Hidden Markov Models

Miquel Perello Nieto

26 de març de 2012

Resum

En aquesta pràctica comprovarem la bondat en l'estimació dels estats generats a partir d'un model, i tot seguit intentarem treure els models de Markov ocults i els contrastarem amb l'original per veure el nivell de validesa que aquest té, i quins paràmetres ens poden portar cap a una estimació del model optim.

1 Introducció

Per fer aquest estudi ens centrarem a un model deshonest en el que comptem amb un parell de daus, un amb totes les cares equiprobables, i l'altre desequilibrat amb una probabilitat del 50% de treure un 6, i equiprobable per els valors restants. A més existeix un canvi de dau amb una probabilitat d'un 90% i probabilitat de començar amb un dels dos daus del 50%. Tota aquesta informació es pot veure en el fitxer dishonest.hmm (codi 1)

Codi Bash 1: Mostrant codi desonest

```
\$ cat dishonest.hmm
M= 6
N= 2
A:
0.9 0.1
0.1 0.9
B:
0.167 0.167 0.167 0.167 0.166
   0.166
0.1 0.1 0.1 0.1 0.1 0.5
pi:
0.5 0.5
```

Primer de tot comprovarem si els estimador d'estats s'aproxima a la realitat de les tirades.

2 Comprovant estimacions

Primer de tot hem generat diferents tirades de daus amb el model deshonest que hem explicat en la introducció. A més de generar els valors de les tirades ens hem guardat els estats que han generat el valor per tal de comprovar si el programa *testvit* gracies al model i als valors de sortida estima correctament quin ha estat l'estat que ho ha generat.

Hem generat diferents seqüències amb les mides 100, 1.000, 10.000, 100.000, 1.000.000 i 5.000.000 per veure si les estimacions que feia eren estables amb la mida. Sembla que la mida de la seqüència no fa millorar ni empitjorar les estimacions ja que sempre ronden el 73,6% d'encerts.

mida	correctes	percentatge
100	68	68
1.000	771	77.10
10.000	7462	74.62
100.000	74419	74.42
1.000.000	736406	73.64
5.000.000	3691153	73.82

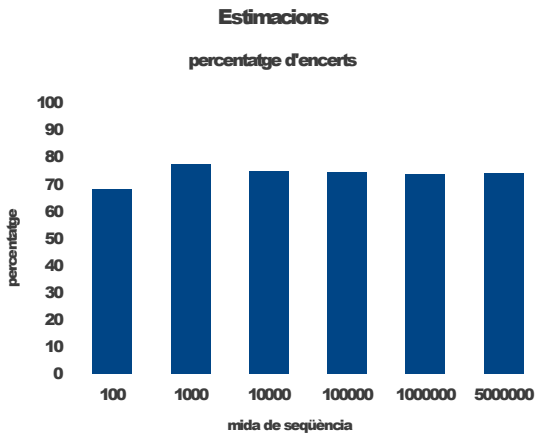

Figura 1: Barplot del percentatge d'encerts del programa *testvit*

3 Trobant models ocults de Markov

Gracies als models ocults de Markov podem intentar predir quins han estat els models que han generat les tirades dels nostres daus. Per fer aquestes proves utilitzarem el programa *esthmm*, el qual ens demanarà la seqüència de valors de sortida que vulguem analitzar, el nombre d'estats que creiem que té el model, i el nombre de símbols. Per ultim nosaltres mateixos haurem d'estimar quines dades són correctes o més realistes, per tant primer de tot provarem incrementant el nombre d'estats (secció 3.1), i comprovarem si podem predir quants estats té realment el model. Tot seguit, ja sabent el nombre de estats que el model real té, farem proves d'execució amb seqüències de diferents mides per veure si els models són més correctes en uns o altres casos (secció 3.2).

3.1 Incrementant el nombre d'estats

En aquesta secció veurem com han anat canviant els resultats dels models ocults de Markov al incrementar el nombre d'estats indicat al programa, i veurem que ens fa pensar en el nombre real d'estats. Ens centrarem individualment en els resultats que s'han vist en els canvis d'estat (secció 3.1.1), en els percentatges dels símbols (secció 3.1.2) i per ultim en l'elecció de l'estat inicial (secció 3.1.3). Per tots els casos l'execució del programa ha estat amb el nombre de símbols fixe a 6, amb una seqüència de 100.000 i incrementant el nombre d'estats (1, 2, 3, 4 i 5).

