Hidden Markov Models

Mathisca de Gunst
Department of Mathematics
Vrije Universiteit

Overview

- Two examples: gene and *CpG* island finding
- Types of models/Markov chains
- Hidden Markov models
- Viterbi algorithm
- Example: sequence alignment
- Conclusion

Modelling DNA (1)

- Structure of DNA: long sequence of nucleotides organized into coding sequences, or genes, separated by long intergenic regions of noncoding sequence.
- In most eukaryotic genes: within genes coding and noncoding sequences, exons and introns.
- Intergenic regions and introns have different statistical properties from those of exons.
- With two different models for each situation, it can be tested whether or not an uncharacterized piece of DNA is part of the coding region of a gene. Models are based on set of training data, taken from characterized sequences.

Types of models (1)

Need stochastic models, involving probabilities.

Consider DNA sequence of length n (single strand).

Define random variables
X_t = nucleotide at location t, t=1, … ,n.

X_t can take values in state space $\Sigma=\{A,T,C,G\}=\{b_1,b_2,b_3,b_4\}$.

$X_1, X_2, \ldots, X_n$ form a stochastic process, indexed by location.

Need to specify probability $P(X_t=b_i)$ for all i and t,
and dependence structure between the different X_t.

Types of models (2)

Stochastic process $X_1, \ldots, X_n$ with state space $\Sigma=\{b_1,\ldots,b_N\}$.

1. Most simple: $X_1, \ldots, X_n$ independent and identically distributed.

For DNA model we may, for instance, assume all nucleotides equally likely:

$P(X_t = b_i) = 1/4$ for all t and i.

This model NOT adequate for modelling DNA sequences.

Types of models (3)

Stochastic process $X_1, \ldots, X_n$ with state space $\Sigma=\{b_1,\ldots,b_N\}$.

2. Simple kind of dependence: state at time t depends only on state before it.
Markov property: for all t and all $x_1,\ldots,x_t$ in state space Σ

$$P(X_t = x_t \mid X_{t-1}=x_{t-1}, X_{t-1}=x_{t-1}, \ldots, X_1=x_1) = P(X_t = x_t \mid X_{t-1}=x_{t-1}) .$$

Process is Markov process, if state space discrete Markov chain.
Markov chain specified by

$A=(a_{ij})$, transition matrix with
$a_{ij} = P(X_t = b_j \mid X_{t-1}=b_i)$, transition probabilities (here independent of t),
$\phi=(\phi_1,\ldots,\phi_N)^T$, initial distribution, i.e. $\phi_i = P(X_1=b_i)$.

For DNA sequence: Markov chain with unequal transition probabilities for different pairs of nucleotides reasonable model (draw graph).

Modelling DNA (2)

- With two different Markov chains modelling coding vs. noncoding regions it can be statistically tested whether short piece of DNA is coding or not (likelihood ratio test).
- Another question: how to find coding regions in long stretch of DNA??
- Possibilities:
  - as for short stretch while using sliding windows: problematic
  - with more complex model.
- Specific question while using complex model: we observe sequence of nucleotides ACCGTAATT (input), but want to know for each nucleotide whether this originates from coding or non-coding region, such as $A^-C^-C^-G^+T^+A^+A^+T^-T^-$ (output), where $^+$ stands for coding and $^-$ stands for noncoding. This information is `hidden'.

Another example: *CpG*-Islands

- In human genome *CpG* (*CG*) is least frequent dinucleotide, because *C* in *CpG* is easily methylated and has the tendency to mutate into T afterwards.
- Methylation is suppressed around genes in a genome: *CpG* appears more frequently within these regions, called *CpG* islands.
- Identifying the *CpG* islands in a genome is important.
- Questions (similar to DNA example):
  - Given a short stretch of genomic sequence, how to decide if it comes from a *CpG* island or not?
  - Given a long sequence, how to find the *CpG* islands in it?

Hidden Markov model (1)

For DNA/*CpG* problem while using complex model:
we observe sequence of nucleotides ACCGTAATT (input), but
want to know hidden sequence of states $A^-C^-C^-G^+T^+A^+A^+T^-T^-$ (output), i.e.
for each nucleotide whether this originates from coding region/*CpG* island or
non-coding region/non-*CpG* island.

