CS221 Homework 2 - Sentiment

Name: Vinod Kumar Senthil Kumar

SUNET ID: vinodkum

Contributors: Xun Wang, Joe Fan

Problem 1:

1a) $Loss_{hinge}(x, y, w) = \max\{0, 1 - w \cdot \phi(x)\, y\}$

$$\nabla_w Loss(x, y, w) = \begin{cases} 0 & , \; w \cdot \phi(x)\, y \geq 1 \\ -\phi(x)\, y & , \; w \cdot \phi(x)\, y < 1 \end{cases}$$

The stochastic gradient descent is,

$$w \leftarrow w - \eta \nabla_w Loss_{hinge}(x, y, w)$$

The input set of reviews are,

x_1 = pretty good $\quad y_1 = +1$
x_2 = bad plot $\quad y_2 = -1$
x_3 = not bad $\quad y_3 = +1$
x_4 = pretty scenery $\quad y_4 = +1$

Let the initial weights be

$$w = \begin{bmatrix} \text{pretty} : 0 \\ \text{good} : 0 \\ \text{bad} : 0 \\ \text{plot} : 0 \\ \text{not} : 0 \\ \text{scenery} : 0 \end{bmatrix}, \quad \eta = 1$$

For the ease of working, the order of keys is fixed

$$W = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Running SGD of (x_1, y_1),

$$W = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - (1)(-1)\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}(+1)$$

(As $w \cdot \phi(x) y < 1$, $\nabla_w \text{Loss}(x, y, w) = -\phi(x) y$)

$$W = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Running SGD on (x_2, y_2),

$$W = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - (1)(-1)\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}(-1)$$

(As $w \cdot \phi(x) y < 1$, $\nabla_w \text{Loss}(x, y, w) = -\phi(x) y$)

$$W = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$$

Running SGD of (x_3, y_3),

$$w = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 0 \\ 0 \end{bmatrix} - (1)(-1) \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} (+1) \qquad \left(\begin{array}{c} \text{As } w \cdot \phi(x) y < 1 \\ \nabla_w \text{Loss}(x, y, w) = \\ -\phi(x) y \end{array} \right)$$

$$w = \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix}$$

Running SGD of (x_4, y_4),

$$w = \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} - (1)(0) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} \qquad \left(\begin{array}{c} \text{As } w \cdot \phi(x) y \geq 1 \\ \nabla_w \text{Loss}(x, y, w) = 0 \end{array} \right)$$

1b) Let the data set be

x_1 = good $\quad y_1 = +1$

x_2 = bad $\quad y_2 = -1$

x_3 = not good $\quad y_3 = -1$

x_4 = not bad $\quad y_4 = +1$

Let the weights be

$$W = \begin{bmatrix} w_{good} \\ w_{bad} \\ w_{not} \end{bmatrix}$$

To classify x_1 correctly, w_{good} must be +ve

Let $w_{good} = g > 0$

To classify x_2 correctly, w_{bad} must be -ve

Let $w_{bad} = b < 0$

Now, to classify x_3 correctly,

w_{not} should be $< -g$

(i.e w_{not} should be -ve)

But to classify x_4 correctly, w_{not} should be $> -b$

$w_{not} > -b$

(i.e w_{not} should be +ve)

As w_{not} cannot be both +ve & -ve, it is impossible for a linear classifier to classify with zero error.

This limitation can be overcome by introducing an additional feature that shows the interaction between features.

If the new feature is "not good", then an example weight that would classify all the data correctly would be

$$w = \begin{bmatrix} w_{good} = 1 \\ w_{bad} = -1 \\ w_{not} = 0 \\ w_{not\,good} = -2 \end{bmatrix}$$

1c) $\text{Loss}(w, (x_1, \ldots, x_m), I)$

$$= \sum_{i \in I} \max\left(0, 1 - \left(w \cdot \phi(x_i) - \max_{j \in \{1,\ldots,m\} - I} (w \cdot \phi(x_j))\right)\right)$$

$$= \sum_{i \in I} \max_{j \in \{1,\ldots,m\}} \left(w \cdot \phi(x_j) - w \cdot \phi(x_i) + 1[i \neq j]\right)$$

2d) I ran a parameter sweep for the hyper-parameters in the following ranges:

- number of iterations numIter in the range 10 to 20 in increments of 1
- step size eta in the range 0.01 to 0.1 in increments of 0.01

Here are the top 25 combinations of hyper-parameters producing the lowest dev error in ascending order of dev error:

numIters	eta	train error	dev error
19	0.02	0.010410805	0.264772088
11	0.06	0.014068655	0.267023073
16	0.04	0.009566685	0.268429938
12	0.04	0.014068655	0.268711311
12	0.05	0.01266179	0.268711311
17	0.04	0.005346089	0.268992684
20	0.01	0.027011818	0.269274057
20	0.04	0.002250985	0.269555431
11	0.04	0.015475521	0.269836804
13	0.06	0.006752954	0.269836804
14	0.04	0.00815982	0.270118177
20	0.02	0.01181767	0.270118177
11	0.08	0.012943163	0.27039955
11	0.09	0.014068655	0.27039955
10	0.07	0.012099043	0.270680923
15	0.06	0.005064716	0.270680923
13	0.03	0.014068655	0.271243669
10	0.05	0.01997749	0.272369162
15	0.02	0.018289252	0.272369162
14	0.02	0.019133371	0.272650535
16	0.03	0.009566685	0.272931908
13	0.02	0.02363534	0.273213281
10	0.06	0.020258863	0.273494654
17	0.1	0.001969612	0.273494654
17	0.06	0.002813731	0.273494654

The tuned hyper-parameters are as follow:

- number of iterations numIter=19
- step size eta=0.02

2e) **Error Analysis:**

Stop words:
Here are two predictions that would have been classified correctly if the classifier had had knowledge about stop words such as I, am, a, an, etc...

1) === as giddy and whimsical and relevant today as it was 270 years ago .
Truth: 1, Prediction: -1 [WRONG]

giddy	1 * 0.3 = 0.3
it	1 * 0.3 = 0.3
and	2 * 0.14 = 0.28
ago	1 * 0.22 = 0.22
today	1 * 0.14 = 0.14
270	1 * 0 = 0
.	1 * -0.04 = -0.04
years	1 * -0.08 = -0.08
whimsical	1 * -0.16 = -0.16
relevant	1 * -0.38 = -0.38
as	2 * -0.24 = -0.48
was	1 * -0.48 = -0.48

The above example would have been classified correctly if the classifier ignored the words 'as' & 'was'

2) === many of benjamins' elements feel like they've been patched in from an episode of miami vice .
Truth: -1, Prediction: 1 [WRONG]

from	1 * 0.34 = 0.34
of	2 * 0.12 = 0.24
an	1 * 0.22 = 0.22
they've	1 * 0.2 = 0.2
many	1 * 0.06 = 0.06
been	1 * 1.31838984174e-16 = 1.31838984174e-16
vice	1 * 0 = 0
miami	1 * 0 = 0
benjamins'	1 * 0 = 0
.	1 * -0.04 = -0.04
episode	1 * -0.06 = -0.06
in	1 * -0.08 = -0.08
patched	1 * -0.08 = -0.08
elements	1 * -0.1 = -0.1
like	1 * -0.18 = -0.18
feel	1 * -0.2 = -0.2

The above example would have been classified correctly if the classifier ignored the words 'from', 'of' & 'an'

Wrong language model:
Here are two predictions that would have been classified correctly if the classifier had had the correct language model.

3) === o ótimo esforço do diretor acaba sendo frustrado pelo roteiro , que , depois de levar um bom tempo para colocar a trama em andamento , perde-se de vez a partir do instante em que os estranhos acontecimentos são explicados .
Truth: -1, Prediction: 1 [WRONG]

de	2 * 0.2 = 0.4
em	2 * 0.16 = 0.32
para	1 * 0.18 = 0.18
o	1 * 0.12 = 0.12
vez	1 * 0.1 = 0.1
do	2 * 0.04 = 0.08
diretor	1 * 0.04 = 0.04
partir	1 * 0.04 = 0.04
os	1 * 0.04 = 0.04
são	1 * 0 = 0
instante	1 * 0 = 0
estranhos	1 * 0 = 0
sendo	1 * 0 = 0
explicados	1 * 0 = 0
trama	1 * 0 = 0
acaba	1 * 0 = 0
colocar	1 * 0 = 0
pelo	1 * 0 = 0
frustrado	1 * 0 = 0
roteiro	1 * 0 = 0
tempo	1 * 0 = 0
perde-se	1 * 0 = 0
andamento	1 * 0 = 0
depois	1 * 0 = 0
ótimo	1 * 0 = 0
esforço	1 * 0 = 0
levar	1 * 0 = 0
bom	1 * 0 = 0
acontecimentos	1 * 0 = 0
.	1 * -0.04 = -0.04
,	3 * -0.02 = -0.06
a	2 * -0.04 = -0.08
que	2 * -0.06 = -0.12
um	1 * -0.34 = -0.34

