Lihan Yao ML, Spring 2017
Homework 3: Sentiment Analysis

1 Introduction

2 Calculating Subgradients

2.1

$g \in \mathbf{R}^d$ is a subgradient of $\delta f_k(x)$. $\forall z$ we have,

$$f_k(z) \geq f_k(x) + g^T(z-x)$$
$$f_k(z) \geq f(x) + g^T(z-x)$$
$$f(z) \geq f_k(z) \geq f(x) + g^T(z-x)$$

Where the second inequality follows from $f_k(x) = f(x)$, and the last inequality follows from f being the pointwise maximum function of the convex function sequence $\{f_i\}_{i=1,\dots,m}$. The resulting inequality implies $g \in \delta f(x)$.

2.2

At a w where $yw^Tx > 1$, $g = 0$ is a subgradient of $J(w)$. First compute the RHS of the subgradient inequality: $J(w) - g^T(z-w) = \max\{0, 1 - yw^Tx\} + 0(z-w) = 0$ for any z. The LHS is $J(z) = \max\{0, 1 - yz^Tx\} \geq 0$. In total this is:

$$J(z) \geq J(w) + 0(z-w)$$

So $g = 0$ is a subgradient at certain values of w.

3 Perceptron

3.1

It is enough to show the perceptron loss on any sample in D equal 0. Consider $(x_i, y_i) \in \mathbf{R}^d \times \{-1, 1\}$. Since the hyperplane $\{x | w^Tx = 0\}$ exists, D is linearly separable and the perceptron algorithm converges to some w. In order to have met the algorithm's termination condition, w must have the property that:

$$y_i x_i^T w > 0$$

The perceptron loss at this sample is then $l(w^Tx_i, y_i) = \max\{0, -y_i x_i^T w\} = 0$.

3.2

From the lecture notes, SSGD is described as:

$$\text{For } k = 1, 2, \ldots \tag{1}$$
$$w^{(k+1)} \leftarrow w^{(k)} - \alpha_k g \tag{2}$$
$$f_{\text{best}}^{(k+1)} \leftarrow \min_{i=1,\ldots,k+1} f(w^{(i)}) \tag{3}$$

where $g \in \delta f(x^{(k-1)})$ and α_k is the step size at time step k. First we modify the update step at (2). Setting $\alpha_k = 1$, and choosing $g = -y_k x_k$, the update step becomes

$$w^{(k+1)} \leftarrow w^{(k)} + y_k x_k$$

Our choice of g follows from differentiating the perceptron loss $l(w) = \max\{h(w), p(w)\}, \quad p(w) = -yw^T x$.

$$\frac{\partial}{\partial w_i} p(w) = -\sum_i y_i x_i$$

With an additional restriction, we may terminate once SSGD reaches step j where $f(w^{(j)}) = 0$. Specifically, after we find $w^{(j)}$ s.t. $l(w^T x_i, y_i) = \max\{0, -w^T x_i y_i\} = 0$, we still need to go through all samples in one more epoch to match the behavior dictated by 'all_correct'. For the first termination condition, in order for lost expression to equal zero, it is equivalent to demand $y_i x_i^T w^{(k)} > 0$ for every sample. We have:

```
While $\sum_i l(w^{(k)T} x_i, y_i) > 0$ :                (4)
  For $i = 1, 2, \ldots$                                 (5)
    If $(y_i x_i^T w^{(k)} \leq 0)$ :                     (6)
      $w^{(k+1)} \leftarrow w^{(k)} + y_k x_k$            (7)
    Else Continue                                         (8)
  Go through samples one more time                        (9)
```

The while condition ensures that the algorithm does not converge if the data is not linearly separable, just as in the perceptron case.

3.3

Proof by induction. Base case $i = 1$:
If $y_1 x_1^T w > 0$, set $a_1 = 0$ and we have $w = 0$. Otherwise $w = y_1 x_1$, a linear combination of the input points. Now by induction hypothesis suppose on step n, we have for some choice of $a_1, \ldots, a_n$:

$$w = \sum_{i=1}^{n} a_i x_i$$

For the $n+1$th sample, if $y_{n+1} x_{n+1}^T w > 0$, set $a_{n+1} = 0$ and we have w as before. If $y_{n+1} x_{n+1}^T w \leq 0$, the algorithm may go back and take another s_i steps of the form $w^{k+1} = w^k + y_i x_i$ on the ith sample.

