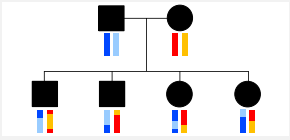
1

BTRY 4840/6840, CS 4775
Computational Genetics and Genomics



September 25, 2018

2

Announcements

- Problem set 2 due today
- Problem set 3 out later today
  - HMMs: Viterbi, Forward-backward, …

3

Today's lecture

- Finish log probabilities
- Continuing Hidden Markov models (HMMs)
  - One form of unsupervised learning
- Application of HMMs to genetics:
  - Background on genetic assays
  - Inferring haplotypes in families
  - Background on inferring haplotypes in unrelated samples

4

Numerical stability in evaluating HMMs

5

Issue: computers store numbers in finite space

- To analyze HMMs, we multiply many numbers < 1
  - Can produce underflow: a number too small for the computer to store
- Example:
  - Want to analyze a sequence of length 10,000 bases
  - $a_{kl} = .1$ for all k, l $\quad e_k(b) = 1$ for some k, b pair
  - Viterbi path score: $10^{-10,000}$
  - ➢ Way too small for standard floating point representations

6

Underflow example

- Underflow example in Python:

```
>>> prob = 1
>>> for i in range(0,10000):
...     prob *= .1
...
>>> prob
0.0
```

- Solution: log probabilities
  - Effective way to compute using very small numbers
  - Have $\log(x^y) = y \log(x)$ – much easier number to represent for large positive/negative exponent y

7

Example: no underflow using log probabilities

```
>>> from math import log
>>> logProb = 0
>>> logTransitionProb = log(.1)
>>> for i in range(0,10000):
...     logProb += logTransitionProb
...
>>> logProb
-23025.850929942502
>>> logProb / log(.1)
10000.00000000089
```

Notes:
- `log()` is expensive
  - ➢ Should precompute probabilities of HMM parameters
- Log probabilities are often more efficient because we sum instead of multiply

8

Issue: need to be able to add probabilities

- Forward & backward algorithms sum probabilities
  - Inconvenient: $\log(x) + \log(y) \neq \log(x + y)$
- Can try converting to normal space
- Approach 1:
  `log( exp(a) + exp(b) )`
  (Here a, b are log probabilities)

Mathematically correct, but
- Can underflow:
  `exp(-1000) == 0.0`
- Expensive: two `exp()` and one `log()` calls

9

Improved sum of log probabilities

- Let a, b be log probabilities, then

$$\exp(a) + \exp(b) = \exp(a) \cdot \left(1 + \frac{\exp(b)}{\exp(a)}\right) = \exp(a) \cdot (1 + \exp(b - a))$$

So: $\log(\exp(a) + \exp(b)) = a + \log(1 + \exp(b - a))$

- Better numerical stability:
  - Worst case for approach 1: `exp(a) + exp(b) == 0.0`, so get `log(0.0)`: $-\infty$ or undefined
  - Worst case for above approach: `exp(b-a) == 0.0`, so get `a + log(1.0 + 0.0) == a` ← **much** better
- Better computation:
  - Now only one `exp()` and one `log()` call

10

Even better sum of log probabilities

- If $\exp(b - a)$ is very small (e.g., 10^{-20}), can have `1+exp(b-a) == 1.0`, and thus `log(1) == 0`
- Solution: `log1p(x)`
  - Computes log(1+x) at higher precision:

```
>>> log1p(1e-20)
9.999999999999995e-21
>>> log(1+1e-20)
0.0
```

- This is related to the lower.tail=FALSE option in R:
  - One-tailed p-value definition: $p = 1 - F(x)$, F the CDF: $P(X \leq x)$

```
pnorm(7) == 1.0  ⇒  1-pnorm(7) == 0.0
pnorm(7, lower.tail=FALSE) == 1.279813e-12
```

11

Python implementation to sum log probabilities

```
from numpy import log1p
from math import exp
def sumLogProb(a, b):
  if a > b: return a + log1p(exp(b - a))
  else:     return b + log1p(exp(a - b))
```

- If statement:
  - Why do we prefer max of `a` and `b`?
    - Trying to sum the corresponding probabilities: is $\geq \max(a, b)$
- For efficiency, Durbin *et al.* suggest generating a table for log(1 + exp(b-a))
  - Idea: `b-a` often close to 0, so don't need large table
  - Use linear interpolation between table values

12

Hidden Markov models

How do we sample from $P(\boldsymbol{z}|\boldsymbol{x})$?

