ICEN 672/ICSI 660 - Detection & Estimation Theory
Project Report

Hadi Habibzadeh, and Omid Rajabi Shishvan

May 08, 2019

1. Part I

The first analysis involves the implementation of a binary test for the given dataset. The four categories are first divided into two hypotheses, each subsuming two different labels. Next, a more general case involving four hypotheses is developed. For both cases, the relevant tests are designed, the required parameters are estimated from a training dataset, and the performance measures are compared.

1.1 Task 1

Regardless of the distribution model, the binary hypothesis test under the Bayes criterion can be formally stated as

$$\Lambda(X) \triangleq \frac{P(X|H_1)}{P(X|H_0)} \gtrless \frac{P_0 \cdot (C_{10} - C_{00})}{P_1 \cdot (C_{01} - C_{11})} \triangleq \gamma, \tag{1.1}$$

where $\Lambda(X)$ denotes the likelihood ratio of observation $X \in \mathbb{R}^{2535}$, P_0 and P_1 respectively represent the prior probabilities of hypothesis 0 (H_0) and hypothesis 1 (H_1), and $P(X|H_j)$ for $j \in \{0,1\}$ is the conditional probability density function (PDF) of X on Hypothesis j. The likelihood ratio test of observation X, as defined in the preceding equation, merely calculates the ratio of conditional probabilities and compares them with a fixed threshold γ. If $\Lambda(X)$ exceeds the threshold, Hypothesis 1 is selected, otherwise, Hypothesis 0 is chosen. Two parameters determine the value of γ: **(i)** the ratio of prior probabilities of given hypotheses and **(ii)** the system's tolerance against various selection outcomes, which are denoted by C_{ij} in Eq. 1.1 for i and $j \in \{0,1\}$. C_{ij} determines the cost associated with the selection of Hypothesis i under the condition that Hypothesis j is true. Design goals determine the values of C_{ij}. The conditional and prior probabilities in Eq. 1.1 are typically derived analytically through the process model. In the absence of such a model, either a nonparametric approach or a supervised algorithm utilizing a training dataset can be used.

1.1.1 Modeling and Estimation

This work analyzes the conditional probabilities of observations by assuming two different models. One presumes a multivariate Bernoulli distribution whereas the other supposes a Multinomial one. In the former case, the conditional PDF of observation $X = (x_1, x_2, ..., x_m)$ under the Hypothesis $j \in \{0,1\}$ can be computed as:

$$\dot{P}(X|H_j) = \prod_{i=1}^{m} P(x_i|H_j)^{b_i} \Big(1 - P(x_i|H_j)\Big)^{1-b_i}, \tag{1.2}$$

where $b_i \in \{1,0\}$ is a binary variable that represents the presence of component x_i in observation X. b_i is set to 1 for every $x_i \neq 0$. Otherwise, $b_i = 0$. Clearly, the evaluation of this distribution hinges on computing the conditional probabilities of individual components, $P(x_i|H_j)$.

This work assumes little about the analytical expression of $P(x_i|H_j)$. Instead, it applies the maximum likelihood estimation to a training dataset to obtain their approximations. If the dataset contains N_j observations pertaining to Hypothesis j and assuming that for N_{ij} number of these observations $b_i = 1$, then the estimate of the conditional probability of feature i, $\hat{P}(x_i|H_j)$, can be computed as:

$$\hat{P}(x_i|H_j) = \frac{N_{ij}+1}{N_j+2}. \tag{1.3}$$

The second model assumes a Multinomial distribution for conditional probabilities. In this case,

$$\overset{\circ}{P}(X|H_j) = \prod_{i=1}^{m} P(x_i|H_j). \tag{1.4}$$

Similar to the previous case, the conditional probability of individual signals can be inferred from the training dataset. To this end, the maximum likelihood estimation yields:

$$\hat{P}(x_i|H_j) = \frac{\sum tf_{ij}+1}{\sum N_j+V}. \tag{1.5}$$

Within the N_j observations of Hypothesis j in the training dataset, $\sum tf_{ij}$ is the sum of all x_i's. $\sum N_j$ measures the sum of all features under the Hypothesis j. Finally V is the number of words in the dataset. Indeed, Eq. 1.2 and Eq. 1.4 are valid as long as features x_i are independent for every i.

