Rutgers CS323 (04), Spring 2017, Homework 2

Due at 8:00am on Feb 27, 2017, submitted via Sakai

Root-Finding for Estimating 2×2 Contingency Tables

You are asked to implement the following three functions:

```
1. function N11_est = Est_IPS(n11,n12,n21,n22,M1,M2,N,iter)
2. function N11_est = Est_MLE(n11,n12,n21,n22,M1,M2,N,iter)
3. function TestMultinomialSampling(Word1, Word2)
```

You will need to test your code on estimating the intersections for two pairs of words: (1) HONG - KONG (2) UNITED - STATES.

1 Iterative Proportional Scaling (IPS) and Newton's Method for MLE

Here, we provide the following matlab script for you to test your IPS and MLE (cubic equation, using Newton's method) functions so that everyone who implemented IPS and MLE correctly will be able to submit results like Figure 1.

```
function TestIPSandMLE

M1 = 300; M2=500; N = 2000;
n11 = round(rand*20+10);   n12 = round(rand*40+20);
n21 = round(rand*80+40);   n22 = round(rand*150+75);

iter = 20;
IPS = Est_IPS(n11,n12,n21,n22,M1,M2,N,iter);
MLE = Est_MLE(n11,n12,n21,n22,M1,M2,N,iter);

figure;
semilogy(2:iter, abs(IPS(2:end)-IPS(1:end-1)),'r-o','linewidth',2);
hold on; grid on;
semilogy(2:iter, abs(MLE(2:end)-MLE(1:end-1)),'b-d','linewidth',2);
title(num2str([n11 n12 n21 n22]));
set(gca,'FontSize',20); xlabel('Iteration'); ylabel('Convergence error');
legend('IPS','MLE');
```

Note that the input data are random (with some restrictions). The return from the function is a vector which stores the estimates for all the iterations (in this example 20 iterations). For IPS, one iteration includes one scaling of all the rows followed by one scaling of all the columns. For MLE, you should start with the "margin-free (MF)" estimate as the first iteration. Two results for two runs are provided in the following figure.

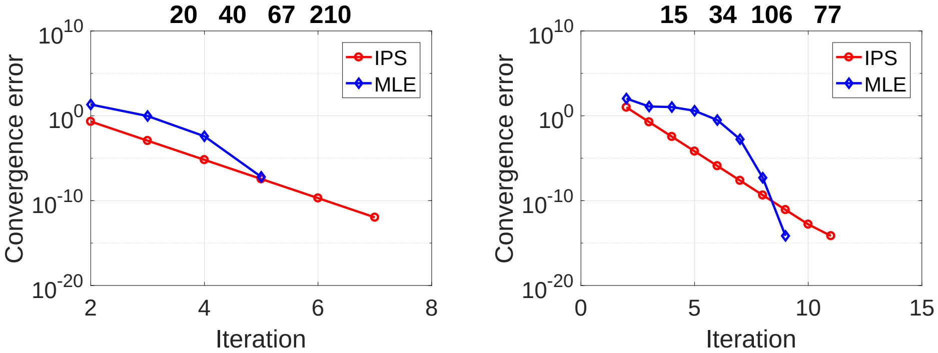

Figure 1: Two runs of the IPS and MLE code. The margins are fixed to be $M_1 = 300$, $M_2 = 500$ and $N = 2000$. The sample contingency tables are random. The title prints the table. For example, the left panel used $n_{11} = 20$, $n_{12} = 40$, $n_{21} = 67$, $n_{22} = 210$.

2 Multinomial Sampling and Estimation

For this problem, you need to complete the following matlab function:

```
function TestMultinomialSampling(Word1, Word2)

N = 2^16; W1 = zeros(N,1); W2 = zeros(N,1);
data = feval('load',[ Word1]);   W1(data(:,2))=1;
data = feval('load',[ Word2]);   W2(data(:,2))=1;
K = [10 20 30 50 80 100 150 200 300 400 500 600 800 1000];
for m = 1:10^3
    ind = randsample(N,max(K));
    S1 = W1(ind); S2 = W2(ind);
    for i = 1:length(K)
        s1 = S1(1:K(i));
        s2 = S2(1:K(i));
        n11 = sum(s1==1 & s2==1);
        %
        n11 = n11+0.1;n12 = n12+0.1;
        n21 = n21+0.1;n22 = n22+0.1;
    n = n11+n12+n21+n22;
    M1 = sum(W1); M2 = sum(W2);
    ips = Est_IPS(n11,n12,n21,n22,M1,M2,N,20);
    IPS(m,i) = ips(end);
    %MF(m,i) =
    %
   end
end

N11 = sum(W1.*W2);
IPS_mse = mean( (IPS - N11).^2);
%
figure;
loglog(K, IPS_mse,'r-o','linewidth',2); hold on; grid on;
%
set(gca,'FontSize',20,'YMinorGrid','off');
xlabel('Sample size');
ylabel('MSE');
text(20,2*10^4,[Word1 '--' Word2],'Color','r','FontWeight','Bold','FontSize',20);
% make sure to include a legend
```

