Language models, Markov chains, hidden Markov models and profiles

Rahul Siddharthan

The Institute of Mathematical Sciences, Chennai, India

iCEL Workshop on Computational Epigraphy, 25 March 2024

Probability

What is “probability”? For example, what do the following mean:

Probability

What is “probability”? For example, what do the following mean:

- The probability of getting “heads” in a coin toss is 0.5

Probability

What is "probability"? For example, what do the following mean:

- The probability of getting "heads" in a coin toss is 0.5
- There is a 30% probability of rain today

Probability

What is “probability”? For example, what do the following mean:

- The probability of getting “heads” in a coin toss is 0.5
- There is a 30% probability of rain today
- There is an 80% probability of NDA forming the next government

Probability: a measure of how likely it is that a proposition is true.

Probability: a measure of how likely it is that a proposition is true.

Frequentist definition

n/N where n = number of occurrences in a large number of "independent, identically distributed" (i.i.d.) trials N.

- "Independent" = one trial does not affect another trial (more precise definition later)
- "Identically distributed" = (one expects) each trial to behave the same way

Probability: a measure of how likely it is that a proposition is true.

Frequentist definition

n/N where n = number of occurrences in a large number of "independent, identically distributed" (i.i.d.) trials N.

- "Independent" = one trial does not affect another trial (more precise definition later)
- "Identically distributed" = (one expects) each trial to behave the same way

Applies to coin tosses... but not to weather, cricket matches, elections!

Bayesian definition

A real number between 0 and 1 quantifying your degree of belief in a proposition.

This made many 20th-century statisticians very uncomfortable... but this methodology is widely accepted now

Probability theory: Some definitions

- **Conditional probability**: The probability of A given that B has occurred.

Probability theory: Some definitions

- **Conditional probability**: The probability of A given that B has occurred.
- **Joint probability**: The probability of both A and B occurring.

$$P(AB) = P(A|B)P(B) = P(B|A)P(A)$$

$$P(A \text{ OR } B) = P(A) + P(B) - P(AB)$$

Probability theory: Some definitions

- **Conditional probability**: The probability of A given that B has occurred.
- **Joint probability**: The probability of both A and B occurring.

$$P(AB) = P(A|B)P(B) = P(B|A)P(A)$$

$$P(A \text{ OR } B) = P(A) + P(B) - P(AB)$$

- **Likelihood**: The probability of a hypothesis given the observed data.

Sequences

- Biology:
  `CTGACAGAGACACCCGATTACTGATTTGGGAAATTTCCCAAATTGGAAATA...`

Sequences

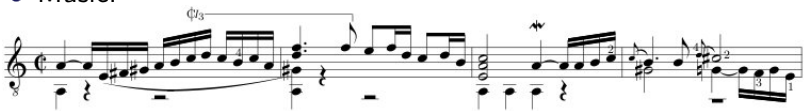
- Biology:
  `CTGACAGAGACACCCGATTACTGATTTGGGAAATTTCCCAAATTGGAAATA...`
- Language:
  `Persons attempting to find a motive in this narrative will be prosecuted; persons attempting to find a moral in it will be banished; persons attempting to find a plot in it will be shot.`

Sequences

- Biology:
  `CTGACAGAGACACCCGATTACTGATTTGGGAAATTTCCCAAATTGGAAATA...`
- Language:
  `Persons attempting to find a motive in this narrative will be prosecuted; persons attempting to find a moral in it will be banished; persons attempting to find a plot in it will be shot.`
  - Letter-level:
    `[P,e,r,s,o,n,s, ,a,...]`
  - Word-level:
    `[Persons, attempting, to, find...]`

Sequences

- Biology:
  `CTGACAGAGACACCCGATTACTGATTTGGGAAATTTCCCAAATTGGAAATA...`
- Language:
  `Persons attempting to find a motive in this narrative will be prosecuted; persons attempting to find a moral in it will be banished; persons attempting to find a plot in it will be shot.`
  - Letter-level:
    `[P,e,r,s,o,n,s, ,a,...]`
  - Word-level:
    `[Persons, attempting, to, find...]`
- Music:



What's common to those sequences?

- They can be written as linear sequences of a discrete, finite set of symbols ("alphabet")
- They can be very long
- They contain meaning
- At short scales, they contain correlations but are not perfectly ordered
- At longer scales, they are uncorrelated
- And more...

Why model sequences?

