Intelligent Systems

Escuela Técnica Superior de Informática

Universitat Politècnica de València

Block 2 Chapter 5: Syntactical/Structural Methods. Markov models.

DSIC – UPV: December, 2015

Index

Index

Syntactic/Structural Pattern Recognition

So far we have used flat, numerical feature vectors of fixed dimensionality to represent the attributes/components of the patterns.

But this does not allow for representing pattern structures taking into account more complex interrelationships.

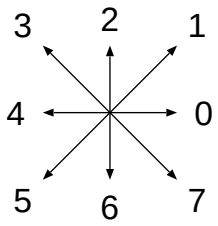
Syntactic or structural Pattern Recognition makes use of symbolic and structured information. The goal is to find a clear structure in the patterns.

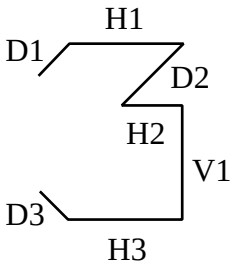
Structured objects in Pattern Recognition

- Representing patterns through numerical feature vectors may result in a loss of information for some applications:
  - Speech Recognition
  - Handwriting Recognition
  - Language identification
  - Chromosome identification
  - Scene recognition from video or images
  - ...
- Solution: to use a structured representation:
  - Sequences of vectors or symbols of variable length
  - Trees, graphs, etc.
- Modeling: Using structural models; for instance, stochastic grammars or hidden Markov models

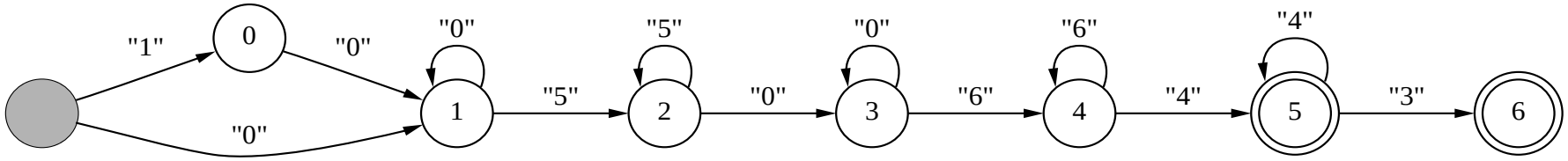
Grammatical modeling of stroke sequences

Syntactic modelling of a hand-written "3"

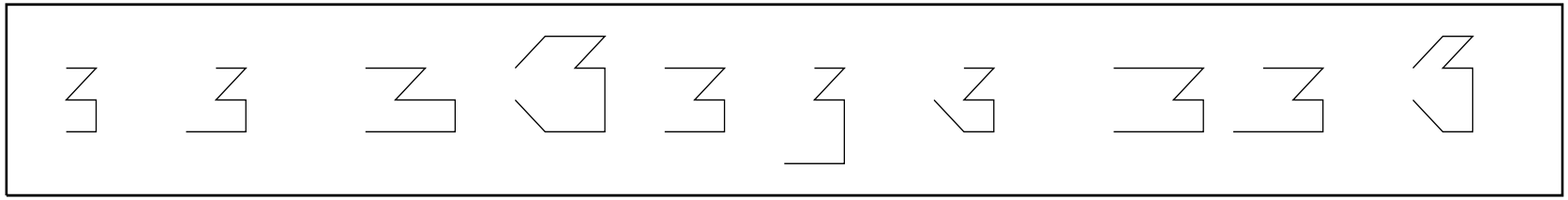




S	-->	D1	H1	D2	H2	V1	H3 D3
D1	-->	"1"	|	λ			
H1	-->	"0"	|	"0"	H1		
D2	-->	"5"	|	"5"	D2		
H2	-->	"0"	|	"0"	H2		
V1	-->	"6"	|	"6"	V1		
H3	-->	"4"	|	"4"	H3		
D3	-->	"3"	|	λ			



tres.gr

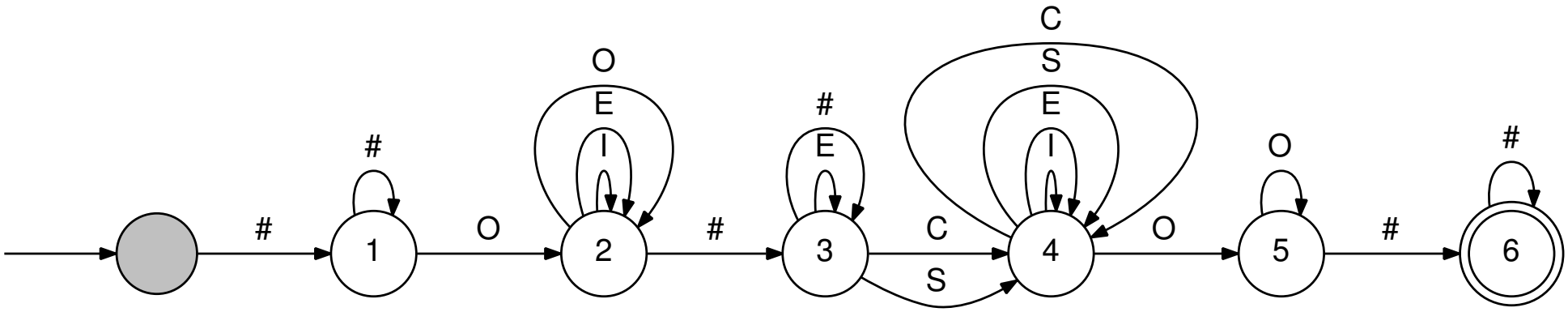


"Three" strokes generated by grammar "tres.gr"

Syntactic modeling of the pronunciation of isolated words

Examples of utterances of the word ”ocho“, represented by acoustic-phonetic strings

```
######OOEIO##ECEEOOO######
#####OOOEI###ECCEEOOO#####
#####OOOE###SCEOOOO######
######OOOIO###CCEOOOO######
######OOEIO##SCCEOOOO######
#####OOOEIE##ECCIEOOOO######
#####OOEO###CCIEOOOO######
######OOOE###ECCEIEOO#####
#####OOOEO##SCCIEEOO######
######OOOEO##SCSCCIEEOOO######
#####OOOEO##ECSCCEIOOOO#####
            . . .
```

