Probabilistic Graphical Models - Tutorial 9

Fernando García Sanz

March 2, 2020

The purpose of this tutorial is to provide a basic understanding of Latent Dirichlet Allocation by using it to implement a topic clustering mechanism applying to several texts contained in a dataset. The implementations done in the provided skeleton, as well as the obtained results, can be found below:

1 Provided Texts

The first provided texts can be classified by 4 different categories, which are: *Religion*, *Cars*, *Hockey* and *Space*.

Once the model has been trained applying the *expectation-maximization* mechanism, it is possible to load the different texts, analyze their content, and given the probability of each word to be related to a specific topic, return a final outcome of which one is the probability of a text to be related to a determined topic.

In this case, the relevant words per topic were classified in the following way:

- Top words for topic 0 :
  ['space' 'nasa' 'year' 'post' 'know' 'program' 'group' 'gov' 'list' 'like' 'use' 'new' 'world' 'data' 'member' 'news' 'include' 'technology' 'good' 'research']

- Top words for topic 1 :
  ['god' 'say' 'people' 'jesus' 'hell' 'know' 'christian' 'believe' 'think' 'thing' 'time' 'religion' 'die' 'father' 'life' 'question' 'faith' 'spirit' 'son' 'come']

- Top words for topic 2 :
  ['car' 'launch' 'use' 'satellite' 'think' 'space' 'mission' 'time' 'orbit' 'project' 'access' 'uiuc' 'say' 'post' 'year' 'saturn' 'dealer' 'nntp' 'net' 'good']

- Top words for topic 3 :
  ['team' 'flyer' 'play' 'game' 'hockey' 'ca' 'gm' 'year' 'season' 'player' 'point' 'goal' 'leaf' 'city' 'good' 'win' 'think' 'post' 'nntp' 'record']

As can be seen, most of the words enclosed in each set are related to a specific topic. Topic 0 seems to be related to space, topic 1 to religion, topic 2 to cars and topic 3 to hockey. Nevertheless, it is necessary to consider that some words can be in texts of several topics, and therefore, be considered as relevant words for some specific topic, even though they don't have any strong relation to it.

When performing the testing stage with different documents, the results are the followings:

Estimated mixture for document 15 is:
topic 0 : 0.9688560775744404
topic 1 : 0.0002762111760749518
topic 2 : 0.00027938964447751433
topic 3 : 0.030588321605007015

Which has the following text:
From: buenneke@monty.rand.org (Richard Buenneke)
Subject: White House outlines options for station, Russian cooperation
X-Added: Forwarded by Space Digest
Organization: [via International Space University]
Original-Sender: isu@VACATION.VENARI.CS.CMU.EDU
Distribution: sci
Lines: 71

As it can be seen, the main topic, with a 96.9% of probability, is topic 0, the one related to the space. It makes sense since the content of the text is about the cooperation with Russia for using their assets in the redesign of the spacial station.

Nevertheless, the results are not always this satisfactory. Depending on the structure and content of the text, it might happen that the model is not able to properly find the topic the text is related to (maybe the probabilities are close to 50% in two different topics), that the topic of the text is not among the four considered...

2 Extra Texts

Besides the texts used in the previous section, we are provided different texts which can be used to train the model with different topics. The results shown in this section are obtained using the Associated Press documents dataset, among those provided:

In this case, the relevant words per topic were classified in the following way:

- Top words for topic 0 :
  ['soviet' 'bush' 'government' 'president' 'party' 'gorbachev' 'union' 'trade' 'committee' 'new' 'official' 'israel' 'united' 'congress' 'people' 'dukakis' 'american' 'leader' 'house' 'administration']

- Top words for topic 1 :
  ['new' '000' 'state' 'reported' 'officials' 'states' 'day' 'children' 'wednesday' 'people' 'air' 'high' 'news' 'california' 'iraq' 'york' 'central' 'southern' 'city' 'americans']

- Top words for topic 2 :
  ['police' 'year' 'people' 'city' 'man' 'years' '000' 'court' 'state' 'new' 'family' 'mrs' 'day' 'attorney' 'old' 'work' 'saturday' 'barry' 'died' 'used']

- Top words for topic 3 :
  ['percent' 'year' 'new' 'prices' 'company' 'million' 'billion' 'oil' 'bank' 'rose' 'rate' 'price' 'market' 'month' 'gold' 'report' 'dollar' 'economy' 'thursday' 'york']

It can be seen that the 4 different topics obtained now are related to (according to my point of view): *USA - Israel - Soviet Union affairs*, *Police Reports in the US*, *Judicial Affairs* and *Economy*.

Using now the MoodyLyrics dataset, we should be able to get the main words employed in each one of the 4 found topics:

- Top words for topic 0 :
  ['away' 'easy' 'god' 'lord' 'joy' 'hey' 'walking' 'goes' 'free' 'gun' 'pain' 'inside' 'run' 'sun' 'blind' 'need' 'huh' 'lost' 'living' 'sexy']

- Top words for topic 1 :
  ['home' 'girl' 'good' 'burn' 'wanna' 'right' 'say' 'think' 'feel' 'tonight' 'hate' 'man' 'night' 'let' 'little' 'change' 'heart' 'fi' 'ooh' 'loving']

- Top words for topic 2 :
  ['away' 'day' 'fame' 'gonna' 'shy' 'angel' 'life' 'good' 'people' 'evil' 'eye' 'shot' 'seen' 'home' 'hey' 'need' 'ooh' 'turn' 'wonder' 'soul']

- Top words for topic 3 :
  ['lonely' 'war' 'need' 'say' 'tell' 'let' 'bed' 'ooh' 'feel' 'mind' 'gone' 'look' 'heart' 'happy' 'chance' 'start' 'right' 'long' 'good' 'pain']

These last topics are theoretically representing the emotions each song exhibits and, therefore, it would be possible to classify a song into one of those four moods by means of its lyrics.

Code Implementation

```
#Imports
import numpy as np
import scipy.special as special
import scipy.optimize
import time

#diGamma func from scipy, use this in your code!
diGamma = special.digamma

#Function definitions for maximizing the VI parameters. This will later be completed by you.
def maxVIParam(phi, gamma, B, alpha, M, k, Wd, eta):

    for d in range(M):
        N = len(Wd[d])
        #Initialization of vars, as shown in E-step.
        phi[d] = np.ones((N,k))*1.0/k
        gamma[d] = np.ones(k)*(N/k) + alpha
        converged = False
        j = 0 #you can use this to print the update error to check your code in the beginning with something like:
        '''if(j%10==0 and d==0):
                    print("u e: ", updateError)'''
        #YOUR CODE FOR THE E-STEP HERE
        prev_gamma = gamma[d]
        while(not converged):
            phi[d] = B[:, Wd[d]].T * np.exp(diGamma(gamma[d]))
            row_sums = phi[d].sum(axis=1)
            phi[d] = phi[d] / row_sums[:, np.newaxis]
            gamma[d] = alpha + phi[d].sum(axis=0)
            if np.sum(abs(prev_gamma - gamma[d])) < eta:
                converged = True
            else:
                prev_gamma = gamma[d]

    return gamma, phi

#Function definitions for maximizing the B parameter. This will later be completed by you.
def MaxB(B, phi, k, V, M, Wd):

    #YOUR CODE FOR THE M-STEP HERE
    B = np.zeros(B.shape)
    for d in range(M):
        for n in range(len(Wd[d])):
            B[:, Wd[d][n]] += phi[d][n]


    return B
```