CSE 250A. Assignment 8

Out: *Tue Nov 21*
Due: *Tue Nov 28*

8.1 EM algorithm for binary matrix completion

In this problem you will use the EM algorithm to build a simple movie recommendation system. Download the files *hw8_movieTitles.txt*, *hw8_studentPIDs.txt*, and *hw8_movieRatings.txt*. The last of these files contains a fifty-column matrix of zeros, ones, and missing elements denoted by question marks. The $\langle i, j \rangle^{\text{th}}$ element in this matrix contains the i^{th} student's rating of the j^{th} movie, according to the following key:

1	recommended,
0	not recommend,
?	not seen.

(a) **Sanity check**

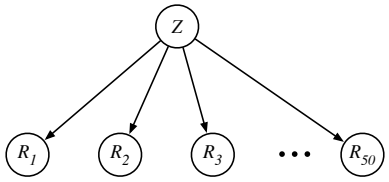
Compute the mean popularity rating of each movie, given by the simple ratio

$$\frac{\text{number of students who recommended the movie}}{\text{number of students who saw the movie}},$$

and sort the movies by this ratio. Print out the movie titles from least popular (*Fifty Shades of Grey*) to most popular (*Inception*). Note how well these rankings do or do not corresponding to your individual preferences.

(b) **Likelihood**

Now you will learn a naive Bayes model of these movie ratings, represented by the belief network shown below, with hidden variable $Z \in \{1, 2, \ldots, k\}$ and partially observed binary variables $R_1, R_2, \ldots, R_{50}$ (corresponding to movie ratings).



This model assumes that there are k different types of movie-goers, and that the i^{th} type of movie-goer—who represents a fraction $P(Z=i)$ of the overall population—likes the j^{th} movie with conditional probability $P(R_j=1|Z=i)$. Let Ω_t denote the set of movies seen (and hence rated) by the t^{th} student. Show that the likelihood of the t^{th} student's ratings is given by

$$P\left(\left\{R_j=r_j^{(t)}\right\}_{j\in\Omega_t}\right) \;=\; \sum_{i=1}^{k} P(Z=i) \prod_{j\in\Omega_t} P\left(R_j=r_j^{(t)}\,\middle|\, Z=i\right).$$

(c) **E-step**

The E-step of this model is to compute, for each student, the posterior probability that he or she corresponds to a particular type of movie-goer. Show that

$$P\left(Z=i \,\middle|\, \left\{R_j=r_j^{(t)}\right\}_{j\in\Omega_t}\right) \;=\; \frac{P(Z=i)\,\prod_{j\in\Omega_t} P\left(R_j=r_j^{(t)}\,\middle|\, Z=i\right)}{\sum_{i'=1}^{k} P(Z=i')\,\prod_{j\in\Omega_t} P\left(R_j=r_j^{(t)}\,\middle|\, Z=i'\right)}.$$

(d) **M-step**

The M-step of the model is to re-estimate the probabilities $P(Z=i)$ and $P(R_j=1|Z=i)$ that define the CPTs of the belief network. As shorthand, let

$$\rho_{it} \;=\; P\left(Z=i \,\middle|\, \left\{R_j=r_j^{(t)}\right\}_{j\in\Omega_t}\right)$$

denote the probabilities computed in the E-step of the algorithm. Also, let T denote the number of students. Show that the EM updates are given by

$$\begin{aligned} P(Z=i) \;&\leftarrow\; \frac{1}{T}\sum_{t=1}^{T}\rho_{it}, \\ P(R_j=1|Z=i) \;&\leftarrow\; \frac{\sum_{\{t|j\in\Omega_t\}}\rho_{it}\,I\left(r_j^{(t)},1\right)+\sum_{\{t|j\notin\Omega_t\}}\rho_{it}\,P(R_j=1|Z=i)}{\sum_{t=1}^{T}\rho_{it}}. \end{aligned}$$

(e) **Implementation**

Download the files *hw8_probZ_init.txt* and *hw8_probRgivenZ_init.txt*, and use them to initialize the probabilities $P(Z=i)$ and $P(R_j=1|Z=i)$ for a model with $k=4$ types[1] of movie-goers. Run 64 iterations of the EM algorithm, computing the (normalized) log-likelihood

$$\mathcal{L} \;=\; \frac{1}{T}\sum_{t=1}^{T}\log P\left(\left\{R_j=r_j^{(t)}\right\}_{j\in\Omega_t}\right)$$

at each iteration. Does your log-likelihood increase (i.e., become less negative) at each iteration? Fill in a completed version of the following table, using the already provided entries to check your work:

iteration	log-likelihood $\mathcal{L}$
0	-23.6819
1	-14.3421
2	
4	
8	
16	-11.6822
32	
64	

[1]There is nothing special about these initial values or the choice of $k=4$, and you should feel free to experiment with other choices.