- Biologists want to understand the function of DNA (and protein) sequence, and design synthetic functional sequence
- Computational linguists want to use computers to parse, process, and create "natural language"
- Computer-created text and music conveys a better understanding of what goes into the "real" stuff
- Scholars want to compare and analyse works, assess authenticity, etc
- And so on...

Questions

Basic questions

Given a "model" that describes the sequence,

- What is the probability ("likelihood") of observing a particular sequence?
- Given a sequence, how do you predict ("generate") the next element?

Questions

Basic questions

Given a "model" that describes the sequence,

- What is the probability ("likelihood") of observing a particular sequence?
- Given a sequence, how do you predict ("generate") the next element?

If we can do the above:

Given multiple models for generating a sequence, how do we choose the more probable model?

Language models

Definition

A probability distribution $p(s)$ over strings s that attempts to reflect how frequently a string s occurs as a sentence.

Chen and Goodman, 1998

Markov chains

- A Markov chain is a sequence of symbols where each symbol depends only on its predecessor (or n predecessors)

Markov chains

- A Markov chain is a sequence of symbols where each symbol depends only on its predecessor (or n predecessors)
- Consider a sequence of symbols

$$S_1 S_2 S_3 S_4 S_5$$

What is the probability of observing this sequence?

Markov chains

- A Markov chain is a sequence of symbols where each symbol depends only on its predecessor (or n predecessors)
- Consider a sequence of symbols

$$S_1 S_2 S_3 S_4 S_5$$

What is the probability of observing this sequence?

  - "Exact" answer:

$$P(S_1)P(S_2|S_1)P(S_3|S_1S_2)P(S_4|S_1S_2S_3)P(S_5|S_1S_2S_3S_4)$$

Markov chains

- A Markov chain is a sequence of symbols where each symbol depends only on its predecessor (or n predecessors)
- Consider a sequence of symbols

$$S_1 S_2 S_3 S_4 S_5$$

What is the probability of observing this sequence?

- "Exact" answer:

$$P(S_1)P(S_2|S_1)P(S_3|S_1S_2)P(S_4|S_1S_2S_3)P(S_5|S_1S_2S_3S_4)$$

- Markov approximation:

$$P(S_1)P(S_2|S_1)P(S_3|S_2)P(S_4|S_3)P(S_5|S_4)$$

Example: DNA sequence

If the DNA is “non-functional”, how to model it?

Example: DNA sequence

If the DNA is "non-functional", how to model it?

- Simplest model: each nucleotide occurs independently with a certain probability. Eg, $P(\mathrm{A}) = P(\mathrm{T}) = 0.3, P(\mathrm{C}) = P(\mathrm{G}) = 0.2$ (4 probabilities)
- However, dinucleotides in DNA are not distributed according to this model!

Example: DNA sequence

If the DNA is "non-functional", how to model it?

- Simplest model: each nucleotide occurs independently with a certain probability.
  Eg, $P(\mathrm{A}) = P(\mathrm{T}) = 0.3, P(\mathrm{C}) = P(\mathrm{G}) = 0.2$ (4 probabilities)
- However, dinucleotides in DNA are not distributed according to this model!
- Next simplest: Each nucleotide depends on its predecessor
  $P(\text{A at site 2}|\text{C at site 1})$, etc. (16 such "conditional probabilities"), "Markov chain"
- Still doesn't account for "trigrams"

Example: DNA sequence

If the DNA is "non-functional", how to model it?

- Simplest model: each nucleotide occurs independently with a certain probability.
  Eg, $P(\mathrm{A}) = P(\mathrm{T}) = 0.3, P(\mathrm{C}) = P(\mathrm{G}) = 0.2$ (4 probabilities)
- However, dinucleotides in DNA are not distributed according to this model!
- Next simplest: Each nucleotide depends on its predecessor
  $P(\mathrm{A}$ at site $2|\mathrm{C}$ at site $1)$, etc. (16 such "conditional probabilities"), "Markov chain"
- Still doesn't account for "trigrams"
- Each nucleotide depends on immediate two predecessors
  $P(\mathrm{A}|\mathrm{CG})$, etc (64 conditional probabilities), 2nd order Markov chain

Example: DNA sequence

If the DNA is "non-functional", how to model it?

