accuracy of conditional Gaussian is 95.925%, and Naive Bayes is around 76.425%. The highest performance is KNN, next followed closely by conditional gaussian, and Naive Bayes is the lowest.
This matches my expectation because KNN is inferring class label based on neighbours, and conditional Gaussian is using shared covariance matrix. Our data set, handwritten digits, have very similar form in one class. In both KNN and conditional Gaussian, we are assuming that images in each class are correlated, and we predict based on what we have seen already in a class.
In Naive Bayes, however, we assume that all data are independent from each other. This assumption unfortunately does not hold in this dataset, and therefore reducing accuracy of our predictions.