CSE 250A. Assignment 1

Hao-en Sung (A53204772)

October 3, 2016

1 Assignment 1

1.1 Conditioning on background evidence [RN 13.16]

(a) Denoting such evidence by E, prove the conditionalized version of the product rule:

$$P(X, Y|E) = P(X|Y, E) \cdot P(Y|E)$$

Sol.

$$\begin{aligned} P(X, Y|E) &= P(X, Y|E) \cdot \frac{P(E)}{P(E)} = \frac{P(X, Y, E)}{P(E)} && (1) \\ &= \frac{P(X|Y, E) \cdot P(Y|E) \cdot P(E)}{P(E)} = P(X|Y, E) \cdot P(Y, E) && (2) \end{aligned}$$

□

(b) Also, prove the conditionalized version of Bayes rule:

$$P(X|Y, E) = \frac{P(Y|X, E) \cdot P(X|E)}{P(Y|E)}$$

Sol.

$$P(X|Y, E) = \frac{P(X, Y, E)}{P(Y|E) \cdot P(E)} = \frac{P(Y|X, E) \cdot P(X|E) \cdot P(E)}{P(Y|E) \cdot P(E)} = \frac{P(Y|X, E) \cdot P(X|E)}{P(Y|E)} \quad (3)$$

□

1.2 Conditional independence [RN 13.17]

Show that the following three statements about random variables X, Y , and E are equivalent:

$$\begin{array}{ll} (1) & P(Y|X, E) = P(Y|E) \\ (2) & P(X|Y, E) = P(X|E) \\ (3) & P(X, Y|E) = P(X|E)P(Y|E) \end{array}$$

Sol.

$$\begin{aligned} (1) \to (2): \quad & P(Y|X, E) = P(Y|E) && (1) \\ \Rightarrow \quad & \frac{P(X, Y, E)}{P(X|E) \cdot P(E)} = P(Y|E) && (2) \\ \Rightarrow \quad & \frac{P(X, Y, E)}{P(Y|E) \cdot P(E)} = P(X|E) && (3) \\ \Rightarrow \quad & P(X|Y, E) = P(X|E) && (4) \end{aligned}$$

$$\begin{aligned} (2) \to (3): \quad & P(X|Y, E) = P(X|E) && (5) \\ \Rightarrow \quad & \frac{P(X, Y, E)}{P(Y|E) \cdot P(E)} = P(X|E) && (6) \\ \Rightarrow \quad & \frac{P(X, Y, E)}{P(E)} = P(X|E) \cdot P(Y|E) && (7) \\ \Rightarrow \quad & P(X, Y|E) = P(X|E) \cdot P(Y|E) && (8) \end{aligned}$$

$$
\begin{aligned}
(3) \to (1): \quad & P(X,Y|E) = P(X|E) \cdot P(Y|E) && (9)\\
\Rightarrow \quad & \frac{P(X,Y,E)}{P(E) \cdot P(X|E)} = P(Y|E) && (10)\\
\Rightarrow \quad & \frac{P(X,Y,E)}{P(X,E)} = P(Y|E) && (11)\\
\Rightarrow \quad & P(Y|X,E) = P(Y|E) && (12)
\end{aligned}
$$

With (1) $\to$ (2) $\to$ (3) $\to$ (1), three statements are proved equivalent. □

1.3 Creative writing

Attach events to the binary random variables X, Y , and Z that are consistent with the following patterns of commonsense reasoning. You may use different events for the different parts of the problem.

(a) Cumulative evidence:

$$P(Z = 1) > P(Z = 1|X = 1) > P(Z = 1|X = 1, Y = 1)$$

Sol. Let $X = 1$ be the event *in winter*, $Y = 1$ be the event *at desert*, and $Z = 1$ be the event of *it rains*. It is easy to tell that

$$P(\textit{it rains}) > P(\textit{it rains} \mid \textit{in winter}) > P(\textit{it rains} \mid \textit{in winter, at desert}).$$

□

(b) Explaining away:

$$P(X = 1|Z = 1) > P(X = 1)$$
$$P(X = 1|Z = 1, Y = 1) < P(X = 1|Z = 1)$$

Sol. Let $X = 1$ be the event *it rains*, $Y = 1$ be the event *at desert*, and $Z = 1$ be the event *in summer*. It is again easy to tell that

$$P(\textit{it rains} \mid \textit{in summer}) > P(\textit{it rains})$$
$$P(\textit{it rains} \mid \textit{in summer, at desert}) < P(\textit{it rains} \mid \textit{in summer}).$$