- Simplest model: each nucleotide occurs independently with a certain probability.
  Eg, $P(\mathrm{A}) = P(\mathrm{T}) = 0.3, P(\mathrm{C}) = P(\mathrm{G}) = 0.2$ (4 probabilities)
- However, dinucleotides in DNA are not distributed according to this model!
- Next simplest: Each nucleotide depends on its predecessor
  $P(\mathrm{A}$ at site $2|\mathrm{C}$ at site $1)$, etc. (16 such "conditional probabilities"), "Markov chain"
- Still doesn't account for "trigrams"
- Each nucleotide depends on immediate two predecessors
  $P(\mathrm{A}|\mathrm{CG})$, etc (64 conditional probabilities), 2nd order Markov chain

If not good enough: go to higher-order Markov.

Example: DNA sequence

If the DNA is "non-functional", how to model it?

- Simplest model: each nucleotide occurs independently with a certain probability.
  Eg, $P(\mathrm{A}) = P(\mathrm{T}) = 0.3, P(\mathrm{C}) = P(\mathrm{G}) = 0.2$ (4 probabilities)
- However, dinucleotides in DNA are not distributed according to this model!
- Next simplest: Each nucleotide depends on its predecessor
  $P(\mathrm{A}$ at site $2|\mathrm{C}$ at site $1)$, etc. (16 such "conditional probabilities"), "Markov chain"
- Still doesn't account for "trigrams"
- Each nucleotide depends on immediate two predecessors
  $P(\mathrm{A}|\mathrm{CG})$, etc (64 conditional probabilities), 2nd order Markov chain

If not good enough: go to higher-order Markov.

Drawback: for alphabet size ℓ, there are ℓ^n n-grams! Lots of data needed to estimate these.

Example: Shannon, 1948

(C. E. Shannon, *A mathematical theory of communication*, 1948)

3. The Series of Approximations to English

To give a visual idea of how this series of processes approaches a language, typical sequences in the approximations to English have been constructed and are given below. In all cases we have assumed a 27-symbol "alphabet," the 26 letters and a space.

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order approximation (digram structure as in English).

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

Example: Shannon, 1948

(C. E. Shannon, *A mathematical theory of communication*, 1948)

5. First-order word approximation. Rather than continue with tetragram, . . . , n-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation. The word transition probabilities are correct but no further structure is included.

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

The resemblance to ordinary English text increases quite noticeably at each of the above steps. Note that these samples have reasonably good structure out to about twice the range that is taken into account in their construction. Thus in (3) the statistical process insures reasonable text for two-letter sequences, but four-letter sequences from the sample can usually be fitted into good sentences. In (6) sequences of four or more words can easily be placed in sentences without unusual or strained constructions. The particular sequence of ten words "attack on an English writer that the character of this" is not at all unreasonable. It appears then that a sufficiently complex stochastic process will give a satisfactory representation of a discrete source.

The zero problem, and smoothing

If you are training from insufficient data, some possible observations may never occur in your data.

Simple example: coin-tossing:

Suppose you have a possibly unfair coin, toss it N times, and see n heads. What is the probability of seeing heads on the next toss?
"Maximum likelihood" answer $= \frac{n}{N}$. Bad answer!

The zero problem, and smoothing

If you are training from insufficient data, some possible observations may never occur in your data.

Simple example: coin-tossing:

Suppose you have a possibly unfair coin, toss it N times, and see n heads. What is the probability of seeing heads on the next toss?
"Maximum likelihood" answer $= \frac{n}{N}$. Bad answer!

Laplace's rule of succession

If you are completely ignorant about the coin's bias, the answer is

$$P(\text{heads}) = \frac{n+1}{N+2}$$

(doesn't usually apply to real coins!)

The zero problem, and smoothing

Laplace's rule: Generalization

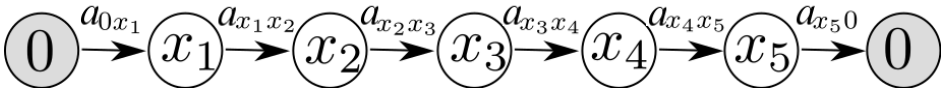
If there are ℓ possible symbols that you can observe, and in N observations you have observed the i'th symbol n_i times; Answer, assuming complete independence of observations and complete ignorance,

$$P(i) = \frac{n_i + 1}{N + \ell}$$

Hidden Markov models

In Markov models

symbols are emitted probabilistically based on the previous symbol.



