HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 1 — January 5

Lecturer: John Novembre *Scribe: (None)*

Note: These lecture notes are still rough, and have only have been mildly proofread.

1.1 HG48600/STAT34550 Lectures

1.1.1 Lecture 1: Introduction to Course and Probability

Introduction

- Themes of this course:
  - Thinking probabilistically
    - The systems we aim to model are inherently **stochastic**
    - Probabilities gives us a language for expressing our uncertainy in precise terms (i.e. we are often going to be thinking as Bayesians)
  - Handling complex probability distributions
    - Those with an index set (i.e. **stochastic processes**)
    - **Heirarchical models** with underlying **latent (hidden)** variables
  - Constructing custom solutions to inference problems in biology
    - Recognizing the biological aspects of a problem and being able to build it into our solutions, i.e. not being beholden to fitting a problem into frameworks already invented
    - That said, we will learn several general purpose models
- Broader context for this course
  - We see three domains are commonly mastered by the best computational biologists.
  - This course will cover 2 of them at an introductory level: Stochastic processes and inference in complex, heirarchical models.
  - The third domain will be the subject of a course that will be taught next year: Computational data structures and algorithms.

Course expectations

- Problem Sets
  - 5 total: You will have at least 1 week to complete them
- Final project
  - Do something interesting leveraging the concepts of this course
  - Use ideas from this course to address a small problem in an area of biology that interests you (need not be your PhD research area)
  - Develop a teaching vignette / lab for a subject area of this course
  - Poster Session on the last day of class
- Scribe duty:
  - You will take notes, most likely on pen and paper.
  - After class you will write them up via latex (or markdown) and post.
  - Please sign up with Evan.

Review: Marginal, Joint, and Conditional distributions, Bayes Rule

- Motivation
  - Most problems we work on involve multiple random variables.
  - To think about multiple random variables at a time it is useful to understand **joint**, **marginal** and **conditional** distributions. There are also analogous forms for expectations, variances, and covariances.
- Example: A basic two-variable discrete joint probability distribution
  - Example 1

X—Y	$Y = 1$	$Y = 2$	$P(X = x)$
$X = 0$	0.08	0.12	0.2
$X = 1$	0.16	0.24	0.4
$X = 2$	0.12	0.18	0.3
$X = 3$	0.04	0.06	0.1
$P(Y = y)$	0.4	0.6	

- Conditional probability and independence:

- The basic definition
$$P(B|A) = \frac{P(A,B)}{P(A)}$$
Note: Trivially generalizes for talking about discrete or continuous random variables.
Also note: we like to replace the formal notation $P(A = a)$ by $P(A)$.
- Independence
  - Two events A,B are said to be indepdent if $P(A,B) = P(A)P(B)$
  - Note from def of conditional probablity this implies: $P(B|A) = P(B)$ (and $P(A|B) = P(A)$)
  - A big theme of the course will be leveraging conditional probabilities and independence to solve problems.
- Marignal distributions and the law of total probability: We can "marginalize" by a summation operation:
$$P(A = a) = \sum_{b:P(B=b)>0} P(A = a, B = b)$$
or
$$P(A = a) = \sum_{b:P(B=b)>0} P(A = a|B = b)P(B = b)$$
or in shorthand
$$P(A) = \sum P(A|B)P(B)$$
Note: As is often the case, the analogous form for continous random variables replaces the summation step with integration.

- Bayes' rule
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$
This has tremendous utility as a tool for taking one conditional probablity ($P(B|A)$) and computing it's "inverse" $P(A|B)$. It also has great utility for inference problems and shows up in the following form. (Matthew will expand on this latter point)
$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)} = fracP(X|\theta)P(\theta)\int P(X|theta)P(\theta)d\theta$$
Where, X are some data, and θ are the parameters of our model.

Review: Introduction to Random Variables

- Basic definitions:
  - Ω : The sample space; points in Ω represent **elementary events**
  - Probability:
    - A function that ascribes a measure to each point (and subset of points) in the sample space, with the important property that the integral of the measure over Ω equals 1.
    - Interpretations: The frequency at which an event will occur, a measure of uncertainty
  - Random variables : Real-valued function over the elementary events in the sample space.
    - Example: X is the sum of two fair die.
      - $X = 2$ if the first die is 1 and the second is 1.
    - Example: An **indicator variable** for whether a single die is even.
      - $I_{odd} = 1$ if die role is single die role is 2, 4, 6; and 0 otherwise.
    - Probabilities can be assigned to the values of random variables
    - Typically we think at the level of random variables and probability distributions/densities (and ignore the more formal construction of the sample space and measure definitions)

- Basic Discrete Random Variables:

Name	parameters	probability mass function	Mean	Variance
Binomial	$n > 0$ and $0 \leq p \leq 1$	$\binom{n}{x} p^x (1-p)^x$	np	$np(1-p)$
Poisson	$\lambda > 0$	$e^{-\lambda} \frac{\lambda^x}{x!}$	λ	λ
Geometric	$0 \leq p \leq 1$	$p(1-p)^{x-1}$	$\frac{1}{p}$	$\frac{1-p}{p^2}$

See Ross Table 2.1

- Basic Continuous Random Variables:

Name	parameters	probability density function	Mean	Variance
Uniform	a,b	$\frac{1}{b-a}$ for $a < x < b$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
Exponential	$\lambda > 0$	$\lambda e^{-\lambda x}$ for $x > 0$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Gamma	n¿0,$\lambda > 0$	$\frac{\lambda e^{-\lambda x} (\lambda x)^{n-1}}{(n-1)!}$ for $x \geq 0$	$\frac{n}{\lambda}$	$\frac{n}{\lambda^2}$
Normal	μ, $\sigma^2 > 0$	$\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$	μ	σ^2
Beta	$\alpha > 0$,$\beta > 0$	$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$

  - Note: See Ross Table 2.2

- Additional random variable distribution that will be of interest for this course
  - Distributions of the exponential family, in particular:
    * Multinomial distribution
    * Dirichlet distribution (a multivariate analog of the beta)
    * Multivariate Normal distribution

- Definition of a stochastic process
  - We will spend a large amount of our time thinking about a special collection of random variables known as a **stochastic process**
  - A stochastic process is a set: $X(t), t \in T$
  - $X(t)$ as the **state** of the system at time t.
  - T as the **index set** of the process. t often interpreted as time variable or a spatial variable.
  - **State space** : The set of possible values of $X(t)$
  - Stochastic processes are a family of random variables that describe the evolution through time of some (physical) process.
  - We will use stochastic processes as models for biological processes, and as a trick to simulate from intractable distributions (this is the idea of MCMC and Gibbs sampling).

Review: Expectation, Variances, Covariances

- Definition of Expectation
  - Discrete case:
$$E[X] = \sum_{x:p(x)>0} xp(x)$$
  - Continous case:
$$E[X] = \int_{-\infty}^{\infty} xf(x)dx$$
  - Expectations of functions
    * g(X) is itself a random variable.
    * In simple cases, $E[g(X)]$ can be computed from $E[X]$. For example:

        - $E[aX + b] = aE[X] + b$
      - In more complicated cases we would have to compute the integral $\int g(x)f(x)dx$, or the discrete analog.
  - Another way to calculate expectations:

$$E[X] = \int_0^\infty [-F(-x) + (1 - F(x))]$$

- Definition of variance

$$Var(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

- Definition of covariance
  - Definition

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])] = E[XY] - E[X]E[Y]$$

  - If X,Y are independent, covariance equals 0.
  - Useful result:

$$Var(aX + bY + c) = a^2Var(X) + b^2Var(Y) + 2abCov(X, Y)$$

The Law of Large Numbers and introduction to Monte Carlo

- **The Strong Law of Large Numbers**: Let $X_1, X_2, \ldots$ be a sequence of independent, identically distributed variables, and let $E[X_i] = \mu$ (where μ is finite). Then,

$$P(lim_{n\to\infty} \frac{X_1 + X_2 + \ldots + X_n}{n} = \mu) = 1$$

- This result forms the basis of "vanilla" Monte Carlo estimators:
  - For expectations:

$$E[g(X)] \approx \frac{1}{M} \sum_{i=1}^{M} g(x_i)$$

    where $x_i \sim f_X(\cdot)$
  - For probabilities (using indicator functions):

$$P(X = x) = E[I_{X=x}] \approx \frac{1}{M} \sum_{i=1}^{M} I_{X=x}(x_i)$$

    where $x_i \sim f_X(\cdot)$

- Thus by being able to simulate instances of a random variable X we can compute probabilities of events dependent on X as well as computing expectations that require integrating over all possible values of X.
- This "Monte Carlo" strategy is a workhorse of modern computational statistics. It also has many variants, several of which we'll learn about in the course (e.g. Gibbs, MCMC).

Conditional expectations and variances

- Definition of Conditional Expectation
  - Discrete case:

$$E[X|Y=y] = \sum_x xP(X=x|Y=y) = \sum_x xp_{X|Y}(x|y)$$

    where $p_{X|Y}(x|y) = p(x,y)/p_Y(y)$
  - Continuous case:

$$E[X|Y=y] = \int_\infty^\infty xf_{X|Y}(x|y)dx$$

    where $f_{X|Y}(x|y) = f(x,y)/f_Y(y)$.
  - Note:
    * Simple, it's just an expectation over a conditional distribution/density function.
    * And note, $E[X|Y=y]$ is a random variable that is a function of y. Thus we can compute it's expectation: $E[E[X|Y]]$. This turns out to be very useful...

- Computing Expectations, Variances and Probabilities by Conditioning
  - Computing expectations of conditional expectations gives us a new route to computing an expectation (**Law of total expectation**):

$$E[X] = E[E[X|Y]]$$

  - We can also compute variances (**Law of total variance**):

$$Var(X) = E[Var(X|Y)] + Var(E[X|Y])$$

– And for computing probabilities (using indicator variables)

$$I_E = \begin{cases} 1 & \text{E happens} \\ 0 & \text{otherwise} \end{cases}$$

$$E[I_E] = 1P(I_E = 1) + 0P(I_E = 0) = P(E)$$

- Examples of using conditioning to compute probabilites:
  - Ross Example 3.10 and 3.19 : Mean and Variance of a Compound Variable
  - Example 3.10: Expected number of accidents in a week is 4 and the number of workers injured in each accident is an indpt RV with mean 2. What is the number of expected injuries during a week?

    Solution: Let N denote the number of accidents, and X_i the number injuries per accident. Our interest is:

    $$E[\sum_1^N X_i] = E[E[\sum_1^N X_i|N]]$$

    Note:

    $$E[\sum_i^n X_i|N = n] = E[\sum_1^n X_i] = nE[X]$$

    and then plugging in get:

    $$E[E[\sum_1^n X_i|N]] = E[nE[X]] = E[N]E[X]$$

    This is kind of obvious but now we've been rigorous about it. More interestingly, what about the variance?
  - Example 3.19: Let S be the compound variable $\sum_1^N X_i$. Find the variance. Let $Var(X) = \sigma^2$ and $E[X] = \mu$. We'll use the conditional variance formula.

    Solution:

    $$Var(S) = E[Var(S|N)] + Var(E[S|N])$$

    First term:

    $$Var(S|N = n) = Var(\sum_{i=1}^n X_i) = n\sigma^2$$

    $$E[Var(S|N)] = E[N]\sigma^2$$

Second term:

$$E[S|N] = n\mu$$

$Var(E[S|N])$ then equals $\mu^2 Var(N)$

So we have: Var(S)=σ^2 E[N]+μ^2 Var(N). In special case where N is Poisson(λ) we have:

$$Var(S) = \lambda\sigma^2 + \lambda\mu^2$$

which note has the simplification: $\lambda E[X^2]$.

Conclusions for the day

- For working on probablity problems...
  - Conditioning often helps
  - Use indicator variables to your advantage
  - Train yourself to recognize probablity distributions when they appear (as in Example 3.23 with the Poissons appear)
  - Sometimes its useful to remember distributions sum (or integrate to 1) (see Ross 3.22 for an example with a Gamma that appears in the simplified form).
  - Use tools from "real analysis":
    - Recognize that many ugly looking sum's or integrals have analytic solutions (e.g. see Example 3.25 or section 3.63). Mathematica can help recognize these
    - Proofs using induction are often needed. Similarly, recursive formulas often arise and can be solved (Example 3.26).
  - Advanced:
    - Using probablistic inequalities to form bounds
    - Using moment generating functions and characteristic functions for solving problems with sums of random variables

Miscellaneous Review

- Cumulative distribution functions and density functions
  - Cumulative distribution function: $F(b) = P(X <= b)$
    - F(b) is non-decreasing in b
    - $\lim_{b \to \infty}$ = F(∞) = 1
    - $\lim_{b \to \infty}$ = F(-∞) = 0

    - CDF's take the form of step functions for discrete RVs
    - For continuous RV's
        - $F(a) = P(X \in (\infty, a)) = \int_{-\infty}^{a} f(x)dx$
        - $\frac{d}{da}F(a) = f(a)$, ie density is the derivative of the cdf

- Definition of Covariance

$$E[X, Y] = E[XY] - E[X]E[Y]$$

  Properties of covariance:

  - $Cov(X, X) = Var(X)$
  - $Cov(X, Y) = Cov(Y, X)$
  - $Cov(cX, Y) = cCov(X, Y)$
  - $Cov(X, Y + Z) = Cov(X, Y) + Cov(X, Z)$

- The **Chain Rule**
  - In its basic form:

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

  - Which generalizes as:

$$P(A_1, A_2, \ldots, A_k) = P(A_1)P(A_2|A_1)\ldots P(A_k|A_{k-1})$$

  - This result holds regardless of the ordering.

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 2 — January 7

Lecturer: John Novembre *Scribe: Matt Bonakdarpour*

Note: These lecture notes are still rough, and have only have been mildly proofread.

2.1 Graphical Models: Introduction

Quite often, we are confronted with the task of understanding a complex system with many dependent components. In this context, probability theory gives us a framework to reason about a collection of possible outcomes and their associated likelihoods. For example, if we are trying to diagnose a patient, there are many potential diseases this patient could have. Also, associated with each patient, we may have hundreds of data points related to their symptoms, personal traits, and diagnostic tests. Each of these characteristics could be thought of as a random variable. Our goal is to reason about this patient given the values of one or more of these random variables. In the framework of probabilistic reasoning, we need to construct a joint distribution over the space of possible assignments to this set of random variables.

If we have five realizations of *binary* random variables for a patient, we must specify a joint distribution over 2^5 possible values – a potentially daunting task! Graphical models provide a framework to compactly encode a joint distribution in high dimensions. They also provide other benefits, including but not limited to:

- A tool for visualizing the structure of a probabilistic model
- Provides insights, such as conditional independence properties, by inspection of the graph
- Complex computations can be expressed as operations on the graph

In this lecture, we will consider directed graphical models, sometimes known as Bayesian networks.

2.1.1 Joint probabilities and independence

Using the definition of conditional probability, we have that the joint distribution between X_i and X_j is:

$$p(X_i, X_j) = P(X_i|X_j)P(X_j)$$

In general, if we have any n random variables $X_1, \ldots, X_n$, we can use the *chain rule* to factor the joint distribution as follows:

$$p(X_1, \ldots, X_n) = p(X_1)p(X_2|X_1)p(X_3|X_1, X_2) \cdots p(X_n|X_1, \ldots, X_{n-1})$$

Recall that X_i is independent of X_j if knowledge of X_j does not change our knowledge of X_i. That is, $p(X_i|X_j) = p(X_i)$. Plugging this in to the expression for the joint distribution above, we have that X_i is independent of X_j if and only if $p(X_i, X_j) = p(X_i)p(X_j)$.
We say that X_i is *conditionally independent* of X_j given X_k if

$$p(X_i, X_j|X_k) = p(X_i|X_k)p(X_j|X_k)$$

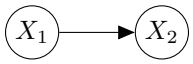
which means

$$p(X_i, X_j, X_k) = p(X_i|X_k)p(X_j|X_k)p(X_k)$$

Notice how the joint distribution factorizes nicely when we have conditional independence. Idea: when faced with a large number of features, use our prior knowledge of independence to simplify the joint distribution.