4) === graças às interações entre seus personagens , o filme torna-se não apenas uma história divertida sobre uma curiosa perseguição , mas também um belo estudo de personagens .
Truth: 1, Prediction: -1 [WRONG]

de	1 * 0.2 = 0.2
uma	2 * 0.08 = 0.16
o	1 * 0.12 = 0.12
também	1 * 0.08 = 0.08
sobre	1 * 0.04 = 0.04
graças	1 * 0 = 0
às	1 * 0 = 0
história	1 * 0 = 0
interações	1 * 0 = 0
entre	1 * 0 = 0
perseguição	1 * 0 = 0
divertida	1 * 0 = 0
torna-se	1 * 0 = 0
estudo	1 * 0 = 0
seus	1 * 0 = 0
curiosa	1 * 0 = 0
personagens	2 * 0 = 0
.	1 * -0.04 = -0.04
,	2 * -0.02 = -0.04
belo	1 * -0.06 = -0.06
filme	1 * -0.06 = -0.06
apenas	1 * -0.08 = -0.08
não	1 * -0.16 = -0.16
mas	1 * -0.2 = -0.2
um	1 * -0.34 = -0.34

Semantics of English:
Here are three predictions that would have been classified correctly if the classifier had had the knowledge about English semantics and sentence structure.

5) === disappointingly , the characters are too strange and dysfunctional , tom included , to ever get under the skin , but this is compensated in large part by the off-the-wall dialogue , visual playfulness and the outlandishness of the idea itself .
Truth: 1, Prediction: -1 [WRONG]

strange	1 * 0.64 = 0.64
skin	1 * 0.56 = 0.56
ever	1 * 0.34 = 0.34
large	1 * 0.28 = 0.28
and	2 * 0.14 = 0.28
are	1 * 0.22 = 0.22
tom	1 * 0.22 = 0.22
part	1 * 0.12 = 0.12
of	1 * 0.12 = 0.12
is	1 * 0.08 = 0.08
playfulness	1 * 0.06 = 0.06

under	1 * 0.04 = 0.04
compensated	1 * 0 = 0
off-the-wall	1 * 0 = 0
outlandishness	1 * 0 = 0
visual	1 * -0.02 = -0.02
this	1 * -0.04 = -0.04
included	1 * -0.04 = -0.04
.	1 * -0.04 = -0.04
in	1 * -0.08 = -0.08
but	1 * -0.08 = -0.08
to	1 * -0.1 = -0.1
,	5 * -0.02 = -0.1
get	1 * -0.16 = -0.16
the	5 * -0.04 = -0.2
idea	1 * -0.22 = -0.22
disappointingly	1 * -0.24 = -0.24
by	1 * -0.3 = -0.3
dysfunctional	1 * -0.32 = -0.32
characters	1 * -0.38 = -0.38
itself	1 * -0.44 = -0.44
too	1 * -0.72 = -0.72
dialogue	1 * -0.84 = -0.84

The above example can be predicted correctly if the classifier had the knowledge of "X but Y" sentence structure thus taking the score of Y or the negation of the score of X.

