a)

$K = ak_1 + bk_2$

$VKV^T = V(ak_1 + bk_2)V^T$

$= V(ak_1)V^T + V(bk_2)V^T$

$= a\,Vk_1V^T + b\,Vk_2V^T$

k_1 and k_2 is valid kernel so, Vk_1V^T and $Vk_2V^T \geq 0$ and a and $b \in \mathbb{R}^+$ so, $VkV^T \geq 0$. Therefore k is semi-definite and valid kernel. TRUE

b) $K = k_1 - k_2$, then k is a valid kernel matrix.

$VKV^T = V(k_1 - k_2)V^T$

$= Vk_1V^T - Vk_2V^T$

If $Vk_2V^T > Vk_1V^T$, VkV^T may be not hold $VKV^T \geq 0$ so k is not valid kernel. FALSE

c) $k = k_1k_2$

if $k = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

$= \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$ $\quad k^T = \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix}$

which is not symmetric so not a valid kernel.

FALSE

d) $\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$ is valid kernel with negative entries so it is not necessary to have all enteries positive.

FALSE

e) $k(x_i,x_j) = k_1(x_i,x_j)\,k_2(x_i,x_j)$, then $k(x_i,x_j)$ is valid kernel function and $k(x_i,x_j)$ can be written as $\phi(x_i)^T\phi(x_j)$ such that $\phi(x) = \phi_1(x) \otimes \phi_2(x_c)$

$$k_1 = \sum_{d} \phi_{1d}(x_i)\,\phi_{1d}(x_j)$$

$$k_2 = \sum_{c} \phi_{2c}(x_i)\,\phi_{2c}(x_j)$$

$$k_1 k_2 = \sum_{d\in D} \phi_{1d}(x_i)\,\phi_{1d}(x_j) \sum_{c\in D} \phi_{2c}(x_i)\,\phi_{2c}(x_j)$$

$$= \left[\phi_{11}(x_i)\,\phi_{11}(x_j) + \phi_{12}(x_i)\,\phi_{12}(x_j) \ldots + \phi_{1D}(x_i)\,\phi_{1D}(x_j)\right]\left[\phi_{21}(x_i)\,\phi_{21}(x_j) + \phi_{22}(x_i)\,\phi_{22}(x_j) \ldots + \phi_{2c}(x_i)\,\phi_{2c}(x_j)\right]$$

$$= \Big[\phi_{11}(x_i)\phi_{11}(x_j)\phi_{21}(x_i)\phi_{21}(x_j) + \phi_{11}(x_i)\phi_{11}(x_j)\phi_{22}(x_i)\phi_{22}(x_j) + \ldots + \phi_{11}(x_i)\phi_{11}(x_j)\phi_{2c}(x_i)\phi_{2c}(x_j)$$

$$\phi_{12}(x_i)\phi_{12}(x_j)\phi_{21}(x_i)\phi_{21}(x_j) + \phi_{12}(x_i)\phi_{12}(x_j)\phi_{22}(x_i)\phi_{21}(x_j) + \ldots + \phi_{12}(x_i)\phi_{12}(x_j)\,\phi_{2c}(x_i)\,\phi_{2c}(x_j)$$

$$\ldots + \phi_{1D}(x_i)\phi_{1D}(x_j)\phi_{21}(x_i)\phi_{21}(x_j) + \ldots + \phi_{1D}(x_i)\phi_{1D}(x_j)\,\phi_{2c}(x_i)\,\phi_{2c}(x_j)\Big]$$

Arranging the terms

$$= \Big[\phi_{11}(x_i)\phi_{21}(x_i)\phi_{11}(x_j)\phi_{21}(x_j) + \phi_{11}(x_i)\phi_{22}(x_i)\phi_{11}(x_j)\phi_{22}(x_j) + \ldots \ldots + \phi_{1D}(x_i)\phi_{2c}(x_i)\phi_{1D}(x_j)\phi_{2c}(x_j)\Big]$$

$$= \left[\phi_{11}(x_i)\phi_{21}(x_i),\ \phi_{11}(x_i)\phi_{22}(x_i),\ \ldots\ \phi_{1D}(x_i)\phi_{2c}(x_i)\right]^T\left[\phi_{11}(x_j)\phi_{21}(x_j),\ \phi_{11}(x_j)\phi_{22}(x_j),\ \ldots\ldots + \phi_{1D}(x_j)\phi_{2c}(x_j)\right]$$

$$(\phi_1(x_i) \otimes \phi_2(x_i))^T(\phi_1(x_j) \otimes \phi_2(x_j))$$ TRUE.

