Homework 2

Due at noon on February 10, 2016

GUIDELINES: You can brainstorm with others, but please solve the problems and write up the answers by yourself. You may use textbooks (Koller & Friedman, Russel & Norvig, etc.), lecture notes, and standard programming references (e.g., online Java API documentation). Please do NOT use any other resources or references (e.g., example code, online problem solutions, etc.) without asking.

SUBMISSION INSTRUCTIONS: Submit this assignment by Dropbox. You submission should include: A PDF containing written answers and all your code.

1 PROGRAMMING: LEARNING/INFERENCE IN HIDDEN MARKOV MODEL (45 POINTS)

Task description. The task we will consider in this assignment is optical character recognition (OCR). The dataset we provide consists of a sequence of words, one character per row. The very first character of each word was capitalized in the original data and has been omitted for simplicity. The format of the data is described in generate_hmm_plots.m, so please look through that file before continuing. This file provides sample code structure to generate plots.
In this problem you will implement maximum likelihood estimation and the forward-backward algorithm for Hidden Markov Models (HMMs). Let X_t denote the t-th letter in a word and O_t^k the value of the k-th pixel for the t-th character. The result should be a stationary model (one that does not depend on t), i.e., you should have a single distribution $p(X_1)$, a single CPT $p(X_t|X_{t-1})$ and 64 CPTs $p(O_t^k|X_t)$ (one for each pixel).

(a) **Parameter Estimation (MLE/MAP) in HMMs:** For this first part, you will set the parameters of the HMM using maximum likelihood and maximum a posteriori estimation using several values of pseudo-count/hyperparameter α.

Your task is to fill in the missing code in the files hmm_learn.m. Note that hmm_learn.m goes over the specifics of what parameters you need to learn. Because we will be comparing HMM to Naive Bayes later, hmm learn.m should also fit a probability model $p(X_t)$ which serves as the class prior for Naive Bayes.

To help debug your code and generate results, generate_hmm_plots.m will plot the transition model that you learn, and the observation model for the letter 'a'. You should

see that the transition model "makes sense", e.g. $p(X_t = u|X_{t-1} = q)$ should be high, and that the observation model looks like a blurry version of the desired letter.

(b) **The Forward Backward Algorithm:** In this part, you will implement the forward-backward algorithm for HMMs and compare its performance to a Naive Bayes approach which classifies each character independently of all others. You have two programming tasks for this sub-part.

i. hmm_fb.m - In this file, you will implement the forward-backward algorithm to compute marginal probabilities $P(X_t|O_1,\ldots,O_T)$. The input to this function is the trained model from hmm_learn.m and the pixel data corresponding to a *single word* (not the entire test set). See the file generate_hmm_plots.m to see how hmm_fb.m is used.

ii. generate_hmm_plots.m - Run this file to train the HMM model and evaluate it on the test data. Naive Bayes will serve as a baseline, but one critical line of code is missing in this file. Remember that Naive Bayes computes,

$$P(X_t|O_t) \propto P(O_t|X_t)P(X_t),$$

and that $P(X_t)$ was computed in hmm_learn.m, and $P(O_t|X_t)$ was computed in hmm_fb.m. You need to fill in this line before the code will run.

What to include in the write up: Try several values of the smoothing/pseudo-counts: $\alpha = 0,1,2,4,8$ and include the plots for the resulting observation model for 'a' and the transition model in the write-up. **Describe** in 1-2 sentences the effect of smoothing. Include a plot of accuracy on the test set vs. smoothing parameter α for HMM and NB. Next, **discuss** (3-4 sentences) how the two algorithms differ in performance, what their performance and errors are, and how/why they differ.

2 KALMAN FILTER (15 POINTS)

Often, we wish to monitor a continuous-state system whose behavior switches unpredictably among a set of k distinct "modes." For example, an aircraft trying to evade a missile can execute a series of distinct maneuvers that the missile may attempt to track. A Bayesian network representation of such a switching Kalman filter model is shown in Figure 2.1.

(a) Suppose that the discrete state S_t has k possible values and that the prior continuous state estimate $P(X_0)$ is a multivariate Gaussian distribution. Show that the prediction $P(X_1)$ is a mixture of Gaussians – that is, a weighted sum of Gaussians such that the weights sum to 1.

(b) Show that if the current continuous state estimate $P(X_t|e_{1:t})$ is a mixture of m Gaussians, then in the general case the updated state estimate $P(X_{t+1}|e_{1:t+1})$ will be a mixture of km Gaussians.

(c) What aspect of the temporal process do the weights in the Gaussian mixture represent?

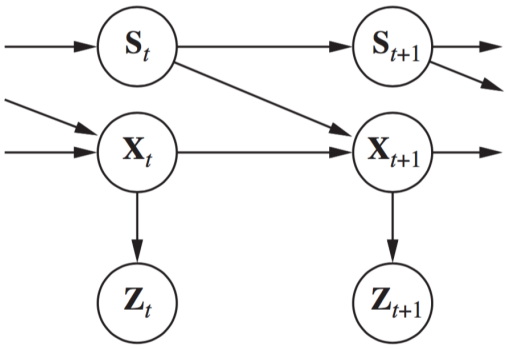

Figure 2.1: A Bayesian network representation of a switching Kalman filter. The switching variable S_t is a discrete state variable whose value determines the transition model of the continuous state variables $\boldsymbol{X}_t$. For any discrete state i, the transition model $P(\boldsymbol{X_{t+1}}|\boldsymbol{X}_t, S_t = i)$ is a linear Gaussian model, just as in a regular Kalman filter. The transition model for the discrete state, $P(S_{t+1}|S_t)$, can be thought as a matrix, as in a hidden Markov model.

