Name: Kevin Zhang

Problem 1.

(a) The probability that $h(A) = h(B)$ is $J(A,B)$ or $\frac{|A \cap B|}{|A \cup B|}$. This is somewhat intuitive. The hash function is $h(A) = min_{x \in A} \pi(x)$ where π is a uniformly random permutation over the dictionary $|U| = n$. In order for $h(A) = h(B)$, the mininium π value must be the same. In other words, the word must be same across A and B. The number of shared words is $|A \cap B|$. The number of total possible attempts is $|A \cup B|$. Therefore, $Pr[h(A) = h(B)] = \frac{|A \cap B|}{|A \cup B|}$.

(b) We have k independent hash functions $h_1, h_2, ..., h_k$ and for documents A and B, we can store $h_1(A), ..., h_k(A)$ and $h_1(B), ..., h_2(B)$. We want to devise an algorithm to produce estimate Z from the stored hashes such that

$$\Pr[|Z - J(A,B)| \geq \epsilon] \leq 1/3$$

Let Z_i be an indicator variable whether $h_i(A) = h_i(B)$. Then, we have

$$\begin{aligned} \mathbb{E}[Z_i] &= Pr[h_i(A) = h_i(B)] \\ &= J(A,B) \\ \text{Var}(Z_i) &\leq 1 \textit{ this is generally true for indicator variable} \end{aligned}$$

With k hash functions, we can produce Z as the mean of these indicators. $Z = \frac{Z_1 + Z_2 + ... + Z_k}{k}$. We can express the expected value and variance of Z in terms of Z_i. Specifically, $\mathbb{E}[Z] = \mathbb{E}[Z_i]$, and $\text{Var}(Z) = \text{Var}(Z_i)/k$.

$$\begin{aligned} \mathbb{E}[Z] &= \mathbb{E}[Z_i] \\ &= J(A,B) \\ \text{Var}(Z) &= \text{Var}(Z_i)/k \\ &\leq 1/k \end{aligned}$$

We can then plug into Chebyshev's Inequality. Furthermore, we can use $k = 3\epsilon^2$.

$$\begin{aligned} \Pr[|Z - \mathbb{E}[Z]| \geq \epsilon] &\leq \frac{\sigma^2}{\epsilon^2} \\ \Pr[|Z - J(A,B)| \geq \epsilon] &\leq \frac{1}{3} \end{aligned}$$

Problem 2.

(a) The problem can be expressed as an LP:

$$\min \frac{1}{n}\sum_{i}^{n} z_i$$
$$y_i - a^T x_i - b \leq z_i \forall i$$
$$-y_i + a^T x_i + b \leq z_i \forall i$$
$$z_i \geq 0$$

(b) Code Below. The average error per test example was 0.509512.

```
import csv
import itertools
from cvxopt import matrix
from cvxopt.modeling import op, variable, sum, min, dot

x = []
y = []
with open('winequality-red.csv', newline='') as input:
    reader = csv.reader(input, delimiter=';')
    for row in itertools.islice(reader, 1, 1501):
        nums = list(map(float, row))
        x.append(nums[:-1])
        y.append(nums[-1])

# d is the dimension of a
d = len(x[0])
# n is the number of training examples
n = len(x)

# Variables
a = variable(11)
b = variable()
z = variable()

# Value Matricies
X = matrix(x)
Y = matrix(y)

# Constraints
c1 = (Y - dot(a,X) - b <= z)
c2 = (-Y + dot(a,X) + b <= z)
c3 = (-z <= 0)
```

```
# LP problem
lp = op(min(sum(z)), [c1, c2, c3])
lp.solve()

# test with the computed a and b
total_error = 0.0
count = 0
with open('winequality-red.csv', newline='') as input:
    reader = csv.reader(input, delimiter=';')
    for row in itertools.islice(reader, 1501, None):
        count += 1
        nums = list(map(float, row))
        x = nums[:-1]
        y = nums[-1]
        error = b.value[0]-y
        for i in range(d):
            error += x[i] * a.value[i]
        total_error += abs(error)

print("Average error per test example is %f" % (total_error/count))
```

Problem 3.

