Genomanalyse


Mario Stanke
Institut für Mathematik und Informatik
Universität Greifswald
References

- my script at http://gobics.de/mario/genomanalyse/script.pdf
- "Biological Sequence Analysis", Durbin, Eddy, Krogh, Mitchison, Cambridge University Press
- "Bioinformatik Interaktiv", Rainer Merkl und Stephan Waack, Wiley, 2009
Markov Chain

**Definition (Markov Chain (Markow-Kette))**

A sequence of random variables \( X_1, X_2, \ldots \) with values in a discrete (here: finite) set \( Q \) is called a **Markov chain** if for all \( i > 0 \) and all \( x_1, x_2, \ldots, x_{i+1} \in Q \)

\[
P(X_{i+1} = x_{i+1} \mid X_1 = x_1, \ldots, X_i = x_i) = P(X_{i+1} = x_{i+1} \mid X_i = x_i),
\]

where defined.

The sequence is called a **homogenous** Markov chain if \( P(X_{i+1} = s \mid X_i = r) \) does not depend on \( i \) \((r, s \in Q)\), otherwise it is called **inhomogeneous**.

For a homogenous Markow chain, the matrix \( A = (a_{r,s})_{r,s \in Q} \) with \( a_{r,s} = P(X_{i+1} = s \mid X_i = r) \) is called the **transition matrix**.

The set \( Q \) is called **state space**. If \( X_i = q \), then we say that the process is in **state** \( q \) at **time** \( i \).

If \( X_i = r, X_{i+1} = s \) then the process is said to make a **transition** from \( r \) to \( s \).
Example: Markov Chain

Example (drunken guy)

After participating in a “pub crawl” a drunken guy is walking around Berlin. At each intersection he is stopping, turning around and randomly picking a direction to go next, independently of what happened in the past (no memory where he has been before and where he came from).

Example (random walk of drunken guy)

\[ X_i := \text{i-th visited intersection} \]
\[ X_1, X_2, \ldots \text{ is a homogenous Markov chain with state space} \]
\[ Q = \text{set of all intersections of Berlin} \]
Markov Chain

Intuitive interpretation

\[
P(X_{i+1} = x_{i+1} \mid X_1 = x_1, \ldots, X_i = x_i) = P(X_{i+1} = x_{i+1} \mid X_i = x_i).
\]

informally means

The future \((X_{i+1})\)

is independent of the past \((X_1, \ldots, X_{i-1})\)
given the present \((X_i)\).

Example (sober guy)

A sober guy is meandering in a foreign city. He remembers some intersections where he has been and previous choices for directions. E.g. he avoids a direction that he has previously taken.

The sequence of visited intersections is not a Markov chain.
Example: Markov Chain

Example

transition graph

\[ \begin{array}{c}
1 & \xrightarrow{.1} & 2 \\
.3 & \xrightarrow{.6} & 2 & \xrightarrow{.5} & 3 \\
.5 & \xrightarrow{.2} & 3 & \xrightarrow{1} & 4 \\
.8 & \xrightarrow{.8} & 1
\end{array} \]

transition matrix

\[ A = (a_{r,s})_{1 \leq r, s \leq 4} = \begin{pmatrix}
.3 & .6 & .1 & 0 \\
0 & .5 & .5 & 0 \\
.8 & 0 & 0 & .2 \\
0 & 0 & 1 & 0
\end{pmatrix} \]

\[ Q = \{1, 2, 3, 4\} \]

\[ P(X_1 = q) = \frac{1}{4} \quad (q \in Q) \]

\[ P(X_{i+1} = x_{i+1} \mid X_i = x_i, \ldots, X_1 = x_1) = a_{x_i, x_{i+1}} \quad (i > 1, x_1, \ldots, x_{i+1} \in Q) \]
Observation

Let $n > 1$ and $x_1, \ldots, x_n \in Q$. Then

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = P(X_1 = x_1) \cdot a_{x_1,x_2} \cdot a_{x_2,x_3} \cdots a_{x_{n-1},x_n}.$$
Simpler Notation

Instead of

\[
P(X_{i+1} = x_{i+1} \mid X_1 = x_1, \ldots, X_i = x_i) \\
= P(X_{i+1} = x_{i+1} \mid X_i = x_i),
\]

for all \( x_1, x_2, \ldots, x_{i+1} \in Q \)

we will write

\[
P(X_{i+1} \mid X_1, \ldots, X_i) = P(X_{i+1} \mid X_i).
\]
Hidden Markov Model

