COMS 4771 Machine Learning (Spring 2018)
Problem Set #1

Jianfu Yang - jy2863 Wenda Xu - wx2195 Fan Yang - fy2232
- wx2195@columbia.edu

February 13, 2018

Problem 1

(i)

$$\int_{-\infty}^{+\infty} p(x|\theta)dx = 1 \;\Rightarrow\; p(x|\theta) = \begin{cases} \frac{1}{2\theta^3}x^2e^{-\frac{x}{\theta}} & x \geqslant 0 \\ 0 & otherwise \end{cases}$$

$$\arg\max_{\theta} \ln \mathcal{L}(\theta|X) = \arg\max_{\theta} \sum_{k=1}^{n}(2\ln x_k - \frac{x_k}{\theta} - 3\ln\theta - \ln 2)$$

$$\frac{d\ln \mathcal{L}(\theta|X)}{d\theta} = \frac{\sum_{k=1}^{n} x_k}{\theta^2} - \frac{3n}{\theta} = 0$$

To maximize $\mathcal{L}$, $\theta_{ML} = \frac{\sum_{k=1}^{n} x_k}{3n}$.

(ii)

$$\int_{-\infty}^{+\infty} p(x|\theta)dx = 1 \;\Rightarrow\; p(x|\theta) = \begin{cases} \frac{1}{2\theta} & -\theta \leqslant x \leqslant \theta \\ 0 & otherwise \end{cases}$$

For any i, if $x_i < -\theta$ or $x_i > \theta$ applies, then $\mathcal{L}(\theta|X) = 0$. Hence, to maximize $\mathcal{L}(\theta|X)$, θ should be no smaller than $max(|X|)$. That is, $\theta \geqslant max(|X|)$. Then:

$$\arg\max_{\theta} \ln \mathcal{L}(\theta|X) = \arg\max_{\theta} \sum_{k=1}^{n}(\ln \frac{1}{2\theta})$$

This is inverse correlation with θ. Hence to maximize it, $\theta_{ML} = max(|X|)$.

(iii) When μ is unknown, as derived in lecture:

$$\begin{aligned}
\bar{X} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\
\mathbb{E}[\sigma_{ML}^2] &= \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})^2] \\
&= \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}((x_i - \mu) + (\mu - \bar{X}))^2] \quad (\textit{Assume } \mu \textit{ is the real mean}) \\
&= \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 + \frac{1}{n}\sum_{i=1}^{n}2(x_i - \mu)(\mu - \bar{X}) + \frac{1}{n}\sum_{i=1}^{n}(\mu - \bar{X})^2] \\
&= \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 + 2(\bar{X} - \mu)(\mu - \bar{X}) + (\mu - \bar{X})^2] \\
&= \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2] - \mathbb{E}[(\mu - \bar{X})^2] \\
&= Var(X) - Var(\bar{X}) = \sigma^2 - \frac{\sigma^2}{n} = \frac{n-1}{n}\sigma^2
\end{aligned}$$

It's clearly, $\sigma_{ML}^2 \neq \sigma^2$. Hence, σ_{ML}^2 isn't an unbiased estimator. To fix it:

$$\sigma_{ML}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2 = \sigma^2$$

(iv) If $g(\theta)$ is a one-to-one function: suppose we have likelihood function $\mathcal{L}(\theta)$. And it can also written as $\mathcal{L}(g^{-1}(g(\theta)))$. These two are both maximized at θ_{ML}. That is:

$$g^{-1}(g(\theta)_{ML}) = \theta_{ML}$$

$$g(\theta_{ML}) = g(\theta)_{ML}$$

Therefore, the MLE of $g(\theta)$ is $g(\theta_{ML})$.
When $g(\theta)$ is a many-to-one function: let $\boldsymbol{\theta} = g^{-1}(x)$ denotes all θ satisfying $g(\theta) = x$. Thus, $\theta_{ML} \in g^{-1}(g(\theta)_{ML})$. $g(\theta_{ML}) = g(\theta)_{ML}$ also applies.
Proven.
From this result we can infer the MLE in Part(iii):

$$g(\sigma^2) = \sqrt{\sigma^2} = \sigma$$

$$\sigma_{ML}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X}_n)^2$$

$$\sigma_{ML} = g(\sigma_{ML}^2) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X}_n)^2}$$

Problem 2

(i) We can span a binary tree as follows:
Each layer represents each dimension of the input data($\vec{X} = [x_1, x_2, ..., x_D]$). Therefore we have at most D layers. Then at each layer(x_i) split the tree to left when $x_i = 0$ or right otherwise. Now we have a tree already.
Finally determine the leaves of the tree:
At each leaf trace back to its parent nodes till the first layer. We obtained the path from first layer to each leaf and can be showed as $\{0,1\}^D$. Then compute $g(x)$ and assign the result to corresponding leaf.
This binary tree is a decision tree and satisfied $T(x) = g(x)$.

