

# ST340 Programming for Data Science

## Assignment 2

*Released: Monday week 5, 2019-10-28; Deadline: 12:00 on Monday week 8, 2019-11-18.*

## Instructions

- Work individually.
- Specify your student numbers and names on your assignment.
- Any programming should be in R. Your report should be created using R markdown. Submit a single knitted pdf document which includes any code you have written.

## Q1 Expectation Maximization

For the EM algorithm with the mixture of Bernoullis model, we need to maximize the function

$$f(\boldsymbol{\mu}_{1:K}) = f(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K) = \sum_{i=1}^n \sum_{k=1}^K \gamma_{ik} \log p(\mathbf{x}_i | \boldsymbol{\mu}_k),$$

where for  $i \in \{1, \dots, N\}$  each  $\mathbf{x}_i \in \{0, 1\}^p$ , for  $k \in \{1, \dots, K\}$  each  $\boldsymbol{\mu}_k = (\mu_{k1}, \mu_{k2}, \dots, \mu_{kp}) \in [0, 1]^p$ , and

$$p(\mathbf{x}_i | \boldsymbol{\mu}_k) = \prod_{j=1}^p \mu_{kj}^{x_{ij}} (1 - \mu_{kj})^{1-x_{ij}}.$$

- (a) Show that the unique stationary point is obtained by choosing for each  $k \in \{1, \dots, K\}$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^n \gamma_{ik} \mathbf{x}_i}{\sum_{i=1}^n \gamma_{ik}}.$$

- (b) The newsgroups dataset contains binary occurrence data for 100 words across 16,242 postings. Postings are tagged by their highest level domain; that is, into four broad topics `comp.*`, `rec.*`, `sci.*`, `talk.*`. The dataset includes `documents`, a  $16,242 \times 100$  matrix whose  $(i, j)$ th entry is an indicator for the presence of the  $j$ th word in the  $i$  post; `newsgroups`, a vector of length 16,242 whose  $i$ th entry denotes the true label for the  $i$ th post (i.e., to which of the four topics the  $i$ th post belongs); `groupnames`, naming the four topics; and `wordlist`, listing the 100 words.
- (i) Run the EM algorithm for the mixture of Bernoullis model on the newsgroups data with  $K = 4$ . You should use some of the code from the EM Lab to help you. A run on the newsgroups dataset could take over 10 minutes so it is recommended to test your code on a small synthetic dataset first.
- (ii) Comment on the clustering provided by your run of the algorithm. Can you measure its accuracy?

## Q2 Two-armed Bernoulli bandits

- (a) Implement both Thompson sampling and the  $\epsilon$ -decreasing strategy in this setting with the unknown success probabilities of the arms being 0.6 and 0.4.
- (b) Describe the behaviour of  $\epsilon$ -decreasing when the sequence  $(\epsilon_n)_{n \geq 1}$  is defined by  $\epsilon_n = \min\{1, Cn^{-1}\}$ , where  $C$  is some positive constant, and check whether it is consistent with your implementation.
- (c) Describe the behaviour of  $\epsilon$ -decreasing when the sequence  $(\epsilon_n)_{n \geq 1}$  is defined by  $\epsilon_n = \min\{1, Cn^{-2}\}$ , where  $C$  is some positive constant, and check whether it is consistent with your implementation.
- (d) Compare and contrast the implementations of  $\epsilon$ -decreasing and Thompson sampling for this problem.

### Q3 $k$ nearest neighbours

- (a) Create a function to do  $k$ NN regression using a user-supplied distance function, i.e.

```
knn.regression.test <- function(k,train.X,train.Y,test.X,test.Y,distances) {  
  # YOUR CODE HERE  
  print(sum((test.Y-estimates)^2))  
}
```

Predicted labels should use the inverse-distance weighting to each neighbour.

- (b) Test your function on the following two toy datasets using `distances.l1` from lab 6. Try different values of  $k$  and report your results.

**Toy dataset 1:**

```
n <- 100  
train.X <- matrix(sort(rnorm(n)),n,1)  
train.Y <- (train.X < -0.5) + train.X*(train.X>0)+rnorm(n,sd=0.03)  
plot(train.X,train.Y)  
test.X <- matrix(sort(rnorm(n)),n,1)  
test.Y <- (test.X < -0.5) + test.X*(test.X>0)+rnorm(n,sd=0.03)  
k <- 2  
knn.regression.test(k,train.X,train.Y,test.X,test.Y,distances.l1)
```

**Toy dataset 2:**

```
train.X <- matrix(rnorm(200),100,2)  
train.Y <- train.X[,1]  
test.X <- matrix(rnorm(100),50,2)  
test.Y <- test.X[,1]  
k <- 3  
knn.regression.test(k,train.X,train.Y,test.X,test.Y,distances.l1)
```

- (c) Load the Iowa dataset (see `?lasso2::Iowa` for details). Try to predict the yield in the years 1931, 1933, ... based on the data from 1930, 1932, ...

```
install.packages("lasso2")  
library("lasso2")  
data(Iowa)  
train.X=as.matrix(Iowa[seq(1,33,2),1:9])  
train.Y=c(Iowa[seq(1,33,2),10])  
test.X=as.matrix(Iowa[seq(2,32,2),1:9])  
test.Y=c(Iowa[seq(2,32,2),10])  
k <- 5  
knn.regression.test(k,train.X,train.Y,test.X,test.Y,distances.l2)
```

- (d) Try different values of  $k$ , and compare your results with ordinary least squares regression and ridge regression.