Jiafeng Ni
CSCI567
Homework #1
02/10/2016

1.Density Estimation, 2.Nearest Neighbor, 3.Additive or Dropout Noising as Regularization, 4.Decision Tree, 5. Programming Problem

1.(a.1) We assume that all samples can only take value between 0 and 1, and they are generated from the Beta distribution with parameter α unknown and $\beta = 1$. Please show how to derive the maximum likelihood estimator of α.

(*a*.1) According to problem, $x_i \sim Beta(\alpha, 1)$, where $\beta = 1$, so we have

$$\begin{aligned} p(x_i|\alpha, 1) &= \frac{1}{B(\alpha, 1)} \cdot {x_i}^{\alpha-1} \\ &= \frac{1}{\int_0^1 t^{\alpha-1} \cdot dt} \cdot {x_i}^{\alpha-1} \\ &= \alpha \cdot {x_i}^{\alpha-1} \end{aligned} \tag{1}$$

Given (1), the likelihood function would be:

$$L(\alpha) = \prod_{i=1}^{n} p(x_i|\alpha, 1) = \alpha^n \cdot \prod_{i=1}^{n} {x_i}^{\alpha-1} \tag{2}$$

If we choose e as base, we can get the ln-likelihood function as:

$$l(\alpha) = \ln L(\alpha) = n \ln \alpha + (\alpha - 1) \sum_{i=1}^{n} \ln x_i \tag{3}$$

We want the α that maximizes ln-likelihood function, then:

$$\begin{aligned} \frac{\partial l(\alpha)}{\partial \alpha} &= \frac{n}{\alpha} + \sum_{i=1}^{n} \ln x_i = 0 \\ \alpha &= -\frac{n}{\sum_{i=1}^{n} \ln x_i} \end{aligned} \tag{4}$$

■

1.(a.2) We assume that all samples are generated from Normal distribution $N(\theta, diag(\theta))$. Please show how to derive the maximum likelihood estimator of $\theta \in R^d$ where $diag(\theta)$ represents a square matrix with diagonal elements equal to vector θ and all other elements 0.

(*a*.2) Since $x_i \sim N(\theta, \sigma)$, i.e.:

$$f(x|\theta,\sigma) = \frac{1}{(2\pi)^{\frac{k}{2}}|\Sigma|^{\frac{1}{2}}} \cdot exp[-\frac{1}{2}(x-\theta)^T\Sigma^{-1}(x-\theta)] \tag{5}$$

Given (5), the likelihood function would be:

$$L(\theta,\Sigma) \propto |\Sigma|^{-\frac{n}{2}} \cdot exp[-\frac{1}{2}\sum_{i=1}^{n}(x_i-\theta)^T\Sigma^{-1}(x_i-\theta)]$$

So we can get the log-likelihood function as:

$$l(\theta,\Sigma) = -\frac{n}{2}\ln|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}(x_i-\theta)^T\Sigma^{-1}(x_i-\theta) + const \tag{6}$$

According to the problem statement, σ =diag(θ), then:

$$\ln|\Sigma| = \sum_{k=1}^{d}\ln\theta^{(k)} \tag{7}$$

$$\sum_{i=1}^{n}(x_i-\theta)^T\Sigma^{-1}(x_i-\theta) = \sum_{k=1}^{d}\sum_{i=1}^{n}\frac{\left(x_i^{(k)}-\theta^{(k)}\right)^2}{\theta^{(k)}} \tag{8}$$

To maximize the likelihood function (6), we require $\frac{dl(\theta)}{\theta^{(k)}} = 0$ for any k, i.e.:

$$-\frac{n}{2}\frac{1}{\theta^{(k)}} - \frac{1}{2}\sum_{i=1}^{n}\left[\frac{\left(x_i^{(k)}\right)^2}{\left(\theta^{(k)}\right)^2}+1\right] = 0$$

$$\left(\theta^{(k)}\right)^2 + \theta^{(k)} - \frac{1}{n}\sum_{i=1}^{n}\left(x_i^{(k)}\right)^2 = 0 \tag{9}$$

Let $\overline{x^{(k)}}$ denote $\frac{1}{n}\sum_{i=1}^{n}\left(x_i^{(k)}\right)^2$, solve the equation (9), we get any element $\theta^{(k)}$ in θ should be(only considering that $\theta^{(k)} \geq 0$ because of Σ):

$$\theta^{(k)} = \frac{-1+\sqrt{1+4\overline{x^{(k)}}}}{2} \tag{10}$$

Where

$$\overline{x^{(k)}} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i^{(k)}\right)^2 \tag{11}$$

■

1.(b.1) Prove given equations (1) and (2).

(*b*.1) According to given distribution, we have:

$$P(y_n \mid x_n, \theta) = \frac{1}{\sqrt{2\pi}\sigma} \cdot exp\{-\frac{[y_n - (\omega_0 + \omega^T x_n)]^2}{2\sigma}\} \tag{12}$$

Given (12), the log-likelihood function is:

$$\begin{aligned} l(\theta) &= \log P(D) = \log \prod_{i=1}^{N} P(y_i \mid x_i, \theta) \\ &= \sum_{i=1}^{N} \log P(y_i \mid x_i, \theta) \\ &= \sum_{i=1}^{N} -\log \sqrt{2\pi}\sigma - \frac{[y_i - (\omega_0 + \omega^T x_i)]^2}{2\sigma^2} \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^{N} [y_i - (\omega_0 + \omega^T x_i)]^2 - \frac{N}{2} \log \sigma^2 - N \log \sqrt{2\pi} \end{aligned} \tag{13}$$

To get the optimal ω_0 and ω, we need:

$$\frac{\partial l(\theta)}{\partial \omega_0} = -\frac{1}{2\sigma^2} \sum_{i=1}^{N} [y_i - (\omega_0 + \omega^T x_i)] \cdot (-1) = 0 \tag{14}$$

$$\frac{\partial l(\theta)}{\partial \omega} = \frac{\partial}{\partial \omega} \sum_{i=1}^{N} [y_i - (\omega_0 + \omega^T x_i)]^2 = 0 \tag{15}$$

For (14), we get:

$$\sum_{i=1}^{N} y_i = N \cdot \omega_0 + \sum_{i=1}^{N} \omega^T x_i$$

$$\omega_0 = \frac{1}{N} \sum_{i=1}^{N} y_i - \frac{1}{N} \sum_{i=1} N \omega^T x_i = \overline{y} - \omega^T \overline{x} \tag{16}$$

Put the value of ω_0 in (16) into (15), we have:

$$\frac{\partial}{\partial \omega} \sum_{i=1}^{N} \left[y_i - \overline{y} - \omega^T (x_i - \overline{x}) \right]^2 = 0$$

$$\frac{\partial}{\partial \omega} \left[\sum_{i=1}^{N} (y_i - \overline{y})^2 - 2 \sum_{i=1}^{N} (y_i - \overline{y}) \omega^T (x_i - \overline{x}) + \sum_{i=1}^{N} (xi - \overline{x})^T \omega \omega^T (xi - \overline{x}) \right] = 0$$

$$0 - 2 \sum_{i=1}^{N} (y_i - \overline{y})(x_i - \overline{x}) + \left(2 \sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})^T \right) \omega = 0$$

$$\hat{\omega} = \left[\sum_{i=1}^{N}(x_i - \overline{x})(x_i - \overline{x})^T\right]^{-1}\left[\sum_{i=1}^{N}(y_i - \overline{y}(x_i - \overline{x}))\right] \tag{17}$$

