1.1

Given K(x, y) = $e^{-||x-y||^2/(2\sigma^2)}$.

We prove and use the following

a) Sum : If k1(x, y) and k2(x, y) are valid kernel then k1(x, y) + k2(x, y) is a valid kernel. : from slides
b) Product : If k1(x, y) and k2(x, y) are valid kernel then k1(x, y) * k2(x, y) is a valid kernel. : from slides
c) Positive scaling. If k1(x,y) is a valid kernel then a*k(x,y) is also valid kernel for a>0, since if φ is the feature map for k1(x,y), √a * φ is a valid feature map for a*k(x,y)
d) Any positive(or even non negative) constant is a valid kernel i.e.k(x,y)=a, for a>=0 is a valid kernel. Since we can construct a feature space $\phi(x)$, such that $\phi(x)$=[√a]
e) If k(x, y) is a valid kernel, then $e^{k(x,y)}$ is also a valid kernel.

   Since expanding the above by Taylor series expansion:

   $e^{k(x,y)} = \sum_{i=0}^{\inf} k(x, y)^i/i!$

   each of the term $k(x, y)^i/i!$ is a valid kernel since $k(x, y)^i$ is a valid kernel because we can apply product(b) to k(x,y) i times to get that. Since i! is just a constant > 0, we apply scaling (c) to $k(x, y)^i$ to get $k(x, y)^i/i!$ also as a valid kernel. For i=0, by (d) the constant 1 is also a valid kernel. Now, the summation for all i is again a valid kernel by (a).

   Thus the whole expression $e^{k(x,y)}$ is also a valid kernel.

Since $e^{k(x,y)}$ is a valid kernel, there exists a feature map $\phi : R^m -> H$ where H is a hilbert space, such that,

$e^{k(x,y)} = \phi(x)^T \phi(y)$

x^Ty is a valid kernel ($\phi(x)$ = x), by c, $x^Ty/(\sigma^2)$ is also a valid kernel. Let

$e^{x^Ty/(\sigma^2)} = \phi(x)^T \phi(y)$

So we have,

$$e^{-\|x-y\|^2/(2\sigma^2)} = e^{-\|x\|^2/2\sigma^2} e^{x^T y/(\sigma^2)} e^{-\|y\|^2/2\sigma^2} = e^{-\|x\|^2/2\sigma^2} \phi(x)^T \phi(y) e^{-\|y\|^2/2\sigma^2}$$

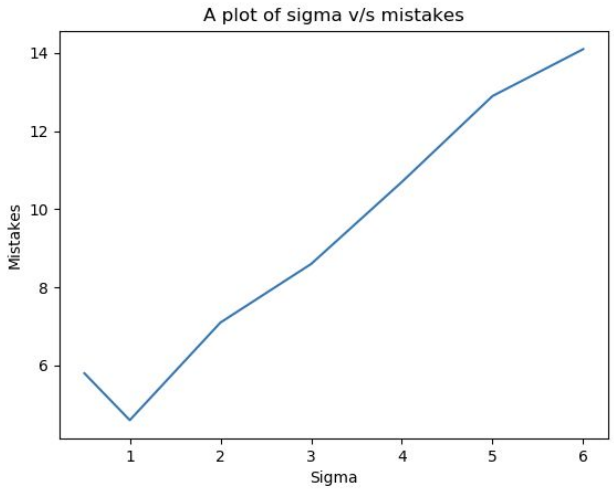
$$= \phi'(x)^T \phi'(y)$$

where $\phi'(x) = e^{-\|x\|^2/2\sigma^2} \phi(x)$.

Hence since a feature map ϕ' exists, $e^{-\|x-y\|^2/(2\sigma^2)}$ is a valid kernel.

1.2

(b) (ii) σ = 1 is the best choice since it minimizes the number of mistakes and maximises accuracy in k-fold cv.

