CSE 250A. Assignment 4

Hao-en Sung (A53204772)
wrangle1005@gmail.com

October 23, 2016

4.1 Maximum likelihood estimation of a multinomial distribution

(a) Log-likelihood

Sol.

$$
\begin{aligned}
L = \log P(\text{data}) &= \log \left(\prod_{t=1}^{T} P(X_t) \right) && (1)\\
&= \sum_{n=1}^{2N} \sum_{t=1}^{C_n} \log P(X_i = n) && (2)\\
&= \sum_{n=1}^{2N} C_n \log p_n && (3)
\end{aligned}
$$

□

(b) Maximum likelihood estimate

Sol. After apply the Lagrange framework, I have

$$
\begin{aligned}
L :&= C_n \log p_n - \lambda \left(\sum_{n=1}^{2N} p_n - 1 \right), && (4)\\
\frac{\partial L}{\partial p_n} &= \frac{C_n}{p_n} - \lambda. && (5)
\end{aligned}
$$

At optimal, it can be derived that

$$
p_n = \frac{C_n}{\lambda}. \tag{6}
$$

Based on the original constraint, it is said

$$
\begin{aligned}
\sum_{n=1}^{2N} p_n &= \frac{\sum_{n=1}^{2N} C_n}{\lambda} = 1 && (7)\\
\lambda &= \sum_{n=1}^{2N} C_n. && (8)
\end{aligned}
$$

Replace λ with $\sum_{n=1}^{2N} C_n$ in Eq. 6, then it is clear that $p_n = \frac{C_n}{\sum_{n=1}^{2N} C_n}$. Since C_n is nonnegative, p_n is also nonnegative. □

(c) Even versus odd

Sol.

$$
\begin{aligned}
&\sum_{n=1}^{2N} (-1)^n p_n = 0\\
\Rightarrow \quad & -\sum_{n=1}^{N} p_{2n-1} + \sum_{n=1}^{N} p_{2n} = 0\\
\Rightarrow \quad & \sum_{n=1}^{N} p_{2n-1} = \sum_{n=1}^{N} p_{2n}
\end{aligned}
$$

□

(d) Maximum likelihood estimate

Sol. From (c) I know that $\sum_{n=1}^{N} p_{2n-1} = \sum_{n=1}^{N} p_{2n} = \frac{1}{2}$. Beside that, I can rewrite Eq. 3 in (a) to $\sum_{n=1}^{N} C_{2n-1} \log p_{2n-1} + \sum_{n=1}^{N} C_{2n} \log p_{2n}$. Since odd die and even die are independently constrained, I can downscale origin problem to two sub-problems. Thus, I need to solve

$$\max_{p_{2n-1}} \sum_{n=1}^{N} C_{2n-1}, \quad \text{s.t.} \sum_{n=1}^{N} p_{2n-1} = \frac{1}{2},$$
$$\max_{p_{2n}} \sum_{n=1}^{N} C_{2n}, \quad \text{s.t.} \sum_{n=1}^{N} p_{2n} = \frac{1}{2}.$$

Follow similar step from Eq. 4 to 8, I can get

$$p_{2n-1} = \frac{C_{2n-1}}{\sum_{n=1}^{N} C_{2n-1}}$$
$$p_{2n} = \frac{C_{2n}}{\sum_{n=1}^{N} C_{2n}}.$$

□

4.2 Maximum likelihood estimation in belief networks

(a) Express the maximum likelihood estimates for the CPTs in G_1 in terms of these counts.

