

problem set no. 1 — due Wednesday 9/28 at 2:35 pm.

problem 1. tasks: find the distribution of $Y = g(X)$ given the distribution of X ; estimate unknown constants in the likelihood (MLE); write some short programs in R.

example with derivations. define the d -dimensional simplex \mathbb{S}^d as the set of points $\vec{u} \in \mathbb{R}^d$ such that the sum $\sum_{i=1}^d u_i < 1$ and individual components $u_i > 0$ for $i = 1 \dots d$. consider $\vec{x} \in \mathbb{R}^d$ with distribution $N_d(\vec{\mu}, \Sigma)$. consider the transformation $\vec{u} = g(\vec{x})$ such that

$$u_i = \frac{e^{x_i}}{1 + \sum_{j=1}^d e^{x_j}} \quad \text{and} \quad x_i = \log\left(\frac{u_i}{u_{d+1}}\right)$$

where $u_{d+1} = 1 - \sum_{j=1}^d u_j$. the points \vec{u} live in \mathbb{S}^d . compute the density of \vec{u} .

this result is known; the original paper by Aitchison & Shen (Biometrika, 1980) will be posted online. an illustration of the calculations involved is appended to this document.

question 1.1. (15 point) define the d -dimensional hypercube $\mathbb{H}^d \subset \mathbb{R}^d$ as the set of points \vec{u} with individual components $u_i \in [0, 1]$, for $i = 1 \dots d$. consider $\vec{x} \in \mathbb{R}^d$ with distribution $N_d(\vec{\mu}, \Sigma)$, where $\Sigma_{ii} = \alpha \forall i$, $\Sigma_{ij} = -\beta \forall ij$, and $\alpha, \beta > 0$. in the spirit of the example above, consider a transformation $\vec{u} = g(\vec{x})$ such that $g : \mathbb{R}^d \rightarrow \mathbb{H}^d$,

$$u_i = \frac{1}{1 + e^{-x_i}} \quad x_i = \log\left(\frac{u_i}{1 - u_i}\right).$$

compute the density of \vec{u} . show every step of your derivations to obtain

$$f_{\vec{U}}(\vec{u}) = \frac{1}{|2\pi\Sigma|^{\frac{1}{2}} \prod_{j=1}^d u_j(1 - u_j)} \exp\left(-\frac{1}{2} (\log(\vec{u}/(1 - \vec{u})) - \vec{\mu})' \Sigma^{-1} (\log(\vec{u}/(1 - \vec{u})) - \vec{\mu})\right).$$

question 1.2. (20 points) compute $E(\vec{u})$, $V(\vec{u})$ in 1.1. (hint. 2nd-order Taylor expand g .)

question 1.3. (20 points) derive maximum likelihood estimators for $\vec{\mu}, \alpha, \beta$ of $f_U(\vec{u})$ in 1.1.

question 1.4. (5 point) next we will write functions in R to work with $f_U(\vec{u})$. please read the section ‘what to submit’ carefully **before** writing any code. you will need write the functions detailed below:

1. `dhypercube(u,mu,alpha,beta)`: computes the density of a point $\vec{u} \in \mathbb{H}^d$,
2. `hypercube.mean(mu,alpha,beta)`: a function to evaluate $E(\vec{u})$,
3. `hypercube.var(mu,alpha,beta)`: a function to evaluate $V(\vec{u})$, and,

4. `hypercube.mle(U)`: a function to estimate the parameters $\vec{\mu}, \alpha, \beta$. this function should return a named list of the form:

```
list("mu.hat"=mu.hat,"alpha.hat"=alpha.hat,"beta.hat"=beta.hat)
```

where `*.hat` are the respective MLEs.

(hint. you can simplify your life substantially by building your functions around the corresponding functions for the multivariate normal in the existing R package `mvtnorm`.)

question 1.5. (15 points) the function `rhypcube` is provided to sample points $\vec{u} \in \mathbb{H}^d$ so that you can check you estimators. pick three configurations of the parameters $\vec{\mu}, \alpha, \beta$ and report the true values versus your point estimates.

question 1.6. (20 points) use the sample of 250 points provided in the file `dataHypercube3D.txt` to compute the estimates for the parameters $\vec{\mu}, \alpha, \beta$. your code should read the file in using the command `read.table("dataHypercube3D.txt",header=TRUE)`. note that for your code to successfully execute this requires that you execute your script in the folder containing `dataHypercube3D.txt`.

for the following parts, please name any figures that you produce in the format:

`myfasusername_fig_x.ext`, where `x` is the figure number (1,2,...), `ext` is the filename extension (pdf, jpg, gif) and where `myfasusername` is replaced by your actual FAS username.

question 1.7. (3 points) how might you assess the variability of the estimates in the previous two questions? derive a computation or describe an algorithm to do this. apply your solution to assess the variability of the estimates you obtained for 1.6.

question 1.8. (2 points) the hypercube model is described is only one specific model for data on \mathbb{H}^d . since the individual components u_i , $i = 1, \dots, d$ fall on $(0, 1)$, we could instead choose to model each u_i as an independent Beta random variable¹. compare the goodness of fit of the Hypercube model for the data in the file `dataHypercube3D.txt` versus the goodness of fit of a set of d independent Beta models? your answer should compare estimates or other statistics computed under both models.