6) === the film's messages of tolerance and diversity aren't particularly original , but one can't help but be drawn in by the sympathetic characters .
Truth: 1, Prediction: -1 [WRONG]

help	1 * 0.32 = 0.32
original	1 * 0.28 = 0.28
film's	1 * 0.26 = 0.26
and	1 * 0.14 = 0.14
of	1 * 0.12 = 0.12
drawn	1 * 0.02 = 0.02
diversity	1 * 0 = 0
tolerance	1 * 0 = 0
,	1 * -0.02 = -0.02
.	1 * -0.04 = -0.04
be	1 * -0.06 = -0.06
particularly	1 * -0.06 = -0.06
the	2 * -0.04 = -0.08
in	1 * -0.08 = -0.08
one	1 * -0.08 = -0.08
aren't	1 * -0.12 = -0.12
but	2 * -0.08 = -0.16
sympathetic	1 * -0.16 = -0.16
messages	1 * -0.24 = -0.24

by 1 * -0.3 = -0.3
characters 1 * -0.38 = -0.38
can't 1 * -0.62 = -0.62

The above example can be predicted correctly if the classifier had the knowledge of "X but Y" sentence structure thus taking the score of Y or the negation of the score of X.

7) === handled correctly , wilde's play is a masterpiece of elegant wit and artifice . here , alas , it collapses like an overcooked soufflé .
Truth: -1, Prediction: 1 [WRONG]
masterpiece 1 * 0.6 = 0.6
play 1 * 0.5 = 0.5
elegant 1 * 0.38 = 0.38
it 1 * 0.3 = 0.3
an 1 * 0.22 = 0.22
and 1 * 0.14 = 0.14
wilde's 1 * 0.12 = 0.12
of 1 * 0.12 = 0.12
wit 1 * 0.1 = 0.1
is 1 * 0.08 = 0.08
here 1 * 0.02 = 0.02
collapses 1 * 0.02 = 0.02
soufflé 1 * 0 = 0
correctly 1 * 0 = 0
overcooked 1 * 0 = 0
a 1 * -0.04 = -0.04
, 3 * -0.02 = -0.06
artifice 1 * -0.08 = -0.08
handled 1 * -0.08 = -0.08
. 2 * -0.04 = -0.08
like 1 * -0.18 = -0.18
alas 1 * -0.2 = -0.2

The above example can be predicted correctly if the classifier had the knowledge of "X, alas, Y" sentence structure thus taking the score of Y or the negation of the score of X.

Split on punctuations:

8) === this cloying , voices-from-the-other-side story is hell .
Truth: -1, Prediction: 1 [WRONG]
hell 1 * 0.5 = 0.5
story 1 * 0.18 = 0.18
is 1 * 0.08 = 0.08
voices-from-the-other-side 1 * 0 = 0
, 1 * -0.02 = -0.02
this 1 * -0.04 = -0.04
. 1 * -0.04 = -0.04
cloying 1 * -0.52 = -0.52

The above example can be predicted correctly if the classifier had split the term "voices-from-the-other-side" in multiple terms as "voices", "from", "the", "other" & "side".

Contextual meaning:
Here are two predictions that would have been classified correctly if the classifier had had the contextual meaning of some words or phrases. This can be achieved using features like words ngrams.

9) === an ill-conceived jumble that's not scary , not smart and not engaging .
Truth: -1, Prediction: 1 [WRONG]

engaging	1 * 0.62 = 0.62
smart	1 * 0.6 = 0.6
that's	1 * 0.42 = 0.42
an	1 * 0.22 = 0.22
and	1 * 0.14 = 0.14
jumble	1 * 0 = 0
,	1 * -0.02 = -0.02
.	1 * -0.04 = -0.04
scary	1 * -0.12 = -0.12
ill-conceived	1 * -0.38 = -0.38
not	3 * -0.26 = -0.78

The above example would have been classified correctly if the classifier understood the words "scary", "smart" & "engaging" along with the preceding "not". The word bigrams would be "not scary", "not smart" & "not engaging"

10) === wickedly funny , visually engrossing , never boring , this movie challenges us to think about the ways we consume pop culture .
Truth: 1, Prediction: -1 [WRONG]

funny	1 * 0.5 = 0.5
engrossing	1 * 0.5 = 0.5
culture	1 * 0.44 = 0.44
ways	1 * 0.32 = 0.32
us	1 * 0.3 = 0.3
wickedly	1 * 0.18 = 0.18
challenges	1 * 0.08 = 0.08
consume	1 * 0 = 0
think	1 * -0.02 = -0.02
visually	1 * -0.02 = -0.02
the	1 * -0.04 = -0.04
this	1 * -0.04 = -0.04
.	1 * -0.04 = -0.04
,	3 * -0.02 = -0.06
to	1 * -0.1 = -0.1
we	1 * -0.1 = -0.1
about	1 * -0.14 = -0.14
movie	1 * -0.2 = -0.2
pop	1 * -0.22 = -0.22

never	1 * -0.48 = -0.48
boring	1 * -1.26 = -1.26

The above example would have been classified correctly if the classifier understood the bigrams "never boring" instead of "never" & "boring".