The prior probabilities of each hypothesis (P_0 and P_1) are also estimated using maximum likelihood estimation. To this end, it was assumed that both these parameters follow a normal distribution by unknown means and variances. That is, $P_0 \sim \mathcal{N}(\mu_0, \sigma_0^2)$ and $P_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$. Although important for analyzing the efficiency of the estimation, σ_0^2 and σ_1^2 are not of much interest in this work. Assuming a normal distribution for prior probabilities significantly simplifies their maximum likelihood estimation as it reduces them to their corresponding sample means. In this case, assuming that the training dataset includes a total of N labeled observations, the expected values of P_0 and P_1 can be computed as:

$$\mu_0 = \frac{1}{N} \cdot \sum_{i=1}^{N} \mathbb{I}\big(X_i \text{ belongs to } H_0\big) \tag{1.6}$$

and

$$\mu_1 = \frac{1}{N} \cdot \sum_{i=1}^{N} \mathbb{I}\big(X_i \text{ belongs to } H_1\big), \tag{1.7}$$

where $\mathbb{I}(\cdot) = 1$ if the given condition is correct and $\mathbb{I}(\cdot) = 0$ otherwise.

1.1.2 Training

The training data are extracted from the given dataset; the first half of observations corresponding to each hypothesis is selected as the training samples. The selection process is deterministic, meaning that the algorithm constructs the training dataset by starting from the first sample and orderly going through observations until the sufficient number of entries is selected. No randomization or cross-validation is implemented as it was not required. After establishing the training dataset, the algorithm uses Eq. 1.6 and Eq. 1.7 to compute P_0 and P_1, respectively. Then, it pre-computes the estimated conditional probabilities for every feature using Eq. 1.3 and Eq. 1.5.

1.1.3 Classification

Once the training phase is complete, the conditional probabilities of test observations can be computed according to Eq. 1.2 or Eq. 1.4, depending on the model of interest. For a given set of costs, substituting these equations in Eq. 1.1 results in two different tests:

$$\prod_{i=1}^{m} \frac{\dot{P}(X|H_1)^{b_i}\left(1-\dot{P}(X|H_1)\right)^{1-b_i}}{\dot{P}(X|H_j)^{b_i}\left(1-\dot{P}(X|H_0)\right)^{1-b_i}} \gtrless \frac{\mu_0\cdot(C_{10}-C_{00})}{\mu_1\cdot(C_{01}-C_{11})} \tag{1.8}$$

and

$$\prod_{i=1}^{m} \frac{\mathring{P}(x_i|H_1)}{\mathring{P}(x_i|H_0)} \gtrless \frac{\mu_0\cdot(C_{10}-C_{00})}{\mu_1\cdot(C_{01}-C_{11})}. \tag{1.9}$$

The evaluation of the preceding equations, however, faces a computational difficulty. Considering the high dimensionality of the feature set, the products in these two equations can quickly drop to minuscule values, which can undermine the performance of the test by introducing numerical inaccuracies. The log likelihood ratio test (LLRT) can remedy this complication. LLRT is computed by taking the logarithm of both sides of Eq. 1.1, which yields

$$\ln\Lambda(X) \triangleq \ln\left(\frac{P(X|H_1)}{P(X|H_0)}\right) \gtrless \ln\left(\frac{P_0\cdot(C_{10}-C_{00})}{P_1\cdot(C_{01}-C_{11})}\right) \triangleq \gamma_2. \tag{1.10}$$

Hence, the LLRT of Eq.1.8 simplifies to

$$\boxed{\sum_{i=1}^{m} \ln\left(\frac{\dot{P}(X|H_1)^{b_i}\left(1-\dot{P}(X|H_1)\right)^{1-b_i}}{\dot{P}(X|H_j)^{b_i}\left(1-\dot{P}(X|H_0)\right)^{1-b_i}}\right) \gtrless \ln\left(\frac{\mu_0\cdot(C_{10}-C_{00})}{\mu_1\cdot(C_{01}-C_{11})}\right).} \tag{1.11}$$

Similarly, taking the logarithm of both sides of Eq. 1.9 results in

$$\boxed{\sum_{i=1}^{m} \ln\left(\frac{\mathring{P}(x_i|H_1)}{\mathring{P}(x_i|H_0)}.\right) \gtrless \ln\left(\frac{\mu_0\cdot(C_{10}-C_{00})}{\mu_1\cdot(C_{01}-C_{11})}\right).} \tag{1.12}$$

LLRT converts the product to summation and significantly reduces the chances of numerical errors. Depending on the desired model, the algorithm uses these two equations to classify the test observations.

1.1.4 Performance

The performance of the classification is assessed using five major measures:

- Accuracy (α)
- False alarm probability (P_F)
- Detection probability (P_D)
- Miss detection probability (P_M)
- Probability of error (P_E)

Let set S_{ij} represent all test samples of Hypothesis j that are classified as Hypothesis i. Also, let M be the number of hypotheses in the classification. For the binary testing, $M = 2$. Using these notations, accuracy is defined as the number of correct classifications to total number of observations in the test dataset. That is

$$\alpha = \frac{\sum_{m=0}^{M-1} |S_{mm}|}{\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} |S_{mn}|} \times 100, \tag{1.13}$$

where $|\cdot|$ is the set cardinality operator.

Probability of false alarm (P_F) and probability of detection (P_D) are defined as

$$\begin{aligned} P_F &= \int_{\gamma}^{\infty} P(\Lambda|H_0)\, d\Lambda \\ P_D &= \int_{\gamma}^{\infty} P(\Lambda|H_1)\, d\Lambda. \end{aligned} \tag{1.14}$$

Computing P_F and P_D based on Eq. 1.14 requires the analytic derivation of $P(\Lambda|H_0)$, which is intractable—if not impossible—for this problem. Instead they can be estimated as follows

$$\begin{aligned} P_F &= \frac{|S_{10}|}{\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} |S_{mn}|} \\ P_D &= \frac{|S_{11}|}{\sum_{m=0}^{M-1}\sum_{n=0}^{M-1} |S_{mn}|}. \end{aligned} \tag{1.15}$$

Probability of error P_E is evaluated as

$$P_E = P_0 \cdot P_F + P_1 \cdot P_M, \tag{1.16}$$

where $P_M = 1 - P_D$.

Table 1 tabulates the classification results of the binary test computed as explained in the preceding discussion. The results are computed for the minimum probability of error. That is $C_{00} = C_{11} = 0$ and $C_{01} = C_{10} = 1$. Overall, the performance of both models is on par, with the multivariate Bernoulli model slightly outperforming the other. A better comparison between the two models can be carried out by contrasting their respective receiver operating characteristic (ROC). As depicted in Fig. 1, the model based on multivariate Bernoulli distribution has less susceptibility to changes in the threshold values.

Table 1: Classification results for binary test using two different models. 'RT' represents the execution time (excluding the training time).

Measure	**Multivariate Bernoulli Model**	**Multinomial Model**
α (%)	98.485	98.106
P_F	0.0136	0.0204
P_D	0.9829	0.9829
P_M	0.0171	0.0171
P_E	0.0151	0.0189
RT (s)	10.32	3.71

1.2 Task 2

When the number of hypotheses (M) increases to beyond two, the optimal Bayes test for observation X reduces to computing

$$\beta_i(X) = \sum_{j=0}^{M-1} C_{ij} \times P(H_j|X) \tag{1.17}$$

for every $i \in \{0, 1, 2, ..., M-1\}$ and picking the one for which $\beta_i(X)$ is smallest. Applying Bayes rule to Eq. 1.18 gives the following

$$\beta_i(X) = \sum_{j=0}^{M-1} C_{ij} \times \frac{P(X|H_j) \times P_j}{P(X)}. \tag{1.18}$$

Again, the objective is to pick a hypothesis for which $\beta_i(X)$ is the smallest. Since $P(X)$ is a positive value that is independent of both i and j, it can be dropped from the preceding

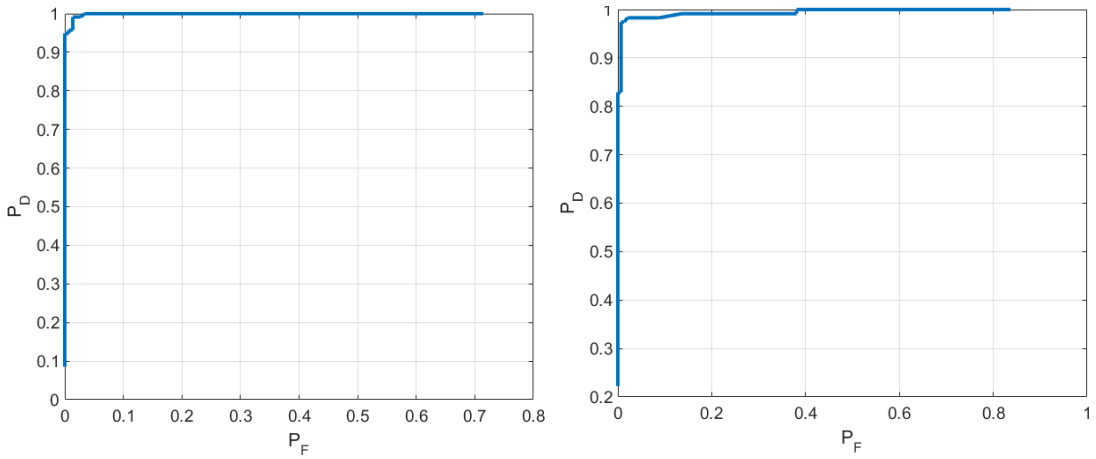

Figure 1: The ROC associated with Multivariate Bernoulli model (left) and Multinomial model (right). Both models perform well, yet the plots clearly show the superior performance of Multivariate Bernoulli model as it is less suscetible to changes in threshold. The ROC is computed for the minimum probability of error criterion.

equation. Hence, the optimal Bayes test reduces to

$$\boxed{\beta_i(X) = \sum_{j=0}^{M-1} C_{ij} \times P(X|H_j) \times P_j.} \tag{1.19}$$

Equation 1.3 and Eq. 1.5 can readily be extended to tests that involve more than two hypotheses. Hence, $P(X|H_j)$ can be computed based on Eq. 1.2 and Eq. 1.4, depending on the model of interest. Similarly, the extension of Eq. 1.6 and Eq. 1.6 to multi-hypotheses test is also straightforward. In particularly, for each $j \in \{0,1,2,...,M-1\}$, the expected value of P_j is given by

$$\mu_j = \frac{1}{N} \cdot \sum_{i=1}^{N} \mathbb{I}\big(X_i \text{ belongs to } H_j\big). \tag{1.20}$$

1.2.1 Training

The training process is very similar to the previous case (See Section 1.1.2) except that for M-test the observations are grouped into four different hypotheses based on their category. Once the training dataset is formed, Eq. 1.20 is used to estimate each P_j. Then, using Eq. 1.3 and Eq. 1.5, the algorithm precomputes and saves the conditional probabilities associated with each feature.

1.2.2 Classification

Using the variables learned during the training, the algorithm computes the conditional probability for a given teset observation X using Eq. 1.2 or Eq. 1.4, depending on the model. For every hypothesis, i, it then computes $\beta_i(X)$ according to Eq. 1.19. The output of

the classification is then determined by returning the hypothesis, for which $\beta_i(X)$ is the smallest.

1.2.3 Performance

The accuracy of the classifier is measured using Eq. 1.13. Let set $I = \{0,1,2,...,M-1\}$. To extend the definition of P_F and P_D to non-binary classifiers, the following definitions are available:

$$P_F = \sum_{i=0}^{M-1} \left(P_i \times \frac{\sum_{j \in I \setminus \{i\}} |S_{ij}|}{\sum_{j=0}^{M-1} |S_{ij}|} \right)$$

$$P_D = \sum_{i=0}^{M-1} \left(P_i \times \frac{|S_{ii}|}{\sum_{j=0}^{M-1} |S_{ij}|} \right). \tag{1.21}$$

To compute P_f, Eq. 1.21 first divides the hypotheses into two different groups; the hypothesis of interest and the rest. This artificially creates a binary test, for which P_F can be measured. The process is repeated until all hypotheses are selected exactly once as the hypothesis of interest. The final result is the expected value of these measurements. P_D can also be computed by a similar extension. Eq. 1.21 implies that for M-Test case, $P_D = \alpha$. Finally, the probability of errro can be computed as

$$P_E = \sum_{i=0}^{M-1} \left[\left((1-P_i) \times \frac{\sum_{j \in I \setminus \{i\}} |S_{ij}|}{\sum_{j=0}^{M-1} |S_{ij}|} \right) + P_i \times \left(1 - \frac{|S_{ii}|}{\sum_{j=0}^{M-1} |S_{ij}|} \right) \right]. \tag{1.22}$$

Table 2 tabulates the results for M-Test under two different models. Unlike the binary case, in this scenario, Multinomial model delivers a superior performance, albeit with a negligible margin. Not surprisingly, the classification performance is lower than that of binary classification. Particularly, the probability of error is increased with a factor of 10. Because the computation of $\beta_i(X)$ does not involve a threshold, no ROC is provided for the M-Test.

2. Part II

Table 2: Classification results for M-test using two different models. 'RT' represents the execution time (excluding the training time).

Measure	Multivariate Bernoulli Model	Multinomial Model
α (%)	96.969	97.348
P_F	0.0302	0.0265
P_D	0.9698	0.9735
P_M	0.0302	0.0265
P_E	0.1101	0.1107
RT (s)	77.6	76.2

2.1 Task 1

Expectation-Maximization (EM) algorithm [1], is an iterative algorithm that solves the maximum likelihood problems with unknown parameters. Although maximum likelihood problems can be directly solved in many cases, when the model is complex, direct solution of the equations is not feasible. EM algorithm can be utilized in these cases to find the solution to the complex equations.

2.1.1 Expectation-Maximization Goals and Formulation

Although EM algorithm is usually introduced as a two-step problem of the *Expectation* step and the *Maximization* step, to explain the formulation of the EM algorithm, we follow the steps and notations presented in [3].

The variables used in the problem are the observed data y, some unknown parameters θ from parameter space Ω, a parameter density function of $p(y|\theta)$, and some unknown data x with a parameter density function of $p(x|\theta)$. Note that x is not known and we only know a realization of x through y.

The goal is to find the maximum likelihood estimate of θ based on y or the maximum of log-likelihood:

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta\in\Omega} p(y|\theta) \tag{2.23}$$

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta\in\Omega} \log p(y|\theta) \tag{2.24}$$

The EM algorithm finds these estimates through an iterative process as following:

1. We make an initial estimate of θ by setting $m = 0$ in $\theta^{(m)}$.

2. We assume that $\theta^{(m)}$ is a correct guess. We then calculate the conditional probability distribution of x given the observed data y and $\theta^{(m)}$: $p(x|y,\theta^{(m)})$.

3. By taking $p(x|y,\theta^{(m)})$ from the previous step, we form the following expression for *conditional expected log-likelihood*, which is known as the Q-function:

$$\begin{aligned} Q\left(\theta|\theta^{(m)}\right) &= \int_{\mathcal{X}(y)} \log p(x|\theta) p\left(x|y,\theta^{(m)}\right) dx \\ &= E_{X|y,\theta^{(m)}}[\log p(X|\theta)] \end{aligned} \tag{2.25}$$

   where $\mathcal{X}(y)$ is the set $\{x|p(x|y,\theta) > 0\}$ and it is assumed to not depend on θ.

4. The next step is finding the θ that maximizes the Q-function. This new θ is taken as the new estimate for θ and is denoted as $\theta^{(m+1)}$.

5. Now we can refer back to step 2 by setting $m := m + 1$ and repeat this process. Note that this process can be repeated for and infinite amount of time, and we can implement some criterion, under which the process can stop. For example we can stop the process if the change in the estimate of $\theta^{(m+1)}$ compared to $\theta^{(m)}$, or the likelihood of these two estimates, is negligible.

The 5 steps described above are usually described as a two-step process.

- **Expectation-step (E-step)**: Computation of The Q-function given in 2.25.
- **Maximization-step (M-step)**: Updating the estimate of θ by the following expression:

$$\theta^{(m+1)} = \arg\max_{\theta\in\Omega} Q\left(\theta|\theta^{(m)}\right) \tag{2.26}$$

Since based on this definition, the Q-function is include in the M-step, we can assume the EM algorithm of an iteration of the Maximization step

2.1.2 Expectation-Maximization Challenges

Although the EM algorithm provides a mean to solve complex maximum likelihood estimation problems, it faces certain challenges.

The main possible problem with the EM algorithm is that, even though it's iterative process is proved to provide a monotonously increasing estimate of the likelihood function, it does not guarantee to provide the best solution [3]. Depending on the initial starting point of the algorithm and the characteristics of the likelihood function and the Q function, EM might converge to a local maximum which is different from the global maximum. One possible solution to this problem is to run the EM algorithm with various initial guesses and choose the result that has the maximum likelihood compared to the other results [3].

Another possible shortcoming of EM algorithm is that it may not necessarily simplify the computation compared to directly maximizing the likelihood function. The last problem for the EM is the speed of convergence. Compared to other numerical optimization approaches, EM has slower convergence speed and although there are techniques that speed up the EM process, the effectiveness of these techniques depends on the specific problems that are being solved [3].

2.2 Task 2

Summary of the research presented in [2].

2.2.1 Problem Definition

The study presented in [2] focuses on unsupervised discovering of interesting places in a city. The proposed scheme, called Physical-Social-aware Interesting Place Discovery (PSIPD), utilizes data gathered through location-based social sensing applications for this purpose. The main topics that are addressed in this study is taking the *physical dependency* of the locations and the *social dependency* of the social sensing applications users into consideration and discovering interesting places in an *unsupervised* approach.

2.2.2 Problem Formulation

To solve the interesting-place finding problem, authors formulate the problem as a Maximum Likelihood Estimation (MLE) problem. There is a set of M users (U_1, U_2, ..., U_M) and a set of N places (P_1, P_2, ..., P_M) where a binary value is assigned to each place ($P_i = I$ or $P_i = \bar{I}$), indicating if it is an interesting place or not

The following matrices are then defined and created based on the input data.