3.1.1 Resultats en els canvis d'estat (A)

Anem a analitzar les probabilitats de canvi d'estats que ens diu el model de Markov quan incrementem el nombre d'estats.

Podem veure que amb un estat ens diu que hi ha un 100% de probabilitat de quedar-se en el mateix estat. Quan li diem que el model pot tenir dos estats veiem que apareix una clara diferencia entre els estats, un amb un 90% i l'altre amb un 10%. Això es un clar indicador que realment existeix més d'un estat, ja que si realment existís un sol estat seria d'esperar que ens digues que la probabilitat de canvi d'un a l'altre rondaria el 50%.

A partir dels tres estats comencem a apreciar que els resultats es solen dividir en dues parts ben diferenciades, per exemple en els tres estats el segon i el tercer tenen uns percentatges de canvi molt similars, el que ens ha de fer dubtar que aquests dos siguin diferents. Amb quatre estats el primer i l'ultim són molt semblant però s'intercanvien la primera i ultima probabilitat de canvi (un porta a l'altre i l'altre al primer) el que ens fa dubtar que siguin diferents. I el segon i tercer són molt semblants per tant es veuen dos models diferenciats. En l'ultim cas amb cinc estats es pot arribar a deduir el mateix amb primer, segon i cinquè representant un estat i el tercer i quart un altre.

Per tant veient aquests resultats podríem dir amb certesa que el model real té dos estats.

REAL				
0,10	0,90			
0,90	0,10			
estats	**1**			
1,00				
estats	**2**			
0,89	0,11			
0,12	0,88			
estats	**3**			
0,87	0,07	0,07		
0,11	0,43	0,46		
0,11	0,59	0,31		
estats	**4**			
0,78	0,09	0,08	0,05	
0,23	0,18	0,29	0,30	
0,25	0,17	0,26	0,32	
0,05	0,08	0,09	0,78	
estats	**5**			
0,29	0,27	0,07	0,08	0,29
0,29	0,27	0,07	0,08	0,29
0,04	0,04	0,64	0,23	0,04
0,06	0,06	0,56	0,27	0,06
0,29	0,27	0,07	0,07	0,30

3.1.2 Resultats en els simbols (B)

En aquest punt ens centrarem en les probabilitats dels símbols en cada estat.

En el primer cas podem veure que el sisè símbol té més probabilitat d'aparèixer que els altres, tenint

en compte que són tirades de daus ja podem començar a sospitar sobre la equitat dels daus, però encara no podem assegurar que hi hagi un sol estat.

Amb dos estats veiem que hi ha una diferencia clara en un i l'altre estat, sobretot en el sisè símbol. El que es pot deduir es que hi ha un dau que està equilibrat ja que tots els símbols tenen una probabilitat similar, en canvi apareix un estat amb un 50% de probabilitat de treure el sisè símbol.

A partir de tres estats com en la secció anterior es pot apreciar que sembla haver una tendència a crear dos grups d'estats bastant similars. Això es un indicador que realment són dos els estats reals.

En concret podem veure que amb tres estats el segon i el tercer són molt semblants, amb quatre estats es genera un grup amb el primer, el segon i el tercer amb alguna diferencia notable en el sisè símbol; que sembla estar arrossegat per el quart estat que té una alta probabilitat del sisè símbol.

Per ultim amb quatre estats es veu el primer el segon i el cinquè estats amb unes probabilitats molt similars i el tercer i el quart també entre ells. Per tant amb aquests resultats també ens porten a creure que hi ha tan sols dos estats.

REAL					
0,17	0,17	0,17	0,17	0,17	0,17
0,10	0,10	0,10	0,10	0,10	0,50
estats	**1**				
0,14	0,14	0,13	0,13	0,13	0,33
estats	**2**				
0,10	0,11	0,10	0,10	0,10	0,50
0,17	0,17	0,18	0,17	0,17	0,15
estats	**3**				
0,17	0,17	0,18	0,17	0,17	0,14
0,11	0,11	0,10	0,11	0,10	0,48
0,10	0,10	0,10	0,10	0,10	0,51
estats	**4**				
0,17	0,18	0,18	0,18	0,17	0,12
0,16	0,14	0,16	0,12	0,17	0,25
0,18	0,12	0,16	0,13	0,13	0,29
0,09	0,10	0,08	0,10	0,09	0,54
estats	**5**				
0,17	0,17	0,18	0,17	0,17	0,15
0,18	0,17	0,18	0,17	0,17	0,15
0,09	0,10	0,09	0,10	0,09	0,53
0,11	0,11	0,11	0,10	0,11	0,47
0,17	0,17	0,17	0,17	0,17	0,14

3.1.3 Resultats en l'estat inicial (pi)

Per finalitzar les proves en l'increment del nombre d'estats comprovem la probabilitat d'inici dels estats. Podem veure que en tots els casos la probabilitat està centrada en un dels estats.

