Statistical Models and Computing Methods, Problem Set 3

November 26, 2020 Due 12/10/2020

Problem 1.

A total of n instruments are used to observe the same astronomical source. Suppose the number of photons recorded by instrument j can be modeled as $y_j \sim \text{Poisson}(x_j\theta + r_j)$ where $\theta \geq 0$ is the parameter of interest, and x_j and r_j are known positive constants. You may think of θ, x_j, r_j as the source intensity, the observation time, and the background intensity for instrument j, respectively. Assume the photon counts across different instruments are independent.

(1) Write down the likelihood function for θ.

(2) Introduce mutually independent latent variables $z_{j1} \sim \text{Poisson}(x_j\theta)$ and $z_{j2} \sim \text{Poisson}(r_j)$ and suppose we observe only $y_j \equiv z_{j1} + z_{j2}$. Under this formulation, derive an EM algorithm to find the MLE of θ.

Table 1: Data (x_j, r_j, y_j)

x	1.41	1.84	1.64	0.85	1.32	1.97	1.70	1.02	1.84	0.92
r	0.94	0.70	0.16	0.38	0.40	0.57	0.24	0.27	0.60	0.81
y	13	17	6	3	7	13	8	7	5	8

(3) Apply your EM algorithm to the data set given by Table 1. What is the MLE?

(4) For these data compute the observed Fisher information and the fraction of missing information. (Recall the observed Fisher information is defined as the negative second derivative of the observed data log-likelihood evaluated at the MLE.)

Problem 2.

Consider a two-dimensional Gaussian mixture model

$$p_\theta(x) = \pi_1\mathcal{N}(\mu_1, \Sigma_1) + \pi_2\mathcal{N}(\mu_2, \Sigma_2), \quad \pi_1 + \pi_2 = 1, 0 \leq \pi_1, \pi_2 \leq 1$$

Here the model parameters are $\theta = \{\pi_1, \pi_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$. Download the data from the course website.

(1) Derive an EM algorithm to find the MLE of θ.

(2) Choose the starting value $\pi_1^{(0)} = \pi_2^{(0)} = 0.5, \mu_1^{(0)} = (0.05, 0), \mu_2^{(0)} = (0, -0.05), \Sigma_1^{(0)} = \Sigma_2^{(0)} = I_2$. Apply your EM algorithm to the data and report the contours of the two estimated Gaussian densities (together with the scatter plot of the data) at iteration $t = 0, 1, 5, 10, 20, 40$. Try other starting values for your EM. Does your EM always converge to the same point estimate? Explain it.

(3) Derive the gradient of the parameters. (Hint: for Σ, you can use Σ^{-1} as the parameter)

(4) Compare the standard gradient descent method to EM. Show $\ell^* - \ell$ as a function of the number of iterations (ℓ is the log-likelihood function and ℓ^* is the optimal) for both methods. Which one is better in this case? Explain it.

Problem 3.
Consider a hidden Markov model (HMM) for predicting protein secondary structure. For each observed amino acid sequence x,

$$p_\theta(x) = \sum_z p_\theta(x, z), \quad p_\theta(x, z) = p_\theta(z_0) \prod_{i=0}^{n-1} p_\theta(z_{i+1}|z_i) \prod_{i=0}^{n} p_\theta(x_i|z_i)$$

where $z = (z_0, \ldots, z_n)$ is the latent variables. Assume each latent variable $z_t \in \{1, \ldots, 8\}$. The model parameters $\theta = \{\pi, A, B\}$, where $p_\theta(z_0) \sim \text{Discrete}(\pi)$, $A \in \mathbb{R}^{8\times 8}$ is the transition probability matrix with $a_{ij} = p_\theta(z_{t+1} = j|z_t = i)$ and $B \in \mathbb{R}^{8\times 20}$ is the emission probability matrix with $b_{ij} = p_\theta(x_t = j|z_t = i)$. Download the data from the course website. Load the data and convert it from text strings into numeric indices with the following code

```
import numpy as np
from os import walk
mypath = 'proteins/'  # use path to data files
_, _, filenames = next(walk(mypath), (None, None, []))

mSeq = len(filenames)        # read in each sequence
x = []
for i in range(mSeq):
    f = open( mypath + filenames[i] , 'r')
    x.append( f.readline()[:-1] )  # strip trailing '\n'
    f.close()

xvals = set()  # extract the symbols used in x
for i in range(mSeq):
    xvals |= set(x[i])
xvals = list( np.sort( list(xvals) ) )
dx = len(xvals)

for i in range(mSeq):       # and convert to numeric indices
    x[i] = np.array([xvals.index(s) for s in x[i]])
```

(1) Derive an algorithm for computing the log-likelihood function using dynamical programming.
(2) Derive the forward-backward algorithm (i.e., two-pass algorithm) for computing the marginal probabilities of the latent variables needed in the E-step.
(3) Initialize the model parameters with the following code

```
np.random.seed(1234)

model.pi = np.random.rand(8)
model.A = np.random.rand(8,8)
```

```
model.B = np.random.rand(8,20)

model.pi /= model.pi.sum()
model.A /= model.A.sum(1, keepdims=True)
model.B /= model.B.sum(1, keepdims=True)
```

Implement your algorithms to evaluate $p(x^{(2)})$, $p(z_0|x^{(3)})$, and $p(z_3, z_4|x^{(1)})$. Here $x^{(i)}$ means the i-th sequence in the data.
(4) Derive the updating formulas in the M-step. Run EM on the data for 200 iterations and report the average log-likelihood (over different sequences) as a function of iterations. You may also want to generate new protein sequences based on your trained HMM model (Just for fun!).