Let $b_i = s_i y_i$, where s_i is the number of steps that the algorithm has taken for ith sample. We have:

$$w = \sum_{i=1}^{n+1} b_i x_i$$

Again a linear combination of x_is.

The support vectors are samples for which at least one correction step has been taken. They alter the w vector. The non-support vectors with coefficient equal 0 are on the correct side of the hyperplane every time it is visited by the algorithm.

4 The Data

4.1

```
def shuffle_data():
    '''
    pos_path is where you save positive review data.
    neg_path is where you save negative review data.
    '''
    pos_path = "C:/Users/Lihan/Documents/WORD/2017 Spring/Machine Learning/hw3/txt_sent
    neg_path = "C:/Users/Lihan/Documents/WORD/2017 Spring/Machine Learning/hw3/txt_sent

    pos_review = folder_list(pos_path,1)
    neg_review = folder_list(neg_path,-1)

    review = pos_review + neg_review
    random.shuffle(review)

    pickle.dump(review[0:1500], open( "train.p", "wb" ) )
    pickle.dump(review[1500:2000], open( "valid.p", "wb" ) )
    return 0
```

5 Sparse Representations

5.1

```
import collections as c
def convert(input_list):
    return c.Counter(input_list)
```

5.2

6 Support Vector Machine via Pegasos

6.1

The SVM objective $J(w)$ is the sum of two convex functions: the vector norm $\frac{\lambda}{2}||w||^2$ and

$$\frac{1}{m}\sum_{i=1}^{m}\max\{0, 1 - y_i w^T x_i\}$$

The latter hinge loss is the maximum of two linear functions, which are individually convex and their maximum is convex. Now invoke the hints, which together implies linearity of the partial derivative operator when applied to convex functions. For given sample (x_i, y_i):

$$\frac{\partial}{\partial w}J(w) = \frac{\lambda}{2}\frac{\partial}{\partial w}||w||^2 + \frac{\partial}{\partial w}\max\left(0, 1 - y_i w^T x_i\right)$$

For w_t at the t-th step, if $y_i w_t^T x_i < 1$, this is:

$$\frac{\partial}{\partial w}J(w) = \lambda w + (-y_i x_i)$$

Now if $y_i w_t^T x_i \geq 1$: $\frac{\partial}{\partial w}J(w) = \lambda w$. Our update step is now

$$\begin{aligned} w_{t+1} &= w_t - \eta_t \frac{\partial}{\partial w_t}J(w_t) & (10) \\ &= w_t - \eta_t(\lambda w_t - y_i x_i) & (11) \\ &= w_t - \eta_t \lambda w_t + \eta_t y_i x_i & (12) \end{aligned}$$

and likewise if $y_i w_t^T x_i \geq 1$, $w_{t+1} = w_t - \eta_t \lambda w_t$. This is what was shown in the pseudocode.

6.2

To update the Pegasos dictionary, I call the update() dictionary method to modify values in place. The termination condition is number of epochs.

```
def pegasos1(X, y, lamb, num_iter):
    w = dict()
    t = 0

    for iteration in range(num_iter):
        for j in range(len(X)):
            t += 1
            step = 1/(t*lamb)

            if y[j]*dotProduct(w,X[j]) < 1:
                w.update( (y, x*(1-lamb*step)) for y, x in w.items())
                increment(w, step*y[j], X[j])
            else:
                w.update( (y, x*(1-lamb*step)) for y, x in w.items())
    return w
```

6.3

The new method is indeed equivalent to the original.

$$w_{t+1} = s_{t+1}W_{t+1} = (1-\eta_t\lambda)s_t(W_t + \frac{1}{s_{t+1}}\eta_t y_i x_i) = (1-\eta_t\lambda)s_t W_t + \eta_t y_i x_i$$

```
def pegasos2(X, y, lamb, num_iter):
    w = dict()
    W = dict()
    t = 0
    s = 1
    for iteration in range(num_iter):
        for j in range(len(X)):
            t += 1
            step = 1/(t*lamb)
            s *= 1-(step*lamb)
            if s == 0:
                s = 1
                W = dict()

            if y[j]*dotProduct(w,X[j]) < 1:
                increment(W, step*y[j]/s, X[j])
            w.update( (y, x*s) for y, x in W.items())
    return w
```

6.4

With a time module wrapper, no preprocessing or feature engineering, the original method completes one epoch on the training data in 16.7 seconds, while the second method completes one epoch in 9.64 seconds. If I access a particular dictionary key such as 'and', they both return the same values, as desired.

