CS 480 Assignment 2
Nikhil Nanda
20727218

Exercise 1:

1. $$\left.\begin{aligned} &\min_{w,z} \|z\|_2^2 + \lambda \|w\|_2^2 \\ &\text{s.t.} \quad z = X^T w - y \end{aligned}\right\} \text{Primal}$$

$$\Rightarrow z - X^T w + y = 0$$

Lagrangian $\left\{ \min_{w,z} \max_{\alpha} \|z\|_2^2 + \lambda \|w\|_2^2 + \alpha(z - X^T w + y) \right.$

$\Updownarrow$ swap

$$\max_{\alpha} \min_{w,z} \|z\|_2^2 + \lambda\|w\|_2^2 + \alpha(z - X^T w + y)$$

$$\frac{\partial}{\partial w} = 2\lambda w - \alpha x = 0$$

$$\frac{\partial}{\partial z} = 2z + \alpha = 0$$

Thus, Substituting back

$$\max_{\alpha} \frac{\|\alpha\|^2}{4} + \frac{\lambda}{4\lambda^2}\|X\alpha\|^2 + \alpha\left[\frac{-\alpha}{2} - X^T \cdot \frac{1}{2\lambda} X\alpha + y\right]$$

$$\Rightarrow \max_{\alpha} -\frac{\|\alpha\|^2}{4} - \frac{1}{4\lambda}\|X\alpha\|^2 + \alpha^T y$$

$$\Rightarrow \max_{\alpha} -\frac{\lambda}{4}\|\alpha\|^2 - \frac{1}{4}\|X\alpha\|^2 + \lambda\alpha^T y$$

$$\Rightarrow -\min_{\alpha} -\frac{\lambda}{4}\|\alpha\|^2 - \frac{1}{4}\|X\alpha\|^2 + \lambda\alpha^T y$$

$$\Rightarrow \min_{\alpha} \frac{\lambda}{4}\|\alpha\|^2 + \frac{1}{4}\|X\alpha\|^2 - \lambda\alpha^T y$$

$$\Rightarrow \min_{\alpha} \frac{1}{4}\alpha^T(X^TX + \lambda I_n)\alpha - \lambda\alpha^T y$$

Let $X^TX = K$,

$$\frac{\partial}{\partial\alpha} = \frac{1}{4}\left((K + \lambda I_n) + (K + \lambda I_n)^T\right)\alpha - \lambda y = 0$$

$$\Rightarrow \lambda y = \frac{2}{4}(K + \lambda I_n)\alpha$$

$$\alpha = 2\lambda (K + \lambda I_n)^{-1} y$$

Since $w = \frac{1}{2\lambda} X\alpha$

$$\Rightarrow w = \frac{1}{2\lambda} \cdot X \cdot 2\lambda \cdot (K + \lambda I_n)^{-1} y$$

$$\Rightarrow \boxed{w^* = X(X^T X + \lambda I_n)^{-1} y}$$

2. $$w^* = (XX^T + \lambda I_d)^{-1} X y \quad — ①$$

and $$w^* = X(X^T X + \lambda I_n)^{-1} y \quad — ②$$

$\therefore$ Can equate ① and ②

$$(XX^T + \lambda I_d)^{-1} X y = X(X^T X + \lambda I_n)^{-1} y$$

$$\Rightarrow (XX^T + \lambda I_d)^{-1} X = X(X^T X + \lambda I_n)^{-1}$$

$$\Rightarrow (XX^T + \lambda I_d)^{-1} XX^T = X(X^T X + \lambda I_n)^{-1} X^T$$

$$\Rightarrow -\frac{1}{\lambda}(XX^T + \lambda I_d)^{-1} XX^T = -\frac{1}{\lambda} X(X^T X + \lambda I_n)^{-1} X^T$$

$$\Rightarrow \frac{1}{\lambda} I_d - \frac{1}{\lambda}(XX^T + \lambda I_d)^{-1} XX^T$$

