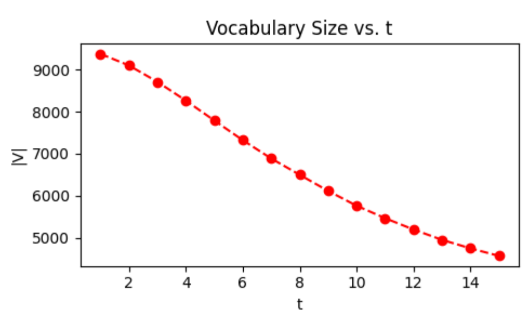
6.C01 PSET 2
Kaden DiMarco

1.1

```python
def build(sents, min_freq=10):
    """
    build the vocabulary from the sentences <sents>, include all words that appear more than <min_freq> times.

    Return:
        a list of strings (words), where each string is a token appeared in <sents>.

    Example:
        sents = [['I', 'like', 'banana'], ['I', 'like', 'apple']]
        min_freq = 1
        Output: ['I', 'like', 'banana', 'apple', '<oov>'] (order doesn't matter)
    """
    v = {}
    v['<oov>'] = 1

    # code to build your vocabulary goes here

    vocab = {}
    for sent in sents:
        for item in sent:
            if vocab.get(item) is None:
                vocab[item] = 1
            else:
                vocab[item] = vocab[item] + 1
            if vocab[item] >= min_freq:
                v[item]=1

    return v.keys()

# Question 1.1
# ##############################
# include updated copy of build funtion in your submission
####################################
sents = load_sent('./positive.txt') + load_sent('./negative.txt')
#words are counted multiple times in each sentement
vocab = build(sents, 10)
print(len(vocab))
#len(vobab) = 5760
```

1.2

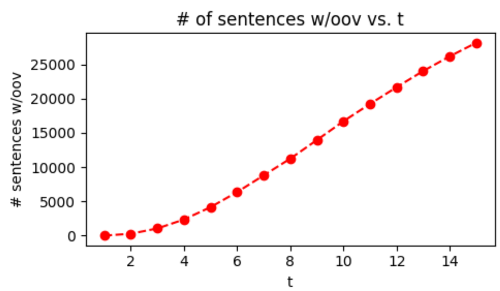
```python
# Question 1.2
vocab_size = [5760]
t = list(range(1,16))
# ##############################
# your code goes here
vocab_size = []
for j in t:
    vocab = build(sents, j)
    vocab_size.append(len(vocab))

####################################

"""
Plotting code: You can use this to plot if you want or write your own.
"""
fig = plt.figure(figsize=(5, 3))
plt.plot(t, vocab_size, '--ro')
plt.ylabel('|V|')
plt.xlabel('t')
plt.title('Vocabulary Size vs. t')
plt.tight_layout()
fig.savefig('./vocab_size_vs_t.png') # Include this figure in your submission.
```



The vocabulary size nonlinearly decreases as the threshold for rare words decreases

1.3

```python
# Question 1.3
oov = [] # list to store number of sentences with atleast 1 oov for each t.
t = list(range(1,16))

# #############################
# Your code goes here
#####################################
sents_with_oov = np.zeros((1,15))[0]
for j in t:
  vocab = build(sents, j)
  for sent in sents:
    for word in sent:
      if word not in vocab:
        sents_with_oov[j-1] = sents_with_oov[j-1] +1
        break

oov = np.ndarray.tolist(sents_with_oov)

"""
Plotting code: You can use this to plot if you want or write your own.
"""
fig = plt.figure(figsize=(5, 3))
plt.plot(t, oov, '--ro')
plt.ylabel('# sentences w/oov')
plt.xlabel('t')
plt.title('# of sentences w/oov vs. t')
plt.tight_layout()
fig.savefig('./oov_size_vs_t.png') # Include this figure in your submission.
```



Increasing t increases the number of sentences with oov words because, as seen in *1.2*, fewer words are included in the vocabulary when the frequency threshold is higher, so words are more likely to be oov.