Equations (16) and (17) are exactly what we want to prove. ∎

1.(b-2) According to above sub-problem, we can firstly compute $\overline{\omega}$ on the centered data, and then estimate ω_0 using equation (1). However, let's temporarily forget the task of estimating regression parameters ω and consider the distribution of $\left[x_i^c, y_i^c\right]$. Assume x and y both follow Gaussian distribution. Then by finding the MLE of $\Sigma_{X^cY^c}$, $\Sigma_{X^cX^c}$, and using the formula for conditional Gaussian, derive equation (2).

to be finished ∎

1.(c) Suppose random variables $X_1, X_2, \ldots, X_n$ are i.i.d sampled according to density function $f(x)$ and the kernel density estimation is in the form of $\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x-X_i}{h})$. Show the bias of the kernel density estimation method.

(*c*.1) According to the definition of expectation:

$$\mathbb{E}\left[\frac{1}{h}K\left(\frac{x - X_i}{h}\right)\right] = \int_{-\infty}^{\infty}\frac{1}{h}K\left(\frac{x-t}{h}\right)f(t)dt \tag{18}$$

Given (18), it can be easily shown that:

$$\mathbb{E}_{X_1,\ldots,X_n}\left[\hat{f}(x)\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\frac{1}{h}K\left(\frac{x - X_i}{h}\right)\right] = \frac{1}{h}\int_{-\infty}^{\infty}K\left(\frac{x-t}{h}\right)f(t)dt \tag{19}$$

(*c*.2) Using Taylor's theorem, and assuming $f(x)$ is order of ν, we can get:

$$f(x - hz) = f(x) + f'(x)\cdot(-hz) + \frac{1}{2!}f''(x)(-hz)^2 + \ldots + \frac{1}{\nu}f^{(\nu)}(x)(-hz)^\nu + o(h^\nu) \tag{20}$$

(*c*.3) Given $z = \frac{x-t}{h}$, apply it to equation (19), we have:

$$\mathbb{E}\left[\hat{f}(x)\right] = \frac{1}{h}\int K(z)f(x - hz)hdz = \int K(z)f(x - hz)dz \tag{21}$$

Combined with equation (20):

$$\mathbb{E}\left[\hat{f}(x)\right] = \int K(z)\left[f(x) + f'(x)\cdot(-hz) + \frac{1}{2!}f''(x)(-hz)^2 + o(h^2)\right]dz \tag{22}$$

By the definition of kernel, $\int K(z)dz = 1$, $\int K(z)zdz = 0$, and let $\sigma_\nu^2 = \int K(z)z^2dz$, then:

$$\mathbb{E}\left[\hat{f}(x)\right] = f(x) + \frac{h^2\sigma_\nu^2 f''(x)}{2} + o(h^2)$$

Thus:

$$\mathbb{E}\left[\hat{f}(x)\right] - f(x) = \frac{h^2\sigma_v^2 f''(x)}{2} + o(h^2) \tag{23}$$

Where

$$\sigma_v^2 = \int K(z)z^2 dz$$

. ■

1.(d) Show that MLE cannot be used to estimate optimal value of h in $\hat{f} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x-X_i}{h})$.

$(\boldsymbol{d})$ Given $\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x-X_i}{h})$, we can write the likelihood function as:

$$L(h) = \prod_{k=1}^{K}\hat{f}(x_k|h) = \prod_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x_k - X_i}{h}) \tag{24}$$

Thus the log-likelihood function is as:

$$l(h) = \sum_{k=1}^{K}\left[-\ln n + \ln\left(\sum_{i=1}^{n}K(\frac{x_k - X_i}{h}) - \ln h\right)\right] \tag{25}$$

To compute the optimal value $\hat{h}$ for h, we need the derivative of $l(h)$ to h, while we don't know what the kernel is, how the kernel distributes. It may be incontinuous, or non-differentiable. So for this log-likelyhood function, we just cannot use MLE to get the optimal value of h. ■

2.(a) Suppose we have the locations (coordinates) of 10 USC students during class time, and we know their majors, as follows:
Mathematics: {(15,49),(7,38),(4,47)}
Electrical Engineering: {(29, 24) , (32, 36) , (37, 43)}
Computer Science: {(18, 9) , (40, 28) , (8, 19) , (11, 12)}
Normalize the data, by following the formula on K-Nearest Neighbor lecture slide. And using different parameters to estimate the major of a student whose coordinate is at (9, 18).

$(\boldsymbol{a}.\mathbf{1})$ From the problem statement, we have:

$$X = \begin{bmatrix} 15 & 49 \\ -7 & 38 \\ -4 & 47 \\ 29 & 24 \\ 32 & 36 \\ 37 & 43 \\ 18 & 9 \\ 40 & -28 \\ -8 & -19 \\ -11 & 12 \end{bmatrix}$$

Let $X^{(1)} = X(:,1), X^{(2)} = X(:,2)$, we can get:

$$\overline{x^{(1)}} = \frac{1}{n}\sum_{i=1}^{n} X_i^{(1)} = 14.1 \tag{26}$$

$$\overline{x^{(2)}} = \frac{1}{n}\sum_{i=1}^{n} X_i^{(2)} = 21.1 \tag{27}$$

Let $\hat{X^{(1)}} = X^{(1)} - ones(n,1) * \overline{x^{(1)}}$, and $\hat{X^{(2)}} = X^{(1)} - ones(n,1) * \overline{x^{(2)}}$, then:

$$\sigma_1 = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} \hat{X^{(1)}}^T \hat{X^{(1)}}} = 20.1243 \tag{28}$$

$$\sigma_2 = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} \hat{X^{(2)}}^T \hat{X^{(2)}}} = 27.27 \tag{29}$$

Given (26), (27), (28), and (29), we can get normalized $X_{norm}^{(1)} = \frac{\hat{X^{(1)}}}{\sigma_1}$, and $X_{norm}^{(2)} = \frac{\hat{X^{(2)}}}{\sigma_2}$, then:

$$X_{norm} = [X_{norm}^{(1)} \;\; X_{norm}^{(2)}] = \begin{bmatrix} 0.0447 & 1.0231 \\ -1.0485 & 0.6197 \\ -0.8994 & 0.9498 \\ 0.7404 & 0.1063 \\ 0.8895 & 0.5464 \\ 1.1379 & 0.8031 \\ 0.1938 & -0.4437 \\ 1.2870 & -1.8005 \\ -1.0982 & -1.4705 \\ -1.2472 & -0.3337 \end{bmatrix}$$

Thus, the normalized locations are as follows:
Mathematics: $(0.0447, 1.0231), (-1.0485, 0.6197), (-0.8994, 0.9498)$
Electrical Engineering: $(0.7404, 0.1063), (0.8895, 0.5464), (1.1379, 0.8031)$
Computer Science: $(0.1938, -0.4437), (1.2870, -1.8005), (-1.0982, -1.4705), (-1.2472, -0.3337)$

