Assignment 2

T yashwanth Reddy
201402163

T. Yashwanth - 201402163

Bayesian Parameter Estimation (BPE)

$$P(\omega_i|x, D_i) = \frac{P(x|\omega_i, D_i)\, P(\omega_i)}{\sum_{j=1}^{c} P(x|\omega_j, D_j)\, P(\omega_j)}$$

fixing (ω_i)

$$P(X|D) = \int P(x, \theta|D)\, d\theta = \int P(x|\theta)\, P(\theta|D)$$

Estimate $P(\theta|D)$ using BPE, Let us identify the parameters μ_n & σ_n for the underlying gaussian determining $P(\mu|D)$

$$P(\theta|D) = \alpha \prod_{k=1}^{n} P(x_k|\theta) \cdot P(\theta) = \alpha \prod_{k=1}^{n} P(x_k|\mu) \cdot P(\mu)$$

univariate:

$$P(\theta|D) = \alpha' \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\left(\frac{\mu - x_k}{\sigma}\right)^2 + \left(\frac{\mu - \mu_0}{\sigma_0}\right)^2\right)\right]$$

$$\therefore P(\mu) = \frac{1}{\sqrt{\sigma_0 2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_0}{\sigma_0}\right)^2\right]$$

$$P(x_k|\mu) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2}\left(\frac{x_k - \mu}{\sigma}\right)^2\right]$$

$$\Rightarrow P(\theta|D) = P(\mu|D) = \alpha'' \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right]$$

$\therefore$ Comparing with $P(\mu|D) = \frac{1}{\sqrt{2\pi\sigma_n}} \exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_n}{\sigma_n}\right)^2\right]$

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2}\right)\bar{x}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2}\mu_0.$$

$$\sigma_n^2 = \frac{\sigma_0^2\sigma^2}{n\sigma_0^2 + \sigma}$$

$$\therefore \quad P(x|D) = \int P(x|\theta) \cdot P(\theta|D) \cdot d\theta$$

$$P(x|D) = \int \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \cdot \frac{1}{\sqrt{2\pi\sigma_n}} \exp\left[-\frac{1}{2}\left(\frac{\mu - \mu_n}{\sigma_n}\right)^2\right] d\mu$$

$$= \frac{1}{2\pi\sigma\sigma_n} \int \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 - \frac{1}{2}\left(\frac{\mu - \mu_n}{\sigma_n}\right)^2\right] d\mu$$

$$= \frac{1}{2\pi\sigma\sigma_n}\int \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma^2}+\frac{\mu^2}{\sigma^2}-\frac{2\mu x}{\sigma^2}+\frac{\mu^2}{\sigma_n^2}+\frac{\mu_n^2}{\sigma_n^2}-\frac{2\mu\mu_n}{\sigma_n^2}\right)\right]$$

$$= \frac{1}{2\pi\sigma\sigma_n}\int \exp\left[-\frac{1}{2}\left(\left(\frac{(x-\mu_n)^2}{\sigma^2+\sigma_n^2}\right)+\frac{\sigma^2+\sigma_n^2}{\sigma^2\sigma_n^2}\left(\mu-\frac{\sigma_n^2 x+\sigma^2\mu_n}{\sigma^2+\sigma_n^2}\right)^2\right)\right]d\mu$$

$\therefore p(x|D) \Rightarrow$ Gaussian with μ_n (mean) & $\sigma^2+\sigma_n^2$ (variance)