□

(c) Conditional dependence:

$$P(X = 1, Y = 1) = P(X = 1) \cdot P(Y = 1)$$
$$P(X = 1, Y = 1|Z = 1) \neq P(X = 1|Z = 1) \cdot P(Y = 1|Z = 1)$$

Sol. Throw a fair dice for twice. Let $X = 1$ be the event *first time dice reveals one*, and $Y = 1$ be the event *second time dice reveals six*, and $Z = 1$ be the event *two dice throws sum to seven*. We know that $X \perp\!\!\!\perp Y$, and thus $P(X = 1, Y = 1) = P(X = 1) \cdot P(Y = 1)$. However, $P(X = 1, Y = 1|Z = 1) = \frac{1}{2}$, which is different from $P(X = 1|Z = 1) \cdot P(Y = 1|Z = 1) = \frac{1}{36}$. □

(d) Conditional independence:

$$P(Y = 1, Z = 1) \neq P(Y = 1) \cdot P(Z = 1)$$
$$P(Y = 1, Z = 1|X = 1) = P(Y = 1|X = 1) \cdot P(Z = 1|X = 1)$$

Sol. Assume there are two dices: one is fair, and one is biased. Throw one of them for twice. Let $X = 1$ be the event *throwing biased coin*, $Y = 1$ be the event *first time dice reveals one*, $Z = 1$ be the event *second time dice reveals six*. After first throw, one can get evidence on whether the dice is fair or biased, so $Y \not\perp\!\!\!\perp Z$. However, given X, it becomes an independent dice throw problem, which indicates that $Y|X \perp\!\!\!\perp Z|X$. □

1.4 Bayes Rule

Suppose that 1% of Olympic athletes use performance-enhancing drugs and that a particular drug test has a 5% false positive rate and a 2% false negative rate.

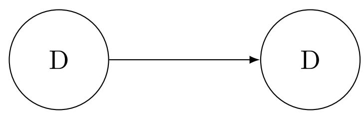
(a) Let $D \in \{0, 1\}$ indicate whether an athlete is doping, and let $T \in \{0, 1\}$ indicate the outcome of the drug test. Draw the belief network for these random variables, and use the information above to deduce the (conditional) probability tables for $P(D)$ and $P(T|D)$.

Sol. Some calculations about joint probabilities.

$$
\begin{aligned}
P(D=0) &= 1-0.01 = 0.99 && (1)\\
P(D=1) &= 0.01 && (2)\\
P(D=0,T=0) &= 0.99\times(1-0.05) = 0.9405 && (3)\\
P(D=0,T=1) &= 0.99\times 0.05 = 0.0495 && (4)\\
P(D=1,T=0) &= 0.01\times 0.02 = 0.0002 && (5)\\
P(D=1,T=1) &= 0.01\times(1-0.02) = 0.0098 && (6)
\end{aligned}
$$

Belief Networks for $P(D)$ and $P(T|D)$ then can be depicted as follows.

Doping	
D = 0	D = 1
0.99	0.01



	Tested	
Doping	T = 0	T = 1
D = 0	0.9405	0.0495
D = 1	0.0002	0.0098

Based on previous Belief Network, we can derive

$$
\begin{aligned}
P(T=0|D=0) &= \frac{P(T=0,D=0)}{P(D=0)} = \frac{0.9405}{0.99} = 0.95 && (7)\\
P(T=1|D=0) &= \frac{P(T=1,D=0)}{P(D=0)} = \frac{0.0495}{0.99} = 0.05 && (8)\\
P(T=0|D=1) &= \frac{P(T=0,D=1)}{P(D=1)} = \frac{0.0002}{0.01} = 0.02 && (9)\\
P(T=1|D=1) &= \frac{P(T=1,D=1)}{P(D=1)} = \frac{0.0098}{0.01} = 0.98. && (10)
\end{aligned}
$$

□

(b) Athlete A tests positive for drug use. What is the probability that Athlete A is not using drugs?

Sol.

$$
P(D=0|T=1) = \frac{P(D=0,T=1)}{P(D=0,T=1)+P(D=1,T=1)} = \frac{0.0495}{0.0495+0.0098} \approx 0.8347 \qquad (11)
$$

□

(c) Athlete B tests negative for drug use. What is the probability that Athlete B is using drugs?

