Homework 6: Hidden Markov Models

UC Irvine CS275P: Graphical Models & Statistical Learning

Homework due at 11:59pm on May 29, 2025

Full Name:

UCINetID (e.g., ucinetid@uci.edu):

Question 0: (5 points)

Canvas includes detailed guidelines on how homework solutions should be formatted for submission to Gradescope. To receive full credit on this question, be sure that you read and follow these guidelines carefully! In particular, please take care that:

- There are two Gradescope assignments for Homework 6. For the first assignment, submit a PDF containing your answers for questions 1-2. For the second assignment, submit your Python code for question 2, which will be checked by an autograder.
- For the PDF submission, you must fill in this PDF template with your answers. All pages must remain in their original order when uploading your assignment to Gradescope. *Do not rearrange, add, or remove any pages from the provided PDF template.*
- For multiple-choice questions, ensure that you *completely fill in the bubble next to your selected answer*, and provide an explanation or justification to support your answer.
- Enter your final answers (to a precision of 3 decimal places) in the answer boxes, and write your explanations and/or math justifications in the space provided below each question. *Always show the work needed to produce your answers, not just the final result.* Math may be handwritten, but if possible, please type any sentences.
- For question 2, use the provided Python file. After adding your code to the template, submit that file separately to the second Gradescope assignment. In addition to uploading your Python code, be sure you provide answers and explanations in the designated areas of the PDF. *Do not include code for question 2 in the PDF.*
- If any question asks you to create a plot, insert an image showing the plot in the PDF.
- Check that your answers (especially scanned math) are readable within Gradescope, and that content has not been cut-off by the page margins.

Question 1: (25 points)

We first examine a simple hidden Markov model (HMM). We observe a sequence of rolls of a four-sided die at an "occasionally dishonest casino", where at time t the observed outcome $x_t \in \{1, 2, 3, 4\}$. At each of these times, the casino can be in one of two states $z_t \in \{1, 2\}$. When $z_t = 1$ the casino uses a fair die, while when $z_t = 2$ the die is biased so that rolling a 1 is more likely. In particular:

$$p(x_t = 1 \mid z_t = 1) = p(x_t = 2 \mid z_t = 1) = p(x_t = 3 \mid z_t = 1) = p(x_t = 4 \mid z_t = 1) = 0.25,$$
$$p(x_t = 1 \mid z_t = 2) = 0.7, \quad p(x_t = 2 \mid z_t = 2) = p(x_t = 3 \mid z_t = 2) = p(x_t = 4 \mid z_t = 2) = 0.1.$$

Assume that the casino has an equal probability of starting in either state at time $t = 1$, so that $p(z_1 = 1) = p(z_1 = 2) = 0.5$. The casino usually uses the same die for multiple iterations, but occasionally switches states according to the following probabilities:

$$p(z_{t+1} = 1 \mid z_t = 1) = 0.8, \qquad p(z_{t+1} = 2 \mid z_t = 2) = 0.9.$$

The other transition probabilities you will need are the complements of these. For all parts below, include work showing how you computed your answer. *Hint: For all of the questions below, you should not need to enumerate all possible state sequences.*

a) Under the HMM generative model, what is $p(z_1 = z_2 = z_3)$, the probability that the same die is used for the first three rolls?

$$p(z_1 = z_2 = z_3) = \boxed{}$$

b) What is $p(x_2 = 1 \mid x_1 = 1)$, the probability that the second roll is a 1 given that the first roll was also 1? Is this different from $p(x_2 = 1)$, the probability that the second roll is a 1 given no information about the first roll?

$$p(x_2 = 1 \mid x_1 = 1) =$$

$$p(x_2 = 1) =$$

c) Suppose that we observe the first two rolls. What is $p(z_1 = 1 \mid x_1 = 2, x_2 = 4)$, the probability that the casino used the fair die in the first roll?

$$p(z_1 = 1 \mid x_1 = 2, x_2 = 4) = \boxed{}$$

d) Consider a sequence of T rolls of the die, and let $z = (z_1, \ldots, z_T)$ and $x = (x_1, \ldots, x_T)$. What state sequence $\hat{z}$ and observation sequence $\hat{x}$ jointly maximize $p(z, x)$? You do not need to compute the numerical value of $p(\hat{z}, \hat{x})$, but justify your answer.