Multivariate:

$$p(\theta|D) = p(\mu|D) = \beta' \cdot \prod_{k=1}^{n} \exp\left[\frac{1}{2}\frac{(x-\mu)^2}{\Sigma}\right] \cdot \exp\left[-\frac{1}{2}\frac{(\mu-\mu_0)^2}{\Sigma_0}\right]$$

$$= \beta'' \cdot \exp\left[-\frac{1}{2}\left(\sum_{k=1}^{n}\frac{(\mu-x_k)^2}{\Sigma}+\frac{(\mu-\mu_0)^2}{\Sigma_0}\right)\right]$$

$$= \beta'' \cdot \exp\left[-\frac{1}{2}\left[\mu^T\left(\frac{n}{\Sigma}+\frac{1}{\Sigma}\right)\mu - 2\mu^T\left(\frac{1}{\Sigma}\sum_{k=1}^{n}x_k+\frac{\mu_0}{\Sigma_0}\right)\right]\right]$$

$$p(\mu|D) = \alpha'' \exp\left(-\frac{1}{2}(\mu-\mu_n)^T \Sigma_n^{-1}(\mu-\mu_n)\right)$$

on comparing we get

$$\Sigma_n^{-1} = n\Sigma^{-1}+\Sigma_0^{-1} \qquad \Sigma_n^{-1}\mu_n = n\Sigma^{-1}\mu_n + \Sigma_0^{-1}\mu_0$$

$$\Sigma_n = \frac{\Sigma_0 \cdot \Sigma}{n\Sigma_0+\Sigma} \qquad \therefore \mu_n = \left(\frac{n\Sigma_0}{n\Sigma_0+\Sigma}\right)\bar{x} + \left(\frac{\Sigma}{n\Sigma_0+\Sigma}\right)\mu_0$$

$$\Rightarrow \quad p(x|D) = \int p(x|\mu)\cdot p(\mu|D)\cdot d\mu$$

$$\therefore p(x|D) = \underbrace{\mu_n}_{\text{mean}} \ \& \ \underbrace{\Sigma+\Sigma_n}_{\text{Varian matrix}}$$

