HMM para reconocimiento de vocales

18/05/2015

Contents

1 Introducción

Los modelos de HMM son modelos estadísticos que se utilizan en procesos que se suponen Markovianos donde los estados no son visibles directamente —son latentes— pero las variables influidas por estos estados si lo son. El objetivo consiste en identificar estos estados ocultos para conocer los estados, las transiciones entre los estados y las probabilidades de cada estado sobre las variables a las que influyen.

Los HMM han sido ampliamente utilizados en problemas de Procesamiento de Lenguaje Natural (NLP) sobre todo para el reconocimiento del habla (Part Of Speech, POS), etiquetado gramatical, reconocimiento de escritura manual, etc. Stamp (2012)

2 Objetivo

Identificación de vocales y consonantes a través de un modelo de HMM estimando sus parámetros.

3 Especificación del modelo

- Utilizamos HMM con el algoritmo Baum-Welch para estimar los parámetros:

1. las probabilidades inciales de los estados
2. las probabilidades de transición entre estados
3. las probabilidades de cada símbolo de pertenecer a uno de los estados

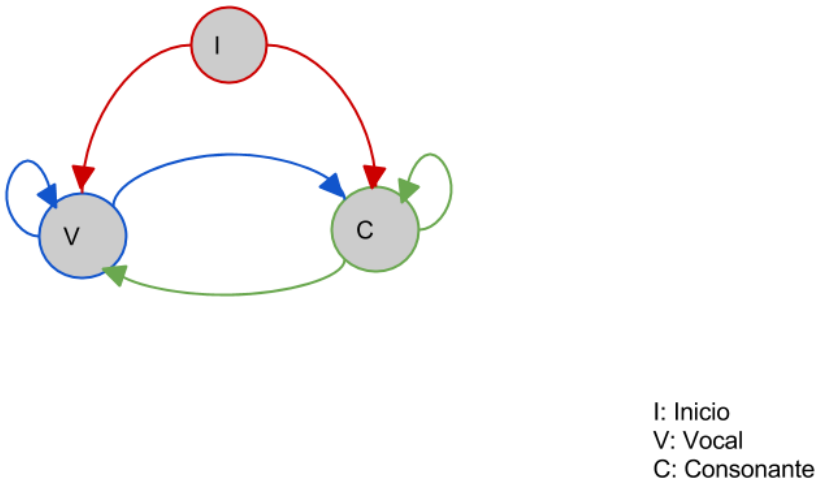

Figure 1: imagen

3.1 Algoritmo Baum-Welch

- Este algoritmo es una variante del EM visto en clase Frazzoli (2010); Bilmes (1998) . Iniciamos con un modelo sin 'conocimiento'

π = probabilidades de iniciar en cada estado

T= matriz de transición de estados

M= matriz de emisiones

$\lambda = (T, M, \pi)$

- En cada iteración los valores de π, A y B se van actualizando hasta convergencia implementando el algoritmo forward-backward.

Definimos $\gamma_k(s) = Pr[X_k = s|Z, \lambda]$, la probabilidad de que el sistema se encuentre en el estado s en el k-ésimo tiempo, dada la secuencia de observaciones Z en el modelo λ (forward-process).

$\gamma_k(s) = \frac{\alpha_k(s)\beta_k(s)}{Pr[Z|\lambda]} = \alpha_k(s)\beta_k(s)\sum_{s\epsilon\chi}\alpha_t(s)$

Definimos $\xi_k(q, s) = Pr[X_k = q, X_{k+1} = s|Z, \lambda]$, la probabilidad de estar en el estado q al tiempo k y en el estado s en el tiempo $k + 1$ dada la secuencia de observaciones en el modelo actual de HMM (backward-process).

$\xi_k(q, s) = \eta_k\alpha_k(q)T_{q,s}M_{s,z_{k+1}}\beta_{k+1}(s)$, donde η_k es un factor de normalización tal que $\sum_{q,s}\xi_k(q, s) = 1$

Calculando $\gamma_k(s)$ y $\xi_k(q, s)$ podemos actualizar el modelo $\lambda' = (T', M', \pi')$

$\pi'_s = \gamma_1(s)$

$T'_{q,s} = \frac{E[\#\ de\ transiciones\ del\ estado\ q\ al\ s]}{E[\#\ de\ transiciones\ del\ estado\ q]} = \frac{\sum_{k=1}^{t-1}\xi_{k(q,s)}}{\sum_{k=1}^{t-1}\gamma_k(q)}$

$M'_{s,m} = \frac{E[\#\ de\ veces\ en\ el\ estado\ s\ cuando\ la\ observacion\ fue\ m]}{E[\#devecesenelestados]} = \frac{\sum_{k=1}t\gamma_k(s)1(z_k=m)}{\sum_{k=1}^{t}\gamma_k(s)}$

3.2 Suposiciones iniciales del modelo

- Nuestra base será suponer que existen 2 estados: **Consonante** y **Vocal**

- No conocemos con qué probabilidad de inicio estamos en Constante o en Vocal
- No conocemos las probabilidades de transición entre estados
- No conocemos las probabilidades de que cada símbolo del lenguaje pertenezca a uno de los estados

4 Datos

Intentamos ocupar los últimos 100 contenidos publicados en Quién.com y CNNExpansión.com sin embargo sus corpus requieren de mucho de preprocesamiento para eliminar encoding y caracteres extraños.

