Deepak Kumar

HW3

1. (a) $K(\vec{x},\vec{z}) = k_1(\vec{x},\vec{z}) + k_2(\vec{x},\vec{z})$

Must be a kernel as sum of two positive semidefinite values is positive semidefinite; thus

$k(\vec{x},\vec{z})$ is a Mercer kernel.

(b) $k(\vec{x},\vec{z}) = k_1(\vec{x},\vec{z}) - k_2(\vec{x},\vec{z})$

Not a kernel. Consider

$$k_2(\vec{x},\vec{z}) = 2k_1(\vec{x},\vec{z}),$$ thus

$k(\vec{x},\vec{z}) = k_1(\vec{x},\vec{z}) - 2k_1(\vec{x},\vec{z}) = -k_1(\vec{x},\vec{z})$, which is <u>not</u> positive semidefinite, thus

$k(\vec{x},\vec{z})$ is not a kernel.

(c) We know $a>0$, $a\in\mathbb{R}$

This is a kernel, as

$k(\vec{x},\vec{z}) = a k_1(\vec{x},\vec{z}) > 0$, thus positive semidefinite.

$\Rightarrow$ Mercer kernel.

~~(d) $k(\vec{x},\vec{z}) = -a k_1(\vec{x},\vec{z})$,~~

~~Because $a>0$, $a\in\mathbb{R}$, we know $-ak_1(\vec{x},\vec{z})<0$, thus $k(\vec{x},\vec{z})$ is negative semidefinite and thus not a mercer kernel.~~

(d) $k(\vec{x},\vec{z}) = -a k_1(\vec{x},\vec{z})$

By defn, we know

$z^T k_1(\vec{x},\vec{z}) z \geq 0$, since $a>0$, $a\in\mathbb{R}$,

$\Rightarrow -a z^T k_1(\vec{x},\vec{z}) z < 0$, so this is not positive semidefinite

$\Longrightarrow$ Not a Mercer kernel.

Scanned by CamScanner

(e) $k_1(\vec{x},\vec{z}) = \sum_i \phi_i^{(1)}(x)\, \phi_i^{(1)}(z)$

$k_2(\vec{x},\vec{z}) = \sum_j \phi_i^{(2)}(x)\, \phi_i^{(2)}(z)$

$k(\vec{x},\vec{z}) = k_1(\vec{x},\vec{z})\, k_2(\vec{x},\vec{z}) = \sum_i \phi_i^{(1)}(x)\, \phi_i^{(1)}(z) \sum_j \phi_j^{(2)}(x)\, \phi_j^{(2)}(z)$

$\Rightarrow \sum_i \sum_j \left(\phi_i^{(1)}(x)\, \phi_j^{(2)}(x)\right)\left(\phi_i^{(1)}(z)\, \phi_j^{(2)}(z)\right)$

$\Rightarrow$ Sub in for kernels, thus $k(\vec{x},\vec{z})$ are kernels $\boxtimes$

(f) $k(\vec{x},\vec{z}) = f(\vec{x})\, f(\vec{z})$

Kernel.

We want to write $k(\vec{x},\vec{z})$ as $\phi(x)^T \phi(x) \Rightarrow$ since $f(\vec{x})$ and $f(\vec{z})$ are scalars, we fit this property if we set $\phi = f$.

(g) $k(\vec{x},\vec{z}) = k_3(\phi(\vec{x}), \phi(\vec{z}))$

Because k_3 is a kernel over $\mathbb{R}^M \times \mathbb{R}^M$, and $\phi(\vec{x})$ + $\phi(\vec{z})$ map $\Rightarrow \mathbb{R}^M$,

$k(\vec{x},\vec{z})$ must be a kernel.

(h) Since $p: \mathbb{R} \Rightarrow \mathbb{R}$ is a polynomial with positive coeffecients, We know the sum returns a kerrel (a), scalar multiplication by positive number returns a kernel (c), product returns a kernel (e), and real valued number returns a kernel (f), which the superposition of which returns a polynomial. Thus, $k(\vec{x},\vec{z}) = p\, k_1(\vec{x},\vec{z})$ is a kernel.

Scanned by CamScanner

2. (a) To solve this, write $w^{(n+1)}$ [final iteration of w] to observe pattern

$$w^{n+1} = w^n + \alpha[t^{n+1} - y(x^{n+1}; w^n)] \cancel{x^{n+1}} \phi(x^{n+1})$$

expand w^n

$$\Rightarrow w^{n-1} + \alpha[t^n - y(x^n; w^{n-1})] \cancel{x^n}\,\phi(x^{(n)}) + \cdots\cdots$$

Continue expanding until you hit w_0, thus, we see

$$w^{n+1} = \sum_{i=1}^{N} a_i \phi(x^{(i)}), \quad a_i = \alpha[t^n - y(x^i; w^{i-1})]$$

Thus, we write the $w^{(i)}$ as a linear combinations of $\phi(x^{(i)})$s.

(b) $f(w^{(i)T} \phi(x^{(i+1)}))$

$$= f\left(\sum_{j=1}^{N} a_j^T \phi(x^{(j)})^T \phi(x^{(i+1)})\right)$$

Use the kernel trick

$$\Rightarrow f\left(\sum_{j=1}^{N} a_j^T k(x^{(j)}, x^{(i+1)})\right),$$ we can use this kernel trick to effeciently compute the prediction on input $x^{(i+1)}$.

(c) $w^{(i+1)} = w^{(i)} + \alpha[t^{(i+1)} - y(\phi(x^{(i+1)}); w^{(i)})]\phi(x^{(i+1)})]$

$$= w^{(i)} + \alpha[t^{(i+1)} - f\left(\sum_{j=1}^{N} a_j^T k(x^{(j)}; x^{(i+1)})\right)]\phi(x^{(i+1)})$$ $\boxtimes$

Scanned by CamScanner

3. (a) $E(\vec{w}, b) = \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \max(0, 1 - t^{(i)}(w^T x^{(i)} + b))$

$= \frac{1}{2}||\vec{w}||^2 + C \sum_{i=1}^{N} \cancel{m}\, I[t^{(i)}(w^T x^{(i)} + b) < 1](1 - t^{(i)}(w^T x^{(i)} + b))$

Solve for two derivatives

$\nabla_w E(\vec{w}, b) = \frac{2}{2}\vec{w} + C \sum_{i=1}^{N} I[t^{(i)}(w^T x^{(i)} + b) < 1]$
$(\cancel{+}\, \nabla_w (1 - [t^{(i)} w^T x^{(i)} + t^{(i)} b)]$

$= \vec{w} + C \sum_{i=1}^{N} I[t^{(i)}(w^T x^{(i)} + b) < 1](-t^{(i)} x^{(i)})$

$= \cancel{w}\ w - C \sum_{i=1}^{N} I[t^{(i)}(w^T x^{(i)} + b) < 1] t^{(i)} x^{(i)}$ ✓

$\frac{\partial}{\partial b} E(\vec{w}, b) = C \sum_{i=1}^{N} I(t^{(i)}(w^T x^{(i)} + b) < 1] [\cancel{0} - t^{(i)} b]$

$= - C \sum_{i=1}^{N} I[t^{(i)}(w^T x^{(i)} + b) < 1](-t^{(i)})$ ✓

Thus, proven. ☒

(b). Given in code

(c) $E^{(i)}(\vec{w},b) = \frac{1}{2N}\|w\|^2 + C\max(0, 1-t^{(i)}(w^T x^{(i)}+b))$

$= \frac{1}{2N}\|w\|^2 + C\,\mathbb{I}[\cancel{1-}\,t^{(i)}(w^T x^{(i)}+b)<1](1-t^{(i)}(w^T x^{(i)}+b))$

