EECS 445 F14 Homework 2

WU Tongshuang
40782356

Group mate:
Eric Shin
Steven Mikes
Kevin Wegienka

1.

a) Kernel (6.17 in lecture notes).
∵ K_1 and K_2 are both symmetric
∴ K is symmetric
∵ K_1 and K_2 are both positive semidefinite: $\forall z,\ z^T K_1 z \geq 0\ and\ z^T K_2 z \geq 0$
∴ $K_1 + K_2$ is positive semidefinite: $\forall z,\ z^T K z = z^T K_1 z + z^T K_2 z \geq 0$

b) Not kernel.
Counterexample:
If $K_1 = 3K_2$: $\forall z,\ z^T K z = z^T K_1 z - z^T K_2 z = -2z^T K_1 z \leq 0$

c) Kernel (6.13 in lecture notes)
∵ $a \in \mathbb{R}^+$,
∴ $\forall z, az^T K_1 z \geq 0$ and, where aK_1 is symmetric.

d) Not kernel.

∵ $a \in \mathbb{R}^+$,

∴ $-a \leq 0$. $If\ a = 1$:

∴ $\forall z, -z^T G_1 z \leq 0$

e) Kernel (6.18 in lecture notes)
∵ K_1 and K_2 are both kernel
∴ $\exists \emptyset$ and $\emptyset'$such that:

$$K_1(x,z) = \emptyset(x)^T \emptyset(z) = \sum_i \emptyset_i(x)\emptyset_i(z)$$

$$K_2(x,z) = \emptyset'(x)^T \emptyset'(z) = \sum_i \emptyset'_i(x)\emptyset'_i(z)$$

∴ $K(x,z) = K_1(x,z)K_2(x,z)$

$$= \sum_i(\emptyset_i(x)\emptyset_i(z)) \sum_i(\emptyset'_i(x)\emptyset'_i(z))$$
$$= \sum_i \sum_j(\emptyset_i(x)\emptyset_i(z))(\emptyset'_j(x)\emptyset'_j(z))$$

$$= \sum_i \sum_j (\emptyset_i(x)\emptyset'_j(x))(\emptyset_i(z)\emptyset'_j(z))$$

$\therefore$ We can derive a new $\emptyset''$, where $\emptyset''(x) = \emptyset_i(x)\emptyset'_j(x)$

$\therefore$ $K(x,z)$ is in Kernel form:

$$K(x,z) = \sum_{i,j} \Big(\emptyset_i(x)\emptyset'_j(x)\Big)\Big(\emptyset_i(z)\emptyset'_j(z)\Big) = \emptyset''(x)^T\emptyset''(z)$$

f) Kernel.

Let $\emptyset(x) = f(x)$.

$\because$ $f(x)$ is a scalar

$\therefore$ $K(x,z) = f(x)f(z) = \emptyset(x)^T\emptyset(z)$.

g) Kernel.

$\because K_3(x,z)$ is a kernel

$\because K_3$ is semi-positive, which will hold true for $K_3\big(\emptyset(x), \emptyset(z)\big)$.

h) Kernel.

$\because$ $K(x,z) = p(K_1(x,z)) = \big(K_1(x,z)\big)^n$ where n is positive

$\therefore$ $K(x,z)$ is the product of many kernels.

$\therefore$ From (e), $K(x,z)$ is a kernel

i)

$$K(x,z) = exp(-\frac{||x-z||^2}{2\sigma^2})$$

$$= exp(-\frac{x^Tx}{2\sigma^2})exp(\frac{x^Tz}{\sigma^2})exp(\frac{z^Tz}{2\sigma^2})$$

$$= exp(-\frac{x^Tx}{2\sigma^2})exp(\frac{x^Tz}{\sigma^2})exp(\frac{z^Tz}{2\sigma^2})$$

$$= exp(-\frac{x^Tx}{2\sigma^2})exp(\frac{z^Tz}{2\sigma^2})\sum_{n=0}^{\infty}\frac{(x^n)^Tz^n}{\sigma^{2|n|}n!}$$

Since $(x^Tx)^T = x^Tx$, $exp(-\frac{x^Tx}{2\sigma^2})exp(\frac{z^Tz}{2\sigma^2})$ is a kernel with

$$\emptyset_1(x) = exp(-\frac{x^Tx}{2\sigma^2})$$

For form $\frac{(x^n)^Tz^n}{\sigma^{2|n|}n!}$, it's also a kernel with$\emptyset_2(x) = \frac{x^n}{\sigma^{|n|}\sqrt{n!}}$

