CS195f Homework 7

Mark Johnson and Erik Sudderth

Homework due at 2pm, 8th December 2009

We begin by learning hidden Markov models (HMMs) which describe 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 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 $y = (y_1, \ldots, y_T)$ denote the observed training sequence, $x = (x_1, \ldots, x_T)$ a hidden state sequence, and $\hat{\theta}_M$ an ML estimate of the parameters for an HMM with M states:

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

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

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

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.

You will need to use Kevin Murphy's Matlab HMM toolbox, available at

`http://people.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`

An example script providing partial solutions, as well as auxiliary functions useful for dealing with character data, are posted here:

`/course/cs195f/asgn/hmm`

WARNING: Training HMMs via the EM algorithm may take a few hours of computation time, so start early!

Question 1:

a) The method `dhmm_em.m`*, provided by the HMM toolbox, is an implementation of the expectation maximization (EM) algorithm for ML parameter estimation in HMMs. Use this*

package to learn HMMs with different hidden state dimensions (for example, try models with $M = 1, 5, 10, 15, 20, 30, 40, 50, 60$ states). Note that the model with $M = 1$ 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 the log–likelihood which each model assigns to the training sequence. Save these models for later sections.

b) *Derive a formula for the number of parameters d in an HMM with M hidden states, and observations taking one of W discrete values. Remember to account for normalization constraints (for example, a discrete distribution on 4 events has only 3 degrees of freedom, since the probabilities of these events must sum to one). Plot the training log–likelihood $\log p(y \mid \hat{\theta}_M)$, AIC_M, and BIC_M versus M for the HMMs learned in part (a). 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* `dhmm_logprob.m`*, evaluate the test chapter's log–likelihood with respect to each HMM learned in part (a). Plot these test log–likelihoods versus M. Which model selection criterion better predicted test performance?*

d) *Using the method* `sampleText`*, 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. Compare and contrast these sequences. What aspects of English text do they capture? What do they miss?*

In addition to computing likelihoods, HMMs lead to an efficient forward–backward algorithm which estimates the posterior probabilities of unobserved state sequences. This problem uses this method to estimate the identities of characters which have been *erased* from a text document.

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

$$\mathrm{P}[Z_t = y_t \mid Y_t = y_t] = 1 - \epsilon \qquad \mathrm{P}[Z_t = * \mid Y_t = y_t] = \epsilon$$

where '*' is a special erasure symbol. Figure 1 shows a graphical model describing this generative process. Note that we never observe an "incorrect" letter; z_t is always either identical to y_t, or the erasure symbol '*'.

Question 2:

a) *Starting with the test sequence from* `aliceTest.txt`*, generate a "noisy" text sequence by randomly erasing letters with probability $\epsilon = 0.2$. See the sample script for code which does this. Print out the first 500 characters of the noisy sequence.*

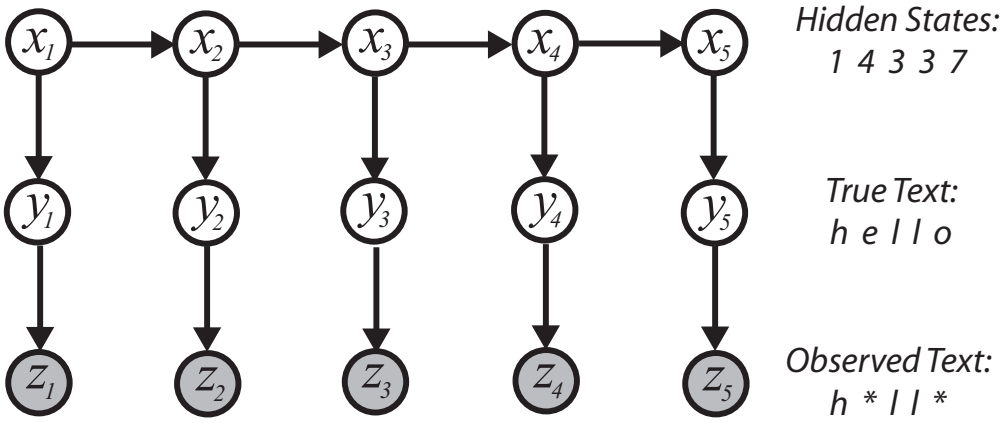

Figure 1: Graphical model illustrating an HMM with hidden states x_t which generate letters y_t. We observe a sequence z in which some of these letters have been erased.