Això no es dona cap informació ja que amb només una seqüència només podem predir quin estat es el principal, però no amb quina probabilitat comença un o l'altre.

REAL				
0,5	0,5			
estats	1			
1,00				
estats	2			
1,00	0,00			
estats	3			
0,00	0,02	0,98		
estats	4			
0,00	0,00	0,00	1,00	
estats	5			
0,00	0,00	0,99	0,01	0,00

3.2 Incrementant la mida de seqüència

En aquesta secció ens centrarem en observar els diferents resultats dels models ocults de Markov al incrementar la mida de la seqüència de sortida.

Analitzarem per separat els resultats en els canvis d'estat (secció 3.2.1), en les probabilitats dels símbols (secció 3.2.2) i per ultim en l'estat inicial (secció 3.2.3).

Per tots els casos l'execució del programa ha estat amb el nombre de símbols fixe a 6, el nombre d'estats a 2 i amb les mides de seqüència 1.000, 10.000, 100.000, 1.000.000 i 5.000.000.

3.2.1 Resultats en els canvis d'estat (A)

Centrem-nos primer en els resultats que apareixen en els canvis d'estat, recordem que el valor real havia de ser 10% i 90%. Podem observar que amb la mida 100.000 i amb la mida 5.000.000 els resultats s'aproximen molt al model real, però havent executat només una prova de cada no podríem dir que aquestes mides maximitzin la certesa. Per tant només podem dir que no podem apreciar una millora notable amb la mida de la seqüència a l'hora de predir el model.

REAL		
0,10	0,90	
0,90	0,10	
mida		
1000	0,85	0,15
	0,21	0,79
10000	0,44	0,56
	0,42	0,59
100000	0,89	0,11
	0,12	0,88
1000000	0,26	0,74
	0,79	0,21
5000000	0,88	0,12
	0,11	0,89

3.2.2 Resultats en els símbols (B)

En aquest apartat podem observar les probabilitats de cada símbol per cada un dels dos estats.

Entre aquests models podem veure que el segon i el quart no s'apropen molt a la realitat, però en canvi el primer i el tercer si que ho fan. Per ultim el cinquè es quasi perfecte ja que totes les probabilitats són correctes menys en el sisè símbol que ha fet la repartició un 1% esbiaixada. En aquest ultim cas sembla que podríem dir que amb una gran quantitat de valors es pot arribar a un resultat optim, però per assegurar-ho s'haurien d'executar moltes més proves.

REAL					
0,17	0,17	0,17	0,17	0,17	0,17
0,10	0,10	0,10	0,10	0,10	0,50
mida	**100**				
0,18	0,16	0,19	0,13	0,20	0,14
0,11	0,10	0,09	0,10	0,01	0,59
mida	**1000**				
0,19	0,12	0,17	0,12	0,10	0,31
0,10	0,15	0,10	0,14	0,17	0,35
mida	**10000**				
0,10	0,11	0,10	0,10	0,10	0,50
0,17	0,17	0,18	0,17	0,17	0,15
mida	**100000**				
0,13	0,13	0,14	0,13	0,14	0,33
0,14	0,14	0,13	0,14	0,13	0,33
mida	**5000000**				
0,17	0,17	0,17	0,17	0,17	0,15
0,10	0,10	0,10	0,10	0,10	0,49

3.2.3 Resultats en l'estat inicial (pi)

En aquest cas tenim els resultats en la predicció de la probabilitat de sortida de l'estat inicial. Com passava en els casos anteriors no li és possible predir quina es aquesta probabilitat, ja que només compta amb una serie per execució, el que fa que l'estat inicial sigui imprevisible.