Definition (HMM)

Let $\Sigma$ be a finite set. A hidden Markov model (HMM) consists of a homogenous Markov chain

$$X_1, X_2, \ldots$$

over some state space $Q$ and a sequence of random variables

$$Y_1, Y_2, \ldots$$

with elements in $\Sigma$ such that

$$P(X_i \mid X_1, \ldots, X_{i-1}, Y_1, \ldots, Y_{i-1}) = P(X_i \mid X_{i-1})$$

and

$$P(Y_i \mid X_1, \ldots, X_i, Y_1, \ldots, Y_{i-1}) = P(Y_i \mid X_i).$$

$Y_i$ is called the $i$-th emission.

$\Sigma$ is called the emission alphabet.

$Y_1, Y_2, \ldots$ is called the emission sequence.

$X_i$ is called the $i$-th hidden state.
### Hidden Markov Model

**HMM**

- in typical applications
  - the emissions $Y_i$ are *observed*
  - the states $X_i$ are *not observed* (“hidden”)
- $P(Y_i \mid X_i)$ is called the emission distribution.
- the states are sometimes called *labels*, $Q$ the *label set*

**Notation**

We will denote the transition probabilities with $a_{r,s}$:

$$a_{r,s} = P(X_i = s \mid X_{i-1} = r)$$

and the emission probabilities with $b(q, s)$:

$$b(q, s) = P(Y_i = s \mid X_i = q) \quad (q \in Q, s \in \Sigma).$$
Example: Hidden Markov Model

Example (The occasionally dishonest casino)

- In a casino two dice are used
  - a fair (F) die: $P(1) = P(2) = \cdots = P(6) = 1/6$
  - a loaded (L) die: $P(6) = 1/2$, $P(1) = \cdots = P(5) = 1/10$
- the dice are otherwise indistinguishable for the players
- a die is repeatedly thrown starting with the fair die
- each time after a die is thrown, there is a chance that the casino exchanges the dice (F->L or L->F)
- when a fair.loaded die was thrown the chance of switching to the loaded/fair is 5% and 15%, respectively

Let $Y_1, Y_2, \ldots \in \Sigma := \{1, 2, \ldots, 6\}$ be the sequence of observed numbers.
Let $X_1, X_2, \ldots \in Q := \{F, L\}$ be the corresponding labels of the dice

Example (Example Output)

$Y_1, Y_2, \ldots = 526414236166616661666252536234666636253615112566335226253322366641443$
$X_1, X_2, \ldots = FFFFFFFFFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLFFFFFFFFFFF
The occasionally dishonest casino

**Dishonest Casino**

The emission sequence \( Y_1, Y_2, \ldots \) and hidden state sequence \( X_1, X_2, \ldots \) constitute a HMM.

The sequence \( X_1, X_2, \ldots \) is a Markov chain with \( X_1 \equiv F \) and transition distribution

\[
P(X_i = s \mid X_{i-1} = r) = a_{r,s}, \quad \text{where } (a_{r,s})_{r,s \in Q} = \begin{pmatrix} .95 & .05 \\ .15 & .85 \end{pmatrix}
\]
The occasionally dishonest casino

Emission distribution

The emission distribution is

\[
 b(q, s) = P(Y_i = s \mid X_i = q) = \begin{cases} 
 1/6 & \text{, if } q = F \\
 1/10 & \text{, if } q = L \text{ and } s \neq 6 \\
 1/2 & \text{, if } q = L \text{ and } s = 6 
\end{cases} 
\]

\((s \in \{1, 2, \ldots, 6\}, q \in \{F, L\})\).
Applications of HMMs

Applications in

- speech recognition
- part-of-speech tagging
- Bioinformatics (gene prediction, models of a protein family, alignments, ...)

“Decoding” an HMM

Decoding: Finding the hidden states

In a typical application of HMMs, the emission sequence is known ("observed") and finite

\[ Y_1 = y_1, \ Y_2 = y_2, \ldots, \ Y_n = y_n \]

and the state sequence is unknown ("hidden") and needs to be predicted.