Use forward probabilities

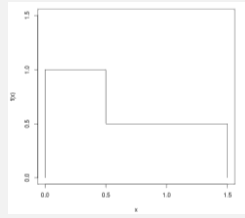
13

Background: what is random sampling?

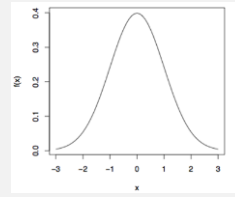
- Sampling: choosing a subset of a population in order to identify characteristics of the full population
  - Example (simple random sampling): Given a population of 10,000 students, select 500 at random with equal probability
- Can sample from a given distribution
  - Want sampling to give a value x with probability proportional to its density (i.e., pdf) $f(x)$

14

Example distributions



A given $x \in [0, 0.5)$ twice as likely as often as a given $x \in [0.5, 1.5]$



$x \in [-1.96, 1.96]$ approximately 95% of the time

Sampling key idea:
Random samples follow original distribution
With infinite data could reconstruct entire distribution

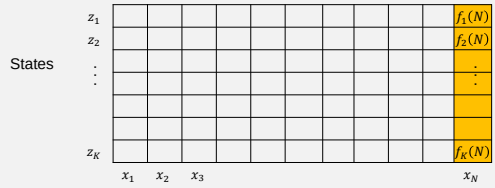
15

Why sample state path?

- Posterior decoding isn't always a valid state path
- Viterbi decoding is the most likely path
  - But only one path: could be many other likely paths
- Can sample $\boldsymbol{z}$ from $P(\boldsymbol{z}|\boldsymbol{x})$
  - Is a distribution over vectors $\boldsymbol{z}$ (typically high dimensional)
  - Can sample $\boldsymbol{z}$ multiple times as alternate to expectation maximization (EM – to be discussed later)
    - More randomized: possibly less prone to find local maxima
- How is this different from generating random data?
  - Generating samples from $P(\boldsymbol{x}, \boldsymbol{z})$ – gives both $\boldsymbol{x}$ and $\boldsymbol{z}$
  - Want a random $\boldsymbol{z}$ sampled from $P(\boldsymbol{z}|\boldsymbol{x})$: fixed observed data

16

Sampling hidden states with forward probabilities



- How should we begin?
  - Sample state k at step N with probability $\frac{f_k(N)}{\sum_{k'=1}^{K} f_{k'}(N)}$
- Iterate:
  - Given state l at step $i+1$, sample state k at step i with probability $\frac{f_k(i)a_{kl}}{\sum_{k'=1}^{K} f_{k'}(i)a_{k'l}}$

17

Randomized training algorithm

1. Initialize $\boldsymbol{\theta}^{(0)} = \left(a_{kl}^{(0)}, e_k^{(0)}(b)\right)$ for all k, l, b
2. Iterate:
   1. Run forward algorithm: compute $f_k(i)$ for all k, i
   2. Sample n hidden state vectors $\boldsymbol{z}$ from $P\left(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}\right)$
   3. Calculate $A_{kl}^{(t)}, E_k^{(t)}(b)$ using all n paths $\boldsymbol{z}$ [as in supervised]
   4. Calculate new $\boldsymbol{\theta}^{(t+1)} = (a_{kl}^{(t+1)}, e_k^{(t+1)}(b))$ for all k, l, b [MLE]
3. Terminate when $\Delta P(\boldsymbol{x}|\boldsymbol{\theta})$ small or fixed # iterations

- Notes:
  - May not converge as quickly as EM: randomized
  - But stochastic so may yield higher $P(\boldsymbol{x}|\boldsymbol{\theta})$

18

Desired uses of HMMs (highlighted done)

- Evaluation:
  - **Given**: observed $\boldsymbol{x}$ and HMM specification
    Question: what is the joint probability of $\boldsymbol{x}$ and a given $\boldsymbol{z}$?
    Question: what is the likelihood of $\boldsymbol{x}$ based on the HMM?
- Decoding:
  - **Given**: observed $\boldsymbol{x}$ and HMM
    Question: what sequence of hidden states produced $\boldsymbol{x}$?
  - Viterbi decoding: most likely hidden state sequence
  - Posterior probability of hidden states: probability of each state z_i producing each x_i
- Learning:
  - **Given**: observed $\boldsymbol{x}$ and HMM without complete probabilities
    Question: what emission, transition probabilities produced $\boldsymbol{x}$?