- An $M \times N$ *User-Place* matrix, called UP, indicates if the users have visited a location or not. In this matrix, if the i-th user has visited the k-th place, U_iP_k is set ot 1, otherwise, it is set to be 0.
- A symmetric $M \times M$ *Social-Dependency* matrix, called SD, indicates the social dependence of the users to each other. If user i and user j have a friend relationship, then $SD_{i,j} = SD_{j,i} = 1$, otherwise $SD_{i,j} = SD_{j,i} = 0$.

Another information inferred from the input data is the joint probability distributions of physical dependency of places to each other. For this piece of data, the N places are divided to R independent groups where the places in each group make a cluster of locations that are physically close to each other. Then for each of these groups, a joint probability distribution PD_r is defined that shows the dependency between different locations in group r.

The paper then introduces some terms that are used to formulate and solve the problem at hand. Since incorporating social dependencies is a part of the solution, the defined terms take that into account.

Two probability terms of Te_i and $Te_{i,j}$ are defined as *independent travel experience* and *dependent travel experience* respectively. Independent travel experience shows the probability of place P_k being interesting, given that user U_i has visited that place. Similarly, dependent travel experience shows the probability of P_k being interesting and a friend of U_i (U_j) visiting it, given that U_i has already visited that place. These probabilities are defined in the following expression:

$$\begin{aligned} Te_i &= \Pr(P_k = I | U_iP_k = 1) \\ Te_{i,j} &= \Pr\left(P_k = I, U_jP_k = 1 | U_iP_k = 1\right) \end{aligned} \tag{2.27}$$

Then the probabilities of independent and dependent users visiting a place, given that that place is interesting, and given that if their friends have visited that place are defined as following:

$$\begin{aligned} E_i &= \Pr(U_i P_k = 1 | P_k = I) \\ E_{i,j} &= \Pr\left(U_i P_k = 1 | U_j P_k = 1, P_k = I\right) \\ F_i &= \Pr\left(U_i P_k = 1 | P_k = \overline{I}\right) \\ F_{i,j} &= \Pr\left(U_i P_k = 1 | U_j P_k = 1, P_k = \overline{I}\right) \end{aligned} \tag{2.28}$$

By defining the probability of user U_i visiting a place as p_i ($p_i = \Pr(U_i P_k = 1)$), and defining the probability of a place being interesting as d ($d = \Pr(P_k = I)$, the terms in 2.28 and 2.27 are related by the following expressions:

$$\begin{aligned} E_i &= \frac{Te_i \times p_i}{d}, \quad F_i = \frac{(1-Te_i)\times p_i}{(1-d)} \\ E_{i,j} &= \frac{Te_{i,j} \times p_i}{Te_j \times p_j}, \quad F_{i,j} = \frac{(1-Te_{i,j})\times p_i}{(1-Te_j)\times p_j} \end{aligned} \tag{2.29}$$

So to sum up, the known information in the problem are the physical dependency distribution PD, the user-place matrix UP, and the social dependency matrix SD and the problem's objective is to find the probability of each location being interesting based on these three inputs and the probability of user U_i finding a place interesting given they have visited that place. These two objectives are defined by the following expression:

$$\begin{aligned} &\forall k, 1 \le k \le N : \Pr(P_k = I | UP, PD, SD) \\ &\forall i, 1 \le i \le M : \Pr(P_k = I | U_i P_k = 1) \end{aligned} \tag{2.30}$$

2.2.3 Proposed Solution

To solve this maximum likelihood estimation problem, an expectation maximization solution is used. The observed data X is consisted of UP, PD, and SD which are defined in section 2.2.2. The parameters are $\Theta = \left(E_1, \ldots, E_M; F_1, \ldots, F_M; E_{1,j}, \ldots, E_{M,j}; F_{1,j}, \ldots, F_{M,j}; d\right)$ where all of its elements are defined in 2.29 and section 2.2.2. The unobserved variables (latent variables) in the problem are defined as Z where they indicate if a place is interesting or not ($z_k = 1$ if P_k is interesting and $z_k = 0$ if P_k is not interesting). Note that the notations used in this formulation are different from the ones used in section 2.1.1 that introduced the EM algorithm. The likelihood ratio for this problem is then written as following:

$$\begin{aligned} L(\Theta; X, Z) &= \Pr(X, Z | \Theta) \\ &= \prod_{r \in R} \Pr(X_r, Z_r | \Theta) = \prod_{r \in R} \Pr(Z_r) \times \Pr(X_r | Z_r, \Theta) \\ &= \prod_{r \in R} \left\{ \sum_{r_1, \ldots, r_h \in \Psi_r} \Pr\left(z_{r_1}, \ldots, z_{r_h}\right) \prod_{k \in r} \prod_{g \in C} \prod_{i \in g} \eta_{k,g,i} \right\} \end{aligned} \tag{2.31}$$

where R is the set of different independent location groups, making r to be a group of physically dependent locations. $\Pr\left(z_{r_1}, \ldots, z_{r_h}\right)$ is the joint probability distribution of places

Table 3: Definition of $\eta_{k,g,i}$

$\eta_{k,g,i}$	Conditions
E_i	$\|g\| == 1, U_i P_k = 1, z_k = I$
$1 - E_i$	$\|g\| == 1, U_i P_k = 0, z_k = I$
$\prod_{j \in g} E_{i,j}$	$\|g\| > 1, U_i P_k = 1, U_j P_k = 1, S_i D_j = 1, z_k = I$
$\prod_{j \in g}^{J} 1 - E_{i,j}$	$\|g\| > 1, U_i P_k = 0, U_j P_k = 1, S_i D_j = 1, z_k = I$
F_i	$\|g\| == 1, U_i P_k = 1, z_k = \bar{I}$
$1 - F_i$	$\|g\| == 1, U_i P_k = 0, z_k = \bar{I}$
$\prod_{j \in g} F_{i,j}$	$\|g\| > 1, U_i P_k = 1, U_j P_k = 1, S_i D_j = 1, z_k = \bar{I}$
$\prod_{j \in g}^{J} 1 - F_{i,j}$	$\|g\| > 1, U_i P_k = 0, U_j P_k = 1, S_i D_j = 1, z_k = \bar{I}$

in r and Ψ_r is all possible combinations of selecting or not selecting $r_1, \ldots, r_h$, so for a group with h number of locations, there are 2^h possible situations that are represented via Ψ_r. Moreover, C shows all different socially independent circles of friends and g iterates on those independent groups. The last variable is $\eta_{k,g,i}$ which is defined in Table 3. Note that $|g|$ in Table 3 shows the number of users in group g, so $|g| = 1$ shows that a user is independent and has no friends, while $|g| > 1$ indicates a group of people with social dependency amongst them.

Based on the likelihood estimation function in 2.31 the Q-function of the EM algorithm is defined as following:

$$\begin{aligned} Q\left(\Theta|\Theta^{(n)}\right) &= E_{Z|X,\Theta^{(n)}}[\log L(\Theta; X, Z)] \\ &= \sum_{r \in R} \Pr\left(z_{r_1}, \ldots, z_{r_h} | X_r, \Theta^{(n)}\right) \times \left\{ \sum_{k \in r} \sum_{g \in C} \sum_{i \in g} \log\left(\eta_{i,k,g}\right) + \log \Pr\left(z_{r_1}, \ldots, z_{r_h}\right) \right\} \end{aligned} \tag{2.32}$$

where $\Pr\left(z_{r_1}, \ldots, z_{r_h} | X_r, \Theta^{(n)}\right)$ is the conditional probability of all places in r, given the current estimate of parameters $\Theta^{(n)}$ and the available information on r via X_r. This conditional probability can be computed as following:

$$\Pr\left(z_{r_1}, \ldots, z_{r_h} | X_r, \Theta^{(n)}\right) = \frac{\Pr\left(z_{r_1}, \ldots, z_{r_h}; X_r, \Theta^{(n)}\right)}{\Pr\left(X_r, \Theta^{(n)}\right)} \tag{2.33}$$

where $\Pr\left(z_{r_1}, \ldots, z_{r_h}; X_r, \Theta^{(n)}\right)$ and $\Pr\left(X_r, \Theta^{(n)}\right)$ are defined by the following two expressions:

$$\begin{aligned} \Pr\left(z_{r_1}, \ldots, z_{r_h}; X_r, \Theta^{(n)}\right) &= \Pr\left(X_r, \Theta^{(n)} | z_{r_1}, \ldots, z_{r_h}\right) \times \Pr\left(z_{r_1}, \ldots, z_{r_h}\right) \\ &= \prod_{k \in r} \prod_{g \in C} \prod_{i \in g} \eta_{k,g,i} \times \Pr\left(z_{r_1}, \ldots, z_{r_h}\right) \end{aligned} \tag{2.34}$$

$$\begin{aligned}\Pr\left(X_r,\Theta^{(n)}\right) &= \sum_{r_1,\ldots,r_h\in\Psi_r}\left[\Pr\left(X_r,\Theta^{(n)}|z_{r_1},\ldots,z_{r_h}\right)\times\Pr\left(z_{r_1},\ldots,z_{r_h}\right)\right]\\ &= \sum_{r_1,\ldots,r_h\in\Psi_r}\left[\left(\prod_{k\in r}\prod_{g\in c}\prod_{i\in g}\eta_{k,g,i}\right)\times\Pr\left(z_{r_1},\ldots,z_{r_h}\right)\right]\end{aligned} \tag{2.35}$$

Now that the Q function is defined, we should find estimations of the parameters for the next iteration. To do this, the probability of location P_k being interesting given the parameters estimation at iteration n is shown by $Y(n,k)$ where $Y(n,k)=\Pr\left(z_k=I|X_k,\Theta^{(n)}\right)$.

Now to maximize the Q-function, its partial derivative with respect to the parameters are taken and set to 0 in each iteration. i.e, the solution to $\frac{\partial Q}{\partial E_i}=0$, $\frac{\partial Q}{\partial F_i}=0$, $\frac{\partial Q}{\partial E_{i,j}}=0$, $\frac{\partial Q}{\partial F_{i,j}}=0$, and $\frac{\partial Q}{\partial d}=0$ is calculated for each iteration. The updated version of these parameter estimates is then calculated with the following expressions:

$$\begin{aligned}E_i^{(n+1)} &= E_i^* = \frac{\sum_{k\in UP_i}Y(n,k)}{\sum_{k=1}^{N}Y(n,k)}\\ F_i^{(n+1)} &= F_i^* = \frac{\sum_{k\in UP_i}(1-Y(n,k))}{\sum_{k=1}^{N}(1-Y(n,k))}\\ E_{i,j}^{(n+1)} &= E_{i,j}^* = \frac{\sum_{k\in UP_{i,j}}Y(n,k)}{\sum_{k=1}(1-Y(n,k))}\\ F_{i,j}^{(n+1)} &= F_{i,j}^* = \frac{\sum_{k=1}^{N}Y(n,k)}{N}\\ d^{(n+1)} &= d^* = \frac{\sum_{k=1}^{N}Y(n,k)}{N}\end{aligned} \tag{2.36}$$

2.2.4 Evaluation Protocol

To evaluate their model and analyze the impact of social and physical dependency on the problem, authors introduce 3 variations of their scheme. (i) A model that only takes physical dependency between location into account (PSIPD-P), (ii) A model that only takes the social dependency between locations into account (PSIPD-S), and (iii) the full model that has both types of dependencies included (PSIPD-PS).

Data traces are imported from public datasets that have the location and time of user check-in information, in addition to the friendship information among the users. Locations of these datasets are then clustered together through a K-means algorithm and then manually checked to completely separate the locations to physical independent groups of places. The ground truth on the of places being interesting is then gathered from several travel advisory websites such as *TripAdvisor*, *Planet Aware* and *San Francisco Travel*. The other information that is gathered from the data is the social dependency information where independent users, in addition to bigger social circles are extracted to create matrix SD.

To evaluate their schemes, authors compare their performance to multiple other similar schemes in the literature. There are two approaches in the comparison of the performance,

one that shows the performance of the *estimation* and another one that shows the performance of *ranking*.

For the performance of the estimation, metrics such as precision, recall, and F1-score are used and in addition to these metrics, the ROC curves of the studied schemes in addition to the baseline models are plotted. For the ranking performance a set of metrics called *normalized discounted cumulative gains (NDCG)* are used to compare the proposed schemes with the baseline models.

The reported results show that all of the three proposed schemes perform better than the baseline in both the estimation and the ranking tasks, with the exception of PSIPD-S falling just short of the baseline in one of the experiments. It is also shown that among the three proposed models, PSIPD-PS either outperforms or has the similar performance compared to both PSIPD-P and PSIPD-S under all evaluated criteria.

2.2.5 Important Outcomes

One of the main outcomes of the paper is that it shows that through an unsupervised approach, and by just depending on basic check-in information of some users, it is possible to discover if a place is interesting or not.

Another important outcome of the paper is showing the importance of physical dependencies of locations and social dependencies of users in the problem of discovering interesting places.

2.3 Task 3

References

[1] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977): 1-22.

[2] Huang, Chao, and Dong Wang. "Unsupervised interesting places discovery in location-based social sensing." 2016 *International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2016.

[3] Gupta, Maya R., and Yihua Chen. "Theory and use of the EM algorithm." *Foundations and Trends® in Signal Processing* 4.3 (2011): 223-296.