Sol.

$$
P(D=1|T=0) = \frac{P(D=1,T=0)}{P(D=0,T=0)+P(D=1,T=0)} = \frac{0.0002}{0.9405+0.0002} \approx 0.0002 \qquad (12)
$$

□

1.5 Entropy

(a) Let X be a discrete random variable with $P(X=x_i)=p_i$ for $i\in 1,2,...,n$. The entropy $\mathcal{H}[X]$ of the random variable X is a measure of its uncertainty. It is defined as:

$$
\mathcal{H}[X] = -\sum_{i=1}^{n} p_i \log p_i,
$$

where log denotes the natural logarithm. Show that the entropy $\mathcal{H}[X]$ is maximized when $p_i=\frac{1}{n}$ for all i. You should do this by computing the gradient with respect to p_i and using Lagrange multipliers to enlet the constraint that $\sum_i p_i = 1$. Later in the course, we will use similar calculations for learning probabilistic models.

Sol. We want to solve

$$
\max_{p_1,p_2,\dots,p_n} \mathcal{H}[X] = -\sum_{i=1}^{n} p_i \log p_i, \text{ s.t. } \sum_{i=1}^{n} p_i = 1. \qquad (1)
$$

Firstly, we write down the Lagrange function as

$$
\mathcal{L}(p_1,p_2...,p_n,\lambda) = -\sum_{i=1}^{n} p_i \log p_i - \lambda\cdot(\sum_{i=1}^{n} p_i - 1). \qquad (2)
$$

Secondly, we let

$$\frac{\partial \mathcal{L}(p_1, p_2, ..., p_n, \lambda)}{\partial p_i} = -(\log p_i + p_i \cdot \frac{1}{p_i}) - \lambda \cdot 1 = -1 - \log p_i - \lambda = 0 \tag{3}$$

$$p_i = e^{-1-\lambda}. \tag{4}$$

Then, we rewrite Lagrange function as

$$\mathcal{L}(\lambda) = -n \cdot e^{-1-\lambda} \cdot (1-\lambda) - \lambda \cdot \left(n \cdot e^{-1-\lambda} - 1\right) = \lambda - n \cdot e^{-1-\lambda}. \tag{5}$$

When optimal occurs, we have

$$\frac{\partial \mathcal{L}(\lambda)}{\partial \lambda} = 1 - n \cdot (-1) \cdot e^{-1-\lambda} = 0 \tag{6}$$

$$\Rightarrow \quad e^{-1-\lambda} = \frac{1}{n} \tag{7}$$

$$\Rightarrow \quad p_i = \frac{1}{n} \tag{8}$$

□

(b) The joint entropy of n discrete random variables $(X_1, X_2, ..., X_n)$ is defined as:

$$\mathcal{H}(X_1, X_2, ..., X_n) = -\sum_{x_1}\sum_{x_2} ... \sum_{x_n} P(x_1, x_2, ..., x_n) \log P(x_1, x_2, ..., x_n),$$

where the sums range over all possible instantiations of $(X_1, X_2, ..., X_n)$. Show that if the variables X_i are independent, then their joint entropy is the sum of their individual entropies: namely,

$$P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i) \quad \text{implies} \quad \mathcal{H}(X_1, X_2, ..., X_n) = \sum_{i=1}^{n} \mathcal{H}(X_i).$$

Sol.

$$\mathcal{H}(X_1, X_2, ..., X_n) = -\sum_{x_1}\sum_{x_2} ... \sum_{x_n} \left[P(x_1, x_2, ..., x_n) \log \left(\prod_{i=1}^{n} P(x_i) \right) \right] \tag{9}$$

$$= -\sum_{x_1}\sum_{x_2} ... \sum_{x_n} \left[P(x_1, x_2, ..., x_n) \sum_{i=1}^{n} \log P(x_i) \right] \tag{10}$$

$$= -\sum_{x_1}\sum_{x_2} ... \sum_{x_n} \left[\sum_{i=1}^{n} [P(x_1, x_2, ..., x_n) \log P(x_i)] \right] \tag{11}$$

It is clear that we can reorder the summation formula in Eq. 11 and sum them through $\log P(x_i)$. Note that: x_i has been determined before the squared brackets in Eq. 12.