Sol. From figure for G_1, I know CPT tables are

$$P_1(X_1 = x_1) = \frac{P_1(X_1 = x_1)}{P_1(X_1)}$$
$$P_{i+1}(X_{i+1} = x_{i+1}|X_i = x_i) = \frac{P_i(X_i = x_i, X_{i+1} = x_{i+1})}{P_i(X_i = x_i)}, \ \forall i \in \{1, ..., n-1\}.$$

Take advantage of the conclusion from Section 4.1, I am aware that the probability p_n maximize log-likelihood when it equals to $\frac{C_n}{\sum_{n=1}^{2N}}$.
Similarly, $P(G_1)$ is maximized as $P^*(G_1)$ when

$$P_1(X_1 = x_1) = \frac{\text{COUNT}_1(X_1 = x_1)}{\text{COUNT}_1(X_1)} = \frac{\text{COUNT}_1(X_1 = x_1)}{T}$$
$$P_{i+1}(X_{i+1} = x_{i+1}|X_i = x_i) = \frac{\text{COUNT}_i(X_i = x_i, X_{i+1} = x_{i+1})}{\text{COUNT}_i(X_i = x_i)}, \ \forall i \in \{1, ..., n-1\}..$$

□

(b) Express the maximum likelihood estimates for the CPTs in G_2 in terms of these counts.

Sol. Similar to (a), I know CPT tables are

$$P_n(X_n = x_n) = \frac{P_n(X_n = x_n)}{P_n(X_n)}$$
$$P_i(X_i = x_i|X_{i+1} = x_{i+1}) = \frac{P_i(X_i = x_i, X_{i+1} = x_{i+1})}{P_{i+1}(X_{i+1} = x_{i+1})}, \ \forall i \in \{1, ..., n-1\}.$$

When $P(G_2)$ is maximized as $P^*(G_2)$,

$$P_n(X_n) = \frac{\text{COUNT}_n(X_n = x_n)}{\text{COUNT}_n(X_n)} = \frac{\text{COUNT}_n(X_n = x_n)}{T}$$
$$P_i(X_i = x_i|X_{i+1} = x_{i+1}) = \frac{\text{COUNT}_i(X_i = x_i, X_{i+1} = x_{i+1})}{\text{COUNT}_{i+1}(X_{i+1} = x_{i+1})}, \ \forall i \in \{1, ..., n-1\}.$$

□

(c) Using your answers from parts (a) and (b), show that the maximum likelihood CPTs for G_1 and G_2 from this data set give rise to the same joint distribution over the nodes $\{X_1, X_2, ..., X_n\}$.

Sol. It can be derived that

$$
\begin{aligned}
P(G_1 = g_1) &= P_1(X_1 = x_1) \cdot \prod_{i=1}^{n-1} P_{i+1}(X_{i+1} = x_{i+1} | X_i = x_i) \\
&= \frac{\text{COUNT}_1(X_1 = x_1)}{T} \cdot \prod_{i=1}^{n-1} \frac{\text{COUNT}_i(X_i = x_i, X_{i+1} = x_{i+1})}{\text{COUNT}_i(X_i = x_i)} \\
&= \frac{\prod_{i=1}^{n-1} \text{COUNT}_i(X_i = x_i, X_{i+1} = x_{i+1})}{T \cdot \prod_{i=2}^{n-1} \text{COUNT}_i(X_i = x_i)} \\
&= \frac{\prod_{i=1}^{n-1} \text{COUNT}_i(X_i = x_i, X_{i+1} = x_{i+1})}{T \cdot \prod_{i=1}^{n-2} \text{COUNT}_{i+1}(X_{i+1} = x_{i+1}) \cdot \text{COUNT}_n(X_n = x_n)} \cdot \text{COUNT}_n(X_n = x_n) \\
&= \prod_{i=1}^{n-1} \frac{\text{COUNT}_i(X_i = x_i, X_{i+1} = x_{i+1})}{\text{COUNT}_{i+1}(X_{i+1} = x_{i+1})} \cdot \frac{\text{COUNT}_n(X_n)}{T} \\
&= \prod_{i=1}^{n-1} \frac{P_i(X_i = x_i, X_{i+1} = x_{i+1})}{P_{i+1}(X_{i+1} = x_{i+1})} \cdot P_n(X_n = x_n) = P(G_2 = g_2).
\end{aligned}
$$

□

(d) Suppose that some but not all of the edges in these DAGs were reversed, as in the graph G_3 shown below. Would the maximum likelihood CPTs for G_3 also give rise to the same joint distribution? (Hint: does G_3 imply all the same statements of conditional independence as G_1 and G_2?)