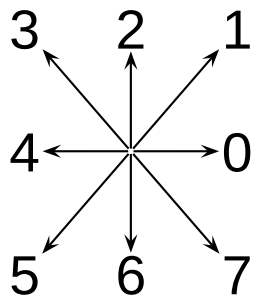


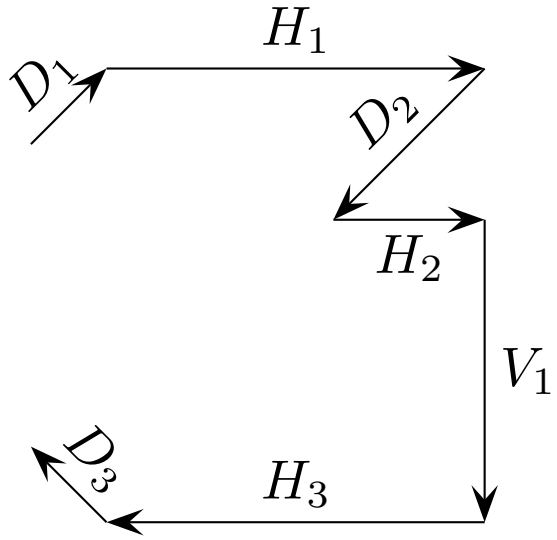
Grammar that models the utterances of the word ”ocho“

Index

A syntactic classifier for digit strokes

Let's get back to the syntactic modeling of a handwritten "3":





$$
\begin{array}{lcl}
S & \Rightarrow & D_1 \\
D_1 & \Rightarrow & 1\,H_1 \mid H_1 \\
H_1 & \Rightarrow & 0\,H_1 \mid 0\,D_2 \\
D_2 & \Rightarrow & 5\,D_2 \mid 5\,H_2 \\
H_2 & \Rightarrow & 0\,H_2 \mid 0\,V_1 \\
V_1 & \Rightarrow & 6\,V_1 \mid 6\,H_3 \\
H_3 & \Rightarrow & 4\,H_3 \mid 4\,D_3 \\
D_3 & \Rightarrow & 3 \mid \lambda
\end{array}
$$

Let's assume we have a grammar like this one, G_c, for each digit c.

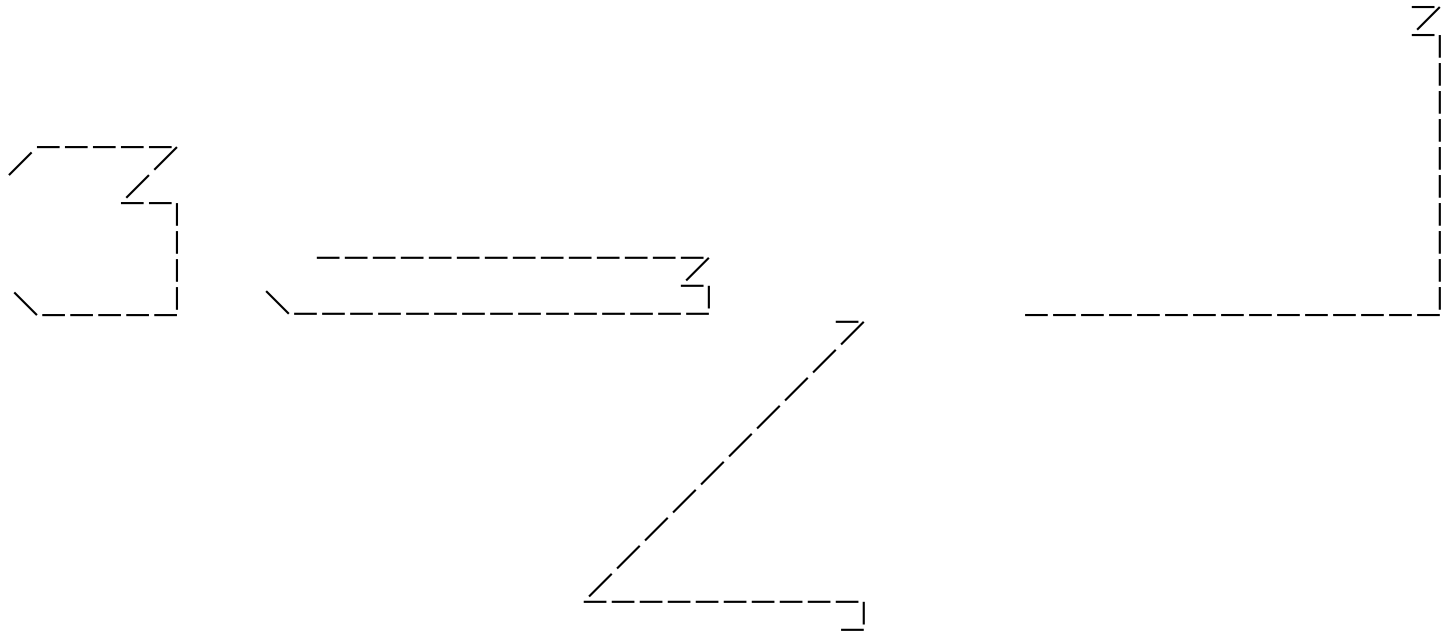
If $x \in \{0, 1, \ldots, 7\}^+$ represents a digit stroke, we can decide which class x belongs to by using a *'pure' syntactic classifier*:

$$
c(x) = \begin{cases}
c & \text{if } G_c \text{ is the only grammar that generates } x \\
\text{"reject"} & \text{if no grammar generates } x \\
\text{"doubt"} & \text{if more than one grammar generates } x
\end{cases}
$$

Pitfalls of conventional grammars and "pure" syntactic classification

- The need for inclusion of the classes "reject" and "doubt" is a clear drawback

- Another drawback is that conventional grammar generate natural outcomes as well as "undesirable" outcomes; for instance:



Common solution: introduce probabilities in grammars $\rightarrow$ ***stochastic grammars***.

Stochastic grammars

- A ***stochastic grammar*** G' is a grammar G with probabilities associated to its rules:

$$G' = (G, p), \quad G = (N, \Sigma, R, S), \quad p : R \to [0, 1]$$

- A *context-free* (or *regular*) grammar is ***proper*** if:

$$\forall A \in N \quad \sum_{\forall \beta : A \to \beta \in R} p(A \to \beta) = 1$$

- ***Probability of a string*** y to be generated by G':

$$\forall y \in \Sigma^* \quad p(y|G') = \sum_{d \in D_G(y)} p(d), \qquad p(d) = \prod_{(A \to \beta) \in d} p(A \to \beta)$$

- A stochastic grammar G' is ***consistent*** if:

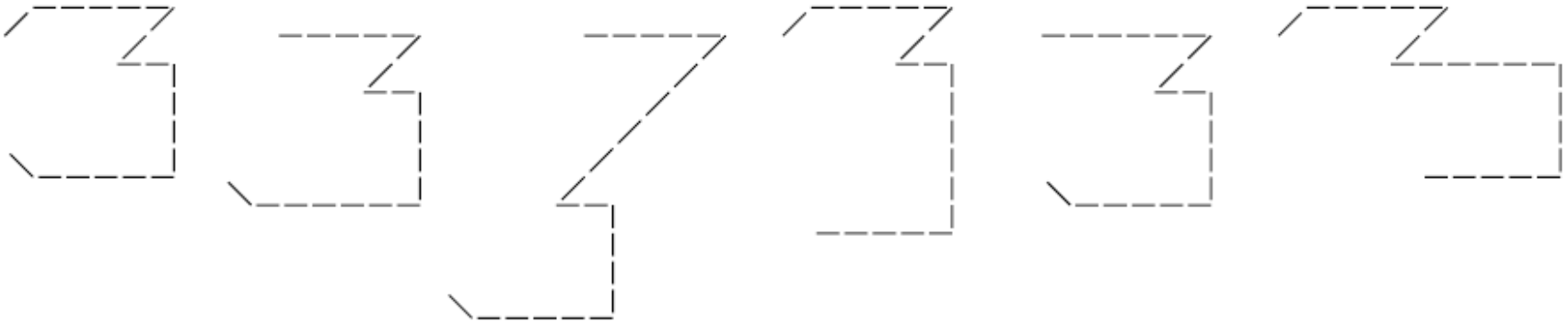
$$\sum_{y \in \Sigma^*} p(y|G') = 1$$

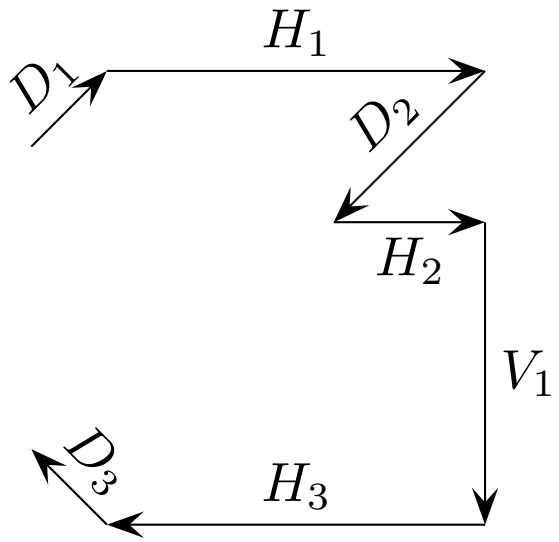
Learning stochastic grammars

- Learning the *rules* or their *"topology"* (structure of the rules):
  - Difficult to automate
  - Approach: pre-defined topology $\Rightarrow$ ***Markov Models***

- Learning probabilities: *estimation*
  - ***Non-ambiguous context-free (or regular) grammars*** G'***:***
    Maximum likelihood estimation from the frequencies of use of the rules during the parsing of a sequence of training strings supposedly generated by G'.
    These estimations get close to the true probabilities when the number of training strings $\rightarrow \infty$.
  - ***Ambiguous regular grammars and/or Markov models:***
    Locally optimal estimation by using ***"Viterbi reestimation"*** or the "Backward-Forward" algorithm.

Learning probabilities: example





$$S \xrightarrow{6/6} D_1$$

$$D_1 \xrightarrow{3/6} 1\,H_1 \qquad D_1 \xrightarrow{3/6} H_1$$

$$H_1 \xrightarrow{25/31} 0\,H_1 \qquad H_1 \xrightarrow{6/31} 0\,D_2$$

$$D_2 \xrightarrow{10/16} 5\,D_2 \qquad D_2 \xrightarrow{6/16} 5\,H_2$$

$$H_2 \xrightarrow{10/16} 0\,H_2 \qquad H_2 \xrightarrow{6/16} 0\,V_1$$

$$V_1 \xrightarrow{20/26} 6\,V_1 \qquad V_1 \xrightarrow{6/26} 6\,H_3$$

$$H_3 \xrightarrow{25/31} 4\,H_3 \qquad H_3 \xrightarrow{6/31} 4\,D_3$$

$$D_3 \xrightarrow{4/6} 3 \qquad D_3 \xrightarrow{2/6} \lambda$$

Index

Markov models

A *Markov models* is a tuple $M = (Q, \Sigma, \pi, A, B)$ where:

- Q is a ***set of states***
  - The model is in a state q_t at every instant $t = 1, 2, \ldots, M$.
  - Q includes a *final state* F

- Σ is a ***set of "observable" symbols*** (observations)
  The models emits a symbol y_t at every instant $t = 1, 2, \ldots, M$.

- $\pi \in \mathbb{R}^Q$ is the ***initial probability vector***:
  M chooses q_1 using π

- $A \in \mathbb{R}^{Q \times Q}$ is a ***transition probability matrix*** (between states):
  M chooses q_{t+1} using q_t and A: $A_{q,q'} = P(q_{t+1} = q' | q_t = q, A)$

- $B \in \mathbb{R}^{Q \times \Sigma}$ is a ***observation/emission probability matrix*** (probability that a state emits an observable symbol):
  M chooses y_t using q_t and B: $B_{q,\sigma} = P(y_t = \sigma \mid q_t = q, B)$

Markov models (cont.)