(ii) The best possible bound we can give on the maximum height of such a decision tree T is D, which is the dimension of our input data. When the height is D, all dimensions have been used to classify x. Hence, a tree with height larger than D doesn't exist.
If a classifier g satisfies that changing any feature of x will change the result, then the decision tree for this classifier g will have a height of D.
From this, let's give an example of g which is the bound tight:
$g : \{0,1\}^D \to \{0,1\}$ and input data is $\vec{X} = [x_1, x_2, ..., x_D]$)

$$g(\vec{x}) = (\sum_{i=1}^{D} x_i) \bmod 2$$

Problem 3

$$\begin{aligned}
Q(g) - Q(f) &= \mathbb{E}_{x,y}[(g(x) - y)^2 - (f(x) - y)^2] \\
&= \mathbb{E}_{x,y}[g(x)^2 - f(x)^2 - 2(g(x) - f(x))y] \\
&= \mathbb{E}_{x,y}[(g(x) - f(x))(g(x) + f(x) - 2y)] \\
&= \mathbb{E}_{x}[(g(x) - f(x))(g(x) + f(x) - 2\mathbb{E}[Y|X])] \\
&= \mathbb{E}_{x}[(g(x) - f(x))(g(x) + f(x) - 2f(x))] \\
&= \mathbb{E}_{x}[(g(x) - f(x))^2] \geqslant 0
\end{aligned}$$

Hence, $Q(f) \leqslant Q(g)$ for any g. That is, $f(x)$ is the optimal predictor w.r.t Q for continuous output spaces.
Proven.

Problem 4

(i)

$$\begin{aligned} M^T &= (A^T A)^T \\ &= (A)^T (A^T)^T \\ &= A^T A = M \end{aligned}$$

Hence, M is symmetric.

$$\begin{aligned} x^T M x &= x^T A^T A x \\ &= (Ax)^T (Ax) \\ &= \| Ax \|_2^2 \geqslant 0 \end{aligned}$$

Hence, M is positive semi-definite.

(ii)

$$\begin{aligned} \beta^{(N)} &= \beta^{(N-1)} + \eta(\nu - M\beta^{(N-1)}) \\ &= \eta\nu + (I - \eta M)\beta^{(N-1)} \\ &= \eta\nu + (I - \eta M)\eta\nu + (I - \eta M)^2 \beta^{(N-2)} \\ &= \sum_{k=0}^{i-1} (I - \eta M)^k \eta\nu + (I - \eta M)^i \beta^{(N-i)} \\ &= \sum_{k=0}^{N-1} (I - \eta M)^k \eta\nu + (I - \eta M)^N \beta^{(0)} \quad (\beta^{(0)} = (0, \cdots, 0)) \\ &= \eta \sum_{k=0}^{N-1} (I - \eta M)^k \nu \end{aligned}$$

Proven.

(iii)

$$\begin{aligned} (I - \eta M)x &= Ix - \eta M x \\ &= x - \eta\lambda_i x \\ &= (1 - \eta\lambda_i)x \end{aligned}$$

Hence, the eigenvalues of $(I - \eta M)$ are $(1 - \eta\lambda_i)$.

$$\begin{aligned} A^k x &= A^{k-1} A x \\ &= A^{k-1} \lambda x \\ &= \lambda^k x \end{aligned}$$

Hence, $(I - \eta M)^k$ has the same eigenvectors with $(I - \eta M)$ (also with M). And the eignvalues are $(1 - \eta\lambda_i)^k$.