(f) **Personal movie recommendations**

Find your student PID in *hw8_studentPIDs.txt* to determine the row of the ratings matrix that stores your personal data. Compute the posterior probability in part (c) for this row from your trained model, and then compute your *expected* ratings on the movies *you haven't yet seen*:

$$P\left(R_\ell = 1 \,\middle|\, \left\{R_j = r_j^{(t)}\right\}_{j \in \Omega_t}\right) \;=\; \sum_{i=1}^{k} P\left(Z = i \,\middle|\, \left\{R_j = r_j^{(t)}\right\}_{j \in \Omega_t}\right) P(R_\ell = 1 | Z = i) \quad \text{for } \ell \notin \Omega_t.$$

Print out the list of these (unseen) movie sorted by their expected ratings. Does this list seem to reflect your personal tastes better than the list in part (a)? Hopefully it does (although our data set is obviously *far* smaller and more incomplete than the data sets at companies like Netflix or Amazon).

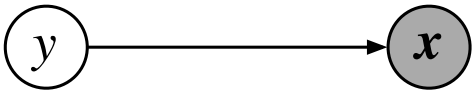
Note: if you didn't complete the survey in time, then you will need to hard-code your ratings in order to answer this question.

(g) **Source code**

Turn in a hard-copy printout of your source code for all parts of this problem. As usual, you may program in the language of your choice.

8.2 Mixture model decision boundary

Consider a multivariate Gaussian mixture model with two mixture components. The model has a hidden binary variable $y \in \{0, 1\}$ and an observed vector variable $\vec{x} \in \mathcal{R}^d$, with graphical model:



$P(y=i) \;=\; \pi_i$ $\qquad$ $P(\vec{x}|y=i) \;=\; (2\pi)^{-\frac{d}{2}}|\Sigma_i|^{-\frac{1}{2}}e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T\Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)}$

The parameters of the Gaussian mixture model are the prior probabilities π_0 and π_1, the mean vectors $\vec{\mu}_0$ and $\vec{\mu}_1$, and the covariance matrices Σ_0 and Σ_1.

(a) Compute the posterior distribution $P(y=1|\vec{x})$ as a function of the parameters $(\pi_0, \pi_1, \vec{\mu}_0, \vec{\mu}_1, \Sigma_0, \Sigma_1)$ of the Gaussian mixture model.

(b) Consider the special case of this model where the two mixture components share *the same* covariance matrix: namely, $\Sigma_0 = \Sigma_1 = \Sigma$. In this case, show that your answer from part (a) can be written as:

$$P(y = 1|\vec{x}) = \sigma(\vec{w} \cdot \vec{x} + b) \quad \text{where} \quad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

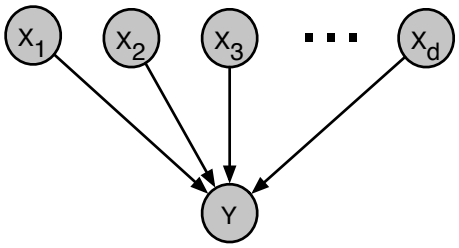
As part of your answer, you should express the parameters $(\vec{w}, b)$ of the sigmoid function explicitly in terms of the parameters $(\pi_0, \pi_1, \vec{\mu}_0, \vec{\mu}_1, \Sigma)$ of the Gaussian mixture model.

(c) Assume again that $\Sigma_0 = \Sigma_1 = \Sigma$. Note that in this case, the decision boundary for the mixture model reduces to a hyperplane; namely, we have $P(y = 1|\vec{x}) = P(y = 0|\vec{x})$ when $\vec{w} \cdot \vec{x} + b = 0$. Let k be a positive integer. Show that the set of points for which

$$\frac{P(y = 1|\vec{x})}{P(y = 0|\vec{x})} \;=\; k$$

is also described by a hyperplane, and find the equation for this hyperplane. (These are the points for which one class is precisely k times more likely than the other.) Of course, your answer should recover the hyperplane decision boundary $\vec{w} \cdot \vec{x} + b = 0$ when $k = 1$.