REAL		
0,5	0,5	
mida		
1000	0,00	1,00
10000	0,05	0,95
100000	1,00	0,00
1000000	0,60	0,40
5000000	0,00	1,00

4 Conclusions

Finalment hem pogut observar diferents resultats en la estimació de models creadors de unes seqüències de dades donades, i valorat aquestes estimacions amb un programa creat específicament per aquest fi. Hem pogut veure que les estimacions ronden el 73% d'encerts els quals en certs àmbits pot ser suficient, però que en altres (com pot ser el genòmic) pot comportar que un resultat no s'apropi a una solució correcte. Per tant es responsabilitat de l'investigador determinar fins a quin punt pot ser valida una estimació o no.

També hem pogut veure els resultats que ens pot donar els algoritmes de cerca de models ocults de Markov, i hem vist que en el nostre cas podíem arribar a deduir correctament que el nostre model tractava amb un parell de daus, dels quals un estava equilibrat i l'altre descompensat cap al 6, i hem vist que gracies a la seqüència de 5.000.000 valors el resultat ha estat quasi exacte. Aquests resultats contenen una incertesa però segons l'aplicació pot ser suficient per determinar la validesa de certs models.

A Codis

En aquesta secció es pot trobar el programa que s'ha creat per calcular el percentatge d'encert del programa *testvit*, i els resultats de les execucions dels models ocults de Markov.

A.1 Validar l'estimació

Aquí es presenta el programa que s'ha emprat per realitzar les validacions del resultat de predicció del programa *testvit*. Aquest programa el que fa es llegir per entrada estandar el nombre de estats a comprovar, tot seguit els estats originals, i per ultim els estats predits. Aquest només compta el nombre de coincidències, el mostra per sortida estandar i finalment mostra el percentatge d'encerts.

Codi C 2: `Percentatge.cpp`

```
1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4  using std::fixed;
5  using namespace std;
6
7  int main (int argc, char **argv)
8  {
9    vector<short> original;
10   unsigned int num, corr;
11   short val;
12   double perc;
13
14   cin >> num;
15   cout << "Llegint " << num << "
         valors" << endl;
16   for (unsigned int i = 0; i <
        num; ++i)
17   {
18     cin >> val;
19     original.push_back(val);
20   }
21
22   perc = 0;
23   cout << "Original llegit" <<
        endl << "Llegint " << num
        << " valors" << endl;
24   for (unsigned int i = 0; i <
        num; ++i)
25   {
26     cin >> val;
27     if (original[i] == val)
28       corr++;
29   }
30
31   perc = ((double)corr/(double)
        num)*100;
32   cout << "Totals \t= " << num
        << "\nCorrectes \t= " <<
        corr << "\nPercent \t= " <<
         perc << endl;
33 }
```

A.2 Incrementant el nombre d'estats

Tot seguit hi ha tots els resultats de les execucions del estimador de models ocults de Markov, en el cas en que es va anar incrementant el nombre d'estats.

Codi Bash 3: `Comprovant Model de Markov ocult`

```
\$ ./esthmm -N 1 -M 6
   dishonest_100000.valors
M= 6
N= 1
A:
1.000000
B:
0.136135 0.135046 0.133977
   0.133377 0.132269 0.334196
pi:
1.000000
```

Codi Bash 4: `Comprovant Model de Markov ocult`

```
\$ ./esthmm -N 2 -M 6
   dishonest_100000.valors
M= 6
N= 2
A:
0.893432 0.107568
0.121892 0.879108
B:
0.103342 0.105048 0.097079
   0.101500 0.101236 0.496794
0.173339 0.169079 0.175839
   0.169544 0.167477 0.149722
pi:
0.999861 0.001139
```

Codi Bash 5: `Comprovant Model de Markov ocult`

```
\$ ./esthmm -N 3 -M 6
   dishonest_100000.valors
M= 6
N= 3
A:
0.866814 0.069300 0.065885
0.113782 0.429565 0.458653
0.106913 0.586300 0.308788
B:
0.174957 0.170777 0.177998
   0.171294 0.169035 0.140937
0.107000 0.107890 0.099092
   0.105702 0.101520 0.483795
0.100603 0.102767 0.095992
   0.097702 0.102557 0.505379
pi:
0.001131 0.021414 0.979454
```