$$\mathcal{H}(X_1, X_2, ..., X_n) = -\sum_{i=1}^{n}\sum_{x_i} \left[\log P(x_i) \cdot \sum_{j=1, j\neq i}^{n} \sum_{x_j} P(x_1, x_2, ..., x_n) \right] \tag{12}$$

From Eq. 12, we can get $\mathcal{H}(X_i)$ by accumulating all other $x_j,\ j \in \{1, ..., n\},\ j \neq i$, that is

$$\mathcal{H}(X_1, X_2, ..., X_n) = -\sum_{i=1}^{n}\sum_{x_i} [\log P(x_i) \cdot P(x_i)] \tag{13}$$

$$= \sum_{i=1}^{n} \left[-\sum_{x_i} P(x_i) \log P(x_i) \right] \tag{14}$$

$$= \sum_{i=1}^{n} \mathcal{H}(X_i) \tag{15}$$

□

1.6 Kullback-Leibler distance

Often it is useful to measure the difference between two probability distributions over the same random variable. For example, as shorthand let

$$p_i = P(X = i|E), \quad q_i = P(X = i|E')$$

denote the conditional distributions over the random variable X for different pieces of evidence $E \neq E'$. Note that $\sum_i p_i = \sum_i q_i = 1$. The Kullback-Leibler (KL) distance between these distributions (also known as the relative entropy) is defined as:

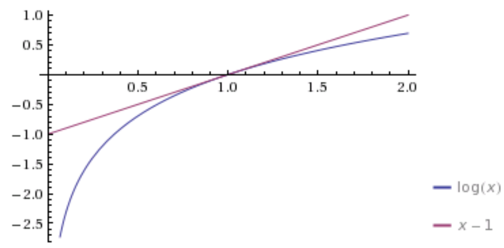
$$\text{KL}(p, q) = \sum_i p_i \log(\frac{p_i}{q_i}).$$

(a) Consider the natural logarithm (in base e). By sketching graphs of $\log(x)$ and $x-1$, verify the inequality:

$$\log(x) \leq x-1,$$

with equality if and only if $x=1$. Confirm this result by differentiation of $\log(x)-(x-1)$.

Sol. Graph sketched by Wolframe Alpha with query "$\log(x),(x-1),x=0$ to 2".



Define $f(x)=\log(x)-(x-1)$, we have

$$\frac{\partial f(x)}{\partial x} = \frac{1}{x}-1 \tag{1}$$

$$\frac{\partial^2 f(x)}{\partial^2 x} = -\frac{1}{x^2} \tag{2}$$

We can tell from $\frac{\partial^2 f(x)}{\partial^2 x}<0,\ \forall x>0$ and $\frac{\partial f(x)}{\partial x}=0$ if and only if $x=1$ that it is a convex function and has maximum value at 1. □

(b) Use the previous result to prove that $\mathrm{KL}(p,q)\geq 0$, with equality if and only if the two distributions p_i and q_i are equal.

Sol. With the conclusion from 1.6 (a), we can derive

$$\mathrm{KL}(p,q) = \sum_i p_i \log(\frac{p_i}{q_i}) = \sum_i -p_i \log(\frac{q_i}{p_i}) \tag{3}$$

$$\geq \sum_i -p_i\left(\frac{q_i}{p_i}-1\right) = -\sum_i q_i + \sum_i p_i = 0. \tag{4}$$

Also, we know that the equality happens if and only if $x=1$, that is

$$\frac{q_i}{p_i} = 1,\ \forall i \tag{5}$$

$$\Leftrightarrow \quad q_i = p_i,\ \forall i \tag{6}$$

□

(c) Using the inequality in (a), as well as the simple equality $\log x = 2\log\sqrt{x}$, derive the tighter lower bound:

$$\mathrm{KL}(p,q) \geq \sum_i \left(\sqrt{p_i}-\sqrt{q_i}\right)^2.$$

Sol. We can alter the conclusion in 1.6 (a) to

$$2\log\sqrt{x} \leq x-1. \tag{7}$$

By taking $x=\sqrt{\frac{q_i}{p_i}}$, we can derive

$$\mathrm{KL}(p,q) = \sum_i p_i\log(\frac{p_i}{q_i}) = \sum_i -p_i\log(\frac{q_i}{p_i}) = \sum_i -4\cdot p_i \log\left(\frac{q_i}{p_i}\right)^{\frac{1}{4}} \tag{8}$$

$$\geq \sum_i -2\cdot p_i\left(\sqrt{\frac{q_i}{p_i}}-1\right) = \sum_i -2\sqrt{q_i\cdot p_i} + 2\cdot\sum_i p_i \tag{9}$$

$$= \sum_i q_i - \sum_i -2\sqrt{q_i\cdot p_i} + \sum_i p_i = \sum_i (\sqrt{q_i}-\sqrt{p_i})^2 \tag{10}$$