Let

\[ Y := (Y_1, \ldots, Y_n) \]
\[ y := (y_1, \ldots, y_n) \]
\[ X := (X_1, \ldots, X_n). \]

One popular way of decoding is to use

\[ \hat{x} \in \arg\max_{x \in Q^n} P(X = x \mid Y = y) \] (1)

as prediction of the unknown sequence of hidden states.
Viterbi-Decoding

Meaning of (1)

- $P(X = x \mid Y = y)$ is called the posterior probability of $x$, also: a posteriori
- $P(X = x \mid Y = y)$ is the probability of state sequence $x$ given that the emission $y$ is observed
- Frequently, a unique $x$ maximizes $P(X = x \mid Y = y)$. If the maximizing $x$ is not unique then

$$\arg\max_{x \in \Sigma^n} P(X = x \mid Y = y)$$

denotes the set of all state sequences having maximal posterior probability.
- $\hat{x}$ is a (the) most likely state sequence given the observed emission sequence
- (1) is called the MAP-estimator (maximum a posteriori)
Joint probability of states and emission

Because

\[ P(X = x \mid Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}, \] (2)

maximizing \( P(X = x \mid Y = y) \) over \( x \) is equivalent to maximizing \( P(X = x, Y = y) \) over \( x \).

Idea: Dynamic Programming

We find the most likely state sequence

\[ \operatorname{argmax}_{x \in Q^n} P(X = x, Y = y) \]

using dynamic programming.
Joint Probability of $x$ and $y$

**Product formula for joint probability of $x$ and $y**

\[
P(X = x, Y = y) = P(X_1 = x_1) \cdot P(Y_1 = y_1 | X_1 = x_1) \\
\cdot \prod_{i=2}^{n} \left\{ P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}, Y_1 = y_1, \ldots, Y_{i-1} = y_{i-1}) \\
\cdot P(Y_i = y_i | X_1 = x_1, \ldots, X_i = x_i, Y_1 = y_1, \ldots, Y_{i-1} = y_{i-1}) \right\} \\
= P(X_1 = x_1) \cdot P(Y_1 = y_1 | X_1 = x_1) \cdot \prod_{i=2}^{n} P(X_i = x_i | X_{i-1} = x_{i-1}) \cdot P(Y_i = y_i | X_i = x_i) \tag{3}
\]

Here, the first line is an application of the general formula $P(A_1 \cap A_2 \cap A_3 \cap \cdots) = P(A_1)P(A_2 | A_1)P(A_3 | A_1 \cap A_2) \cdots$ applied to the sequence of events $A_1 = \{ X_1 = x_1 \}, A_2 = \{ Y_1 = y_1 \}, A_3 = \{ X_2 = x_2 \}, \cdots$.

(3) follows from the two independence requirements in the definition of HMMs.

Substituting the variables for transition and emission probabilities, we get

\[
P(X = x, Y = y) = P(X_1 = x_1) \cdot b(x_1, y_1) \cdot \prod_{i=2}^{n} a_{x_{i-1}, x_i} b(x_i, y_i).
\]
Viterbi Algorithm

**Definition (Viterbi variables)**

Let

\[ \gamma(q, i) := \max_{x_1, \ldots, x_{i-1} \in Q} P((X_1, \ldots, X_i) = (x_1, \ldots, x_{i-1}, q), (Y_1, \ldots, Y_i) = (y_1, \ldots, y_i)) \]

\[ = \text{joint probability for most likely state sequence up to } i \]
\[ \text{that ends in state } q \]

for states \( q \in Q \) and time points \( 1 \leq i \leq n \).

**Viterbi recursion**

\[ \gamma(q, i) = b(q, y_i) \cdot \max_{q' \in Q} a_{q', q} \cdot \gamma(q', i - 1) \quad (q \in Q, 1 < i \leq n) \]

Initial cases:

\[ \gamma(q, 1) = P(X_1 = q) \cdot b(q, y_1) \]
Proof of Viterbi recursion...