19

Applying HMMs to genetic data

Background: genetic assays

20

Genetic variants of interest for many analyses

- Genomes of any two humans are > 99.9% identical
- However, variation exists:
  - Approximately $.001 \times 3\times10^9 = 3\times10^6$ variants between a pair of human genomes
- Why do we care about variation?
  - Genetic variation drives biological variation
    - ➢ Want to characterize it to understand biological impacts of variants
  - Genetic variation reveals population genetics
    - ➢ Must account for/model these dynamics to avoid false discoveries related to biology
    - ➢ Of direct interest to studying evolutionary & demographic history
      - Will see examples of both the above later

21

Variation levels differ widely across species

- Plot: average number of pairwise differences between two genomes in a given species
  - Per base pair as a %
- Log10 scale
  - Max ~8%
  - Min ~0.01%

Leffler *et al.* (2012)

22

How do we characterize genetic variation?

Historically labor intensive: typically assayed ~10 variants for a study

Today:

Saliva
Blood

Whole genome sequencing of human ~$1000
- ➢ 30× coverage

High throughput sequencing

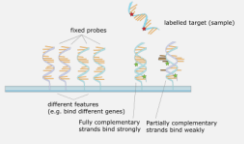
DNA microarray
(SNP microarray)

- ➢ Information on variable positions (SNPs) ~300,000 to 2.5 million sites
- ➢ Can be ~$100 or less

23

SNP microarrays

- Microarray: flat substrate (often glass or silicone) with numerous probes
  - Many kinds of microarrays: protein, antibody, DNA, etc.



1. Sample DNA is labeled with florescent dye
2. DNA allowed to hybridize with probes on microarray
3. Lasers used to detect binding to probe sequences

24

SNP microarrays

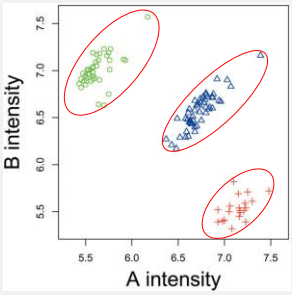
- To design probes, need:
  - Sequence flanking SNP
  - Ideally whole genome sequence of species to ensure probe sequence is unique

ACTTTCAGTGTCGCAT
ACTTTCATTGTCGCAT } Homologous sequences: G/T SNP
Probe target

- Probe sequences:
  - Are complementary to flanking sequence
    - Example: ATGCGACAA (recall: pair is reverse complement)
      ATGCGACAC
  - Are relatively long (often 50 bp)

25

SNP genotypes given by probe intensities



Called clustering enables genotype calling
Have three genotypes: A/A, A/B, B/B

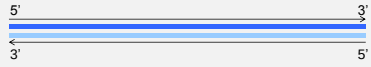
Lamy *et al.* (2006)

26

High throughput sequencing uses PCR

Sequencing uses polymerase chain reaction (PCR)

Sample DNA (fragmented)

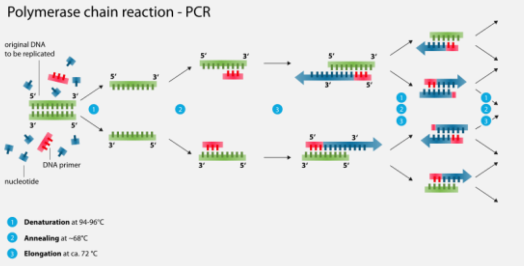


Primers: portion of 5' sequence from each strand

1. When heated, DNA denatures (comes apart)
2. Primer sequences anneal to 3' end of complement
3. Polymerase synthesizes the complementary strand beginning at the primer sequence
   - Now have 2× more DNA
4. Repeat: exponential increase in DNA product

27

PCR schematic



Schematic by Enzoklop

28

High throughput sequencing

- Sequencing details complicated, but at a high level:
  - Sequencer performs PCR
  - Nucleotides added during elongation are florescently labeled; colors differ between A, C, G, T
  - Machine detects color of each incorporated nucleotide
- Above is done across millions of DNA fragments
- Produces "reads" of specific lengths
  - Often 100 or 150 bp
- For humans, 30× coverage, 100 bp reads ≈ 960 million reads!

