Algorithmic Learning Theory
MSIS 26:711:685
Homework 3

Instructor: Farid Alizadeh
Due Date: Wednesday November 20, 2019, at 11:50PM

last updated on December 5, 2019

Instructions: Please read carefully

Please answer the following questions in **electronic** form and upload and submit your files to Sakai Assignment site, before the due date. Make sure to click on the **submit** button.

For this homework you should submit **three files** for each of questions 1, 2 and 3 as an R (or Python) scripts. For each part of questions 1, 2 and 3 have the necessary R/Python code along with answers to questions in the form of an output. For example, here is a sample R code for homework answers:

```
# Question 1)
print("Question 1a)\n")
knnModel <- knn (y ~ x, data = someData)
print(summary(knnModel))
print("Answer to Q1a):\n")
print("error rate is ...\n")
readline("Hit Enter to continue\n")
#Question 1b)
. . . .
```

Include ample comments explaining what you are doing for each part of your code.

For questions 3 you may use word processor software like Word or similar, or use hand-written answer. Either way, transform your solution into a **single pdf file for both questions** and upload and submit it on the course Sakai site.

1. **R/Python Project:** The imdb-5000 data is a collection of information about 5000 movies made in the US and around the world. Information about this data set can be obtained from Kaggle.com site. Go to this site, read the background and characteristics of this data set.

   1a) Download the zip file that contains the data. Using R data frames, or Python pandas, transform this datset into a data frame. In doing so you need to perform the following filtering and cleaning of the data:

   - Unzip the data, and read into a data frame called `movieDat`
   - For this homework we need only those features which are numerical. Create a new data frame called `nmovieDat` and copy only columns of `movieDat` which are numerical in it.

   Once you have prepared the `nmovieDat` data frame, print the head and tail to make sure you have read everything correctly.

   1b) Find the mean, variance and standard deviation of each numerical feature and print your results. Comment on whether it is wise to do a scaling of the data before running any type of clustering or not.

   1c) Now apply the principal component analysis on the data using `prcomp` in R (or the equivalent in Python.) Based on your response on part 1b) above decide whether the option `scale` should be true or false. Save the output of `prcomp` in an object called `usarrests.pca` To see the components of this object, use the `names` function and see what information is in it. Print the result. Extract the `center` and `scale`. Explain why these values are different (or the same as) the mean and variances above.

   1d) Print the rotation matrix.

   1e) Extract the standard deviation of the principal components, these are the singular values of the data matrix. Verify this by using matrix operations (svd, matrix multiplication, etc.)

   1f) Print and graph the contribution of each principal component by graphing their variance (from the highest to lowest). You may use the `screeplot` function in R, see its documentation. Do you think it would be justified to keep all four features, or would it be wise to drop one or more? Justify your answer.

   1g) Use the `biplot` function of R (see its documentation) to plot the "*factor loadings*" of each data point on the first two principal components. Observe all vectors corresponding to all numerical features. Each one has a PC1 and a PC2 components. Observe the graph. Which features have roughly similar factor loading (coordinates in each PCA direction), and which one(s) are visibly different from the others? Observe the `actor1_facebook_likes` vector. Look at the projection of each state on this vector. What type of states project to the end of the vector, and what types project to the opposite direction?

See attached R script.

2. **R/Python Project:** R comes with a pre-loaded data called `USArrests`. Check the documentation of this dataset to see what it is about. This dataset is also available in Python for those of you using Python. In this homework you will run some experiments with the k-means method for clustering.

   2a) For each value of `k` from 2 to 10, run the k-means algorithm. For each run set the `nstart` parameter to 20 so that the algorithm is run with 20 different random starting points. To make sure all results are reproducible before starting run the command `set.seed(3)`. This will make all random numbers generated the same (this makes it easier to grade). Create an empty vector `ssw`. For each `k` from 2 to 10, compute `ssw=(total within ss)/(total ss)`. This is the ratio of within-cluster sum-of-square distances to the total sum-of-square of all distances. By looking at the plot decide what would be a good choice of `k`.

   2b) Set `k`, the number of clusters to 3. Run the k-means algorithm for $k = 3$ and with `nstart=20`. Print the cluster of each state based on the result.