1.4

Vocab: { “a”, “and”, “the”, “was”, “atmosphere”, “scene”, “restaurant”, “food”, “drink”, “bad”, “game”, “service”, “not”, “dull”, “tasteless”, “good”, “<oov>”}

Sentence: the food was not bad and the service was relatively good.

Bag of words representation [0, 1, 2, 2, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1]

Negative sentence with same representation: the food was not good and the service was relatively bad.

1.5

You could distinguish the two sentences’ bag of words representation by using an n-gram model. In other words, you could make the vocabulary contain clusters of word.

Better Vocab: {“a”, “and”, “the”, “was good”, “was bad”, “atmosphere”, “scene”, “restaurant”, “food”, “drink”, “game”, “service”, “not bad”, “not good,” “dull”, “was tasteless”, “not tasteless”, “<oov>”}

2.1

The success rates for Treatments A and B for Subtype 1 are 80% and 76.67%, respectively. The success rates for Treatments A and B for Subtype 2 are 66.67% and 60%, respectively. Based on this interpretation, Treatment A is more effective.

2.2

The success rates for Treatments A and B for the disease as a whole are 70% and 72.5%, respectively. Based on this interpretation, Treatment B is more effective. Despite the marginal association favoring treatment B, the partial association favors treatment A. This is likely because subtype 1 is more treatable and that since there are more data for treatment B on subtype 1, treatment B appears to be better overall.

2.3a

There need to be 14 more trials to guarantee treatment A is better than treatment B overall.
$(28 + 0.8x)/(40+x) = 0.725 \rightarrow x = 13.333$

3.1 Purchase Log, $D = \big((a_1, b_2), \ldots, (a_K, b_K)\big)$ user a_i purchased item b_i

$$Y_{ab} = \begin{cases} 1 & \text{if } (a,b) \in D \\ 0 & \text{if } (a,b) \notin D \end{cases}$$

$$J(X,Y) = \sum_{(a,b) \in D} \frac{(X_{ab} + b_{ab} - Y_{ab})^2}{2} + \frac{\lambda}{2} \sum_{(ab)} (X_{ab} + b_{ab})^2$$

$$\frac{\partial J(X,Y)}{\partial X_{ab}} = X_{ab} + \overset{\text{bias}}{b_{ab}} - Y_{ab} + \lambda(X_{ab} + b_{ab}) = X_{ab}(1+\lambda) + b_{ab} - Y_{ab} + \lambda b_{ab} \Rightarrow X_{ab} = \begin{cases} \frac{Y_{ab} - b_{ab}(1+\lambda)}{1+\lambda} & \text{if } (a,b) \in D \\ 0 & \text{if } (a,b) \notin D \end{cases}$$

3.2) $$J(X,Y) = \sum_{(ab) \in D} \frac{(U_a V_b + b_{uv} - Y_{ab})^2}{2} + \frac{\lambda}{2} \sum_a (U_a + b_u)^2 + \frac{\lambda}{2} \sum_b (V_b + b_v)^2$$

d is the rank of the factored matrices

d is a hyperparameter & it could be optimized using cross-validation.

3.3) $$\underset{d\times n}{U^T} \cdot \underset{n\times 1}{x_a} = \underset{d\times 1}{U^T x_a} \longrightarrow \left[\underset{1\times d}{W^T} \cdot \left(\underset{d\times 1}{U^T x_a} + \underset{d\times 1}{V^T x_b} \right) + b \right] = z \rightarrow y = \text{Sigmoid}(z)$$

$$\underset{d\times m}{V^T} \cdot \underset{m\times 1}{x_b} = \underset{d\times 1}{V^T x_b} \longrightarrow$$

