University of Pittsburgh
CS2750 Machine Learning
Professor Milos Hauskrecht

Handout 8
March 24, 2022

Problem assignment 8

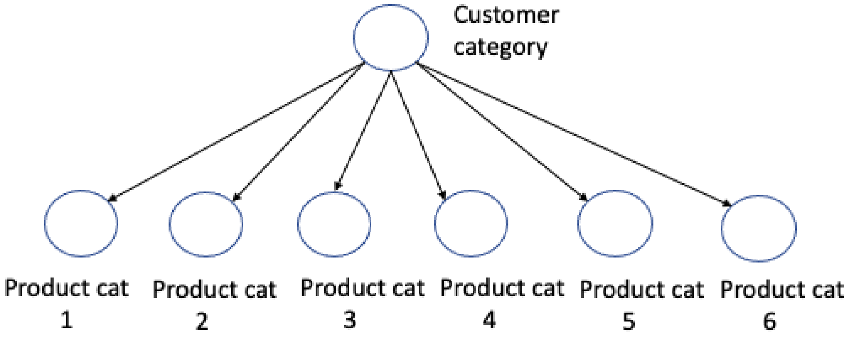
Due: Thursday, March 31, 2022

Problem 1. Customer profiling and predictions with a latent variable model and EM

In this problem we use the the dataset of consumers preferences for different product brands. The dataset consists of 6 attributes corresponding to individual product categories, the values correspond to different brands in respective categories. For the sake of simplicity, we use numerical labels to distinguish the brands. We assume that each product can appear only in one category. Each customer corresponds to one entry (row) in the dataset. The number of brand products in each product category is listed in the following table.

Product category	1	2	3	4	5	6
Number of brands	5	3	3	4	5	4

The dataset we have available is not complete and some entries reflecting the preferences of the customer towards different brands are missing. The label 0 indicates the missing value. Despite the missingness, our goal is to use the data for prediction. We will use the Naive Bayes model to achieve this goal. We construct the Naive Bayes model in the following way: the root node of the model represents groups of customers (not observable), the leaves are product categories with values corresponding to different brand names. The model is shown in the figure below. The assumption introduced by the Naive Bayes model is that the preferences of the customer towards brands in different categories are conditionally independent given the group the customer belongs to. The relations between the group and any product category are stochastic and may vary for different individuals. The key idea here is that identifying the customers' group helps us to make the prediction about the product preference.



Part a. In the first part we need to learn the parameters of the Naive Bayes model with the hidden class variable and missing values in the dataset. We assume that there are four different groups of customers. Thus the hidden root variable can have 4 different values.

To learn the model we want to find the ML estimate of parameters of the model given data. We will use EM algorithm to obtain them. What follows is a description of the EM for this case that will help you to implement the procedure. You will need to **turn in your EM code** in the file *NB_EM_learning.py*. Briefly, your code in *NB_EM_learning.py* should load the training data in file *training_data.csv*, implement the EM algorithm and save the model and its parameters in the format of your choice so that you can load it to implement the prediction related inferences in Part b.

EM for the Naive Bayes model

Different customers types in the system are represented through a class variable C. We assume there is a fixed number of classes $M = 4$. Let $X = \{X_1, X_2, \cdots, X_n\}$ be a set of relevant attribute variables. Let $D = \{D^1, D^2, \cdots, D^N\}$ be a data set such that each D^l is a sample of X with possibly missing values. For example D^l can consist of $\{X_1^l = 3, X_3^l = 1\}$ or $\{X_1^l = 4, X_3^l = 1, X_4^l = 1\}$.

Our goal is find the set of parameters Θ maximizing the likelihood of D, that is, $p(D|\Theta)$. In the case of the Naive Bayes model parameters in Θ correspond to

- a set of priors on classes $\pi_1, \cdots, \pi_M$ (s.t. $\sum_{j=1}^{M} \pi_j = 1$).
- a set of parameters θ_{ijk} representing components of conditional distributions $p(X_i = k|C = j)$.

Let H represents the set of hidden variables. As discussed during the lecture hidden variables correspond to the different indicator variables. These represent values the class variable C can be assign to for each entry in the dataset and values corresponding to missing attribute values in the dataset. For example, if the value for X_2^l is missing in D^l we capture this using variables in H.

The optimization of the likelihood $p(D|\Theta)$ can be carried in a standard way in terms of the loglikelihood optimization. There are various ways to approach this optimization problem. We apply the EM algorithm.

In the EM algorithm we define:

$$Q(\Theta|\Theta') = \sum_{j=1}^{M} N'_{c=j} \log \pi_j + \sum_{i=1}^{d} \sum_{k=1}^{|X_i|} \sum_{j=1}^{M} N'_{C=j,X_i=k} \log \theta_{ijk}$$

where:

$$N'_{C=j} = \sum_{l=1}^{N} E_{H|D,\Theta'}(\delta_{C^l=j})$$

and

$$N'_{C=j,X_i=k} = \sum_{l=1}^{N} E_{H|D,\Theta'}(\delta_{C^l=j,X_i^l=k}).$$

The expectation terms are:

$$E_{H|D,\theta'}(\delta_{C^l=j}) = p(C^l = j|D^l, \theta')$$

$$E_{H|D,\theta'}(\delta_{C^l=j,X_i^l=k}) = \begin{cases} p(C^l = j|D^l, \Theta') & \text{if } X_i^l = k \text{ is in } D \\ p(C^l = j|D^l, \Theta')p(X_i^l = k|C^l = j, D^l, \Theta') & \text{if } X_i^l \text{ is missing in } D \\ 0 & \text{otherwise} \end{cases}$$

In EM we maximize $Q(\Theta|\Theta')$. The key advantage here is that this maximization can be carried in the closed form, leading to new parameter estimates:

$$\pi_j = \frac{N'_{C=j}}{\sum_{j=1}^{M} N'_{C=j}}$$

$$\theta_{ijk} = \frac{N'_{C=j,X_i=k}}{\sum_{k=1}^{|X_i|} N'_{C=j,X_i=k}}.$$

Additional tasks:

- The EM algorithm may take a while to converge as parameters can change very slowly at the end. Thus, one typically stops the EM iterations once only small changes in Q functions are observed. Choose a small constant (0.001) to stop the algorithm.
- Try to run the EM algorithm multiple times on different sets of initial parameters. Please avoid using 0 parameters while initializing the parameters. Analyze and explain your observations in your report.

Part b. Once we learn the model we can use it to support predictions. Write and submit *NB_test.py* file that loads the NB model learned in Part a, testing set in file *testing_data.csv* and predicts the value of attribute 6 for each customer from the testing set. Please note that there are missing values also in the testing set. For the prediction use the ML estimates of parameters Θ obtained in Part a. To do the prediction simply compute the probability $p(x_6^l = k|D_{mod}^l, \Theta_{ML})$ of attribute 6 having value k for each customer in the test dataset, and do it for all possible values the attribute can take. D_{mod}^l is the data entry for the customer l with attribute 6 removed. As a prediction output select the value with the highest probability.

After the prediction please compute the confusion matrix and mean misclassification error for predicting attribute 6 based on values for other 5 attributes. To compute the confusion matrix and the misclassification error compare the predicted and actual value of the attribute. Turn in the program code in file *NB_test.py*

Part c. Assume a predictor in which the value of attribute 6 is selected uniformly at random. What is the worst case mean misclassification error in this situation? Compare the worst case result to the result obtained through the Naive Bayes model.

Extra credit. Run the randomized uniform predictor on the test set. Is the result obtained by the Naive Bayes predictor statistically significantly different from the randomized predictor at significance level 0.05?