□

(d) Provide a counterexample to show that the KL distance is not a symmetric function of its arguments:

$$\mathrm{KL}(p,q) \neq \mathrm{KL}(q,p).$$

Despite this asymmetry, it is still common to refer to $\mathrm{KL}(p,q)$ as a measure of distance. Many algorithms for machine learning are based on minimizing KL distances between probability distributions.

Sol. Assume both p and q are real-value vector of three dimensions. Let $p = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ and $q = [\frac{1}{12}, \frac{1}{12}, \frac{5}{6}]$, we can obtain

$$\mathrm{KL}(p, q) = \frac{1}{3} \cdot \log(\frac{\frac{1}{3}}{\frac{1}{12}}) + \frac{1}{3} \cdot \log(\frac{\frac{1}{3}}{\frac{1}{12}}) + \frac{1}{3} \cdot \log(\frac{\frac{1}{3}}{\frac{5}{6}}) \approx 0.618766 \tag{11}$$

$$\mathrm{KL}(p, q) = \frac{1}{12} \cdot \log(\frac{\frac{1}{12}}{\frac{1}{3}}) + \frac{1}{12} \cdot \log(\frac{\frac{1}{12}}{\frac{1}{3}}) + \frac{5}{6} \cdot \log(\frac{\frac{5}{6}}{\frac{1}{3}}) \approx 0.532527 \tag{12}$$

□

1.7 Mutual information

The mutual information $I(X, Y)$ between two discrete random variables X and Y is defined as

$$I(X, Y) = \sum_x \sum_y P(x, y) \log \left[\frac{P(x, y)}{P(x)P(y)} \right],$$

where the sum is over all possible values of the random variables X and Y. Note how the mutual information is related to the definitions in the previous two problems.

(a) Show that the mutual information $I(X, Y)$ is non negative. (The steps of the previous exercise should be useful here.)

Sol. Define $p_i = P(x, y)$ and $q_i = P(x)P(y)$, $\forall i \in (x, y)$ it can be observed that

- $0 \leq p_i, q_i, \leq 1$, because of probabilistic property
- $\sum_i p_i = 1$, because of joint marginal property of joint distribution
- $\sum_i q_i = 1$, because of distributive law from $\sum_x P(x) \cdot \sum_y P(y)$, which equals to 1

Thus, we can apply the conclusion from 1.6 (b) to prove $\sum_i p_i \log \left(\frac{p_i}{q_i} \right)$. □

(b) Show that the mutual information $I(X, Y)$ vanishes if and only if X and Y are independent random variables. (Thus, $I(X, Y)$ provides one quantitative measure of dependence between X and Y.)

Sol.

Case 1. $I(X, Y) = 0 \rightarrow X \perp\!\!\!\perp Y$

$I(X, Y) = 0$ indicates that $P(x, y) \log \left[\frac{P(x,y)}{P(x)P(y)} \right] = 0$, $\forall (x, y)$. Since $P(x, y)$ cannot be zero by definition, otherwise, it is not a "possible" pair of values of random variable X and Y, we get $P(x, y) = P(x)P(y)$, $\forall (x, y)$, which exactly meets the definition of $X \perp\!\!\!\perp Y$.