From (a), $\sum_{n=0}^{\infty}\frac{(x^n)^Tz^n}{\sigma^{2|n|}n!}$ is a kernel, with $\emptyset_3(x)$ being an infinite dimensional vector, with each entry of form

$$\frac{x^n}{\sigma^{|n|}\sqrt{n!}}, n \in [0,\infty)$$

From (e), $\emptyset(x)$ for Gaussian Kernel can be computed as:

$$\emptyset(x) = \emptyset_1(x)\emptyset_3(x)$$

Which is also an infinite dimensional vector.

2.

a)

$$w := w + \alpha(t^{(i)} - y(\emptyset(x^{(i)}); w))\emptyset(x^{(i)})$$

w will always be a linear combination of the $\emptyset(x^{(i)})$: after incorporating training points,

$\exists \beta_k$ such that $w^{(i)} = \sum_{k=1}^{i} \beta_k \emptyset(x^{(k)})$

$\therefore$ w can be represented by β_k (coefficient of the linear combination): i real numbers after having incorporated I training points.

The initial value $w^{(0)}$ corresponds to cases when the summation has no terms.

b)

$$y(\emptyset(x^{(i)}); w^{(i)}) = f({w^{(i)}}^T \emptyset(x^{(i+1)}))$$

$$= f(\sum_{k=1}^{i} \beta_k \emptyset(x^{(k)})^T \emptyset(x^{(k+1)}))$$

$$= f(\sum_{k=1}^{i} \beta_k K(x^{(k)}, x^{(k+1)})),$$

which can be computed efficiently.

c)

Compute $\beta_i = \alpha(t^{(i)} - f({w^{(i-1)}}^T \emptyset(x^{(i)})))$ with the kernel trick described in (b), then w can be efficiently updated.

3.

a)

When $1 - t^{(i)}(w^T x^{(i)} + b) > 0$:

$I(t^{(i)}(w^T x^{(i)} + b) < 1)$=1.

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

$$= \nabla_w(\frac{1}{2}||\mathrm{w}||^2 + \mathrm{C}\sum_{i=1}^{N}(1 - t^{(i)}(w^T x^{(i)} + b))$$

$$= \nabla_w(\frac{1}{2}w^T\mathrm{w} + \mathrm{C}\sum_{i=1}^{N}(1 - t^{(i)}(w^T x^{(i)} + b))$$

$$= \mathrm{w} - \mathrm{C}\sum_{i=1}^{N} t^{(i)} x^{(i)}$$

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

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

$$= \frac{\partial}{\partial \mathrm{b}}(\frac{1}{2}||\mathrm{w}||^2 + \mathrm{C}\sum_{i=1}^{N}(1 - t^{(i)}(w^T x^{(i)} + b))$$

$$= -\mathrm{C}\sum_{i=1}^{N} t^{(i)}$$

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

When $1 - t^{(i)}(w^T x^{(i)} + b) \leq 0$:

$I(t^{(i)}(w^T x^{(i)} + b) < 1)$=0.

$$\nabla_w E(w,b) = \nabla(\tfrac{1}{2}||\mathrm{w}||^2 + \mathrm{C}\sum_{i=1}^{N} max(0, 1 - t^{(i)}(w^T x^{(i)} + b)))$$

$$= \nabla_w(\tfrac{1}{2}||\mathrm{w}||^2)$$

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

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

$$= \frac{\partial}{\partial \mathrm{b}}(\tfrac{1}{2}||\mathrm{w}||^2)$$

$$= 0$$

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

Therefore,

$$\nabla_w E(w,b) = \mathrm{w} - \mathrm{C}\sum_{i=1}^{N} I(t^{(i)}(w^T x^{(i)} + b) < 0)t^{(i)}x^{(i)}$$

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

(b)

Num. iteration	w	b	accuracy
5	$\begin{bmatrix}123.9595\\-26.1923\\268.4059\\98.9796\end{bmatrix}$	-0.0419	0.50
50	$\begin{bmatrix}10.8493\\-10.1315\\27.5918\\15.5309\end{bmatrix}$	-0.3094	0.50
100	$\begin{bmatrix}-0.5009\\-12.9514\\14.1483\\11.0535\end{bmatrix}$	-0.3479	0.96
1000	$\begin{bmatrix}-0.1655\\-2.6619\\0.1146\\3.8643\end{bmatrix}$	-0.3753	0.50
5000	$\begin{bmatrix}0.0367\\-0.5321\\0.0488\\2.0702\end{bmatrix}$	-0.3901	0.96

6000	$\begin{bmatrix} 0.0638 \\ -0.4931 \\ 0.0541 \\ 1.9503 \end{bmatrix}$	-0.3919	0.96

(c)

When $1 - t^{(i)}(w^T x^{(i)} + b) > 0$:

$I(t^{(i)}(w^T x^{(i)} + b) < 1)$=1.