$$= \frac{1}{\lambda} I_d - \frac{1}{\lambda} X(X^T X + \lambda I_n)^{-1} X^T$$

$$\Rightarrow \frac{1}{\lambda}(XX^T + \lambda I_d)^{-1}(XX^T + \lambda I_d) - \frac{1}{\lambda}(XX^T + \lambda I_d)^{-1} XX^T$$

$$= \frac{1}{\lambda} I_d - \frac{1}{\lambda} X(X^T X + \lambda I_n)^{-1} X^T$$

$$\Rightarrow (XX^T+\lambda I_d)^{-1}\left[\frac{1}{\lambda}(XX^T+\lambda I_d)-\frac{1}{\lambda}XX^T\right]$$

$$= \frac{1}{\lambda}I_d - \frac{1}{\lambda}X(X^TX+\lambda I_n)^{-1}X^T$$

$$\Rightarrow (XX^T+\lambda I_d)^{-1} = \frac{1}{\lambda}I_d - \frac{1}{\lambda}X(X^TX+\lambda I_n)^{-1}X^T$$

Exercise 2:

1. $$\min_{w,b} \frac{c}{2}\|w\|_2^2 + \sum_{i=1}^{n} (\rho - y_i\hat{y}_i)_+ ,$$

$$\hat{y}_i = w^T x_i + b$$

→ Given that $\rho > 0$

Thus, on dividing by $\rho\ (>0)$

$$\min_{w,b} \frac{c}{2\rho}\|w\|_2^2 + \sum_{i=1}^{n} \left(1 - \frac{y_i\hat{y}_i}{\rho}\right)_+$$

Let $w \leftarrow w/\rho$ and $c \leftarrow c\rho$

Thus, we reduce the optimization problem to a simpler form with $\rho = 1$

$$\min_{w,b} \frac{c}{2}\|w\|_2^2 + \sum_{i=1}^{n} (1 - y_i\hat{y}_i)_+$$

This just means that the error/loss being calculated by the above objective function is just translated by the change in ρ.

Thus, it doesn't affect the training of the classifier. Only the value of error / loss has been translated by change in ρ.

2.

$$\min_{w,b,\rho} \frac{c}{2}\|W\|_2^2 - \nu\rho + \sum_{i=1}^{n}(\rho - y_i\hat{y}_i)_+ ,$$

$$\hat{y}_i = w^T x_i + b$$

On dividing by $c\ (>0)$,

$$\min_{w,b,\rho} \frac{1}{2}\|W\|_2^2 - \frac{\nu}{c}\rho + \sum_{i=1}^{n}\left(\frac{\rho - y_i\hat{y}_i}{c}\right)_+$$

The above scaling, only changes the minimum value that the function return and not the minimizers of the function. The original value returned by the objective function before scaling can be retrieved by just a simple translation of the value returned by the objective function after scaling.

Thus, we can reduce our original objective function to a simpler case with $c = 1$:

$$\min_{w,b,\rho} \frac{1}{2}\|w\|_2^2 - \nu\rho + \sum_{i=1}^{n} (\rho - y_i \hat{y}_i)_+ ,$$

where $\hat{y}_i = w^T x_i + b$

3. $C = 1, \quad \nu \geq 0$

$$\min_{w, b, \rho} \frac{1}{2} \|w\|_2^2 - \nu \rho + \sum_{i=1}^{n} (\rho - y_i \hat{y}_i)_+ ,$$

$$\hat{y}_i = w^T x_i + b$$

if $\nu > 0$, Let's look at the optimization varible ρ in the objective function:

For the term $-\nu \rho$,

it will be negative if $\rho > 0$, since $\nu > 0$

Thus for any minizer ρ^*, we necessarily have $\rho^* \geq 0$

For the other term $(\rho - y_i \hat{y}_i)_+$,

$$(\rho - y_i \hat{y}_i)_+ = \max\{0, \rho - y_i \hat{y}_i\}$$

In order to minimize $(\rho - y_i \hat{y}_i)_+$, it would be desirable to have $\rho - y_i \hat{y}_i \leq 0$