Normalization conditions for π, A, B:

- Probability of the initial state:

  $0 \leq \pi_q \leq 1, \quad \sum_{q \in Q} \pi_q = 1, \quad \pi_F = 0$

- Probabilities of transition between states:

  $0 \leq A_{q,q'} \leq 1, \quad \sum_{q' \in Q} A_{q,q'} = 1, \quad A_{F,q} = 0$

- Probabilities of emitting observable symbols:

  $0 \leq B_{q,\sigma} \leq 1, \quad \sum_{\sigma \in \Sigma} B_{q,\sigma} = 1, \quad B_{F,\sigma} = 0$

Markov models: example

.

Q={1,2,3,F}; Σ = {a,b,c}; $\pi_1 = 1$; $\pi_2 = \pi_3 = \pi_F = 0$

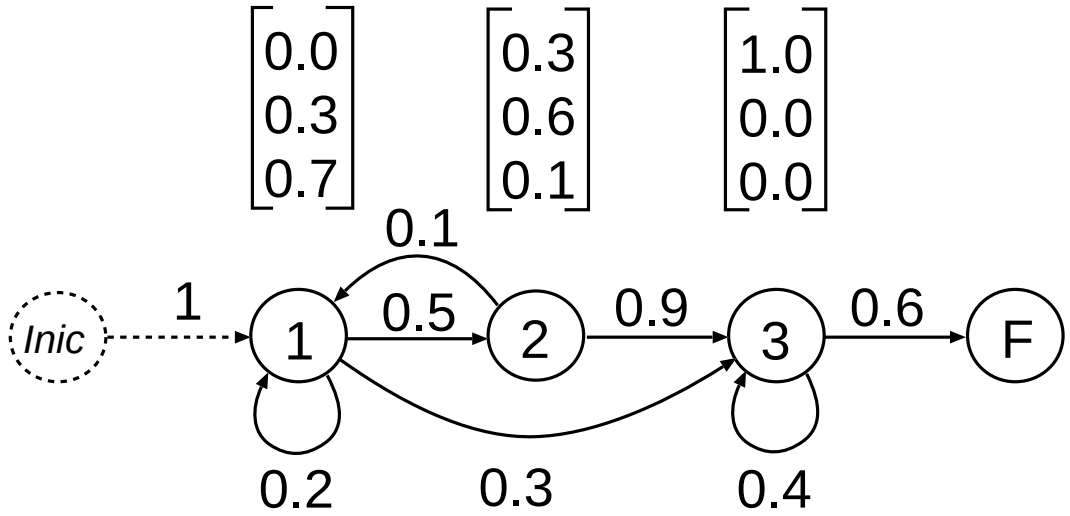
p(q'|q) ≡ A(q,q')

	1	2	3	F
1	0.2	0.5	0.3	0.0
2	0.1	0.0	0.9	0.0
3	0.0	0.0	0.4	0.6

p(σ|q) ≡ B(q,σ)

	a	b	c
1	0.0	0.3	0.7
2	0.3	0.6	0.1
3	1.0	0.0	0.0

Representación Gráfica Equivalente:



Markov models: example

Meaning of the probabilities:

$P(2|1) = A_{1,2} = 0.5$: probability of reaching state 2 given that we are in state 1 (present state)
$P(F|1) = A_{1,F} = 0$: probability of reaching state F given that we are in state 1 (present state)
$P(3|2) = A_{2,3} = 0.9$: probability of reaching state 3 given that we are in state 2 (present state)
...

$P(a|1) = B_{1,a} = 0$: probability of emitting observation (symbol) a coming from state 1
$P(c|2) = B_{2,c} = 0.1$: probability of emitting observation (symbol) c coming from state 2
$P(a|3) = B_{3,a} = 1$: probability of emitting observation (symbol) a coming from state 3
...

Probability of generating a sequence of states

A stochastic process has the Markov property if the conditional probability distribution of future states depends only upon the present state, not on the sequence of states that preceded it.

Through the application of the Markov chain property, the probability of a sequence of states is:

$$P(s_1, s_2, \ldots, s_n) = P(s_n|s_1, \ldots, s_{n-1})P(s_1, \ldots, s_{n-1}) = P(s_n|s_{n-1})P(s_1, \ldots, s_{n-1}) =$$
$$P(s_n|s_{n-1})P(s_{n-1}|s_1, \ldots, s_{n-2})P(s_1, \ldots, s_{n-2}) =$$
$$P(s_n|s_{n-1})P(s_{n-1}|s_{n-2})P(s_1, \ldots, s_{n-2}) = \ldots =$$
$$P(s_n|s_{n-1})P(s_{n-1}|s_{n-2})\ldots P(s_2|s_1)P(s_1) = P(s_1)\prod_{i=n}^{2} P(s_i|s_{i-1})$$

Examples from page 17:

Probability of generating the sequence of states $123F$:
$P(123F) = P(F|3)P(3|2)P(2|1)P(1) = A_{3,F} \cdot A_{2,3} \cdot A_{1,2} \cdot \pi_1 = 0.6 \cdot 0.9 \cdot 0.5 \cdot 1 = 0.27$

Probability of generating the sequence of states $113F$:
$P(113F) = P(F|3)P(3|1)P(1|1)P(1) = A_{3,F} \cdot A_{1,3} \cdot A_{1,1} \cdot \pi_1 = 0.6 \cdot 0.3 \cdot 0.2 \cdot 1 = 0.036$

Probability of generating a sequence of observations (string) with a sequence of states

Let $M = (Q, \Sigma, \pi, A, B)$ be a Markov model with a final state q_F:

1. *Choose an initial state* $q \in Q$ using $P(q) \equiv \pi_q$
2. *Select an observation* $\sigma \in \Sigma$ using $P(\sigma|q) \equiv B_{q,\sigma}$; emit σ
3. *Choose the next state* $q' \in Q$ using $P(q'|q) \equiv A_{q,q'}$
4. If $q = q_F$ *end*; else, ***go to step 2***

Let:

- $y = y_1, y_2, \ldots, y_m \in \Sigma^+$: be a sequence of observations produced by M
- $z = q_1, q_2, \ldots, q_F \in Q^+$: be a sequence of states that generate y