(*a*.2) Now we get the new point $x = (9, 18)$, it can be normalized as

$$x_{norm} = \left(\frac{x(1,1) - \overline{x^{(1)}}}{\sigma_1}, \; \frac{x(1,2) - \overline{x^{(2)}}}{\sigma_2} \right) = [-0.2534, \; -0.1137]$$

We define:

$$dif = X_{norm} - ones(10,1) * x_{norm} = \begin{bmatrix} 0.2981 & 1.1368 \\ -0.7951 & 0.7334 \\ -0.6460 & 1.0635 \\ 0.9938 & 0.2200 \\ 1.1429 & 0.6601 \\ 1.3913 & 0.9168 \\ 0.4472 & -0.3300 \\ 1.5404 & -1.6868 \\ -0.8448 & -1.3568 \\ -0.9938 & -0.2200 \end{bmatrix}$$

Considering L_2 distance metric, we can get the distance from x_{norm} to each of the training points:

$$D_{L_2} = sum(dif.^2,\ 2) = \begin{bmatrix} 1.3812 \\ 1.1701 \\ 1.5483 \\ 1.0361 \\ 1.7419 \\ 2.7763 \\ 0.3089 \\ 5.2182 \\ 2.5545 \\ 1.0361 \end{bmatrix}$$

Case 1 [$K = 1$ and using L_2 distance]:
If $K = 1$, the $7th$ point is the nearest neighbor of x, so $Major(K = 1, L_2) = major(X_7) =$ **Computer Science**.

Case 2 [$K = 3$ and using L_2 distance]:
If $K = 3$, the label with majority vote should still be **Computer Science**, since X_7, X_{10}, X_4 rank the first three according to the distance to x, and both X_7 and X_{10} are labeled as Computer Science.

Similarly, considering L_1 distance metric, we can get the distance vector:

$$D_{L_1} = sum(abs(dif),\ 2) = \begin{bmatrix} 1.4349 \\ 1.5285 \\ 1.7095 \\ 1.2138 \\ 1.8030 \\ 2.3081 \\ 0.7772 \\ 3.2272 \\ 2.2016 \\ 1.2138 \end{bmatrix}$$

Case 3 [$K = 1$ and using L_1 distance]:
If $K = 1$, the $7th$ point is the nearest neighbor of x, so $Major(K = 1, L_2) = major(X_7) =$ **Computer Science**.

Case 4 [$K = 3$ and using L_1 distance]:
If $K = 3$, the label with majority vote should still be **Computer Science**, since X_7, X_{10}, X_4 rank the first three according to the distance to x, and both X_7 and X_{10} are labeled as Computer Science.

Comparison:
In our case, all metrics with different K's come out with the same result. But if we do the Leave-one-out test, we can get the accuracy for each model. We can get:

$$A^{train}(K = 1, L_2) = 80\%$$

$$A^{train}(K = 3, L_2) = 60\%$$

$$A^{train}(K = 1, L_1) = 80\%$$

$$A^{train}(K = 3, L_1) = 80\%$$

Thus, given this specific training set, we can say that the model with L_2 distance metric and $K = 3$ performs the worst, the other three have the same accuracy rate.

■

2.(b.1) Given all definitions in the problem, using the fact that $\sum_c K_c = K$, derive the formula for unconditional density $p(x)$.

(b.1) Since $p(x|Y = c) = \frac{p(x,Y=c)}{p(Y=c)}$, given $p(x|Y = c)$ and $p(Y = c)$, we can get:

$$\begin{aligned} p(x, Y = c) &= p(x|Y = c) \cdot p(Y = c) \\ &= \frac{K_c}{N_c \cdot V} \cdot \frac{N_c}{N} \\ &= \frac{K_c}{N \cdot V} \end{aligned} \tag{30}$$

Based on equation (30) can get:

$$\begin{aligned} p(x) &= \sum_c p(x, Y = c) \\ &= \sum_c \frac{K_c}{N \cdot V} \\ &= \frac{K}{N \cdot V} \end{aligned} \tag{31}$$

■

2.(b-2) Using Bayes rule, derive the formula for the posterior probability of class membership $p(Y = c|x)$.

$(\boldsymbol{b.2})$According to Bayes rule, we get:

$$\begin{aligned} p(Y = c|x) &= \frac{p(x|Y = c) \cdot p(Y = c)}{p(x)} \\ &= \frac{\frac{K_c}{N_c \cdot V} \cdot \frac{N_c}{N}}{\frac{K}{N \cdot V}} \\ &= \frac{K_c}{K} \end{aligned} \tag{32}$$

■

3.(a) By making use of $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i \epsilon_j] = \delta_{i,j}\sigma^2$, show that minimizing E averaged over the noise distribution $E(\omega) = \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\left\{y(\tilde{x}_n^{(m)}, \omega) - t_n\right\}^2$ is equivalent to minimizing the sum-of-squares error for noise-free input variables with the addition of a weight-decay regularization term, in which the bias parameter ω_0 is omitted from the regularizer.

$(\boldsymbol{a})$According to the problem,

$$\begin{aligned} y(\tilde{x}_n^{(m)}, \omega) &= \omega_0 + \omega^T \tilde{x}_n^{(m)} \\ &= \omega_0 + \omega^T (x_n + \epsilon_m) \\ &= y(x_n, \omega) + \omega^T \epsilon_m \end{aligned} \tag{33}$$

Based on (33) we can get:

$$\begin{aligned} E(\omega) &= \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\left\{y(\tilde{x}_n^{(m)}, \omega) - t_n\right\}^2 \\ &= \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\left\{y(x_n, \omega) - t_n + \omega^T \epsilon_m\right\}^2 \\ &= \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[y(x_n, \omega) - t_n\right]^2 + \omega^T \epsilon_m {\epsilon_m}^T \omega + \omega^T \epsilon_m [y(x_n, \omega) - t_n] \end{aligned} \tag{34}$$

Given that $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i \epsilon_j] = \delta_{i,j}\sigma^2$, we have:

$$\mathbb{E}\left[\omega^T \epsilon_m [y(x_n, \omega) - t_n]\right] = 0 \tag{35}$$

$$\mathbb{E}[\omega^T \epsilon_m {\epsilon_m}^T \omega] = \sigma^2 \|\omega\|_2^2 \tag{36}$$

Thus,

$$\mathbb{E}[E(\omega)] = \frac{1}{2}\sum_{n=1}^{N}\{[y(x_n,\omega)-t_n]^2 + \sigma^2\|\omega\|_2^2\} \tag{37}$$

Notice that the second term above is as the same format in the ridge regression, without the bias parameter ω_0, and the first term is exactly the empirical loss function when we haven't include the Gaussian noise on x_n. Proof done. ∎

3.(b.1) Show that $E[\tilde{x}_n] = x_n$, and $Var[\omega^T\tilde{x}_n] = \frac{\delta}{1-\delta}\sum_{d=1}^{D} {x_{n,d}}^2{\omega_d}^2$.