8.3 Gradient ascent versus EM



Consider the belief network shown above, with binary random variables $X_i \in \{0,1\}$ and $Y \in \{0,1\}$. Suppose that the conditional probability table (CPT) for node Y has the form

$$P\left(Y=1|\vec{X}=\vec{x}\right) \;=\; 1-\exp\left(-\vec{v}\cdot\vec{x}\right),$$

which is parameterized in terms of nonnegative weights $v_i \geq 0$, one for each parent X_i.

(a) **Log (conditional) likelihood**

Consider a fully observed data set of i.i.d. examples $\{\vec{x}_t, y_t\}_{t=1}^T$ where for each example $\vec{x}_t \in \{0,1\}^d$ and $y_t \in \{0,1\}$. Compute the log (conditional) likelihood,

$$\mathcal{L}(\vec{v}) \;=\; \sum_{t=1}^{T} \log P(y_t|\vec{x}_t)\,,$$

in terms of the weights v_i. You will want to simplify your expression as much as possible for the next part of this question.

(b) **Gradient**

As shorthand in this problem, let $\rho_t = P(Y=1|\vec{x}_t)$. The gradient of the conditional log-likelihood from part (a) is given by one of the following expressions:

$$\text{(i)} \quad \frac{\partial \mathcal{L}}{\partial \vec{v}} \;=\; \sum_{t=1}^{T} (y_t - \rho_t)\,\vec{x}_t$$

$$\text{(ii)} \quad \frac{\partial \mathcal{L}}{\partial \vec{v}} \;=\; \sum_{t=1}^{T} \left[\frac{y_t - \rho_t}{\rho_t(1-\rho_t)}\right]\vec{x}_t$$

$$\text{(iii)} \quad \frac{\partial \mathcal{L}}{\partial \vec{v}} \;=\; \sum_{t=1}^{T} \left(\frac{y_t - \rho_t}{\rho_t}\right)\vec{x}_t$$

Compute the gradient, and show that your final answer matches (the correct) one of these expressions.

(c) **Noisy-OR**

With a simple transformation, we can express the conditional probability model for this problem in a more familiar form. Let

$$v_i \;=\; -\log(1-p_i),$$

where the new parameters $p_i \in [0,1]$ lie in the unit interval. Use this transformation to express the conditional probability

$$P(y{=}1|\vec{x}) \;=\; 1 - e^{-\vec{v}\cdot\vec{x}}$$

in terms of the parameters p_i. Your answer should reproduce the result for a noisy-OR model with parameters p_i.

(d) **Chain rule**

We can perform gradient ascent in either the weights v_i or the noisy-OR parameters p_i. Use the transformation in part (c) and the chain rule from calculus to show that:

$$\frac{\partial \mathcal{L}}{\partial p_i} \;=\; \left(\frac{1}{1-p_i}\right)\frac{\partial \mathcal{L}}{\partial v_i}.$$

(e) **Gradient ascent versus EM**

Consider the update rule for gradient ascent (GA) in the noisy-OR parameters $p_i \in [0,1]$. This update takes the simple form:

$$p_i \;\leftarrow\; p_i \;+\; \eta\left(\frac{\partial \mathcal{L}}{\partial p_i}\right). \qquad (\mathbf{GA})$$

In the homework, you also derived an EM update for the noisy-OR parameters p_i in this belief network. It had the following form:

$$p_i \;\leftarrow\; \frac{p_i}{T_i}\left[\sum_{t=1}^{T}\left(\frac{y_t x_{it}}{\rho_t}\right)\right], \qquad (\mathbf{EM})$$

where on the right hand side of this update we have used shorthand for the conditional probability $\rho_t = P(y{=}1|\vec{x}_t)$ and the sums of inputs $T_i = \sum_{t=1}^{T} x_{it}$.

