Copy number Aberrations

Normal cells:
Red and gene probes to regions on X chromosome: >2 copies of green region

Cancer cells:
Extensive gene duplication/deletion

http://cs.brown.edu/courses/csci1950-z/
DNA Microarrays

Comparative Genomic Hybridization (CGH)

Measuring Mutations in Cancer
**CGH Analysis (1)**

Divide genome into segments of equal copy number

![Graph showing genomic position with deletion and amplification areas marked]

**CGH Analysis (2)**

Identify aberrations common to multiple samples

Chromosome 3 of 26 lung tumor samples on mid-density cDNA array. Common deletion located in 3p21 and common amplification – in 3q.

![Heatmap showing genomic position with samples and deletion/amploification areas marked]
CGH Analysis (1)
Divide genome into segments of equal copy number

Input: $X_i = \log_2 \frac{T_i}{R_i}$, clone $i = 1, ..., N$
Output: Assignment $s(y_i) \in \{S_1, ..., S_K\}$
$S_i$ represent copy number states

An Approach to CGH Segmentation
• Circular Binary Segmentation (CBS), Olshen et al. 2004
• Use hypothesis test to compare means of two intervals using t-test (whiteboard)
Interval Score

Assume:
• $X_i$ are independent, normally distributed
• $\mu$ and $\sigma$ denote the mean and standard deviation of the normal genomic data.

Given an interval $I$ spanning $k$ probes, we define its score as:

$$S(I) = \frac{\sum_{i \in I} (X_i - \mu)}{\sqrt{|I| \cdot \sigma}}$$

Significance of Interval Score

Assume:
• $X_i \sim N(\mu, \sigma)$

$$S(I) = \frac{\sum_{i \in I} (X_i - \mu)}{\sqrt{|I| \cdot \sigma}}$$

$$P(S(I) > z) = \frac{1}{\sqrt{2\pi}} \frac{1}{z} e^{-\frac{1}{2}z^2}$$
**The MaxInterval Problem**

**Input:** A vector $X=(X_1...X_n)$

**Output:** An interval $I \subset [1...n]$, that maximizes $S(I)$

\[
S(I) = \sum_{j \in I} (X_j - \mu) / \sqrt{|I| \cdot \sigma}
\]

Other intervals with high scores may be found by recursively calling this function.

Exhaustive algorithm: $O(n^2)$

---

**MaxInterval Algorithm I: LookAhead**

Assume given:
- $m$: An upper bound for the value of a single element $X_i$
- $t$: A lower bound on the maximum score

\[
I = [i,...,i+k-1] \quad I' = [i,...,i+k+r-1]
\]

\[
\begin{align*}
\text{sum} & \quad s = \sum_{j \in I} X_j \\
\text{length} & \quad k \\
\text{score} & \quad S(I) = s / \sqrt{k} \\
\end{align*}
\]

\[
S(I') \leq \frac{(s + m r)}{\sqrt{(k + r)}}
\]

Solve for first $r$ for which $S(I)$ may exceed $t$.

**Complexity:** Expected $O(n^{1.5})$ (unproved)
MaxInterval Algorithm II: 
Geometric Family Approximation (GFA)

For $\varepsilon > 0$ define the following geometric family of intervals:

$$k_j = (1 + \varepsilon)^j, \quad \Delta_j = \varepsilon \cdot k_j$$

$$\Omega(j) = \left\{ [i\Delta_j, i\Delta_j + k_j - 1]: 0 \leq i \leq \frac{n - k_j}{\Delta_j} \right\}$$

$$\Omega = \bigcup_j \Omega(j)$$

**Theorem:**

Let $I^*$ be the optimal scoring interval.
Let $J$ be the leftmost longest interval of $\Omega$ fully contained in $I^*$.
Then $S(J) \geq S(I^*)/\alpha$, where $\alpha \propto (1 - \varepsilon^2)^{-1}$.

**Complexity:** $O(n)$

---

**Benchmarking**

Linear regression suggests that the complexities of the Exhaustive, LookAhead and GFA algorithms are $O(n^2)$, $O(n^{1.5})$, $O(n)$, respectively.
Applications: Single Samples

Chromosome 16 of HCT116 colon carcinoma cell line on high-density oligo array (n=5,464).