Case 2. $X \perp\!\!\!\perp Y \rightarrow I(X, Y) = 0$

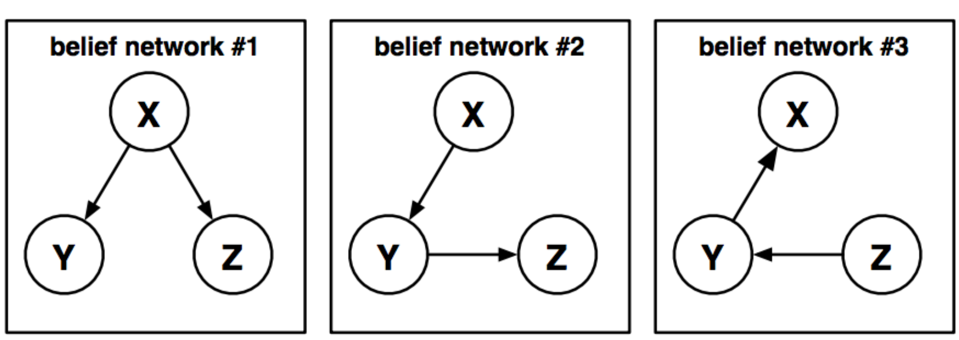
$$P(x, y) = P(x) \cdot P(y) \tag{1}$$

$$\Rightarrow \quad I(X, Y) = \sum_x \sum_y P(x, y) \cdot \log(1) = 0 \tag{2}$$

□

1.8 Compare and contrast

Consider the different belief networks (BNs) shown below for the discrete random variables X, Y, and Z.



(a) Does the second belief network imply a statement of marginal or conditional independence that is not implied by the first? If yes, provide an example.

Sol. For *belief network #2*, we can derive

$$
\begin{aligned}
& P(X,Y,Z) = P(X) \cdot P(Y|X) \cdot P(Z|Y) && (1)\\
\Rightarrow\ & P(X,Z|Y) = \frac{P(X)}{P(Y)} \cdot P(Y|X) \cdot P(Z|Y) && (2)\\
\Rightarrow\ & P(X,Z|Y) = P(X|Y) \cdot P(Z|Y), && (3)
\end{aligned}
$$

which means X and Z are conditional independent given Y. However, this kind of relationship does not apply in *belief network #1*. □

(b) Does the third belief network imply a statement of marginal or conditional independence that is not implied by the second? If yes, provide an example.

Sol. For *belief network #3*, we can derive

$$
\begin{aligned}
& P(X,Y,Z) = P(Z) \cdot P(Y|Z) \cdot P(X|Y) && (4)\\
\Rightarrow\ & P(X,Z|Y) = \frac{P(Z)}{P(Y)} \cdot P(Y|Z) \cdot P(X|Y) && (5)\\
\Rightarrow\ & P(X,Z|Y) = P(Z|Y) \cdot P(X|Y), && (6)
\end{aligned}
$$

which means X and Z are conditional independent given Y. According to the conclusion in (a), we know *belief network #2* and *belief network #3* have the same conditional independence property. □

(c) Does the first belief network imply a statement of marginal or conditional independence that is not implied by the third? If yes, provide an example.

Sol. For *belief network #1*, we can derive

$$
\begin{aligned}
& P(X,Y,Z) = P(X) \cdot P(Y|X) \cdot P(Z|X) && (7)\\
\Rightarrow\ & P(Y,Z|X) = P(Y|X) \cdot P(Z|X) && (8)\\
& && (9)
\end{aligned}
$$

which means Y and Z are conditional independent given X. However, this kind of relationship does not apply in *belief network #3*. □

1.9 Hangman

Description of coding problem is omitted.

(a) Download the file *hw1_word_counts_05.txt* that appears with the homework assignment. The file contains a list of 5-letter words (including names and proper nouns) and their counts from a large corpus of Wall Street Journal articles (roughly three million sentences). From the counts in this file compute the prior probability $P(w) = \text{COUNT}(w)/\text{COUNT}_{\text{total}}$. **As a sanity check, print out the fifteen most frequent 5-letter words, as well as the fourteen least frequent 5-letter words. Do your results make sense?**

Most frequent (15)	Less frequent (14)
THREE [273077]	BOSAK [6]
SEVEN [178842]	CAIXA [6]
EIGHT [165764]	MAPCO [6]
WOULD [159875]	OTTIS [6]
ABOUT [157448]	TROUP [6]
THEIR [145434]	CCAIR [7]
WHICH [142146]	CLEFT [7]
AFTER [110102]	FABRI [7]
FIRST [109957]	FOAMY [7]
FIFTY [106869]	NIAID [7]
OTHER [106052]	PAXON [7]
FORTY [94951]	SERNA [7]
YEARS [88900]	TOCOR [7]
THERE [86502]	YALOM [7]
SIXTY [73086]	

Table 1: Partial Word Frequence

Sol. □

(b) Consider the following stages of the game. For each of the following, indicate the best next guess — namely, the letter l that is most likely (probable) to be among the missing letters. Also report the probability $P(L_i = l \text{ for some } i \in \{1,2,3,4,5\}|E)$ for your guess l. Your answers should fill in the last two columns of this table. (Some answers are shown so that you can check your work.)