How are these two different updates related? In this problem you will prove a surprising result. Suppose that at each GA-update, we use a specialized learning rate η_i for each parameter p_i, and that we set

$$\eta_i \;=\; \frac{p_i(1-p_i)}{T_i}.$$

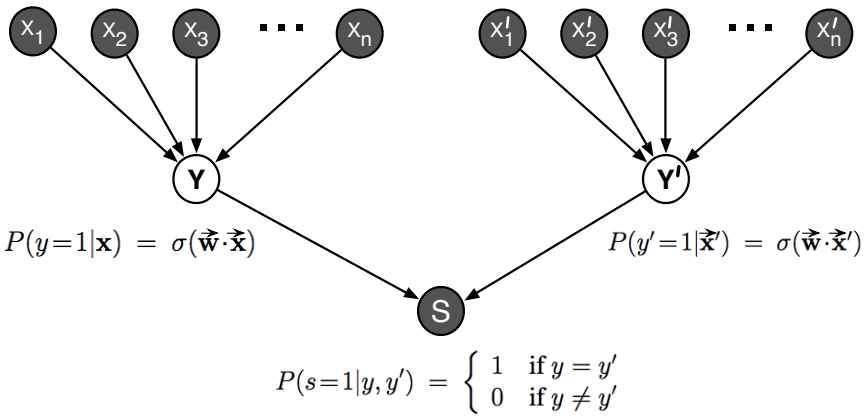
For this particular *adaptive* choice of the learning rates η_i (which *rescale* the individual components of the gradient), show that the resulting **GA**-update is identical to the **EM**-update.

8.4 Similarity learning with logistic regression

In class we introduced logistic regression as a probabilistic model of *classification* that mapped vector inputs $\vec{x} \in \Re^n$ to binary outputs $y \in \{0, 1\}$. We also showed how to train the model from labeled examples $\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_T, y_T)\}$; this was done by maximizing the log (conditional) likelihood $\sum_{t=1}^{T} \log P(y_t|\vec{x}_t)$ of correct classification.

Sometimes it is necessary to train models of logistic regression from examples that are not so explicitly labeled. An interesting case is the problem of *similarity learning*. In this problem we are told only whether *pairs* of inputs belong to the same class—that is, whether or not they share the same label—but not the actual labels of the inputs themselves.

The figure below shows a belief network for this problem. The observed inputs $\vec{x}, \vec{x}' \in \Re^n$ have unknown (hidden) labels $y, y' \in \{0, 1\}$, and the observed variable $s \in \{0, 1\}$ indicates whether or not $y = y'$. Here $\sigma(z) = [1 + e^{-z}]^{-1}$ denotes the sigmoid function. Note that the same weight vector $\vec{w} \in \Re^n$ parameterizes the logistic regressions for $P(y|\vec{x})$ and $P(y'|\vec{x}')$.



In this problem you will be guided to derive an EM algorithm for models of this form. (The virtue of the EM algorithm for this model is that its M-step consists of ordinary logistic regressions.)

(a) **Inference for similar examples**

Show how to compute the posterior probability $P(y = 1, y' = 1|\vec{x}, \vec{x}', s = 1)$ in terms of the weight vector $\vec{w}$ and the inputs $\vec{x}, \vec{x}'$. Simplify your answer as much as possible.

(b) **Inference for dissimilar examples**

Show how to compute the posterior probability $P(y = 1, y' = 0|\vec{x}, \vec{x}', s = 0)$ in terms of the weight vector $\vec{w}$ and the inputs $\vec{x}, \vec{x}'$. Simplify your answer as much as possible.

(c) **E-Step**

The posterior probabilities needed for the EM algorithm can be derived (fairly easily) from the ones that you just computed. Choose the correct answer for each of the following.

The posterior probability $P(y = 1|\vec{x}, \vec{x}', s=1)$ is equal to:

(a) $P(y=1, y'=1|\vec{x}, \vec{x}', s=1)$
(b) $P(y=1, y'=0|\vec{x}, \vec{x}', s=0)$
(c) $1 - P(y=1, y'=1|\vec{x}, \vec{x}', s=1)$
(d) $1 - P(y=1, y'=0|\vec{x}, \vec{x}', s=0)$

The posterior probability $P(y' = 1|\vec{x}, \vec{x}', s=1)$ is equal to:

(a) $P(y=1, y'=1|\vec{x}, \vec{x}', s=1)$
(b) $P(y=1, y'=0|\vec{x}, \vec{x}', s=0)$
(c) $1 - P(y=1, y'=1|\vec{x}, \vec{x}', s=1)$
(d) $1 - P(y=1, y'=0|\vec{x}, \vec{x}', s=0)$

The posterior probability $P(y = 1|\vec{x}, \vec{x}', s=0)$ is equal to:

(a) $P(y=1, y'=1|\vec{x}, \vec{x}', s=1)$
(b) $P(y=1, y'=0|\vec{x}, \vec{x}', s=0)$
(c) $1 - P(y=1, y'=1|\vec{x}, \vec{x}', s=1)$
(d) $1 - P(y=1, y'=0|\vec{x}, \vec{x}', s=0)$

The posterior probability $P(y' = 1|\vec{x}, \vec{x}', s=0)$ is equal to:

(a) $P(y=1, y'=1|\vec{x}, \vec{x}', s=1)$
(b) $P(y=1, y'=0|\vec{x}, \vec{x}', s=0)$
(c) $1 - P(y=1, y'=1|\vec{x}, \vec{x}', s=1)$
(d) $1 - P(y=1, y'=0|\vec{x}, \vec{x}', s=0)$

(d) **Log-likelihood**

Consider a data set $\{(\vec{x}_1, \vec{x}'_1, s_1), \ldots, (\vec{x}_T, \vec{x}'_T, s_T)\}$ that consists of pairs of inputs $(\vec{x}_t, \vec{x}'_t)$ and judgments of similarity $s_t \in \{0, 1\}$. Derive an explicit expression for the log-likelihood

$$\mathcal{L}(\vec{w}) \;=\; \sum_t \log P(s_t|\vec{x}_t, \vec{x}'_t)$$

in terms of the weight vector $\vec{w}$ and the examples in the data set. *Hint:* You will need to sum over (allowed) values of the hidden variables (y, y').

(e) **M-step**

As shorthand, let $\bar{y}_t = P(y = 1|\vec{x}_t, \vec{x}'_t, s_t)$ and $\bar{y}'_t = P(y' = 1|\vec{x}_t, \vec{x}'_t, s_t)$ denote the posterior probabilities computed in the E-step of the EM algorithm. The M-step for this model takes the following form:

$$\vec{w} \leftarrow \operatorname*{argmax}_{\vec{w}} \left\{ \sum_t \Big[\bar{y}_t \log \sigma(\vec{w} \cdot \vec{x}_t) + (1 - \bar{y}_t) \log \sigma(-\vec{w} \cdot \vec{x}_t) + \bar{y}'_t \log \sigma(\vec{w} \cdot \vec{x}'_t) + (1 - \bar{y}'_t) \log \sigma(-\vec{w} \cdot \vec{x}'_t) \Big] \right\}$$

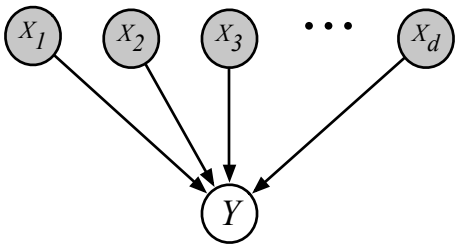
It is generally much simpler to maximize the right hand side of this expression (which should look familiar to you) than the log-likelihood $\mathcal{L}(\vec{w})$ in part (d).

Let $\eta > 0$ denote a small learning rate. Derive the iterative update rule to perform the above maximization using *gradient ascent*. Complete the expression below with your final answer.

$$\vec{w} \leftarrow \vec{w} + \eta \left\{ \sum_t \Big[\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Big] \right\}$$

Note: compute the gradient for the maximization shown above, **not** the gradient of your answer to part (d) of this problem (which is much more complicated).

8.5 Logistic regression across time

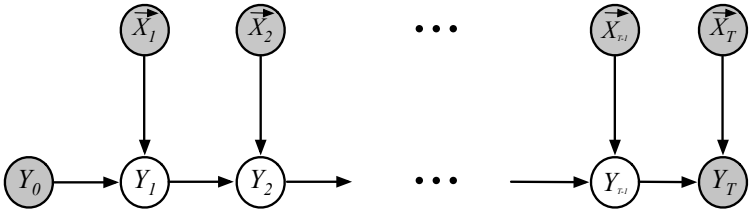


In class we studied a simple model of logistic regression represented by the belief network shown above. Predictions were modeled by the conditional probability

$$P(Y = 1|\vec{x}) \;=\; \sigma(\vec{w} \cdot \vec{x}),$$

where $\sigma(z) = [1 + e^{-z}]^{-1}$ was the sigmoid function, and the model as a whole was parameterized by the weight vector $\vec{w} \in \Re^d$. Below we show a more compact way to draw this same belief network, with all the inputs represented by a single vector-valued node.