The eigenvalues of $\eta \sum_{k=0}^{N-1}(I-\eta M)^k$ is:

$$\begin{aligned}\eta \sum_{k=0}^{N-1}(1-\eta\lambda_i)^k &= \frac{1-(1-\eta\lambda_i)^N}{1-(1-\eta\lambda_i)}\eta \\ &= \frac{1-(1-\eta\lambda_i)^N}{\lambda_i} \quad (i=1\dots d)\end{aligned}$$

(iv)

$$\begin{aligned}\| \beta^{(N)} - \hat{\beta} \|_2^2 =& \| \eta \sum_{k=0}^{N-1}(I-\eta M)^k M\hat{\beta} - \hat{\beta} \|_2^2 \\ =& \| (\eta \sum_{k=0}^{N-1}(I-\eta M)^k M - I)\hat{\beta} \|_2^2 \\ \leqslant& \| \eta \sum_{k=0}^{N-1}(I-\eta M)^k M - I \|_2^2 \quad \| \hat{\beta} \|_2^2\end{aligned}$$

Since M is symmetric, $\sum_{k=0}^{N-1}(I-\eta M)^k \eta M - I$ is also symmetric. Then the 2-Norm of it equals its max eigenvalue. Besides, $\eta \sum_{k=0}^{N-1}(I-\eta M)^k$ has the same eigenvectors with M as discussing above. Hence:

$$\begin{aligned}\| \sum_{k=0}^{N-1}(I-\eta M)^k \eta M - I \|_2^2 \quad \| \hat{\beta} \|_2^2 &= ((\frac{1-(1-\eta\lambda_i)^N}{\lambda_i}\lambda_i - 1)_{max})^2 \| \hat{\beta} \|_2^2 \\ &= (1-\eta\lambda_{min})^{2N} \| \hat{\beta} \|_2^2 \\ &\leqslant (e^{-\eta\lambda_{min}})^{2N} \| \hat{\beta} \|_2^2 \\ &= e^{-2\eta\lambda_{min}N} \| \hat{\beta} \|_2^2\end{aligned}$$

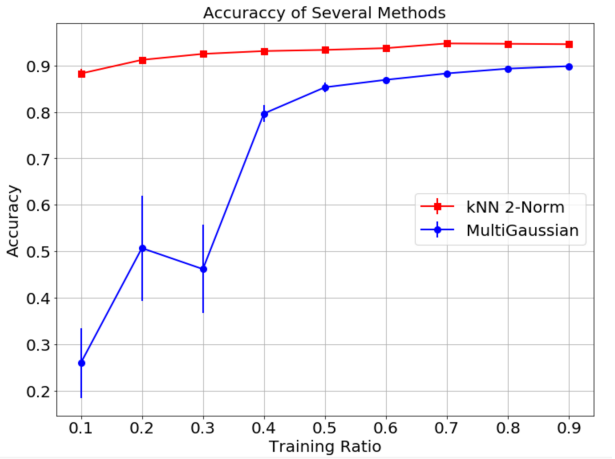
Proven.

Problem 5

(i) The code is submitted through Courseworks.

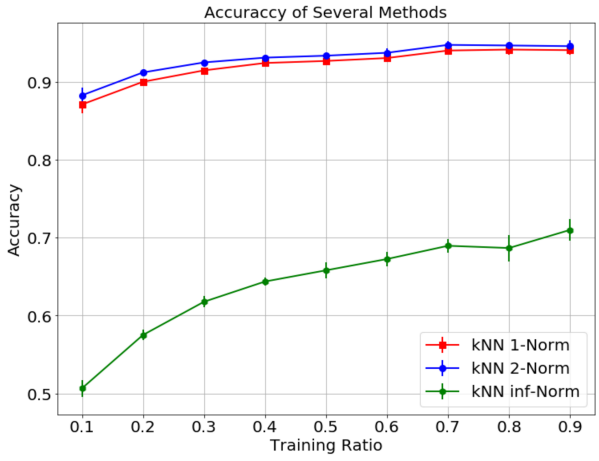
(ii) The code is submitted through Courseworks.
In this question, using order=2 to calculate Euclidean distance.

(iii) The accuracy of multigaussian bayes and kNN 2-Norm is shown below:



Clearly, with smaller training data size, kNN has higher accuracy and is more stable than multigaussian bayes. With training datasize increases, this difference decreases.

(iv) The accuracy of kNN methods with different distance types is shown below:

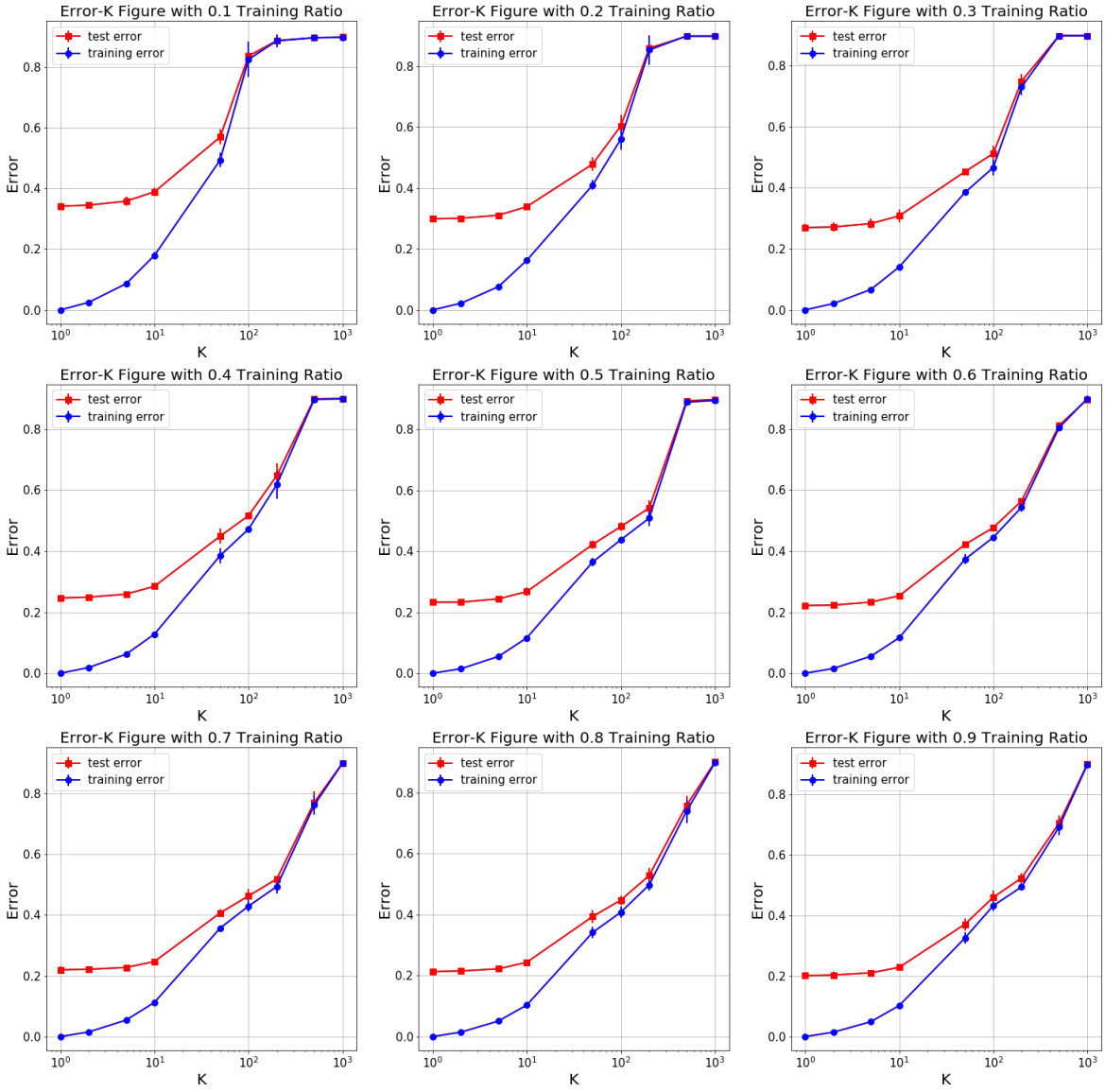


1-Norm and 2-Norm has similar performance while they both have higher accuracy and are more stable than ∞-Norm. Besides, 2-Norm is slightly better than ∞-Norm.

Problem 6

(i) The code is submitted through Courseworks.
The hyperparameter K used in this problem denotes the maximum number of points in those cells whose uncertainty larger than 0. When K is small, the length of each leaf is small, causing the tree-depth becomes larger.

(ii)&(iii) The training error and test error w.r.t K for each training ratio is plotted as below:



K takes 1, 2, 5, 10, 50, 100, 200, 500 and 1000.
The model complexity has inverse correlation with K. It's clearly that both training error and test error will decrease with K decreasing for all the training ratio.
Training error decreases faster than test error, and could finally be 0 when $K = 1$, while test error couldn't.
With training ratio increasing, test error will decrease.

(iv) When K is small, i.e., the model complexity is large enough, training error could be extremely small. Test error decreases slowly when $K < 10$. This is because the model is already good enough, which means continueing splitting the cell will only help a little. Besides, the points with different label in this cell may be a noise. Hence, test error decreases slower than training error and couldn't archieve 0.

(v) In this problem, since the data is really good, a smaller K always behaves better than a larger K. Hence we could get the best perfomance when taking $K = 1$. Considering the time consumption, taking $K = 5$ could save much time and has approximating accuracy with $K = 1$.