(*b*.1) Given the definition of $\tilde{x}_{n,d}$, we can get:

$$\begin{aligned}\mathbb{E}[\tilde{x}_{n,d}] &= 0\cdot\delta + [x_{n,d}/(1-\delta)]\cdot(1-\delta)\\ &= x_{n,d}\end{aligned} \tag{38}$$

$$\begin{aligned}Var[\tilde{x}_{n,d}] &= \mathbb{E}[(\tilde{x}_{n,d}-\mathbb{E}[\tilde{x}_{n,d}])^2]\\ &= \delta\cdot(0-x_{n,d})^2 + (1-\delta)\cdot(x_{n,d}/(1-\delta)-x_{n,d})^2\\ &= \delta x_{n,d}^2 + \frac{\delta^2}{1-\delta}x_{n,d}^2\\ &= \frac{\delta}{1-\delta}x_{n,d}^2\end{aligned} \tag{39}$$

From (38) and (39), we get the expectation and variance of each element in vector $\tilde{x_n}$, thus:

$$\mathbb{E}[\tilde{x_n}] = x_n \tag{40}$$

$$Var[\omega^T\tilde{x}_n] = \frac{\delta}{1-\delta}\sum_{d=1}^{D} x_{n,d}^2\omega_d^2 \tag{41}$$

∎

3.(b.2) By making use of above expressions of $E[\tilde{x}_n]$ and $Var[\omega^T\tilde{x}_n]$, show that minimizing E averaged over the noise distribution $E(\omega) = \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\{y(\tilde{x}_n^{(m)},\omega)-t_n\}^2$ is equivalent to minimizing the sum-of-squares error for noise-free input variables with the addition of a weight-decay regularization term, in which the bias parameter ω_0 is omitted from the regularizer.

(*b*.2)

$$\begin{aligned}E(\omega) &= \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\{y(\tilde{x}_n^{(m)},\omega)-t_n\}^2\\ &= \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\{\omega_0+\omega^T x_n - t_n + \omega^T(\tilde{x}_n^{(m)}-x_n)\}^2\\ &= \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\omega^T(\tilde{x}_n^{(m)}-x_n)(\omega_0+\omega^T x_n - t_n)\\ &\quad + (\omega_0+\omega^T x_n - t_n)^2 + [\omega^T(\tilde{x}_n^{(m)}-x_n)]^2\end{aligned} \tag{42}$$

Given equation (40), we can get:

$$\mathbb{E}[\tilde{x}_n^{(m)} - x_n] = 0 \tag{43}$$

$$\mathbb{E}[\omega^T(\tilde{x}_n^{(m)} - x_n)(\omega_0 + \omega^T x_n - t_n)] = 0 \tag{44}$$

Taking advantage of equation (41), we also have:

$$\mathbb{E}[\omega^T(\tilde{x}_n^{(m)} - x_n)^2] = Var[\omega^T \tilde{x}_n^{(m)}] = \frac{\delta}{1-\delta}\sum_{d=1}^{D} x_{n,d}^2 \omega_d^2 \tag{45}$$

Combine (43), (44), and (45), we finally arrive at:

$$\mathbb{E}[E(\omega)] = \frac{1}{2}\sum_{n=1}^{N}\{[y(x_n,\omega) - t_n]^2 + \frac{\delta}{1-\delta}\sum_{d=1}^{D} x_{n,d}^2 \omega_d^2\} \tag{46}$$

Notice that the first term in the above equation is exactly the empirical loss function when we haven't include the dropout noise, and the second term corresponds to weight decay regularization term in ridge regression while having different λ for each element ω_d in parameter vector ω, with the bias parameter ω_0 omitted. Proof done.

■

4.(a) Suppose you want to grow a decision tree to predict the Play-Tennis Feasibility based on the following data which provides the feasibility of playing tennis in 14 observations. Which predictor variable will you choose to split in the first step to maximize the information gain?

$(\boldsymbol{a})$According to statistics given by the problem statement, we can calculate the conditional information gain as follows:

$$IG(Outlook) = 1 - \left[\frac{5}{14}I(\frac{2}{5},\frac{3}{5}) + \frac{4}{14}I(\frac{4}{4},\frac{0}{4}) + \frac{5}{14}I(\frac{3}{5},\frac{2}{5})\right] \doteq 0.246bits$$

$$IG(Temperature) = 1 - \left[\frac{4}{14}I(\frac{2}{4},\frac{2}{4}) + \frac{6}{14}I(\frac{4}{6},\frac{2}{6}) + \frac{4}{14}I(\frac{3}{4},\frac{1}{4})\right] \doteq 0.089bits$$

$$IG(Humidity) = 1 - \left[\frac{7}{14}I(\frac{3}{7},\frac{4}{7}) + \frac{7}{14}I(\frac{6}{7},\frac{1}{7})\right] \doteq 0.212bits$$

$$IG(Wind) = 1 - \left[\frac{8}{14}I(\frac{6}{8},\frac{2}{8}) + \frac{6}{14}I(\frac{3}{6},\frac{3}{6})\right] \doteq 0.108bits$$

Obviously, $IG(Outlook)$ is the largest, i.e. we can get the most information if we choose Outlook in the first step. ■

4.(b) Which attributes will you choose for the second level of the tree? You need to show all the calculations.

$(\boldsymbol{b})$Given the first step, we can get conditional information gain:

$$IG(Temperature|Sunny) = 1 - \left[\frac{2}{5}I(\frac{0}{2},\frac{2}{2}) + \frac{2}{5}I(\frac{1}{2},\frac{1}{2}) + \frac{1}{5}I(1,0)\right] = 0.6bits$$

$$IG(Humidity|Sunny) = 1 - \left[\frac{3}{5}I(\frac{0}{3},\frac{3}{3}) + \frac{2}{5}I(\frac{2}{2},\frac{0}{2})\right] = 1bits$$

$$IG(Wind|Sunny) = 1 - \left[\frac{3}{5}I(\frac{1}{3},\frac{2}{3}) + \frac{2}{5}I(\frac{1}{2},\frac{1}{2})\right] = 0.049bits$$

$$IG(Temperature|Rain) = 1 - \left[\frac{3}{5}I(\frac{2}{3},\frac{1}{3}) + \frac{2}{5}I(\frac{1}{2},\frac{1}{2})\right] = 0.049bits$$

$$IG(Humidity|Rain) = 1 - \left[\frac{2}{5}I(\frac{1}{2},\frac{1}{2}) + \frac{3}{5}I(\frac{2}{3},\frac{1}{3})\right] = 0.049bits$$

$$IG(Wind|Rain) = 1 - \left[\frac{3}{5}I(\frac{3}{3},\frac{0}{3}) + \frac{2}{5}I(\frac{0}{2},\frac{2}{2})\right] = 1bits$$

According to the above results, we can easily choose **Humidity** for the **Sunny branch** in the second step, and choose **Wind** for the **Rain branch** in the second step. For the **Overcase branch**, we don't need to do anything. ∎

4.(c) In training decision trees, the ultimate goal is to minimize the classification error. However, the classification error is not a smooth function; thus, several surrogate loss functions have been proposed. Two of the most common loss functions are the Gini index and Cross-entropy, see [MLaPP, Section 16.2.2.2] or [ESL, Section 9.2.3] for the definitions. Prove that, for any discrete probability distribution p with K classes, the value of the Gini index is less than or equal to the corresponding value of the cross-entropy. This implies that the Gini index is a better approximation of the misclassification error.
Definitions: For a K-valued discrete random variable with probability mass function $p_i, i = 1, \ldots, K$ the Gini index is defined as: $\sum_{k=1}^{K} p_k(1 - p_k)$ and the cross-entropy is defined as $-\sum_{k=1}^{K} p_k \log p_k$.