Initial case: Let $q \in Q$. Then by definition of $\gamma$ and of the emission distribution we have

$$
\gamma(q, 1) = P(X_1 = q, Y_1 = y_1) = P(X_1 = q) \cdot P(Y_1 = y_1 | X_1 = q) = P(X_1 = q) \cdot b(q, y_1).
$$

Now, let $1 < i \leq n$.

$$
\gamma(q, i) = \max_{x_1, \ldots, x_{i-1} \in Q} P((X_1, \ldots, X_i) = (x_1, \ldots, x_{i-1}, q), (Y_1, \ldots, Y_i) = (y_1, \ldots, y_i))
= \max_{x_1, \ldots, x_{i-1} \in Q} \left\{ P((X_1, \ldots, x_{i-1}) = (x_1, \ldots, x_{i-1}), (Y_1, \ldots, Y_{i-1}) = (y_1, \ldots, y_{i-1})) \right. \\
\left. \cdot P(X_i = q | (X_1, \ldots, y_{i-1})) \cdot P(Y_i = y_i | X_i = q, (X_1, \ldots, y_{i-1})) \right\}
= \max_{x_1, \ldots, x_{i-1} \in Q} \left\{ P((X_1, \ldots, y_{i-1})) \cdot P(X_i = q | X_{i-1} = x_{i-1}) \right. \\
\left. \cdot P(Y_i = y_i | X_i = q) \right\} 
$$

(Equally colored expressions are equal.)

Here, the second line follows as $P(A, B, C) = P(A) \cdot P(B | A) \cdot P(C | B, A)$ holds for any events $A, B, C$.
The left term in (4) is equal to to the corresponding term in the previous line because $P(X_i | X_1, \ldots, X_{i-1}, Y_1, \ldots, Y_{i-1}) = P(X_i | X_{i-1})$ by definition of an HMM.
The right term in (4) is equal to to the corresponding term in the previous line because $P(Y_i | X_1, \ldots, X_i, Y_1, \ldots, Y_{i-1}) = P(Y_i | X_i)$ by definition of an HMM.
...Proof of Viterbi recursion.

We continue

\[
\gamma(q, i) = b(q, y_i) \cdot \max_{x_{i-1} \in Q} \{ \max_{x_1, \ldots, x_{i-2} \in Q} P(X_i = q \mid X_{i-1} = x_{i-1}) \cdot P((X_1, \ldots, y_{i-1} )) \} \tag{5}
\]

\[
= b(q, y_i) \cdot \max_{x_{i-1} \in Q} \{ a_{x_{i-1}, q} \cdot \max_{x_1, \ldots, x_{i-2} \in Q} P((X_1, \ldots, y_{i-1} )) \} \tag{6}
\]

\[
= b(q, y_i) \cdot \max_{x_{i-1} \in Q} \{ a_{x_{i-1}, q} \cdot \gamma(x_{i-1}, i - 1) \} \tag{7}
\]

Equation 5 holds because \( b(q, y_i) = P(Y_i = y_i \mid X_i = q) \) is independent of \( x_1, \ldots, x_{i-1} \).

Equation 6 holds because \( a_{x_{i-1}, q} = P(X_i = q \mid X_{i-1} = x_{i-1}) \) is independent of \( x_1, \ldots, x_{i-2} \).

Equation 7 holds by definition of \( \gamma \) applied to state \( x_{i-1} \) and time \( i - 1 \).

Replacing the maximizing variable \( x_{i-1} \) with \( q' \) yields the claim from the Viterbi recursion. \( \square \)
Viterbi Algorithm

1: for $q \in Q$ do
2:   $\gamma(q, 1) \leftarrow P(X_1 = q) \cdot b(q, y_1)$ // initialization of boundary cases
3: end for
4: for $i = 2$ to $n$ do
5:   for $q \in Q$ do
6:     update $\gamma(q, i)$ according to Viterbi recursion
7:   end for
8: end for
9: // backtracking starts
10: $x \leftarrow <$empty list$>$ // $x$ will hold most likely state sequence
11: $i \leftarrow n$
12: $q \leftarrow \text{argmax}_{q \in Q} \gamma(q, n)$
13: repeat
14:   add $q$ to front of $x$
15: $q \leftarrow \text{argmax}_{q' \in Q} a_{q', q} \cdot \gamma(q', i - 1)$
16: $i \leftarrow i - 1$
17: until $i = 0$
18: output $x$ as solution
Complexity of Viterbi Algorithm