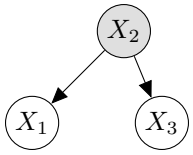
2.1.2 Some basics

A directed graph is a set of vertices (or nodes) along with a set of directed edges (or arrows) between nodes. Here is a simple graph with two vertices and one directed edge:



Here, each vertex represents a random variable. When a specific random variable in the graph is observed, we shade that particular vertex. For example, this is what the graph above would look like if we observed X_1, but X_2 was still latent:

A directed graphical model encodes conditional independence in the following way. A specific random variable (or vertex) is conditionally independent of all other nodes, given its parents (i.e. all nodes that points to it). For example, in the following graph X_1 is independent of X_3 given X_2:



The general form for the joint distribution for a directed graphical model is therefore:

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|\text{pa}_i) \tag{2.1}$$

where pa_i denotes the parents of X_i.

2.1.3 A motivating example

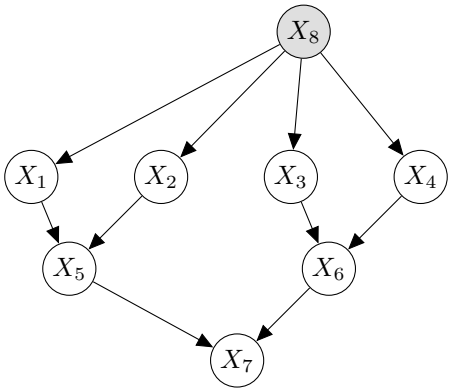
Assume we have 3 possible genotypes {AA,Aa,aa}. To each person, we will assign a random variable X_i according to their genotype:

Genotype	X_i
AA	0
Aa	1
aa	2

Let's also assume that the genotype of each person is binomially distributed, so that:

$$\begin{aligned} X_i|p &\sim \text{Bin}(2, p) \\ p(X_i = 2|p) &= p^2 \\ p(X_i = 1|p) &= 2p(1-p) \\ p(X_i = 0|p) &= (1-p)^2 \end{aligned}$$

In this context, we represent inheritance in a 3-generation pedigree with the following graphical model. In what follows, we're assuming $X_8 = p$.



Assume X_8 is observed and that we are interested in $p(X_5|X_1, X_2)$. We could represent this conditional probability as a table, enumerating all possibly quantities that X_1, X_2 and X_5 take. In general, we can represent $P(X_i|X_j, X_k)$ where X_i is the child of X_j and X_k using the basic laws of Mendelism:

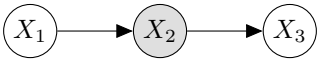
Father	Mother	Child (X_i)		
X_j	X_k	0	1	2
0	0	1	0	0
0	1	$\frac{1}{2}$	$\frac{1}{2}$	0
0	2	0	1	0
1	0	$\frac{1}{2}$	$\frac{1}{2}$	0
1	1	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
1	2	0	$\frac{1}{2}$	$\frac{1}{2}$
2	0	0	1	0
2	1	0	$\frac{1}{2}$	$\frac{1}{2}$
2	2	0	0	1

From the graphical model above, and using equation 2.1, we can factorize the joint distribution in the following way, in which the values of the table above can be used to compute the joint distribution:

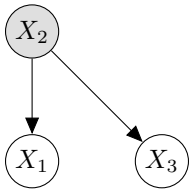
$$p(X_1, \ldots, X_8) = p(X_1|X_8)p(X_2|X_8)p(X_3|X_8)p(X_4|X_8)\times \\ p(X_5|X_1, X_2)p(X_6|X_3, X_4)p(X_7|X_5, X_6)p(X_8)$$

2.1.4 Some classes of 3-node graphs

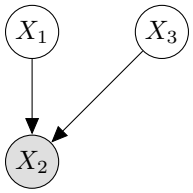
With the linear chain graph, we have that $X_1 \perp X_3|X_2$:



As we saw in a previous section, in this multiple offspring graph, we have that $X_1 \perp X_3|X_2$:



The v-structure graph has the following structure:



In this case, it is *not true* that $X_1 \perp X_3|X_2$. For example, assume X_2 was the event that your house alarm was going off, X_1 was the event that there was an earthquake, and X_3 was the event that there was a burglar in your house. Assuming your alarm was going off, the additional information about whether a burglar was in your house would affect your knowledge of whether or not an earthquake was happening.

2.1.5 D-separation

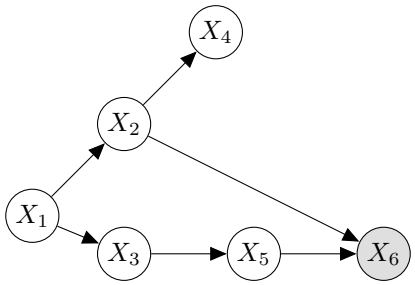
Take any undirected path (ignoring arrows) in the graph G. This path is called an *active trail* for observed variables $O \subset \{X_1, \ldots, X_n\}$, if for every consecutive triple of variables X, Y, Z on the path:

- $X \to Y \to Z$ and $Y \notin O$
- $X \leftarrow Y \leftarrow Z$ and $Y \notin O$
- $X \leftarrow Y \to Z$ and $Y \notin O$
- $X \to Y \leftarrow Z$ and Y or any of Y's descendants are in O

Any two variables X_i and X_j for which there does not exist an active trail for observations O are called *d-separated* by O, written d-sep$(X_i; X_j|O)$. Two sets of vertices A and B are d-separated by O if d-sep$(X, Y|O)$ for all $X \in A$ and $Y \in B$.
The key result is the following: if X and Y are d-separated by a set O, then $X \perp Y|O$. In words, X is conditionally independent of Y given O if there does not exist any active trail between X and Y for observations O. Intuition: active trails allow the dependencies to flow.

2.1.6 Elimination Algorithm

Assume we have a graph G, and we take two disjoint subsets of nodes X_E and X_Q. Our goal is to calculate $p(X_Q|X_E)$. In this section, we will focus on the case when X_Q is one node called the *query node*, and the set of nodes X_E are called *evidence nodes*.
Assume we have the following graph:



Let our evidence set be $\{X_6\}$ and our query set be $\{X_1\}$. We want to compute:

$$p(X_1|X_6) = \frac{p(X_1, X_6)}{p(X_6)} = \frac{p(X_1, X_6)}{\sum_{x_1} p(X_1 = x_1, X_6)}$$

The elimination algorithm provides an effective computational method for making these calculations.

From the expression above, it seems that to compute the conditional distribution, we just need to be able to compute the joint distribution $p(X_1, X_6)$. We could start off by marginalizing the full joint distribution for all variables:

$$p(X_1, X_6) = \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} p(X_1, X_2, \ldots, X_6)$$

In this case,if every X_i can take on one of k values, this expression would have k^4 terms. However, using the conditional independence relations from the graph, we get:

$$\begin{aligned} p(X_1, X_6) &= \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} p(X_1)p(X_2|X_1)p(X_3|X_1)p(X_4|X_2)p(X_5|X_3)p(X_6|X_2, X_5) \\ &= p(X_1)p \sum_{x_2} (X_2|X_1) \sum_{x_3} p(X_3|X_1) \sum_{x_4} p(X_4|X_2) \sum_{x_5} p(X_5|X_3)p(X_6|X_2, X_5) \end{aligned}$$

We can further simplify this summation by introducing the following notation. Let $m_i(x_{S_i})$ denote the expression that arises when performing the sum $\sum_{x_i}$ where x_{S_i} are the variables, other than x_i that appear in the summand. Employing this notation, we get:

$$\begin{aligned} p(X_1, X_6) &= p(X_1)p \sum_{x_2} (X_2|X_1) \sum_{x_3} p(X_3|X_1) \underbrace{\sum_{x_4} p(X_4|X_2)}_{m_4(x_2)} \underbrace{\sum_{x_5} p(X_5|X_3)p(X_6|X_2, X_5)}_{m_5(x_2,x_3)} \\ &= p(X_1)p \sum_{x_2} (X_2|X_1)m_4(x_2) \underbrace{\sum_{x_3} p(X_3|X_1)m_5(x_2, x_3)}_{m_3(x_1,x_2)} = p(X_1)p \sum_{x_2} (X_2|X_1)m_4(x_2)m_3(x_1, x_2) \\ &= p(x_1)m_2(x_1) \end{aligned}$$

Using this result, we get the desired conditional probability:

$$\Rightarrow p(X_1|X_6) = \frac{p(x_1)m_2(x_1)}{\sum_{x_1} p(x_1)m_2(x_1)}$$

This exercise gives rise to a general algorithm for computing marginal probabilities. To see the details of this algorithm, refer to the handout given in class entitled *The Elimination Algorithm*.

In statistical genetics, Felsenstein's tree-pruning algorithm (1981, JME) computes the likelihood of an evolutionary tree from nucleic acid sequence data. This pruning algorithm was an early instance of the elimination algorithm. As a simplified illustration, we can represent a phylogeny as a tree in which the leaves correspond to the *observed* values of a site across different species. The non-leaves are values of the site for the ancestors. Each edge of the tree can represent the *evolutionary distance* to be estimated which can be encoded as the conditional probability of a state given its ancestral state. Using Felsenstein's pruning algorithm, we can obtain the joint probability across all sites. This can be used in conjunction with the EM algorithm to find the maximum likelihood tree.
Relevant online course notes:

- `http://www.inf.ed.ac.uk/teaching/courses/pmr/slides/elim-2x2.pdf`
- `https://www.cs.cmu.edu/~aarti/Class/10701/readings/graphical_model_Jordan.pdf`
- `http://www.cs.columbia.edu/~blei/fogm/2015F/notes/inference.pdf`
- `http://www.cs.berkeley.edu/~jordan/courses/281A-fall04/`

For drawing graphical models

- `https://github.com/jluttine/tikz-bayesnet`

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 3 — January 12

Lecturer: Matthew Stephens *Scribe: Arjun Biddanda*

Note: These lecture notes are still rough, and have only have been mildly proofread.

3.1 Introduction

The primary theme of todays lecture is on likelihoods, likelihood ratios, and their interpretation. The writeup here *heavily* samples from the excellent vignettes found here: `http://stephens999.github.io/fiveMinuteStats/analysis/likelihood_ratio_simple_models.html`.

3.2 Example : DNA Barcoding of Poached Elephant Tusks

Elephant ivory is a commonly poached item, and we would like to determine which particular environment the elephants are coming from. The elephants can either come from (1) the savannah or (2) the forest. We denote the following two models and data from an elephant tusk:

M_s : Tusk comes from a savannah elephant

M_f : Tusk comes from a forest elephant

Marker	Allele	f_S	f_F
1	1	0.4	0.8
2	0	0.12	0.2
3	1	0.21	0.11
4	0	0.12	0.17
5	0	0.02	0.23
6	1	0.32	0.25

f_S and f_F represent the allele frequency of the "1" allele in the savannah and forest elephant populations respectively. The likelihood of the model M_S can be defined as the probability of the data being generated under the model.

$$
\begin{aligned}
L(M_S) &= P(Data|M_S) \\
&= \prod (f_S)^x \times (1 - f_S)^{1-x} \\
&= (0.4)(1 - 0.12)(0.21)(1 - 0.12)(1 - 0.02)(0.32) \\
&= 0.020399 \\
L(M_F) &= P(Data|M_F) \\
&= \prod (f_F)^x \times (1 - f_F)^{1-x} \\
&= (0.8)(1 - 0.2)(0.11)(1 - 0.17)(1 - 0.23)(0.25) \\
&= 0.0112 \\
LR(M_S/M_F) &= \frac{L(M_S)}{L(M_F)} \\
&= \frac{0.0204}{0.0112} \approx 1.8
\end{aligned}
$$

We have defined the likelihood ratio as the ratio of the likelihood of the model M_S over the likelihood of M_F. We also note that the models M_S and M_F are *fully-specified*, that is to say that there are no free parameters in the model. We will explore unspecified models later on in the lecture.
Two distinct questions arise as a result of calculating this likelihood ratio : (1) how to interpret the likelihood ratio and (2) when can we claim that we believe a model over another model given a likelihood ratio?

3.2.1 Context of Individual Likelihoods

The purpose of a likelihood ratio is to examine the evidence (data) in the context of one model against another. As a brief toy example let us consider a fair coin tossed 100 times and landing with 50 heads and 50 tails. What is the likelihood of the model that this coin is a fair coin? $L(M_{fair}) = \left(\frac{1}{2}\right)^{100}$. However this very small number is simply a probability, and actually getting any set of 100 tosses with a fair coin results in this likelihood! Thus we can see that likelihoods are important by providing a context under which they can be interpreted, by comparing two models against each other.

3.2.2 Notational Things

- We often work in "log-space" since the individual likelihoods may be quite small. The Log-Likelihood Ratio (LLR) is defined as $log(LR)$
- In english we would say $P(Data|M_0)$ as "The likelihood under the model M_0"

- Sometimes semicolons are used to denote the data, and curly a "L" for the likelihood (i.e. $log(LR) = log\left(\frac{\mathcal{L}(M_S;D)}{\mathcal{L}(M_F;D)}\right)$)

3.3 Example : Continuous Measurement of Protein in Blood

Suppose that we a protein that is measured in the blood and we call this random variable X. We want to see if this protein varies in concentration according to disease or non-disease status We wish to test the following two models :

$$M_n : X \sim Gamma(0.5, 2)$$
$$M_d : X \sim Gamma(1, 2)$$

Where M_n is the model under a non-diseased state and M_d is a model under the diseased state. If we observe the data as $X = 4.02$ the likelihood ratio can be detemined as:

$$LR = \frac{f_{X|M_n}}{f_{x|M_d}}$$

. However we would only like to compare the density around the measurement we have actually obtained, so we will use a quick trick and assume the precision of the measurement of X to be ± 0.05 making $X \in [4.015, 4.025]$. Now that we have discretized this measurement we can integrate the respective conditional densities over this range, making the likelihood ratio :

$$LR = \frac{\int_{4.015}^{4.025} f_{X|M_n}}{\int_{4.015}^{4.025} f_{X|M_d}}$$

There are a couple of caveats to this approach as well that translate broadly to the calculation of likelihood ratios :

- The density function must be well-behaved in the integration bounds
- The data must be held the same between the models that we are comparing (transforming one and not the other is illegal!)
- If a likelihood ratio is 0 we can certainly say that

Different Support of Random Variables

Likelihood Ratios still work properly when we have different support for the models. Suppose we have a random variable X which corresponds to the roll of a standard 6-sided dice and the following two models:

$$M_6 : \text{All the dice rolls are a 6}$$
$$M_{fair} : \text{The dice is a fair dice}$$

We can imagine two scenarios from this : (1) when the roll is a 6 and (2) when the roll is not a 6. When X is a 6 we have a likelihood ratio of $LR = \frac{P(X|M_6)}{P(X|M_{fair})} = 1/(1/6) = 1/6$. However when $X \neq 6$ we can say that the likelihoo ratio is 0 since $P(X \neq 6|M_6) = 0$ and this is the numerator of our likelihood ratio.

3.4 Continuation of Disease Example

Let us assume that $Z_i = 1$ if patient is diseased, else $Z_i = 0$.

$$P(Z_i = 1|X_i = x) = \frac{P(X_i = x|Z_i = 1) \cdot P(Z_i = 1)}{P(X_i = 1)}$$
$$P(Z_i = 0|X_i = x) = \frac{P(X_i = x|Z_i = 0) \cdot P(Z_i = 0)}{P(X_i = 0)}$$
$$\frac{P(Z_i = 1|X_i = x)}{P(Z_i = 0|X_i = x)} = \frac{P(Z_i = 1) \cdot P(X_i = x|Z_i = 1)}{P(Z_i = 0) \cdot P(X_i = x|Z_i = 0)}$$
$$Odds_{Posterior} = Odds_{Prior} \times \text{Bayes Factor}$$

When the model is fully-specified the Bayes Factor is equal to the Likelhood Ratio. However the role of the prior odds also plays a large role in our interpretation of the posterior odds. For instance if we believe that the disease is very rare, then we will have to have a much higher likelihood ratio in order to truly believe that we have the disease. It is very important to consider the prior odds of having the disease.

3.5 Likelihood Functions and Partially Specified Models

Let us review our elephant example now. But let us assume that we have sampled 100 elephants only from the savannah and look at their alleles at one particular marker. We obtain 40 samples that carry the "1" and 60 samples that carry the "0" allele. We then want to evaluate a particular model and its likelihood :

$$M_q : \text{Allele frequency of the 1 allele is q}, q \in [0,1]$$
$$\mathcal{L}(M_q) = P(Data|M_q) = q^{40}(1-q)^{60}$$

One way in which we can compare two potentially different values of q would be to look at their difference in log-likelihood units. We would then look at $log(\mathcal{L}(M_{q_1})) - log(\mathcal{L}(M_{q_2}))$. If we get a log-likelihood difference of 2 then we know that $LR = e^2 \approx 7.4$.

