Intelligent Systems

Escuela Técnica Superior de Informática

Universitat Politècnica de València

Block 2 Chapter 7:
Estimation of Markov Models

Index

Learning: estimation of probabilities of a Markov model criterium to optimize

Basic problem:

Estimate/learn the probabilities/parameters of a Markov model M; that is, learn A, B and π.

Use a set of training strings $Y = \{y_1, \ldots, y_n\}$ drawn independently according to the probability rule $P(y|M)$.

Since the strings have been drawn independently:

$$P(Y|M) = \prod_{k=1}^{n} P(y_k|M)$$

The *maximum likelihood estimator* of M is:

$$\hat{M} = \underset{M}{\text{argmax}} \prod_{k=1}^{n} P(y_k|M) \approx \underset{M}{\text{argmax}} \prod_{k=1}^{n} \tilde{P}(y_k|M)$$

Estimation by Viterbi algorithm

Basic idea:

Parse all the strings in Y, counting the frequencies of use of transitions between states, frequencies of generation of symbols in each state, etc., and normalize to obtain the probabilities

Problem:
How can we parse a string if the model probabilities are not known and so we cannot calculate the sequence of states?

A possible solution:

1. Initialize the model probabilities “properly” (we obtain an initial Markov model)
2. Parse each string in $y \in Y$ using the Viterbi algorithm and obtain the corresponding sequence of states
3. Count the required frequencies (transitions between states -A-, generation of symbols in each state -B-) for the sequence of states of each string
4. Normalize frequencies to obtain the new model probabilities
5. Repeat steps 2-4 until convergence (the model probabilities converge)

This is called ***re-estimation***: we are given an initial Markov model M with initial values A, B, and π, and we have to ***re-estimate*** these values by parsing a set of input strings.

Viterbi re-estimation: example (1)

Recall: **(1)** we are given a set of strings and an initial model M with values A, B and π; **(2)** we apply the Viterbi algorithm to obtain the optimal sequence of states for each string; **(3)** we count the required frequencies; **(4)** we normalize these frequencies, thus obtaining the new values for A, B, and π; **(5)** we repeat steps 2-4 until de model probabilities converge.

Example of steps 3 and 4: We have three strings that represent three hand-written digits (digit "*siete*"). Each symbol in the string ('a', 'b', 'c', 'd') represents one out of the possible 4 directions in the written digit.

Suppose we obtain the *optimal sequence of states* for each *string* by applying the Viterbi algorithm:

String 1: **aaaaaddcdcdcdcdccbabaabababccccb**
Optimal state sequence: **11111222222222222333333333344444F**

String 2: **aaaaaddcdcdcdccdcbababababaabcccdcbb**
Optimal state sequence: **111112222222222223333333333444444 4F**

String 3: **aaaadcdcdcdcdcdcbababababccdccbaab**
Optimal state sequence: **1111222222222222333333333444444444F**

Viterbi re-estimation: example (2)

Counting frequencies and normalization

11111222222222222333333333344444F
11111222222222222333333333344444444F
111122222222222233333333344444444 4F

$$\pi_1 = 3/3 = 1 \quad \pi_2 = \pi_3 = \pi_4 = 0$$

A	1	2	3	4	F
1	$4+4+3$	$1+1+1$	0	0	0
2	0	$11+11+11$	$1+1+1$	0	0
3	0	0	$9+9+8$	$1+1+1$	0
4	0	0	0	$4+6+8$	$1+1+1$

$\Rightarrow$

A	1	2	3	4	F
1	$\frac{11}{14}$	$\frac{3}{14}$	0	0	0
2	0	$\frac{33}{36}$	$\frac{3}{36}$	0	0
3	0	0	$\frac{26}{29}$	$\frac{3}{29}$	0
4	0	0	0	$\frac{18}{21}$	$\frac{3}{21}$