In addition, I called sklearn's DictVectorizer to convert the training and validation dictionaries into arrays, and instead of using our given 'dotProduct' and 'increment', I use corresponding numpy operations. The performance of this Pegasos version is under the name 'Pegasos3':

```
v_l, t_l, w = pegasos3(X_train_v, y_train, lamb = 1, num_iter = 1)
#print(v.inverse_transform(w)[0])
c =v.inverse_transform(w)[0]['long able']
```

```
a= pegasos1(X_train, y_train, lamb = 1, num_iter = 1)['long able']
b= pegasos2(X_train, y_train, lamb = 1, num_iter = 1)['long able']
print(a, b, c)
```

Output for this jupyter cell is:

```
'pegasos3' 6.87056 sec
'pegasos1' 136.05831 sec
'pegasos2' 75.81657 sec
-0.0006666666666666668 -0.0006666666666666667 -0.000666666666666667
```

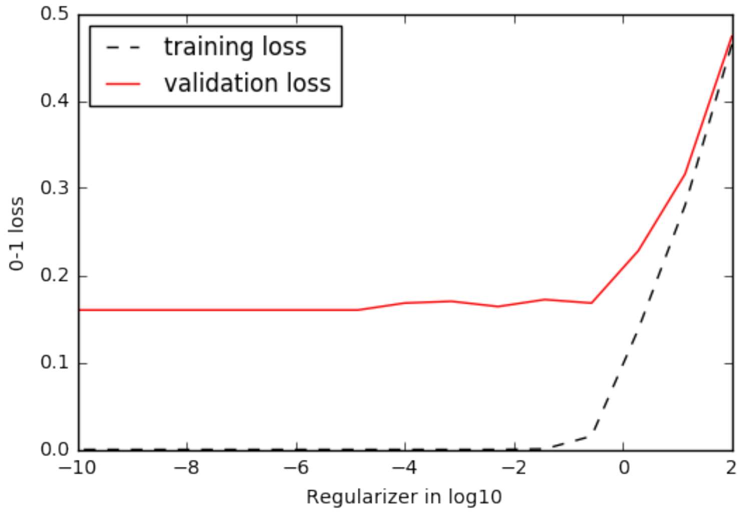
6.5

The loss function below tallies the boolean evaluation $y_i \neq \text{sgn}(w^T x_i)$ in a loss variable.

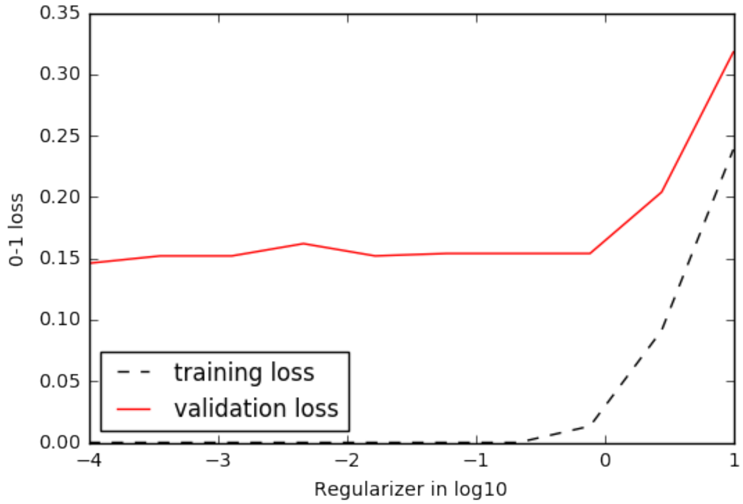
```
def loss(w, X, y):
    #zero one loss of linear predictor
    loss = 0
    for i in range(X.shape[0]):
        loss += y[i] != np.sign((X[i].dot(w.T)))
        # add boolean to loss whenever pred is incorrect
    return (loss/X.shape[0])
```

6.6

After 90 epochs, with no preprocessing or feature engineering, the λ value I chose is $10e-3$. The close up of the graph in the range $[10e-4, 10e-2]$ is a nearly flat line so I have decided not to include it:



On later problems, I have since had to recalibrate my regularization parameter. The graph below is produced after 70 epochs per point, with a non-alphabet filter, lemmatization and bigrams. From the graph data, I again chose $\lambda = 10e - 3$:



6.7

6.8

7 Error Analysis

7.1

Features that contributed most to bad decisions include '.', 'have', and 'and', with the period symbol having the highest contribution $|w_i x_i|$ in all three examples. For example, in the most confident mistake with score of 1.44 (review 1432 below), '.' has a weight of -0.02 but by sheer number of occurrences in the review, has contributed -1.91 to the classification of the review, when it should have the $+1$ label. From these bad features I can conclude that removing stopwords, removing non-alphabet symbols, and adding bigrams (for example 'not good' should be a tell-tale sign of a negative review, the exact opposite of 'good') will have a positive effect on SVM performance.

Review 1432. In this instance there were a lot of ellipses (the added forward-slash character by the database alters the output). I will list the top three negative features in the format of (word, w_i score, $|w_i x_i|$), we may infer number of occurrences accordingly:
('.' , -0.022, 1.91)
('have' , -0.05, 0.315)
('as', 0.03, 0.3)

'a couple of months ago , when i first downloaded the face/off trailer from the net , my initial reaction was a fourteen carrot gold ýawn´ ;followed swiftly by a press of my computers delete key , not wanting to waste six or seven megabytes of precious space on this piss-poor trailer . ;then i started reading the first wave of reviews from the u . s . . . . . unique . . . excellent . . . . must-see . ;well , i thought still skeptical , i suppose i might as well go see it when it gets here . ;h́ere,́ of course , was still three months away . ;iĺl admit , when i trotted off to see this film , the only john woo movie i had seen before was the fairly enjoyable but highly forgetable broken arrow . ;id́ heard good things about his previous work with movies like hard boiled , but his films were definitely not on my must-see list . ;that , let me tell you , has changed completely . ;i knew this fact only five minutes into the film , after the brilliantly shot and acted opening sequence where sean archer loses his son blew me off my feet . ;the acting throughout the film is staggeringly good for an action flick . ;seriously . ;iv́e never been a big john travolta fan , but he , like cage , perfectly suited his role in the movie . . . . . sorry , ;make that ŕolesín the movie . ;even travoltaś great performance , however , paled in comparison to cageś character portrayals . ;my favourite cage scene was definitely when he was crashed out in compardre dietrich hasslerś hideout . ;half drugged out of his mind , he sits there reclining back in a chair talking about his sonś death from castorś perspective - " doesnt́ it just break your heart . " he mutters coldly . ;now dont́ go thinking from the previous comments about brilliant acting that this is a drama focused movie - itś not . ;thatś what really makes this movie unique . ;itś an action movie with brilliantly portrayed characters . . . . . not a common mix at all . ;suspension of disbelief is paramount in this movie though . . . . . thatś ;the only way to overlook the fact that travolta and cage fire around 5671 rounds at each other . . . . and ;never hit . ;several action scenes are just so well choreographed that they just make wish that you could press ŕewindánd watch it over again . ;the part where archer and troy have a stand off on either side of a double-sided mirror is just plain brilliant . ;whilst cageś and travoltaś performances would be enough alone to sustain most movies , the lesser characters are just as intriguing . ;joan allen , who usually sticks to the straight drama movies , plays her part perfectly as archerś long suffering wife . ;i like the fact that her character didnt́ end up toting a gun at the end of the movie . . . it ;would have wrecked her potrayal . ;gina gershon was surprising to say the least as castor troyś mistreated girlfriend . . . . . make that * one of * castor troyś mistreated girlfriends . ;other movies would have used her character as just window dressing . . . a ;sex object , but instead her character is very strong and independent . ;dominique swain , who plays archerś daughter , also does a nice job , though her character is not as explored as much as allenś or gershonś . ;faults ? well . . . there were some i have to say . ;first , the movieś ending , whilst being very good overall , was a bit too drawn out for itś own good . ;after the two combatants begin duelling again after the final boat crash you cant́ help but think - " geez , are you guys nuclear powered or what ? " . ;there were certain bits that werent́ handled properly , like where archer appears safe and sound on solid ground after jumping off the converted oil platform/prison - more explanation here would have been nice . ;overall , this movie was not perfect . ;but i thought it was about as close as an action movie has ever came to perfect . ;many critics have claimed that this movie will change the way action movies are made . . . . . i ;certainly hope so .'