$(\boldsymbol{c})$Let $f(p_k) = 1 - p_k + \log p_k$, where $p \in [0, 1]$ and the base of log is 2, then:

$$\frac{\partial f}{p_k} = -1 + \frac{1}{p_k \cdot \ln 2}$$

For all $p_k \in [0, 1]$, $\frac{\partial f}{p_k} > 0$, so the maximum point of f is at $p_k = 1$, where:

$$f(p_k)|_{p_k=1} = 1 - 1 + \log 1 = 0$$

Thus, $f(p_k) \leq 0$ as $p_k \in [0, 1]$. We get:

$$p_k(1 - p_k + \log p_k) \leq 0$$
$$i.e. \quad p_k(1 - p_k) \leq -p_k \log p_k \quad for \;\; p_k \in [0, 1]$$

In sum, $\sum_{k=1}^{K} p_k(1 - p_k) <= -\sum_{k=1}^{K} p_k \log p_k$, i.e. Gini index is guaranteed to be less than or equal to cross-entropy. ∎

5.(1.1) For each of 4 training datasets, i.e. data1, data2, data3,data4, train a linear regression model with and without outlier sample (the last one in each dataset). Then plot 1) the data points and 2) fitted lines in each subplot.

(1.1) The trained models with and without outlier for each dataset are as follows: ■

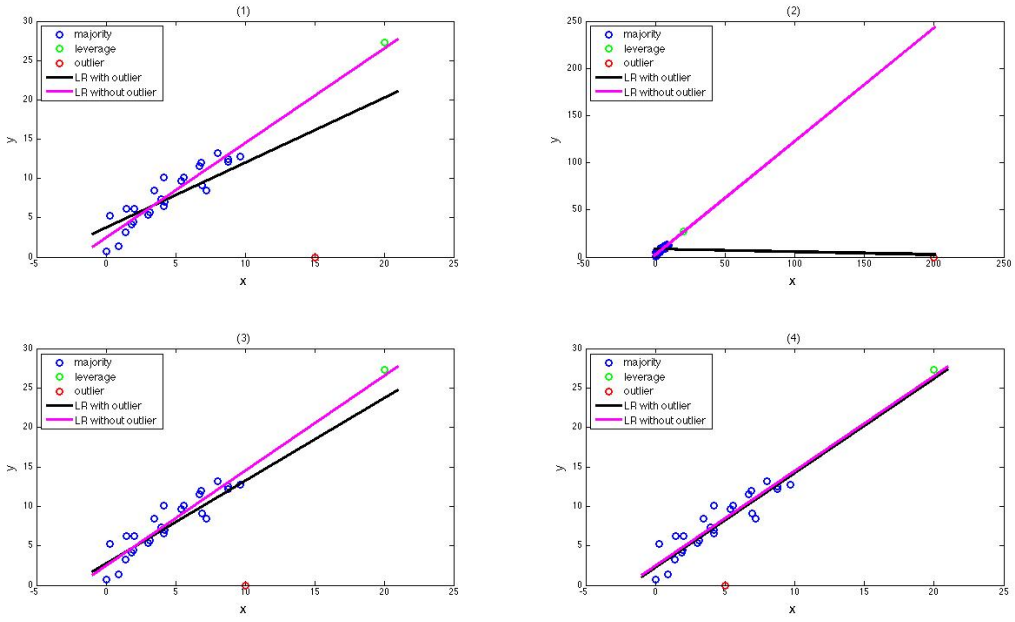

Figure 1: Linear Regression Models With and Without Outliesr

5.(1.2) For each of 4 training datasets, train linear regression models using weight decay coefficients $\lambda = 0.1, 1, 10$ respectively and plot data points and 3 fitted lines in each subplot. Could weight decay on parameter significantly reduce the influence of outlier sample? Why or why not?

(1.2) The trained models using weight decay coefficients for each dataset are as follows:

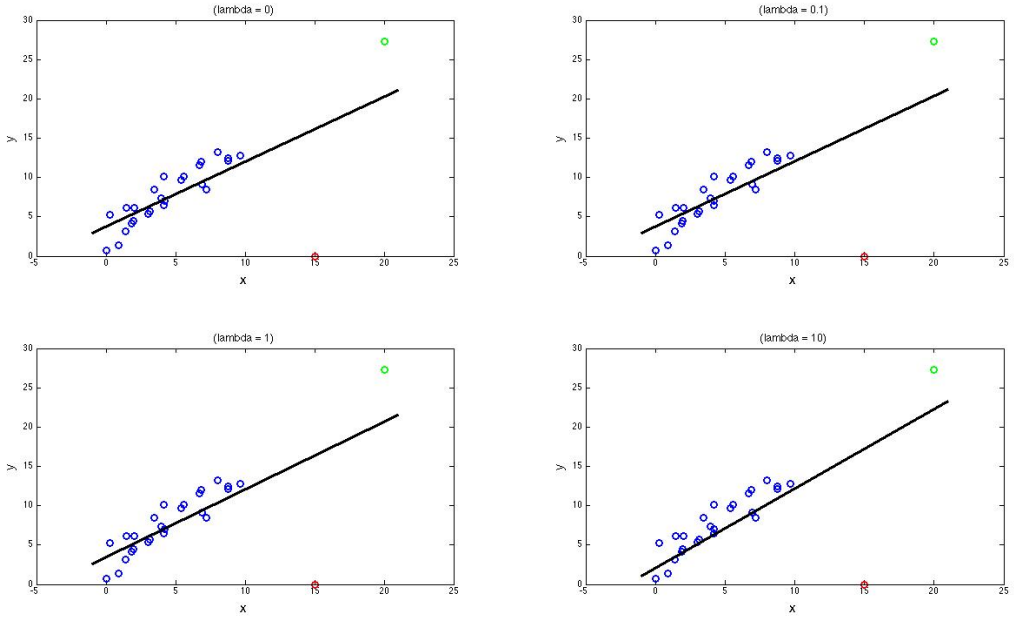

Figure 2: Ridge Regression with Different lambda for dataset 1

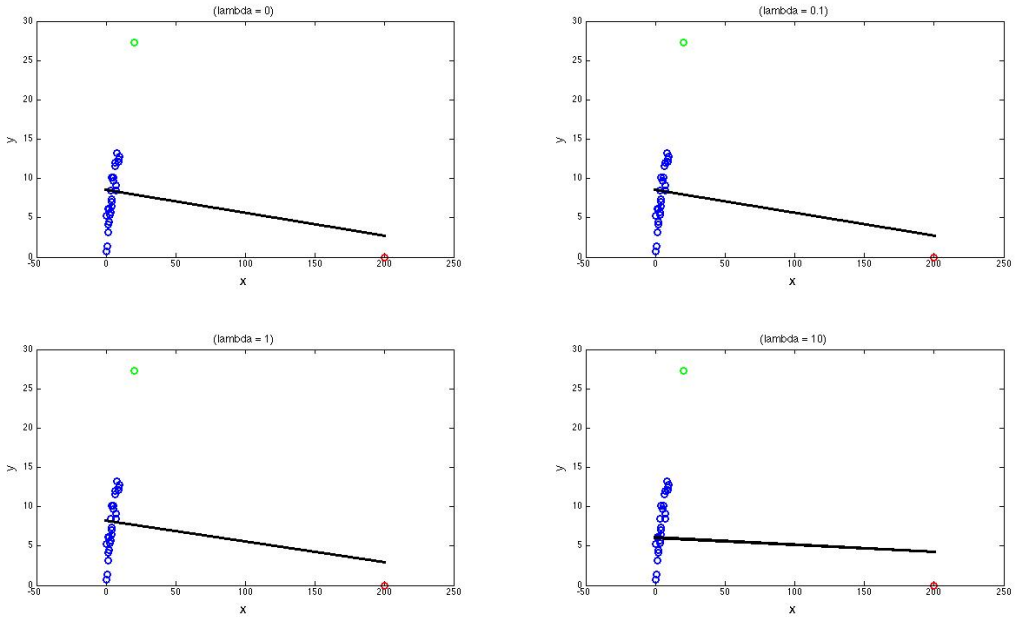

Figure 3: Ridge Regression with Different lambda for dataset 2

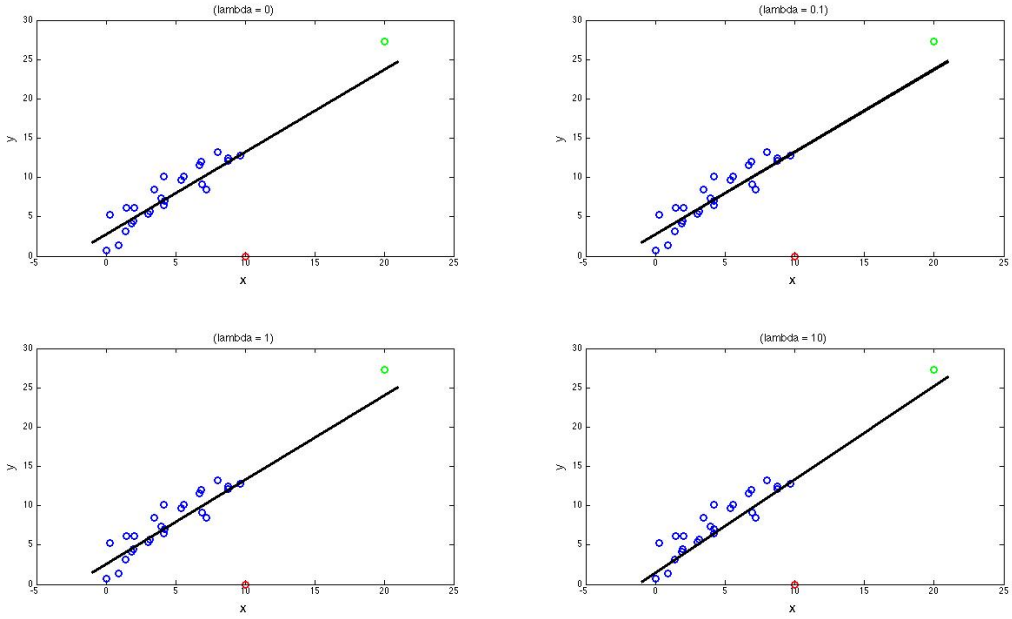

Figure 4: Ridge Regression with Different lambda for dataset 3

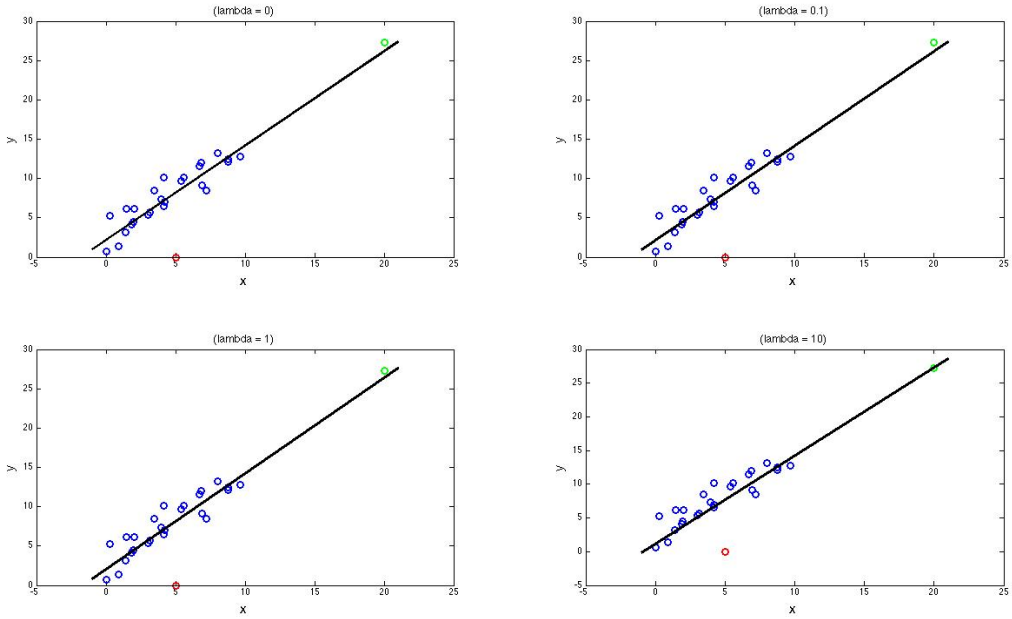

Figure 5: Ridge Regression with Different lambda for dataset 4

No, the weight decay on parameter cannot significantly reduce the influence of outlier sample. Weight decay, or ridge regression here just add bias in exchange to reduce variance, i.e., it is just a way to avoid overfitting by shrinking the parameters we trained. It is still very sensible to outliers, because we still depend on the least square minimization technique and this does not allow large residuals. Hence the regression line, plane or hyperplane will be drawn towards the outliers. Of course, when lambda is big, the trend would be reduced to some extent, but still not that big to significantly reduce the influence.

■

5.(1.3) If we model the noise in the regression models using zero-mean Laplace distribution, instead of Gaussian distribution, could we significantly reduce influence our outliers, i.e. achieve robustness to outliers? Why or why not? Could you write the learning objective function for linear regression in this case?