General:
$x_1,\ldots,x_n$ observed sequence of observations, realizations from stochastic process $X_1,\ldots,X_n$.
$\pi_1,\ldots,\pi_n$ hidden sequence, realizations of stochastic process $\Pi_1,\ldots,\Pi_n$.
.

Hidden Markov model (2)

General: $X_1,\ldots,X_n$ observed sequence, $\Pi_1,\ldots,\Pi_n$ hidden sequence.

The processes $\Pi_1,\ldots,\Pi_n$ and $X_1,\ldots,X_n$ together form a hidden Markov model (HMM) if

1. $\Pi_1,\ldots,\Pi_n$ is a Markov chain with state space $Q=\{q_1,\ldots,q_N\}$, transition matrix $A=(a_{ij})$, and initial distribution $\phi=(\phi_1,\ldots,\phi_N)^T$.
2. $X_1,\ldots,X_n$ is an observable process with outcome (symbol,emission) space $\Sigma=\{b_1,\ldots,b_M\}$.
3. $\Pi_1,\ldots,\Pi_n$ and $X_1,\ldots,X_n$ are related via conditional probabilities $e_i(.)$ (emission probabilities) for $i=1,\ldots,N$:
   $P(X_t=b_j|\ \Pi_t=q_i)=e_i(b_j)$, for $j=1,\ldots,M$.
4. Given de hidden states, the observations are independent.

Sets Q and Σ are assumed to be known.
Parameter set of HMM is $\lambda=\{A=(a_{ij}),E=(e_i(b_j)),\phi=(\phi_1,\ldots,\phi_N)^T\}$, can be unknown.

Hidden Markov model (3)

- Can be viewed as abstract machine with **N** *hidden* states that emits symbols from an alphabet **Σ** of size **M**.
- Each state has its own probability distribution, and machine switches between states according to this probability distribution.
- While in a certain state, machine makes 2 decisions:
  - **What state should I move to next?**
  - **What symbol - from the alphabet Σ - should I emit?**
- Observers can see the emitted symbols of an HMM but have *no ability to know which state the Markov chain of the HMM is currently in.*
- HMMs first used in speech recognition in 1970's (Rabiner et al.); nowadays in many other areas, e.g. weather prediction, finance, statistical genetics and bioinformatics.

Hidden Markov model (4)

There are three main problems that arise in the context of HMM modelling:

i) Given the parameters and the observed series (input), what is the most likely underlying unobserved series of hidden states (output) ?
ii) Given the parameters and the observed series (input), what is the probability of the series of observations (output) ?
iii) Given the observations (input), which parameters maximize the probability of the observations (output) ?

For each problem intuitive approach computationally not feasible, but efficient algorithms are available.

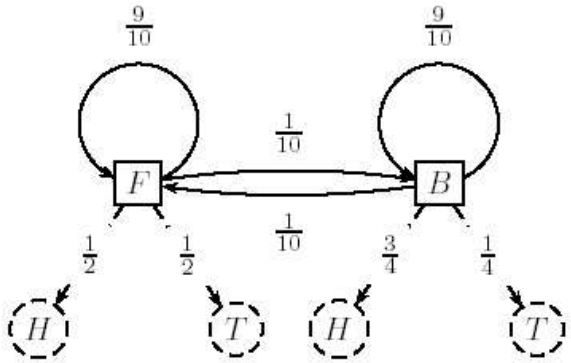
Observations are also called training data.

The second question in DNA/CpG island problems belongs to i).

The Fair Bet Casino

- DNA and *CpG* island problems can be modelled similar to a problem named "Fair Bet Casino".
- Dealer flips one of two coins, a fair one and a biased one, each of which has only two possible outcomes: head or tail.
- Fair coin will give heads and tails with same probability 1/2; biased coin will give heads with probability 3/4, tails with probability 1/4.
- Dealer changes between fair and biased coin with probability 0.1.