The results in (a) and (b) show that the representation of the posterior grows without limit even for switching Kalman filters, which are among the simplest hybrid dynamic models.

3 Graph and Independence Relations (20 points)

For $i = 1,2,3$, let X_i be an indicator variable for the event that a coin toss comes up heads (which occurs with probability q). Supposing that that the X_i are independent, define $Z_4 = X_1 \oplus X_2$ and $Z_5 = X_2 \oplus X_3$ where $\oplus$ denotes addition in modulo two arithmetic.

(a) Compute the conditional distribution of (X_2, X_3) given $Z_5 = 0$; then, compute the conditional distribution of (X_2, X_3) given $Z_5 = 1$.

(b) Draw a directed graphical model (the graph and conditional probability tables) for these five random variables. What independence relations does the graph imply?

(c) Draw an undirected graphical model (the graph and compatibility functions) for these five variables. What independence relations does it imply?

(d) Under what conditions on q do we have $Z_5 \perp X_3$ and $Z_4 \perp X_1$? Are either of these marginal independence assertions implied by the graphs in (b) or (c)?

4 Restricted Boltzmann Machines (20 points)

Restricted Boltzmann Machines (RBMs) are a class of Markov networks that have been used in several applications, including image feature extraction, collaborative filtering, and recently in deep belief networks. An RBM is a bipartite Markov network consisting of a visible

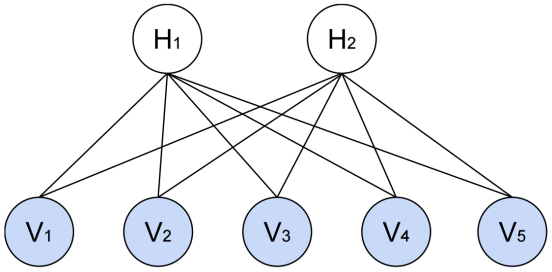

Figure 4.1: An example RBM with 5 visible units and 2 hidden units.

(observed) layer and a hidden layer, where each node is a binary random variable. One way to look at an RBM is that it models latent factors that can be learned from input features. For example, suppose we have samples of binary user ratings (like vs. dislike) on 5 movies: Finding Nemo (V_1), Avatar (V_2), Star Trek (V_3), Aladdin (V_4), and Frozen (V_5). We can construct the following RBM:
Here, the bottom layer consists of visible nodes $V_1, \ldots, V_5$ that are random variables representing the binary ratings for the 5 movies, and H_1, H_2 are two hidden units that represent latent factors to be learned during training (e.g., H_1 might be associated with Disney movies, and H_2 could represent the adventure genre). If we are using an RBM for image feature extraction, the visible layer could instead denote binary values associated with each pixel, and the hidden layer would represent the latent features. However, for this problem we will stick with the movie example. In the following questions, let $V = (V_1, \ldots, V_5)$ be a vector of ratings (e.g. the observation $v = (1,0,0,0,1)$ implies that a user likes only Finding Nemo and Aladdin). Similarly, let $H = (H_1, H_2)$ be a vector of latent factors. Note that all the random variables are binary and take on states in $\{0,1\}$. The joint distribution of a configuration is given by

$$P(V = v, H = h) = \frac{1}{Z} e^{-E(v,h)}, \tag{4.1}$$

where

$$E(v,h) = -\sum_{ij} w_{ij} v_i h_j - \sum_i a_i v_i - \sum_j b_j h_j \tag{4.2}$$

is the energy function, $\{w_{ij}\}, \{a_i\}, \{b_j\}$ are model parameters, and

$$Z = Z(\{w_{ij}\}, \{a_i\}, \{b_j\}) = \sum_{v,h} e^{-E(v,h)}$$

is the partition function, where the summation runs over all joint assignments to V and H.

(a) Using Equation (4.1), show that $p(H|V)$, the distribution of the hidden units conditioned on all of the visible units can be factorized as

$$p(H|V) = \prod_j p(H_j|V) \tag{4.3}$$

where

$$p(H_j = 1|V = v) = \sigma(b_j + \sum_i w_{ij} v_i)$$

and $\sigma(s) = \frac{e^s}{1+e^s}$ is the sigmoid function. Note that $p(H_j = 0|V = v) = 1 - p(H_j = 1|V = v)$.

(b) Give the factorized form of $p(V|H)$, the distribution of the visible units conditioned on all of the hidden units. This should be similar to what's given in part 1, and so you may omit the derivation.

(c) Can the marginal distribution over hidden units $p(H)$ be factorized? If yes, give the factorization. If not, give the form of $p(H)$ and briefly justify.

(d) Based on your answers so far, does the distribution in Equation (4.1) respect the conditional independencies of Figure 4.1? Explain why or why not. Are there any independencies in Figure 4.1 that are not captured in Equation (4.1)?