   2c) Create a scatter plot with the X axis assault rate, and the Y axis as the murder rate. Color those states in cluster 1 as blue, those in cluster 2 as orange, and those in cluster 3 as maroon. Also, in addition to "dots" on the scatter plot, print the name of the state as well. For this check documentation for the `text` function of R.

   2d) Repeat 2c) but using Rape on the X axis and Murder on the Y axis.

   2e) Repeat 2c) but using Assault on the X axis and Rape on the Y axis.

   2f) Repeat 2c) but using UrbanPop on the X axis and Murder on the Y axis.

   2g) Now load the `rgl` library. Plot a 3D graph with Murder on the X axis, Rape on the Y axis and Assault on the Z axis. Use the same color code as the 2D plots for each cluster. Also add the name of each state on each data point using `text3d` function (check the documentation).

   See attached R script.

3. In this exercise you are going to derive the ML and MAP estimates for the exponential distribution. Recall that a random variable X follows the exponential distribution if its pdf is given by

$$f_e(t \mid \lambda) = \lambda \exp(-t\lambda)$$

   Recall also that this distribution is useful for modeling "waiting time" for an event to happen (such as arrival of a bus, or arrival of a customer to store, or a hit to a website), where the arrivals are

independent of each other and the rate of arrival (here λ) is fixed. So λ is the number of arrivals per unit of time. As a result, $\beta = \frac{1}{\lambda}$ is the average waiting time for the first arrival. You know that $\mathbb{E}(X) = \beta = \frac{1}{\lambda}$.

3a) Fatma assumes that waiting time for her bus in the morning follows approximately the exponential distribution. She wishes to estimate the rate λ through the maximum likelihood method. She records for N days arrival times of $t_1, t_2, \ldots, t_N$. Write down the likelihood function for these observations, clearly indicating what are the variables, and what are known constants.
The likelihood function

$$\text{lik}(\lambda \mid t_1, \ldots, t_N) = \prod_{i=1}^{N} f_e(t_{t_i} \mid \lambda) = \prod_{i=1}^{N} \Big(\lambda \exp(-t_i \lambda)\Big) = \lambda^N \exp\Big(-\lambda \sum_i t_i\Big)$$

3b) Write down the algebraic expression for the log likelihood function.

$$\log \text{lik}(\lambda \mid t_1, \ldots, t_N) = N \log \lambda - \lambda \sum_i t_i$$

3c) Find the ML estimation λ_{ML} from the log likelihood function. Show your work.

$$\frac{d}{d\lambda} \log \text{lik}(\lambda \mid t_1, \ldots, t_N) = \frac{N}{\lambda} - \sum_i t_i = 0$$

$$\lambda_{ML} = \frac{N}{\sum_i t_i} \quad \text{or} \quad \beta_{ML} = \frac{1}{\lambda_{ML}} = \frac{\sum_i t_i}{N}$$

So, the most likely *waiting time* is the average of the observed times. The most likely λ (number of arrivals per unit of time) is the inverse of the most likely average waiting time.

3d) You are now going to develop the Bayesian maximum posterior likelihood function. First you need to encode the prior degree of knowledge or ignorance about λ through a probability distribution. To this end we can assume that our prior estimate of λ is given by λ_p, and we encode our confidence about this by thinking that we have made α_p "*virtual*" observations and from these observations we arrived at λ_p estimate. (For now we assume that α_p is a nonnegative integer.) So if α_p is large, it indicates stronger belief and confidence in λ_p (more virtual observations), and if it is small it indicates less confidence.

It turns out that to express this in the form of a pdf, we need to use the *gamma distribution* with parameters λ and α. Basically the gamma distribution indicates the random variable waiting time for α events to happen, if on average λ arrivals occur in a unit of time. The exponential distribution is a special case where $\alpha = 1$, meaning we are waiting for the first arrival. In fact,

the gamma random variable can be thought of as sum of α independent exponential random variables, $X_1, X_2, \ldots, X_\alpha$. The pdf for the gamma distribution id given by

$$f_\Gamma(t\,|, \alpha, \lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)} t^{\alpha-1} \exp(-t\lambda)$$

where $\Gamma(\alpha) = (\alpha - 1)!$ the factorial of $\alpha - 1$. The reason we use $\Gamma(\cdot)$ instead of the factorial notation is that this distribution, in fact, is defined for every *nonnegative real number* α and not just integers. In the Bayesian context, since our prior is based on virtual observations (as a means to encode our degree of belief) we can pretend that we have estimated our prior λ_p based on, say $\alpha_p = 3.5$ observations (more confident than 3, less than 4)[1].