The probability that M produces the sequence of symbols (string) y with the sequence of states z is:

$$P(y, z) = P(z) \cdot P(y \,|\, z) = P(q_1) \prod_{t=2}^{m} P(q_t \,|\, q_{t-1})\, P(q_F \,|\, q_m) \cdot \prod_{t=1}^{m} P(y_t \,|\, q_t)$$

Example of the probability of generating a string with a sequence of states

Which is the probability of generating the string cba with the sequence of states 113?

$$P(y,z) = P(z) \cdot P(y \mid z) = P(q_1) \prod_{t=2}^{m} P(q_t \mid q_{t-1})\, P(q_F \mid q_m) \cdot \prod_{t=1}^{m} P(y_t \mid q_t)$$

$$P(cba, 113F) = P(cba|113F)P(113F) = P(c|1)P(b|1)P(a|3)P(113F) =$$

$$P(c|1)P(b|1)P(a|3)P(F|3)P(3|1)P(1|1)P(1) =$$

$$0.7 \cdot 0.3 \cdot 1.0 \cdot 0.6 \cdot 0.3 \cdot 0.2 \cdot 1 = 0.0076$$

Probability of generating a string (sequence of observations) with a Markov model

Probability that M generates the string $y = y_1 \dots y_m \in \Sigma^+$:

$$P(y \mid M) = \sum_{z \in Q^+} P(y, z)$$

$$= \sum_{q_1,\dots,q_m \in Q^+} P(q_1) \prod_{t=2}^{m} P(q_t \mid q_{t-1}) \; P(q_F \mid q_m) \cdot \prod_{t=1}^{m} P(y_t \mid q_t)$$

$$= \sum_{q_1,\dots,q_m \in Q^+} \pi_{q_1} B_{q_1,y_1} \left(\prod_{t=2}^{m} A_{q_{t-1},q_t} B_{q_t,y_t} \right) A_{q_m,q_F}$$

The following holds:

$$0 \leq P(y|M) \leq 1, \qquad \sum_{y \in \Sigma^+} P(y|M) = 1$$

Example of generating a string with a Markov Model (1)

Which is the probability of generating the string cba with the Markov Model in page 17?

First, we have to find all possible sequences of states that generate the string cba. We annotate the states that can generate the symbols of the string:

c	b	a
1	1	2
2	2	3

Observations:

- The only state that reaches the final state F is 3, so the sequence cannot end with state 2
- The only initial state is 1, so the sequence of states cannot start with state 2

c	b	a
1	1	
	2	3

Sequences of states that generate cba: 113 and 123

$$P(cba|M) = P(cba, 113F) + P(cba, 123F) =$$
$$P(cba|113F)P(113F) + P(cba|123F)P(123F)$$
$$P(c|1)P(b|1)P(a|3)P(113F) + P(c|1)P(b|2)P(a|3)P(123F) =$$
$$0.0076 + 0.1134 = 0.121 \approx \quad 0.12$$

Example of generating a string with a Markov Model (2)

Which is the probability of generating the string $bcbaa$ with the Markov Model in page 17?

First, we have to find all possible sequences of states that generate the string cba. We annotate the states that can generate the symbols of the string:

b	c	b	a	a
1	1	1	2	2
2	2	2	3	3

Observations:

- The only state that reaches the final state F is 3, so the sequence cannot end with state 2
- The only initial state is 1, so the sequence of states cannot start with state 2
- State 2 cannot repeat itself so this discards combinations 22

b	c	b	a	a
1	1	1	2	
	2	2	3	3

Valid combinations of states $bcbaa$: 11123, 11133, 11233, 12123, 12133

$$P(\texttt{bcbaa}|M) =$$
$$P(bcbaa, 11123F) + P(bcbaa, 11133F) + P(bcbaa, 11233F) + P(bcbaa, 12123F) +$$
$$P(bcbaa, 12133F) = \ldots$$

Example of generating a string with a Markov Model (3)

$$
\begin{array}{rcccccl}
P(\mathtt{cba}|M) & = & (\pi_1 \cdot B_{1,\mathtt{c}}) & (A_{1,2} \cdot B_{2,\mathtt{b}}) & (A_{2,3} \cdot B_{3,\mathtt{a}}) & A_{3,F} & \\
& + & (\pi_1 \cdot B_{1,\mathtt{c}}) & (A_{1,1} \cdot B_{1,\mathtt{b}}) & (A_{1,3} \cdot B_{3,\mathtt{a}}) & A_{3,F} & \\
& = & (1 \cdot 0.7) & (0.5 \cdot 0.6) & (0.9 \cdot 1) & 0.6 & \\
& + & (1 \cdot 0.7) & (0.2 \cdot 0.3) & (0.3 \cdot 1) & 0.6 & = 0.1134 + 0.00756 \\
& & & & & & \approx 0.12
\end{array}
$$

$$P(\mathtt{bcbaa}|M) = P(y, z_1) + P(y, z_2) + P(y, z_3) + P(y, z_4) + P(y, z_5)$$

$$
\begin{array}{rccccccl}
y = & \mathtt{b} & \mathtt{c} & \mathtt{b} & \mathtt{a} & \mathtt{a} & & \\
z_1 = & 1 & 1 & 1 & 2 & 3 & F & \\
P(y, z_1) = & (1 \cdot 0.3) & (0.2 \cdot 0.7) & (0.2 \cdot 0.3) & (0.5 \cdot 0.3) & (0.9 \cdot 1) & 0.6 & = 0.000204 \\
z_2 = & 1 & 1 & 1 & 3 & 3 & F & \\
P(y, z_2) = & (1 \cdot 0.3) & (0.2 \cdot 0.7) & (0.2 \cdot 0.3) & (0.3 \cdot 1) & (0.4 \cdot 1) & 0.6 & = 0.000181 \\
z_3 = & 1 & 1 & 2 & 3 & 3 & F & \\
P(y, z_3) = & (1 \cdot 0.3) & (0.2 \cdot 0.7) & (0.5 \cdot 0.6) & (0.9 \cdot 1) & (0.4 \cdot 1) & 0.6 & = 0.002722 \\
z_4 = & 1 & 2 & 1 & 2 & 3 & F & \\
P(y, z_4) = & (1 \cdot 0.3) & (0.5 \cdot 0.1) & (0.1 \cdot 0.3) & (0.5 \cdot 0.3) & (0.9 \cdot 1) & 0.6 & = 0.000036 \\
z_5 = & 1 & 2 & 1 & 3 & 3 & F & \\
P(y, z_5) = & (1 \cdot 0.3) & (0.5 \cdot 0.1) & (0.1 \cdot 0.3) & (0.3 \cdot 1) & (0.4 \cdot 1) & 0.6 & = 0.000032
\end{array}
$$

$$P(y|M) = 0.003175$$

Index

Exercise

Let M be a Markov model with states $Q = \{1, 2, 3, 4, 5, F\}$; alphabet (observable symbols) $\Sigma = \{a, b\}$; initial probabilities $\pi_1 = 1$, $\pi_2 = \pi_3 = \pi_4 = \pi_5 = 0$; and transition and observation probability matrix:

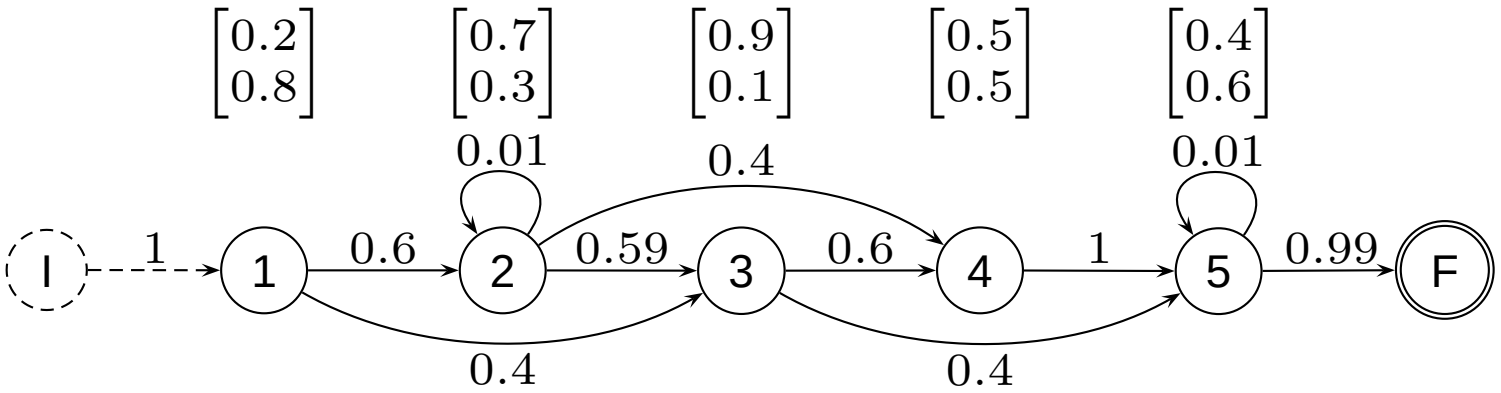
A	1	2	3	4	5	F
1		0.6	0.4			
2		0.01	0.59	0.4		
3				0.6	0.4	
4					1.0	
5					0.01	0.99

B	a	b
1	0.2	0.8
2	0.7	0.3
3	0.9	0.1
4	0.5	0.5
5	0.4	0.6

1. Represent M graphically.
2. Calculate the probability that M generates a sequence of 3 observable symbols.
3. Can we say that the most likely strings are sequences whose length is between 3 and 5 symbols?

Exercise (solution)

1)



2) L=length of the string. There are 8 strings such that $L = 3 : aaa, aab, aba, abb,$ baa, bab, bba, bbb. There is only one possible state sequence to generate the 8 strings: $1, 3, 5, F$.

$$
\begin{aligned}
P(L = 3 \mid M) &= \sum_{y \in \{a,b\}^3} P(y \mid M) = \sum_{y \in \{a,b\}^3} \pi_1 \, B_{1,y_1} \left(A_{1,3} \, B_{3,y_3} \; A_{3,5} \, B_{5,y_5} \right) \; A_{5,F} \\
&= \pi_1 \, A_{1,3} \, A_{3,5} \, A_{5,F} \sum_{y \in \{a,b\}^3} \left(B_{1,y_1} \, B_{3,y_3} \, B_{5,y_5} \right) \\
&= 0.1584 \left(B_{1,a} \, B_{2,a} \, B_{3,a} + B_{1,a} \, B_{2,a} \, B_{3,b} + \cdots + B_{1,b} \, B_{2,b} \, B_{3,b} \right) \\
&= 0.1584 \left(B_{1,a} + B_{1,b} \right) \left(B_{2,a} + B_{2,b} \right) \left(B_{3,a} + B_{3,b} \right) \\
&= 0.1584 \cdot 1 \cdot 1 \cdot 1 \; = \; \mathbf{0.1584}
\end{aligned}
$$

3) ***Yes***. The products involving $A_{2,2}$ or $A_{5,5}$ (both $= \; 0.01$) do not really affect the sumations. a) $P(L = l \mid M) = \mathbf{0} \; \forall l < 3$; b) $P(L = 4 \mid M) = 1 \cdot 0.6 \cdot 0.4 \cdot 1 \cdot 0.99 \; + 1 \cdot 0.4 \cdot 0.6 \cdot 1 \cdot 0.99 \; + 1 \cdot 0.6 \cdot 0.59 \cdot 0.4 \cdot 0.99 + \ldots \approx \mathbf{0.62}$; c) $P(L = 5 \mid M) = 1 \cdot 0.6 \cdot 0.59 \cdot 0.6 \cdot 1 \cdot 0.99 + \ldots \approx \mathbf{0.21}$; d) $\forall l \geq 6$, $P(L = l \mid M) < \mathbf{0.005}$, and the probability decreases exponentially with l.