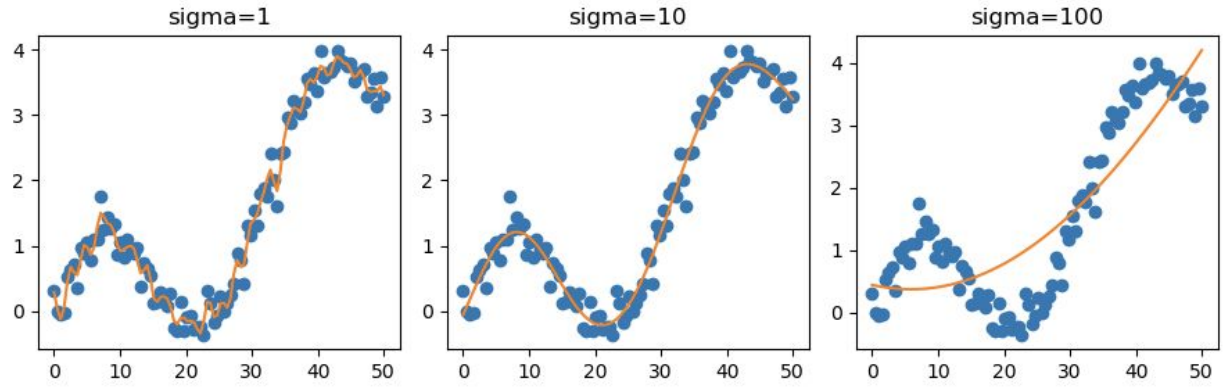


(b)(iii) sigma decides the contribution of the alpha of each train point to the prediction at the test point. Larger it is,more do we allow points farther away than the test point to contribute to the prediction(this kernel has square of Euclidean distance in exponent)

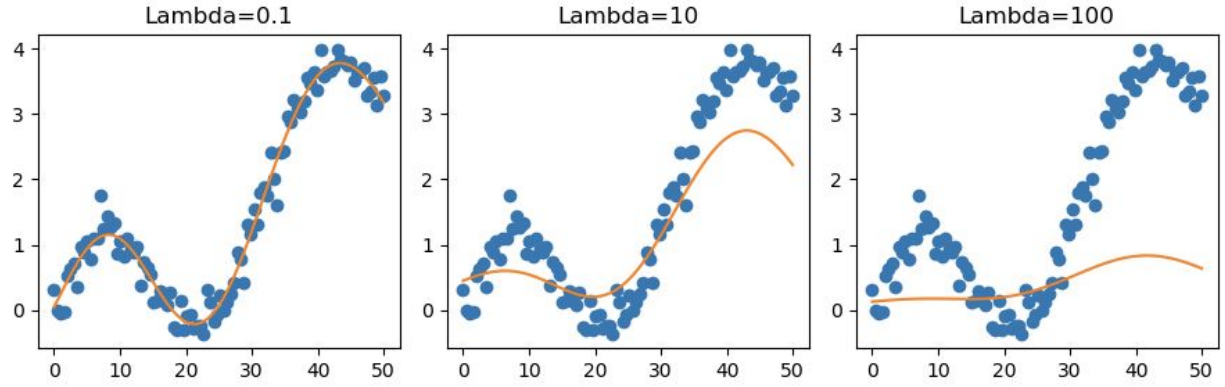
For small sigma, only the points(y) close to a given point (x) can make a significant contribution to the prediction, making it more of a local classifier. If sigma is high,the farther points also contribute, hence it tends towards a global classifier. A too large sigma leads to oversimplification taking into consideration the contribution of points far away also, leading to a smooth boundary(underfitting) and hence an increase in error.

For low sigma, only close points are considered, leading to a strict boundary which tends to overfit, and we start to fit on noise of points (overfitting). The graph shows this behaviour, mistakes increase for low and high values of sigma.

(c)(ii)



As shown in 1.2b(iii) above, high sigma means that the gaussian kernel is flatter leading to smoother regression curve whereas low sigma gaussian has a sharper peak, hence the regression curve depends on a small number of points around the given point and hence the curve is more erratic and curvy. This is observed in the plots, for low sigma=1, the regression curve is overfitting and is very curvy. For high sigma(100), the regression curve is too simple and smooth and hence underfits the data. The intermediate value of sigma=10 seems optimal.



Lambda is the regularisation parameter. Large lambda imposes penalty on the weight vector values and promotes a sparser weight vector, hence a smoother curve, preventing overfitting. Smaller lambda can lead to overfitting since the weight vector can become large and curvy and overfit data. The plots show this, as lambda increases, the

curve becomes flatter since the coefficients of the weight vector become close to 0, leading to worse fit as seen(underfitting). The lambda 0.1 seems optimal out of these, if we decrease lambda even more, it can lead to overfitting and a bad curvy fit.