NOW, $\langle c_+, c_+ \rangle = \langle \frac{1}{M_+} \sum_{i \in I^+} \phi(x_i), \frac{1}{M_+} \sum_{i \in I^+} \phi(x_i) \rangle$

$$= \frac{1}{2M_+^2} \sum_{i,j \in I^+} \phi(x_i)\phi(x_j)$$

$$= \frac{1}{M_+} \sum_{i,j \in I^+} k(x_i, x_j)$$

$$\langle c_-, c_- \rangle = \frac{1}{2M_-^2} \sum_{i,j \in I^-} k(x_i, x_j)$$

$$\therefore \; b = \frac{1}{2M_-^2} \sum_{i,j \in I^-} k(x_i, x_j) - \frac{1}{2M_+^2} \sum_{i,j \in I^+} k(x_i, x_j)$$

$$\langle \phi(x), c_+ - c_- \rangle = \phi(x)^T \left[\frac{1}{M_+} \sum_{i \in I^+} \phi(x_i) - \frac{1}{M_-} \sum_{i \in I^-} \phi(x_i) \right]$$

$$= \frac{1}{M_+} \sum_{i \in I^+} \phi(x)^T \phi(x_i) - \frac{1}{M_-} \sum_{i \in I^-} \phi(x)^T \phi(x_i)$$

$$= \sum_{i=1}^{m} \alpha_i \phi(x)^T \phi(x_i)$$

where $\alpha_i = \begin{cases} \frac{1}{m_+} & \text{if } y_i = +1 \\ \frac{-1}{m_-} & \text{if } y_i = -1. \end{cases}$

Then (1) become

$$h(x) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i k(x, x_i) + b \right)$$

