AML Quiz 1 **August 30th, 2019** **Total Marks: 50**

Ans1. Assume $K_1(x,x')$ and $K_2(x,x')$ are kernels then three 'construction rules' can be defined as:

(scaling) $f(x)K_1(x,x')f(x'),\ \ f(x) \in R$

(Sum) $K_1(x,x') + K_2(x,x')$

(Product) $K_1(x,x')K_2(x,x')$

(a) (5 points) Let $\phi^{(1)}(x)$ and $\phi^{(2)}(x)$ be the feature vectors corresponding to kernels $K_1(x,x')$ and $K_2(x,x')$ respectively. These feature vectors may be of different lengths. Show that the product kernel $K_1(x,x')\ K_2(x,x')$ is a kernel by showing that its feature vectors are given explicitly by $\phi(x)$ whose $(i,\ j)$ component (doubly indexed vector) is $\phi_i^{(1)}(x)\phi_j^{(2)}(x)$.

Ans: $K(x,x') \ = \ K_1(x,x') \times K_2(x,x')$

$$= (\phi^{(1)}(x)^T \phi^{(1)}(x')) \times (\phi^{(2)}(x)^T \phi^{(2)}(x'))$$

$$= (\sum_{i=1}^{n_1} (\phi_i^{(1)}(x)\ \phi_j^{(1)}(x')) \times (\sum_{j=1}^{n_2} \phi_j^{(2)}(x)\ \phi_j^{(2)}(x'))$$

$$= \sum_{k=1}^{n_1 x\ n2} \phi_k(x)\phi_k(x')$$

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

which is a valid inner product. Thus, the product kernel, K(x, x0), is a valid kernel.

(b) (10 points) Use the construction rules to build a normalized cubic polynomial kernel

$$K(x,x') = (\ 1 + (\tfrac{x}{||x||})^T (\tfrac{x'}{||x'||}))^3$$

You can assume that you already have a constant kernel $K_0(x,x') = 1$ and a linear kernel $K_1(x,x') = x^T x'$. Identify which rules you are employing at each step.

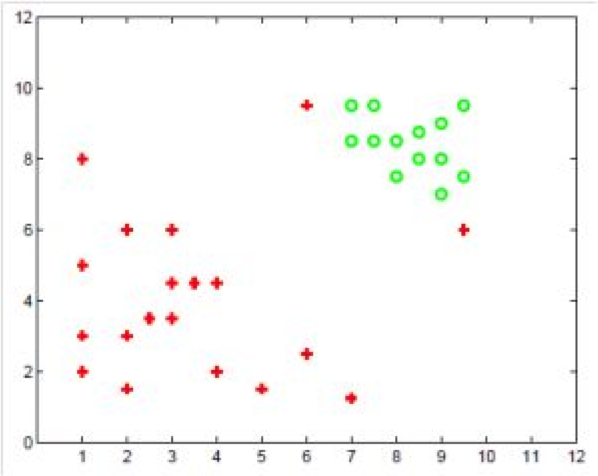
Ans: Step 1: Scaling. (Using $f(x) = (1/\|x\|)$, we get a new kernel $K_3(x,x')$

Step 2: Sum. $(K_4(x,x') = 1 + K_3(x,x'))$

Step 3: Product, twice. (to obtain $K_4(x,x')^3$)

Ans2 [Total 8, 2 for each]. The goal of this problem is to correctly classify test data points, given a training data set. You have been warned, however, that the training data comes from sensors which can be error-prone, so you should avoid trusting any specific point too much.

For this problem, assume that we are training an SVM with a quadratic kernel– that is, our kernel function is a polynomial kernel of degree 2. You are given the data set presented in Figure 1. The slack penalty C will determine the location of the separating hyperplane. Please answer the following questions qualitatively. Give a one-sentence answer/justification for each and draw your solution in the appropriate part of the Figure at the end of the problem.