¹for those of you who are not familiar with the beta distribution you may want to start at: http://en.wikipedia.org/wiki/Beta_distribution

problem 2 This problem is based on replicating an analysis from “Malaria risk on the Amazon frontier” by de Castro et. al. (PNAS 2006). They use an unsupervised method known as a “grade of membership” or GoM model; this is a special form of mixture model. Your task will be to replicate a subset of their model-building and computation. This will involve probability modeling and fundamental optimization methods.

The data consists of features for a set of plots in the border of the Amazon inhabited by settlers. The objective of this analysis is to identify features associated with high and low risks of contracting malaria; for example, having high-quality walls for one’s home would be expected to correspond to lower risk, whereas living close to water would likely raise one’s risk.

We assume that each plot has features that originate from one of two distributions: high or low risk. Our objective is then to estimate both the parameters associated with each risk profile and the probability that each plot is in the high-risk group. Note that the approach used here does not make use of the actual occurrence of malaria; instead, the authors assume there are two types of plots and attempt to identify them using only other features.

All features are categorical and differ in the number of levels; for example, there are only two levels for `chainsaw`, but there are five levels for `dist-hosp`. The data generating process underlying this model is:

Given $G_{N \times 2}$ and $\Theta_{2 \times p}$

for $n = 1, \dots, N$ **do**

for $p = 1, \dots, P$ **do**

$$P(X_{n \times p} | \vec{g}_n, \vec{\theta}_{L,p}, \vec{\theta}_{H,p}) = g_{L,n} \cdot \text{Mult}(\vec{\theta}_{L,p}, 1) + g_{H,n} \cdot \text{Mult}(\vec{\theta}_{H,p}, 1)$$

 where $\vec{g}_n = (g_{L,n}, g_{H,n})$

end for

end for

Here, $X_{n \times p}$ is a matrix of features for plot n , $\theta_{L,p}$ is the vector of multinomial probabilities for feature p in the low-risk group, $\theta_{H,p}$ is an analogous vector for the high-risk group. Finally, $g_{L,n}$ and $g_{H,n}$ are the probabilities that plot n belongs to the low or high-risk group, respectively ($g_{L,n} = 1 - g_{H,n}$).

The data for this problem can be found in the `dat` folder. The file `data1985_area2.csv` contains the features for this problem, along with a column labelled `id`. This is an identifier for each plot and should **not** be included in your X matrix. The file `theta0list.Rdata`

contains an R list with the initial values of $\theta_{H,p}$ and $\theta_{L,p}$ for each feature. If you are not using R, the same values are contained in the set of 49 CSVs found in `malaria_theta0.zip`. We have also included a CSV containing descriptions of each variable and its levels (`variables_theta0_1985.csv`) for your information.

question 2.1. (15 points) Write out the log-likelihood for this model. Be sure to specify all relevant variables and watch your subscripts.

question 2.2. (5 points) Write an function `ll(G, theta, X)` that computes the log-likelihood derived in 2.1.

question 2.3 (25 points) Find the MLE for (G, Θ) using a coordinate ascent approach, maximizing over parameters in the following sequence:

1. $g_{L,n}$ for $n = 1, \dots, N$
2. $\theta_{L,j}$ for $j = 1, \dots, J$
3. $\theta_{H,j}$ for $j = 1, \dots, J$

This should be implemented as function `gomMLE(X, G0, theta0)` where `G0` and `theta0` are the provided starting values for your iterations. Your function should return a list of the form `list(G.hat=G.hat, theta.hat=theta.hat, maxlik=maxlik)`, where the first two entries are the MLEs and the last is the maximized likelihood. When running this on the provided data, set `G0` to 1/2 for all plots, and use the provided values for `theta0`.

Note: Your code can assume that the mixture has only two components (H and L), but it should not assume that the number of covariates, the number of levels per covariate, or the number of observations is fixed. It should be flexible enough to run on similar data without modification. We will test your code on an independent set of data with different number of plots, features and different covariates.

computing note. It is strongly recommended that all students use **R** for the computing tasks required in this course. We will, however, accept **MATLAB** or **Python** code solutions to all assignments. Please note that **no support, troubleshooting or assistance** will be provided for **MATLAB** or **Python** code. Any **MATLAB** or **Python** code that produces an incorrect answer is likely to receive less partial credit than similar **R** code. Please also note that all coding restrictions and guidelines described must also be followed by **MATLAB** or **Python** users (with appropriate file extension changes: **.R** to **.M** or **.Py**). In short: if you are an experienced **MATLAB** or **Python** user and are confident you can complete the assignments throughout the course, then you are welcome to use **MATLAB** or **Python** instead of **R** (acknowledging all warnings).