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 4 — January 14

Lecturer: Matthew Stephens *Scribe: Justin Hsu*

Note: These lecture notes are still rough, and have only have been mildly proofread.

4.1 Likelihood Analysis

Consider the set of data of 100 tusks, 40 of which have the "1" allele, 60 with the "0" allele Then the data has the likelihood function

$$L(q) = P(Data|M_q) \tag{4.1}$$

. We can write this as

$$L(q) = q^{40}(1-q)^{60} \tag{4.2}$$

Now consider the log of the likelihood function:

$$\ell(q) = 40log(q) + 60log(1-q) \tag{4.3}$$

We can estimate q by finding the value of q that maximizes $L(q)$. This is known as the Maximum Likelihood estimator (mle), which we denote as $\hat{q}$. A useful feature is that the value that maximizes the likelihood function also maximizes the log likelihood function.

$$\begin{aligned} \hat{q} &= \operatorname*{argmax}_q L(q) \\ &= \operatorname*{argmax}_q \ell(q) \end{aligned} \tag{4.4}$$

This is useful because it is sometimes easier to find the maximum of $\ell(q)$.
Returning to the elephant tusk example, we find the maximum of the likelihood function by taking the derivative of the log likelihood.

$$\begin{aligned} \ell'(q) &= \frac{40}{q} - \frac{60}{1-q} \\ 0 &= \frac{40}{q} - \frac{60}{1-q} \\ \hat{q} &= \frac{40}{100} \end{aligned} \tag{4.5}$$

We can extend this generally so that given two populations n_1 and n_0, we have a likelihood function

$$L(q) = q^{n_1}(1-q)^{n_0} \tag{4.6}$$

and a log likelihood function

$$\ell(q) = n_1 log(q) - n_0 log(1-q) \tag{4.7}$$

Then the maximum likelihood estimate will have the form

$$\hat{q} = \frac{n_1}{n_1 + n_0} \tag{4.8}$$

This is the maximum likelihood of the Binomial Distribution.

4.2 Mixture Models

We now move on to mixture models, which our models that consist of a mixture of two or more distributions. As an example, consider the heights of all humans of these worlds. What would be the distribution of these heights. We could assume that they are normally distributed, but what if the male heights come from a different distribution than the female heights?
Suppose we have

$$\begin{aligned} maleheight &\sim N(\mu_m, \sigma_m^2) \\ femaleheight &\sim N(\mu_f, \sigma_f^2) \end{aligned} \tag{4.9}$$

and suppose the population is 50% male and 50% female.
Let X be the height of a randomly chosen person. What would be the density function for X?
If X was discrete, then

$$Pr(X = x) = Pr(X = x|male)Pr(male) + Pr(X = x|female)Pr(female) \tag{4.10}$$

The continuous analogue would be:

$$\begin{aligned} f_x(x) &= \frac{1}{2} f_{x|male}(x) + \frac{1}{2} f_{x|female}(x) \\ &= \frac{1}{2} N(X; \mu_m, \sigma_m^2) + \frac{1}{2} N(X; \mu_f, \sigma_f^2) \end{aligned} \tag{4.11}$$

We call the probabilities $Pr(male) = \frac{1}{2}$ and $Pr(female) = \frac{1}{2}$ the "mixture proportions". We call $f_{x|male}(x)$ and $f_{x|female}(x)$ the "component densities".
Returning to our elephant tusk example, suppose we have data $X = (X_1, X_2, ..., X_n)$ on n tusks, and that we know the allele frequencies.

Let the proportion of elephants that are Savannah be Π_S.
Let $Z_i \in \{S, F\}$ denote whether tusk i came from either a Savannah or Forest elephant. We call $\{S, F\}$ the "component labels".
Then we have the mixture model

$$\begin{aligned} P(X_i = x_i|\Pi_S) &= Pr(Z_i = S)Pr(X_i = x_i|Z_i = S) + Pr(Z_i = F)Pr(X_i = x_i|Z_i = F) \\ &= \Pi_S Pr(X_i = x_i|Z_i = S) + (1 - \Pi_S)Pr(X_i = x_i|Z_i = F) \end{aligned} \tag{4.12}$$

More generally

$$PrX_i = x_i = \sum_k \Pi_k Pr(X_i = x_i|Z_i = k) \tag{4.13}$$

where $\Pi_1, ...\Pi_k$ are nonnegative and sum to 1.
The likelihood function of this mixture model is

$$\begin{aligned} L(\Pi_S) &= P(X|\Pi_S) \\ &= \prod_{i=1}^{n} P(X_i = x_i|\Pi_S) \end{aligned} \tag{4.14}$$

When we take the log of this likelihood function, we get

$$\begin{aligned} \ell(\Pi_S) &= \sum_{i=1}^{n} log(Pr(X_i = x_i|\Pi_S)) \\ &= \sum_{i=1}^{n} log[\Pi_S P(X_i = x_i|Z_i = S) + (1 - \Pi_S)P...] \end{aligned} \tag{4.15}$$

Unlike the example with the binomial distribution, this log likelihood is difficult to differentiate, so to find the maximum, we must rely on numerical methods.

4.3 EM Algorithm

The Expectation Maximization (EM) Algorithm is a method for finding maximum likelihood estimates for a model. The key idea behind the EM algorithm is "data augmentation". It is data which we do not have but wish we would have. Suppose our data is X, then the augmented data would be (X, Z), where Z is the "missing data".
Let $L(\theta) = P(X|\theta)$ be the "marginal likelihood"/"observed likelihood". The "complete likelihood" is $L_{comp}(\theta) = P(X, Z|\theta)$.
The steps of the EM algorithm are as follows:

1. Choose some θ_0

2. E step: Form the "expected" complete log likelihood by taking the expectation over Z. In other words find
$$Q(\theta, \theta_0) = E_{Z|X,\theta_0}[\ell_{comp}(\theta; Z, X)] \tag{4.16}$$

3. M step: Choose the value of θ which maximizes $Q(\theta, theta_0)$.

4. The maximizes theta is your new θ_0. Repeat the E and M steps until $\ell(\theta)$ does not change very much.

The advantage of the EM algorith is that the likelihood will always increase with each iteration.
Sometimes the algorithm will converge to a local optimum rather than a global optimum. In practice the algorithm is run multiple times.
Returning to elephant tusk mixture model, which has a complete likelihood

$$\begin{aligned} L(\Pi_S) = P(X, Z|\Pi_S) = \prod_{i=1}^{n} P(X_i, Z_i|\Pi_S) \\ = \prod_{i=1}^{n} P(Z_i|\Pi_S) \propto \prod_{i=1}^{n} \Pi_S^{\mathbb{1}_{Z_i=S}} (1-\Pi_S)^{\mathbb{1}_{Z_i=F}} \end{aligned} \tag{4.17}$$

$\mathbb{1}$ stands for the indicator function, which is 1 for the given event and 0 otherwise.
Taking the log of this expression, we get

$$log(\prod_{i=1}^{n} P(Z_i|\Pi_S)) = \underset{constant}{C} + \sum_i \mathbb{1}(Z_i = S) log(\Pi_S) + \sum_i \mathbb{1}(Z_i = F) log(1-\Pi_S) \tag{4.18}$$

If we take the expectation of the sum of indicator functions, we find the probability of that event occurring. We find that the log likelihood above is maximizes at

$$\frac{\sum_i E(\mathbb{1}(Z_i = S)|X, \theta)}{\sum_i E(...) = n} \tag{4.19}$$

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 5 — January 19

Lecturer: Matthew Stephens *Scribe: Alan Hutchison*

Note: These lecture notes are still rough, and have only have been mildly proofread.

5.1 Introduction to Bayesian Inference

See github.com/stephens999/stat302 for further notes by M. Stephens on Bayesian Inference (Lects. 1-4)

From last class: Posterior Odds = Prior Odds * Likelihood Ratio (aka Bayes Factor)

$$\frac{Pr(M_1|x)}{Pr(M_0|x)} = \frac{P(M_1)}{P(M_0)} * \frac{P(x|M_1}{P(x|M_0)}$$

This is the two model M_1 vs. M_0 case. We saw two examples of this in the previous lecture, regarding 1) Identifying Elephant Tusks and 2) Protein measurements in the blood. In these cases we were using likelihood ratios (LRs) for imagining a population that we're screening to tell us the conditional probability of a state (elephant type of tusk / disease state of patient) given data.

What if we only have 1 tusk and a LR of 1.8? We can think of the LR as a measure of the uncertainty of event (tusk is Savannah, tusk is Forest) based on everything we know. Instead of thinking in terms of a 'measure of frequency' we can think in terms of a 'measure of uncertainty'.

1. "50% chance of snow tomorrow" This is simplest understood as the uncertainty of snow, not as a prediction of a hypothetical population of days.
2. The answer to "Did it snow in Denver?" can be expressed as an uncertainty, even though the event 'snowed in Denver' is not a random variable.

One way of thinking about uncertainty is by "comparing an event to a standard", such as a flip of a (fair) coin or a roll of a (fair) dice. It is not unreasonable to start with these standards (usually uniformly distributed probabilities). Heuristically, if one's prior odds are off by a factor of 10, then it starts to be more problematic.

5.2 Comparison of more than two models

Now, we will look at k different models instead of 2 models, as we have done previously.

In the elephant case, for example, instead of thinking of just Forest vs. Savannah elephants, we could think about further subdividing these categories by genetic differences amongst the different cardinal directions (N E S W).

Let's refer to the different models as $Z_i \in \{1, 2, ..., k\}$, the likelihood of data $P(x_i|Z_i = k)$ as L_{ik} and the prior probability $P(Z_i = k)$ as Π_k. We are using subscript i to denote a series of tusks, but we could just as easily drop the $i's$.

Using the Law of Total Probability (LTP), we can write

$$P(x_i) = \sum_{k'} P(x_i, Z_i = k') = \sum_{k'} P(x_i|Z_i = k')P(Z_i = k') = \sum_{k'} L_{ik'}\Pi_{k'}$$

$$P(Z_i = k|x_i) = \frac{P(x_i|Z_i = k)P(Z_i = k)}{P(x_i)} = \frac{L_{ik}\Pi_k}{\sum_{k'} L_{ik'}\Pi_{k'}}.$$

Note that the denominator is simply the sum over k of all possible numerators. We use the symbol k' to denote the same set of states k, only for the sum across all k of them. This means we can simply compute $L_{ik}\Pi_k$ for each k, then divide by the sum of them at the end. If all k models are equally likely, then we end up just normalizing by the sum of the likelihoods.

Instead of using equalities, we can use proportions to avoid thinking about the normalizing factor. We can think about this as informing our posterior uncertainty by what we learn from the data (the likelihood) with what we knew or believed beforehand (the prior).

$$Posterior \propto Likelihood * Prior$$

Remember that the likelihoods themselves don't really matter, it's the ratios that give you information about the different models. In ratios the normalizing constant ends up canceling out altogether.

5.3 Continuum of models case: Parametric model

5.3.1 Estimating an allele frequency

As previously, sample 100 alleles, see 40 of type '1' and 60 of type '0'. D: Data. M_q: "freq. of allele '1' is q"

Originally we used,

$$L(q) = q^{40}(1-q)^{60}$$

Maximizing this provides a Maximum Likelihood Estimate (MLE) point estimate for q, but we may be interested incorporating prior information or belief for q. We can compute the posterior distribution of q using Bayes Theorem. Let our prior distribution on q be $P(q)$.

$$P(q|D) \propto q^{40}(1-q)^{60} * p(q)$$

Assuming that all values of q are equally probable $P(q) = 1$

$$P(q|D) \propto q^{40}(1-q)^{60}$$

now we need to normalize by $\int q'^{40}(1-q')^{60}dq'$. This normalization step can be a complicated integral, and there are some shortcuts that can be used here. Instead of solving out the integral, it is easier if we can recognize it as a standard distribution whose behavior we know.

Since we are used the assumption that $P(q) = 1$, i.e. all values of q are equally probable, we can also write this uniform distribution as a Beta distribution of q, $Be(q; \alpha, \beta) = q^{1-\alpha}(1-q)^{1-\beta}$ with parameters $\alpha = 1$ and $\beta = 1$, $P(q) = Be(q; 1, 1) = q^0(1-q)^0 = 1$.

$$P(q|D) \propto q^{40}(1-q)^{60} * p(q) = q^{40}(1-q)^{60}q^0(1-q)^0 = q^{40}(1-q)^{60}$$

Rewritten as a Beta distribution, we have

$$P(q|D) \propto q^{40}(1-q)^{60} = q^{\alpha-1=40}(1-q)^{\beta-1=60} = Be(q;\alpha,\beta) = Be(q;41,61)$$

Having an expression proportional to the posterior distribution $P(q|D)$, we have the means for summarizing q as either a point estimate or as an interval. For point estimates, common choices include the *posterior mean*, *posterior median*, and *posterior mode*. For an interval estimate, *posterior quantiles* allow us to describe the spread of likely values of q. Giving the 2.5% quantile and 97.5% quantile provides the interval for which there is a 95% probability that q lies in it.

There are two takeaways from this section

1. For computing the posterior distribution in the continuous case, we use the same procedure as the the discrete case but we have the trick of using conjugate prior distributions to make the math simpler.

2. Once we have the posterior distribution, we can summarize it with both point estimates and interval estimates.

5.3.2 Conjugacy

For many types of likelihood functions there are corresponding prior *conjugate* distributions such that when the two are multiplied together the resulting distribution is also in the same family of conjugate distributions. More formally, *conjugacy means that the posterior distribution comes from the same family as the prior distribution.* These priors may not be realistic interpretations of the underlying mechanism of the prior distribution, but they are powerful and useful. For the Beta distributions in particular, any distribution with support of [0,1] can be approximated as a mixture of Beta distributions.

A question in class asked if there were any properties of likelihood functions that would indicate whether a conjugate prior distribution exists. The short answer is no, though it has been proven that exponential likelihood families have conjugate priors. Many common distributions have been identified as having conjugate priors, and they are summarized in Table 1.

Let's return to the above example. Instead of assuming the uniform prior distribution $P(q) = Be(q;1,1)$, let's leave it as a general Beta distribution.

Observed Likelihood Function	Conjugate Prior
Binomial	Beta
Multinomial	Dirichlet
Mean of Normal	Normal
Variance of Normal	Inverse Normal
Mean of Poisson	Gamma

Table 1. Observed Likelihood Functions and their Conjugate Priors

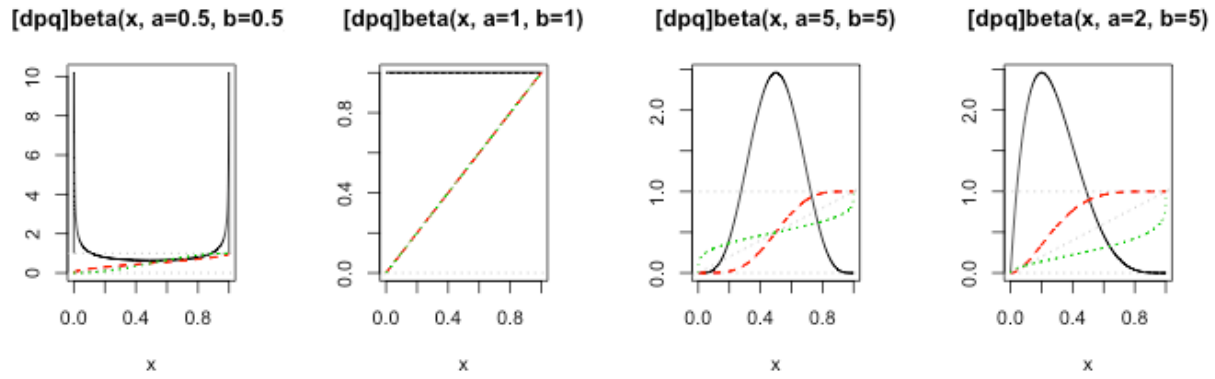


Figure 5.1. Beta distributions for different α and β values. Black refers to the density, red to the distribution function, and green the quantile function.

$$P(q|D) \propto q^{40}(1-q)^{60} * p(q) = q^{40}(1-q)^{60} q^{\alpha-1}(1-q)^{\beta-1} = q^{40+\alpha-1}(1-q)^{60+\beta-1} = Be(q; 40+\alpha, 60+\beta)$$

Here we can see exactly how the choice of the prior distribution parameters α and β can include the resulting posterior distribution. Figure 5.3.2 shows the Beta distribution for a variety of α, β combinations.