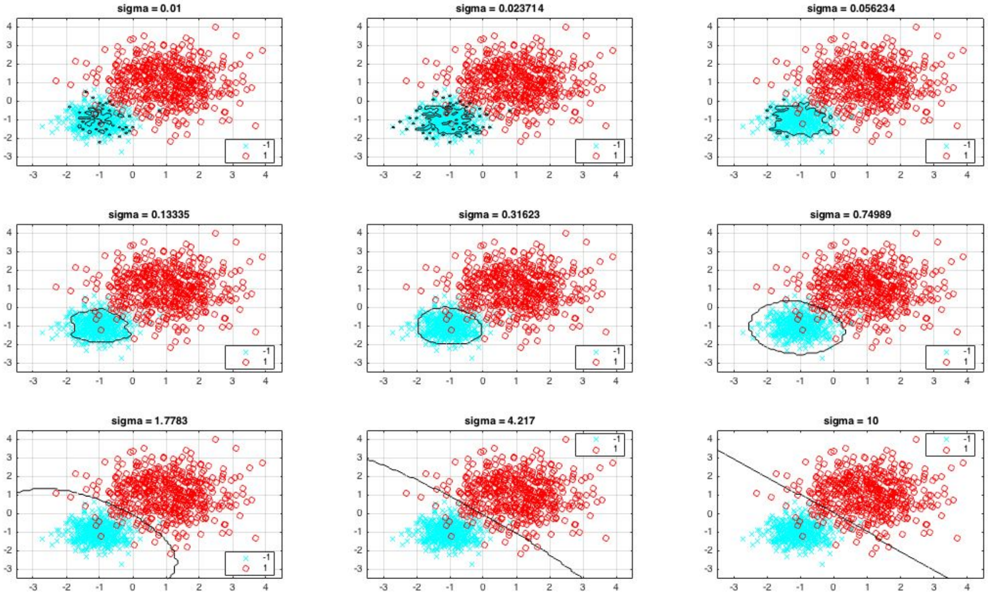
2

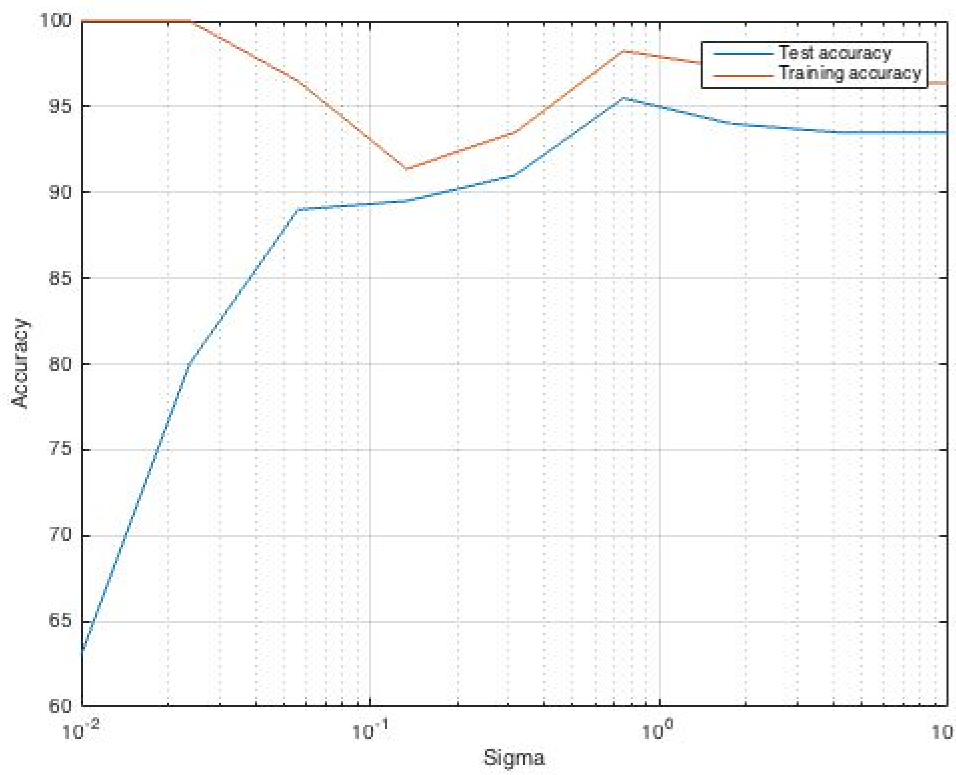
```
function K = gaussian_kernel(X, Z, sigma)
  [n, d] = size(X);
  [m, d] = size(Z);
  K = zeros(n, m);
  for i = 1:n
    x = X(i,:);
    individual_term = sum(bsxfun(@minus, Z, x).^2, 2);
    K(i,:) = individual_term'./ (-2 * sigma *sigma);
  end
  K = exp(K );
End
```

3.

```
function y_pred = parzen_classify(ktrain_train, ktrain_test, y)
  index_pos = find(y == 1);
  index_neg = find(y == -1);
  b_pos = 0;
  b_neg = 0;

  b_pos = sum(sum(ktrain_train(index_pos, index_pos),1)) ./ (2 * length(index_pos).^2);

  b_neg = sum(sum(ktrain_train(index_neg, index_neg),1)) ./ (2 * length(index_neg).^2);

  b = b_neg - b_pos;

  [n, m] = size(ktrain_test');

  alpha = ones(m, 1) ./ length(index_pos);
  alpha(index_neg) = -1./ length(index_neg);

  y_pred = sign((alpha' * ktrain_test) + b);

end
```

4.


sigma = 0.01
sigma = 0.023714
sigma = 0.056234
sigma = 0.13335
sigma = 0.31623
sigma = 0.74989
sigma = 1.7783
sigma = 4.217
sigma = 10


Test accuracy
Training accuracy
Accuracy
Sigma

First figure shows that as the sigma increases boundary line to separate the red and blue dots is getting larger and larger. More the sigma value more the spreadness of the boundary circle. As we see the first subplot with sigma 0.01 has small boundary. Second plot shows that first few sigma has high training error and low test error which mean model is overfit for first sigmas. Smaller the sigma value high change of model overfit. The optimal sigma for the given dataset is 0.74989.