$U^{n\times d}$ $V^{m\times d}$

X_a X_b

$n\times 1$ $m\times 1$

$Y \in R^1$

$U^T x_a \in R^{d\times 1}$

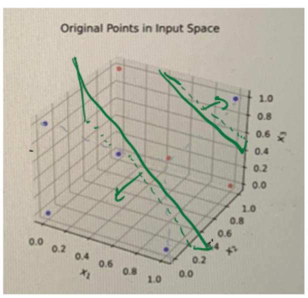
3.4 a) $\begin{bmatrix} 3 & 2 & \\ & 2.5 & 1 \\ 1.5 & & \end{bmatrix} = \begin{bmatrix} 1 \\ 1.25 \\ 0.5 \end{bmatrix} \begin{bmatrix} 3 & 2 & 0.8 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 0.8 \\ 3.75 & 2.5 & 1 \\ 1.5 & 1 & 0.4 \end{bmatrix}$

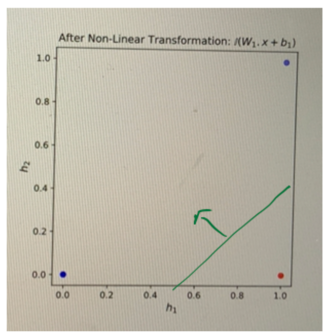
$$J(u,v) = \sum_{(ai) \in D} \frac{(u_a v_i - y_{ai})^2}{2} = 0$$

b) (a,i) wouldn't be in the set of (a,i) with outputs (D), so they could be anything, since loss isn't regularized here.

4.1) Left: $\phi(x_1, x_2) = x_1 + x_2$ Right: $\phi(x_1, x_2) = x_1^2 + x_2^2$

4.2 a)





$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = W_1 x + b_1$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{12} \end{bmatrix} = \begin{bmatrix} x_1 a_{11} + x_2 a_{12} + x_3 a_{13} + b_{11} \\ x_1 a_{21} + x_2 a_{22} + x_3 a_{23} + b_{12} \end{bmatrix} = z$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = x \cdot W_1 = (z - b_1) = \left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right)$$

$$\begin{bmatrix} x_1 a_{11} + x_2 a_{12} + x_3 a_{13} \\ x_1 a_{21} + x_2 a_{22} + x_3 a_{23} \end{bmatrix} = \begin{bmatrix} z_1 - b_{11} \\ z_2 - b_{12} \end{bmatrix} \left\} \begin{array}{l} x_1 a_{11} + x_2 a_{12} + x_3 a_{13} = z_1 - b_1 \quad 1. \\ x_1 a_{21} + x_2 a_{22} + x_3 a_{23} = z_2 - b_2 \quad 2. \end{array} \right.$$

$$\boxed{W_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad b_1 = \begin{bmatrix} -1.5 \\ -2.5 \end{bmatrix}}$$

4.2 b) $y = I\left([w_{21} \ w_{22}] \cdot \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + b_2 \right)$

$\boxed{b_2 = -0.5}$ since $h_1, h_2 = 0 \rightarrow w_2 \cdot h + b_2 = -0.5 = b_2$

$\boxed{W_2 = [1 \ -1]}$

$[w_{21} \ w_{22}] \begin{bmatrix} 0 \\ 1 \end{bmatrix} - 0.5 \Rightarrow$

$w_{22} \cdot 1 - 0.5 = 0.5$

$\Rightarrow w_{22} = 1$

$[w_{21} \ \ 1] \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 0.5 = -0.5$

$w_{22} = -1$

4.2 c) $\dfrac{d\ell}{dw_1} = \dfrac{\partial \ell}{\partial y} \cdot \dfrac{\partial y}{\partial z_y} \cdot \dfrac{\partial z_y}{\partial h} \cdot \dfrac{\partial h}{\partial z_h} \cdot \dfrac{\partial z_h}{\partial w_1}$

$$\boxed{\frac{\partial \ell}{\partial w_1} = (y - y^i) \cdot \left(\text{Sigmoid}(z_y)\left(1 - \text{Sigmoid}(z_y)\right)\right) \cdot W_2 \cdot \left(1 - \tanh^2(z_h)\right)\left(x^i\right)}$$

4.3) $a \rightarrow c$ corresponds to regularization decreasing. Regularization causes the elements of w to become favored towards zero. So (a) has the narrowest decision boundary & hence the most regularization.