2.1

(i) K(x, x') is a valid kernel. this implies there exists $\phi(x) : R^m -> H$ where H is a hilbert space, such that

$$K(x, x') = \phi(x)^T \phi(y)$$

Now,

$$K'(x, x') = K(g(x), g(x')) = \phi(g(x))^T \phi(g(x')) = \phi_g'(x)^T \phi_g'(x')$$

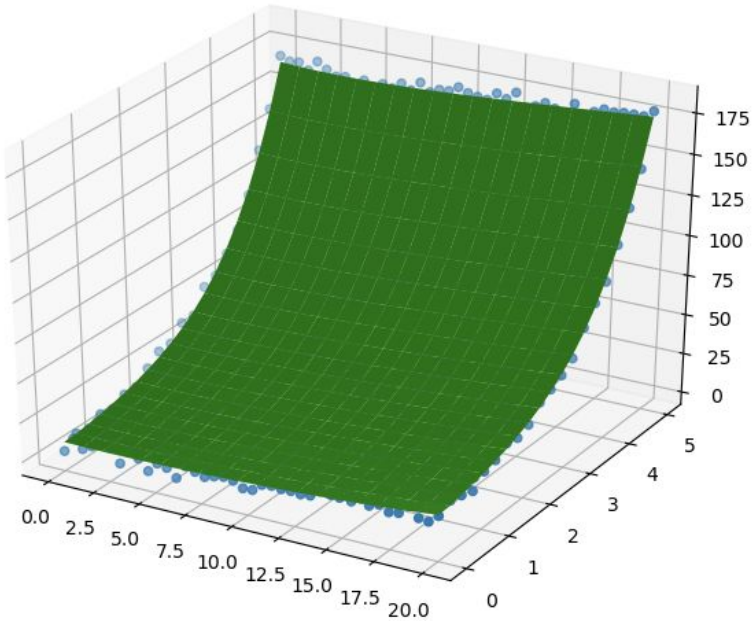
where $\phi_g'(x) = \phi(g(x))$, ϕ_g' is also a feature map from $R^m -> H$, since g is from $R^m -> R^m$. Due to the existence of this feature map ϕ_g', K'(x, x') = $K(g(x), g(x'))$ is also a valid kernel.

(ii) $q(K(x, x')) = \sum_{i=0}^{n} a_i * K(x, x')^i$

By using the properties mentioned in 1.1, each of the term $a_i * K(x, x')^i$ is a valid kernel. Since for a_i = 0, it is a valid kernel ($\phi(x)$ =[0])(property 1.1 d), for a_i != 0, suppose if i=0, since q has non-negative coefficients, by property (1.1 d) , it is a valid kernel . For i!=0, $K(x, x')^i$ is a valid kernel because we can apply product (1.1b) to K(x,x') i times to get that. Since a_i is just a constant > 0 (a_i = 0 already considered), we apply scaling (1.1 c) to $K(x, x')^i$ to get $a_i * K(x, x')^i$ also as a valid kernel. Now, the summation for all i is again a valid kernel by sum property (1.1a).

Hence $K'(x, x') = q(K(x, x'))$ is a valid kernel.

2.2



My design is polynomial kernel with the 4th power

$K(x, y) = (1 + x^T y)^4$

Polynomial kernel was tried with different powers, at the value of 4, the fit seemed good. At higher and lower powers, the error increases due to overfitting and underfitting respectively

3.1

Let centroid of first cluster: $c = (\Sigma^{m}_{i=1} x_i)/ m$
Let centroid of second cluster: $d = (\Sigma^{n}_{i=m+1} x_i)/ (n-m)$

For all points x belonging to first cluster:

$||x-c||^2 < ||x-d||^2$
$||x||^2 + ||c||^2 - 2x.c < ||x||^2 + ||d||^2 - 2x.d$
$2(d-c).\ x + (||c||^2 - ||d||^2) < 0$

Similarly, for points x belonging to second cluster:

$||x-c||^2 > ||x-d||^2$
$||x||^2 + ||c||^2 - 2x.c > ||x||^2 + ||d||^2 - 2x.d$
$2(d-c).\ x + (||c||^2 - ||d||^2) > 0$

Hence, the hyperplane a.x + b = 0 linearly separates the two clusters.

where, a = 2(d-c)

b = ($||c||^2$ - $||d||^2$)