Sol. One can observe that in (c) there is a T in denominator, which is caused by the one root node — either X_1 in G_1 or X_n in G_2. However, in G_3 there are two root nodes X_2 and X_{n-1}, which generates two T in denominator. It is clear that G_3 will not have the same joint probability as G_1 or G_2. □

4.3 Statistical language modeling

(a) Compute the maximum likelihood estimate of the unigram distribution $P_u(w)$ over words w. Printout a table of all the tokens (i.e., words) that start with the letter "A", along with their numerical unigram probabilities (not counts). (You do not need to print out the unigram *probabilities* for all 500 tokens.)

Sol. Unigram probability for words starting with character "A" is shown as Table 1.

□

(b) Compute the maximum likelihood estimate of the bigram distribution $P_b(w'|w)$. Print out a table of the five most likely words to follow the word "THE", along with their numerical bigram probabilities.

Sol. Bigram probability for words starting with word "THE" is shown as Table 2.

□

(c) Consider the sentence "Last week the stock market fell by one hundred points." Ignoring punctuation, compute and compare the log-likelihoods of this sentence under the unigram and bigram models. In the equation for the bigram log-likelihood, the token $\langle s \rangle$ is used to mark the beginning of a sentence. Which model yields the highest log-likelihood?

Sol.

$$
\begin{aligned}
L_u &= -64.509440 \\
L_b &= -44.740469
\end{aligned}
$$

It is clear that the bigram model yields higher probability. □

(d) Consider the sentence "Last week the stock market fell by one hundred points." Ignoring punctuation, compute and compare the log-likelihoods of this sentence under the unigram and bigram models. Which pairs of adjacent words in this sentence are not observed in the training corpus? What effect does this have on the log-likelihood from the bigram model?

Sol.

$$
\begin{aligned}
L_u &= -41.643460 \\
L_b &= -\infty
\end{aligned}
$$

The pairs (NINETEEN, OFFICIALS) and (SOLD, FIRE) do not show up in corpus, which makes log(0) and pushes the overall log probability to $-\infty$. □

A	0.018407
AND	0.017863
AT	0.004313
AS	0.003992
AN	0.002999
ARE	0.002990
ABOUT	0.001926
AFTER	0.001347
ALSO	0.001310
ALL	0.001182
A.	0.001026
ANY	0.000632
AMERICAN	0.000612
AGAINST	0.000596
ANOTHER	0.000428
AMONG	0.000374
AGO	0.000357
ACCORDING	0.000348
AIR	0.000311
ADMINISTRATION	0.000292
AGENCY	0.000280
AROUND	0.000277
AGREEMENT	0.000263
AVERAGE	0.000259
ASKED	0.000258
ALREADY	0.000249
AREA	0.000231
ANALYSTS	0.000226
ANNOUNCED	0.000227
ADDED	0.000221
ALTHOUGH	0.000214
AGREED	0.000212
APRIL	0.000207
AWAY	0.000202

Table 1: Unigram Probability for words start with character "A"

<UNK>	0.615020
U.	0.013372
FIRST	0.011720
COMPANY	0.011659
NEW	0.009451

Table 2: Bigram Probability for words start with word "THE"

(e) Consider the so-called mixture model that predicts words from a weighted interpolation of the unigram and bigram models. Compute and plot the value of this log-likelihood L_m as a function of the parameter $\lambda \in [0, 1]$. From your results, deduce the optimal value of λ to two significant digits.

Sol. The maximum probability of mixed unigram and bigram algorithm is -39.953680, which appears at $\lambda = 0.41$. I also examined different λ with 0.01 precision, as shown in Fig. 1.