In Review 372 there were ellipses (again) and quotations, which contributed to a negative classification. There were also plenty of stopwords, which end up contributing a lot to the output prediction even if they are low weight in w_i, which I do not consider to be informative features:
('.' , -0.022, 0.955)
('"' , -0.02, 0.53)
('and', 0.03, 0.48)

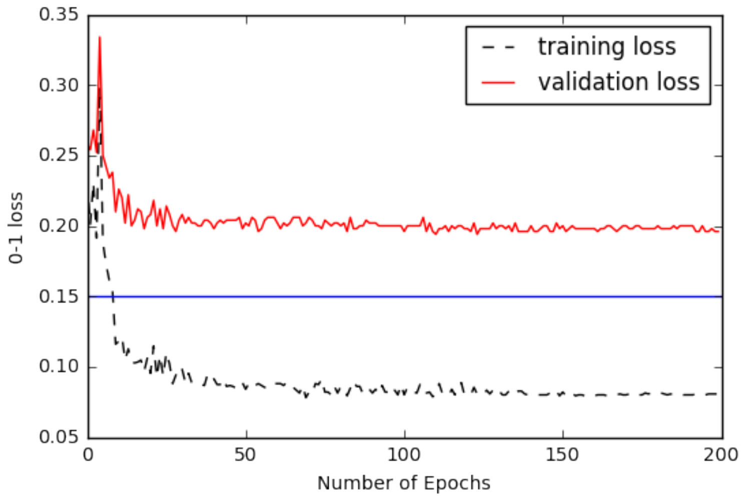
'i didnt́ realize how apt the name of this movie was until i called the mpaa (the motion picture association of america - the folks who decide whatś g , nc 17 , pg , r or x) to ask why the preview was rated r . so that we can make some sense of their response , let me tell you about the movie itself . ; " the celluloid closet " is a documentary about how homosexuality has been portrayed in the movies over the past several decades . ;itś brilliant , funny , naughty and extremely poignant . ;it tore at my heart to watch a gifted lesbian screenwriter explain that , as a rule , gay audiences hunger for any hint of homosexuality on screen . ;regardless of how veiled - or how sordid - the presence of a gay or lesbian person allows others to lessen their sense of isolation and makes them feel as if theyŕe not quite so invisible as america seems to want them to be . ;the movie itself is rated r - and for good reason . ;it contains scenes of bloody , violent gay bashing and graphic , uninhibited , sex . ;as with any movie , i appreciate knowing about these things ahead of time , so i can decide for myself whether to see the movie with a friend , a date , my 11 year old niece , alone or not at all . ;but , thatś the movie . ;now back to the preview . ;prior to this film being theatrically released (it was originally filmed as a documentary for hbo) i had seen the coming attractions trailer for it at least six times . ;there was no nudity , no violence , no bad language , nothing that i could see that would be offensive or inappropriate for a general audience (okay , whoopi goldberg did refer to someone " boning " someone , but the last i knew that wasnt́ one of the seven words you cant́ say on tv) . ;except for a scene of two fully clothed men kissing . ;hmmmmm . ;when i inquired about the rating on the trailer , a very nice woman at the mpaa quoted from " the handbook " that a trailer approved for all audiences could contain " no homosexuality or lesbianism and no going down on someone . " ;hello ? i was in the office and it was the middle of the day . ;bravely , i pursued . ; " iv́e seen that trailer , oh . . . ;probably half a dozen times , " i gulped . . . ; " and i dont́ remember that scene . " ; " well , " she chirped . ; " itś there . ;our little eyes are trained to see that . " ;no really . ;in the words of dave barry , " i am not making this up . " ;they are " trained " to " see that ? " ;when someone who was shocked at the rating the first time and made a note to watch it carefully the following five times or so managed to let it slip past her ? ;gosh , i certainly dont́ mean to question the mpaa , or " the handbook " . ;i would , however , like to suggest that itś they who are in the closet on this one . ;and the light aint́ good in there . ;but , having seen " the celluloid closet , " and being one of a handful of straight people involved in a primarily gay and lesbian weekly bible study (email me and iĺl give you the details) , none of this was any big surprise . ;the point of the movie was that homosexuality , even in the politically correct 90s , is ridiculously perceived as a threat to a mostly heterosexual society . ;a point well made in this candid and honest film . ;now , i could go off on the mpaaś ruling that a trailer must contain " no homosexuality or lesbianism " and ask how that is defined , particularly in light of some of the things , both sexual and non-sexual , that iv́e watched straight people do in trailers . ;i just dont́ feel the need to go there , because it seems so obvious . ;iĺl instead suggest that the mpaa re-evaluated their evaluation criteria . ;let the ratings reflect not subject content , like " sex " and " violence " . ;let them reflect attitude content . ;in the future , id́ be interested in knowing whether the movie is rated d for disrespectful or s for stereotyped . ;then id́ truly be able to make an informed decision about how i spent my time .'