Given that it is reasonable to assume that the prior distribution of allele frequencies in population genetics is U-shaped, we can use a prior distribution of $Be(q; 0.5, 0.5)$.

$$P(q|D) \propto q^{40}(1-q)^{60} q^{1/2-1}(1-q)^{1/2-1} = q^{40-1/2}(1-q)^{60-1/2} = Be(q; 40.5, 60.5)$$

As you can see, this slightly changes our posterior distribution with a uniform prior, from $Be(q; 41, 61)$ to $Be(q; 40.5, 60.5)$

More generally, we can think of the observed alleles in our sample as n_1 of type "1" and n_0 of type "0". This means our posterior distribution will be

$$P(q|D) \sim \ Be(q; n_1 + \alpha, n_0 + \beta)$$

and the resulting posterior mean estimate for q would be

$$E[P(q|D)] = \frac{n_1 + \alpha}{n_0 + n_1 + \alpha + \beta}$$

Using a Maximum Likelihood Estimate (MLE) approach instead, our result would be independent of the prior distribution

$$E[L(q|D)] = \frac{n_1}{n_0 + n_1}$$

Likewise, estimating q solely off the prior distribution would be independent of observed data n_0 or n_1.

$$E[Be(q; \alpha, \beta)] = \frac{\alpha}{\alpha + \beta}$$

What this comparison of point estimates of q shows is that the posterior mean of q is a combination or tradeoff or compromise of the MLE estimate and the prior estimate: the results is 'weighted' by the evidence presented by n_0 and n_1, but is balanced against the prior information or uncertainty about the population encoded in α and β. If α and β are big relative to n_0 and n_1, then the posterior mean will be close to the prior mean. If n_0 and n_1 are big relative to α and β, then the posterior mean will be close to the MLE. "In sufficient quantities, data can overwhelm belief."

Note: in some situations, α and β can be thought of as pseudo-counts. This can be useful in a situation where you want a prior that strongly discounts the possibility of an event but you don't want to make it zero. In that case, you could use a prior of $Be(1, 100)$ to indicate the prior belief that it is extremely unlikely and then let the data determine whether the posterior probability of the event increases.

5.3.3 Example: Estimating Normal Mean

1 observation: data $X \sim N(\theta, \sigma^2)$
assuming σ^2 is known (e.g. measurement error).
Prior on θ, $\theta \sim N(\mu_0, \sigma_0^2)$. We use the subscript '0' to denote that these are prior parameters.

$$P(\theta|X = x) \propto \ prior * likelihood$$

$$P(\theta|X = x) \propto \ \exp[\frac{-(\theta - \mu_0)^2}{2\sigma_0^2}] \exp[\frac{-(x - \theta)^2}{2\sigma^2}]$$

$$P(\theta|X = x) \propto \ \exp[\theta^2[-\frac{1}{2\sigma_0^2} + \frac{1}{2\sigma^2}] + \theta[\frac{\mu_0}{\sigma_0^2} + \frac{x}{\sigma^2}]]$$

Given we are solving for the normal mean using a normal conjugate prior, we know our posterior distribution is going to have the form

$$P(\theta|X = x) \propto \ N(\mu_1, \sigma_1^2) = \exp[\frac{-(\theta - \mu_1)^2}{2\sigma_1^2}]$$

which can be re-written without the constants as

$$P(\theta|X = x) \propto \ \exp[\theta^2 \frac{-1}{2\sigma_1^2} + \theta \frac{-\mu_1^2}{2\sigma_1^2}]$$

Comparing this with the above equation, we can determine that

$$\frac{1}{\sigma_1^2} = \frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}$$

The inverse of the variance $\frac{1}{\sigma^2}$ is referred to as the precision, which is mathematically sometimes more tractable. This equation means that our posterior precision is the sum of our prior precision and our precision from our data. Our precision gets bigger, and our variance gets smaller.

$$\frac{\mu}{\sigma_1^2} = \frac{\mu_0}{\sigma_0^2} + \frac{x}{\sigma^2}$$

This means that the product of our posterior mean and our posterior precision is the sum of the product of our prior mean and prior precision with product of our observed x and observed precision.

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 6 — January 21

Lecturer: John Novembre *Scribe: Marcos Vieira*

Note: These lecture notes are still rough, and have only have been mildly proofread.

Discrete-time Markov Chains

$$X_1 \to X_2 \to ... \to X_t$$

Markov property:

$P(X_t = j \mid X_{t-1} = i_{t-1},\ X_{t-2} = i_{t-2},\ ...,\ X_1 = i_1) = P(X_t = j \mid X_{t-1} = i_{t-1})$

The chain is time-homogeneous if, for all t:

$P(X_t = j \mid X_{t-1} = i) = P_{ij}$

Since the system must move to one of the states:

$\sum_{j \in S} P_{ij} = 1$

We can collect the transition probabilities into a matrix $\mathbf{P}$. Because the rows of such a matrix specify probability distributions, the matrix is said to be a "stochastic matrix". For example:

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$$

An example from phylogenetics gives transition probabilities among nucleotides A, T, C and G, represented in the rows and columns:

$$\mathbf{P} = \begin{bmatrix} 0.999 & \frac{0.001}{3} & \frac{0.001}{3} & \frac{0.001}{3} \\ \frac{0.001}{3} & 0.999 & \frac{0.001}{3} & \frac{0.001}{3} \\ \frac{0.001}{3} & \frac{0.001}{3} & 0.999 & \frac{0.001}{3} \\ \frac{0.001}{3} & \frac{0.001}{3} & \frac{0.001}{3} & 0.999 \end{bmatrix}$$

An example from population genetics is the Wright-Fisher model for how the the number X_t of copies of an allele changes in a population over time:

$X_t \mid X_{t-1} = i \sim Binomial(N, \frac{i}{N})$

What happens in n steps?

$P_{ij}^{(n)}$: nth step transition probabilities.

$P_{ij}^{(n)} = P(X_{n+k} = j \mid X_k = i)$, for $n \geq 0$, $i, j \geq 0$

Chapman-Kolmogorov equations:

$P_{ij}^{(n+m)} = \sum_k {P_{ik}}^{(n)} {P_{kj}}^{(m)}$ for all $n, m \geq 0$

Or, equivalently, in matrix algebra:

$\mathbf{P}^{(n+m)} = \mathbf{P}^{(n)}\mathbf{P}^{(m)}$

In general, for discrete-time Markov chains:

$\mathbf{P}^{(n)} = \mathbf{P}^n$

Reducible vs. irreducible chains:

State j is *accessible* to state i if it's possible for the chain to move from i to j. If i is accessible to j, j is accessible to i, or both, states i and j *communicate*.

Class of states: a set of states that communicate.

Irreducible Markov chain: a Markov chain with a single class (all states communicate).

Recurrent or transient states:

f_i : probability of returning to i if starting at i, as $t \to \infty$

State i is:

Recurrent, if $f_i = 1$.
Transient, if $f_i < 1$.

Periodic vs. non-periodic states

A state is periodic if the chain can't stay in it and has to leave before moving to the same state again:

$P(X_{t+1} = i \mid X_t = i) = 0$

Otherwise, the state is non-periodic.

Ergodic Markov chains:

A discrete-time Markov chain is *ergodic* if it is irreducible and all of its states are recurrent and non-periodic.

If a discrete-time Markov chain is ergodic, then it's guaranteed to have a *stationary distribution* (also known as an equilibrium distribution). That is, from any initial probability distribution of the states $\boldsymbol{\pi}^{(0)}$, there is a unique $\boldsymbol{\pi}$ such that:

$$\lim_{t\to\infty} (\boldsymbol{\pi}^{(0)})^T \mathbf{P}^t = \boldsymbol{\pi}^T$$

And $\boldsymbol{\pi}$ satisfies:

$\boldsymbol{\pi}^T \mathbf{P} = \boldsymbol{\pi}$

Which we can solve for $\boldsymbol{\pi}$.

Eigenvalue decomposition of $\mathbf{P}$:

If $\mathbf{V}$ and $\boldsymbol{\Lambda}$ are the eigenvector and eigenvalue matrices of $\mathbf{P}$, respectively, then:

$\mathbf{P} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$

$\mathbf{P}^k = \mathbf{V}\boldsymbol{\Lambda}^k\mathbf{V}^{-1}$

Usually, $\lambda_1 = 1$, $\lambda_2, \lambda_3, ... < 1$.

Time reversibility:

$$Q_{ij} = P(X_{t-1} = i \mid X_t = j) = \frac{P(X_t = i \mid X_{t-1} = j)P(X_{t-1} = j)}{P(X_t = i)}$$

If we are at the stationary distribution $\boldsymbol{\pi}$, this becomes:

$$Q_{ij} = \frac{P_{ij}\pi_j}{\pi_i}$$

In the special case where

$$P_{ij} = \frac{P_{ji}\pi_j}{\pi_i}$$

The chain is said to be *time-reversible*:

$\pi_i P_{ij} = \pi_j P_{ji}$

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 7 — January 26

Lecturer: Prof. Stephens *Scribe: Rahul S. and Daniel Z.*

Note: These lecture notes are still rough, and have only have been mildly proofread.

7.1 Markov Chain Monte Carlo

Given $X, Y \in 0, 1$ such that $P(X = x, Y = y) = :$

X / Y	0	1
0	0.6	0.1
1	0.15	0.15

An alternative way to simulate from this joint distribution using the conditional probabilities would be:

$$Pr(X = 0|Y = 0) = \frac{0.6}{0.75} = 0.8 \tag{7.1}$$

$$Pr(X = 1|Y = 0) = \frac{0.6}{0.75} = 0.2 \tag{7.2}$$

$$Pr(X = 0|Y = 1) = \frac{0.1}{.25} = 0.4 \tag{7.3}$$

$$Pr(X = 1|Y = 1) = \frac{.3}{.5} = .6 \tag{7.4}$$

Method:
Set $x_0 = 0$, $y_0 = 0$,
Iterate:
at step $i = 1, 2, 3..$

- simulate x_i from the conditional distribution $Pr(x|y = y_{i-1})$
- simulate x_i from the conditional distribution $Pr(y|x = x_{i-1})$

The theory: for large enough N: $Pr(X_n = x_n, Y_n = y_n) = Pr(X = x, Y = y)$ based only on the conditional probabilities.
In this example we simulated a Markov Chain. The stationary distribution of this Markov Chain converges to $Pr(X = x, Y = y)$.
Chain $c_1, c_2, c_3,$. where $c_1 = (x_1, y_1), c_2 = (x_2, y_2), c_3 = (x_3, y_3)$. $c_i$only depends on c_{i-1} and is therefore a Markov Chain.

7.2 Proof

Proof of convergence to the stationary distribution $Pr(X = x, Y = y)$ (intuitive explanation):
Once we reach a stationary distribution π we remain there: $\pi * P = P$. Suppose we reached the stationary distribution s.t. $Pr(X = x, Y = y)$: At the first step of the iteration:

$$Pr(X_i = x, Y_i = y) = Pr(X_{i=x}|Y_{i=y})Pr(Y_{i=y}) \tag{7.5}$$

$$= Pr_{x|y}(X_i = x|Y_i = y)Pr_y(Y = y) \tag{7.6}$$

= (Conditional distribution from the table)*(marignal distribution from table)

$$Pr(X_i = x, Y_i = y) = Pr(X_i = x|Y_{i=y})Pr(Y_{i=y}) \tag{7.7}$$

$$= Pr_{x,y}(X = x, Y = y) \tag{7.8}$$

As long as x and y come from the right marginal after step 1 and step 2 x_{i+1}, y_{i+1} are from the correct target distribution. As such, if we start the algorithm from the stationary distribution we stay at the joint distribution.
note: this Markov Chain is irreducible, recurrent, finite and aperiodic therefor it will converge to its stationary distribution. There are different definitions for the convergence of distributions such as almost surely.
This proof can be extended to N variables e.g. x,y,z: a) simulate x conditional on y and z b) simulate y conditional on x and z c) simulate z conditional on x and y
alternatively, we can use the algorithms for subsets of variables: a)simulate x given y and z b)simulate y and z given x
our proof still works when we add conditionals which is useful for Bayesian methods.
Example: Genetic data on elephant tusks: Some elephants are from a savanna population some from a forest population, no prior knowledge on which is which. Cluster data based on a mixture model using Haploid Elephants markers:
Data: $x = x_1, .x_n$ n tusks π_1, π_2= proportion from each of the two groups F_1, F_2 = marker allele frequency in the two groups Latent variables: $z = z_1, .z_n$ where z_i is the group origin of tusk i

"Complete" Data Likliehood: $Pr(x, z|F, \pi) =$

$$= \prod_{i=1}^{n} P(X_i|Z_i, F, \pi) * P(Z_i|f, \pi) \tag{7.9}$$

$$Likelihood = P(x|f, \pi) = \pi_i * \sum_k \pi_k P(X_i|Z_i, F, \pi) \tag{7.10}$$

7.3

7.3.1 Likelihood calculation

Complete data likelihood.

$$\prod_{i=1}^{n} P(z_i|F, \pi) \cdot P(x_i|z_i) = \prod_{i=1}^{n} \pi_{z_i} \cdot \prod_j F_{k1j}^{k_j} \cdot (1 - F)^{1-k_j} \tag{7.11}$$

product across individual i marker j where F_{k1j} is the frequency of an allele of a of a marker in a given population group k Likelihood.

$$Likelihood = P(x|F, \pi) = \prod_{i=1} (\sum_k \pi_k \cdot P(x_i|z_i = k, F, \pi)) \tag{7.12}$$

to simulate data decide on group 1 or 2, and simulate alleles.

7.3.2 Gibbs sampling

Gibbs sampling for $P(z, pi, Fx)$:
1) sample from $z|\pi, F, (x)$ (use current value of π and F to generate z group origins)
2) sample from $\pi, F|z, (x)$

where $\pi, F|z, (x) \propto P(\pi, F|Z, x) \propto P(\pi)P(F)P(z, x|\pi, F)$
if $\pi \sim Be(\alpha_\pi, \beta_\pi)$ is the prior then:
$\pi|z, x \sim Be(\alpha_{pi} + \#k_i = 1, \beta_\pi = \sum z_i = z$

$\pi|z, x \sim Be(\alpha_\# Fj + \#, \beta_{k,j} + \#0_s...$ (allele number j from pop. k

$$p(\pi, f|z, x) \alpha p(\pi) * p(f) * p(z, x|pi, f) \tag{7.13}$$

this algorithm converges to a distribution rather than a point estimate which allows for calculating confidence bounds.

Gibbs sampling is one way of generating a markov chain when we can directly sample from the conditional probabilities but not from the joint distribution

7.4 Metropolis Hastings Algorithm

How do we generate a sample from a Markov chain with stationary distribution $\pi(x)$

1) π is reffered to as the target distribution and has to be defined up to a constant of proprtion α
2) Makrov transition matrix Q (also known as the "kernel") from which it is possible to simulate from: s.t. for any given current state of x you can simulate the next state of the Markov Chain with transition matrix Q. We have to know how to compute Q or what is the probability of transitioning.

7.4.1 algorithm

Providing Q, π
i) initialize $x \in X$
ii) at step i=1,2 ....

- let x be the current value of $x = x_{i-1}$
- generate a proposed value of x, x' by simulating one step of Q
- with probablity A set $x = x'$ otherwise set $x_i = x(x_i = x_{i-1})$ where

$$A = \frac{\pi(x')Q(x' -> x)}{pi(x)Q(x -> x')} \bigwedge 1 \tag{7.14}$$

the $\frac{Q()}{Q()}$ part is called the hastings part
this is a bit reminiscent of the detalied balance equation

7.4.2 example

Example of Q: add a small random deviate to x if $\pi \sim exp(1)$ and Q adds random normal the probability of moving back and forth $+ - 1$ is the same since Q is normal which is symetric

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 9 — February 2

Lecturer: John Novembre *Scribe: Frank Wen*

Note: These lecture notes are still rough, and have only have been mildly proofread.