$$\Sigma+\Sigma_n = \Sigma + \frac{\Sigma_0 - \Sigma}{n\Sigma_0+\Sigma}$$

$$= \frac{(n+1)\Sigma_0\cdot\Sigma + \Sigma\cdot\Sigma}{n\Sigma_0+\Sigma}$$

$$\Sigma+\Sigma_n = \frac{\Sigma(\Sigma+(n+1)\Sigma_0)}{n\Sigma_0+\Sigma}$$

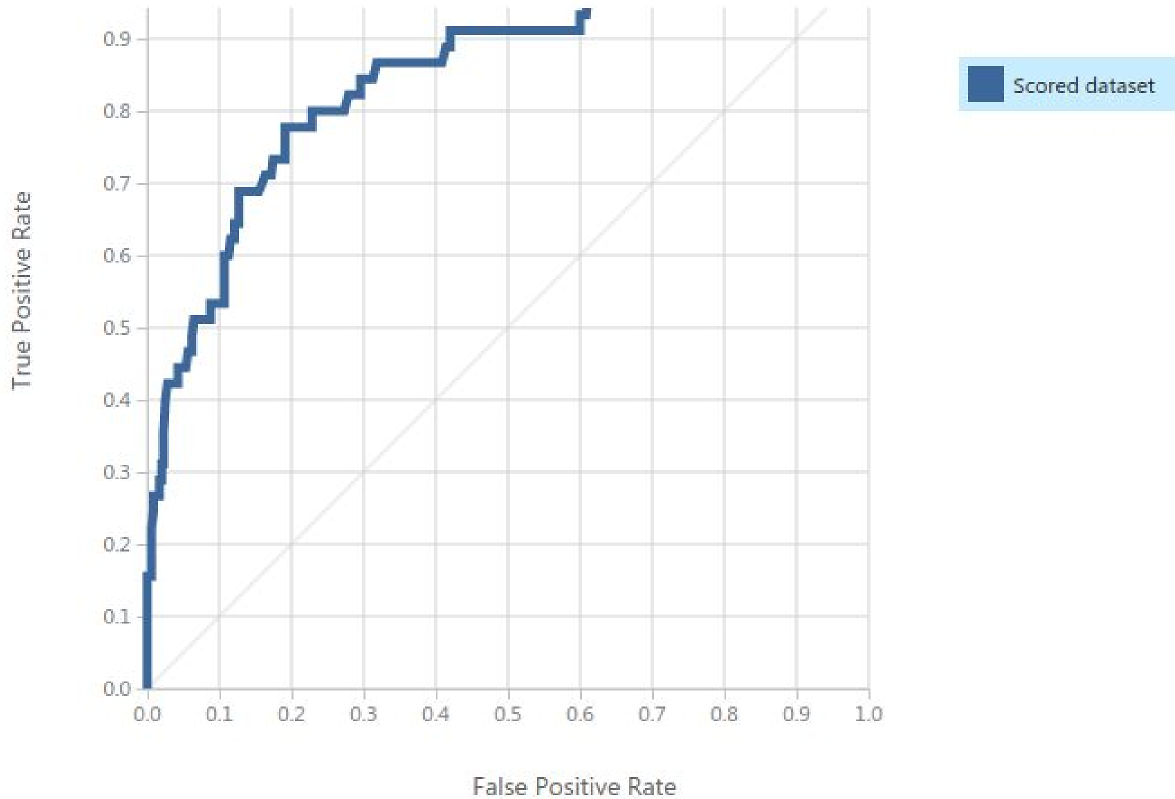
1

Accuracy	Precision	Recall	F-Score	AUC	Average Log Loss	Training Log Loss
0.9544368 02818083	0.6982332 15547703	0.3613752 74323336	0.4762593 39599904	0.9361372 08513201	0.1308243 05818383	40.411469 8602503

3

Accuracy	Precision	Recall	F-Score	AUC	Average Log Loss	Training Log Loss
0.9025	0.8	0.1777777 77777778	0.2909090 90909091	0.8500156 49452269	0.3677266 55063227	-4.553795 57249289

After applying PCA and downscaling to 500 features



Score Bin	Positive Examples	Negative Examples	Fraction Above Threshold	Accuracy	F1 Score	Precision	Recall	Negative Precision	Negative Recall	Cumulative AUC
(0.900,1.000]	0	0	0.000	0.887	0.000	1.000	0.000	0.887	1.000	0.000
(0.800,0.900]	0	0	0.000	0.887	0.000	1.000	0.000	0.887	1.000	0.000
(0.700,0.800]	2	0	0.005	0.892	0.085	1.000	0.044	0.892	1.000	0.000
(0.600,0.700]	0	0	0.005	0.892	0.085	1.000	0.044	0.892	1.000	0.000
(0.500,0.600]	6	2	0.025	0.902	0.291	0.800	0.178	0.905	0.994	0.001
(0.400,0.500]	12	17	0.098	0.890	0.476	0.513	0.444	0.931	0.946	0.019
(0.300,0.400]	17	86	0.355	0.718	0.396	0.261	0.822	0.969	0.704	0.187
(0.200,0.300]	6	126	0.685	0.417	0.270	0.157	0.956	0.984	0.349	0.506
(0.100,0.200]	2	113	0.973	0.140	0.207	0.116	1.000	1.000	0.031	0.819
(0.000,0.100]	0	11	1.000	0.113	0.202	0.113	1.000	1.000	0.000	0.850

Python code for PCA

```
import csv
import pandas as pd
from sklearn.decomposition import PCA
from numpy import zeros as np
file_=open('1train_dorothea.csv')
w_file=open('VV_train_dorothea.csv','w')
X=np((800,100000),dtype=int)
#print X[1]
row=0
```

```
for line in file_:
        line=line.strip()
        line_list=line.split(',')
        for count in line_list:
                count=int(count)
                X[row][count-1]=1
        row=row+1
#print X[0]
df = pd.DataFrame(data=X)
df = df.transpose()
pca = PCA(n_components=500)
pca.fit(df)
#print pca.components_
Y=pca.components_
Y=Y.transpose()
print X.shape,Y.shape
row =0
cloumn=0
while row < 800:
        cloumn=0
        while cloumn < 500:
                w_file.write(str(Y[row][cloumn]))
                if cloumn != 499:
                        w_file.write(",")
                cloumn=cloumn+1
        w_file.write("\n")
        row=row+1
```