b) Using the method `fwdback.m`*, compute the posterior distributions* $p(x_t \mid z)$ *for the four models from problem 1(d). To do this, exploit the fact that*

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

This implies that if we sum or marginalize over the possible values of the letters y_t*, we recover a standard hidden Markov model in which the observations* z_t *are independent given the hidden state sequence* x*.*

c) Suppose that we observe a letter $z_t \neq *$ *at position* t*. Argue that this implies that* $y_t = z_t$ *with probability one.*

d) Suppose that we observe an erasure at position t*, so that* $p(z_t = * \mid y_t) = \epsilon$ *remains* constant *as* y_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* y_t *is*

$$\begin{aligned} p(y_t \mid z, z_t = *) &\propto p(z_t \mid y_t) p(y_t \mid z_1, \ldots, z_{t-1}, z_{t+1}, \ldots, z_T) \\ &\propto \sum_{x_t} p(y_t \mid x_t) p(x_t \mid z) \end{aligned}$$

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

e) Using the marginal distributions $p(x_t \mid z)$ *from part (b), and the equation from part (d), determine the most likely missing letter for each erasure. Or, equivalently, implement the decision rule which minimizes the expected number of incorrect characters.*

f) Determine the percentage of missing letters which were correctly estimated by each model. What would chance performance be for this task? Print the first 500 characters of the denoised text produced by each model from problem 1(d), and comment on any differences.

In the final question, we revisit the Bayesian linear regression model from HW4. As before, given M basis functions $\phi_j(x)$, $j = 1, \ldots, M$, we model the dependence of response variables y_i on input covariates x_i as follows:

$$p(y_i \mid x_i, w, \beta) = \mathcal{N}(y_i \mid w^T \phi(x_i), \beta^{-1}) \qquad p(w \mid \alpha) = \mathcal{N}(w \mid 0, \alpha^{-1} I_M)$$

Rather than searching over a discrete grid of potential values for the hyperparameters α and β, we will instead estimate them via the EM algorithm. In the E-step, we compute the expected value of certain statistics of the M-dimensional vector of regression coefficients w. In the M-step, we use these to produce new estimates of α and β. For a more detailed overview, see Sec. 9.3.4 of Bishop's textbook, *Pattern Recognition and Machine Learning*. The previously distributed solutions for HW4 may also be useful.

Question 3: (200-level graduate credit)

a) For the normal distributions assumed above, derive the form of the expected "complete-data" log likelihood, $\mathrm{E}[\log p(y, w \mid x, \alpha, \beta)]$*, given* N *observations* $y = (y_1, y_2, \ldots, y_N)$*. What particular statistics do we need to determine the expectations of in the E-step, in order to concretely evaluate this expression?*

b) Take the derivative of the expression in part (a) with respect to α*, set it to zero, and determine the M-step update of* α*.*

c) Take the derivative of the expression in part (a) with respect to β*, set it to zero, and determine the M-step update of* β*.*

d) Using the equations from the preceding parts, implement the EM algorithm for this model. Consider two different families of basis functions, the polynomial and radial basis functions from HW4, both with order $M = 100$*. For each family, initialize* $\alpha^{(0)} = 0.01$*,* $\beta^{(0)} = 0.0025$*, and run EM until changes in the likelihood fall below* 10^{-6}*. Plot the resulting sequences of parameters* $\alpha^{(t)}$ *and* $\beta^{(t)}$*, as well as the corresponding likelihoods* $p(y \mid \alpha^{(t)}, \beta^{(t)})$ *(see the formula from HW4, problem 3(a)). What is the test accuracy of the resulting models, using the squared-error loss from HW4?*