Suppose the prior distribution for the unknown parameter $\beta = \frac{1}{\lambda}$ (that is the average waiting time parameter) follows the gamma distribution with parameters α_p and rate λ_p. Fatma has observed arrivals $t_1, t_2, \ldots, t_N$. Derive the posterior likelihood function and clearly indicated what is unknown and what is known constant.

3e) Derive the log posterior likelihood.

Since λ has a gamma prior distribution with hyper-parameters α_p and $\lambda_p = \frac{1}{\beta_p}$, its posterior pdf is given by

$$\begin{aligned} f_{\text{post}}(\lambda \mid t_1, \ldots, t_N, \lambda_p, \alpha_p) &= \frac{f(t_1, \ldots, t_N \mid \lambda) f_{\text{prior}}(\lambda \mid \lambda_p, \alpha_p)}{f(t_1, \ldots, t_i)} \\ &\propto \lambda^N \exp\Big(-\lambda \sum_i t_i\Big) \frac{\beta_p^{\alpha_p}}{\Gamma(\alpha_p)} \lambda^{\alpha_p - 1} \exp(-\lambda \beta_p) \end{aligned}$$

The function $f(t_1, \ldots, t_N \mid \beta) = \prod_i f(t_i \mid \beta)$ since the data are assumed to be iid, and each follows the exponential distribution. The log likelihood function is

$$\log \text{lik}\big(\lambda \mid t_1, \ldots, t_N\big) \propto N \log \lambda - \lambda \sum_i t_i + (\alpha_p - 1) \log(\lambda) - \beta_p \lambda + \text{Const.}$$

3f) Minimize the log posterior likelihood and derive the MAP estimate λ_{MAP}. Comment on which of the data $t_1, \ldots, t_N$, or the prior exerts influence in the outcome.

Find the derivative and set it equal to zero and solve for λ:

$$\frac{d}{d\lambda} \log \text{lik}(\lambda \mid t_1, \ldots, t_N) = \frac{N}{\lambda} - \sum_i t_i + \frac{\alpha_p - 1}{\lambda} - \beta_p = 0$$

$$\beta_{\text{MAP}} = \frac{1}{\lambda_{\text{MAP}}} = \frac{\sum_i t_i + \beta_p}{N + \alpha_p - 1}$$

[1] The gamma function is generalization of the factorial and is defined for all real numbers except zero and negative integers at which points its values are ∞. The exact definition of the gamma function is through an integral $\Gamma(t) = \int_0^\infty x^{t-1} \exp(t) dx$. It can be shown that for integers $\Gamma(n) = (n-1)!$

We see that the prior exerts influence since the prior belief of λ_p is added to the observed t_i. And the number of observations is added by the number of *virtual observations* $\alpha_p - 1$. In fact, the MAP solution can be written as

$$\lambda_{MAP} = \frac{\sum_i t_i + \beta_p}{N + \alpha_p - 1} = \frac{\frac{\sum_i t_i}{N} + \frac{\beta_p}{N}}{1 + \frac{\alpha_p}{N}} = \frac{1}{1 + \frac{\alpha_p}{N}}\bar{t} + \frac{\frac{\alpha_p - 1}{N}}{1 + \frac{\alpha_p - 1}{N}}\left(\frac{\beta_p}{\alpha_p - 1}\right)$$

So, the MAP estimate is a weighted average of observations $\bar{t}$ and the prior estimate $\frac{\beta_p}{\alpha_p - 1}$. Note that $\frac{\beta_p}{\alpha_p - 1} = \frac{1}{(\alpha_p - 1)\lambda_p}$, is the implied prior belief for average time that the $\alpha_p - 1$ arrivals are expected. The posterior distribution is now again gamma with

$$\alpha_{posterior} = \alpha_p + N - 1$$
$$\beta_{posterior} = \sum_i t_i + \lambda_p$$