□

(f) Submit a hard copy of your source code for the previous parts of this problem.

I first use C++ to write down all main functions to solve (a) to (d). For (e), I use C++ generate probabilities for mixed unigram and bigram model with various λ, then use MATLAB to render the figure. Both C++ and MATLAB are included in Appendix as 1 and 2, respectively.

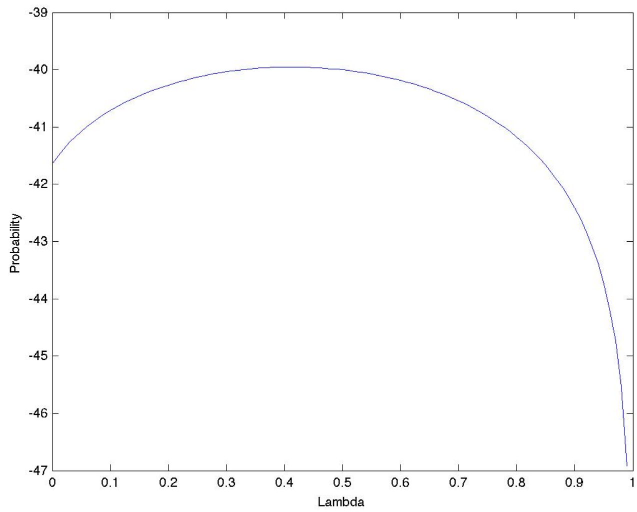

Figure 1: Mixed unigram and bigram model with different lambdas

Appendix

```
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include <cfloat>
#include <vector>
#include <utility>
#include <string>
#include <unordered_map>
#include <algorithm>
#include <assert.h>

using namespace std;

const int MAXL = 10000;
const int NUMV = 500;
const int NUMS = 100;

char in[MAXL + 10];

unordered_map<string,int> umsi;
unordered_map<int,string> umis;

vector<string> vocab(NUMV, "");
vector<int> uni(NUMV, 0);
vector<vector<int>> bi(NUMV, vector<int>(NUMV, 0));

// helper function to calculate prob with different lambda
double calLProb(vector<string>& lst, double lambda, bool warning) {
    int size = lst.size();

    int usum = 0;
    for (int i = 0; i < NUMV; i++) {
        usum += uni[i];
    }

    double lprob = 0;
    for (int i = 0; i < size; i++) {
        int tar = i == 0 ? umsi.at("<s>") : umsi.at(lst[i-1]);
        int bsum = 0;
        for (int j = 0; j < NUMV; j++) {
            bsum += bi[tar][j];
        }

        if (bi[tar][umsi.at(lst[i])] == 0 and lambda != 0 and warning) {
            fprintf(stderr, "Warning: (%s, %s) does not appear in documents\n",
              umis.at(tar).c_str(), lst[i].c_str());
        }

        lprob += log((1-lambda) * uni[umsi.at(lst[i])] / usum +
          lambda * bi[tar][umsi.at(lst[i])] / bsum);
    }

    return lprob;
}

int main() {
    { // read vocabulary
        string FN = "../dat/vocab.txt";
        FILE *pf = fopen(FN.c_str(), "r");
        if (pf == NULL) {
            fprintf(stderr, "Vocab file cannot read\n");
```

```
            exit(EXIT_FAILURE);
        }

        int cnt = 0;
        while (fscanf(pf, "%s", in) != EOF) {
            umsi[in] = cnt;
            umis[cnt] = in;
            vocab[cnt++] = in;
        }

        fclose(pf);
        assert(vocab.size() == NUMV);
    }

    { // read unigram
        string FN = "../dat/unigram.txt";
        FILE *pf = fopen(FN.c_str(), "r");
        if (pf == NULL) {
            fprintf(stderr, "Unigram file cannot read\n");
            exit(EXIT_FAILURE);
        }

        int d;
        int cnt = 0;
        while (fscanf(pf, "%d", &d) != EOF) {
            uni[cnt++] = d;
        }

        fclose(pf);
        assert(cnt == NUMV);
    }

    { // read bigram
        string FN = "../dat/bigram.txt";
        FILE *pf = fopen(FN.c_str(), "r");
        if (pf == NULL) {
            fprintf(stderr, "Bigram file cannot read\n");
            exit(EXIT_FAILURE);
        }

        int u, v, d;
        while (fscanf(pf, "%d\t%d\t%d", &u, &v, &d) != EOF) {
            u -= 1;