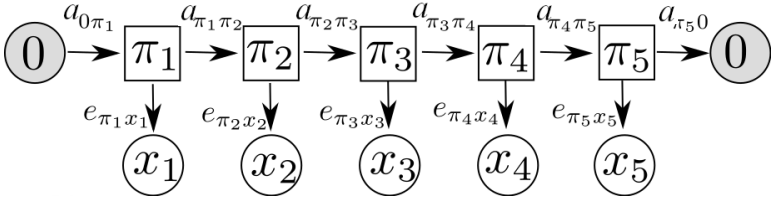
Hidden Markov models

In Markov models

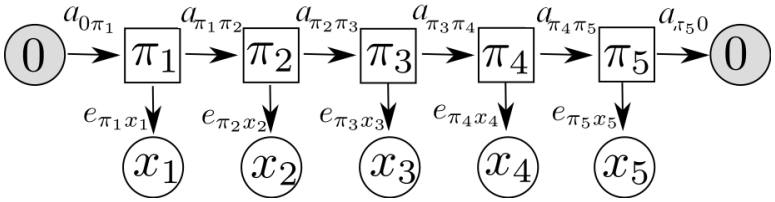
symbols are emitted probabilistically based on the previous symbol.

In hidden Markov models

an invisible "state" follows a Markov process. This state emits symbols that are visible. Each hidden state emits a different pattern of symbols.



Hidden Markov models



Tasks

- Infer hidden states (Viterbi algorithm)
- Infer likelihood of sequence (forward/backward algorithms)

Hidden Markov models

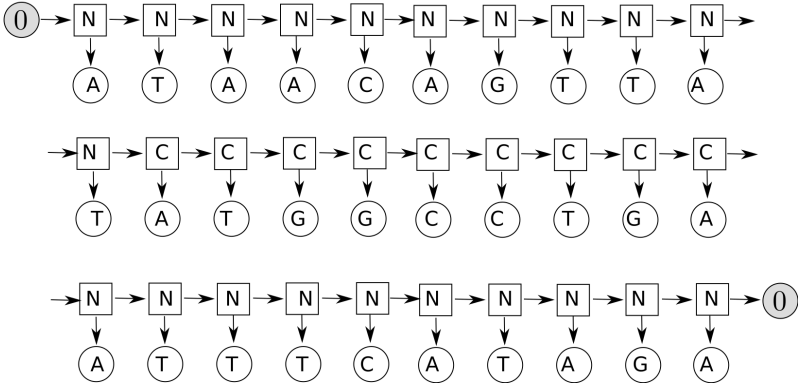
Silly example

Ma jolie, how do you do?
Mon nom est Jean-Guy Thibault-Leroux
I come from east of Gatineau
My name is Jean-Guy, ma jolie
J'ai une maison à Lafontaine
Where we can live, if you marry me
Une belle maison à Lafontaine
Where we will live, you and me
Oh Louise, ma jolie Louise ...

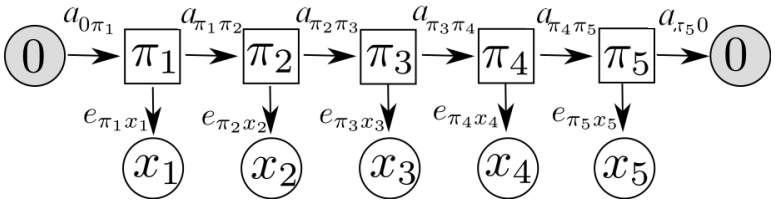
(Isabelle Boulay / written by Daniel Lanois)

Somewhat more realistic example

Imagine a very simple model of DNA where there are two kinds of regions – coding (C) and noncoding (N) – characterised mainly by differences in nucleotide densities. Non-coding regions are AT rich, coding regions are more GC rich. This is a possible (but oversimplified) HMM.



Most probable path: Viterbi algorithm



Given a sequence $x = x_1 x_2 \dots x_n$ and a set of hidden states $\pi = \pi_1 \pi_2 \dots \pi_n$, we can calculate the joint likelihood

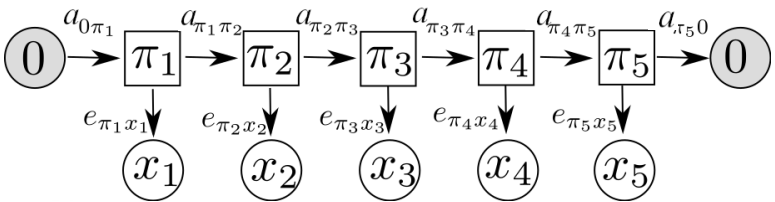
$$P(x, \pi) = a_{0\pi_1} \left(\prod_{i=1}^{n} e_{\pi_i x_i} a_{\pi_i \pi_{i+1}} \right)$$

where $\pi_{n+1} = 0$.