Complexity

- **time**: $O(|Q|^2 \cdot n)$
  (line 6 takes time proportional to $|Q|$ and is executed $|Q| \cdot (n - 1)$ times)
- **space**: $O(|Q| \cdot n)$
  (to store the array $\gamma$)
Posterior Probabilities of States

Probabilities of States

- **prior (a priori)** probability of states $P(X_i = q)$ is given through Markov chain alone
- **posterior (a posteriori)** probability of states $P(X_i = q \mid Y = y)$ depends on the observed data
- e.g. $P(X_i = L \mid Y = y)$ is relatively high if many '6' are rolled around time $i$
Posterior Probabilities of States

We have

\[
P(X_i = q \mid Y = y) = \frac{P(X_i = q, Y = y)}{P(Y = y)} = \alpha(q, i) \cdot \beta(q, i)
\]

The last equation holds because \( Y_{i+1}, \ldots, Y_n \) depend on \( X_i, Y_1, \ldots, Y_i \) only through \( X_i \).

Define variables \( \alpha(q, i), \beta(q, i) \) for \((q \in Q, 1 \leq i < n)\) called forward and backward probabilities, respectively.

**Motivation:** Forward and backward probabilities can be calculated using Dynamic Programming.
Forward Algorithm

Definition (Forward Variables)

\[ \alpha(q, i) := P(X_i = q, (Y_1, \ldots, Y_i) = (y_1, \ldots, y_i)) \quad (q \in Q, i \geq 1) \]

Forward Recursion

We have for \( q \in Q \)

\[ \alpha(q, i) = b(q, y_i) \sum_{q' \in Q} a_{q', q} \cdot \alpha(q', i - 1) \quad (i > 1) \]

Initial case

\[ \alpha(q, 1) = P(X_1 = q)b(q, y_1) \]

Proof: Exercise
Forward Algorithm

1: \textbf{for } i = 1 \textbf{ to } n \textbf{ do}
2: \hspace{1em} \textbf{for } q \in Q \textbf{ do}
3: \hspace{2em} \text{compute } \alpha(q, i) \text{ according to forward recursion}
4: \hspace{1em} \textbf{end for}
5: \textbf{end for}

\textbf{time: } O(|Q|^2 \cdot n)
\textbf{space: } O(|Q| \cdot n)
Backward Algorithm

Definition (Backward Variables)

Let $n > 1$ be given. Define

$$\beta(q, i) := P((Y_{i+1}, \ldots, Y_n) = (y_{i+1}, \ldots, y_n) | X_i = q) \quad (q \in Q, i < n)$$

Backward Recursion

We have for all $q \in Q$

$$\beta(q, i) = \sum_{q' \in Q} a_{q, q'} b(q', y_{i+1}) \cdot \beta(q', i + 1) \quad (1 \leq i < n)$$

Initial case

$$\beta(q, n) := 1.$$
Backward Algorithm

1: for \( i = n \) to 1 do
2: \hspace{1em} for \( q \in Q \) do
3: \hspace{2em} compute \( \beta(q, i) \) according to backward recursion
4: \hspace{1em} end for
5: end for

time: \( O(|Q|^2 \cdot n) \)
space: \( O(|Q| \cdot n) \)
Probabilities of Emission

**Probability of Emission** \( y = (y_1, \ldots, y_n) \)

We can now compute \( P(Y = y) \) with the law of total probability using either the forward variables

\[
P(Y = y) = \sum_{q \in Q} P(X_n = q, Y = y) = \sum_{q \in Q} \alpha(q, n)
\]

or the backward variables

\[
P(Y = y) = \sum_{q \in Q} P(X_1 = q, Y = y)
\]

\[
= \sum_{q \in Q} P(X_1 = q) b(q, y_1) \cdot P((Y_2, \ldots, Y_n) = (y_2, \ldots, y_n) \mid X_1 = q)
\]

\[
= \sum_{q \in Q} P(X_1 = q) b(q, y_1) \cdot \beta(q, 1).
\]
Now we can compute the posterior probability of a state $q$ at time point $i$

$$P(X_i = q \mid Y = y) = \frac{\alpha(q, i) \cdot \beta(q, i)}{P(Y = y)}$$

Example (posterior probability of F)

52641423616661666252536234666636253615112566335226253322366641443

In each column $i$ the number of stars measures $P(X_i = F \mid Y = y)$. 1* = 10%