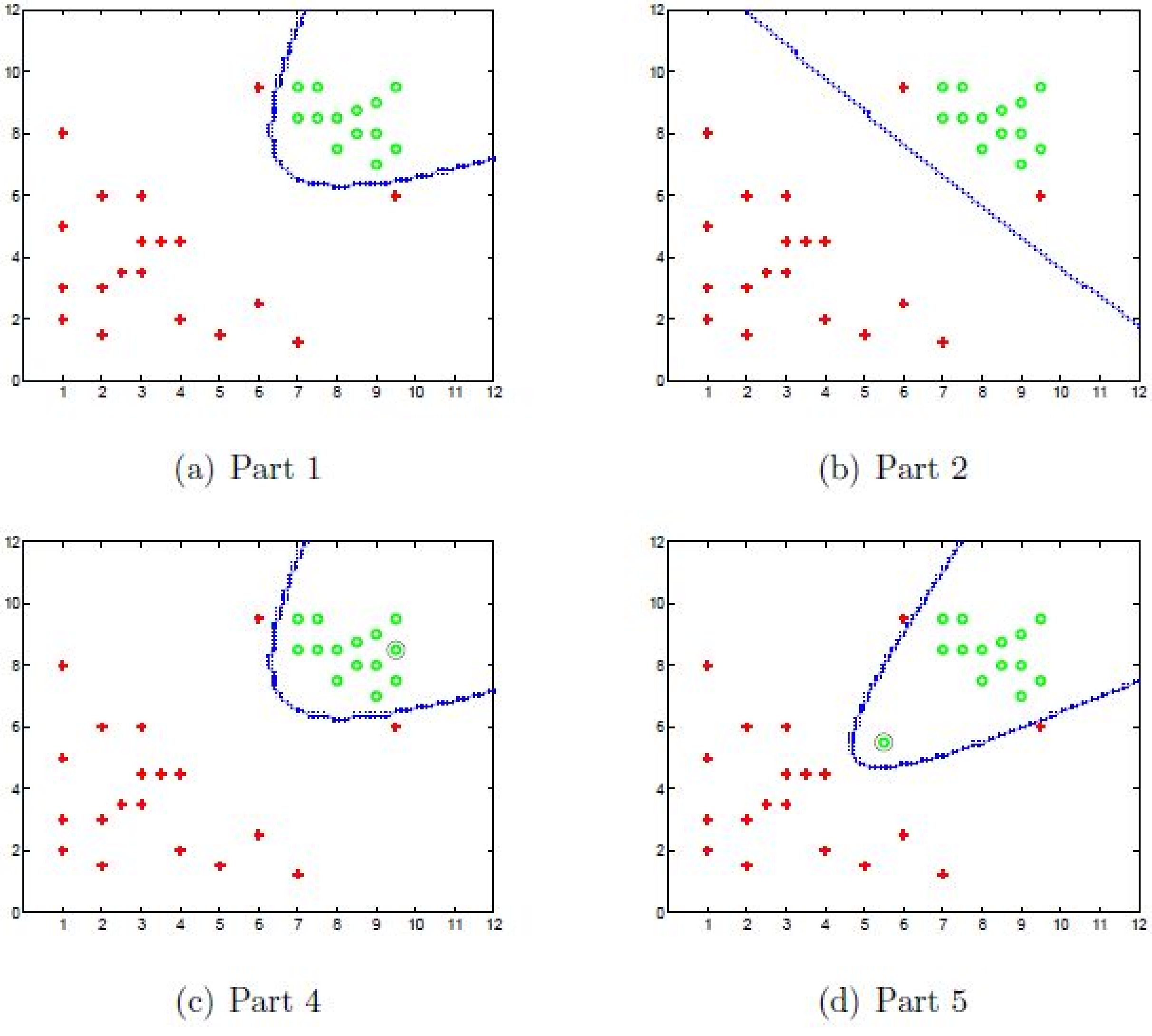
(a) Where would the decision boundary be for very large values of C (i.e., C → ∞)? (remember that we are using an SVM with a quadratic kernel.) Draw on the figure and justify your answer.

(b) For C ≈ 0, indicate in the figure below, where you would expect the decision boundary to be? Justify your answer.

1(c) Draw a data point that will not change the decision boundary learned for very large values of C. Justify your answer.