Index

Equivalence between Markov models and stochastic regular grammars

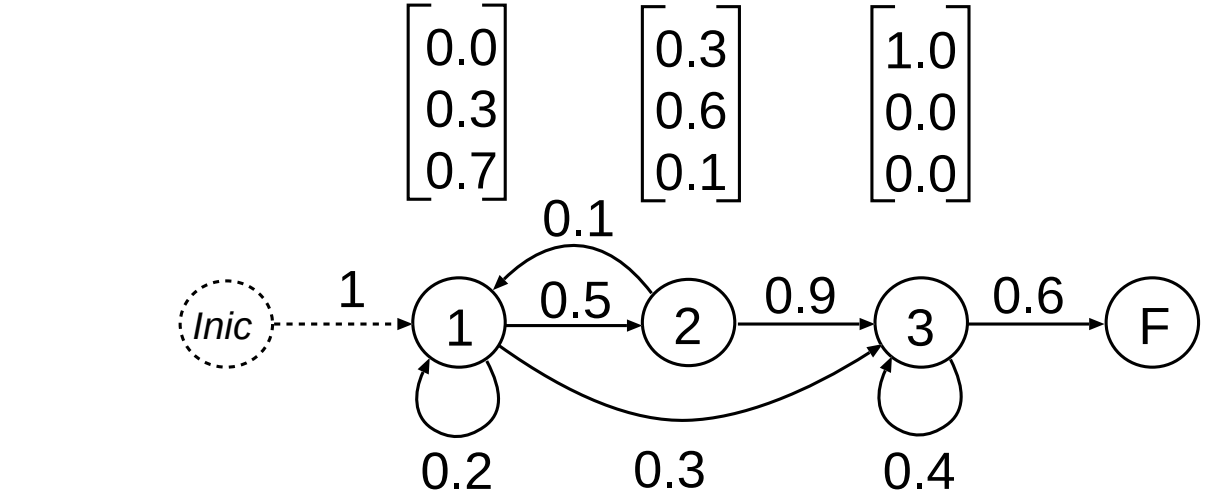
Given a Markov model M, there exists a stochastic regular grammar G such that $P(y|M) = P(y|G)\ \forall y \in \Sigma^*$

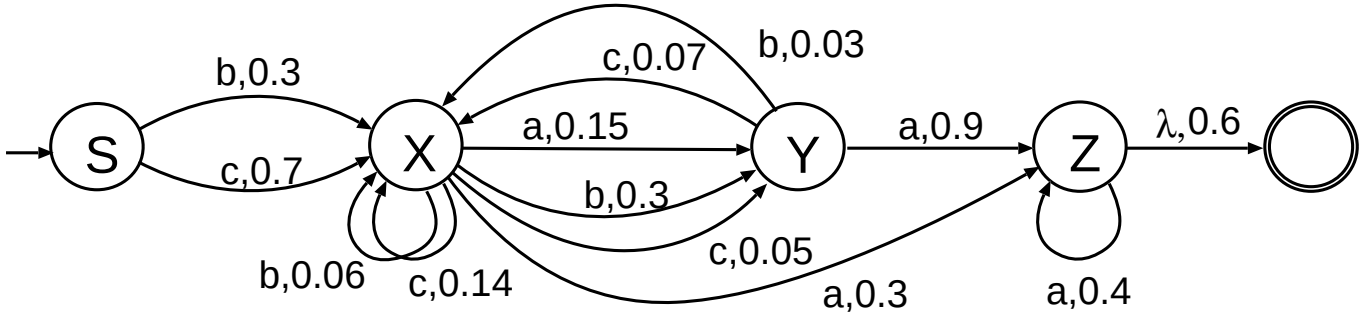
It is easily proved by construction.

Given a stochastic regular grammar G, there exists a Markov model M such that $P(y|M) = P(y|G)\ \forall y \in \Sigma^*$ (except for degenerated cases)

It is proved by construction (a bit more complex than in the previous case).

Equivalence between Markov models and stochastic grammars





0.3: S -> bX
0.7: S -> cX

0.06: X -> bX
0.14: X -> cX
0.15: X -> aY
0.30: X -> bY
0.05: X -> cY
0.30: X -> aZ

0.90: Y -> aZ
0.03: Y -> bX
0.07: Y -> cX

0.4: Z -> aZ
0.6: Z -> λ

Index

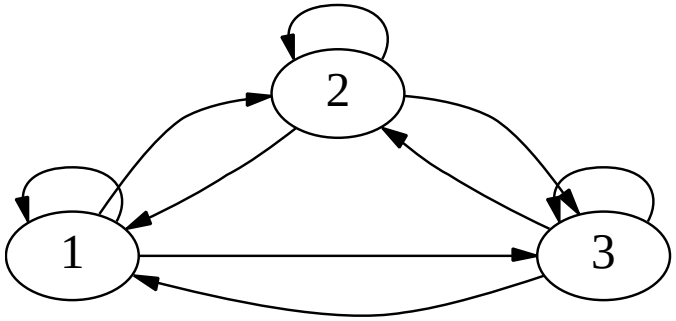
Structure or "topology" of a Markov model

The *topology* of a Markov model is the underlying graph. It is determined by the structure (number and location of zeroes in the state-transition matrix A). The most common topologies are:

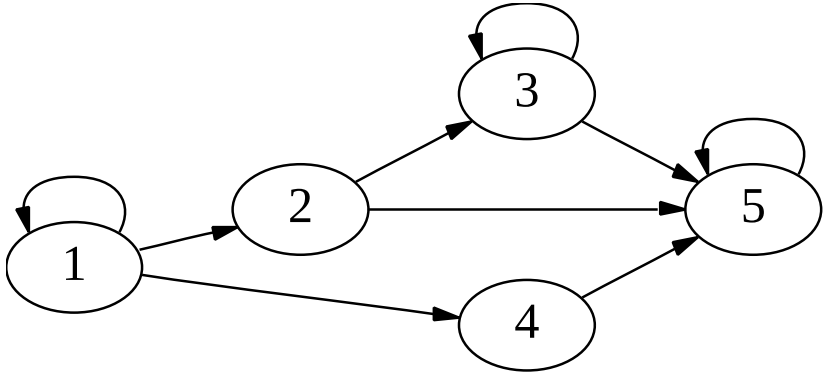
- ***Ergodic:*** complete graph, no zeroes in A.
- ***Left-to-right:*** the graph is *directed and acyclic* (DAG), though there can be individual loops in the states. A is triangular.
- ***Linear:*** the graph is a restricted DAG (possibly with loops in the states) where transitions leaving the i-th state can only reach states $i+1, \ldots, i+k$. The non-null elements in A are in $k+1$ adjacent diagonals. These transitions are called *skips*.
- ***Strictly linear:*** the graph is a concatenation of states (possibly with loops in the states). The non-null elements in A are in two adjacent diagonals.