Chromosome 17 of several breast carcinoma cell lines on mid-density cDNA array (n=364).

Another Approach to CGH Segmentation

Use Hidden Markov Model (HMM) to “parse” sequence of probes into copy number states.
Example: The Dishonest Casino

A casino has two dice:

- Fair die
  \[ P(1) = P(2) = P(3) = P(5) = P(6) = 1/6 \]
- Loaded die
  \[ P(1) = P(2) = P(3) = P(5) = 1/10 \]
  \[ P(6) = 1/2 \]

Casino player switches back-&-forth between fair and loaded die once every 20 turns

**Game:**
1. You bet $1
2. You roll (always with a fair die)
3. Casino player rolls (maybe with fair die, maybe with loaded die)
4. Highest number wins $2
Question # 1 – Evaluation

GIVEN

A sequence of rolls by the casino player

124526462146136136616646616366163616515615115146123562344

QUESTION

How likely is this sequence, given our model of how the casino works?

This is the EVALUATION problem in HMMs.

Prob = $1.3 \times 10^{-35}$

Question # 2 – Decoding

GIVEN

A sequence of rolls by the casino player

124526462146136136616646616366163616515615115146123562344

FAIR       LOADED       FAIR

QUESTION

What portion of the sequence was generated with the fair die, and what portion with the loaded die?

This is the DECODING question in HMMs.

This is what we want to solve for CGH analysis.
Question # 3 – Learning

GIVEN

A sequence of rolls by the casino player

```
245526462146136136616646616366163661515
5115146123562344
```

Prob(6) = 64%

QUESTION

How “loaded” is the loaded die? How “fair” is the fair die? How often does the casino player change from fair to loaded, and back?

This is the LEARNING question in HMMs

Definition of a hidden Markov model

**Definition:** A hidden Markov model (HMM)

- **Alphabet** $\Sigma = \{ b_1, b_2, \ldots, b_M \}$
- **Set of states** $Q = \{ 1, \ldots, K \}$
- **Transition probabilities** between any two states
  
  $a_{ij} = \text{transition prob from state } i \text{ to state } j$

  $a_{i1} + \ldots + a_{iK} = 1, \text{ for all states } i = 1\ldots K$

- **Start probabilities** $a_{0i}$

  $a_{01} + \ldots + a_{0K} = 1$

- **Emission probabilities** within each state

  $e_i(b) = P( x_i = b | \pi_t = k)\$

  $e_i(b_1) + \ldots + e_i(b_M) = 1, \text{ for all states } i = 1\ldots K$
The dishonest casino model

A HMM is memory-less

At each time step $t$, the only thing that affects future states is the current state $\pi_t$.

\begin{align*}
P(\pi_{t+1} = k \mid "\text{whatever happened so far}" ) &= \ P(\pi_{t+1} = k \mid \pi_1, \pi_2, \ldots, \pi_t, x_1, x_2, \ldots, x_t) \\
P(\pi_{t+1} = k \mid \pi_t) &= \end{align*}
A parse of a sequence

Given a sequence \( x = x_1 \ldots x_N \),
A parse of \( x \) is a sequence of states \( \pi = \pi_1, \ldots, \pi_N \)

Likelihood of a Parse

Simply, multiply all the orange arrows!
(transition probs and emission probs)
Likelihood of a parse

Given a sequence \( x = x_1 \ldots x_N \) and a parse \( \pi = \pi_1, \ldots, \pi_N \)

To find how likely is the parse:
(given our HMM)

\[
P(x, \pi) = P(x_1, \ldots, x_N, \pi_1, \ldots, \pi_N) = \prod_{j=1}^{N} P(x_j | \pi_j) P(\pi_j | \pi_{j-1}) \prod_{j=1}^{N} P(\pi_j | \pi_{j-1}) \ldots \prod_{j=1}^{N} P(\pi_j | \pi_1) P(x_1 | \pi_1) \ldots a_{\pi_1}\ldots a_{\pi_N} e_{\pi_1}(x_1) \ldots e_{\pi_N}(x_N)
\]

A compact way to write

\[a_{\pi_1} a_{\pi_2} \ldots a_{\pi_{N-1}} e_{\pi_1}(x_1) \ldots e_{\pi_N}(x_N)\]