(d) Draw a data point that will significantly change the decision boundary learned for very large values of C. Justify your answer.

Ans:



(a) Part 1

(b) Part 2

(c) Part 4

(d) Part 5

Ans3 [8=5+3].

(a) SVM tries to maximize the margin between two classes. Therefore, the optimal decision boundary is diagonal and it crosses the point (3,4). It is perpendicular to the line between support vectors (4,5) and (2,3), hence it is slope is m = -1. Thus the line equation is

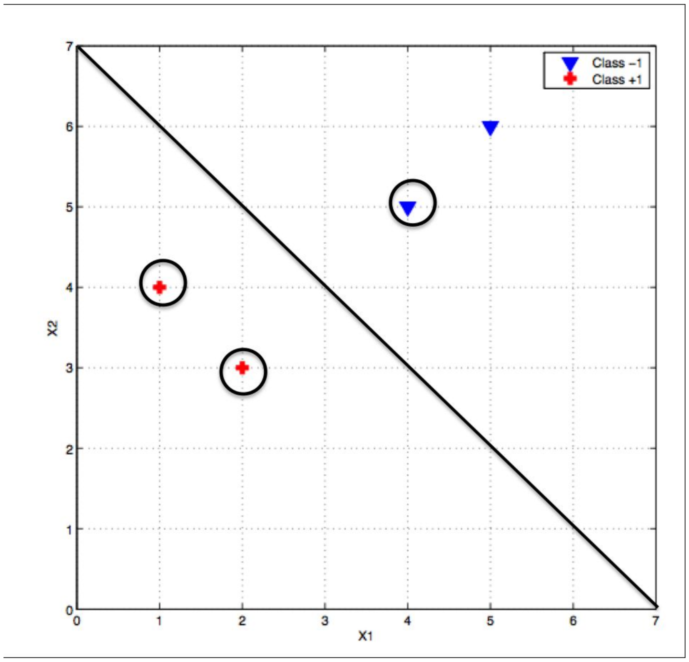
$(x2 - 4) = -1(x1 - 3)$

$x1 + x2 = 7$. From this equation, we can deduce that the weight vector has to be of the form (w1,w2), where w1 = w2. It also has to satisfy the following equations:

$2w1 + 3w2 + b = 1$ and $4w1 + 5w2 + b = -1$

Hence $w1 = w2 = -1/2$ and $b = 7/2$

(b)



Ans4 [Total 4, 1 for each point]. (Subjective i.e., not specific to following points)

The problem with the parameter C is:

1. that it can take any positive value

2. that it has no direct interpretation.

It is therefore hard to choose correctly and one has to resort to cross validation or direct experimentation to find a suitable value.

In response Schölkopf et al. reformulated SVM to take a new regularization parameter nu. This parameter is:

1. bounded between 0 and 1
2. has a direct interpretation

Interpretation of nu

The parameter nu is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors relative to the total number of training examples. For example, if you set it to 0.05 you are guaranteed to find at most 5% of your training examples being misclassified (at the cost of a small margin, though) and at least 5% of your training examples being support vectors.

Basically they are the same thing but with different parameters. The range of C is from zero to infinity but nu is always between [0,1]. A nice property of nu is that it is related to the ratio of support vectors and the ratio of the training error.

C ranges from 0 to infinity and can be a bit hard to estimate and use. A >modification to this was the introduction of nu which operates between 0-1 >and represents the lower and upper bound on the number of examples that >are support vectors and that lie on the wrong side of the hyperplane.

http://www.statsoft.com/textbook/support-vector-machines

https://stackoverflow.com/questions/11230955/what-is-the-meaning-of-the-nu-parameter-in-scikit-learns-svm-class

https://stats.stackexchange.com/questions/312897/c-classification-svm-vs-nu-classification-svm-in-e1071-r

2C SVM:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\gamma \sum_{i\in I_+} \xi_i + C(1-\gamma)\sum_{i\in I_-} \xi_i$$

$$\text{s.t.} \quad y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{for } \ i = 1, 2, \ldots, n.$$

The 2C-SVM assigns two different costs to each type of error: Cγ for a false negative and C(1-γ) for a false positive.