what to submit. Please submit a pdf file with written answers (using latex or word) to all relevant questions and a zip file including all the codes separately via Canvas before 2:35 pm on Wednesday 9/28. Pick a username (lastname_firstname is recommended) throughout this class and submit a PDF and a ZIP file named as **username_ps1.pdf** and **username_ps1.ZIP** to Canvas. The ZIP file should contain an R script named as **username_ps1.R**. Your R script file must contain all of the functions detailed in 1.4., must read the supplied data file, estimate the parameters, and produce the plots, assuming all files are in the same folder. The ZIP file should also contain the PDF file named as **username_ps1.pdf**, as well as the TEX or DOC file with the solution, named as **username_ps1.tex** or **username_ps1.doc**. The latex source of this pset is provided for your convenience. If you have any questions about code formatting and organization then please post your question on **Discussion** panel in Canvas. Happy coding!

derivations for the simplex example. We define the d -dimensional simplex as

$$\mathcal{S}^d = \left\{ \vec{u} \in \mathbb{R}^d : \vec{u} > \vec{0}, \sum_{i=1}^d u_i < 1 \right\}. \quad (1)$$

Let \vec{x} be a random vector in \mathbb{R}^d with density function $p[x]\vec{x} = \mathcal{N}\vec{x} \mid \vec{\mu}, \Sigma$. Consider the transformation $g : \mathbb{R}^d \rightarrow \mathcal{S}^d$, where $\vec{u} = g(\vec{x})$ is defined as

$$u_i = \frac{e^{x_i}}{1 + \sum_{j=1}^d e^{x_j}}, \quad x_i = \log \left(\frac{u_i}{u_{d+1}} \right), \quad (2)$$

where $u_{d+1} = 1 - \sum_{j=1}^d u_j$. The density of \vec{u} can be computed by

$$p[u]\vec{u} = p[x]g^{-1}(\vec{u})J, \quad (3)$$

where the Jacobian is

$$J = |\det J|, \quad J = \frac{\partial \vec{x}}{\partial \vec{u}} = \left\{ \frac{\partial x_i}{\partial u_j} \right\}. \quad (4)$$

Assume $i \neq j$. We compute the elements of the Jacobian matrix as

$$\begin{aligned}\frac{\partial x_i}{\partial u_j} &= \frac{\partial}{\partial u_j} \log \left(\frac{u_i}{u_{d+1}} \right) \\ &= \left(\frac{u_{d+1}}{u_i} \right) \left(\frac{u_{d+1} \cdot 0 - u_i \cdot (-1)}{(u_{d+1})^2} \right) = \frac{1}{u_{d+1}}.\end{aligned}\tag{5}$$

For the case when $i = j$, the elements of the Jacobian matrix are

$$\begin{aligned}\frac{\partial x_i}{\partial u_j} &= \frac{\partial x_i}{\partial u_i} = \frac{\partial}{\partial u_i} \log \left(\frac{u_i}{u_{d+1}} \right) \\ &= \left(\frac{u_{d+1}}{u_i} \right) \left(\frac{u_{d+1} \cdot (1) - u_i \cdot (-1)}{(u_{d+1})^2} \right) \\ &= \frac{1}{u_i} \left(1 + \frac{u_i}{u_{d+1}} \right) \\ &= \frac{1}{u_i} + \frac{1}{u_{d+1}}.\end{aligned}\tag{6}$$

By these equations, we see that we can express the Jacobian matrix as

$$J = D + \vec{v}\vec{v}^T,\tag{7}$$

where $D = \text{diag}(1/u_1, \dots, 1/u_d)$ and \vec{v} is a vector with all entries equal to $(u_{d+1})^{-\frac{1}{2}}$. By Sylvester's determinant theorem,

$$\begin{aligned}\det J &= \det (D + \vec{v}\vec{v}^T) \\ &= \det (D) (1 + \vec{v}^T D^{-1} \vec{v}) \\ &= \left(\prod_{i=1}^d \frac{1}{u_i} \right) \left(1 + \frac{1}{u_{d+1}} \sum_{j=1}^d u_j \right) \\ &= \left(\prod_{i=1}^d \frac{1}{u_i} \right) \left(1 + \frac{1 - u_{d+1}}{u_{d+1}} \right) \\ &= \left(\prod_{i=1}^d \frac{1}{u_i} \right) \left(\frac{1}{u_{d+1}} \right) = \prod_{i=1}^{d+1} \frac{1}{u_i},\end{aligned}\tag{8}$$

and so we have

$$J = |\det J| = \left| \prod_{i=1}^{d+1} \frac{1}{u_i} \right| = \prod_{i=1}^{d+1} \frac{1}{u_i}.\tag{9}$$

for $\vec{u} > 0$. Consequently, the probability density function of \vec{u} is

$$p_u(\vec{u}) = \det 2\pi \Sigma^{-\frac{1}{2}} J \exp \left[-\frac{1}{2} (g^{-1}(\vec{u}) - \vec{\mu})^T \Sigma^{-1} (g^{-1}(\vec{u}) - \vec{\mu}) \right]\tag{10}$$

where $\{g^{-1}(\vec{u})\}_i = \log\left(\frac{u_i}{u_{d+1}}\right)$ and $J = \left(\prod_{i=1}^{d+1} u_i\right)^{-1}$.