How do we find the *most probable path*

$$\pi^* = \mathrm{argmax}_\pi P(x, \pi)?$$

Viterbi algorithm: example



Suppose someone gives you a sequence of coin tosses, but the person has two coins and is randomly switching from one coin to another. One coin is fair ("F"). The other coin is biased ("B") and tosses heads 80% of the time. After each toss, the player can keep the same coin (probability 0.8), switch to the other coin (0.15), or end the game (0.05). Also, the probability of starting with the fair coin is 0.8.

The sequence of tosses is HTTHHHHHTH. What is the most probable hidden path?

Viterbi algorithm: example

i \ k	F	B	x_i
1	0.4	0.12	H
2			T
3			T
4			H
5			H
6			H
7			H
8			H
9			T
10			H

Define a matrix v_{ki} = probability of most probable path up until state i, if $\pi_i = k$. Then $v_{l,i+1} = e_{l x_{i+1}} \max_k (v_{ki} a_{kl})$.

- Initialize $v_{00} = 1$, $v_{k0} = 0 \forall k > 0$.
- Fill in the matrix until the bottom right, *each time pointing back to the previous row entry that gave the best answer*
- Trace back arrows from largest entry on bottom row

Viterbi algorithm: example

i \ k	F	B	x_i
1	0.4	0.12	H
2	0.16		T
3			T
4			H
5			H
6			H
7			H
8			H
9			T
10			H

Define a matrix v_{ki} = probability of most probable path up until state i, if $\pi_i = k$. Then $v_{l,i+1} = e_{lx_{i+1}} \max_k (v_{ki} a_{kl})$.

- Initialize $v_{00} = 1$, $v_{k0} = 0 \forall k > 0$.
- Fill in the matrix until the bottom right, *each time pointing back to the previous row entry that gave the best answer*
- Trace back arrows from largest entry on bottom row

Viterbi algorithm: example

i \ k	F	B	x_i
1	0.4	0.12	H
2	0.16	0.0192	T
3			T
4			H
5			H
6			H
7			H
8			H
9			T
10			H

Define a matrix v_{ki} = probability of most probable path up until state i, if $\pi_i = k$. Then $v_{l,i+1} = e_{lx_{i+1}} \max_k (v_{ki} a_{kl})$.

- Initialize $v_{00} = 1$, $v_{k0} = 0 \forall k > 0$.
- Fill in the matrix until the bottom right, *each time pointing back to the previous row entry that gave the best answer*
- Trace back arrows from largest entry on bottom row

Viterbi algorithm: example

i \ k	F	B	x_i
1	0.4	0.12	H
2	0.16	0.0192	T
3	0.064		T
4			H
5			H
6			H
7			H
8			H
9			T
10			H

Define a matrix v_{ki} = probability of most probable path up until state i, if $\pi_i = k$. Then $v_{l,i+1} = e_{lx_{i+1}} \max_k (v_{ki} a_{kl})$.

- Initialize $v_{00} = 1$, $v_{k0} = 0 \forall k > 0$.
- Fill in the matrix until the bottom right, *each time pointing back to the previous row entry that gave the best answer*
- Trace back arrows from largest entry on bottom row

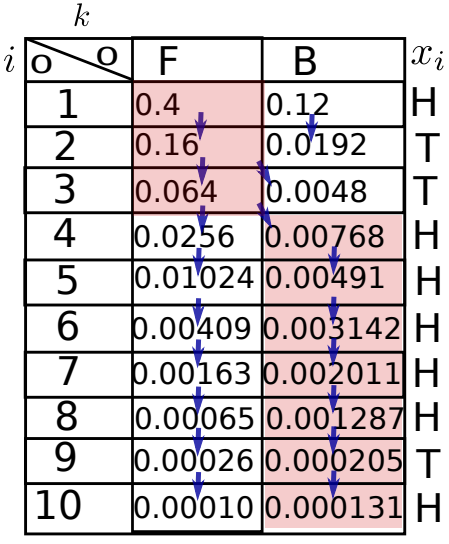
Viterbi algorithm: example

i \ k	F	B	x_i
1	0.4	0.12	H
2	0.16	0.0192	T
3	0.064	0.0048	T
4			H
5			H
6			H
7			H
8			H
9			T
10			H

Define a matrix v_{ki} = probability of most probable path up until state i, if $\pi_i = k$. Then $v_{l,i+1} = e_{lx_{i+1}} \max_k (v_{ki} a_{kl})$.

