TP-3-SD-TSIA-214-Bejaoui-Ahmed-Mejri-Aymen

June 12, 2018

1 TP 3 : Text segmentation using Hidden Markov Models BEJAOUI AHMED - MEJRI AYMEN

1.0.1 Question 1 : Give the value of the vector of the initial probabilitiesű

Etant donnée que l'état initial de départ est l'entête nous constatons donc que le vecteur d'initialisation :

$$\pi = [1, 0]$$

1.0.2 Question 2 : What is the probability to move from state 1 to state 2 ? What is the probability to remain in state 2 ? What is the lower/higher probability ? Try to explain why

$$A = \begin{bmatrix} 0.999218078035812 & 0.000781921964187974 \\ 0 & 1 \end{bmatrix}$$

Disposant de la matrice A , nous pouvons conclure que la probabilité de passer de l'état 1 vers l'état 2 correspond à $A_{12} = 0.000781921964187974$ et la probabilité de rester dans l'état 2 est $A_{22} = 1$ la plus faible probabilité est est p(2 | 1) = 0 ce qui est tout à fait prévisible parce que nous ne pouvons pas passer à l'état 1 header lorsque nous nous trouvons à l'état 2 (corps du mail) . La plus grande probabilité est p(2 | 2) = 1 c'est à dire lorsque nous sommes dans l'état 2 (corps du mail), nous y restons

1.0.3 Question 3 : What is the size of the corresponding matrix ?

La taille de la matrice est 256x2 . En effet nous avons 256 caractères possibles dans le code ASCII donc nous avons 256 lignes dans la matrice et nous disposons seulement de 2 états possible qui sont s=1 et s=2 donc nous avons 2 colonnes .

1.0.4 Question 4 : Print the track and present and discuss the results obtained on mail11.txt to mail30.txt

```
In [1]: from glob import glob
        import os.path as op
        import numpy as np

        filenames = sorted(glob(op.join('dat','*.dat')))
        texts_pred = [np.loadtxt(f,dtype=int) for f in filenames]
        len(texts_pred)
```

Out[1]: 30

Initialisation :

```
In [2]: A=np.zeros((2,2))
        A[0,0]=0.999218078035812; A[0,1]=0.000781921964187974 ; A[1,0]=0 ; A[1,1]=1
        B = np.loadtxt('P.text', dtype=float)
        pi=[1,0]
        q=[0,1]
```

La Fonction Viterbi :

```
In [3]: def Viterbi_function(A,B,pi,O,q):
            """
            Input :

            A : Matrice de transition des différents états
            B : Matrice de transition des observations sachant les états
            pi: vecteur d'initialisation
            O : vecteur des observations
            q : Vecteur des différents états possibles

            Output:

            qhat : la séquence d'état prédite.

            """

            dl=np.zeros((len(q),len(O)))
            phi=np.zeros((len(q),len(O)))

            inter=0
            qhT=0

            for i in range(len(q)):
                dl[i,0]=B[O[0],i]*pi[i]

            for t in range(len(O)-1):
                for j in range(len(q)):
                    inter=np.max([A[i,j]*dl[i,t] for i in range(len(q))])
                    dl[j,t+1]=inter*B[O[t+1],j]
                    phi[j,t+1]=np.argmax([A[i,j]*dl[i,t] for i in range(len(q))])
                dl[:,t+1] = dl[:,t+1]*(10**(-1*(int(np.log10(np.max(dl[:,t+1])))+1)))
            qh=np.array([])
            qhT=np.argmax([dl[j,len(O)-1] for j in range(len(q))])
            qh=np.append(qhT,qh)
            for i in range (len(O)-1,0,-1):
                qhT=int(qhT)
```

```
            qhT=phi[qhT,i]
            qh=np.append(qhT,qh)
            qh=np.array([int(i) for i in qh])
        return qh
```

