

Lecture Notes: Markov Models

Basic probability background

$P(a)$ - the probability of an event, a

Probability is always a number in the interval $[0,1]$

where 0 means "never true"

1 means "always true"

Example: $P(\text{rain}) = 0.4$

$P(A = a)$ means "the probability that variable A has value a "

Example: $P(W=\text{rain}) = 0.4$

If A has several possible values, a_1, a_2, \dots, a_n , then:

$$\sum_{i=1}^n A_i = 1$$

JOINT PROBABILITY: the probability that two events A and B both happen

Write this as: $P(A,B)$.

Similarly, $P(A = a, B = b)$ means "the probability that variable A has value a and variable B has value b ."

Example: $P(\text{rain,weekend}) = 0.2$

Definition: A and B are INDEPENDENT iff $P(A, B) = P(A)P(B)$

CONDITIONAL PROBABILITY: $P(A = a|B = b)$ is the probability that $A = a$, given that $B = b$.

(Notational shorthand: $P(A = a|B = b)$ is the same as $P(a|b)$).

DEFINITION:

$$P(a|b) = \frac{P(a, b)}{P(b)}$$

$$P(b|a) = \frac{P(b, a)}{P(a)}$$

Equivalently:

$$P(a, b) = P(a|b)P(b)$$

Bayes' Law

$$P(a|b) = \frac{P(a, b)}{P(b)} = \frac{P(b|a)P(a)}{P(b)} \quad (1)$$

Modeling sequence data

Example: Suppose we want to model a domain with just 4 events: ATG, GTG, TTG, CTG

$$\begin{aligned} & P(X=ATG) = 0.8 \\ \text{M1} \quad & P(X=GTG) = 0.1 \\ & P(X=TTG) = 0.07 \\ & P(X=CTG) = 0.03 \end{aligned}$$

This set of probability assignments is a MODEL, M1. This might represent start codons in a hypothetical bacterium, say *M. nasty1*.

Here is a second model, M2 (*M. nasty2*):

$$\begin{aligned} & P(X=ATG) = 0.6 \\ \text{M2} \quad & P(X=GTG) = 0.1 \\ & P(X=TTG) = 0.2 \\ & P(X=CTG) = 0.1 \end{aligned}$$

Now, suppose we observe the event:

$$X = ATG$$

If we assume we're in the world of model M1, then $P(X) = 0.8$.

More accurately, we write:

$$P(X|M1) = 0.8$$

$$P(X|M2) = 0.6$$

Usually, we don't know which model is correct — that's what we want to know. The question might be, e.g., “what organism did this DNA come from?”

We would like to decide whether X came from M1 or M2. Using Bayes' Law, we ask whether the probability of M1 is greater than M2:

In other words, we prefer M1 if:

$$P(M1|X) > P(M2|X)$$

$P(M|X)$ can be read as “the probability that X is drawn from the model M, given that the value of X is x.”

Using Bayes' Law:

$$P(M1|X) = \frac{P(X|M1)P(M1)}{P(X)}$$

$$P(M2|X) = \frac{P(X|M2)P(M2)}{P(X)}$$

So we are asking:

$$\frac{P(X|M1)P(M1)}{P(X)} > \frac{P(X|M2)P(M2)}{P(X)}$$

$$P(X|M1)P(M1) > P(X|M2)P(M2)$$

$$0.8P(M1) > 0.6P(M2)$$

Notice that it doesn't matter what $P(X)$ is. However, we do need to know the probability of each model. (N.B. These are called the *prior* probabilities.)

Suppose both models are equally likely; for example, X could be a sequence drawn from a mixture of equal parts M1 and M2. Then M1 is more likely. But if we have some reason to suspect M2 more strongly, then M2 could win.

Markov chains

The previous example treated a 3-base sequence as an event. Next let's treat the identity of each base in a sequence as a separate event.

Markov chains are a simple type of *Markov model*. In a Markov chain, we model a sequence of data by considering the probabilities of each item in the sequence.

In Markov chains, the probability of any character in a sequence depends only on what the preceding few characters were.

The **Markov assumption**: *the probability of an event X depends only on a fixed number of previous events.*

This assumption makes it possible to compute things that otherwise wouldn't be computable.

Here are 0^{th} and 1^{st} order models. In a 0^{th} order model, each base depends on zero previous bases. In a 1^{st} order model, each base depends on the base immediately previous.

0^{th} order model:

$$P(A) = 0.1 \quad P(C) = 0.3 \quad P(G) = 0.2 \quad P(T) = 0.4$$

1^{st} order model:

$$P(A|A) = 0.1 \quad P(C|A) = 0.3 \quad P(G|A) = 0.2 \quad P(T|A) = 0.4$$

$$P(A|C) = 0.2 \quad P(C|C) = 0.1 \quad P(G|C) = 0.4 \quad P(T|C) = 0.3$$

$$P(A|G) = 0.1 \quad P(C|G) = 0.2 \quad P(G|G) = 0.3 \quad P(T|G) = 0.4$$

$$P(A|T) = 0.3 \quad P(C|T) = 0.1 \quad P(G|T) = 0.4 \quad P(T|T) = 0.2$$

Note that for the 1^{st} order model, the sum of each row is 1.

SCORING A SEQUENCE with a Markov chain

Example sequence: GGTACC

0th order model score:

$$0.2 * 0.2 * 0.4 * 0.1 * 0.3 * 0.3 = 0.000144$$

1st order model score:

$$0.2 * 0.3 * 0.4 * 0.3 * 0.3 * 0.1 = 0.000216$$

The computation is very simple: just multiply probabilities.

This is statistically okay due to the Markov assumption. The definition of independence allows us to multiply probabilities.

Computational trick: we often use the fact that:

$$\log ab = \log a + \log b$$

to speed up computation.

Instead of storing a table of probabilities, store the logs. Then the computation (scoring) uses addition, which is faster than multiplication.

This also avoids the problem of storing very tiny numbers, which always arises when the sequences get to be long. A sequence of 5,000 bases will have a probability in the range of 2^{-10000} , too small to represent on a computer. But $\log 2^{-10000} = -10000$, which poses no problems computationally.

An n^{th} order Markov model of DNA requires a table of 4^{n+1} probabilities. An n^{th} order model for protein sequences requires 20^{n+1} probabilities.

Training a Markov chain

Training is simple: the probabilities can be computed from data.

For example, to get $P(C|T)$ (the probability of C given that the previous base was T):

1. Definition: $P(C|T) = \frac{P(C,T)}{P(T)}$.
2. Count how many T's appear in the data (which could be any sequence, from a small fragment up to a complete genome). Suppose we observe 100 T's.
3. Count how many times T is followed by C. Suppose this happens 10 times.
4. Let G = length of the whole sequence.
5. $P(C|T) = \frac{10/G}{100/G} = 0.1$.

Why is it called a chain? See Figure 1.

Microbial gene finding with Markov chains

5th order chains have proved the best. This is the main idea behind GeneMark (Borodovsky and McIninch, 1993).

Why 5th order?

- Looking at 5 previous bases to score a base means we are looking at two consecutive codons.

- Bacterial genomes are long enough to provide good estimates for $4^6 = 4096$ probabilities.
- 8^{th} order might work even better — this would use three consecutive codons instead of two. But $4^9 = 262,144$ probabilities, and a 2 megabase genome gives only 8 events (on average) to estimate each of these values.

Scoring AAGACGTGCAT with a 5^{th} order chain:

$$P(A) * P(A|A) * P(G|AA) * P(A|AAG) * P(C|AAGA) * P(G|AAGAC) * P(T|AGACG) \dots$$

Higher order Markov chains are better! (if enough data is available)

Example (proof):

Suppose a 1^{st} order model is the “true” model.

Then in the sequence GGCGA, $P(A|GGCG) = P(A|G)$.

Suppose we “mistakenly” use a 2^{nd} order chain. Thus we use $P(A|CG)$ to compute the probability of the final “A.”

Since the 1^{st} order model is correct, the position two bases prior to A is irrelevant. Therefore

$$P(A|CG) = P(A|AG) = P(A|GG) = P(A|TG) = P(A|G)$$

I.e., the probability of A is determined only by the fact that a G occurred in the previous position.

If we observe enough data to have an accurate estimate of $P(A|CG)$, then it will give us the same value as $P(A|G)$.

On the other hand, suppose the 2^{nd} order model is the “true” model:

If we use the 1^{st} order model, it will use incorrect estimates for some of the probabilities.

Nonhomogeneous Markov chains

See Figure 2.

Algorithm for gene finding

1. Train a model using “known” genes (very long ORFs, database hits).
2. Train a 0^{th} order Markov model as a “default” model.
3. For every open reading frame, score it in all 6 possible reading frames (3 forward, 3 backward). (Use a nonhomogeneous model.)
4. If the highest scoring frame is the “open” one, then call it a gene.
5. For overlapping ORFs, score the overlap region separately to break ties.

Interpolated Markov models

IMMs are the basis of Glimmer, the gene finder we’re now using at TIGR for bacterial DNA sequences (Salzberg, Delcher, Kasif, and White, 1998).

Instead of using a 5^{th} order model, use all these Markov models:

$$0^{th}, 1^{st}, 2^{nd}, 3^{rd}, 4^{th}, 5^{th}, 6^{th}, 7^{th}, 8^{th}$$

This is guaranteed to work better than any one of the models alone, *if* it is done properly.

Idea: use all the models, giving each one a different weight depending on how much data you've seen.

When insufficient data is available (a common situation for highest order models), don't use the model.

Scoring AAGACGTGCAT with a 5th order chain:

$$S(A) * S(A|A) * S(G|AA) * S(A|AAG) * S(C|AAGA) * S(G|AAGAC) * S(T|AGACG) \dots$$

where $S(x)$ is a combination of all models. Example:

$$\begin{aligned} S(T|AGACG) = & \lambda_{agacg}(P(T|AGACG) + \lambda_{gacg}P(T|GACG) + \\ & \lambda_{acg}P(T|ACG) + \lambda_{cg}P(T|CG) + \\ & \lambda_gP(T|G) + \lambda_0P(T) \end{aligned}$$