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

$$= \nabla_w(\frac{1}{2N}||\mathrm{w}||^2 + \mathrm{C}(1 - t^{(i)}(w^T x^{(i)} + b)))$$

$$= \nabla_w(\frac{1}{2N} w^T \mathrm{w} + \mathrm{C}\sum_{i=1}^{N}(1 - t^{(i)}(w^T x^{(i)} + b))$$

$$= \frac{1}{N}\mathrm{w} - \mathrm{C}t^{(i)}x^{(i)}$$

$$= \frac{1}{N}\mathrm{w} - \mathrm{C} \cdot I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}x^{(i)}$$

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

$$= \frac{\partial}{\partial \mathrm{b}}(\frac{1}{2N}||\mathrm{w}||^2 + \mathrm{C}(1 - t^{(i)}(w^T x^{(i)} + b)))$$

$$= -\mathrm{C}t^{(i)}$$

$$= -\mathrm{C} \cdot I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}$$

When $1 - t^{(i)}(w^T x^{(i)} + b) \leq 0$:

$I(t^{(i)}(w^T x^{(i)} + b) < 1)$=0.

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

$$= \nabla_w(\frac{1}{2N}||\mathrm{w}||^2)$$

$$= \frac{1}{N}\mathrm{w} - \mathrm{C} \cdot I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}x^{(i)}$$

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

$$= \frac{\partial}{\partial \mathrm{b}}(\frac{1}{2N}||\mathrm{w}||^2)$$

$$= 0$$

$$= -\mathrm{C} \cdot I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}$$

Therefore,

$$\nabla_w E^{(i)}(w, b) = \frac{1}{N}\mathrm{w} - \mathrm{C} \cdot I(t^{(i)}(w^T x^{(i)} + b) < 1)t^{(i)}x^{(i)}$$

$$\frac{\partial}{\partial \mathrm{b}} E^{(i)}(w, b) = -\mathrm{C} \cdot I(t^{(i)}(w^T x^{(i)} + b) < 1) t^{(i)}$$

d)

Num. iteration	w	b	accuracy
5	$\begin{bmatrix} 0.9695 \\ -4.7780 \\ 3.2424 \\ 2.7991 \end{bmatrix}$	-0.0167	0.96
50	$\begin{bmatrix} 1.1196 \\ -3.1281 \\ -0.0098 \\ 2.6457 \end{bmatrix}$	-0.0253	0.96
100	$\begin{bmatrix} 1.1134 \\ -2.3517 \\ -0.3725 \\ 2.4465 \end{bmatrix}$	-0.0277	0.96
1000	$\begin{bmatrix} 0.0330 \\ -0.5242 \\ -0.0642 \\ 2.1668 \end{bmatrix}$	-0.0485	0.96
50000	$\begin{bmatrix} -0.0248 \\ -0.4330 \\ 0.0694 \\ 1.7294 \end{bmatrix}$	-0.0769	0.96
6000	$\begin{bmatrix} -0.0247 \\ -0.4346 \\ 0.0682 \\ 1.7197 \end{bmatrix}$	-0.0802	0.96