Sol. Table are filled up as follows.

correctly guessed	incorrectly guessed	best next guess	$P(L_i = l \text{ for some } i \in \{1,2,3,4,5\} \vert E)$
− − − − −	{}	E	0.5394
− − − − −	{A,I}	E	0.6214
A − − − R	{}	T	0.9816
A − − − R	{E}	O	0.9913
− − U − −	{O,D,L,C}	T	0.7045
− − − − −	{E,O}	I	0.6366
D − − − I −	{}	A	0.8207
D − − − I −	{A}	E	0.7521
− U − − −	{A,E,I,O,S}	Y	0.6270

Table 2: Partial Game Table

□

(c) Turn in a **hard-copy printout** of your source code. Do not forget the source code: it is worth many points on this assignment.

Sol. Runnable code appended as follows.

```
#include <cstdio>
#include <cstdlib>
#include <vector>
#include <utility>
#include <string>
#include <algorithm>
#include <unordered_set>

using namespace std;

const int LEN = 5;
const int MAXLEN = 30;

int main() {
    // [Read Data]
    string path = "../dat/hw1_word_counts_" +
      to_string(LEN/10) + to_string(LEN%10) + ".txt";
    FILE* pfile = fopen(path.c_str(), "r");
    if (pfile == NULL) {
        fprintf(stderr, "Read file failed\n");
        exit(EXIT_FAILURE);
    }

    char word[MAXLEN];
    int cnt;
    vector<pair<int, string>> vct;
    while (fscanf(pfile, "%s%d", word, &cnt) != EOF) {
        vct.emplace_back(cnt, word);
    }

    fclose(pfile);



    int size = vct.size();
    {   // [Problem (a)]
        sort(vct.begin(), vct.end());
        fprintf(stdout, "Most frequent fifteen words\n");
        for (int i = 0; i < 15; i++) {
            fprintf(stdout, "%s\n", vct[size-1-i].second.c_str());
        }
        fprintf(stdout, "\n");
        fprintf(stdout, "Less frequent forteen words\n");
        for (int i = 0; i < 14; i++) {
            fprintf(stdout, "%s\n", vct[i].second.c_str());
        }
        fprintf(stdout, "\n");
    }



    {   // [Problem (b)]
        // [Input]
        //   * Guess string
        //     * capital characters only
        //     * empty character with - sign
        //   * Charater string: {A,...,Z}
        //     * no additional space is allowed
        char str[MAXLEN];
        char chr[MAXLEN];
        while (scanf("%s%s", str, chr) != EOF) {
            unordered_set<int> uset;
```

```
            int nchr = strlen(chr);
            for (int i = 0; i < LEN; i++) {
                uset.emplace(str[i]-'A');
            }
            for (int i = 1; i < nchr-1; i+=2) {
                uset.emplace(chr[i]-'A');
            }

            vector<pair<double,int>> res;
            for (int k = 0; k < 26; k++) {
                if (uset.find(k) != uset.end()) {
                    continue; // queried, just skip
                }
                int num = 0;
                int div = 0;
                for (int i = 0; i < size; i++) {
                    bool suc = true;
                    int match = 0;
                    for (int j = 0; j < LEN; j++) {
                        if (str[j] == '-') {
                            if (uset.find(vct[i].second[j]-'A')
                                != uset.end()) {
                                suc = false;
                            }
                            if (vct[i].second[j]-'A' == k) {
                                match += 1;
                            }
                        } else {
                            if (str[j] != vct[i].second[j]) {
                                suc = false;
                            }
                        }
                    }
                    if (suc == true) {
                        div += vct[i].first;
                        if (match > 0) {
                            num += vct[i].first;
                        }
                    }
                }
                res.emplace_back(1.0 * num / div, k);
            }

            int res_size = res.size();
            sort(res.begin(), res.end());
            fprintf(stdout, "Best: %c\n", res[res_size-1].second + 'A');
            fprintf(stdout, "Prob: %.4f\n\n",res[res_size-1].first);
        }
    }
}
```

□