Etude de la performance de l'algorithme:

```
In [4]: filenames = sorted(glob(op.join( 'dat', '*.txt')))
        train = [len(open(f).read()) for f in filenames if f.rstrip().endswith("h.txt")]
        texts = [np.loadtxt(f.rstrip().replace("h.txt",".dat"),dtype="int") for f in filenames
        state_pred=np.array([])
        for text in texts:
            state_pred=np.append(state_pred,np.bincount(Viterbi_function(A,B,pi,text,q))[0])

        score=np.array([])
        score=1-np.abs(train-state_pred)/state_pred

In [5]: states_pred=np.array([])
        for text in texts_pred:
            states_pred=np.append(states_pred,np.bincount(Viterbi_function(A,B,pi,text,q))[0])

In [6]: print('L\'algorithme a une précision de : ',score.mean())

L'algorithme a une précision de :  0.983381544512
```

initialisation sur un ensemble de texte de mail11 jusqu'à mail30 : nous prenons ici les deux fichiers mail11.txt et mail30.txt pour ces deux fichiers nous avons respectivement 2850 et 2249 caractères dans le header. nous avons déterminer ces résultat en regarder le texte avec wordPad.

```
In [7]: train=[2850,2249]
        state_pred=np.array([])
        O1= np.loadtxt('dat/mail11.dat',dtype=int)
        O2= np.loadtxt('dat/mail30.dat',dtype=int)
        state_pred=np.append(state_pred,np.bincount(Viterbi_function(A,B,pi,O1,q))[0])
        state_pred=np.append(state_pred,np.bincount(Viterbi_function(A,B,pi,O2,q))[0])
        score=np.array([])
        score=1-np.abs(train-state_pred)/state_pred

In [8]: print('L\'algorithme a une précision de : ',score.mean())

L'algorithme a une précision de :  0.982337278255
```

Visualisation sur un texte : Pour la visualisation du texte nous avons eu un problème avec la commande perl. Nous avons donc choisi de développer notre propre fonction.

```
In [9]: def segment(path,file_name,size=None):
            text=None
            with open(file_name) as f:
                text=f.read()
            n =np.argmax(path)
            a=0
            b=len(text)
            if size != None:
                a=max(0,n-size)
                b=min(len(text),n+size)
            print(text[a:n]+ '\n================================== coupez ici ================
            return None

In [10]: O2= np.loadtxt('dat/mail13.dat',dtype=int)
         segment(Viterbi_function(A,B,pi,O2,q),'dat/mail13.txt')
```

```
From ilug-admin@linux.ie  Thu Aug 22 16:27:21 2002
Return-Path: <ilug-admin@linux.ie>
Delivered-To: zzzz@localhost.netnoteinc.com
Received: from localhost (localhost [127.0.0.1])
        by phobos.labs.netnoteinc.com (Postfix) with ESMTP id 7A28A43F99
        for <zzzz@localhost>; Thu, 22 Aug 2002 11:27:21 -0400 (EDT)
Received: from phobos [127.0.0.1]
        by localhost with IMAP (fetchmail-5.9.0)
        for zzzz@localhost (single-drop); Thu, 22 Aug 2002 16:27:21 +0100 (IST)
Received: from lugh.tuatha.org (root@lugh.tuatha.org [194.125.145.45]) by
    dogma.slashnull.org (8.11.6/8.11.6) with ESMTP id g7MFQmZ12280 for
    <zzzz-ilug@spamassassin.taint.org>; Thu, 22 Aug 2002 16:26:48 +0100
Received: from lugh (root@localhost [127.0.0.1]) by lugh.tuatha.org
    (8.9.3/8.9.3) with ESMTP id QAA07188; Thu, 22 Aug 2002 16:25:32 +0100
Received: from moe.jinny.ie ([193.120.171.3]) by lugh.tuatha.org
    (8.9.3/8.9.3) with ESMTP id QAA07145 for <ilug@linux.ie>; Thu,
    22 Aug 2002 16:25:24 +0100
X-Authentication-Warning: lugh.tuatha.org: Host [193.120.171.3] claimed to
    be moe.jinny.ie
Received: from jlooney.jinny.ie (unknown [193.120.171.2]) by moe.jinny.ie
    (Postfix) with ESMTP id 938BD7FC46; Thu, 22 Aug 2002 16:25:23 +0100 (IST)
Received: by jlooney.jinny.ie (Postfix, from userid 500) id 4F57189D;
    Thu, 22 Aug 2002 16:25:45 +0100 (IST)
Date: Thu, 22 Aug 2002 16:25:45 +0100
From: "John P. Looney" <valen@tuatha.org>
To: linux-raid@vger.kernel.org
Cc: ilug@linux.ie
Message-Id: <20020822152545.GJ3670@jinny.ie>
Reply-To: valen@tuatha.org
Mail-Followup-To: linux-raid@vger.kernel.org, ilug@linux.ie
References: <200208172056.g7HKuHm05754@raq.iceblink.org>
    <1029624922.14769.119.camel@atherton> <20020819140815.GY26818@jinny.ie>
```

MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
In-Reply-To: <20020819140815.GY26818@jinny.ie>
User-Agent: Mutt/1.4i
X-Os: Red Hat Linux 7.3/Linux 2.4.18-3
X-Url: http://www.redbrick.dcu.ie/~valen
X-Gnupg-Publickey: http://www.redbrick.dcu.ie/~valen/public.asc
Subject: [ILUG] Re: Problems with RAID1 on cobalt raq3
Sender: ilug-admin@linux.ie
Errors-To: ilug-admin@linux.ie
X-Mailman-Version: 1.1
Precedence: bulk
List-Id: Irish Linux Users' Group <ilug.linux.ie>
X-Beenthere: ilug@linux.ie

On Mon, Aug 19, 2002 at 03:08:16PM +0100
================================= coupez ici ==============================
, John P. Looney mentioned:
> This is likely because to get it to boot, like the cobalt, I'm actually
> passing root=/dev/hda5 to the kernel, not /dev/md0.

 Just to solve this...the reason I was booting the box with
root=/dev/hda5, not /dev/md0 was because /dev/md0 wasn't booting - it
would barf with 'can't find init'.

 It turns out that this is because I was populating md0 with tar. Which
seems to have 'issues' with crosslinked files - for instance, it was
trying to make a hard link of glibc.so to hda - and failing. It was only
as I did it again with a friend present, that he spotted the errors, and
queried them. We noticed that the hard linked files just didn't exist on
the new rootfs.

 When we duplicated the filesystems with dump instead of tar, it worked
fine, I was able to tell lilo to use root=/dev/md0 and everything worked.

 Woohoo.

Kate

--
Irish Linux Users' Group: ilug@linux.ie
http://www.linux.ie/mailman/listinfo/ilug for (un)subscription information.
List maintainer: listmaster@linux.ie

1.0.5 Question 5 : How would you model the problem if you had to segment the mails in more than two parts (for example : header, body, signature) ?

On utilisera dans ce cas aussi le model HMM mais l'entrée de l'algorithme est différente : l'espace d'état sera de taille 3 qui prendra les valeurs suivantes : header(=0),body(=1),et signature(=2) et le vecteur d'initialisation est de dimention 3 qui commence toujours par le header et par conséquent $\pi = (1, 0, 0)$

Concernant la matrice de transition A, elle est de taille 3x3 et telle que $A_{13} = 0$, $A_{21} = 0$,$A_{13} = 0$ et $A_{23} = 0$

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ 0 & A_{22} & A_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

et la matrice B qui représente la matrice des probabilités $P(c|s)$ devient une matrice de taille 256x3 .

1.0.6 Question 6 : How would you model the problem of separating the portions of mail included, knowing that they always start with the character ">".

Pour cette question , nous aurons 4 etats possible qui sont : header,body_included ,body(le texte) et la signature . le vecteur d'initailisation π est de taille 4x1 :

$$\pi = [1, 0, 0, 0]$$

La matrice de transition A est de dimension 4x4 :

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{12} & 0 \\ 0 & A_{22} & A_{23} & A_{24} \\ 0 & A_{32} & A_{33} & A_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

la probabilité du caractère appartenant à l'état 3 est plus élevée par rapport aux autres états - cette information doit être incluse dans la matrice de probabilité conditionnelle B La matrice B sera de taille 256x4.