HMM graph for Fair Bet Casino



HMM for *Fair Bet Casino*

HMM for Fair Bet Casino

- The *Fair Bet Casino* in *HMM* terms:
  $\Sigma = \{0, 1\}$ (0 for ***T***ails and 1 ***H***eads)
  $Q = \{F,B\}$ – F for fair & B for biased coin.
- Transition probabilities A, emission probabilities E :

	Fair	Biased
Fair	$a_{FF} = 0.9$	$a_{FB} = 0.1$
Biased	$a_{BF} = 0.1$	$a_{BB} = 0.9$

	Tails(0)	Heads(1)
Fair	$e_F(0) = ½$	$e_F(1) = ½$
Biased	$e_B(0) = ¼$	$e_B(1) = ¾$

HMM parameters for Fair Bet Casino

$A = (a_{ij})$: a N x N transition probability matrix of probabilities of changing from state i to state j,

$a_{FF} = 0.9$ $a_{FB} = 0.1$

$a_{BF} = 0.1$ $a_{BB} = 0.9$.

$E = (e_i(b_j))$: a N x M matrix of probabilities of emitting symbols b_j while being in state i,

$e_F(0) = ½$ $e_F(1) = ½$

$e_B(0) = ¼$ $e_B(1) = ¾$.

Some initial distribution ϕ.

The Fair Bet Casino problem

Problem i) Given the parameters and the observed series (input), what is the most likely underlying unobserved series of hidden states (output) ?

- **Input:** A sequence $x = x_1, \ldots, x_n$ of coin tosses made by two possible coins (*F* or *B*).
- **Output:** A sequence $\pi = \pi_1, \ldots, \pi_n$ with each π_i being either *F* or *B* indicating that x_i is the result of tossing the fair or biased coin, respectively.

Problem…

Fair Bet Casino Problem

Any observed outcome of coin tosses could have been generated by any sequence of states!

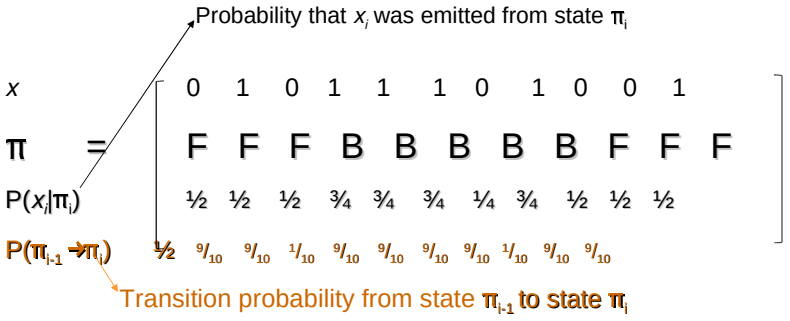
Need to incorporate a way to grade different sequences differently.

⇩

Decoding Problem

Hidden paths

A path $\pi = \pi_1, \ldots, \pi_n$ in the HMM is defined as a sequence of states.
Consider path π = FFFBBBBBFFF and
emission sequence x =01011101001 .



P(*x*|π) Calculation

$P(x|\pi)$: probability that sequence x was generated by the path π:

$$P(x|\pi) = \phi(\pi_1) \cdot \Pi_{i=1}^{n} P(x_i|\pi_i) \cdot P(\pi_i \rightarrow \pi_{i+1})$$

$$= \phi(\pi_1) \Pi e_{\pi_i}(x_i) \cdot a_{\pi_i, \pi_{i+1}} .$$

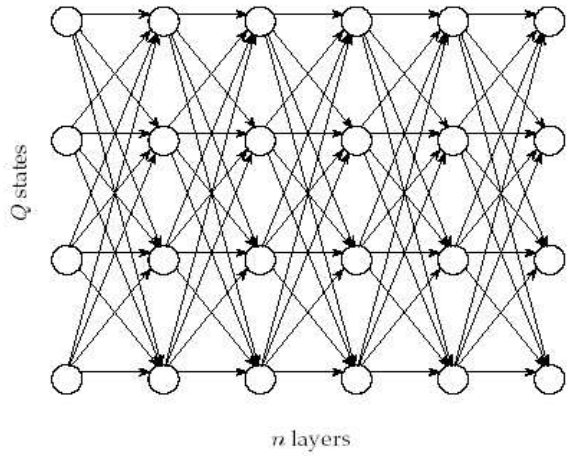
Decoding problem becomes:

find path that maximizes $P(x|\pi)$ over all possible paths π.