(1.3) Suppose there's noise $\eta \sim Laplace(0, b)$, so $y = \omega_0 + \omega^T x + \eta$, and $y \sim Laplace(\omega_0 + \omega^T x, b)$, then pdf is as follows:

$$f(y_n|x_n, b) = \frac{1}{2b} \cdot exp\{\frac{|y_n - (\omega_0 + \omega^T x_n)|}{b}\} \tag{47}$$

Based on (47), we can get the log-likelihood function as:

$$\begin{aligned} l(b) &= \log \prod_{n=1}^{N} P(y_n|x_n, b) \\ &= \sum_{n=1}^{N} \log P(y_n|x_n, b) \\ &= \sum_{n=1}^{N} -\log 2b - \frac{|y_n - (\omega_0 + \omega^T x_n)|}{b} \\ &= const - \sum_{n=1}^{N} \frac{|y_n - (\omega_0 + \omega^T x_n)|}{b} \end{aligned} \tag{48}$$

We can see from equation (48) that using Laplace noise, unlike Gaussian noise giving quadratic influence, the empirical loss for each sample point is kind of linear to the residual, which makes outlier have much less influence on it. So the conclusion is that by using Laplace distribution noise, we can reduce the influence of outliers. And the learning objective would be :

$$\hat{\omega_0} = min \ \ arg \sum_{n=1}^{N} \frac{|y_n - (\omega_0 + \omega^T x_n)|}{b} \tag{49}$$

$$\hat{\omega} = min \ \ arg \sum_{n=1}^{N} \frac{|y_n - (\omega_0 + \omega^T x_n)|}{b} \tag{50}$$

■

5.(1.4) For each data sample in 4 training datasets, calculate $\{h_i, t_i, d_i\}$ for each data point, select the samples with largest $\{h_i, t_i, d_i\}$ respectively and plot these points using $'cd'$, $'ms'$ and $'k >'$ shape & color. Could outlier sample in every dataset be picked out according to Cooks distance?

(1.4) Yes, according to what is shown in the figure as following: ■

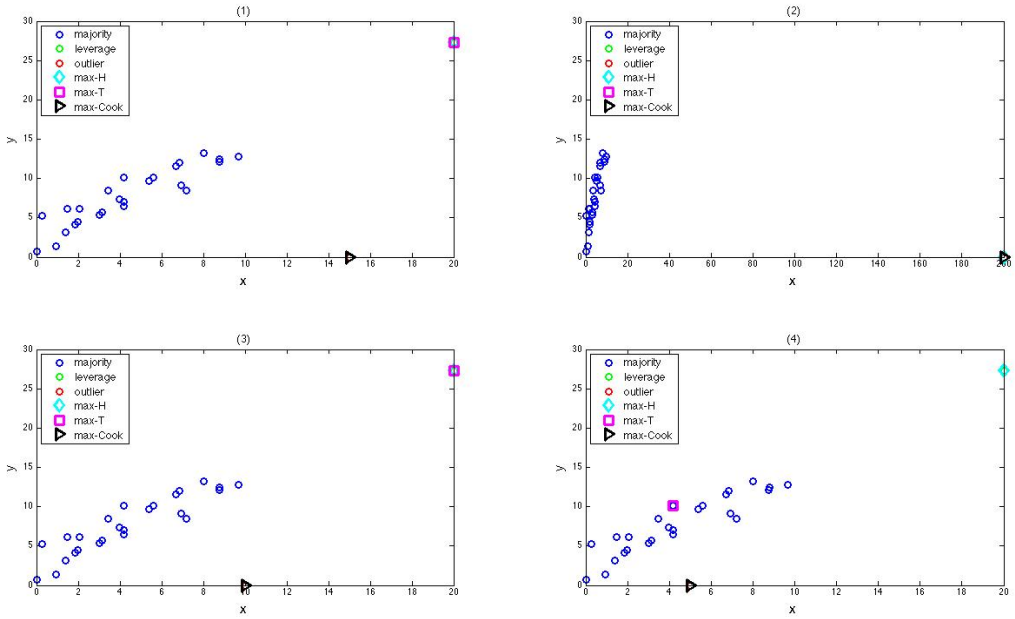

Figure 6: Samples with largest $\{h, t, d\}$

5.(2.a & 2.b) Data preprocessing and Implement KNN.

(2.*a*&2.*b*)

K	Training Accuracy	Validation Accuracy	Test Accuracy
1	0.583396	0.617021	0.567442
3	0.929057	0.617021	0.567442
5	0.905660	0.617021	0.567442
7	0.929057	0.617021	0.576744
9	0.942642	0.617021	0.567442
11	0.946415	0.617021	0.567442
13	0.946415	0.617021	0.567442
15	0.945660	0.617021	0.567442

Table 1: Accuracy for KNN

■

5.(2.c) Compare KNN and Decision Tree Algorithms.

(**2.***c*)

leaf #	Train-Gini	Validation-Gini	Test-Gini
1	0.978868	0.611702	0.627907
2	0.978868	0.611702	0.627907
3	0.978868	0.611702	0.627907
4	0.979623	0.680851	0.627907
5	0.978113	0.670213	0.627907
6	0.977358	0.654255	0.627907
7	0.977358	0.611702	0.637209
8	0.972830	0.611702	0.637209
9	0.959245	0.638298	0.595349
10	0.958491	0.664894	0.595349

Table 2: Gini Accuracy for Decision Tree

leaf #	Train-Entropy	Validation-Entropy	Test-Entropy
1	0.982642	0.638298	0.637209
2	0.982642	0.638298	0.637209
3	0.982642	0.638298	0.637209
4	0.982642	0.638298	0.637209
5	0.978113	0.627660	0.637209
6	0.977358	0.611702	0.637209
7	0.977358	0.611702	0.637209
8	0.977358	0.611702	0.637209
9	0.968302	0.638298	0.595349
10	0.967547	0.664894	0.595349

Table 3: Entropy Accuracy for Decision Tree

■

5.(2.d) Draw the decision boundary of given data.

(**2.***d*) The smoothness increases as K increases in general, the reason is that small K may lead to overfitting, large K helps to prevent overfitting. In other words, when K increases, the decision boundary will be smoother, because we are generalize the model itself. While if K is too large, then it will lead to underfitting, which should also be prevented. Thus K actually is a sensitive hyperparameter here, we must choose it carefully.

The decision boundaries for different K are as follows:

■

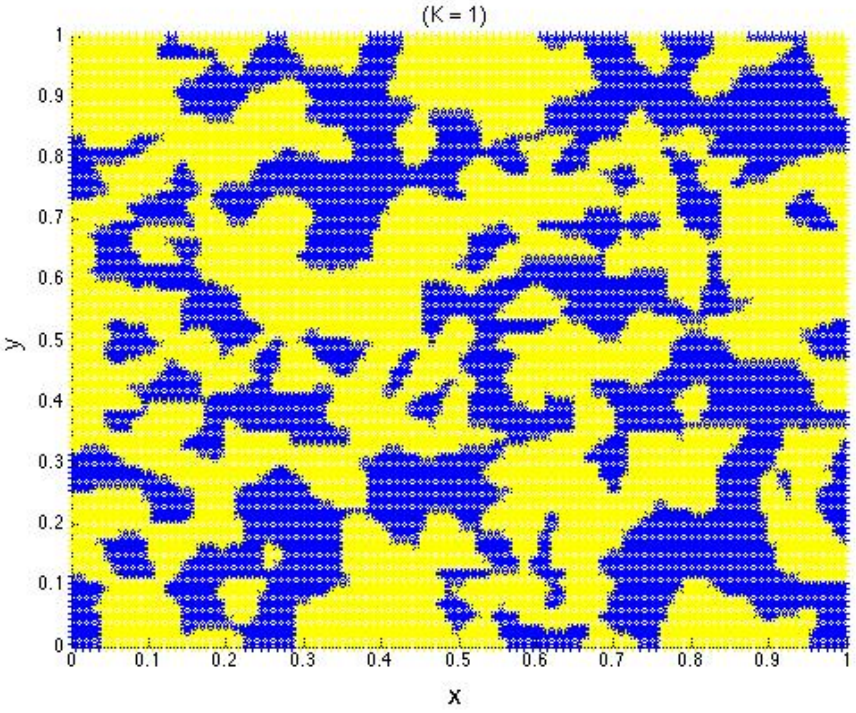

Figure 7: Decision Boundary with K=1

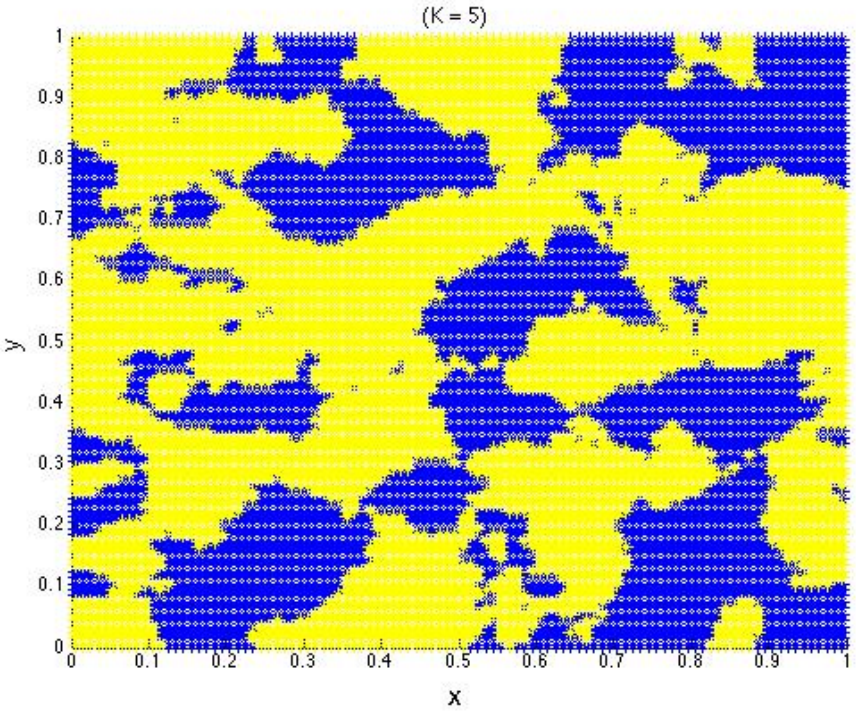

Figure 8: Decision Boundary with K=5

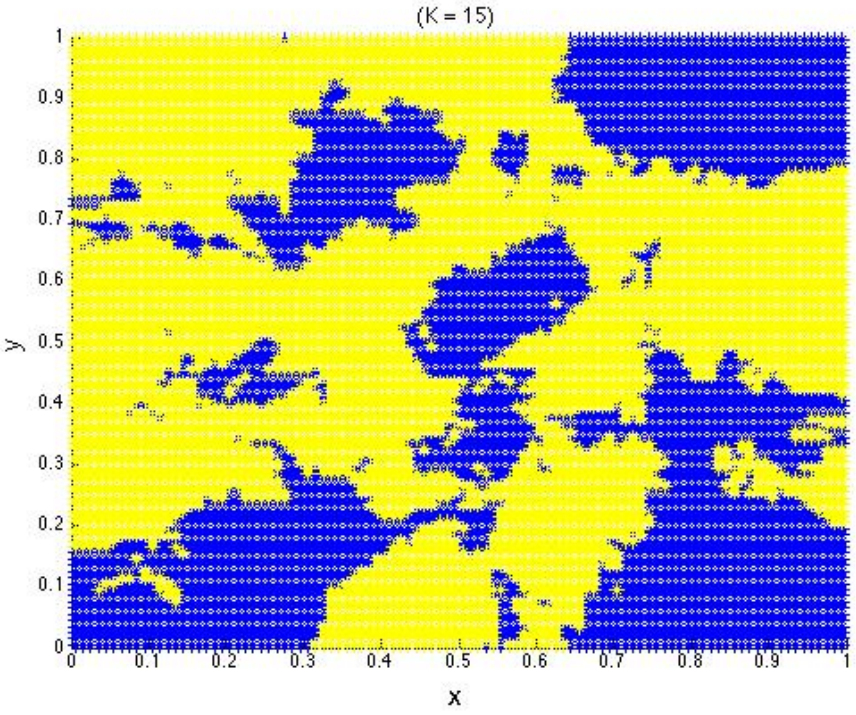

Figure 9: Decision Boundary with K=15

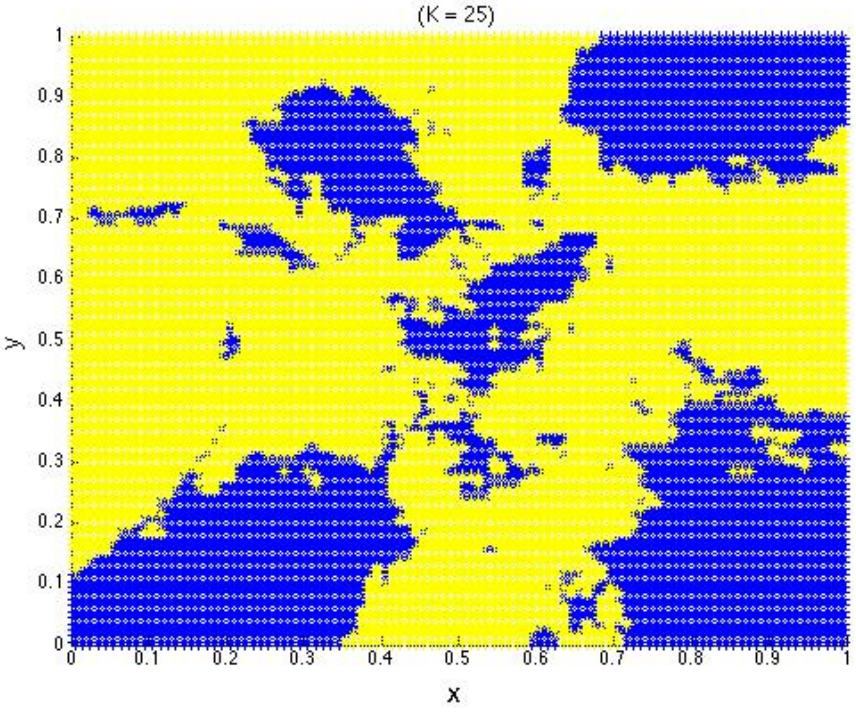

Figure 10: Decision Boundary with K=25