B	a	b	c	d
1	$5+5+4$	0	0	0
2	0	0	$6+6+6$	$6+6+6$
3	$5+5+4$	$5+5+5$	0	0
4	$0+0+2$	$1+2+2$	$4+4+4$	$0+1+1$

$\Rightarrow$

B	a	b	c	d
1	$\frac{14}{14}$	0	0	0
2	0	0	$\frac{18}{36}$	$\frac{18}{36}$
3	$\frac{14}{29}$	$\frac{15}{29}$	0	0
4	$\frac{2}{21}$	$\frac{5}{21}$	$\frac{12}{21}$	$\frac{2}{21}$

Viterbi re-estimation algorithm

Input: $M^0 = (Q^0, \Sigma^0, \pi^0, A^0, B^0)$ /* Initial model */
$Y = \{y_1, \ldots, y_n\}$ /* training sets */
Output: $M = (Q, \Sigma, \pi, A, B)$ /* Optimized model */

$M = M^0$
repeat $M' = M$; $\pi = 0$; $A = 0$; $B = 0$
  for $k = 1$ **to** n **do**
    $m = |y_k|$ /* most probable state sequence for y_k, */
    $\tilde{q}_1, \ldots, \tilde{q}_m = \mathrm{argmax}_{q_1,\ldots,q_m} P(y_k, q_1, \ldots, q_m \mid M')$ /* by Viterbi */
    $\pi_{\tilde{q}_1}$++; $B_{\tilde{q}_1, y_{k,1}}$++ /* counter update */
    for $t = 2$ **to** m **do** $A_{\tilde{q}_{t-1}, \tilde{q}_t}$++; $B_{\tilde{q}_t, y_{k,t}}$++ **done**; $A_{\tilde{q}_m, F}$ ++
  done
  $s = \sum_{q \in Q} \pi_q$
  forall $q \in Q$ **do** /* counter normalization */
    $\pi_q = \pi_q / s$
    $a = \sum_{q' \in Q} A_{q,q'}$; **forall** $q' \in Q$ **do** $A_{q,q'} = A_{q,q'}/a$
    $b = \sum_{\sigma \in \Sigma} B_{q,\sigma}$; **forall** $\sigma \in \Sigma$ **do** $B_{q,\sigma} = B_{q,\sigma}/b$
  done
until $M = M'$

Viterbi re-estimation algorithm: exercise (1)

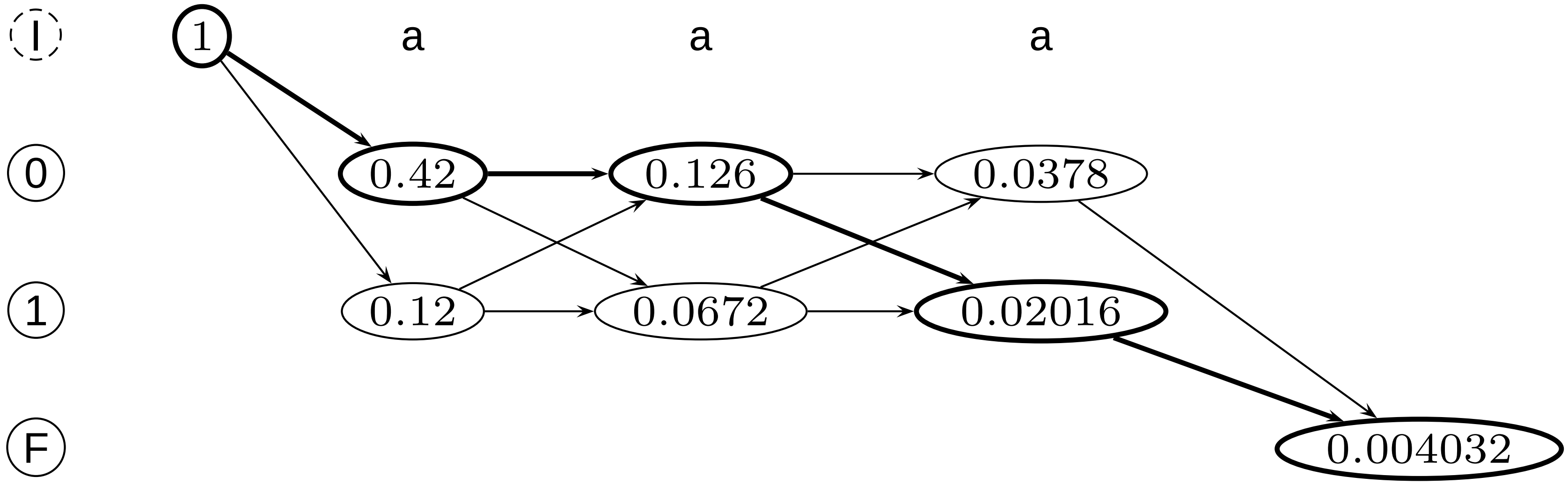
Let M be a Markov model with states $Q = \{0, 1, F\}$; alphabet $\Sigma = \{a, b\}$; prior probabilities $\pi_0(0) = 0.7, \pi_0(1) = 0.3$; transition probabilities and emission probabilities:

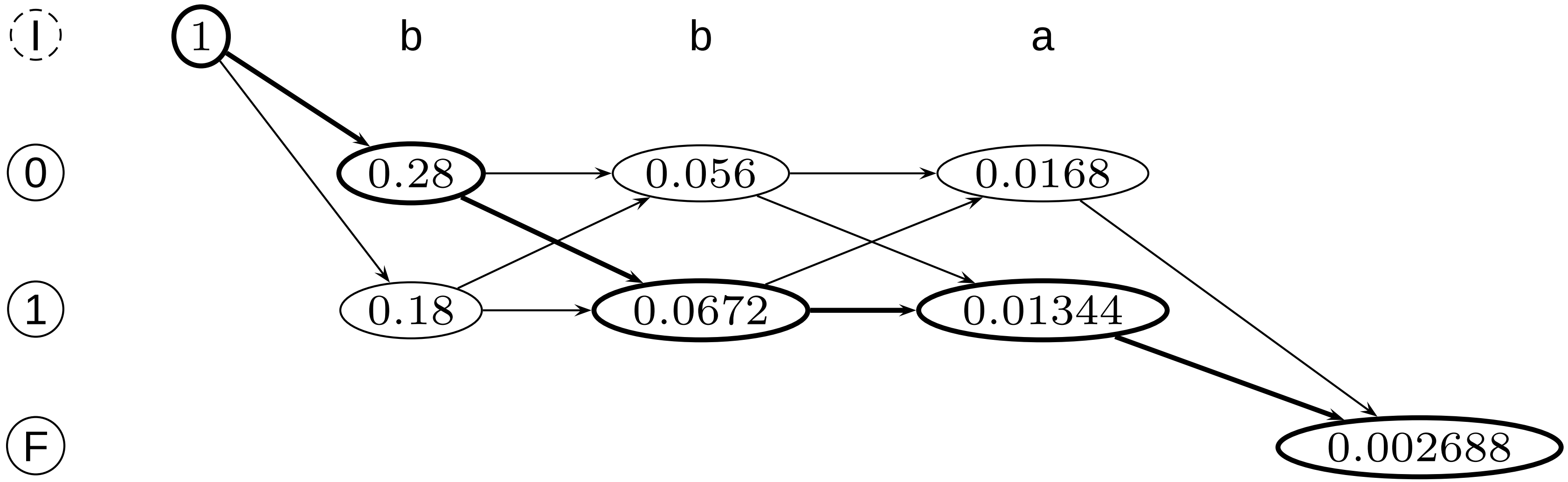
A	0	1	F
0	0.5	0.4	0.1
1	0.3	0.5	0.2

B	a	b
0	0.6	0.4
1	0.4	0.6

Re-estimate the parameters of M through one iteration of Viterbi re-estimation algorithm from the training sets "a a a" and "b b a".