In lecture we considered this model for *i.i.d.* data. In this problem you will explore a model for data that does not satisfy this assumption. Since the *i.i.d.* assumption does not hold, we will try to model the dependence across time explicitly.



$$P(Y_t{=}1 \,|\, \vec{x}_t, Y_{t-1}{=}0) \;=\; \sigma(\vec{w}_0 \cdot \vec{x}_t)$$
$$P(Y_t{=}1 \,|\, \vec{x}_t, Y_{t-1}{=}1) \;=\; \sigma(\vec{w}_1 \cdot \vec{x}_t)$$

Consider the belief network shown above for predicting a time series of binary outputs $y_t \in \{0, 1\}$ from a time series of inputs $\vec{x}_t \in \Re^d$, where $t = 1, 2, \ldots, T$. For times $t \geq 1$, the belief network has the sigmoid conditional probability tables (CPTs):

$$
\begin{aligned}
P(Y_t{=}1 \,|\, \vec{x}_t, Y_{t-1}{=}0) &= \sigma(\vec{w}_0 \cdot \vec{x}_t), \\
P(Y_t{=}1 \,|\, \vec{x}_t, Y_{t-1}{=}1) &= \sigma(\vec{w}_1 \cdot \vec{x}_t),
\end{aligned}
$$

where $\vec{w}_0 \in \Re^d$ and $\vec{w}_1 \in \Re^d$ are different weight vectors. In other words, we predict y_t from $\vec{x}_t$ differently depending on the value of y_{t-1}: if $y_{t-1} = 0$, then we use $\vec{w}_0$ in the sigmoid CPT for predicting y_t, and if $y_{t-1}{=}1$, then we use $\vec{w}_1$ in the sigmoid CPT.

In addition, imagine for this application that only two outputs are observed, an initial one at time $t = 0$ and a final one at time $t = T$. These observations are indicated as usual by shaded nodes.

(a) **Likelihood**

As shorthand, let $\alpha_{it} = P(Y_t = i | y_0, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_t)$ where $i \in \{0, 1\}$. Derive an efficient recursion for computing the probabilities $\alpha_{j,t+1}$ in terms of the probabilities α_{it} at the previous time step as well as the model parameters $\vec{w}_0$ and $\vec{w}_1$. *Show your work for full credit*, justifying each step.

(b) **Most likely hidden states**

Let $i \in \{0, 1\}$, and denote the log-probability of the most likely set of hidden states up to time t, such that $y_t = i$, by

$$
\ell^*_{it} \;=\; \max_{y_1, \ldots, y_{t-1}} \Big[\log P(y_1, y_2, \ldots, y_t{=}i | y_0, \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_t) \Big].
$$

Derive an efficient recursion for computing the log-probabilities $\ell^*_{j,t+1}$ from those at the previous time step as well as the model parameters $\vec{w}_0$ and $\vec{w}_1$. *Show your work for full credit*, justifying each step.

(c) **Prediction** (2 pts)

To use this model for prediction, given parameters $\vec{w}_0$ and $\vec{w}_1$, we must be able to compute the binary outputs $\{y^*_t\}_{t=1}^{T-1}$ that maximize the posterior probability

$$
P(y_1, y_2, \ldots, y_{T-1} | y_0, y_T, \vec{x}_1, \ldots \vec{x}_T).
$$

Show that your results from part (b) can be used to perform this computation; in particular, complete the two blocks of pseudocode sketched below.

for $t = 1$ to $T{-}1$
  for $j = 0$ to 1
    $\Phi_{t+1}(j) \;=\; \text{argmax}_{i \in \{0,1\}} \Big[$ $\qquad\qquad\qquad\qquad$ $\Big]$

for $t = T{-}1$ to 1
  $y^*_t \;=$