2g) I ran a parameter sweep for the hyper-parameters in the following ranges:

- number of iterations, numIter, in the range 10 to 20 in increments of 1
- step size, eta, in the range 0.01 to 0.1 in increments of 0.01
- character ngrams, n, in the range 1-10 in increments of 1

Here are the top combinations of hyper-parameters for each value of n that gave the lowest devError:

n	numIters	eta	trainError	devError
6	15	0.02	0	0.266460326
7	15	0.02	0	0.266460326
8	15	0.02	0	0.266460326
9	15	0.02	0	0.266460326
10	15	0.02	0	0.266460326
5	16	0.01	0	0.269836804
4	15	0.01	0	0.27855937
3	20	0.02	0.000844119	0.314575127
2	17	0.01	0.291502532	0.396173326
1	18	0.06	0.446539111	0.464828362

The tuned hyper-parameters are as follow:

- character ngrams, n=6
- number of iterations numIter=15
- step size eta=0.02

The smallest devError=0.266460326 which is nearly equal to the smallest devError produced by word features which is 0.264772088. I think this is due to the fact that the average word size in the corpus is 6 and thus, in average, produces features similar to word features. The character 6gram also engulfs multiple stop words, which mostly are within a length of 4, into a single feature thus reducing negative contributions to the score.

Here is a review that was classified wrongly using word features but was classified correctly using character 6grams.

=== rain is a small treasure , enveloping the viewer in a literal and spiritual torpor that is anything but cathartic .
Truth: 1, Prediction: -1 [WRONG]

treasure	1 * 0.42 = 0.42
small	1 * 0.22 = 0.22
is	2 * 0.08 = 0.16
and	1 * 0.14 = 0.14
spiritual	1 * 0.06 = 0.06
enveloping	1 * 0.06 = 0.06
cathartic	1 * 0 = 0
,	1 * -0.02 = -0.02
the	1 * -0.04 = -0.04
rain	1 * -0.04 = -0.04
.	1 * -0.04 = -0.04
viewer	1 * -0.06 = -0.06
a	2 * -0.04 = -0.08
in	1 * -0.08 = -0.08
but	1 * -0.08 = -0.08
literal	1 * -0.08 = -0.08
that	1 * -0.1 = -0.1
torpor	1 * -0.26 = -0.26
anything	1 * -0.28 = -0.28

=== rain is a small treasure , enveloping the viewer in a literal and spiritual torpor that is anything but cathartic .
Truth: 1, Prediction: 1 [CORRECT]

easure	1 * 0.24 = 0.24
hingbu	1 * 0.14 = 0.14
ingthe	1 * 0.12 = 0.12
asmall	1 * 0.1 = 0.1
treasu	1 * 0.1 = 0.1
alands	1 * 0.1 = 0.1
spirit	1 * 0.1 = 0.1
reasur	1 * 0.1 = 0.1
thingb	1 * 0.08 = 0.08
raland	1 * 0.08 = 0.08
thatis	1 * 0.08 = 0.08
envelo	1 * 0.06 = 0.06
pingth	1 * 0.06 = 0.06
nvelop	1 * 0.06 = 0.06
isasma	1 * 0.04 = 0.04
asure,	1 * 0.04 = 0.04
isanyt	1 * 0.04 = 0.04
sasmal	1 * 0.04 = 0.04
eralan	1 * 0.04 = 0.04

sanyth	1 * 0.04 = 0.04
ingbut	1 * 0.02 = 0.02
dspiri	1 * 0.02 = 0.02
loping	1 * 0.02 = 0.02
ndspir	1 * 0.02 = 0.02
tisany	1 * 0.02 = 0.02
landsp	1 * 0.02 = 0.02
elopin	1 * 0.02 = 0.02
piritu	1 * 0.02 = 0.02
velopi	1 * 0.02 = 0.02
ltreas	1 * 0.02 = 0.02
iritua	1 * 0.02 = 0.02
ritual	1 * 0.02 = 0.02
viewer	1 * 0.02 = 0.02
inalit	1 * 0.02 = 0.02
rinali	1 * 0 = 0
utcath	1 * 0 = 0
lltrea	1 * 0 = 0
aliter	1 * 0 = 0
,envel	1 * 0 = 0
sure,e	1 * 0 = 0
athart	1 * 0 = 0
malltr	1 * 0 = 0
itualt	1 * 0 = 0
ltorpo	1 * 0 = 0
butcat	1 * 0 = 0
rthati	1 * 0.0 = 0.0
ualtor	1 * 0 = 0
tharti	1 * 0 = 0
alltre	1 * 0 = 0
nisasm	1 * 0 = 0
altorp	1 * 0 = 0
artic.	1 * 0 = 0
tcatha	1 * 0 = 0
hatisa	1 * 0.0 = 0.0
gbutca	1 * 0 = 0
orport	1 * 0 = 0
portha	1 * 0 = 0
rainis	1 * 0 = 0
terala	1 * 0 = 0
nalite	1 * 0 = 0
cathar	1 * 0.0 = 0.0
tualto	1 * 0 = 0
ainisa	1 * 0 = 0
hartic	1 * 0 = 0
erinal	1 * 0 = 0
smallt	1 * 0 = 0
ure,en	1 * 0 = 0

inisas	1 * 0 = 0
atisan	1 * 0 = 0
werina	1 * 0 = 0
re,env	1 * 0 = 0
e,enve	1 * 0 = 0
rporth	1 * 0 = 0
nythin	1 * -0.02 = -0.02
ewerin	1 * -0.02 = -0.02
opingt	1 * -0.02 = -0.02
ieweri	1 * -0.02 = -0.02
andspi	1 * -0.02 = -0.02
litera	1 * -0.04 = -0.04
torpor	1 * -0.04 = -0.04
orthat	1 * -0.04 = -0.04
gthevi	1 * -0.04 = -0.04
ngthev	1 * -0.04 = -0.04
thevie	1 * -0.06 = -0.06
heview	1 * -0.06 = -0.06
eviewe	1 * -0.06 = -0.06
anythi	1 * -0.08 = -0.08
iteral	1 * -0.08 = -0.08
ngbutc	1 * -0.1 = -0.1
ything	1 * -0.18 = -0.18

2h) I have implemented word ngrams along with the word features for the extra credit features. The reason why word ngrams make the feature set better is because these ngrams provides some contextual knowledge to the classifier to do better prediction.

For example, using just word features would give us
=== wickedly funny , visually engrossing , never boring , this movie challenges us to think about the ways we consume pop culture .
Truth: 1, Prediction: -1 [WRONG]

funny	1 * 0.5 = 0.5
engrossing	1 * 0.5 = 0.5
culture	1 * 0.44 = 0.44
ways	1 * 0.32 = 0.32
us	1 * 0.3 = 0.3
wickedly	1 * 0.18 = 0.18
challenges	1 * 0.08 = 0.08
consume	1 * 0 = 0
think	1 * -0.02 = -0.02
visually	1 * -0.02 = -0.02
the	1 * -0.04 = -0.04
this	1 * -0.04 = -0.04
.	1 * -0.04 = -0.04
,	3 * -0.02 = -0.06
to	1 * -0.1 = -0.1
we	1 * -0.1 = -0.1
about	1 * -0.14 = -0.14
movie	1 * -0.2 = -0.2
pop	1 * -0.22 = -0.22
never	1 * -0.48 = -0.48
boring	1 * -1.26 = -1.26

But using word trigrams along with word features would give us
=== wickedly funny , visually engrossing , never boring , this movie challenges us to think about the ways we consume pop culture .
Truth: 1, Prediction: 1 [CORRECT]

funny	1 * 0.48 = 0.48
us	1 * 0.36 = 0.36
culture	1 * 0.34 = 0.34
ways	1 * 0.24 = 0.24
,	3 * 0.08 = 0.24
to think about	1 * 0.16 = 0.16
wickedly	1 * 0.1 = 0.1
engrossing	1 * 0.1 = 0.1
we	1 * 0.08 = 0.08
the ways we	1 * 0.08 = 0.08
visually	1 * 0.02 = 0.02
consume	1 * 0 = 0
movie challenges us	1 * 0 = 0
wickedly funny ,	1 * 0 = 0

, never boring	1 * 0 = 0
challenges	1 * 0.0 = 0.0
, visually engrossing	1 * 0 = 0
never boring ,	1 * 0 = 0
think about the	1 * 0 = 0
consume pop culture	1 * 0 = 0
challenges us to	1 * 0 = 0
pop culture .	1 * 0 = 0
visually engrossing ,	1 * 0 = 0
we consume pop	1 * 0 = 0
funny , visually	1 * 0 = 0
boring , this	1 * 0 = 0
ways we consume	1 * 0 = 0
about the ways	1 * 0 = 0
us to think	1 * 0 = 0
this movie challenges	1 * 0 = 0
engrossing , never	1 * 0 = 0
think	1 * -0.02 = -0.02
, this movie	1 * -0.02 = -0.02
to	1 * -0.06 = -0.06
pop	1 * -0.06 = -0.06
.	1 * -0.08 = -0.08
the	1 * -0.12 = -0.12
this	1 * -0.12 = -0.12
about	1 * -0.14 = -0.14
movie	1 * -0.18 = -0.18
never	1 * -0.2 = -0.2
boring	1 * -0.7 = -0.7

Initially the features “never” & “boring” contributed -1.74 to the score but using word trigrams along with the word features made the features “never”, “boring” & “never boring” contributed just -0.9.

Problem 3:

3a) $D_{train} = \{x_1, x_2, x_3, x_4\}$

$\phi(x_1) = [1, 0]$

$\phi(x_2) = [2, 1]$

$\phi(x_3) = [0, 0]$

$\phi(x_4) = [0, 2]$

1) Let the initial cluster location be

$\mu_1 = [0, -1]$

$\mu_2 = [2, 2]$

Iteration 1:

a) Assigning data points to clusters:

$dist(\phi(x_1), \mu_1) = (1-0)^2 + (0+1)^2 = 2$

$dist(\phi(x_1), \mu_2) = (1-2)^2 + (0-2)^2 = 5$

x_1 is assigned to μ_1

$dist(\phi(x_2), \mu_1) = (2-0)^2 + (1+1)^2 = 8$

$dist(\phi(x_2), \mu_2) = (2-2)^2 + (1-2)^2 = 1$

x_2 is assigned to μ_2

$$dist\left(\phi(x_3),\ \mu_1\right) = (0-0)^2 + (0+1)^2 = 1$$
$$dist\left(\phi(x_3),\ \mu_2\right) = (0-2)^2 + (0-2)^2 = 8$$

x_3 is assigned to μ_1

$$dist\left(\phi(x_4),\ \mu_1\right) = (0-0)^2 + (2+1)^2 = 9$$
$$dist\left(\phi(x_4),\ \mu_2\right) = (0-2)^2 + (2-2)^2 = 4$$

x_4 is assigned to μ_2

b) Updating clusters:

$$\mu_1 = \left[\frac{1+0}{2},\ \frac{0+0}{2}\right] = \left[\frac{1}{2},\ 0\right]$$

$$\mu_2 = \left[\frac{2+0}{2},\ \frac{1+2}{2}\right] = \left[1,\ \frac{3}{2}\right]$$

Iteration 2:

a) Assigning data points to clusters:

$$dist\left(\phi(x_1),\ \mu_1\right) = \left(1-\frac{1}{2}\right)^2 + (0-0)^2 = 1/4$$
$$dist\left(\phi(x_1),\ \mu_2\right) = (1-1)^2 + \left(0-\frac{3}{2}\right)^2 = 9/4$$

x_1 is assigned to μ_1

$$dist(\phi(x_2), \mu_1) = (2 - 1/2)^2 + (1-0)^2 = 1 + \frac{9}{4}$$
$$dist(\phi(x_2), \mu_2) = (2-1)^2 + (1 - \frac{3}{2})^2 = 1 + \frac{1}{4}$$

x_2 is assigned to μ_2

$$dist(\phi(x_3), \mu_1) = (0 - 1/2)^2 + (0-0)^2 = 1/4$$
$$dist(\phi(x_3), \mu_2) = (0-1)^2 + (0 - \frac{3}{2})^2 = 1 + 9/4$$

x_3 is assigned to μ_1

$$dist(\phi(x_4), \mu_1) = (0 - 1/2)^2 + (2-0)^2 = 4 + 1/4$$
$$dist(\phi(x_4), \mu_2) = (0-1)^2 + (2 - 3/2)^2 = 1 + 1/4$$

x_4 is assigned to μ_2

~~b) Updating~~

As the assignments did not change in comparison to iteration 1, we can declare the clusters as,