9.1 Continuous time processes (the exponential distribution)

The exponential distribution often appears as the distribution of waiting times until an event (i.e. arrival times).

$$f(x) = \lambda e^{-\lambda x}$$

Let T be a random variable representing the waiting time until an event happens. Then for $T \sim exp(\lambda)$ with expected value $E(T) = \frac{1}{\lambda}$. The cumulative distribution function is given by

$$F(X) = P(T \leq X) = 1 - e^{-\lambda x}$$
$$P(T > X) = e^{-\lambda x}$$

An important property of the exponential distribution is that it is *memoryless*. Suppose that T is the waiting time until an event occurs. Then, for any given waiting times t and s,

$$P(T > t + s | T > t) = P(T > s)$$

This is to say that if we have already waited t minutes for an event to occur, then the remaining time s for the event to occur is the same as if we hadn't waited the first t minutes to begin with. To show this, we use our definition of conditional probability:

$$P(T > t + s | T > t) = \frac{P(T > t + s, T > t)}{P(T > t)}$$

Since we have already waited t minutes, $P(T > t+s, T > t) = P(T > t+s)$. So,

$$\begin{aligned} P(T > t+s | T > t) &= \frac{P(T > t+s)}{P(T > t)} \\ &= \frac{e^{-\lambda(t+s)}}{e^{-\lambda t}} \\ &= e^{-\lambda s} = P(T > s) \end{aligned}$$

9.2 Discrete time processes (the geometric distribution)

The geometric distribution can be thought of as the discrete analog of the exponential distribution. It is the the probability distribution of X number of Bernoulli trials with success probability p before one success is obtained.

$$Geom(p) = P(X = k) = (1-p)^{k-1}p$$

Because the trials are independent, the geometric distribution is also memoryless.

9.3 Poisson processes

9.3.1 Counting process and the poisson distribution

A poisson process can be defined as a counting process, where we are interested in the number of events (arrivals) that have occurred up until some time $N(t)$. Four properties of this counting process are the following
1) $N(0) = 0$
2) $N(t), t \geq 0$ has independent increments, or what happens in one time interval is independent of what happens in any other interval
3) $P(N(t_0 + t) - N(t_0) \geq 2) = o(t)$ as $t \to 0$
As the time between events gets smaller, the probability of observing two or more events in that time step goes to zero. In other words, events do not happen simultaneously.
4) $P(N(t_0 + t) - N(t_0) = 1) = \lambda t + o(t)$ as $t \to 0$
i.e. the number of events in an interval of length t is Poisson distributed with rate λt. We write this as

$$P(N(t) = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$$

Here, $f(x) = o((g(x))$ means that $|f(x)| \leq k|g(x)|$ for all k. Intuitively this means that $g(x)$ grows faster than $f(x)$

9.3.2 Relationship to the binomial distribution

The poisson distribution can be derived as the limit of a binomial distribution as the number of trials n goes ti infinity, and the probability of success p goes to zero, such that $np = \lambda$. To show this, we take an interval $(s, s+t]$ and divide it into n intervals of size t/n. The number of events in the ith interval is given by N_i. Since events do not happen simultaneously,

$$P(N_i \geq 2) = o(t/n), t \to 0$$

This is the equivalent of saying that $N_i \sim Bernoulli(p)$, where $p = \lambda(t/n) + o(t/n)$.

The total number of events in the interval is then Binomially distributed

$$N(t+s) - N(s) \sim Binom(n, p) = Binom(n, \lambda t/n)$$

So, as n approaches infinity, np approaches λt, and $N(t+s) - N(s)$ $Pois(\lambda t)$.

9.3.3 Inter-arrival times

Waiting times between events (inter-arrival times) in a poisson process are exponentially distributed. To show this, we consider the waiting times between the first and second arrival T_1 and T_2 respectively.

$$\begin{aligned} P(T_1 > t) &= P(N(t) = 0) \\ &= \frac{(\lambda t)^0 e^{-\lambda t}}{0!} = e^{-\lambda t} \end{aligned}$$

So the waiting time until the first arrival is exponentially distributed. For the second arrival event, given the first arrival occurred at time s, the waiting time is

$$\begin{aligned} P(T_2 > t) &= \int_s P(T_2 > t | T_1 = s) P(T_1 = s) \\ &= \int_s P(0 \text{ events in } (s, s+t] | T_1 = s) P(T_1 = s) \end{aligned}$$

By independence,

$$\begin{aligned} &= \int_s P(0 \text{ events in } (s, s+t]) P(T_1 = s) \\ &= e^{-\lambda t} \int_s P(T_1 = s) \\ &= e^{-\lambda t} \end{aligned}$$

By the same logic, the inter-arrival time (waiting time in between events) for any ith interval is also exponentially distributed $T_i \sim exp(\lambda)$. Then, the waiting time until the nth arrival is the sum of all i events $S_n = \sum_{i=1}^{n} T_i$. Since the inter-arrival times are exponentially distributed, the nth arrival time is gamma distributed $S_n \sim Gamma(\lambda, n)$
If we know the number of events in some given interval $(0, t]$, then the distribution of inter-arrival times conditioned on the number of events is uniformly distributed on $(0, t]$. We show this for the simple case with one event.

$$\begin{aligned} P(T_1 > s | N(t) = 1) &= \frac{P(T_1 > s, N(t) = 1)}{P(N(t) = 1)} \\ &= P(0 \text{ events in } (0, s], 1 \text{ event in } (s, t]) \\ &= \frac{e^{-\lambda s}\lambda(t-s)e^{-\lambda(t-s)}}{\lambda t e^{-t}} \\ &= \frac{t-s}{t} \end{aligned}$$

To get to the uniform distribution,

$$\begin{aligned} P(T_1 \leq s | N(t) = 1) &= 1 - P(T_1 > s | N(t) = 1) \\ &= \frac{s}{t} \end{aligned}$$

9.3.4 Splitting and superposition

In order to develop point processes suitable for certain models, one can employ mathematical operations that remove points from Poisson process (splitting i.e. thinning), or combining points from multiple Poisson processes (superposition).

Superposition

If we combine the points from two Poisson processes with rates r_1 and r_2, the result is a Poisson process with rate $r_1 + r_2$. More formally, suppose we have two Poisson processes of rate λ_1 and λ_2), given by $\{N_1(t); t > 0\}$ and $\{N_2(t); t > 0\}$. The two are independent if for all $t_1, ..., t_n$, the random variables $N_1(t_1), ..., N_1(t_n)$ are independent of $N_2(t_1), ..., N_1(t_n)$. Suppose $N(t) = N_1(t) + N_2(t)$ for all $t > 0$. Then $\{N_t; t > 0\}$then a Poisson process with rate $\lambda_1 + \lambda_2$.

Splitting

Now suppose we have a Poisson process $\{N_t; t > 0\}$ with rate λ. Suppose each arrival is switched to $\{N_1(t); t > 0\}$ with probability p and switched to $\{N_2(t); t > 0\}$ with probability $(1-p)$. Then $\{N_1(t); t > 0\}$ is a Poisson process with rate λp and $\{N_2(t); t > 0\}$ is a Poisson process with rate $\lambda(1 - p)$

9.3.5 Compound Poisson processes

Compound Poisson processes assign a random value or weight to each point in a Poisson process.

9.3.6 Non-homogenous Poisson processes

So far, we have considered Poisson processes where the rate parameter λ is constant for all time $t > 0$. A nonhomogenous Poisson process is one where the 'rate,' referred to as an intensity function $\lambda(t)$ can vary over time. $\{N_t; t > 0\}$ is a nonhomogenous or inhomogenous Poisson process if
1. $N(0) = 0$
2. $\{N(t), t \geq 0\}$ has independent increments
3. $P(N(t+h) - N(t) \geq 2) = o(h)$
4.$P(N(t+h) - n(t) = 1) = \lambda(t)h + o(h)$
Note that a nonhomogenous Poisson process with $\lambda(t) = \lambda$ for all $t > 0$ is a homogenous Poisson process.
Also note that for the nonhomogenous Poisson process, the interarrival times are no longer exponentially distributed, nor are they independent.

9.3.7 Compound or mixed 'Poisson' processes

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 10 — February 4

Lecturer: John Novembre *Scribe: Chris Stamper*

Note: These lecture notes are still rough, and have only have been mildly proofread.

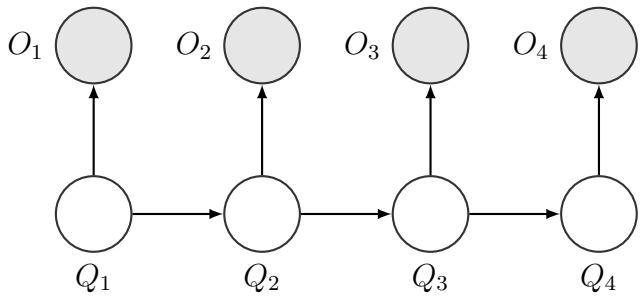
10.1 Hidden Markov Models

10.1.1 Hidden Markov Models Introduction

Hidden Markov Models have a variety of applications and are widley used in many disparate fields. One common application is their use in speech recognition. (Please see Lawrence Rabiner's 'A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition')

10.1.2 Uses of Hidden Markov Models

Hidden Markov Models are perfect for an underlying Markov Model with noisy data.



Meaning that if one conditioned on Q_3 the observation would only depend on it and nothing else.

10.1.3 Basic Structure of Hidden Markov Models

The basic structure of a Hidden Markov Model is:

- A Transition Probability Matrix: $(A = \{a_{ij}, i = 1, ...N, j = 1...N\})$ with

$$a_i j = P(q_t + 1 = S_j | q_t = S_i), \quad 1 \leq i, j \leq N, 1 \leq t \leq T - 1. \tag{10.1}$$

- An Emission Probability Matrix: B_{NXM} is a matrix with all $b_j(i)$

$$P(O_t = i | Q_t = j) = b_j(i), \quad 1 \leq j \leq N, 1 \leq i \leq M, 1 \leq t \leq T. \tag{10.2}$$

- An Initial State Distribution: $\pi = \{\pi_i, i = 1..., N\}$ or with

$$\pi_i = P(q_1 = S_i), \quad 1 \leq i \leq N. \tag{10.3}$$

Which means there are three main problems for Hidden Markov Models.

1. $P(O|\lambda)$: How do we compute efficiently?
2. Given O and λ, what is the most probable sequence, Q.
3. How can we estimate λ? AKA How can we find a $rp_{max}\lambda P(O|\lambda)$?

10.2 Algorithms for solving the problmes of Hidden Markov Models

10.2.1 Forward Algorithm

For problem 1 of HMMs we need a way to solve

$$P(O|\lambda) = \sum_Q P(O, Q|\lambda)$$

Which we can accomplish by using the forward algorithm.

- Initialize

$$\alpha_1(i) = \pi_i b_i(O_1) = P(O, Q_1) \tag{10.4}$$

- Induction

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i)a_{ij}]b_j(O_{t+1}), \quad 1 \leq t \leq T-1 \ \textit{and} \ 1 \leq j \leq N \tag{10.5}$$

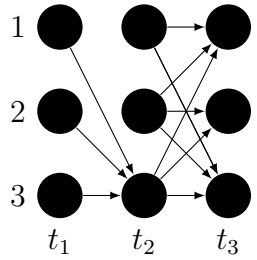
- Termination

$$P(O|\lambda) = \sum_j \alpha_T(j) \tag{10.6}$$

10.2.2 Viterbi Algorithm

For Problem 2 we need a way for the argmax of $P(O, Q|\lambda)$. In other words we need a way to consider all possible Q and find the Q that is maximal.

To visualize this we can imagine a lattice graph:



In which we need to take the maximum of the three paths coming from some i at time 2 and going to some j at time 3.

To do this we can make use of the Viterbi Algortithm.
In which we have:

- An array defined as: $\psi_{t+1}j = argmax_i[\delta_t(i)a_{ij}], \quad 1 \leq j \leq N, 1 \leq t \leq T-1$
- An auxillary variable, δ: $\delta_t(i) = max_{q_1 q_2 ... q_{t-1}} P\{q_1, q_2, ...q_{t-1} = i, O_1, O_2, ..., O_{t-1}|\lambda\}$

Which we can use in the algortihm.

- Initialize

$$\begin{aligned} \delta_1(i) &\leftarrow \pi_i b_i(O_1), \quad 1 \leq i \leq N \\ \psi_1(i) &\leftarrow 0, \quad 1 \leq i \leq N \\ t &\leftarrow 1 \end{aligned} \tag{10.7}$$

- Repeat

$$\begin{aligned} \psi_{t+1}(j) &\leftarrow argmax_i[\delta_t(i)a_(ij)], \quad 1 \leq j \leq N \\ \delta_{t+1}(j) &\leftarrow \delta_t(\psi_{t+1}(j))a_{\psi_{t+1}(j),j}b_j(O_{t+1}), \quad 1 \leq j \leq N \\ t &\leftarrow t+1 \end{aligned} \tag{10.8}$$

- Until

$$\begin{aligned} t &= T \\ P^* &\leftarrow max_{1 \leq i \leq N}[\delta_T(i)] \\ i_T^* &\leftarrow argmax_{1 \leq i \leq N}[\delta_T(i)] \\ &\text{State Sequence Backtracking:} \\ q_T^* &\leftarrow S_{i_T^*} \\ t &\leftarrow T \end{aligned} \tag{10.9}$$

- Repeat

$$\begin{aligned} i_{t-1}^* &\leftarrow \psi_t(i_t^*) \\ q_{t-1}^* &\leftarrow S_{i_{t-1}^*} \\ t &\leftarrow t-1 \\ &\text{until} \\ t &= 1 \\ Q^* &\leftarrow q_1^*, ...q_T^* \end{aligned} \tag{10.10}$$

10.2.3 Forward-Backward Algorithm

To solve the final problem, efficiently learning the parameters of a HMM, we can make use of a backward recursive procedure.
Our goal is that we want the backward variable, $B_t(i)$, or:
$B_t(i) = P(O_{t+1}, ..., O_T|Q_t = j, \lambda)$

such that

$\alpha_t(j)B_t(j) = P(O_1, ..., O_T|Q_t = j, \lambda) \propto P(Q_t = j|O, \lambda)$
and

$P(Q_t = j|O, \lambda) = \frac{\alpha_t(j)B_t(j)}{\sum_i \alpha_t(i)B_t(i)} = \lambda_t(j)$

To compute this we can use the Backwards Algorithm, AKA the Forward-Backward Algorithm:

- Initialize

$$B_T(i) = 1 \tag{10.11}$$

- Induction

$$B_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_t) B_t(j), \quad t = T-1, T-2, ..., 1 \text{ and } 1 \leq i \leq N \tag{10.12}$$

- Continue unitl

$$P(Q_t = j|O, \lambda) = \frac{\alpha_t(j) B_t(j)}{\sum_i \alpha_t(i) B_t(i)} = \lambda_t(j) \tag{10.13}$$

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 11 — February 9

Lecturer: John Novembre *Scribe: Nicholas Knoblauch*

Note: These lecture notes are still rough, and have only have been mildly proofread.

