ASEN 6519 Probabilistic Algorithms for Aerospace Autonomy
Spring 2019
Exercise 2: Maximum Likelihood Parameter Estimation
Out: Tuesday 02/26/2019 (posted on Canvas)
Due: Tuesday 03/12/2019, 5 pm (via Canvas)

Use whatever programming environment you wish, i.e. Matlab, Python, R, Julia, etc. – but do not simply use toolboxes or built-in functions in place of writing core algorithms for yourself. You need not hand in an overly formal report, but what you turn in should be neatly prepared, and address the questions provided below for each part in a clearly organized manner. Be sure to explain the logic of your solution, and appropriately comment your code (code should be in an appendix). ***You may work in groups of up to 2 students to turn in a single assignment (include both names on the assignment; only one person needs to turn it in). You may discuss your work with other students/groups, but work must be your own.***

Application Background This assignment will explore how maximum likelihood parameter estimation can be used to identify model parameters for a simple generative classification model and for variants of the HMM from Exercise 1.

Questions *(complete both problems)*

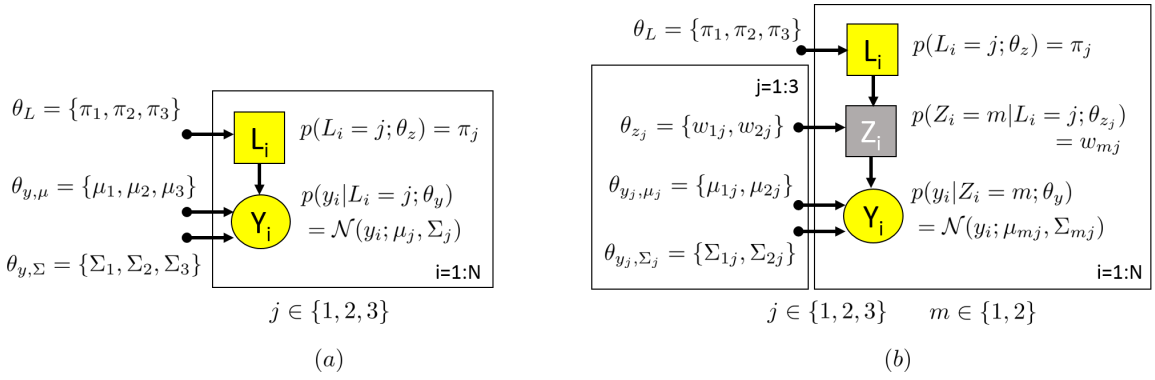

Figure 1: Two possible models for generative classification problem.

1. **Generative Classification:** Consider how arbitrary feature vectors $Y \in \mathbb{R}^2$ with realizations x can be assigned to a ternary discrete label set $L \in \{1, 2, 3\}$, using either of the two graphical models shown in Figure 1. In model (a), the so-called class-conditional feature pdfs $p(y|L = j)$ for each $j \in \{1, 2, 3\}$ are given by Gaussians; in model (b), the class-conditional feature pdfs are given by Gaussian mixtures. In both models, the class priors $P(L = j)$ are described by a CPT distribution. Such models are examples of *probabilistic generative classification models*, since (for a given set of model parameters) we would be able to generate features Y that are consistent for a selected class label L by performing logic sampling. To use either model for a

practical classification problem, it is typical to use Bayes' rule to find the label which maximizes the posterior as $\ell^* = \arg\max P(L = \ell | Y = y; \theta)$, for some estimated set of parameters θ and some new given instance of features $Y = y$.

Use the data in the posted 'ThreeClass_log.csv' file to estimate the unknown model parameters for model (a) via maximum likelihood, and repeat for model (b). The data log contains the discrete class label L_i and feature vector Y_i for each data sample in each row. In both cases, report the resulting log-likelihood and generate the corresponding 2D surface plot showing the posterior $P(L = 1 | Y = y; \hat{\theta}_{ML})$ as a function of $y \in \mathbb{R}^2$ (in some appropriate neighborhood). Which of the two models (a or b) better describes the data? (**Hint:** you need to do/show a little extra work to arrive at the appropriate maximum likelihood update equations for model (b); this can be done by starting from the ICLL and then making a suitable argument about how this can be optimized).

2. **Baum-Welch for Supervisory Operator HMM:** Implement the Baum-Welch algorithm for the posted set of $y_{1:T}$ observation data logs from the original Exercise 1 HMM in the 'nominal_hmm_multi_logs.csv' file (each column is an independent sequence of $y_{1:T}$ observations, with time indexed by row starting at $k = 1$). Compare the resulting estimated HMM CPTs $\hat{P}(x_k | x_{k-1})$ and $\hat{P}(y_k | x_k)$ to their true values from Exercise 1 (note that the true $P(x_0)$ distribution here is uniform, unlike in Exercise 1) – in particular, how does the accuracy of the parameter estimates change as the number of observed data sequences used for Baum-Welch varies, e.g. if only using the first 10, 20, or 50 data logs vs. all available data logs? How does the log-likelihood of the observed data sequences change?

Extra Credit Challenge Questions Attempt as many of these as you like, in any order you wish – **partial/full credit given only if you turn in a *complete* solution for any particular problem (i.e. no half-finished/incomplete attempts).**

C1. Consider a modified version of the previous HMM, which describes a similar application but a slightly different interface than the original UAV operator interface used by Boussesmart and Cummings. The number of states N_x needed for this new HMM is not known a priori, but the observation data grammar for the discrete y_k is the same as before with $N_y = 15$. Different sets of observation sequences $y_{1:T}$ for the new interface are given in the 'unknown_hmm_multi_logs.csv' file. Use Baum-Welch/EM and the Bayes Information Criterion (BIC) score to respectively estimate the HMM model parameters, i.e. $P(x_k | x_{k-1})$ and $P(y_k | x_k)$, and model order N_x for this new interface and data set. You should learn and compare HMMs with $N_x \in [2, 7]$ – report the BIC scores using a plot, along with the estimate of the parameters for the best model.

C2. Consider yet another modification to the original Boussesmart and Cummings HMM, where the discrete y_k observation grammar from before is replaced by a set of transformed/compressed set of continuous coordinates describing the human supervisor's mouse pointing and clicking actions on the UAV supervisor interface. Suppose these new observations are modeled as two dimensional conditional Gaussian random vectors $y_k \in \mathbb{R}^2$, where for $i \in \{1, ..., N_x\}$

$$p(y_k | x_k = i) = \mathcal{N}_{y_k, i}(\mu_i, \Sigma_i), \ \mu_i \in \mathbb{R}^2, \ \Sigma_i \in \mathbb{R}^{2 \times 2},$$

where vector μ_i and symmetric posdef covariance matrix Σ_i are the unknown model parameters for the conditional observation likelihood $p(y_k|x_k = i)$ (this conditional pdf replaces the $P(y_k|x_k)$ CPT from the original HMM). Starting from the ICLL for this new model, derive the required EM updates for the HMM using this new definition for y_k. Be sure to show the key steps needed to derive the E step and M step updates, and detail the actual parameter and probability updates that would be coded up (i.e. do not leave these in 'generic' form – one should be able to immediately implement the updates derived for both the E and M steps). Discuss: what is the relationship between this modified HMM and the GM model discussed in lecture?