Thus,

$\rho \geq 0$ to minimize $-\nu\rho$ (where $\nu > 0$)

and $\rho \leq y_i \hat{y}_i$ to minimize $(\rho - y_i \hat{y}_i)_+$

But $\rho \leq y_i \hat{y}_i$ depends on the training data which we have no control on

Therefore, if $\nu > 0$, at any minimizer ρ^*, we necessarily have $\rho^* \geq 0$

→ For the Lagrangian Dual,

$$\min_{w, b, \rho, \xi} \frac{1}{2}\|w\|_2^2 - \nu\rho + \sum_{i=1}^{n} \xi_i$$

$$\text{s.t. } (\rho_i - y_i(w^T x_i + b))_+ \leq \xi_i \rightarrow \begin{cases} \rho - y_i \hat{y}_i \leq \xi_i \\ 0 \leq \xi_i \end{cases}$$

$$\min_{w, b, \rho, \xi} \max_{\substack{\alpha \geq 0 \\ \beta \leq 0}} \frac{1}{2}\|w\|_2^2 - \nu\rho + \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i (\rho - y_i(w^T x_i + b) - \xi_i) + \sum_{i=1}^{n} \beta_i \xi_i$$

$$\max_{\substack{\alpha \geq 0 \\ \beta \leq 0}} \min_{w, b, \rho, \xi} \frac{1}{2} \|w\|_2^2 - \nu \rho + \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i \left(\rho - y_i (w^T x_i + b) - \xi_i\right) + \sum_{i=1}^{n} \beta_i \xi_i$$

$$\frac{\partial}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\Rightarrow \boxed{w = \sum_{i=1}^{n} \alpha_i y_i x_i}$$

$$\frac{\partial}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0$$

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

$$\frac{\partial}{\partial \rho} = -\nu + \sum_{i=1}^{n} \alpha_i = 0$$

$$\Rightarrow \boxed{\nu = \sum_{i=1}^{n} \alpha_i}$$

$$\frac{\partial}{\partial \xi_i} = 1 - \alpha_i + \beta_i = 0$$

$$\Rightarrow \boxed{\alpha_i - \beta_i = 1}$$

Substituting it back,

$$\max_{\substack{\alpha \geq 0 \\ \beta \leq 0}} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \rho \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} \xi_i$$

$$+ (\beta_i - \alpha_i) \sum_{i=1}^{n} \xi_i + \rho \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\Rightarrow \max_{\substack{\alpha \geq 0 \\ \beta \leq 0}} -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Since $\alpha_i - \beta_i = 1 \Rightarrow \beta_i = \alpha_i - 1 \leq 0$

$$\Rightarrow \alpha_i \leq 1$$

$$\Rightarrow \boxed{\begin{aligned} &\min_{0 \leq \alpha \leq 1} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ &\text{s.t.} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ &\quad\ \sum_{i=1}^{n} \alpha_i = \nu \end{aligned}}$$

OR

$$\min_{0 \leq \alpha \leq 1} \frac{1}{2} \left\| \sum_{i=1}^{n} \alpha_i y_i x_i \right\|_2^2$$

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

$$\sum_{i=1}^{n} \alpha_i = \nu$$

4.

$$\min_{p} \; -kp + \sum_{i=1}^{n} (p - a_i)_+ \, ,$$

$a \in \mathbb{R}^n$

On sorting $a \Rightarrow \; a_j \leq a_{j+1} \; \forall j = 1, \ldots, n$

Let $x \in \mathbb{N}$ such that:

$\forall i > x \Rightarrow (p - a_i)_+ = 0$

and $\forall \; i \leq x \Rightarrow (p - a_i)_+ = p - a_i$

Thus, now the objective function is differentiable

$$\min_{p} \; -kp + \sum_{i=1}^{x} (p - a_i)$$

$$\Rightarrow \frac{\partial}{\partial p} = -k + \sum_{i=1}^{x} 1 = 0$$

$$\Rightarrow k = \sum_{i=1}^{x} 1 = x$$

$$\Rightarrow \boxed{k = x}$$

Thus, $\forall\, i > x = k \Rightarrow (p^* - a_i)_+ = 0$

$\Rightarrow p^* \leq a_i$

Since, a is sorted $\Rightarrow p^* \leq a_{x+1} \leq a_{x+2} \cdots$

and $p^* \geq a_x$ (since $(p - a_i)_+ = p - a_i$ for $i \leq x = k$)

$$\boxed{\therefore \quad a_x \leq p^* \leq a_{x+1}},$$

where $k = x$

5.