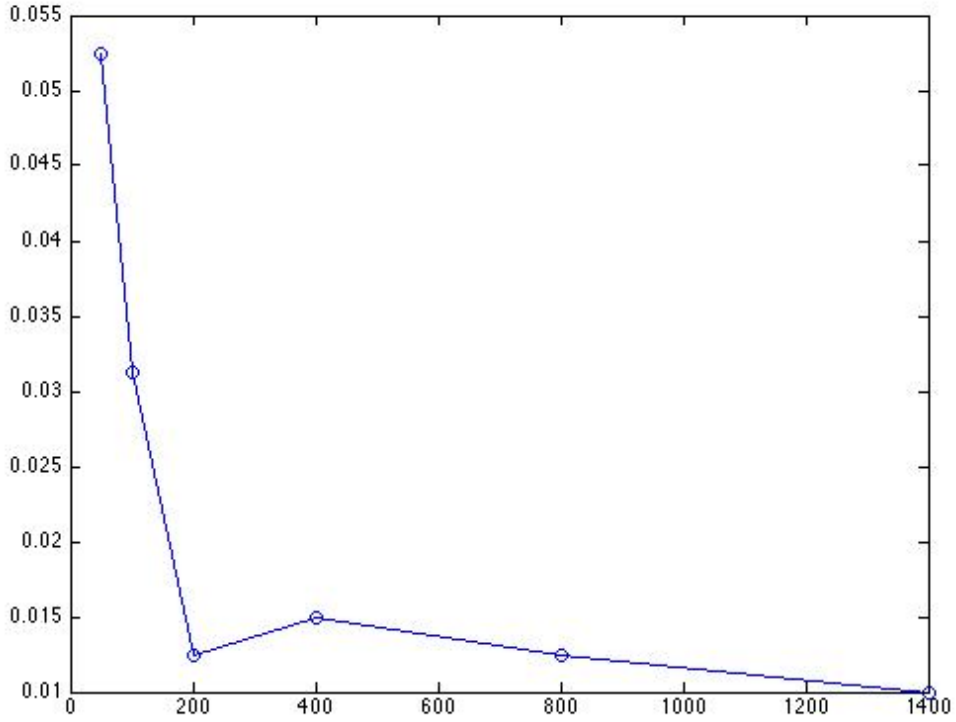
e) SGD converges more quickly. From the result above, it reaches the highest accuracy possible: 96% in 5 iterations, and the value of w and b doesn't vary greatly thereafter. However, when the iteration number is set to be high, it takes much more time for SGD to finish running.

4.

a)

Training set size	Test set error
50	5.25%
100	3.12%
200	1.25%
400	1.50%
800	1.25%

1400	1.00%
Full size	0.63%



b)

Compare with results got in homework 2:

Training set size	**Test set error**
50	3.87%
100	2.62%
200	2.62%
400	1.87%
800	1.75%
1400	1.63%
Full size	1.63%

The conclusion are as following:

- Naive Bayes learns quickly with less data, but has higher asymptotic error.
- SVM classifier has relatively higher error on very small training sets, but performs better than Naive Bayes when the training set is reasonably large.

5.

Three line of code:

```
classifier = NaiveBayesClassifier.train(train_set)
spam_accuracy = nltk.classify.accuracy(classifier, test_spam)
ham_accuracy = nltk.classify.accuracy(classifier, test_ham)
```

Output screenshot:

```
Test Spam accuracy: 100.00%
Test Ham accuracy: 99.00%
Most Informative Features
                   click = True              spam : ham    =    102.1 : 1.0
                     web = True              spam : ham    =     87.1 : 1.0
                newslett = True              spam : ham    =     59.6 : 1.0
               subscript = True              spam : ham    =     56.1 : 1.0
                 mortgag = True              spam : ham    =     49.0 : 1.0
                 easiest = True              spam : ham    =     47.2 : 1.0
                subscrib = True              spam : ham    =     44.2 : 1.0
                    mime = True              spam : ham    =     41.0 : 1.0
                   expir = True              spam : ham    =     36.5 : 1.0
                 enterpr = True              spam : ham    =     35.8 : 1.0
               promotion = True              spam : ham    =     34.7 : 1.0
                    earn = True              spam : ham    =     34.0 : 1.0
                 anytim = True               spam : ham    =     32.9 : 1.0
                    aol = True               spam : ham    =     32.9 : 1.0
                    opt = True               spam : ham    =     30.9 : 1.0
                   loan = True               spam : ham    =     29.4 : 1.0
                    www = True               spam : ham    =     29.4 : 1.0
              confidenti = True              spam : ham    =     29.4 : 1.0
                     fl = True               spam : ham    =     29.4 : 1.0
                   bonu = True               spam : ham    =     29.4 : 1.0
```

Compared with the self-constructed code, Liblinear performs better in terms of the accuracy (i.e. much less error rate). Indicative words are exported with much difference.

The most informative words, based on the output, are the most relevant features. It is computed to be the highest value of probability ratio:

$$max[\frac{P(feature\ = True\ |spam)}{P(feature\ = True\ |ham)}]$$