11.1 Continuous Time Markov Chains

11.1.1 Differential equations that lead to the poisson distribution

We can describe the probability that the number of events that have occured by time t as N_t, and can write the probability as: $P(N_t = i) = P_i(t)$
With the rate parameter λ, the probability that $N_t = i$ at $t + h$ is given by:
$P_i(t + h) = P_{i-1}(t)(\lambda h + o(h)) + P_i(t)(1 - \lambda h + o(h))$
Here we are summing over the probability of two scenarios. The first represents that $N_t = i - 1$ and that there was another event in time h. The second is that $N_t = i$ and that there was no event in time h. For small vaues of h we can ignore the probabilty of two steps in time h.
The limit as h goes to 0 is: $\frac{P_i(t+h)-P_i(t)}{h} = \frac{d}{dt}P_i(t) = P_{i-1}(t)(\lambda h + o(h)) + P_i(t)(1 - \lambda h + o(h)) - P_i(t)$
Cancel the one and divide by h to get:
$(\lambda)P_{i-1}(t) - P_i(t)P_0(t + h) = P_0(t)(1 - \lambda h + o(h))$
$\frac{dP_0(t)}{dt} = -\lambda P_0(t)$
The solution to this differential equation is: $P_0(t) = e^{-\lambda t}$ plus some constant
$\frac{P_i(t)}{dt} = \lambda P_0(t) - \lambda P_i(t) = \lambda e^{-\lambda t} - \lambda P_1(t)$
To get the solution for P_1 we use an integrating factor
$\frac{dP_1(t)}{dt} + \lambda P_1(t) = \lambda e^{-\lambda t}$
$\int_0^T e^{-\lambda t}\frac{dP_1(t)}{dt} + \lambda e^{-\lambda t}P_1(t) = \int_0^T \lambda$
Remembering the product rule: $(fg)' = f'g + g'f$

$$\int_0^T \frac{d}{dt} = \int_0^T \frac{d}{dt}(e^{\lambda t}P_1(t)) = \int_0^T \lambda dt = P_1(t) = \lambda t e^{-\lambda t}$$

We can keep iterating: $P_2(t) = \frac{(\lambda t)^2}{2}e^{-\lambda t}$ And eventually we have $P_n(t) = \frac{(\lambda t)^n}{n!}e^{-\lambda t}$
This is the poisson process

11.1.2 A More General Process, the Pure Birth Process

More general than the poisson process is a "birth process", or "pure birth process"
In a pure birth process there is a state dependent rate of arrival:
$\frac{d}{dt}P_i(t) = \lambda_{i-1}P_{i-1}(t) - \lambda_i P_i(t)$
The poisson process is a special case where λ_i is constant.
Another example is the **linear birth process**, where the $\lambda_i = i\lambda$ (This is also known as a Yule process)
$P_j(t) = \binom{j-1}{k-1}e^{-\lambda t}(1-e^{\lambda t})^{k-i}$ where k is the starting size. This is a negative binomial distribution (and is a transient solution)
Another major class is a birth-death process. We have a set of birth rates and a set of death rates. (λ from 0 to infinity, and μ from 1 to infinity)
$\frac{d}{dt}P_i(t) = \lambda_{i-1}P_{i-1}(t) + \mu_{i-1}P_{i+1}(t) - (\lambda_i + \mu_i)P(i)(t)$
There is a linear birth-death process $\lambda_i = i\lambda$ and $\mu_i = i\mu$
There is also linear birth-death with immigration
$\lambda_i = i\lambda + \theta$ $\mu_i = i\mu$

11.1.3 Continuous version of the Markov property

Independence of the past before the immediate past $P(X(t+s) = j|X(s) = i, X_u = x(u), 0 \leq u < s) = P(X(t+s) = j|X(s) = i)$

11.1.4 Rate matrix Q

We have been thinking about going from 0 to 1, but we can think in general about going from i to j Defining the generator matrix or Rate Matrix Q $P_{i,j}(h) = q_{ij}h + o(h)$ $P_{i,i}(h) = 1 - v_i h + o(h)$ Where $v_i = \sum_j q_{ij}$
If we define $q_{ii} = -v_i$ Then $Q = q_{ij}$
Inter-event times depend on i and they are exponential with rate v_i $P_{ij} = \frac{q_{ij}}{v_i}$ This looks like a discrete markov chain. There is an idea that we have a discrete markov chain embedded in a continuous markov chain

Q matrix for poisson

For the poisson process, Q is infinite in both directions. The diagonal has $-\lambda$ along the diagonal. One to the right of the diagonal is λ, the rate at which we arrive at the next state. (There is no return to previous states in the poisson process)

Q matrix for the birth-death process

Matrix goes on infinitely in both directions. Along the diagonal we have $\lambda 0$ for 0,0 followed by $-(\mu_k + \lambda_k)$ at k, k at $k, k-1$ we have μ_k and at $k, k+1$ we have λ_k For P, we have $\frac{\mu_k}{\lambda_k+\mu_k}$, at $k, k-1$. At $k, k+1$ we have $\frac{\lambda_k}{\lambda_k+\mu_k}$, and at the diagonal we have 1 minus the two off diagonals
$\frac{P_{ij}(t)}{dt} = v_j Pij(t) + \sum_{k \neq j} P_{ik}(t) q_{kj}$ This is the probability of going from i to j, plus the probability of going form i to k and then going to j
$\frac{dP_t}{dt} = P_t Q$ $P_t = e^{Qt}$
If Q is diagonalizable, then we can write $Qt = UDU^{-1}$ where D is a diagonal matrix, then $e^{Qt} = Ue^D U^{-1}$
There are a branch of methods called Krylov methods for exponentiating matrices.

11.1.5 What about stationary distributions?

The Global Balance Equations

Define P_i as the stationary probability of being in state i which is the limit as t goes to infinity of $P_{ij}(t)$ $v_j P_j = \sum_k q_{kj} P_k$ This equation describes a situation where the rate out of state j (weighted by the probability of being in state j, or the flux) is equal to the flux into state j
If a Continuous Time Markov Chain is time reversible, then $P_i q_{ij} = P_j q_{ji}$ and it satisfies the local balance equations. The flux from i to j is equal to the flux from j to i
What is the flux out for state 0?

State	Flux out	Flux in
0	$\lambda_0 P_0$	$\mu_1 P_1$
1	$(\lambda_1 + \mu_1) P_1$	$\lambda_0 P_0 + \mu_2 P_2$
2	$(\lambda_2 + \mu_2) P_2$	$\lambda_1 P_1 + \mu_3 P_3$
n	$(\lambda_n + \mu_n) P_n$	$\lambda_{n-1} + \mu_{n+1} P_{n+1}$

These all have to be equal
$\lambda_0 P_0 = \mu_1 P_1$ $\lambda_1 P_1 = \mu_2 P_2$ $\lambda_n P_n = \mu_{n+1} P_{n+1}$
We can start solving everything in terms of P_0
$P_1 = \frac{\lambda_0 P_0}{\mu_1}$ $P_3 = \frac{\lambda_2}{\mu_3} P_2$ $P_n = \frac{\lambda_{n-1} \ldots \lambda_1}{\mu_n \ldots \mu_1} P_0$
We know that $1 = P_0 + \sum_{n=1}^{\infty} P_n P_0$
in the linear birth-death model $P_n = \frac{\lambda}{\mu}^n (\frac{1}{1+\sum_{i=1} \infty \frac{\lambda}{\mu}^n}$
Even though this is an infinte sum, it turns out that: $P_n = \frac{\lambda}{\mu}^n (1 - \frac{\lambda}{\mu})$
We know this because $\sum_{n=1}^{\infty} p(1-p)^{n-1} = 1$

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 12 — February 11

Lecturer: Matthew Stephens *Scribe: Katherine Silliman*

Note: These lecture notes are still rough, and have only have been mildly proofread.

Computing practical on implementing MCMC in R. See *github.com/stephens999/mcmc-examples/blob/master/MCMC/IntroMCMC.R* for original code example.

12.1 Ex.1: Sampling from an exponential distribution using MCMC

Any MCMC scheme aims to produce (dependent) samples from a "target" distribution $\pi(x)$. In this case we are going to use the exponential distribution with mean 1 as our target distribution. So we start by defining our target density:

```
target = function(x){
    if(x<0){
        return(0)}
    else {
        return( exp(-x))}
}
```

Recall that the Metropolis-Hastings algorithm is useful for sampling from a distribution that is proportional to our target distribution. It involves the following steps:

1. Initialization: pick an initial state x (usually at random)
2. Randomly draw a state x' from the proposal distribution $Q(x \to x')$
3. Accept the state according to the acceptance distribution H:

$$H = min\left(1, \frac{\pi(x')\,Q(x' \to x)}{\pi(x)\,Q(x \to x')}\right). \tag{12.1}$$

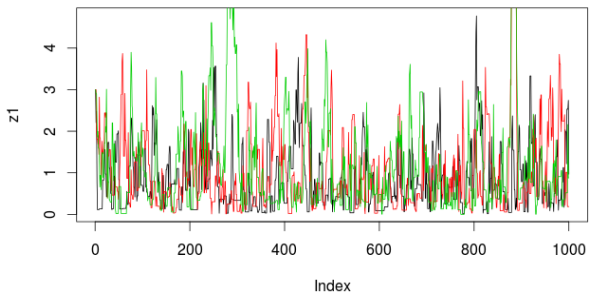
   If not accepted state remains at x, otherwise transitions to x'
4. Save state x, go to #2
5. Repeat desired number of iterations

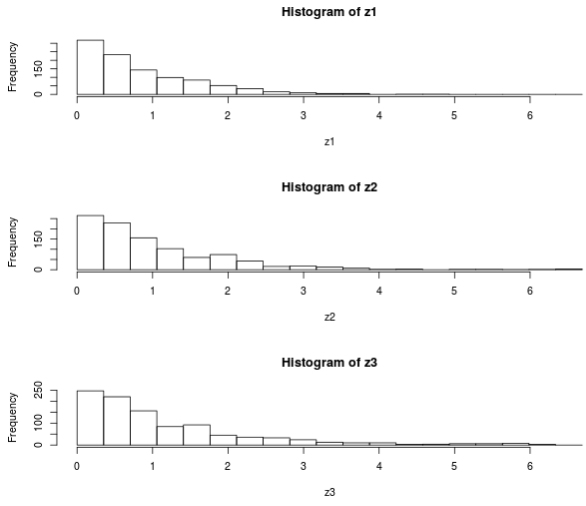
We can code a Metropolis-Hastings scheme in R like this:

```
easyMCMC = function(niter, startval, proposalsd){
    x = rep(0,niter)
    x[1] = startval
    for(i in 2:niter){
        currentx = x[i-1]
        proposedx = rnorm(1,mean=currentx,sd=proposalsd)
        A = target(proposedx)/target(currentx)
        if(runif(1)<A){
            x[i] = proposedx     # accept move with probabily min(1,A)}
        else {
        x[i] = currentx      # otherwise "reject" move, and stay where we are} }
    return(x) }
```

$*$ Note that since $\frac{Q(x' \to x)}{Q(x \to x')} = 1$, we are only computing $\frac{\pi(x')}{\pi(x)}$ to determine acceptance. Plotting x shows us the **trace** of our Markov chain. When x is high dimensional it is more difficult or impossible to view a trace of x.

```
#Run MCMC 3 times and look at how similar the results are.
z1=easyMCMC(1000,3,1)
z2=easyMCMC(1000,3,1)
z3=easyMCMC(1000,3,1)
plot(z1,type="l")
lines(z2,col=2)
lines(z3,col=3)
```





By playing around with the proposal standard deviation, number of iterations, and starting value we can see how these affect the MCMC output.

Starting value: If the starting value is way outside of the target distribution, it will take longer to converge. The code also will not run with a negative starting value.

Proposal SD: A very small proposal SD will make it take longer to converge, whereas a large SD will result in the chain getting "stuck" a lot as values will more frequently get rejected. Intuitively, an SD of 0 will result in a flat trace as x never changes.

Number of iterations: An adequate nuber of iterations are required in order for the chain to converge.

"sticky chain": high autocorrelation among steps
We can also change the target distribution.

```
target = function(x){
    return((x>0 & x <1) + (x>2 & x<3))
}
```

This target will have a bimodal distribution. If the SD is too small (i.e. 0.1) it will not work.

12.1.1 Tuning the MCMC

Tuning your proposal involves finding the best values of the proposal standard deviation in order for the chain to mix faster. This is usually accomplished by ruunning a test chain and looking at the proportion of steps that are accepted. If the proportion of accepted steps is too high, it may slow down mixing. In general, when you have higher dimensions in your target values the optimal optimal acceptance rate is lower.
adaptivity: where you change the rules (i.e. the proposal SD) for an ongoing MCMC chain based on what you see in the trace. This is risky as it may end up not being a true Markov chain. Another important consideration is that the probability of storing an observation must be unbiased and not depend on what stage the chain is currently in. A common way to *thin* values is to set a consistent interval for recorded observations (i.e. every 10 iterations).

12.2 Ex.3: Estimating an allele frequency and inbreeding coefficient (Gibbs Sampler)

In class we glossed over Example 2 in the IntroMCMC.R, which goes over how to implement an MCMC to estimate the allele frequency in a population showing Hardy-Weinberg equilibrium. Example 3 illustrates how to implement a 2 dimensional Markov chain to estimate an

allele frequency and inbreeding coefficient. From IntroMCMC.R: " A slightly more complex alternative than HWE is to assume that there is a tendency for people to mate with others who are slightly more closely-related than "random" (as might happen in a geographically-structured population, for example). This will result in an excess of homozygotes compared with HWE. A simple way to capture this is to introduce an extra parameter, the "inbreeding coefficient" f, and assume that the genotypes AA, Aa and aa have frequencies $fp+(1-f)\,p^2$, $(1-f)\,2p\,(1-p)$, and $f\,(1-p)+(1-f)\,(1-p)^2$ In most cases it would be natural to treat f as a feature of the population, and therefore assume f is constant across loci. For simplicity we will consider just a single locus.

Note that both f and p are constrained to lie between 0 and 1 (inclusive). A simple prior for each of these two parameters is to assume that they are independent, uniform on [0,1]. Suppose that we sample n individuals, and observe nAA with genotype AA, nAa with genotype Aa and naa with genotype aa.

One way we can implement and MCMC routine to get f and p is to update both f and p each iteration and assume that they are independent. Another way is to implement a Gibbs Sampler. To do this, we use a "latent variable" Z_i, a representation of whether an individual came from an inbred mating f or non-inbred mating $(1-f)$. This way we can implement a posterior dstribution for p that does not depend on f.

$$Z_i = \begin{cases} 1, & wpf \\ 0, & wp\,(1-f) \end{cases} \tag{12.2}$$

$Z_i \sim Bernoulli\,(f)$

This makes the likelihoods of the data(genotypes) given p and Z not depend on f:

$$p\,(AA|z_i=1) = p \tag{12.3}$$
$$p\,(AA|z_i=0) = p^2 \tag{12.4}$$
$$p\,(Aa|z_i=1) = 0 \tag{12.5}$$
$$p\,(Aa|z_i=0) = 2p\,(1-p) \tag{12.6}$$
$$p\,(aa|z_i=1) = 1-p \tag{12.7}$$
$$p\,(aa|z_i=0) = (1-p)^2 \tag{12.8}$$

To implement the Gibbs sampler, you would iterate over the following steps:

1. Sample Z from $p\,(z|g,f,p)$
2. Sample f,p from $p\,(f,p|g,z)$

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 14 — February 18

Lecturer: Matthew Stephens *Scribe: Lauren E. Blake*

Note: These lecture notes are still rough, and have only have been mildly proofread.

14.1 Class notes

1. There are two vignettes on the multivariate normal on Matthew's website:
   https://github.com/stephens999/fiveMinuteStats/tree/gh-pages/analysis
2. The reading for this lecture can be found in Chapter 10 of the Ross textbook.

14.2 Background

If X_1 and X_2 are independent normals (~ N(0, 1)), then aX_1 + bX_2 ~ N(0, $a^2 + b^2$)

14.3 Introduction to the multivariate normal

- There are many ways to definite the multivariate normal, but the definition from Wikipedia states that Vector X = (X_1, ... X_r) is r-variate normal or "r-dimensional multivariate normal" if every alternate linear combination of X, $\lambda_1 X_1 + \ldots + \lambda_r X_r$, is univariate normal (as long as $\lambda \neq 0$).
- Suppose Z = (Z_1, ... Z_n) are iid ~ N(0,1). Let A by an r x n matrix and μ be an r-vector. Then X = μ + AZ is a multivariate normal with mean E(X_j) = μ_j and covariance matrix Cov(X_i, X_j) = (A A′)$_{ij}$.
- For example, Cov(X_1, X_1) = Var(X_1) so the diagonal of the covariance matrix is one.

14.4 An example

There is an example in the vignette, Z_1, Z_2, Z_3 where

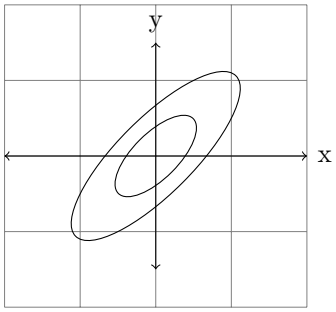
$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

and where $X_1 = Z_1$ and Z_2 and $X_2 = Z_1 + Z_3$.

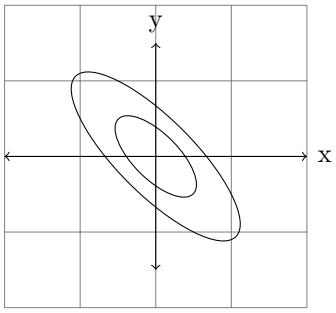
From this, we get X = AZ $\sim$ N(0, Σ = A A$'$)

- Note that Σ = A A$'$ =
$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$
- The covariance comes from the fact that they share 2 elements (Z_1) and don't share two elements (Z_2 and Z_3).
- Corr =
$$\frac{Cov}{\sqrt{(Var)}\sqrt{(Var)}}$$
=
$$\frac{1}{2}$$
=
$$\frac{Cov}{\sqrt{(2)}\sqrt{(2)}}$$
.

  This gives us a 3D ellipse that is tilted towards the right:



- If we were to change the covariance matrix so A A$'$ =
$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$
then we would have an 3D ellipse with the same shape but tilted to the left:



14.5 Simulating from the multivariate

- We can use the fact that a multivariate normal are linear combinations of univariate normals
- For $N_r(\mu, \Sigma_r)$, you can simulate from this provided that you can find A such that A $A' = \Sigma$.
- Note: Σ must be symmetric to hold. This means that Σ must be positive and semi-definite (so it has all non-negative eigenvectors).
- Inverting matrices can be unstable, so we need at least one other way to find the inverse of a matrix. In the Cholesky decomposition of Σ in the lower triangular matrix L has a value such that L $L' = \Sigma$. This allows use to find L and the from that find Z and then find X. We can then add μ if we are so inclined.
- To find the inverse of a matrix in R, use the command *chol2 inv (chol(Σ))*.
- It is important to note that we use this because inverting matrices can be unstable.
- We don't always have a density but if it is invertable (has all positive eigenvalues), then the density of the multivariate normal is the following:

$$p(x) = (\frac{1}{\sqrt{(2\pi)|\Sigma|}})^{\frac{r}{2}} exp(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)) \tag{14.1}$$

14.5.1 Maximizing p(x) for the multivariate normal

- It is not straightforward to show how to maximize p(x).
- If $X_1 \ldots X_n$ (that are iid) $\sim N_r(\mu, \Sigma)$, then the MLE for μ and Σ is $\hat{\mu}_{ij} = \frac{1}{n} \sum_i x_{ij}$. This is the average of vectors.
- Here, we are going to average j matrices: $\hat{\Sigma} = \frac{1}{n} \sum_i (\underline{x_i} - \underline{\hat{\mu}})(\underline{x_i} - \underline{\hat{\mu}})'$
- Therefore, $\hat{\Sigma} = \frac{1}{n} \sum_i (x_{ij} - \hat{\mu}_j)(\underline{x_{ik}} - \hat{\mu}_k)$
- The likelihood:

$$L(\mu, \Sigma) = \prod_i (\frac{1}{\sqrt{(2\pi)|\Sigma|}})^{\frac{r}{2}} exp(-\frac{1}{2}(x_i - \mu)'\Sigma^{-1}(x_i - \mu)) \tag{14.2}$$

14.6 Introduction to Gaussian Processes and Brownian Models

- If X_1, X_2 ... is a Markov chain, then $X_{t+1} \mid X_t \sim$ N(X_t, 1). It is a normal Markov chain where be begin at the origin (0,0) and $X_1 \sim$ N(0,1). As a result, we might call the a random walk, where each time we go an amount from a normal distribution.
- If we were to stop at X = 1000, then it would be a 1000-D multivariate normal where

  $X_1 = Z_1$
  $X_2 = X_2 + Z_1 = Z_1 + Z_2$
  $X_3 = X_2 + Z_3 = Z_1 + Z_2 + Z_3$

  so $\underline{X} = AZ$.
- Note: With a multivariate normal, they are uncorrelated if and only if they are independent.
- In the example $\underline{X} = AZ$, the points closer together are going to appear more correlated than points that are farther away.

- $\Sigma = \text{A A}' =$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

- But this is really interesting what you take the inverse $\Omega = \Sigma^{-1} =$

$$\begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

  where Ω is the precision matrix.

- The structure of the inverse of the covariance matrix can indicate that this is a Markov chain because they have the following special property:

  1. If X is multivariate normal, $\underline{\text{X}} \sim N_r(0, \Omega^{-1})$
  2. $\Omega^{ij} = 0$ if and only if X_i and X_j are conditionally independent given all other Xs
  3. $\Sigma_{ij} = 0$ if and only if X_i and X_j are independent

14.7 (Undirected) Gaussian graphical models

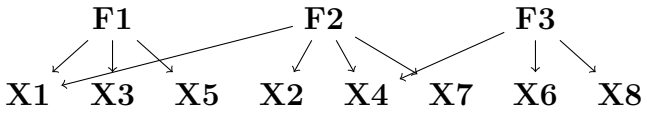
14.7.1 Introduction

- For any precision matrix, you want to make a connection if there is a non-zero covariance. This has many applications, including a gene network. For any precision matrix, you would make a connection between genes that have a non-zero covariance.

- The advantage of Gaussian graphical models is that you decrease the number of parameters that you are trying to estimate

- In the gene network example, you may have hub genes or master regulators that control a bunch of other genes. The signal from the controlled genes are observed, but the signal from the hub genes are not observed. Given that these are unobserved, there are no conditional independents. For example, the precision matrix for X_1, X_2, ... (observed signals) is dense but the precision matrix with F_1 ... (from the unobserved signals)

and X_1 ... (from the observed signals) is sparse. Removing an unobserved value won't drastically change a covariance matrix (it will change at only 1 row), but it will change the precision matrix everywhere.



- It is appealing for genetics and other applications, however, we are plagued by noisy data (e.g. expression data from RNA-seq). Since the true value of the measurement is unknown, we have to treat everything as unobserved, which is problematic.

14.7.2 Utilization

- Assume the observations have come to a relatively small number of factors so that the relationships between the observations are relatively simple.

$$\mathbf{X} = LF + E \tag{14.3}$$

where L represents the loadings, F represents the factors, and E represents the error terms. **X** has dimensions p x n. L has dimensions p x k and F has dimensions k x n where k is a small number of factors. E has dimensions p x n.

- In a genetics application, F is individual transcription factors, where each TF has different effects across the observations (e.g. TF binding sites near genes).

- If we assume that k factors F are independent, $N_n(0, \text{I})$ and rows of E are independent $N_n(0, \psi)$, then $X_i \sim \text{N}(0, \text{L L}' + \psi)$. In words, X_i is represented by a multivariate normal with rank k matrix and diagonal, ψ.

- We are making 3 assumptions in order to decrease the number of parameters to be estimated:

  1. The covariance matrix is sparse.
  2. The inverse covariance matrix is sparse.
  3. The covariance matrix is rank k normal with a diagonal. This is also known as factor models

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 15 — February 23

Lecturer: Matthew Stephens *Scribe: Nan Xiao*

Note: These lecture notes are still rough, and have only have been mildly proofread.

This lecture discussed more properties and applications of multivariate normal distributions, and introduced Gaussian processes (Brownian motion) with examples.

15.1 Multivariate normal distributions

15.1.1 Sparse factor models

From last lecture, we know the MLE of multivariate normals (a high-dimensional covariance matrix estimation problem) gives us too many parameters to estimate, so we need some assumptions to reduce the number of parameters. Here we briefly review the three methods for such purposes:

1. Assumption of sparse covariance matrix
2. Assumption of sparse precision matrix (mostly appears in undirected Gaussian graphical models, which leads to the important concept of conditional independence)
3. Sparse factor models (low-rank factorization)

Sparse factor models:

$$\underbrace{Y}_{n\times p} = \underbrace{L}_{n\times k} \cdot \underbrace{F}_{k\times p} + \underbrace{E}_{n\times p} \tag{15.1}$$

This could be also named matrix factorization, since this model will factorize a matrix into two low-rank matrices. We assumed that

$$\begin{aligned} Y_{\cdot j} &\sim \mathcal{N}(0, LL' + \psi) && (15.2)\\ \text{if} \quad F_{\cdot j} &\sim \mathcal{N}(0, I) && (15.3)\\ E_{\cdot j} &\sim \mathcal{N}(0, \psi) && (15.4) \end{aligned}$$

Further more, there is an equivalence between two assumptions:

$$\Sigma \text{ (covariance matrix) is low rank and diagonal} \Leftrightarrow \Sigma^{-1} \text{ low rank and diagonal.} \tag{15.5}$$

15.1.2 Properties of multivariate normal distributions

Property 1. Linear Combination of MVNs

Sums and linear combinations of MVNs are still MVN.

If

$$\underbrace{X}_{r\times 1} \sim \mathcal{N}_r(\underbrace{\mu}_{r\times 1}, \underbrace{\Sigma}_{r\times r}) \tag{15.6}$$

then

$$\underbrace{A}_{p\times r}\underbrace{X}_{r\times 1} \sim \mathcal{N}_p(\underbrace{A\mu}_{p\times 1}, \underbrace{A\Sigma A^T}_{p\times p}) \tag{15.7}$$

Moreover,

$$\underbrace{A}_{p\times r}\underbrace{X}_{r\times 1} + \underbrace{b}_{p\times 1} \sim \mathcal{N}_p(\underbrace{A\mu + b}_{p\times 1}, \underbrace{A\Sigma A^T}_{p\times p}) \tag{15.8}$$

Imagine for a bivariate normal distribution, the term b will only shift the mean, it will not change the shape of the distribution (the covariance is kept the same).

Property 2. Conditional distributions

Given a MVN X, if we condition on a subset of X, say X', then $X|X'$ is still MVN, and only the mean is changed, the covariance will not change.

Also, if

$$\begin{pmatrix} \underbrace{X_1}_{r_1} \\ \underbrace{X_2}_{r_2} \end{pmatrix} \sim \mathcal{N}_{r_1+r_2}\left(\begin{pmatrix} \underbrace{\mu_1}_{r_1} \\ \underbrace{\mu_2}_{r_2} \end{pmatrix}, \begin{pmatrix} \underbrace{\Sigma_{11}}_{r_1} & \underbrace{\Sigma_{12}}_{r_2} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right)$$

then $X_1|X_2 = a$ is MVN. In fact,

$$X_1|X_2 \sim \mathcal{N}(\bar{\mu}, \bar{\Sigma}). \tag{15.9}$$

where

$$\bar{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2) \tag{15.10}$$

$$\bar{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{15.11}$$

See the Wikipedia page on conditional distributions of MVNs (`https://en.wikipedia.org/wiki/Multivariate_normal_distribution#Conditional_distributions`) for details.

A regression analog of the above equation is, if:

$$Y = X\beta + E \tag{15.12}$$

and we know this is actually

$$E(Y|X) = X\beta \tag{15.13}$$

then

$$\bar{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2) \tag{15.14}$$

So $\Sigma_{12}\Sigma_{22}^{-1}$ is also called *regression coefficients*.

15.1.3 Application of MVN in HMM

Recall the Hidden Markov Model we used in the previous lectures (Figure 15.1):

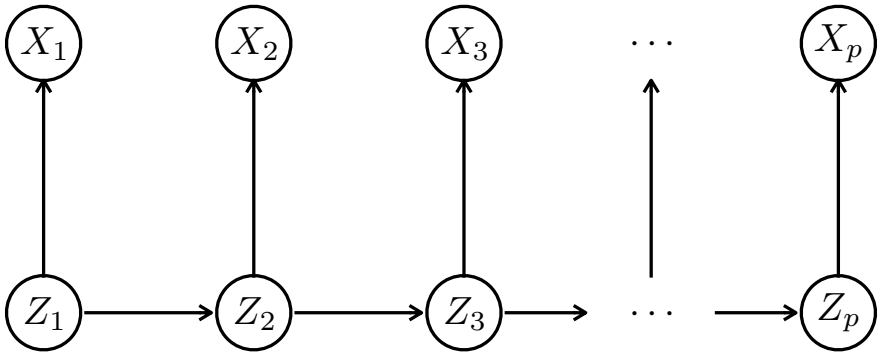


Figure 15.1. A Hidden Markov Model

Assume everything is MVN, suppose

$$Z_{t+1}|Z_t \sim \mathcal{N}_r(Z_t, \Sigma_1) \tag{15.15}$$

$$X_t|Z_t \sim \mathcal{N}_r(Z_t, \Sigma_2) \tag{15.16}$$

If Z_i and X_i are r-dimensional vectors, then $X_1, X_2, \ldots, X_p$ and $Z_1, Z_2, \ldots, Z_p$ are two $p \times r$ multivariate normals.

Equation 15.14 is useful in forwardbackward algorithms, since the algorithm involves multivariate normal calculations, and particularly *conditional distributions* of $X_1|X_2 = a$ is MVN. Additionally, Kalman filter is also related to this property.

15.1.4 Linear algebra tricks to compute $\bar{\mu}$

Trick 1. Avoid computing matrix inverse directly

$\bar{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(a - \mu_2)$. This formula involves computing the matrix inverse Σ_{22}^{-1} numerically. In matrix computations, we must avoid computing the inverse directly, instead, we should use matrix decomposition to do it.

For example, if $y = Ax$, we want to get x by solving $x = A^{-1}y$, but it turns out we don't need to solve A^{-1} directly. In reality, it is possible to use multiple types of matrix decompositions to get x, here we use Cholesky decomposition to do this.
Cholesky decomposition factorizes a matrix into the product of a lower triangular matrix and its conjugate transpose, namely:

$$A = LL' \tag{15.17}$$

then we have

$$y = LL'x \Rightarrow L'x = L^{-1}y \tag{15.18}$$

Since L^{-1} is lower triangular, we can get L^{-1} here easily by using the backsolve algorithm (the corresponding R function is `backsolve()`), then we get $L^{-1}y$, use backsolve again to get x easily, since L' is upper triangular.

Cholesky decomposition exists when A is a positive semi-definite (PSD) matrix, so it is very useful in factorizing covariance matrices, since covariance matrices are always PSD.

Trick 2. Avoid re-computing matrix inverses

If we already knew what A^{-1} is (e.g. already computed it somehow), then $(A+uv')^{-1}$ is easy to compute, by using the *Sherman–Morrison formula*, and not compute it directly. Here uv' represents rank-1 changes to A, but actually this works on any low-rank modifications of A. We can use *Sherman–Morrison formula* to compute the rank-1 updates of A:

$$(\underbrace{A}_{n\times n} + \underbrace{u}_{n\times 1}\underbrace{v'}_{1\times n})^{-1} = A^{-1} - \frac{A^{-1}uv'A^{-1}}{1 + v'A^{-1}u} \tag{15.19}$$

The *Sherman-Morrison-Woodbury formula* is an natural extension for the above formula, on rank-k modification of A, i.e.

$$(A + UV)^{-1} = A^{-1} - A^{-1}U(I + V^TA^{-1}U)^{-1}V^TA^{-1} \tag{15.20}$$

One example of this is factor models. In fact, $LL' = l_1l_1' + l_2l_2' + \cdots$, equivalent to changing k columns in L.

Another example is variable selection regressions. If we have the linear model

$$\underbrace{Y}_{n\times 1} = \underbrace{X}_{n\times p}\underbrace{\beta}_{p\times 1} + \underbrace{E}_{n\times p} \tag{15.21}$$

we can use MCMC to fit it, and we assume that β is sparse (only a few of the coefficients are non-zero). This will require changing one β_i from 0 to non-zero, which is a small modification to the original matrix.

Trick 3. Avoid computing ratios directly by division

In Metropolis–Hastings algorithm, we have to compute the ratio of two densities:

$$\frac{\pi(x')}{\pi(x)} \tag{15.22}$$

In this case, we should compute the logarithm of the two densities and then exponentiate the difference between them, instead of computing the division directly, i.e.

$$\exp(\log \pi(x') - \log \pi(x)) \tag{15.23}$$

since $\pi(x')$ and $\pi(x')$ are close to 0, the division between two near 0 numbers will be numerically unstable.

For multivariate normals, we should also do this:

$$\cdots \propto \frac{1}{|\Sigma^{p/2}|}\exp(\cdots) \tag{15.24}$$

When we compute likelihood ratios, we should also compute the two log-likelihoods, then do the exponentiation, rather than dividing two likelihood directly.

15.2 Brownian motion

Read Ross book Chapter 10 (Brownian Motion and Stationary Processes) for more details about this section.

Considering a symmetric random walk, which in each time unit we take a step either to the left or to the right. Suppose for each Δt time unit we take a step of size Δx either to the left or the right with equal probabilities. If $X(t)$ is the position at time t, then

$$X(t) = \Delta x(X_1 + \cdots + X_{[t/\Delta t]}) \tag{15.25}$$

where

$$X_i = \begin{cases} +1, & \text{if } i\text{-th step of length } \Delta x \text{ is to the right} \\ -1, & \text{if } i\text{-th step of length } \Delta x \text{ is to the left} \end{cases} \tag{15.26}$$

and $[t/\Delta t]$ represents the largest integer less than or equal to $t/\Delta t$, and X_i are independent,

$$P(X_i = 1) = P(X_i = -1) = \frac{1}{2}. \tag{15.27}$$

This is essentially a Markov chain with $P_{i,i+1} = P_{i,i-1} = 1/2, i = 0, \pm 1, \ldots$

Suppose that we take smaller and smaller steps in smaller and smaller time intervals. If we go to the limit in the right manner, what we obtain is defined as *Brownian motion*.

Take the limit, we get:

$$X(t) \sim \mathcal{N}(0, \sigma^2 t) \tag{15.28}$$

When $\sigma = 1$, the process is called *standard Brownian motion* (we can get a standard Brownian motion by standardizing any Brownian motion by $B(t) = X(t)/\sigma$), then

$$X(t) \sim \mathcal{N}(0, t) \tag{15.29}$$

Independent Increment Property:

$$X(t) - X(s) \sim \mathcal{N}(0, (t-s)), \quad \forall s < t. \tag{15.30}$$

Recall the normal Markov Chain simulations we did in the past lectures (`http://stephens999.github.io/fiveMinuteStats/analysis/normal_markov_chain.html`):

$$\begin{aligned} X_1 &\sim \mathcal{N}(0,1) \\ X_2 &= X_1 + \mathcal{N}(0,1) \\ X_3 &= X_2 + \mathcal{N}(0,1) \\ &\ldots \end{aligned}$$

Intuitively, this property implies such a continuous Markov chain with $\mathcal{N}(0, (t-s))$ embedded in it.

15.2.1 Application of Brownian motion in genetics and evolution

Figure 15.2 shows an example of Brownian motion on a species tree of certain trait (e.g. height). Assume we start from $t = 0$, and $z_i \sim \mathcal{N}(0, t_i)$. We can see that the bottom nodes of the tree are linear combinations of z_i.
Since $z_i \sim \mathcal{N}(0, t_i)$, we know that $\sum z_i$ are multivariate normals, with covariance matrix associated with the length of the changes.

Such evolution of traits can be also modelled using the *Ornstein–Uhlenbeck processes*, which introduces certain strength of attraction for each step, in addition to the simple Brownian motion model.

For more details of the model, search "Brownian motion model of trait evolution".

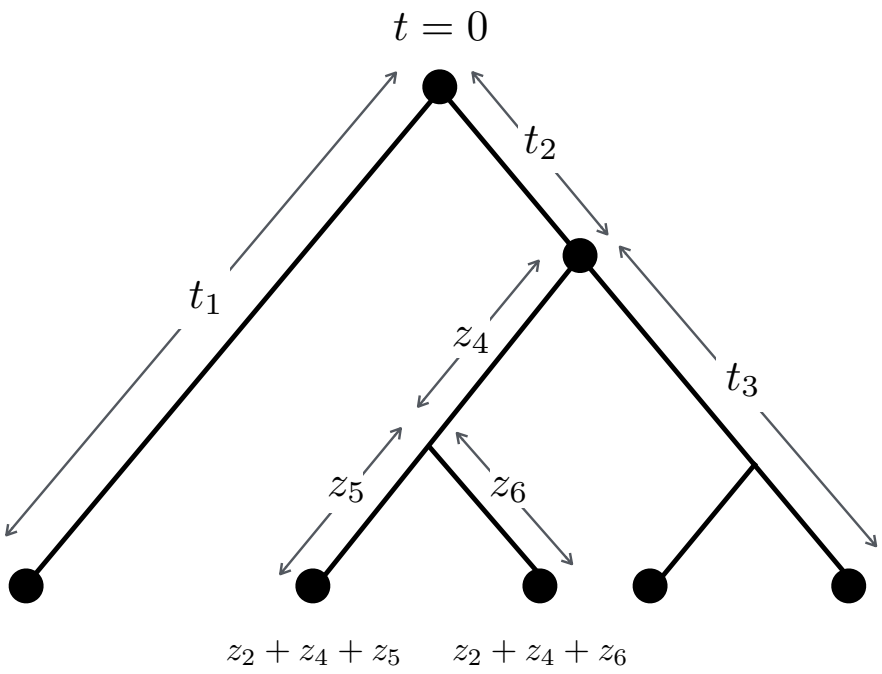


Figure 15.2. Brownian motion along an evolution tree

15.2.2 General definition of a Gaussian Process

Definition. A time-continuous stochastic process is a Gaussian Process $\iff$ For every set of indices in the index set T, $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ is multivariate Gaussian.

As a matter of fact, continuous time can always be discretized. Usually, the covariance matrix is proportional to the time length (or we can somehow know their definition of pairwise covariances); the mean is always 0. Therefore, a Gaussian Process is often defined by specifying $\mathrm{Cov}(X(t_1), X(t_2)), \forall\, t_1, t_2$.

We usually use K to denote this function:

$$K(t_1, t_2) = \mathrm{Cov}(X(t_1), X(t_2)). \tag{15.31}$$

If $K(t_1, t_2)$ only depends on $|t_1 - t_2|$, then it is said to be *stationary*.

This indicates that the nearer the two points are, the more correlated they will be. The farther the two points are, the less correlated they will be, as shown in Figure 15.3.

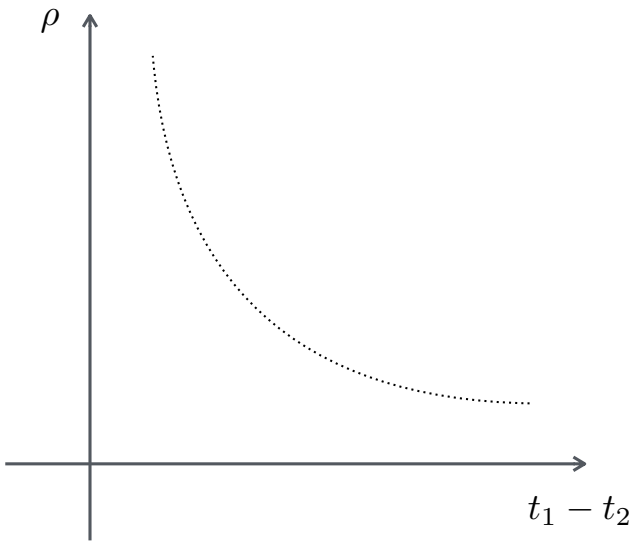


Figure 15.3. The relationship between time length and correlation of two points.

This implies the *smoothness* for the "local" time points, and the *scattered* pattern for the points in the "global" perspective.

In fact, most processes we use are stationary. Brownian motion is one exception.

HGEN 48600/STAT 35450: Fundamentals of Computational Biology

Lecture 17 — March 10

Lecturer: John Novembre *Scribe: Joseph Marcus*

Note: These lecture notes are still rough, and have only have been mildly proofread.

Introduction

Today's lecture introduced undirected probabilistic graphical models and briefly discussed diffusion processes.

Undirected Probabilistic Graphical Models

Review of Directed Graphs

Recall the canonical directed graph that we have used for numerous examples during lecture (*Figure 1*). We can take advantage of the factorization of directed probabilistic graphical models to efficiently compute the joint distribution of the random variables or various combinations of conditional probabilities. Let $X = \{X_1, X_2, X_3, X_4, X_5, X_6\}$ denote the set of random variables, i.e. nodes, in the graph. We can then write down the probability of X i.e. joint distribution of $X_1, X_2, \ldots, X_6$ as:

$$Pr(X) = \prod_i Pr(X_i \mid Parents(i))$$

Where $Parents(i)$ represents the conditioning on the parent nodes of child node i. We can use algorithms of the flavor of the Sum-Product algorithm to compute the joint distribution of the above random varibles through the use of message passing functions. For example, we covered Felsenstein's Pruning Algorithm for computing the likelihood of a tree. While the directed probabilistic graph is useful representation of conditional dependencies of random variables there are other graph-based approaches we can take. Specifically we can use an undirected graph structure to represent a probabilistic model.

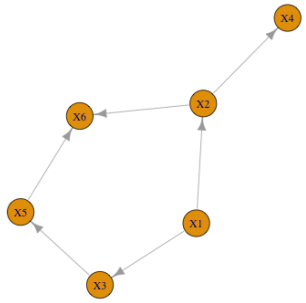


Figure 17.1. Directed probabilistic graphical model where each node represents a random variable and edges represent conditional dependencies

d-separation

To understand the use of undirected graphs for probabilistic modeling it is helpful to introduce the notion of *d-separation*. Two nodes X_1, and X_2 are d-separated if when conditioned on a third node they are independent. For example lets take another look at our canonical graph but zoomed in on the sub-graph $\{X_2, X_5, X_6\}$ (*Figure 2*). X_5 and X_2 are not d-separated because when conditioned on X_6 they are not independent.

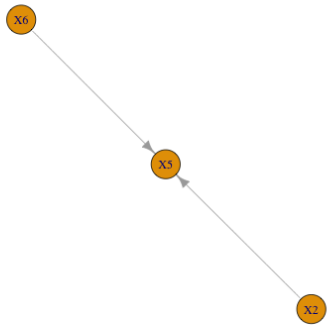


Figure 17.2. Sub-graph of the above example to illustrate d-separation

In a undirected graph there is a natural visual interpretation of d-separation. Particularly if every path between node X_A and node X_B contains X_C then X_A and X_B are d-separated $\equiv X_A \perp\!\!\!\perp X_B \mid X_C$. This can be easily visualized in a undirected graph by tracing the paths between two nodes. It is a helpful exercise to the reader to try to condition on various nodes in the Markov Random Fields below.

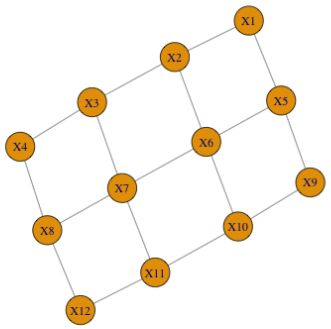


Figure 17.3. Markov Random Field Example

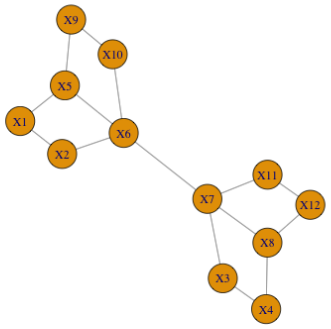


Figure 17.4. Markov Random Field Example for d-separation

Graph Moralization

Graph moralization is a term for converting a directed graph into a undirected graph. Every graph has a corresponding undirected graph. If we revisit figure 2 we can perform graph moralization by "marrying" the "un-married" parents and then removing directions on the edges. Specifically we add an edge between X_2 and X_5 and remove all the directions of edges. Another useful definition is the Markov blanket which is the set of neighbors of a particular node. The Markov blanket helps us consider what random variables to condition on in the Gibbs sampling algorithm i.e. $P(X_i \mid Neighbors(X_i))$. Some useful advantages immediately become clear when using undirected graphs for probabilistic models:
1. We can easily visualize conditional independencies
2. We no longer need d-separation we can just simply use graph separation
3. Makes clear the Markov blanket i.e. what variables that need to be conditioned on in Gibbs sampling

Potential Factorization

We can define a clique of a undirected graph a set of nodes connected to each other. There is an equivalent factorization as we saw in the directed graph setting for undirected graphs. Particularly we can take the product of a potential function for cliques within the graph to compute the joint distribution of the random variables we are modelling:

$$P(X) = \prod_C \Psi_c(X_c)$$

Where we define $\Psi(.)$ as a potential function.

Factor Graph

Another method of computing joint probability distributions on undirected graphs is by defining factor functions between different nodes of a graph. Starting from the top factor graph and going down the column we can write down the joint distribution of random variables of the graph in terms of products of the factor functions (Figure 5):

$$P(X) = f_1(X_1, X_3) f_2(X_1, X_2) f_3(X_2, X_3)$$
$$P(X) = f_2(X_1, X_2) f_1(X_2, X_3)$$
$$P(X) = f(X_1, X_2, X_3)$$

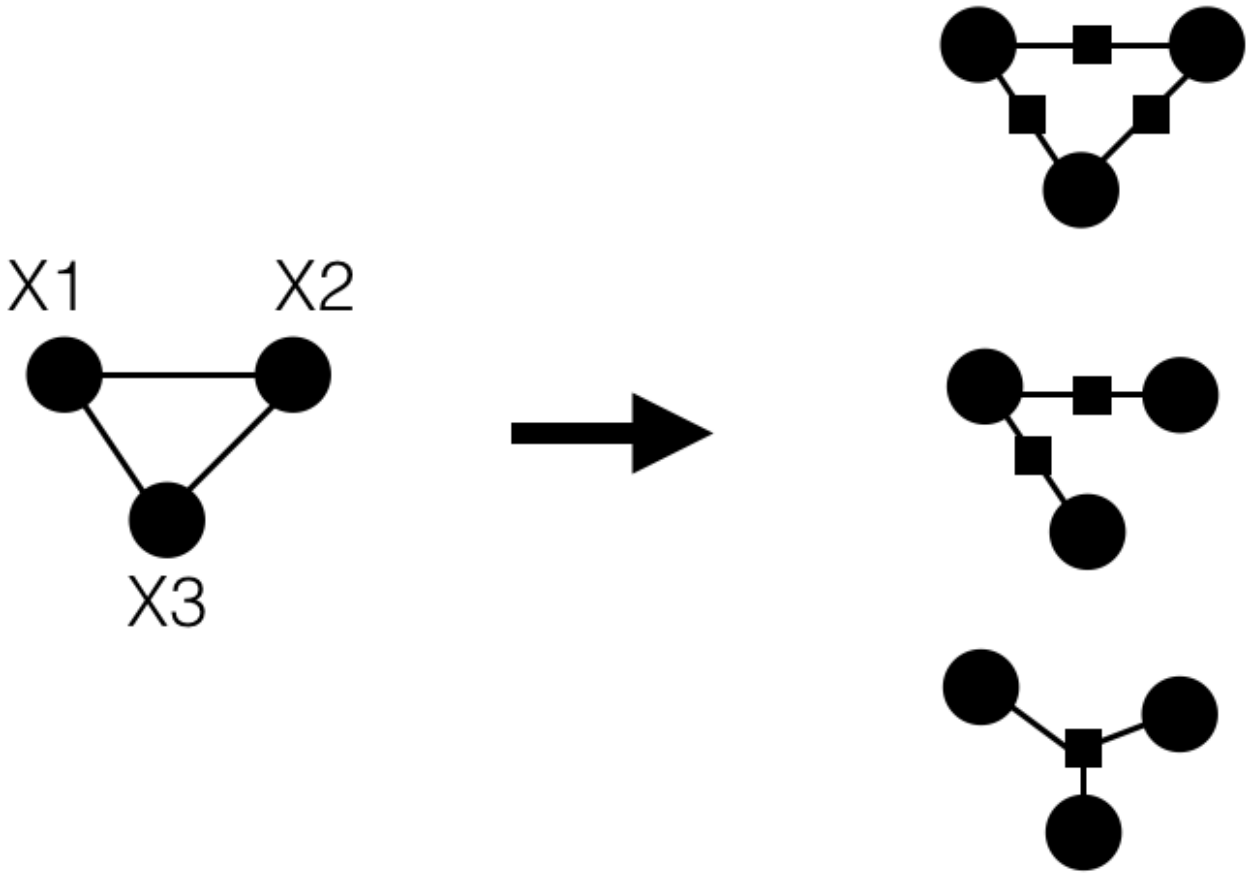


Figure 17.5. Factor Graph Example

Factors graphs allow one to deal with more complicated graph structures. One should note that if the factor graph is a tree then the general sum-product algorithm gives efficient computation of all singleton marginals $P(X_i)$ for all $i \in V$ where V is the set of vertices. Also note to look into the junction tree algorithm.

Diffusion Theory

Jump processes which include many Markov chains, Poisson processes have low transition probabilities but when transitions occur large changes in state are made. Diffusion processes such as Brownian motion and other Markov chains have high transition probabilities but have when a transition occurs small transitions in state are made. We can model the behavior of diffusion process using second order partial differential equations:

$$P_t(x) = P(X(t) = x)$$

$$\frac{\partial P_t(x)}{\partial t} = \frac{\partial}{\partial t}[P_t(x)M(x)] + \frac{\partial^2}{\partial t^2}[P_t(x)M(x)]$$

$$x(t + \delta t) - x(t) = \delta_x$$

$$E[\delta_x] = M(x)\delta t + O(\delta t)$$

$$Var[\delta_x] = V(x)\delta t + O(\delta t)$$

We can use the above diffusion equations to solve for the stationary distributions and transient distributions of a variety of stochastic processes.