Decidimos tomar un corpus en español de un ejercicio realizado en Métodos Analíticos correspondiente a un periodico español con 309,918 noticias

4.1 Limpieza de Datos

- Eliminamos signos de puntuación
- Eliminamos dígitos
- Eliminamos tabulaciones
- Cambiamos todo el corpus a minúsculas
- Dividimos cada palabra en sus letras respetando los espacios

5 Metodología

1. Limpieza de datos
2. Separar cada palabra en sus respectivas letras respetando espacios
3. Establecemos nuestro conocimiento a priori sobre las probabilidades iniciales de estados inicializando π. Dado que no conocemos mucho del proceso las establecemos muy cercanas a 0.5 (suman a 1)
4. Establecemos nuestro conocimiento a priori sobre las probabilidades de transición entre estados inicializando A. Dado que no conocemos mucho del proceso las establecemos cercanas a 0.5 pero agregando que creemos que es más probable la transición de vocal a consonante que de vocal a vocal, al igual que de consonante a vocal de que de consontante a consonante.
5. Establecemos nuestro conocimiento a priori sobre las probabilidades de cada símbolo a cada uno de los estados propuestos inicializando B. Dado que no conocemos mucho del proceso las establecemos dividiendo 1 entre el número de símbolos posibles en el set de datos.
6. Inicialización de la HMM con los parámetros establecidos en el paso 4,5 y 6.
7. Correr el algoritmo de Baum-Welch

6 Paquetes utlizadas

- Paquete HMM de R Himmelmann (2010)
- Algoritmo de Baum-Welch para estimación de parámetros de una HMM

7 Resultados

Inicial sin conocimiento:

V	C
0.5337	0.4662

Inicial después de Baum-Welch

V	C
0.5337	0.4662

Transiciones sin conocimiento

	V	C
V	0.3099	0.6900

	V	C
C	0.5200	0.4799

Transiciones después de Baum-Welch

	V	C
V	0.3045	0.6954
C	0.993	0.006

```
$hmm$emissionProbs
      symbols
states           a            b            c           d          e            f           g
     v 0.118369315 1.018630e-65 3.344456e-76 0.008194266 0.19487864 1.743170e-66 0.008411338
     c 0.005519715 1.172973e-02 7.037839e-02 0.140688529 0.04784931 1.172973e-02 0.011348951
      symbols
states           h            i j k           l            m           n           o            p
     v 4.732532e-83 5.702808e-02 0 0 4.622025e-80 1.983201e-92 1.489029e-19 1.466436e-01 8.415060e-72
     c 1.172973e-02 7.955236e-62 0 0 1.290270e-01 3.518919e-02 1.524865e-01 7.102166e-28 4.691892e-02
      symbols
states           q           r          s           t           u            v w x y           z
     v 2.196755e-72 0.008237816 0.01800018 9.776189e-02 6.517495e-02 1.212351e-93 0 0 0 6.455586e-96
     c 3.518919e-02 0.117166365 0.11484041 7.666003e-07 1.653220e-87 2.345946e-02 0 0 0 1.172973e-02
      symbols
states           á é í ó            ú           ñ
     v 0.26915304 0 0 0 0  8.146868e-03 2.268708e-92
     c 0.01128861 0 0 0 0 3.850728e-194 1.172973e-02
```

Figure 2: resultados

8 Código

```
library(HMM)

periodico <- scan(file='./datos/Es_Newspapers.txt', sep="\n", what = character())

#limpieza de datos
for(i in 1:length(periodico)) {
  periodico[i] <- gsub("[[:punct:]]","",unlist(periodico[i]))
  periodico[i] <- gsub("[[:digit:]]","",unlist(periodico[i]))
  periodico[i] <- tolower(periodico[i])
  periodico[i] <- gsub("[[:space:]]"," ",unlist(periodico[i]))
}


#separando a letras
for (i in 1) {
  texto <- periodico[i]
  if(i == 1){
    obsv <- as.data.frame(unlist(strsplit(texto, split="")), stringsAsFactors=FALSE)
  } else {
    obsv <- rbind(obsv, as.data.frame(unlist(strsplit(texto, split="")), stringsAsFactors=FALSE))
  }
}
names(obsv) <- "letter"


states = c("v","c")
symbols = c("a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r",
            "s","t","u","v","w","x","y","z"," ","á","é","í","ó","ú","ñ")
#probabilidades iniciales de lso estados
random_num <- runif(1,0.5,0.7)
startProbs = matrix(c(random_num, 1-random_num), 2)

random_num <- runif(1, 0.5,0.7)
random_num_2 <- runif(1, 0.5,0.7)
transProbs <- matrix(c(1-random_num, random_num_2,
                       random_num, 1-random_num_2), 2)

random_nums <- data.frame(runif(33,0.030,0.034), runif(33,0.030,0.034))
random_nums <- data.frame(rep(1/33,33), rep(1/33,33))
emissionProbs=as.matrix(random_nums)
inithmm <- initHMM(states, symbols, startProbs=startProbs, transProbs=transProbs,
                   emissionProbs=emissionProbs)
bw <- baumWelch(inithmm, obsv$letter)
bw
```

9 Bibliografía