This way, the data (binary, 0/1) are written two vectors W_1 and W_2 with length $N = 2^{16}$. The steps are:

1. Randomly select n coordinates from $[1, 2, ..., N]$. For this assignment, we use "sample without replacement."

2. Finding the sample contingency table from the n samples, using our convention: $n_{11} = \#\{W_1 = 1 \text{ and } W_2 = 1\}$. $n_{12} = \#\{W_1 = 1 \text{ and } W_2 = 0\}$, $n_{21} = \#\{W_1 = 0 \text{ and } W_2 = 1\}$, and $n_{22} = \#\{W_1 = 0 \text{ and } W_2 = 0\}$.

3. Estimate the original table, in particular N_{11} by three methods. Method 1: IPS (iter=20). Method 2: (Approximate) MLE; Method 3: Margin-Free (MF). You can convince yourself that directly using the Newton's method will likely lead to very miserable estimates. Thus, we recommend an approximate formula (which the instructor and his intern mentor derived in 2005)

$$\hat{N}_{11} = \frac{M1(2n_{11} + n_{21}) + M2(2n_{11} + n_{12}) - \sqrt{(M1(2n_{11} + n_{21}) - M2(2n_{11} + n_{12}))^2 + 4M1M2n_{12}n_{21}}}{2(2n_{11} + n_{12} + n_{21})}$$

4. Repeat the procedure by varying sizes n. For this assignment, you will need to use all $n \in \{10, 20, 30, 50, 80, 100, 150, 200, 300, 400, 500, 600, 800, 1000\}$. Note that there is a small trick for efficiency. You just need to sample the maximum n (in this case 1000), then re-use this set of samples for other n values. For example, when $n = 10$, you just need to use the first 10 samples.

5. Repeat this for 10^4 times so that we can compute the bias and mean square errors (MSE). Basically, after running this experiment 10^4 times, you will have three matrices of size $10^4 \times 14$ (for 14 different n values), from which we can compute the empirical bias and empirical MSE. Suppose the true value is V. Given m estimates $\hat{V}_i$, $i = 1$ to m. The empirical bias is $\frac{1}{m}\sum_{i=1}^{m} \hat{V}_i - V$, and the empirical MSE is $\frac{1}{m}\sum_{i=1}^{m}\left(\hat{V}_i - V\right)^2$.

6. You need to provide **2** figures, for plotting the MSE for three methods as well as the (approximate) theoretical MSE of the MLE.

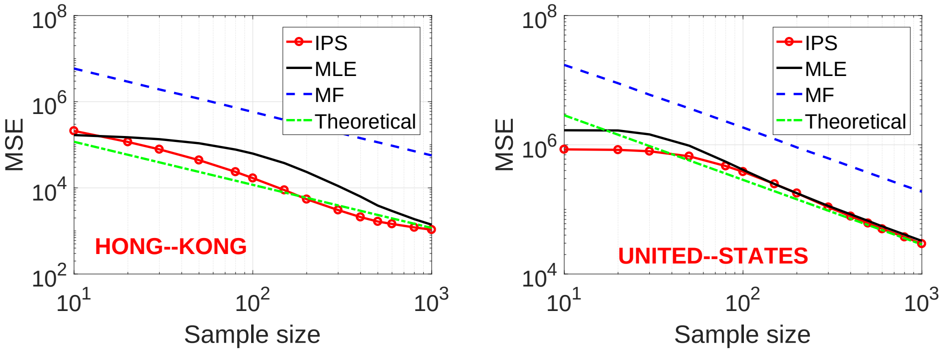

Figure 2: Empirical and theoretical MSE curves.

3 Submission Instructions

Your submission should include **7** files with the following names:

- 3 matlab files:

  [1] `Est_IPS.m`
  [2] `Est_MLE.m`
  [3] `TestMultinomialSampling.m`

- 2 figure files similar to Figure 1.

  [4] `IPSandMLE1.fig`
  [5] `IPSandMLE2.fig`

- 2 figure files similar to Figure 2.

  [6] `HONG_KONG.fig`
  [7] `UNITED_STATES.fig`

All the files should be submitted to Sakai in one zipped file (using WinZip).