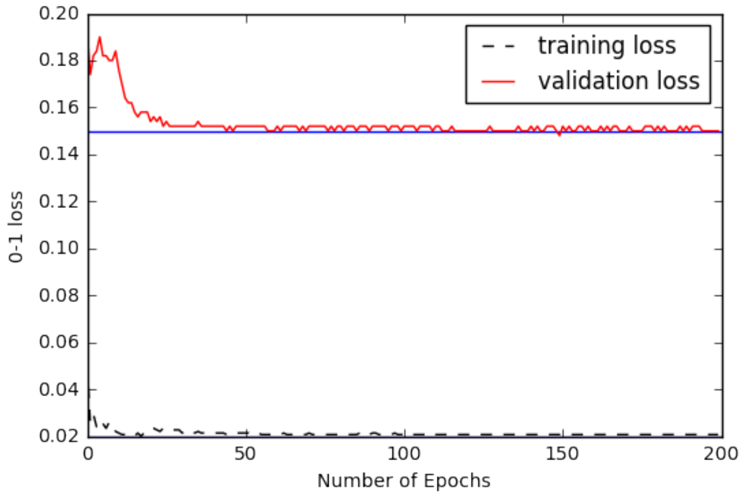
Both of these examples led me to consider closely the elimination of uninformative features in the form of preprocessing. I have taken out stopwords and used regex to eliminate non-alphabeticals, though admittedly these features may be informative in counterintuitive ways.

8 Features

8.1

To act on observations from the previous question, I have since included regex to filter out non-alphabeticals, lemmatization and bigrams to the preprocessing and feature engineering stages. With $\lambda = 10e - 5$, for vanilla pegasos, my validation error converged to 21 % after 200+ iterations. For my model with improved features, validation error hovers around 15% after 200+ iterations.





```
def process_data(data):
    X = []
    y = []

    for review in data:
        y.append(review[-1])
        #nltk stopword removal as part of pre-processing
        s_filtered = [re.sub(r"[^A-Za-z]+", '', word) for word in review[0:-2]] #elimin
        s_filtered = [word for word in s_filtered if word not in stopwords.words('engli
        s_filtered = list(filter(None, s_filtered))
        #POS tagging then convert to bag of words dicts
        POS_tagged = pos_tag(s_filtered)
        POS_tagged = [(word, p) for word, p in POS_tagged if word.isalpha()]

        #lemmad = []
        for word, p in POS_tagged:
            p = get_wordnet_pos(p)
            lemmad.append(lem.lemmatize(word, pos = get_wordnet_pos(p)))

        #add bigrams
        for bgram1, bgram2 in ngrams(s_filtered, 2):
            lemmad.append(bgram1+' '+bgram2)
        X.append(c.Counter(lemmad))

        ###vanilla
        #X.append(c.Counter(review[0:-2]))
    del data
    return X,y

#Transform a vectorized version of datasets:
from sklearn.feature_extraction import DictVectorizer
from scipy.sparse import csr_matrix
v = DictVectorizer()
X_train_v = v.fit_transform(X_train)
X_valid_v = v.transform(X_valid)
```

The standard errors is

$$\sqrt{\frac{p(1-p)}{n}} = \sqrt{\frac{0.15(0.85)}{500}} = 0.016$$

Since SE is a measure of the estimate precision, a small SE implies low standard deviation in our iid selection from D, which is a 500-sized validation set. The sample mean is 0.15. We may construct a 98% confidence interval of

$$\left(\bar{x} - z^* \frac{\sigma}{\sqrt{n}}, \bar{x} + z^* \frac{\sigma}{\sqrt{n}}\right) = \left(0.15 - 2.326\frac{0.016}{\sqrt{500}}, 0.15 + 2.326\frac{0.016}{\sqrt{500}}\right) = (0.148, 0.152)$$

So the model's true error is very much free of how we sample our reviews.

8.2