The cost asymmetry $\gamma\, \varepsilon$ [0, 1] controls the ratio of false positives and false negatives.

Dual Form:-

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_i \alpha_i$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C\gamma \quad \text{for } i \in I_+$$

$$0 \leq \alpha_i \leq C(1-\gamma) \quad \text{for } i \in I_-$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

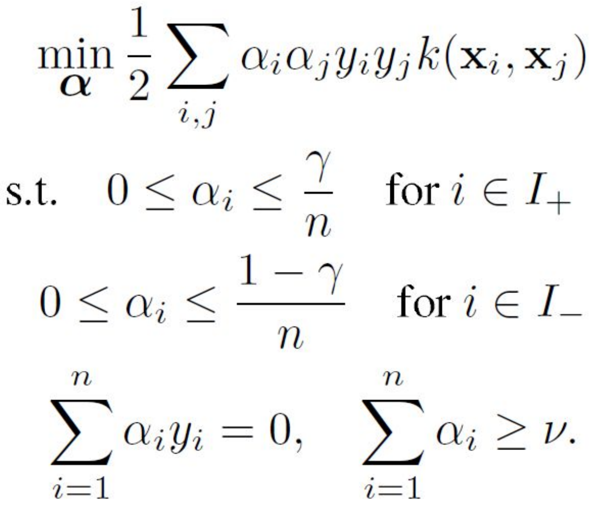
2nu-SVM

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\rho} \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{\gamma}{n}\sum_{i\in I_+} \xi_i + \frac{1-\gamma}{n}\sum_{i\in I_-} \xi_i$$

$$\text{s.t.} \quad y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) \geq \rho - \xi_i, \quad \xi_i \geq 0 \quad \text{for } i = 1, 2, \ldots, n$$

$$\rho \geq 0$$

Dual Form:-



$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq \frac{\gamma}{n} \quad \text{for } i \in I_+$$

$$0 \leq \alpha_i \leq \frac{1-\gamma}{n} \quad \text{for } i \in I_-$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \quad \sum_{i=1}^{n} \alpha_i \geq \nu.$$

Compared to the C in the standard SVM, ν has more intuitive meaning; precisely, ν serves as an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors. The ν-SVM, however, is proved to solve the same problem as the C-SVM.

http://web.eecs.umich.edu/~cscott/code/balanced2v-svm.pdf

Ans5 [Total 15, 5 for each].

Let S a finite set of points $x^1, \ldots.., x^l$ and let K_1 and K_2, be the corresponding kernel matrices obtained by restricting k_1 and k_2 to these points. Consider any vector $\alpha \varepsilon \Re^l$. Recall that a matrix K is positive semi-definite if and only if $\alpha' K \alpha \geq 0$, for all α.

(a)

Answer 1

$$\alpha'(K_1 + K_2)\alpha = \alpha' K_1 \alpha + \alpha' K_2 \alpha \geq 0$$

And so $K_1 + K_2$ is a positive semi-definite and $k_1 + k_2$ a kernel function.

Answer 2

- Given any final set of instances {x_1,..., x_n}, let K_1 (resp., K_2) be the n x n Gram matrix associated with k_1 (resp., k_2). The Gram matrix associated with k_1 + k_2 is just K = K_1 + K_2.
- K is PSD because any $v \in R_n$, $v_T (K_1 + K_2)v = (v_T K_1 v) + (v_T K_2 v) \geq 0$ as $V^T K_1 v \geq 0$ and $v^T K_2 v \geq 0$ follows from K_1 and K_2 being positive semidefinite.
- K is valid kernel

(b)

Ans 1: Similar to Ans 1 of (a)

$\alpha' a K_1 \alpha = a \alpha' K_1 \alpha \geq 0$ verifying that ak_1 is a kernel

$$k(x, y) = (\sqrt{a}\phi_1^1(x), ..., \sqrt{a}\phi_N^1(x))(\sqrt{a}\phi_1^1(y), ..., \sqrt{a}\phi_1^N(y) = ak_1(x, y)$$

Ans 2:

Assuming that $a \geq 0$,

Let Gram matrix for aK1(x,y) is K. Now K is PSD because for any $v \in R_n$,
$v_T (aK_1(x,y))v = a(v_T K_1(x,y)v) \geq 0$ that implies K is PSD hence a valid kernel.