            v -= 1;
            assert(u >= 0 and u < NUMV and v >= 0 and v < NUMV);
            bi[u][v] = d;
        }

        fclose(pf);
    }

    { // 4.3 (a)
        int usum = 0;
        for (int i = 0; i < NUMV; i++) {
            usum += uni[i];
        }
        printf("4.3 (a)\n");
        for (int i = 0; i < NUMV; i++) {
            if (vocab[i][0] == 'A') {
                printf("%s %f\n", vocab[i].c_str(), 1.0 * uni[i]/usum);
            }
        }
        printf("\n\n");
    }

    { // 4.3 (b)
        int tar = umsi.at("THE");
        int sum = 0;
        vector<pair<int,int>> vct;
        for (int j = 0; j < NUMV; j++) {
            sum += bi[tar][j];
            vct.emplace_back(bi[tar][j],j);
        }
        sort(vct.rbegin(), vct.rend());
        printf("4.3 (b)\n");
        for (int i = 0; i < 5; i++) {
            printf("%s %f\n", vocab[vct[i].second].c_str(), 1.0 * vct[i].first/sum);
        }
        printf("\n\n");
    }

    { // 4.3 (c)
        vector<string> lst{"LAST", "WEEK", "THE", "STOCK", "MARKET", "FELL", "BY", "ONE", "HUNDRED", "POINTS"};
        printf("4.3 (c)\n");
        { // I. unigram
            double lprob = calLProb(lst, 0, true);
            printf("unigram: %f\n", lprob);
        }
        { // II. bigram
            double lprob = calLProb(lst, 1, true);
```

```
                printf("bigram: %f\n", lprob);
            }
            printf("\n\n");
        }

        { // 4.3 (d)
            vector<string> lst{"THE", "NINETEEN", "OFFICIALS", "SOLD", "FIRE", "INSURANCE"};
            printf("4.3 (d)\n");
            { // I. unigram
                double lprob = calLProb(lst, 0, true);
                printf("unigram: %f\n", lprob);
            }
            { // II. bigram
                double lprob = calLProb(lst, 1, true);
                printf("bigram: %f\n", lprob);
            }
            printf("\n\n");
        }

        { // 4.3 (e)
            string FN = "../res/lambda_data.csv";
            FILE* pf = fopen(FN.c_str(), "w");
            if (pf == NULL) {
                fprintf(stderr, "Lambda file cannot write\n");
                exit(EXIT_FAILURE);
            }

            vector<string> lst{"THE", "NINETEEN", "OFFICIALS", "SOLD", "FIRE", "INSURANCE"};
            double mmax = -DBL_MAX;
            double pmax = -1;
            for (int i = 0; i < NUMS; i++) {
                double pos = 1.0 * i / NUMS;
                double lprob = calLProb(lst, pos, false);
                if (lprob > mmax) {
                    mmax = lprob;
                    pmax = pos;
                }
                fprintf(pf, "%f, %f\n", pos, lprob);
            }

            printf("4.3 (d)\n");
            printf("max prob: %f (at %.2f)", mmax, pmax);
            fclose(pf);
        }
}
```

Listing 1: Main Code

```
% read in data
M = csvread('../res/lambda_data.csv');

% plot figure
res = figure('visible', 'off');
plot(M(:,1), M(:,2));
saveas(res, '../res/lambda.jpg');
```

Listing 2: Draw Lambda Plot