Cluster 1 : $\{x_1, x_3\}$

Cluster 2 : $\{x_2, x_4\}$

2) Let the initial clusters be

$\mu_1 = [2, 0]$

$\mu_2 = [-1, 0]$

Iteration 1:

a) Assigning data points to clusters:

$dist(\phi(x_1), \mu_1) = (1-2)^2 + (0-0)^2 = 1$

$dist(\phi(x_1), \mu_2) = (1+1)^2 + (0-0)^2 = 4$

x_1 is assigned to μ_1

$dist(\phi(x_2), \mu_1) = (2-2)^2 + (1-0)^2 = 1$

$dist(\phi(x_2), \mu_2) = (2+1)^2 + (1-0)^2 = 10$

x_2 is assigned to μ_1

$dist(\phi(x_3), \mu_1) = (0-2)^2 + (0-0)^2 = 4$

$dist(\phi(x_3), \mu_2) = (0+1)^2 + (0-0)^2 = 1$

x_3 is assigned to μ_2

$dist(\phi(x_4), \mu_1) = (0-2)^2 + (2-0)^2 = 8$

$dist(\phi(x_4), \mu_2) = (0+1)^2 + (2-0)^2 = 5$

x_4 is assigned to μ_2

b) Updating the clusters:

$$\mu_1 = \left[\frac{1+2}{2}, \frac{0+1}{2}\right] = \left[\frac{3}{2}, \frac{1}{2}\right]$$

$$\mu_2 = \left[\frac{0+0}{2}, \frac{0+2}{2}\right] = [0, 1]$$

Iteration 2:

a) Assigning the data points to clusters:

$$\text{dist}(\phi(x_1), \mu_1) = (1 - 3/2)^2 + (0 - 1/2)^2 = 1/2$$
$$\text{dist}(\phi(x_1), \mu_2) = (1 - 0)^2 + (0 - 1)^2 = 2$$

x_1 is assigned to μ_1

$$\text{dist}(\phi(x_2), \mu_1) = (2 - 3/2)^2 + (1 - 1/2)^2 = 1/2$$
$$\text{dist}(\phi(x_2), \mu_2) = (2 - 0)^2 + (1 - 1)^2 = 4$$

x_2 is assigned to μ_1

$$\text{dist}(\phi(x_3), \mu_1) = (0 - 3/2)^2 + (0 - 1/2)^2 = 5/2$$
$$\text{dist}(\phi(x_3), \mu_2) = (0 - 0)^2 + (0 - 1)^2 = 1$$

x_3 is assigned to μ_2

$$dist(\phi(x_4), \mu_1) = (0 - 3/2)^2 + (2 - 1/2)^2 = 9/2$$

$$dist(\phi(x_4), \mu_2) = (0 - 0)^2 + (2 - 1)^2 = 1$$

x_4 is assigned to μ_2

As the assignment did not change in comparison with iteration 1, we can declare the clusters as,

Cluster 1 : $\{x_1, x_2\}$

cluster 2 : $\{x_3, x_4\}$

3c) * Let S be the set of example pairs ~~that~~ where each example pair belongs in the same cluster.

Eg: $S = \{(1,2), (2,3), (4,5)\}$

* Process the set S to do an Union-Find and group all data points that belong to the same cluster.

$$S = \{(1,2,3), (4,5)\}$$

* Replace all sets of data points in S with a surrogate data point which is the average of all the data points in the set.

Let $A = avg(1,2,3)$
$B = avg(4,5)$

$$S = \{A, B\}$$

* Perform kmeans clustering with the new set of data where data that belong to the same cluster are represented by its surrogate point.

* After kmeans clustering, replace all the surrogate points with its original data points and report the clusters.

* As all the data points that must belong to the same cluster are represented by a single data point during clustering, it can be guaranteed that all the data points that must belong to the same cluster do belong in the same cluster.