29

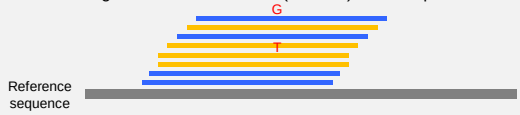
Sequencing next step: read mapping

- SNP microarrays have probes at known locations
  - Probes chosen to be unique (based on human reference genome)
- Sequencing reads can come from anywhere
  - This leads to the problem of read mapping / alignment
    - Must determine where in the genome a given 100 bp sequence comes from
  - May discuss methods for this later this term

30

Final step: genotype calling

- After read mapping: many reads overlap a position
  - Reads are mapped from both homologs (blue, orange)
  - Homologs will have differences (variants) at some positions



Genotype calling: method for deciding genotypes based on some input (SNP array intensity plot or mapped reads)

- Won't discuss, but interesting; can view as hypothesis testing
- Some packages (e.g., GATK) use machine learning techniques

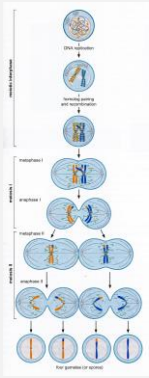
31

Applying HMMs to genetic data
Inferring haplotypes in family data

32

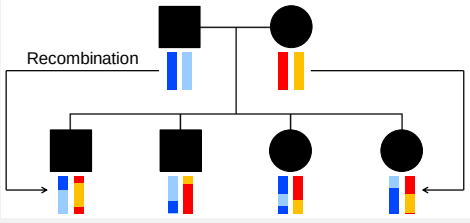
Homologous recombination occurs during meiosis

- Produces mosaic chromosomes inherited by offspring



33

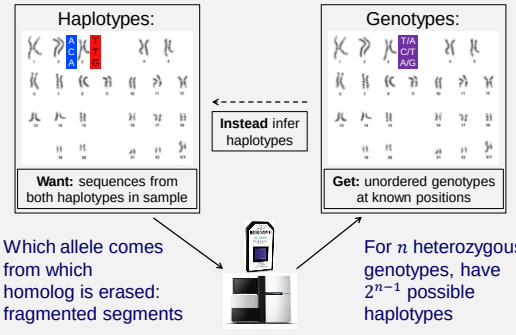
Haplotype transmissions in pedigrees



Pedigree representation
Squares and circles: males and females, respectively
Parents have line joining them and connected to children

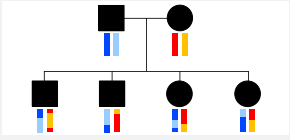
34

Genotypes do not give haplotype information



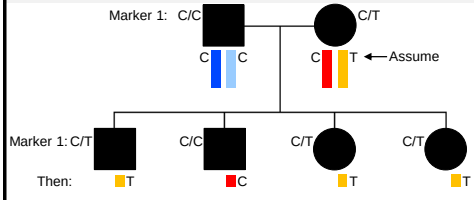
35

Thinking through haplotype inference problem

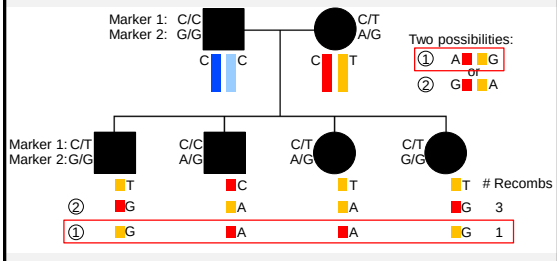


- Assume we have > 100,000 SNP genotypes
- Hidden state: haplotypes of parents, children at each marker
- Huge help: recombination relatively rare
  - Rough approximation: 1 crossover per 100 million bp (human)
    - Crossover: one form of recombination
      Can ignore other type (non-crossover: very short ~100-1,000 bp)
  - Human chromosome 1 is ~249 million bp

Phasing example



Phasing example



Final notes

- Summary:
  - Function to precisely sum log probabilities
  - Sampling from posterior distribution of states $P(\boldsymbol{z}|\boldsymbol{x})$
  - Beginning HMM formulation for haplotype inference in families