Examples of topologies of Markov models

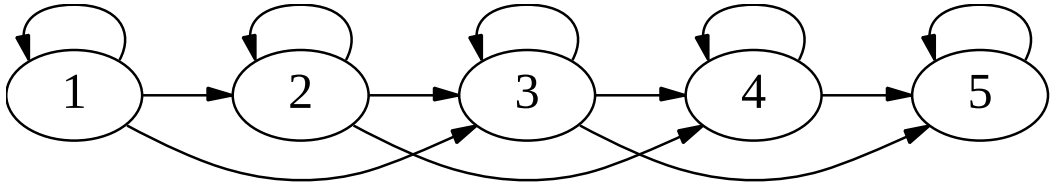
Ergodic



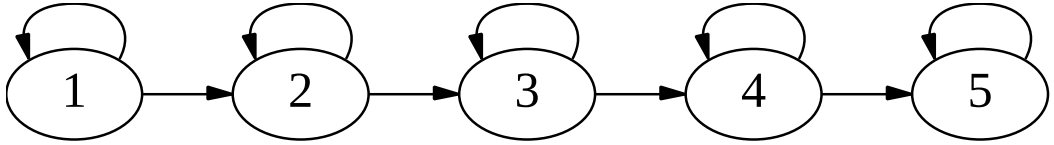
Left-to-right



Linear



Strictly linear



Index

Grammars

- ***Free Monoid*** Σ^****:*** Given a finite set Σ, Σ^+ is the set of all finite-length strings of elements that belong to Σ. Moreover, $\Sigma^* = \Sigma^+ \cup \{\lambda\}$ (the *empty string*).

- ***Grammar:*** $G = (N, \Sigma, R, S)$
  - N: Finite set of *non-terminals*
  - Σ: Finite set of *terminals or primitives*
  - $S \in N$: Initial non-terminal symbol or *"Axiom"*
  - $R \subset (N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$: set of *rules*.

  A rule is written as:

  $\boldsymbol{\alpha \rightarrow \beta}, \quad \alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^*, \ \ \beta \in (N \cup \Sigma)^*$

  If many rules share their left part, they can be written in short as:

  $\boldsymbol{\alpha \rightarrow \ \beta_1 \mid \beta_2 \mid \cdots}$

Grammars and languages

- ***Elementary derivation :*** $\underset{G}{\Longrightarrow}$***:***

  $\mu\,\alpha\,\delta \underset{G}{\Longrightarrow} \mu\,\beta\,\delta$ sii $\exists(\alpha \to \beta) \in R,\ \ \mu, \delta \in (N \cup \Sigma)^*$

- ***Derivation*** $\overset{*}{\underset{G}{\Longrightarrow}}$***:***

  A derivation is a *finite sequence of elementary derivations*. A derivation d can be expressed as the corresponding sequence of rules in G.

  The *derivation set* of $y \in \Sigma^*$ (such that $S \overset{*}{\underset{G}{\Longrightarrow}} y$) is denoted as $D_G(y)$.

  A grammar G is *ambiguous* if $\exists y \in \Sigma^*$ such that $|D_G(y)| > 1$

- ***A language generated by a grammar*** G, $\mathcal{L}(G)$***:***

$$\mathcal{L}(G) = \{\, y \in \Sigma^* \mid S \overset{*}{\underset{G}{\Longrightarrow}} y \,\}$$

Types of grammars and languages

CHOMSKY HIERARCHY FOR RECURSIVE LANGUAGES

0: No-restricted languages

1: Context languages

$$\alpha \rightarrow \beta \ , \qquad\qquad |\alpha| \ \leq \ |\beta|$$

2: Context-free languages

$$B \rightarrow \beta \ , \qquad\qquad B \in N$$

3: ***Regular of "finite-state" languages***

$$A \rightarrow aB \quad \text{o} \quad A \rightarrow a \ , \quad A, B \in N, \ a \in \Sigma \cup \{\lambda\}$$

Regular grammars and finite automata

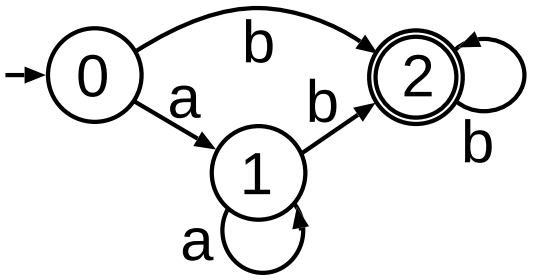
- ***Regular grammars:*** $G = (N, \Sigma, R, S)$,
  Rules in R of the form: $A \to aB \ \vee \ A \to a, \quad A, B \in N, \quad a \in \Sigma$

- ***Finite automata:*** $\mathcal{A} = (Q, \Sigma, \delta, q_0, F), \quad q_0 \in Q, \quad F \subseteq Q, \quad \delta : Q \times \Sigma \to 2^Q$

- ***Equivalence:*** For every regular grammar there exists a finite automaton that recognizes the same language (*Warning! The reverse is not always true for* **stochastic** *languages!*).

Example:

$$
\begin{aligned}
&G = (N, \Sigma, R, S);\\
&\Sigma = \{a, b\}; \ N = \{S, A_1, A_2\};\\
&R = \{ \ S \ \to aA_1 \mid bA_2 \mid b,\\
&\qquad A_1 \to aA_1 \mid bA_2 \mid b,\\
&\qquad A_2 \to bA_2 \mid b\}
\end{aligned}
$$



$$
\begin{aligned}
&\mathcal{A} = \{Q, \Sigma, \delta, q_0, F\};\\
&Q = \{0, 1, 2\},\\
&\Sigma = \{a, b\},\\
&q_0 = 0, \ F = \{2\}
\end{aligned}
$$

$$\mathcal{L}(G) = \{b, ab, bb, aab, abb, bbb, \ldots, aaabbbb, \ldots\} = \mathcal{L}(\mathcal{A})$$