$\nabla_w E^{(i)}(\vec{w},b) = \frac{\|w\|}{N} + \nabla_w C\,\mathbb{I}[t^{(i)}(w^T x^{(i)}+b)<1)]\,(1-t^{(i)}w^T x^{(i)}+t^{(i)}b))$

$= \frac{\|w\|}{N} - C\,\mathbb{I}[t^{(i)}(w^T x^{(i)}+b)<1](t^{(i)}x^{(i)})$ ✓

$\frac{\partial}{\partial b}E^{(i)}(\vec{w},b)$

$= -C\,\mathbb{I}[t^{(i)}(w^T x^{(i)}+b)<1]\,t^{(i)}$ ✓

Thus, proven. ☒

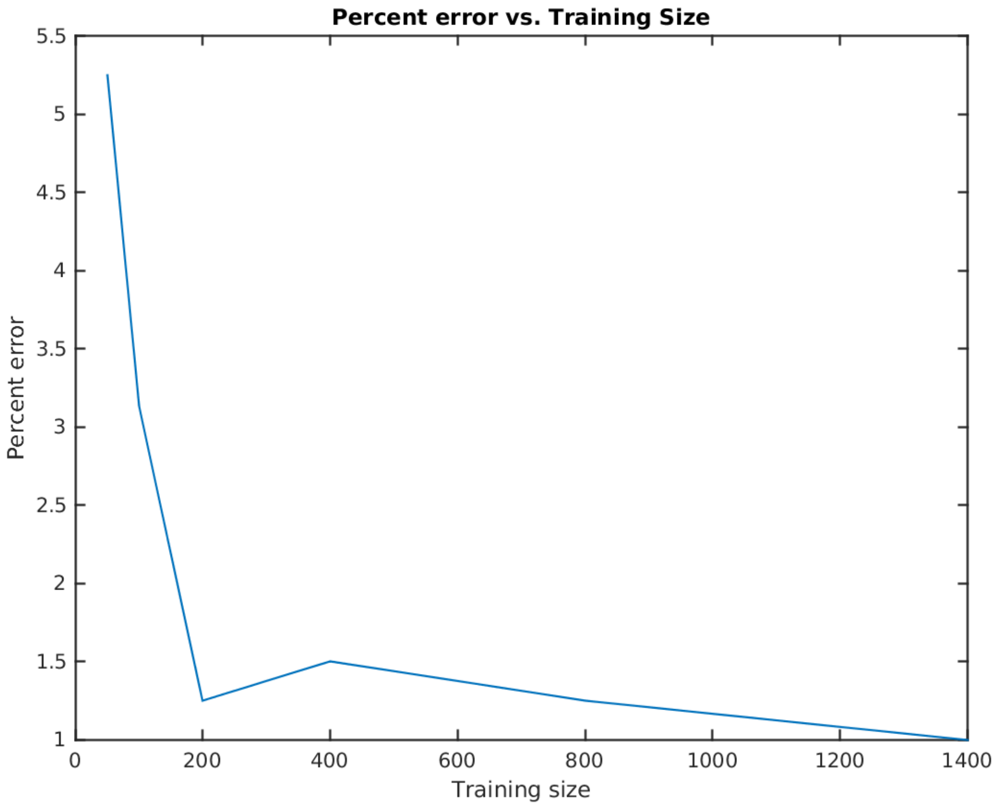
(d) Given in code

(e) We conclude SGD converges much faster than BGD. Observe that at 5 iterations, BGD was at ~55% accuracy, but at the same number, SGD was at 96% accuracy.

4 (a) Given in code.

4 (b) Analysis of the two models yields SVM to be greater than NB in terms of accuracy → though the trends of the two do follow a similar path. With more training examples, error decreases.

Scanned by CamScanner


Percent error vs. Training Size
Percent error
Training size
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
200
400
600
800
1000
1200
1400