$\hat{z} = (\hat{z}_1, \ldots, \hat{z}_T) =$

$\hat{x} = (\hat{x}_1, \ldots, \hat{x}_T) =$

Question 2: (45 points)

We next learn hidden Markov models (HMMs) of the statistics of English text. In this application, each discrete "time" point corresponds to a single letter. For training, we use a chapter from Lewis Carroll's *Alice's Adventures in Wonderland*, available in `aliceTrainRaw.txt`. To simplify the modeling task, we first converted letters to lower-case and removed all punctuation. The resulting text, stored in `aliceTrain.txt`, is a sequence composed of 27 distinct characters (26 letters, as well as whitespace encoded via an underscore '_').

In many applications of HMMs, there is insufficient data or computational resources to select the model order via cross-validation. In these situations, the state dimension is often selected via either the *Akaike information criterion (AIC)* or *Bayesian information criterion (BIC)*. Let $x = (x_1, \ldots, x_T)$ denote the observed training sequence, $z = (z_1, \ldots, z_T)$ a hidden state sequence, and $\hat{\theta}_M$ an ML estimate of the parameters for an HMM with M states:

$$\hat{\theta}_M = \arg\max_{\theta_M} p(x \mid \theta_M) = \arg\max_{\theta_M} \sum_z p(x \mid z, \theta_M) p(z \mid \theta_M), \qquad x_t \in \{1, \ldots, M\}.$$

For this model, the AIC and BIC take the following form:

$$\begin{aligned} \text{AIC}_M &= \log p(x \mid \hat{\theta}_M) - d(M), \\ \text{BIC}_M &= \log p(x \mid \hat{\theta}_M) - \frac{1}{2} d(M) \log(T). \end{aligned}$$

Here, d(M) is the *number* of parameters (degrees of freedom) for an HMM with M states. The "best" model is then the one for which AIC_M or BIC_M is largest.

The `CategoricalHMM` class from the `hmmlearn` Python package provides an implementation of the expectation maximization (EM) algorithm for ML parameter estimation of HMMs for discrete data. *WARNING: Training HMMs of multiple orders via the EM algorithm may take an hour of computation time, so start early!*

a) Use the `CategoricalHMM.fit` *method, with default initialization, to learn HMMs with state dimensions of* $M = 1, 5, 10, 15, 20, 30, 40, 50, 60$. *(With* $M = 1$ *the HMM is a unigram model, which assumes that characters are independent.) For each of these models use a single random initialization, and run the EM algorithm for at most 500 iterations, or until the change in log-likelihood falls below* 10^{-6}. *Compute and store the log-likelihood which each model assigns to the training sequence. Save these models for later sections. (You do not need to show anything for this part.)*

b) Derive a formula for the number of parameters $d(M)$ in an HMM with M states, and observations taking W discrete values. Remember to account for normalization constraints: a discrete distribution on 4 events has only 3 degrees of freedom, since the probabilities must sum to one. Which criterion favors simpler models?

$d(M) =$

Which model selection criterion favors simpler models?

◯ ***AIC*** ◯ ***BIC***

Derivation of $d(M)$ and explanation for which criterion favors simpler models:

c) To test our learned HMMs, we use the text from a different chapter of Alice's Adventures in Wonderland*, available in* `aliceTest.txt`*. Using* `CategoricalHMM.score`*, evaluate the test chapter's log-likelihood for each HMM learned in part (a). Plot these test log-likelihoods versus M, together with the training log-likelihood* $\log p(x \mid \hat{\theta}_M)$*,* AIC_M*, and* BIC_M *versus M for the HMMs learned in part (a). Which model selection criterion better predicted test performance?*

Which model selection criterion better predicted test performance?

◯ ***AIC*** ◯ ***BIC***

d) Using the methods `CategoricalHMM.sample` *and* `num_to_text`*, generate a random 500-character sequence from four different HMMs: the model with no sequential dependence ($M = 1$), the model with the highest* BIC_M*, the model with the highest* AIC_M*, and the most complex model ($M = 60$). Compare and contrast these sequences. What aspects of English text do they capture? What do they miss?*

Discussion of what aspects of English text are captured and missed by the HMMs:

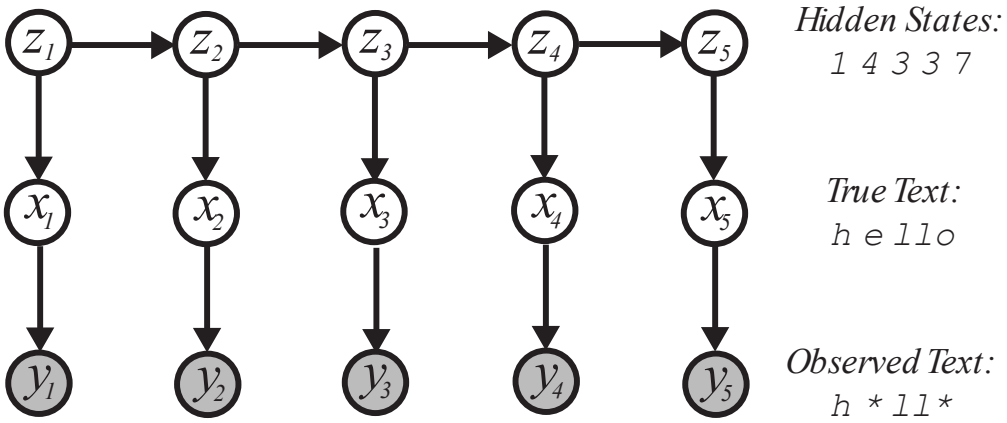

Figure 1: Graphical model illustrating an HMM with hidden states z_t which generate letters x_t. We observe a sequence y in which some of these letters have been erased.

In addition to computing likelihoods, HMMs lead to an efficient forward-backward algorithm which estimates the posterior probabilities of unobserved state sequences. We will now use this algorithm to estimate characters which have been *erased* from a text document.

Let $z_t \in \{1, \ldots, M\}$ denote the hidden state at position t, and x_t the "true" character at position t of some document. Suppose that instead of observing x, we observe an alternative sequence y in which some letters have been erased. We assume that each letter is independently erased with probability ϵ, so that

$$p(Y_t = x_t \mid X_t = x_t) = 1 - \epsilon, \qquad p(Y_t = * \mid X_t = x_t) = \epsilon.$$

Here, '*' is a special erasure symbol. Figure 1 shows a graphical model describing this generative process. Note that we never observe an "incorrect" letter; y_t is always either identical to x_t, or the erasure symbol '*'. The demo code shows how to make a noisy version of the test sequence `aliceTest.txt` by randomly erasing letters with probability $\epsilon = 0.2$.

e) Using the method `CategoricalHMM.predict_proba`*, compute the posterior distributions* $p(z_t \mid y)$ *for the four models from part (d). To do this, exploit the fact that*

$$p(y_t \mid z_t) = \sum_{x_t} p(y_t \mid x_t) p(x_t \mid z_t).$$

This implies that if we marginalize over the possible true letters x_t*, we recover a standard HMM in which the observations* y_t *are independent given the hidden state sequence* z*. (You do not need to show anything for this part.)*

f) Suppose that we observe an erasure at position t, so that $p(y_t = * \mid x_t) = \epsilon$ remains constant *as x_t is varied (since erasures provide no information about the underlying letter). Using the factored form of the generative model (as illustrated by the graph in Fig. 1), and the form of the observation model, show that the posterior distribution of x_t is*

$$p(x_t \mid y, y_t = *) = \sum_{z_t} p(x_t, z_t \mid y, y_t = *) = \sum_{z_t} p(x_t \mid z_t) p(z_t \mid y),$$

where $p(z_t \mid y)$ is the posterior distribution of z_t given the full *noisy sequence y.*

Derivation of equation for posterior distribution of x_t:

g) Using the marginal distributions $p(z_t \mid y)$ from part (e), and the equation from part (f), determine the most likely missing letter for each erasure. Or, equivalently, implement the decision rule which minimizes the expected number of incorrect characters. Note that when we observe a letter $y_t \neq *$ at position t, this implies that $x_t = y_t$.

Most likely missing letter at position t, expressed in terms of $p(x_t \mid y)$ from part (f):

$\hat{x}_t =$

h) Determine the percentage of missing letters which were correctly estimated by each model from part (d). What would chance performance be for this task? Print the first 500 characters of the denoised text produced by each model.

Accuracy for randomly choosing one of the 27 letters:

Accuracy for HMM with $M = 1$:

Accuracy for HMM with M selected by BIC:

Accuracy for HMM with M selected by AIC:

Accuracy for HMM with $M = 60$:

First 500 characters of denoised text for each of the four HMM models: