STAT516: Stochastic Modeling of Scientific Data I Fall 2021

Final Exam

Due December 19th by 11:59pm

Instructor: Vincent Roulet *Grader: Alex Jiang*

Upload your answers to the following question on Gradescope in a pdf.

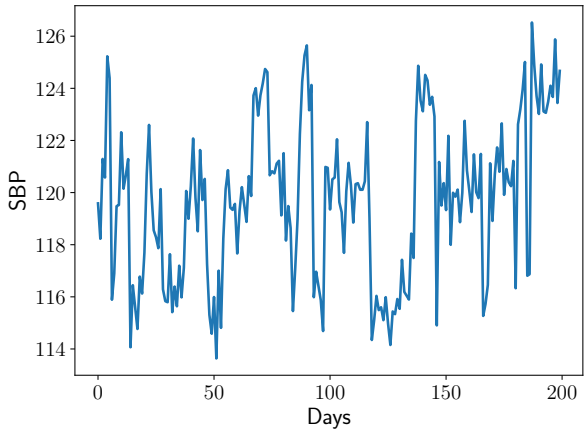
Make a new page for each exercise that will facilitate the grading.

Read the whole homework before starting it. Some early decisions in your code may have bad consequences at the end.

This exam is quite linear. However if you are stuck on some parts, feel free to send me a mail and I can give you for example the MLE for you to implement the Viterbi algorithm.

Analyzing personal health data

Consider that an individual measures its daily average Systolic Blood Pressure (SBP) with their smartwatch every day for 200 days which gives us the following data available in the file `SBP.csv`



We would like to analyze the data to understand if these patterns may correspond to some levels of stress the individual is going through over the days. Namely, we will consider that the daily average SBP can be modeled as an HMM with observations Y_t and hidden states X_t s.t.

$$Y_t \mid X_t \sim \mathcal{N}(\mu_{X_t}, 1)$$
$$X_1, X_2, \dots \quad \text{form a Markov chain on a finite state space } S \text{ of size } s$$

Our goal will be to estimate the underlying parameters $\mu_1, \dots, \mu_s$ and the transition probabilities of the data, understnad the patterns of this data (i.e., the hidden states), and predict the future SBP of this individual (will this person be able to relax for Christmas or not).

Getting some intuition form the data

Before even starting to code anything, a quick look at your data is valuable. For a Bayesian analysis, you would need to choose some priors. Even in the frequentist approach you will need good guesses of the parameters of the model. Indeed, EM or a gradient ascent are not guaranteed to find the minimum but if you start from a good initial guess it would surely help! So first, you shall try to fix the number of hidden states. After a quick look at your data, you may infer that $s = 3$ is a good choice.

1. Now do you think that the means would be equal? Do you think that starting an EM algorithm for example with an initial guess $\mu_1 = 1, \mu_2 = 2, \mu_3 = 3$ would be a good choice? So make your

guesses for the initial means (for a Bayesian analysis the priors on the means should be normal distributions centered around your guesses)

2. Now for the transition probabilities, what would you observe if the transition probabilities were all equal? Now what would you observe if p_{ii} were too be smaller than the p_{ij} for $j \neq i$? Make your guess for the transition probabilities

3. For the initial distribution, there is no way to make a good guess so a uniform distribution seems to be a good choice. In fact as we do not have access to multiple observations of this HMM, we may just consider the initial distribution to be fixed as a uniform distribution and estimate only the other parameters.

Coding part

Before starting coding, you shall think about a good representation of your parameters. The means are fine, that's the usual way you would tackle normal distributions, i.e., by simply estimating the means.

For the transition probabilities, there are two issues: first as said during the course, computing product of probabilities is a no-go when coding the actual algorithms. For example, no one is ever maximizing a likelihood. The usual path is to minimize the negative **log**-likelihood. So here rather than working with the matrix of transition probabilities, I would recommend you to work with $Q = (\log(p_{ij}))_{i,j \in S}$.

The second issue is that the sum of the transition probabilities of each row must be one. When coding EM or doing a Bayesian approach this would not be an issue as you would directly get parameters that satisfy that constraint (for the EM algorithm for example you compute directly the maximizers and so they naturally satisfy the constraints, for a Bayesian approach by sampling from a Dirichlet you know that you are sampling parameters that satisfy your constraints). But when coding a gradient descent that may be a problem as you want your iterates to satisfy the constraints. If the problem was convex, you could project your iterates on the set of constraints and guarantee some convergence. Here we will rather consider a simpler approach which consists in considering $\tilde{Q} \in \mathbb{R}^{s \times s}$ and define Q s.t. $Q_{ij} = \tilde{Q}_{ij} / \log(\sum_{k \in S} \exp(\tilde{Q}_{ik}))$, you should easily check that for any $\tilde{Q}$ the resulting $P = (\exp(Q_{ij}))_{i,j]inS}$ would satisfy $\sum_j p_{ij} = 1$.

Finally for the initial distribution, the same considerations apply: consider rather $v = (\log(\nu_i))_{i \in S}$ and if needed you'll optimize on $\tilde{v}$ where v would be defined as $v = \tilde{v}_i / \log(\sum_{j \in S} \tilde{v}_j)$.

Likelihood evaluation

1. Code a function that computes the logarithm of forward and backward probabilities associated to the model above given the observations, and given parameters (initial distribution parameterized by v as defined above, transition probabilities parameterized by Q as defined above, means $\mu_1, \ldots, \mu_s$ for any s)

   Hints:

   (a) You may simply code it for $s = 3$, we will consider our pipeline for $s \neq 3$ only for the last question that is optional so if you don't want to do the last question, you can simply code it for $s = 3$.

   (b) When encountering operations such as $\log(\sum_{i \in I} \exp(x_i))$ (known as log-sum-exp), you may want to stabilize your computations with the following trick:

   $$\log(\sum_{i \in I} \exp(x_i)) = \log(\exp(x_{\max}) \sum_{i \in I} \exp(x_i - x_{\max}))) = x_{\max} + \log(\sum_{i \in I} \exp(x_i - x_{\max})))$$

   Why? The exponential can easily produce very large values, by the above trick you ensure that the exponentials you compute ($\exp(x_i - x_{\max})$)) are always small.

2. Code a function that outputs then the negative log-likelihood of the observations given set of parameters (with the parameterization presented above). Test it on your initial guesses. Note that you can manually refine your guess by trying out different values here.

Parameters estimation

Choose one or more of the following questions (extra credits if you are implementing two or three of the following methods)

1. Code an EM algorithm. Be careful in your implementation as you may be using logarithms. Report the MLEs you computed.

2. Code a gradient descent to optimize the negative log-likelihood of your data. If you go for the gradient descent, use the parameterization described above. I would then suggest you to implement your code in Pytorch to easily have access to the gradients of your negative likelihood, see the tutorial https://pytorch.org/tutorials/beginner/basics/intro.html (all you need is the definitions of "tensors" and the part on automatic differentiation). You should also implement a backtracking line-search to ensure that your algorithm makes some progress at each step. Report the MLEs you computed.

3. Code a Bayesian data augmentation algorithm and report either the mean a posteriori or the maximum a posteriori of your parameters (for the transition probabilities, you may simply report the marginal means/modes of each probability and normalize them to get something relevant). For that approach explain your choices of priors (for the means, a good conjugate prior is another normal distribution, for the transition probabilities I let you see).

Hidden state inference

1. Code a Viterbi algorithm, which given some parameters return the most probable sequence of hidden states

2. Given your MLE or your Bayesian estimates or just a guess of the parameters of the HMM, estimate the hidden states. Make a plot that superpose the data and the mean of the corresponding inferred state (as in the earthquake example in the course)

Forecasting

1. Write down the forecasting probabilities for the observation in h days. To be rigorous, consider computing the cdf of $Y_{n+h} \mid Y_{1:n}$ and then derive the resulting pdf.

2. Propose an algorithm to predict the observations in the future

3. (Optional) Implement your proposed algorithm

Model selection (Optional)

1. Compute the AIC and the BIC for different values of the number of states s.

2. Conclude about the best possible set of parameters