Manhattan for Decoding problem

- Andrew Viterbi used Manhattan grid model to solve this Decoding problem.
- Every choice of $\pi = \pi_1, \ldots \pi_n$ corresponds to a path in the graph.
- Only valid direction in the graph is *eastward.*
- This graph has $N^2(n-1)$ edges.

Edit Graph for Decoding Problem



Decoding Problem as finding a longest path in a DAG

- The *Decoding Problem* is reduced to finding a longest path in the *directed acyclic graph (DAG)* above.
- **Note:** length of the path is *product* of its edges' weights, not *sum.*
- Every path in the graph has the probability $P(x|\pi)$.
- The Viterbi algorithm finds the path that maximizes $P(x|\pi)$ among all possible paths.
- The Viterbi algorithm runs in $O(nN^2)$ time.

Another example: sequence alignment

- Finding distant members of a protein family:
- Distant cousin of functionally related sequences in protein family may have weak pairwise similarities with each member of family and thus fail significance test.
- However, cousin may have weak similarities with ***many*** members of family.
- Goal: align a sequence to ***all*** members of family at once.
- Use: family of related proteins can be represented by their multiple alignment and the corresponding profile plus HMM.

Profile representation of protein families

Aligned DNA sequences of length n can be represented by 4 ·n profile matrix reflecting frequencies of nucleotides in every aligned position:

A	.72	.14	0	0	.72	.72	0	0
T	.14	.72	0	0	0	.14	.14	.86
G	.14	.14	.86	.44	0	.14	0	0
C	0	0	.14	.56	.28	0	.86	.14

Protein family can be represented by 20·n profile representing frequencies of amino acids.

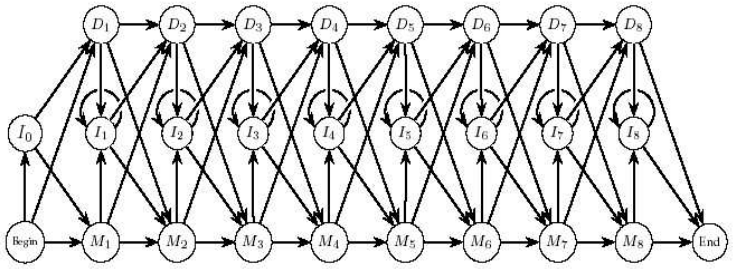
Profiles and HMMs

- HMMs can be used for aligning a sequence against a profile representing profile family:
- Profile HMM is a probabilistic representation of a multiple alignment.
- A given multiple alignment of protein family is used to build profile HMM.
- HMM model may be used to find and score less obvious potential matches of new protein sequences (problem **ii**) of HMM).

Building a profile HMM

- How is multiple alignment used to construct HMM model?
- A 20·n profile P corresponds to n sequentially linked match states $M_1,\ldots, M_n$ in profile HMM of P, i.e. each column corresponds to a match state in HMM.
- Add to each match state insertion and deletion state.
- Estimate emission probabilities according to amino acid counts in column. Different positions in the protein will have different emission probabilities.
- Estimate transition probabilities between match, deletion and insertion states also from profile.
- HMM model gets `trained' to derive the optimal parameters.

Profile HMM



A profile HMM

Conclusion

- This was introduction to HMMs.
- Couple of examples shown.
- Some insight into computing for decoding problem **i).**
- More details about Viterbi algorithm for problem **i)** and algorithms for problems **ii)** and **iii)** next week.
- Acknowledgement:
some slides adjusted from http//bioalgorithms.info/slides.htm.

Literature

- Durbin, R., Eddy, S., Krogh, A. and Mitchison, G. (1998). *Biological sequence analysis; probabilistic models of proteins and nucleic acids.* Cambridge University Press.
- Ewens, W.J. and Grant, G.R. (2001). *Statistical methods in bioinformatics. Springer.*
- Jones, N.C. and Pevzner, P.A. (2004). *An introduction to bioinformatics algorithms.* The MIT Press.
- Rabiner, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* **77**, 257–286.
- Rabiner, L.R. and Juang, B.H. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine* **3**, 4–16.