Exercise (2): calculating the most probable state sequences by Viterbi





Exercise (3): re-estimating the parameters of M

$$\hat{\pi}_0(0) = \frac{2}{2} = 1$$

$$\hat{\pi}_0(1) = \frac{0}{2} = 0$$

A	0	1	F
0	$\frac{1}{3}$	$\frac{2}{3}$	0
1	0	$\frac{1}{3}$	$\frac{2}{3}$

B	a	b
0	$\frac{2}{3}$	$\frac{1}{3}$
1	$\frac{2}{3}$	$\frac{1}{3}$

Now, we should repeat the same calculations with this new model M; that is, compute the optimal sequences for "a a a" and "b b a" by using Viterbi with the new M, count frequencies and normalize. And so on until M converges.

Index

Initialization for Viterbi re-estimation

What happens if we are not given an initial Markov model M? How can we parse the strings if the model probabilities are unknown? How can we compute the optimal sequences of states for each string?

Solution: Initialize all probabilities following a uniform distribution

Problem: This usually produces convergence problems or a convergence to inadequate local maxima

A useful idea for linear or left-to-right models:

- Split each string in Y in as many segments (approximately) as states in the Markov model.
- Assign the symbol of each segment to one state
- Count the frequencies of transition and generation
- Normalize frequencies to obtain the *required initial probabilities*

Initialization by linear segmentation: example

Obtain a Markov model with $N = 3$ states by linear segmentation from the strings

$y_1 =$ aabbbcc $\qquad$ $y_2 =$ aaabbccc

$$Q = \{1, 2, 3, F\} \qquad \Sigma = \{a, b, c\}$$

$$q = \left\lfloor \frac{t \cdot N}{\mid y \mid +1} \right\rfloor + 1 :$$

aabbbcc	aaabbccc
1122233	11222333

$$\pi_1 = \tfrac{2}{2}, \quad \pi_2 = \pi_3 = 0$$

A	1	2	3	F
1	$\frac{2}{4}$	$\frac{2}{4}$	0	0
2	0	$\frac{4}{6}$	$\frac{2}{6}$	0
3	0	0	$\frac{3}{5}$	$\frac{2}{5}$

B	a	b	c
1	$\frac{4}{4}$	0	0
2	$\frac{1}{6}$	$\frac{5}{6}$	0
3	0	0	$\frac{5}{5}$

Once we have obtained an initial model M by linear segmentation, we can now apply Viterbi re-estimation.

Initialization by linear segmentation for Viterbi re-estimation

Input: $Y = \{y_1, \ldots, y_n\}$, N /* training strings, number of states */

Output: $M = (Q, \Sigma, \pi, A, B)$ /* model */

$Q = \{1, 2, \ldots, N, F\}$; $\Sigma = \{y \in y_k \in Y\}$ /* states and symbols */
$\pi = 0$; $A = 0$; $B = 0$ /* initialization of counters*/

for $k = 1$ **to** n **do** /* counter updating using */
  $q = 1$; π_q++; $B_{q,y_{k,1}}$++ /* linear alignment of y_k with the states */
  for $t = 2$ **to** $|y_k|$ **do** $q' = q$; $q = \left\lfloor \frac{t}{|y_k|+1} N \right\rfloor + 1$; $A_{q',q}$++; $B_{q,y_{k,t}}$++ **done**
  $A_{q,F}$ ++
done

$s = \sum_{q \in Q} \pi_q$

forall $q \in Q$ **do** /* counter normalization */
  $\pi_q = \pi_q \,/\, s$
  $a = \sum_{q' \in Q} A_{q,q'}$; **forall** $q' \in Q$ **do** $A_{q,q'} = A_{q,q'} \,/\, a$
  $b = \sum_{\sigma \in \Sigma} B_{q,\sigma}$; **forall** $\sigma \in \Sigma$ **do** $B_{q,\sigma} = B_{q,\sigma} \,/\, b$
done