$$\min_{\substack{\alpha \geq 0,\ 1^T\alpha = 1 \\ \beta \geq 0,\ 1^T\beta = 1}} \frac{1}{2} \left\| \sum_{i:y_i=1} \alpha_i x_i - \sum_{i:y_i=-1} \beta_i x_i \right\|_2^2$$

From Part 3 Dual :

$$\min_{0 \leq \alpha \leq 1} \frac{1}{2} \left\| \sum_{i=1}^{n} \alpha_i y_i x_i \right\|_2^2$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i = \nu$$

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

$$\alpha \begin{cases} \to p \Rightarrow y_i = 1 \\ \to n \Rightarrow y_i = -1 \end{cases}$$

On multiplying both sides of constraint by $2/\nu$

$$0 \leq p \leq 2/\nu$$

$$0 \leq n \leq 2/\nu$$

and $\left.\begin{array}{l} 1^T p = 1 \\ 1^T n = 1 \end{array}\right\} \to$ since $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $\sum_{i=1}^{n} \alpha_i = 2$ (by setting $\alpha \leftarrow 2\alpha/\nu$)

$$\therefore \quad \min_{p,n} \frac{1}{2} \left\| \sum_{i:y_i=1} p_i x_i - \sum_{i:y_i=-1} n_i x_i \right\|_2^2$$

$$\text{s.t.} \quad 0 \le p \le 2/\nu \,, \quad 1^T p = 1$$

$$0 \le n \le 2/\nu \,, \quad 1^T n = 1$$

Difference is that p, n are upper bounded by $2/\nu$

If $\nu \in (0, 2] \Rightarrow$ both will be exactly same

Exercise 3:

1.

$$L = \min_{w} \sum_{i=1}^{n} \log\left(1 + e^{-y_i w^T x_i}\right)$$

Since $w = \sum_{i=1}^{n} \alpha_i \phi(x_i)$ from Representer Theorem

$$x_i \mapsto \phi(x_i)$$

$$L = \min_{\alpha} \sum_{i=1}^{n} \log\left(1 + e^{-y_i \sum_j \alpha_j \phi(x_j)\phi(x_i)}\right)$$

$$= \min_{\alpha} \sum_{i=1}^{n} \log\left(1 + e^{-y_i \sum_j \alpha_j k(x_j, x_i)}\right)$$

$$L = \min_{\alpha} \sum_{i=1}^{n} \log\left(1 + e^{-y_i \alpha^T K_{:i}}\right),$$

where $K_{:i}$ is the i-th column of the Kernel Matrix

$$\frac{\partial L}{\partial \alpha} = \sum_i \frac{-y_i K_{:i}}{1 + \exp(-y_i \alpha^T K_{:i})} \cdot \exp(-y_i \alpha^T K_{:i})$$

Thus, we can run Gradient Descent with the above derivate and learn α

To predict results on the Test Set,

$$w^T \phi(x') = \sum_i \alpha_i [\phi(x_i)]^T [\phi(x')]$$

$$= \sum_i \alpha_i k(x_i, x')$$

Thus, we predict positive result when $w^T \phi(x') > 0.5$

where x' ← Test Dataset instance

x_i ← Training Dataset instance

α ← Learnt from Gradient Descent

2. python3 klev.py

Exercise 4 :

1.

$k_n(x, z)$ is a kernel with kernel matrix K_n

Thus, K_n is Positive semi definite

$$\Rightarrow \forall \alpha \in \mathbb{R}^n$$

$$\alpha^T K_n \alpha \geq 0$$

$$\Rightarrow \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i K_n(x_i, x_j) \alpha_j \geq 0$$

As n approaches infinity, the non-negativity holds for the limit

Hence,

$$\lim_{n \to \infty} K_n = K_n \text{ is also}$$

Positive semi definite

As K_n is also symmetric for all n, the symmetry also holds for the limit as n approaches infinity

Therefore, we can conclude that $\lim_{n \to \infty} K_n(x, z) = K(x, z)$ is a Kernel

2.

From Taylor expansion

$$\exp(k) = \sum_{n=0}^{\infty} \frac{k^n}{n!} = 1 + k + \frac{k^2}{2!} + \frac{k^3}{3!} + \cdots$$

The terms in the expansion are either constant $(=1)$, single kernel $(=k)$ or product of kernels

As k is a kernel,

$$\left.\begin{aligned} k^2 &= k \cdot k \\ k^3 &= k^2 \cdot k \\ &\vdots \\ k^n &= k^{n-1} \cdot k \end{aligned}\right\}$$ also kernel's (since product of kernels is also a kernel)

$\frac{k^n}{n!}$ is also a kernel as $\frac{1}{n!} \geq 0$, for whole numbers,

Since λk is a kernel when
$\lambda \geq 0$ and k is a kernel

1 is also a Kernel as $\lambda . k$ is
a kernel
where $\lambda = \frac{1}{k} \geq 0$

Hence, sum of all Kernels in the expansion is also a Kernel

$\therefore$ $\exp(k)$ is a Kernel,
given k is a Kernel

3.

$k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a kernel

$$k_\varphi(x, z) = \varphi(x)\, k(x, z)\, \varphi(z)$$

Since k is a kernel

$$\Rightarrow k_\varphi(x, z) = \varphi(x)\, \phi^T(x)\, \phi(z)\, \varphi(z)$$

$$= \varphi(x)\, \varphi(z)\, \phi^T(x)\, \phi(z)$$

$$= [\varphi(x)\, \phi(x)]^T [\varphi(z)\, \phi(z)]$$

$$= [\phi'(x)]^T [\phi'(z)]$$

$\therefore k_\varphi(x, z)$ is also a kernel

4.

$$k(x, z) = \exp\left(-\|x - z\|_2^2 / \sigma\right), \quad \sigma > 0$$

$$= \exp\left(\frac{-\|x\|_2^2 - \|z\|_2^2 + 2x^Tz}{\sigma}\right)$$

$$= \exp\left(\frac{-\|x\|_2^2}{\sigma}\right) \exp\left(\frac{-\|z\|_2^2}{\sigma}\right) \cdot \exp\left(\frac{2x^Tz}{\sigma}\right)$$

The term $\exp\left(\frac{2x^Tz}{\sigma}\right)$ can be written as $\exp(k'(x, z))$ where $k'(x, z)$ is a kernel as $\frac{2}{\sigma} \geq 0$ is multiplied by a kernel (x^Tz) which results in a Kernel

From Part 2 $\Rightarrow$ $\exp(k'(x, z))$ is also a kernel

Note: x^Tz is a kernel as $x^Tz = \phi^T(x) \cdot \phi(z)$ where $\phi(x) = x$ and $\phi(z) = z$

7) $k(x, z) = \exp\left(\frac{-\|x\|_2^2}{\sigma}\right) \exp\left(\frac{-\|z\|_2^2}{\sigma}\right).$

$\exp(k'(x,z))$

$= \varphi(x) \exp(k'(x,z)) \varphi(z)$

$= \varphi(x)\ k''(x,z)\ \varphi(z)$

$\therefore$ From Part 3, $k(x,z)$ is a Kernel

5.

$$k(x,z) = \frac{1}{1+\|x-z\|_2^2}$$

For any $t>0$,

$$\frac{1}{1+\|x-z\|_2^2} = \int_0^\infty \exp(-t(1+\|x-z\|_2^2))\,dt$$

$$= \lim_{n\to\infty} \int_0^n \exp(-t(1+\|x-z\|_2^2))\,dt$$

$$= \lim_{n\to\infty} \int_0^n \exp(-t).\exp(-t\|x-z\|_2^2)\,dt$$

From Part 4 we know that

$\exp(-t\|x-z\|_2^2)$ is a kernel

as $1/t > 0$ (since we assume initially $t>0$)

Since $\exp(-t) \in [0,\infty)$

$\Rightarrow \exp(-t) \geq 0$

Thus $\exp(-t)\exp(-t\|x-z\|_2^2)$ is a Kernel

From Part 1 we can conclude that $k(x,z) = \dfrac{1}{1+\|x-z\|_2^2}$ is a Kernel

as the RHS above is a limit of finite sums,

Which in turn is a sum of Kernels (From Part 1).

