STAT 37710 Homework 4.

1. By the definition of p.s.d.,

symmetric matrix K is p.s.d. when $v^T K v \geq 0$, for any $v \in \mathbb{R}^d$

Since K is symmetric matrix, K can be written as $K = P(\lambda_1 \cdots \lambda_d) P^T$,

where P is orthogonal matrix, i.e. $K = \sum_{i=1}^{d} \lambda_i u_i u_i^T$ $\quad P = (u_1, u_2, \cdots u_d)$

Thus, $v^T K v = \sum_{i=1}^{d} \lambda_i v^T u_i u_i^T v = \sum_{i=1}^{d} \lambda_i (v^T u_i)^2$

Since $v^T K v \geq 0$ is true for any $v \in \mathbb{R}^d$,

$\iff \sum_{i=1}^{d} \lambda_i (v^T u_i)^2 \geq 0$ for any $v \in \mathbb{R}^d$.

This inequality holds if and only if $\lambda_i \geq 0$, $i = 1, 2, \cdots d$.

i.e. symmetric matrix K is p.s.d. if and only if its eigenvalues are nonnegative

2. Consider the matrix, $M = \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T$

where $\lambda_1 = 4$ $\lambda_2 = -2$, $v_1 = \frac{1}{\sqrt{2}}(1, 1)^T$, $v_2 = \frac{1}{\sqrt{2}}(1, -1)^T$

$$M = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$

M is positive, but not p.s.d. (since its eigenvalue $-2 < 0$)

Let $\mathcal{X}$ be two-element space $\mathcal{X} = (1, 2)$ define $k(1,1) = k(2,2) = 1$, $k(1,2) = k(2,1) = 3$.

This function is symmetric, but not positive semidefinite,

The linear kernel $k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ define $k(x, x') = x^T x'$, this kernel is p.s.d., but not positive.

3. Let m be any number $x_1, \cdots x_m \in \mathcal{X}$, and $a_1, \cdots a_m \in \mathbb{R}$, Let

$y_i = \phi(x_i) \in \mathcal{Y}$, for $i = 1, 2, \cdots, m$, Then

$$\sum_{i=1}^{m}\sum_{j=1}^{m} a_i a_j k(x_i, x_j) = \sum_{i=1}^{m}\sum_{j=1}^{m} k'(y_i, y_j)$$

Since k' is psd on $\mathcal{X}'$, $\sum_{i=1}^{m}\sum_{j=1}^{m} k'(y_i, y_j) \geq 0$.

Therefore, $\sum_{i=1}^{m}\sum_{j=1}^{m} a_i a_j k(x_i, x_j) \geq 0$ for any $x_1, \dots x_m$ and $a_1, \dots a_m$.

i.e. k is psd on $\mathcal{X}$.

4. (a) For any $m \geq 1$, $x_1, \dots x_m \in \mathcal{X}$, and $c_1, \dots c_m \in \mathbb{R}$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j k(x_i, x_j) = \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j k_1(x_i, x_j)}_{①} + \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j k_2(x_i, x_j)}_{②}$$

① ≥ 0 since k_1 is psd on $\mathcal{X}$,

② ≥ 0 since k_2 is psd on $\mathcal{X}$,

$\Rightarrow$ ① + ② ≥ 0

Therefore, k is psd on $\mathcal{X}$.

(b) For any $m \geq 1$, and $(x_1^1, x_1^2), \dots (x_m^1, x_m^2) \in \mathcal{X} \times \mathcal{X}$, and $c_1, \dots c_m \in \mathbb{R}$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j k_\oplus((x_1, x_2), (x_1', x_2')) = \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{m} k_1(x_i^1, x_j^1)}_{①} + \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{m} k_2(x_i^2, x_j^2)}_{②}$$

① ≥ 0 since k_1 is psd.

② ≥ 0 since k_2 is psd.

Therefore k is psd.

(c) $k_\angle(x, x') = x^T x' / (\|x\| \|x'\|)$, let $\varphi(x) = \frac{x}{\|x\|}$

$k_\angle(x, x') = k_0(\varphi(x), \varphi(x'))$, k_0 is linear kernel. k_0 is psd.

By the result of problem 3,

$k_\angle$ is also psd.

3. counterexample. $X = \{(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}), (\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}), (+\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})\}$

def $\phi: R^2 \to R^2$.

$$(x,y) \to \begin{cases} (2x, 2y) & \text{if } x \geq 0,\ y \geq 0 \\ (-2x, -2y) & \text{if } x \geq 0,\ y < 0 \\ (3x, -3y) & \text{if } x < 0,\ y \geq 0 \\ (x, y) & \text{otherwise.} \end{cases}$$

def $k: R^2 \times R^2 \to R$

$(x,y) \to \langle \phi(x), \phi(y) \rangle$, where $\langle , \rangle$ is inner product on R^2.