(a) Let $x_{i,j}$ indicate whether edge from $i \in X$ to $j \in Y$ is selected. Then, we can formulate the maximum set of edges such that no two edges share common endpoint as a LP:

$$\max \sum_{i \in X, j \in Y} x_{i,j}$$
$$\sum_{i \in X} x_{i,j} = 1 \forall j$$
$$\sum_{j \in Y} x_{i,j} = 1 \forall i$$
$$0 \leq x_{i,j} \leq 1$$

(b) The dual of the problem above can be expressed as an idea: The minimum number of edges that can be removed (if every vertex is connected) so that no vertex is shared. If we let $y_{i,j}$ indicate whether we remove an edge $i \in X$ to $j \in Y$, then we can express the dual LP below. The thought process is that the final number of connected edges to a vertex i or j must be ≤ 1.

$$\min \sum_{i \in X, j \in Y} y_{i,j}$$
$$\sum_{i \in X} 1 - \sum_{i \in X} y_{i,j} \leq 1 \forall j$$
$$\sum_{j \in Y} 1 - \sum_{i \in Y} y_{i,j} \leq 1 \forall i$$
$$0 \leq y_{i,j} \leq 1$$

(c) Left Blank For Now

Problem 4.

(a) We have a set of elements $V = \{e_1, e_2, ..., e_n\}$ and m subsets $S_1, S_2, ..., S_m$. Each set has weight $w_i \geq 0$. We want to find a collection of subsets that covers all of V and minimizes the total weight. If we let x_i indicate whether a subset S_i is selected, we can frame this as an LP:

$$\min \sum_{i=1}^{m} w_i * x_i$$
$$\sum_{e \in S_i} x_i \geq 1 \forall e \in V$$
$$0 \leq x_i \leq 1$$

(b) Suppose we have a fractional solution to the LP above. We can round the solution to produce an integral solution. This is accomplished as follows: the set S_i is picked with probability of $\min(1, 2x_i \ln n)$. We want to show that the probability that an element e_j is covered is at least $1 - \frac{1}{n^2}$.

We can start by examining the possibility that e_j is not selected at all:

$$\Pr[e_j \text{ not selected}] = \prod_{e_j \in S_i} (1 - \min(1, 2x_i \ln n))$$

The min is problematic, so we can use case analysis. In case 1, we have $2x_i \ln n \geq 1$. Then the expression reduces to:

$$\Pr[e_j \text{ not selected}] = \prod_{e_j \in S_i} (1 - 1) = 0$$

In case 2, we have $2x_i \ln n \leq 1$. Then, we can reduce the expression as follows. Note that $1 - x \leq e^{-x}$.

$$
\begin{aligned}
\Pr[e_j \text{ not selected}] &= \prod_{e_j \in S_i} (1 - 2x_i \ln n) \\
&\leq \prod_{e_j \in S_i} e^{-2x_i \ln n} \\
&\leq e^{-2\ln n \times \sum_{e_j \in S_i} x_i} \\
&\leq e^{-2\ln n} \\
&\leq \frac{1}{n^2}
\end{aligned}
$$

In both cases, the following statements hold:

$$
\begin{aligned}
\Pr[e_j \text{ is selected}] &= 1 - \Pr[e_j \text{ is not selected}] \\
&\geq 1 - \frac{1}{n^2}
\end{aligned}
$$

(c) We can generalize part B further. A feasible solution to the LP is one in which every e_j is selected. This is the same as 1 – Pr[there is a e not selected]. By the union bound, we can express the second half:

$$
\begin{aligned}
\Pr[\text{There is a } e_j \text{ is not selected}] &\leq \sum_{e_j \in V} \Pr[e_j \text{ is not selected}] \\
&\leq n(1 - \frac{1}{n^2}) \\
&\leq n - \frac{1}{n}
\end{aligned}
$$

Secondly, on the condition that the solution is feasible, we want to show that the expected cost is within O(ln$n$) times the objective value of the LP. We can use the same case analysis as earlier to break this down:

$$
\begin{aligned}
\text{Obj(LP)} &= \sum_{i=1}^{m} (w_i \times x_i) \\
\mathbb{E}[cost] &= \sum_{i=1}^{m} (w_i \times \Pr[S_i \text{ is selected}]) \\
&= \sum_{i=1}^{m} (w_i \times \min(1, 2x_i \ln n)) \\
\textbf{Case 1:} &\geq \sum_{i=1}^{m} (w_i \times 1) \\
\textbf{Case 2:} &\leq \sum_{i=1}^{m} (w_i \times 2x_i \ln n) \\
&\leq 2\ln n \times \sum_{i=1}^{m} (w_i \times x_i) \\
&\leq 2\ln n \times \text{Obj(LP)}
\end{aligned}
$$