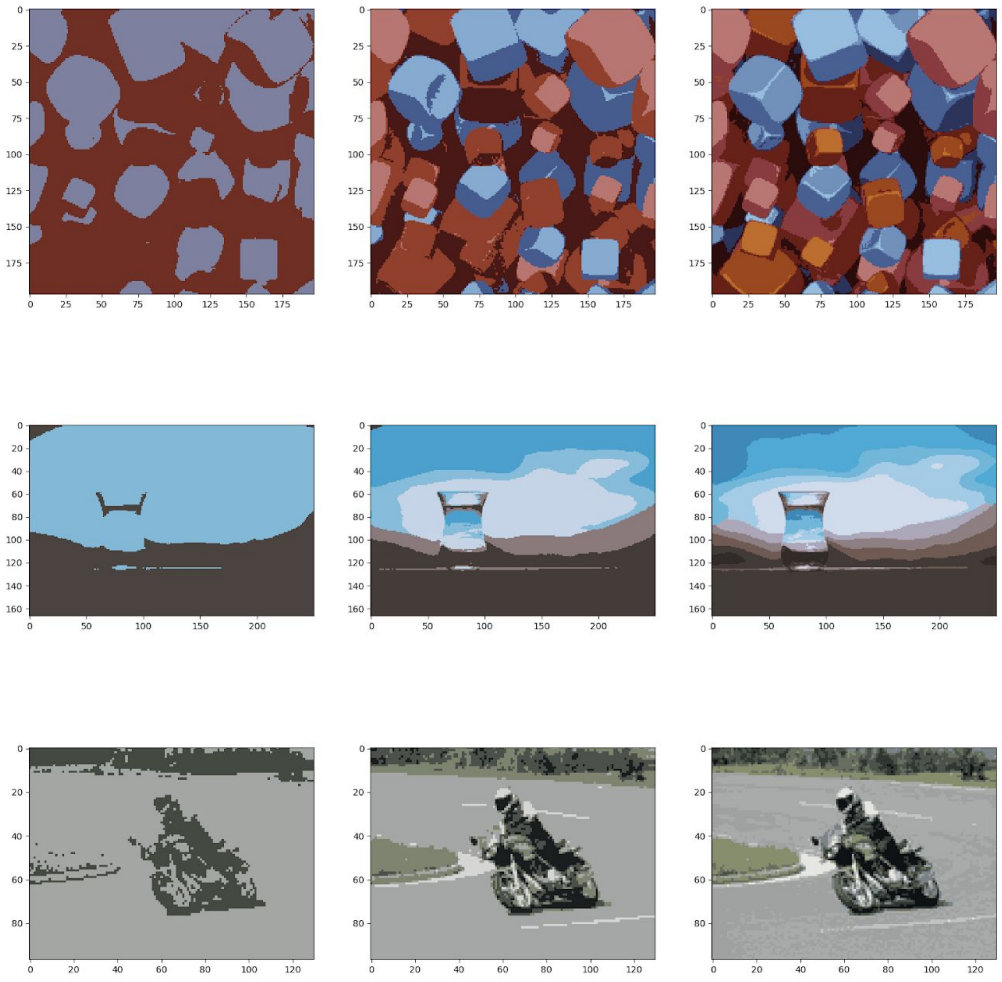
Where c = ($\Sigma^m_{i=1} x_i$)/ m and d =($\Sigma^n_{i=m+1} x_i$)/ (n-m)

Hence a= -2(($\Sigma^m_{i=1} x_i$)/ m - ($\Sigma^n_{i=m+1} x_i$)/ (n-m))

b= || ($\Sigma^m_{i=1} x_i$)/ m ||^2 - || ($\Sigma^n_{i=m+1} x_i$)/ (n-m) ||^2

3.2

(ii)



Pictures are in the order of k=2,5,10

As the number of clusters centres increase, the images get clearer since the number of distinct colours (r,g,b) that appear in the image increase. This gives space for better

representation of colors as ones that are closer to them. In some images having more variety of different colors and gradients, the increase in clarity is more.

(iii) The number of cluster centres in K-means denotes the number of distinct colors(r,g,b) that can be retained in the image. All other colors are mapped to the closest cluster centre and hence their original value is lost. Hence increasing the number of clusters allows more colors to be present, and also the remaining colors can be represented by a closer color than with the case of a smaller number of clusters(intuitively, on increasing k, the cost function of K-means decreases, leading to less loss in the image representation).

Hence images with a small number of colors (or similar colors), like image 1 can be represented quite well with a smaller cluster number like k=5,10. Increasing k from 5 to 10 doesn't affect much since k=5 already captures the distinct colors in the original image quite well.

The visible change of increasing k is more significant when the original image has lots of different colors (or shades), since then a large k allows more accurate representation of the colors of the image. Like in image 2, an increase from k=5 to k=10 brings in new shades of blue colors which were not able to be distinguished for smaller values of k. Similarly in image 3, increasing k allows better and better representation, since the original image has a wide variety of small patches of colors, which were earlier merged into single clusters for small k, but get better and better representation as k increases.