(c)

Ans 1:

$$K = K_1 \otimes K_2$$

be the tensor product of the matrices K_1 and K_2 obtained by replacing each entry of K1 by K2 multiplied by that entry. The tensor product of two positive semi-definite matrices is itself positive semidefinite since the eigenvalues of the product are all pairs of products of the eigenvalues of the two components. The matrix corresponding to the function $k_1 k_2$ is known as the Schur product H of K_1 and K_2 with entries the products of the corresponding entries in the two components. The matrix H is a principal submatrix of K defined by a set of columns and the same set of rows. Hence for any $\alpha \in \Re^l$, there is a corresponding $\alpha_1 \in \Re^{l^2}$, such that

$\alpha H \alpha = \alpha_1 ' K \alpha_1 \geq 0,$ and so H is positive semi-definite as required.

Ans 2:

First show that C s.t. $C_{ij} = A_{ij} \times B_{ij}$ is PSD:

One way to show it:

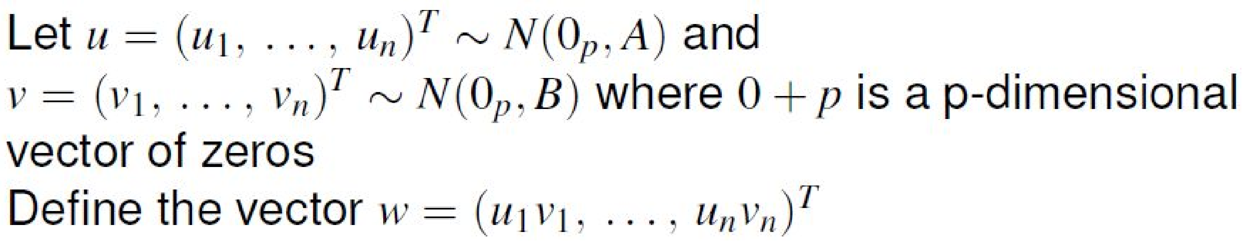
Any PSD matrix Q is a covariance matrix.

Any PSD matrix Q is a covariance matrix. To see this, think of a p-dimensional random variable x with a covariance matrix I_p, the identity matrix. (Q is $p \times p$) Because Q is PSD it admits a non-negative symmetric square root $Q^{1/2}$. Then:

$$cov(Q^{1/2})\, x = Q^{1/2}\, cov(x))\, Q^{1/2} = Q^{1/2} I\, Q^{1/2} = Q$$

And therefore Q is a covariance matrix.

We also know that any covariance matrix is PSD. So given A and B PSD, we know that they are covariance matrices. We want to show that C is also a covariance matrix and therefore PSD.



Let $u = (u_1, \ldots, u_n)^T \sim N(0_p, A)$ and
$v = (v_1, \ldots, v_n)^T \sim N(0_p, B)$ where $0 + p$ is a p-dimensional vector of zeros
Define the vector $w = (u_1 v_1, \ldots, u_n v_n)^T$

$$cov(w) = E[(w - \mu^w)(w - \mu^w)^T] = E[ww^T]$$

This is because $\mu_i^w = 0$ for all i. This is because u and v are independent so $\mu^w = \mu^u \times \mu^v = 0_p$

$$\begin{aligned} cov(w)_{i,j} &= E[w_i w_j^T] = E[(u_i v_i)(u_j v_j)] = E[(u_i u_j)(v_i v_j)] \\ &= E[u_i u_j]\,E[v_i v_j] \end{aligned}$$

This is again because u and v are independent.

$$cov(w)_{i,j} = E[u_i u_j]\,E[v_i v_j] = A_{i,j} \times B_{i,j} = C_{i,j}$$

Therefore C is a covariance matrix and therefore PSD

Since any kernel matrix created from

$k(x_i, x_j) \ = k_1(x_i, x_j) \times k_2(x_i, x_j)$ is PSD, then k is PSD