- Initialize $v_{00} = 1$, $v_{k0} = 0 \forall k > 0$.
- Fill in the matrix until the bottom right, *each time pointing back to the previous row entry that gave the best answer*
- Trace back arrows from largest entry on bottom row

Viterbi algorithm: example



Define a matrix v_{ki} = probability of most probable path up until state i, if $\pi_i = k$. Then $v_{l,i+1} = e_{l x_{i+1}} \max_k (v_{ki} a_{kl})$.

- Initialize $v_{00} = 1$, $v_{k0} = 0 \forall k > 0$.
- Fill in the matrix until the bottom right, *each time pointing back to the previous row entry that gave the best answer*
- Trace back arrows from largest entry on bottom row

Likelihood of sequence: forward algorithm

The Viterbi algorithm gives you the most probable value of $P(x, \pi)$. But what if you only care about

$$P(x) = \sum_{\pi} P(x, \pi)$$

and not about the hidden path?
"Forward algorithm" lets you do that: define

$$f_{ki} = P(x_1 \dots x_i | \pi_i = k).$$

Then

$$f_{l,i+1} = e_{l,x_{i+1}} \sum_{k} f_{ki} a_{kl}.$$

In terms of this,

$$P(x) = \sum_{k} f_{kN} a_{k0}.$$

Backward algorithm, posterior decoding

Define

$$b_{ki} = P(x_{i+1} \dots x_N | \pi_i = k).$$

Then

$$P(x) = \sum_i a_{0k} e_{kx_1} b_{k1}.$$

But we can do better: we can infer the probability of any hidden state π_i.

$$P(\pi_i = k|x) = \frac{f_{ki} b_{ki}}{P(x)}.$$

HMMs applied to various tasks in linguistics

Pattern Recognition
Volume 27, Issue 10, October 1994, Pages 1345-1363

Connected and degraded text recognition using hidden Markov model ☆

Chinmoy B Bose ‡, Shyh-Shiaw Kuo §

Signal Processing Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974, U.S.A.

Pattern Recognition Letters
Volume 32, Issue 8, 1 June 2011, Pages 1081-1088

Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking

Jawad H AlKhateeb [a], Jinchang Ren [d], Jianmin Jiang [b], Husni Al-Muhtaseb [c]

Published as a conference paper at ICLR 2023

HIDDEN MARKOV TRANSFORMER FOR SIMULTANEOUS MACHINE TRANSLATION

Shaolei Zhang [1,2], **Yang Feng** [1,2]*
[1]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2]University of Chinese Academy of Sciences, Beijing, China
{zhangshaolei20z,fengyang}@ict.ac.cn

Profile HMMs

```
Helix                  AAAAAAAAAAAAAAAA   BBBBBBBBBBBBBBBBCCCCCCCCCCCC
HBA_HUMAN  ---------VLSPADKTNVKAAWGKVGA--HAGEYGAEALERMFLSFPTTKTYFPHF
HBB_HUMAN  --------VHLTPEEKSAVTALWGKV----NVDEVGGEALGRLLVVYPWTQRFFESF
MYG_PHYCA  ---------VLSEGEWQLVLHVWAKVEA--DVAGHGQDILIRLFKSHPETLEKFDRF
GLB3_CHITP ----------LSADQISTVQASFDKVKG------DPVGILYAVFKADPSIMAKFTQF
GLB5_PETMA PIVDTGSVAPLSAAEKTKIRSAWAPVYS--TYETSGVDILVKFFTSTPAAQEFFPKF
LGB2_LUPLU --------GALTESQAALVKSSWEEFNA--NIPKHTHRFFILVLEIAPAAKDLFS-F
GLB1_GLYDI ---------GLSAAQRQVIAATWKDIAGADNGAGVGKDCLIKFLSAHPQMAAVFG-F
Consensus             Ls....  v a W kv .  .    g . L.. f . P .   F  F
```