Codi Bash 6: Comprovant Model de Markov ocult

```
\$ ./esthmm -N 4 -M 6
   dishonest_100000.valors
M= 6
N= 4
A:
0.775748 0.090594 0.083783
   0.052875
0.234110 0.181471 0.292017
   0.295402
0.250954 0.168485 0.258735
   0.324827
0.047079 0.083200 0.091093
   0.781627
B:
0.171513 0.180160 0.176891
   0.181823 0.170559 0.124054
0.156800 0.142840 0.161448
   0.122287 0.173050 0.248575
0.178984 0.116164 0.160711
   0.125084 0.130193 0.293864
0.088570 0.102080 0.083560
   0.099173 0.091256 0.540360
pi:
0.001097 0.001481 0.001633
   0.998789
```

Codi Bash 7: Comprovant Model de Markov ocult

```
\$ ./esthmm -N 5 -M 6
   dishonest_100000.valors
M= 6
N= 5
A:
0.291084 0.270512 0.072106
   0.075414 0.294885
0.288055 0.271472 0.074827
   0.078149 0.291497
0.042702 0.043237 0.642011
   0.233457 0.042593
0.057350 0.058113 0.560538
   0.270806 0.057193
0.290542 0.270260 0.071555
   0.074859 0.296785
B:
0.173208 0.170842 0.175616
   0.171299 0.168711 0.145325
0.176345 0.166293 0.179786
   0.167643 0.166589 0.148343
0.094450 0.098984 0.087614
   0.098492 0.094484 0.530976
0.112700 0.108882 0.106145
   0.097683 0.106162 0.473428
0.172500 0.171855 0.174729
   0.172057 0.169142 0.144717
pi:
0.001143 0.001148 0.993622
   0.006944 0.001142
```

A.3 Incrementant la mida de seqüència

Tot seguit hi ha tots els resultats de les execucions del estimador de models ocults de Markov, en el cas en que es va anar incrementant la mida de la seqüència d'entrada.

Codi Bash 8: Comprovant Model de Markov ocult

```
\$ ./esthmm -N 2 -M 6
   dishonest_1000.valors
M= 6
N= 2
A:
0.848305 0.152695
0.207470 0.793530
B:
```

```
0.180773 0.155781 0.191197
    0.134785 0.203297 0.139167
0.108103 0.099586 0.086986
    0.104455 0.011957 0.593913
pi:
0.002286 0.998714
```

Codi Bash 9: Comprovant Model de Markov ocult

```
\$ ./esthmm -N 2 -M 6
    dishonest_10000.valors
M= 6
N= 2
A:
0.444477 0.556523
0.415719 0.585281
B:
0.185218 0.115958 0.172547
    0.120504 0.101794 0.308979
0.096235 0.146537 0.100634
    0.137037 0.171066 0.353491
pi:
0.050190 0.950810
```

Codi Bash 10: Comprovant Model de Markov ocult

```
\$ ./esthmm -N 2 -M 6
    dishonest_100000.valors
M= 6
N= 2
A:
0.893432 0.107568
0.121892 0.879108
B:
0.103342 0.105048 0.097079
    0.101500 0.101236 0.496794
0.173339 0.169079 0.175839
    0.169544 0.167477 0.149722
pi:
0.999861 0.001139
```

Codi Bash 11: Comprovant Model de Markov ocult

```
\$ ./esthmm -N 2 -M 6
    dishonest_1000000.valors
M= 6
N= 2
A:
0.257112 0.743888
0.788797 0.212203
B:
0.131637 0.132365 0.135654
    0.134774 0.136882 0.333689
0.136707 0.137234 0.132869
    0.135178 0.129532 0.333479
pi:
0.602184 0.398816
```

Codi Bash 12: Comprovant Model de Markov ocult

```
\$ ./esthmm -N 2 -M 6
    dishonest_5000000.valors
M= 6
N= 2
A:
0.881328 0.119672
0.108035 0.892965
B:
0.169592 0.169855 0.170390
    0.170529 0.169772 0.154863
0.101827 0.102480 0.102410
    0.101678 0.101775 0.494831
pi:
0.001141 0.999859
```

Referències

[1] Tapas Kanungo, ÜMDHMM: Hidden Markov Model Toolkit,ïn "Extended Finite State Models of Language,"A. Kornai (editor), Cambridge University Press, 1999. http://www.kanungo.com/software/software.html.

[2] Conjunt de programes de Hidden Markov Models `http://www.lsi.upc.edu/~peypoch/docencia/ri/hmm/umdhmm-v1.02.tar`