Thus we know that sum of Kernels is also a Kernel.

Exercise 5:

1.

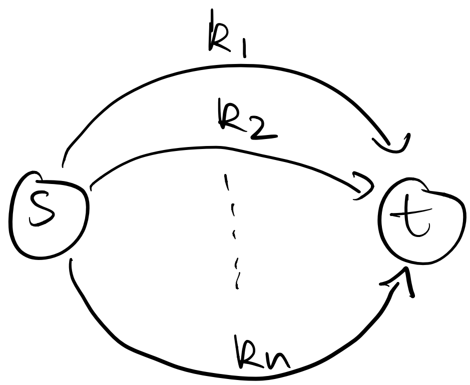
$$k_G(1,-1) = k_{P=\{s,a,c,d,t\}}(1,-1) + k_{P=\{s,a,b,d,t\}}(1,-1)$$

$$= 1.(1+xz)^2 . e^{-|x-z|} . 1 + 1.(xz). e^{-(x-z)^2} . 1$$

$$= 1.(1-1)^2 . e^{-|1-(-1)|} . 1 + 1.(-1) . e^{-(1-(-1))^2} . 1$$

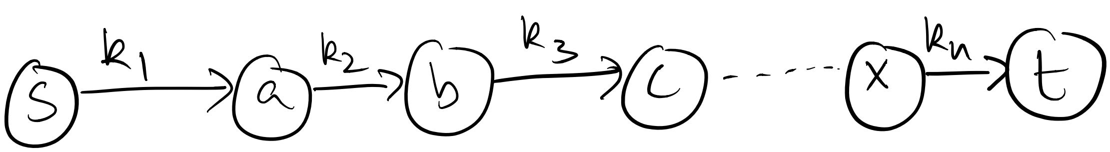
$$\boxed{k_G(1,-1) = -e^{-4}}$$

2.



$$\boxed{k_G = \sum_{i=1}^{n} k_i}$$

3.



$$\boxed{k_G = \bigwedge_{i=1}^{n} k_i}$$

4.

→ Algorithm:

1. Do Topological Sort on DAG (Directed Acyclic Graph) with $|V|$ vertices
2. Initialize array $k[0 \ldots V-1]$ to zero
3. Set $k[V-1]$ to 1
4. Start from end of Topological Sort (vertex t) as well as end of array (index $i = V-2$)
5. $k[i] \leftarrow \sum_{(i,j) \in E} k[j] * \text{edge}(i,j)$
6. $i \leftarrow i-1$
7. Repeat step 5 and 6 until $i = -1$
8. Return $k[0]$

The above algorithm runs in $O(2(|V| + |E|)) = O(|V|+|E|)$ time

→ Doing Topological sort takes $O(|V|+|E|)$ time

→ Then, we go through every vertex and every edge of the DAG exactly once, which takes $O(|V|+|E|)$ time

Thus, algorithm takes $O(|V|+|E|)$ time

→ Topological sort rearranges the node in a DAG such that all arcs go from "smaller" node (s is the smallest node) to a "bigger" node (t is the biggest node)

→ We set $R[V-1]$ to 1 because it the value of the biggest node (t) and this node has no outgoing edges

- The value stored in the array at index i is the k_G value, where G consists of the graph with the last i nodes after Topological sorting the Nodes

- At every index i, if node i has multiple outgoing edges, This means there are multiple paths from node s to node t and the product of kernels of each path need to be added together to get the kernel of the entire graph

Thus, the algorithm is able to calculate k_G in $O(|V| + |E|)$ time.