Sample alignment of globin proteins

(from Durbin et al,"Biological sequence analysis" 1998)

```
HBA_HUMAN   ...VGA--HAGEY...
HBB_HUMAN   ...V----NVDEV...
MYG_PHYCA   ...VEA--DVAGH...
GLB3_CHITP  ...VKG------D...
GLB5_PETMA  ...VYS--TYETS...
LGB2_LUPLU  ...FNA--NIPKH...
GLB1_GLYDI  ...IAGADNGAGV...
                ***  *****
```

Close-up of sample alignment of globin proteins

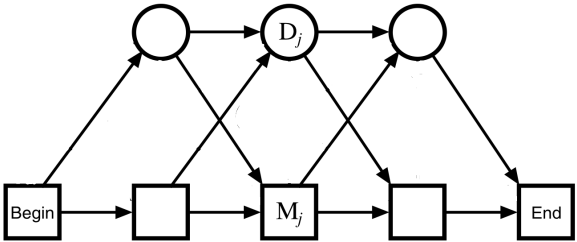
(from Durbin et al,"Biological sequence analysis" 1998)

```
HBA_HUMAN    ...VGA--HAGEY...
HBB_HUMAN    ...V----NVDEV...
MYG_PHYCA    ...VEA--DVAGH...
GLB3_CHITP   ...VKG------D...
GLB5_PETMA   ...VYS--TYETS...
LGB2_LUPLU   ...FNA--NIPKH...
GLB1_GLYDI   ...IAGADNGAGV...
                  ***  *****
```

Modelling match states

(from Durbin et al,"Biological sequence analysis" 1998)

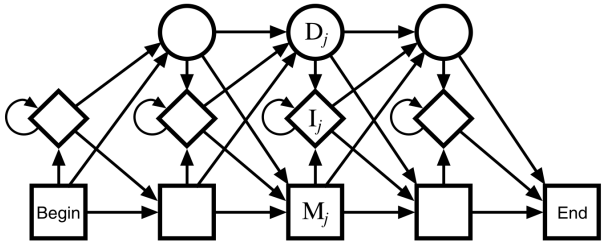
```
HBA_HUMAN    ...VGA--HAGEY...
HBB_HUMAN    ...V----NVDEV...
MYG_PHYCA    ...VEA--DVAGH...
GLB3_CHITP   ...VKG------D...
GLB5_PETMA   ...VYS--TYETS...
LGB2_LUPLU   ...FNA--NIPKH...
GLB1_GLYDI   ...IAGADNGAGV...
                ***  *****
```



Modelling match states and delete states

(from Durbin et al,"Biological sequence analysis" 1998)

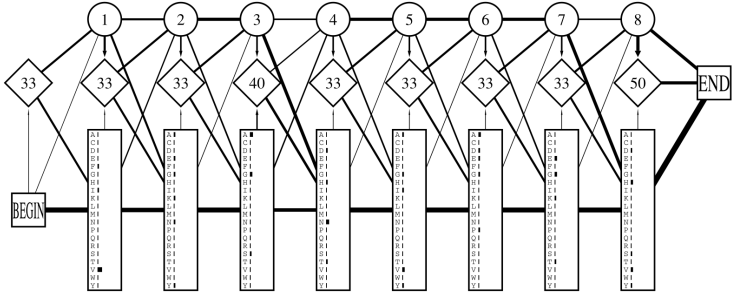
```
HBA_HUMAN    ...VGA--HAGEY...
HBB_HUMAN    ...V----NVDEV...
MYG_PHYCA    ...VEA--DVAGH...
GLB3_CHITP   ...VKG------D...
GLB5_PETMA   ...VYS--TYETS...
LGB2_LUPLU   ...FNA--NIPKH...
GLB1_GLYDI   ...IAGADNGAGV...
                ***  *****
```



Modelling match states, delete states and insert states

(from Durbin et al,"Biological sequence analysis" 1998)

```
HBA_HUMAN   ...VGA--HAGEY...
HBB_HUMAN   ...V----NVDEV...
MYG_PHYCA   ...VEA--DVAGH...
GLB3_CHITP  ...VKG------D...
GLB5_PETMA  ...VYS--TYETS...
LGB2_LUPLU  ...FNA--NIPKH...
GLB1_GLYDI  ...IAGADNGAGV...
               ***  *****
```



Trained HMM on globin alignment

(from Durbin et al,"Biological sequence analysis" 1998)

Hidden Markov models have been applied to a wide range of problems in NLP, speech recognition, bioinformatics, and more. Even in the deep-learning age, they remain a useful tool.

Thank you