Def $T_G = \{i, e\}$, i.e. $i(x_1, x_2) = i(x_1, x_2)$ $e(x) = (x_2, x_1)$

Thus, for X, kernel matrix for k_{max} is

$$k = \begin{pmatrix} 4 & -2 & -4 \\ -2 & 1 & 3 \\ -4 & 3 & 9 \end{pmatrix}$$

which has negative eigenvalue, $\Rightarrow$ k_{max} isn't p.s.d.

$$k_G(x, x') = \sum_{g, g' \in G} k(T_g(x), T_{g'}(x'))$$

$$= \sum_{g, g' \in G} k(T_g(T_{g_0}(x)), T_{g'}(T_{g_0'}(x')))$$

$$= \sum_{g, g' \in G} k(T_{g g_0}(x), T_{g' g_0'}(x'))$$

$$= \sum_{g, g' \in G} k(T_g(x), T_{g'}(x')) \quad \text{(by property of symmetry group)}$$

Also, for any c_i, c_j.

$$\sum_i \sum_j c_i c_j k_G(x_i, x_j) = \sum_{g, g' \in G} \underbrace{\sum_i \sum_j c_i c_j k(T_g(x), T_{g'}(x'))}_{\geq 0},$$

$\therefore \sum_i \sum_j c_i c_j k_G(x_i, x_j) \geq 0$

$\therefore k_G(x, x')$ is psd.

Scanned by CamScanner

RKHS:

1. Define $k_G: \mathcal{X} \to R^{|G|}$

$k_G(x') = \sum_{g \in G} k(T_g(x), x')$

2. Define H_k^{pre} as the space of linear combinations

$f(x) = \sum_{i=1}^{m} \alpha_i k_{x_i}(x)$

for any $m \in N$, any $x_1, \ldots x_m \in \mathcal{X}$, and $\alpha_1, \ldots \alpha_m \in R$.

3. Define $\langle k_x, k_{x'} \rangle = k_G(x, x')$ and extend it to the rest of H_k^{pre} by linearity.

4 Add to H_k^{pre} the limits points of all Cauchy sequences

Here, we prove $k_{max}(T_G(x), T_G(x')) = \max_{g_a, g_b \in G} k(T_{g_a}(x), T_{g_b}(x'))$

$$= \max_{g, g' \in G} k(T_g(x), T_{g'}(x'))$$

$$= k_{max}(x, x')$$

since $G_a = G_b G$

Thus, we get k_{max} is invariant to T_G.

$$k_G(T_g(x), x') = \sum_{g, g' \in G} k(\overset{T_{g g_0}(x),}{\cancel{\ldots}}\, T_{g'}(x')) = \sum_{g, g' \in G} k(T_g(x), T_{g'}(x')) = k_G(x, x')$$

$\therefore h(T_g(x)) = h(x)$

6. (a) The intuitive meaning of this kernel is to count the number of contiguous substrings of length k that are both included in string $\bar{a}$ and $\bar{b}$,

For any m, $c_1, c_2, \ldots c_m$ and $\bar{x}_1, \bar{x}_2, \ldots \bar{x}_m$, we have

$$\sum_i \sum_j c_i c_j k_K(\bar{x}_i, \bar{x}_j) = \sum_{\bar{s}\in\Sigma^k} \sum_{i=1}^{m} \sum_{j=1}^{m} c_i c_j n_{\bar{s}}(\bar{x}_i) n_{\bar{s}}(\bar{x}_j)$$

$$= \sum_{\bar{s}\in\Sigma^k} \sum_{i=1}^{m} c_i n_{\bar{s}}(\bar{x}_i) \sum_{j=1}^{m} c_j n_{\bar{s}}(\bar{x}_j)$$

$$= \sum_{\bar{s}\in\Sigma^k} \left(\sum_{i=1}^{m} c_i n_{\bar{s}}(\bar{x}_i)\right)^2 \geq 0$$

Thus, it's psd by definition

Pseudocode.

Let a and b be the strings, la is the length of a, lb is the length of b.

```
count = 0
for i=0 : la-k
    for j=0 : lb-k
        if a(i:(i+k-1)) == b(j:(j+k-1))
            count = count+1
        end
    end
end
```

(b) The intuitive meaning of this kernel here is the number of gapped strings of length k that are both included in the both strings

For any m, $c_1, c_2, \ldots c_m$, and $\bar{x}_1, \bar{x}_2, \ldots \bar{x}_m$.

$$\sum_{i=1}^{m} \sum_{j=1}^{m} c_i c_j k_k^*(\bar{x}_i, \bar{x}_j) = \sum_{\bar{s}\in\Sigma^k} \sum_{i=1}^{m} \sum_{j=1}^{m} c_i c_j n_{\bar{s}}^*(\bar{x}_i) n_{\bar{s}}^*(\bar{x}_j)$$

$$= \sum_{\bar{s}\in\Sigma^k} \sum_{i=1}^{m} c_i n_{\bar{s}}^*(\bar{x}_i) \sum_{j=1}^{m} c_j n_{\bar{s}}^*(\bar{x}_j)$$

$$= \sum_{\bar{s}\in\Sigma^k} \left(\sum_{i=1}^{m} c_i n_{\bar{s}}^*(\bar{x}_i)\right)^2 \geq 0$$

Thus, it's p.s.d.

(i) Here, we could count the number of each character in the alphabet and then multiply them. This is equivalent.

Establish a mapping function $g: \Sigma \to \{0, 1, \ldots, n-1\}$. map 'a' to 0, 'b' to 1, .. 'z' to 25

Denote $|\Sigma| = n$

Pseudo code:

```
m = zeros(n, 1)
n = zeros(n, 1)
for i=0:(la-1)
    m(g(a(i))) = m(g(a(i)))+1.

for j=0:(lb-1)
    n(g(b(j))) = n(g(b(j)))+1

count=0
for i=0:(n-1)
    count = count + m(i)*n(i)
~~end~~
```

count is value of this kernel.

(ii) Since the length of both strings a and b are both p, they only have one possible gapped strings of length p. Hence, what we need to do is to check they are ~~equal~~ same or not.

pseudo code.

```
res=1
for i=0:p-1,
    if a(i)!=.b(i)
        res=0
        break.
return res.
```

Scanned by CamScanner

(iii)

(A) For any gapped strings, ~~they~~ there are only two cases,

① it doesn't involving the last elements of $a_{0:N+1}$, ~~which is a_{N+1}~~

② it ~~doesn't~~ involves the last element of $a_{0:N+1}$, which is a_{N+1}.

For situation ①, we have

$$\sum_{\xi\in\Sigma^p} n_\xi(a_{0:N+1})\, n_\xi(b_{0:j}) = \sum_{\xi\in\Sigma^p} n_\xi(a_{0:N})\, n_\xi(b_{0:j}) = \sum_{\xi\in A} n_\xi(a_{0:N})\, n_\xi(b_{0:j})$$

Here we define A as the set of all possible gapped substring in $a_{0:N}$.

we also define B as the set of all possible gapped substrings in $a_{0:N+1}$ and its last element is a_{N+1}.

Hence, for ②, we have.

$$\sum_{\xi\in\Sigma^p} n_\xi(a_{0:N+1})\, n_\xi(b_{0:j}) = \sum_{\xi\in B} n_\xi(a_{0:N+1})\, n_\xi(b_{0:j})$$

if ~~the~~ last element of strings in a and b are same, then it becomes

$$\sum_{\xi\in B} n_\xi(a_{0:N+1})\, n_\xi(b_{0:j}) = \sum_{q=p}^{j} K^*_{p-1}(a_{0:N}, b_{0:(q-1)})\, [a_{N+1}=b_q]$$

So, we have $K^*_p(a_{0:N+1}, b_{0:j}) = K^*_p(a_{0:N}, b_{0:j}) + \sum_{q=p-1}^{j} K^*_{p-1}(a_{0:N}, b_{0:(q-1)})\, [a_{N+1}=b_q]$.

(B) Let A be the set of all possible gapped substrings in b such that last element is not involved, B be the set of gapped substrings that involves the last element.

$$K_p^*(a_{0:i}, b_{0:N+1}) = \sum_{\xi\in A} n_\xi(a_{0:i})\, n_\xi(b_{0:N}) + \sum_{\xi\in B} n_\xi(a_{0:i})\, n_\xi(b_{0:N+1})$$

$$= K_p^*(a_{0:i}, b_{0:N}) + \sum_{q=p-1}^{i} K^*_{p-1}(a_{0:(q-1)}, b_{0:N})\, [b_{N+1}=a_q]$$

The justifying process is same as in (A).

(C) Let $K(p,i,j) = K_p(\overline{a_{0:i}}, b_{0:j})$

Let $S_1(1,i,j) = \sum_{q=p-1}^{j} K^*_{p-1}(a_{0:i}, b_{0:q-1})\, [a_{i+1}=b_q]$

Let $S_2 = \sum_{q=i:p-1}^{i} k^*_{p-1}(a_{0:q-1}, b_{0:j})[b_{j+1} = a_q]$

If $p=1$, use the algorithm in (i)

if $p>1$.

if $i<p$ or $j<p$ $k(p,i,j)=0$.

if $i=p$ and $j=p$ use the algorithm in (ii) the cost of time is $O(p)$

if $i>p$ or $j>p$, use the formula in (iii) part (α) and (β)

$S_1(p,i,j) = S_1(p,i,j-1) + k^*_{p-1}(a_{0:i}, b_{0:j-1})[a_{i+1} = b_j]$ ①

$S_2(p,i,j) = S_2(p,i-1,j) + k^*_{p-1}(a_{0:i-1}, b_{0:j})[b_{j+1} = a_i]$ ②

$k(p,i,j) = k(p,i,j-1) + S_1(p,i,j-1) + k^*_{p-1}(a_{0:i}, b_{0:j-1})[a_{i+1} = b_j]$ ③

$k(p,i,j) = k(p,i-1,j) + k^*_{p-1}(a_{0:i-1}, b_{0:j})[a_i = b_{j+1}] + S_2(p,i-1,j)$ ④

Pseudo codes:

Initialize 3-dim matrix S_1, S_2 and k, size is $(k+1, l_a, l_b)$

```
for p=1:k
    for i=0:(la-1)
        for j=0:(lb-1)
            if i==0 and j==0
                k(p,i,j)=k(p,0,0)
            else if i==0, j>0.
                compute S2 using ②
                compute k(p,i,j) using ④
            else if i>0 and j==0
                compute S1 using ①
                compute k(p,i,j) using ③
            else
```

compute S_1 using ①

compute S_2 using ②

compute $K(p, i, j)$ using ③ or ④

(c) (i) since there is only one element in $\bar{s}$, $i_{|s|} - i_1 - (|s|-1) = 0$.

$$W_{\bar{s}}(\bar{x}) = \sum_{0 \le i_1 \le |x|_1} [\bar{x}_{i_1} = \bar{s}_1]$$

Hence, it's the same as in (b) (i).

The algorithm is also the same.

(ii) the length of $\bar{a}_{0:p-1}$ and $\bar{b}_{0:p-1}$ are both p,

So $K_p^{n}(\bar{a}_{0:p-1}; \bar{b}_{0:p-1}) = [\bar{a}_{0:p-1} = \bar{b}_{0:p-1}]$

which is also same as in (b) (ii).

The algorithm is also same.

(iii)

$$K_p^{n}(\bar{a}, \bar{b}) = \sum_{\bar{s} \in \Sigma^p} W_{\bar{s}}(\bar{a}) W_{\bar{s}}(\bar{b})$$

$$= \sum_{(\vec{i}, \vec{j}) \in I_p \times I_p,\ \bar{a}(\vec{i}) = \bar{b}(\vec{j})} \lambda^{l(\vec{i}) + l(\vec{j}) - 2p}$$

Here we define $\vec{i} = (i_1, i_2, \ldots i_p)$ $\vec{j} = (\vec{j}_1, \vec{j}_2, \ldots \vec{j}_p)$ ~~$l(\vec{i}) = i_p - i_1$ $l(\vec{j}) = j_p - j_1$~~

$l(\vec{i}) = i_p - i_1$, $l(\vec{j}) = j_p - j_1$

$$\therefore K_p^{n}(\bar{a}, \bar{b}) = \sum_{(\vec{i}, \vec{j}) \in I_p \times I_p :\ a(\vec{i}) = \bar{b}(\vec{j})} \lambda^{l(\vec{i}) + l(\vec{j}) - 2p}$$

$$= \sum_{i=1}^{|\bar{a}|} \sum_{j=1}^{|\bar{b}|} \sum_{(\vec{i}, \vec{j}) \in I_p^i \times I_p^j :\ \bar{a}(\vec{i}) = \bar{b}(\vec{j})} \lambda^{l(\vec{i}) + l(\vec{j}) - 2p}$$

$$= \sum_{i=1}^{|\bar{a}|} \sum_{j=1}^{|\bar{b}|} K_p^{s}(\bar{a}(0:i-1), \bar{b}(0:j-1)) \cdot / \lambda^{2p}$$

Here we denote $K_p^{s}(\bar{a}(0:i-1), \bar{b}(0:j-1)) = \sum_{(\vec{i}, \vec{j}) \in I_p^i \times I_p^j :\ \bar{a}(\vec{i}) = b(\vec{j})} \lambda^{l(\vec{i}) + l(\vec{j})}$

Scanned by CamScanner

This function is to compute strings ~~ends~~ end in $\bar{a}(i-1)$ and $\vec{b}(j-1)$

i.e. ~~kps~~ $k_p^S(\bar{a}(0:i-1), \bar{b}(0:j-1))$ computes the sum of length-p ~~substring~~ substring co-occurences again ignoring the normalization $1/\lambda^{2p}$, that end ~~at~~ positions $i-1$ and $j-1$, we have the following recurrence,

$$k_p^S(\bar{a}(0:i-1), \bar{b}(0:j-1)) = \begin{cases} \lambda^2 [\bar{a}(i-1) = \bar{b}(j-1)] & \text{if } p=1 \\ \sum\limits_{k<i-1} \sum\limits_{l<j-1} \lambda^{i-1-k+j-1-l} k_{p-1}^S(\bar{a}(0:k), \bar{b}(0:l)) [\bar{a}(i-1) = \bar{b}(j-1)] \end{cases}$$

For ~~simplicity~~ samplicity, we denote $k_p^S(\bar{a}(0:i-1), \bar{b}(0:j-1)) = k_p^S(i,j)$,

Thus, we reform above equations.

$$k_p^S(i,j) = \begin{cases} \lambda^2 [\bar{a}(i-1) = \bar{b}(j-1)] & \text{if } p=1 \\ \sum\limits_{k<i} \sum\limits_{l<j} k_{p-1}^S(k,l) \cdot \lambda^{i-k+j-l} [\bar{a}(i-1) = \bar{b}(j-1)] \end{cases}$$

Again we define $S_p(k,l) = \sum\limits_{i<k} \sum\limits_{j<l} \lambda^{k-i+l-j} k_{p-1}^S(i,j)$

$$k_p^S(k,l) = [\bar{a}(k-1) = \bar{b}(l-1)] \lambda^2 S(k-1, l-1)$$

$S_p(k,l)$ can be written as

$$S_p(k,l) = k_{p-1}^S(k,l) + \lambda S_p(k-1,l) + \lambda S_p(k,l-1) - \lambda^2 \cdot S_p(k-1,l-1)$$

Hence, as we can see, the complexity of computation required to compute S_p is $O(|\bar{a}||\bar{b}|)$, so the complexity of computing $k_p^n(\bar{a},\bar{b})$ is equal to $O(p|\bar{a}||\bar{b}|)$

Pseudocode is as follows

~~kps(0:n-1, 0:m)~~

```
kps(1:n, 1:m)=0
for i= 1:n
  for j=1:m
    if ā(i-1)= b̄(j-1)
      kps(i,j)= λ².
```

$sp(0, 0:m) = 0$

$sp(1:n, 0) = 0$

for $l = 2:p$

$k(l) = 0$

for $i = 1:n-1$

for $j = 1:m-1$

$sp(i,j) = kps(i,j) + \lambda sp(i-1, j) + \lambda sp(i, j-1) - \lambda^2 sp(i-1, j-1)$

if $\bar{a}(i-1) = \bar{b}(j-1)$

$kps(i,j) = \lambda^2 sp(i-1, j-1)$

$k(l) = k(l) + kps(i,j)$,

~~Clearly the com~~

$k(p) = k_p(\bar{a}, \bar{b})$, clearly, the complexity of this pseudocode is $O(p|\bar{a}||\bar{b}|)$

(d) The intuitive meaning of this kernel is to compute all possible matches with substrings of length k in $\bar{a}$ and $\bar{b}$, Here, matches are defined as

$$\prod_{t=1}^{k} k_\Sigma(a_{i_t}, b_{j_t})$$

$$\sum_i \sum_j c_i c_j k_k^\Delta(\bar{a}_i, \bar{b}_j) = \sum_i \sum_j c_i c_j \sum_{0 \le i_1 < i_2 < \cdots \le |\bar{a}_i|} \sum_{0 \le j_1 < j_2 < \cdots \le |\bar{b}_j|} \prod_{t=1}^{k} k_\Sigma(a_{i_t}, b_{j_t})$$

$$= \sum_{0 \le i_1 < \cdots < |\bar{a}_i|} \sum_{0 \le j_1 < \cdots < |\bar{b}_j|} \underbrace{\sum_i \sum_j \prod_{t=1}^{k} k_\Sigma(a_{i_t}, b_{j_t})}_{\ge 0}$$

$$\ge 0$$

$\therefore k_k^\Delta(\bar{a}, \bar{b})$ is p.s.d.