Number all parameters \( a_i \) and \( e_i(b) \); \( n \) params

Example:

\[a_{0\text{Fair}} : \theta_1; a_{0\text{Loaded}} : \theta_2; \ldots e_{\text{Loaded}}(6) = \theta_{18}\]

Example: the dishonest casino

Let the sequence of rolls be:

\[x = 1, 2, 1, 5, 6, 2, 1, 5, 2, 4\]

Then, what is the likelihood of

\[\pi = \text{Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair}\]

(say initial probs \( a_{0\text{Fair}} = \frac{1}{2}, a_{0\text{Loaded}} = \frac{1}{2} \))

\[
\frac{1}{2} \times P(1 | \text{Fair}) P(\text{Fair} | \text{Fair}) P(2 | \text{Fair}) P(\text{Fair} | \text{Fair}) \ldots P(4 | \text{Fair}) = \\
\frac{1}{2} \times \left(\frac{1}{6}\right)^{10} \times (0.95)^9 = 0.0000000521158647211 \approx 0.5 \times 10^{-9}
\]
Example: the dishonest casino

So, the likelihood the die is fair in this run is just $0.521 \times 10^{-9}$

OK, but what is the likelihood of

$\pi = \text{Loaded, Loaded, Loaded, Loaded, Loaded, Loaded, Loaded, Loaded, Loaded, Loaded}$?

$\frac{1}{2} \times P(1 \mid \text{Loaded}) \times P(\text{Loaded}, \text{Loaded}) \times \ldots \times P(4 \mid \text{Loaded}) =$

$\frac{1}{2} \times (\frac{1}{10})^9 \times (\frac{1}{2})^1 \times (0.95)^9 = .00000000015756235243 \approx 0.16 \times 10^{-9}$

Therefore, it somewhat more likely that all the rolls are done with the fair die, than that they are all done with the loaded die

Example: the dishonest casino

Let the sequence of rolls be:

$x = 1, 6, 6, 5, 6, 2, 6, 6, 3, 6$

Now, what is the likelihood $\pi = F, F, \ldots, F$?

$\frac{1}{2} \times (\frac{1}{6})^{10} \times (0.95)^9 = 0.5 \times 10^{-9}$, same as before

What is the likelihood

$\pi = L, L, \ldots, L$?

$\frac{1}{2} \times (\frac{1}{10})^4 \times (\frac{1}{2})^6 \times (0.95)^9 = .00000049238235134735 \approx 0.5 \times 10^{-7}$

So, it is 100 times more likely the die is loaded
HMM Model for CGH data

Fridlyand et al. (2004)

A model for CGH data

K states copy numbers

Emissions: Gaussians

Homozygous Deletion (copy =0)
Heterozygous Deletion (copy =1)
Normal (copy =2)
Duplication (copy >2)
The three main questions on HMMs

1. **Evaluation**
   GIVEN: HMM M, and a sequence x,
   FIND: \( \text{Prob}[\,x \mid M\,] \)

2. **Decoding**
   GIVEN: HMM M, and a sequence x,
   FIND: sequence \( \pi \) of states that maximizes \( P[\,x, \pi \mid M\,] \)

3. **Learning**
   GIVEN: HMM M, with unspecified transition/emission
   pros. and a sequence x,
   FIND parameters \( \theta = (e_i(.), a_{ij}) \) that maximize \( P[\,x \mid \theta\,] \)

Let’s not be confused by notation

\( P[\,x \mid M\,] \): The probability that sequence x was generated by the model

The model is: architecture (#states, etc)
   + parameters \( \theta = (a_{ij}, e_i(.)) \)

So, \( P(x \mid M) \) is the same with \( P(x\mid \theta) \), and \( P(\,x\,\mid \theta) \), when the architecture, and the parameters, respectively, are implied

Similarly, \( P(\,x, \pi \mid M\,) \), \( P(\,x, \pi \mid \theta\,) \) and \( P(\,x, \pi\,) \) are the same when the architecture, and the parameters, are implied

In the LEARNING problem we always write \( P(x\mid \theta) \) to emphasize that we are seeking the \( \theta^* \) that maximizes \( P(x\mid \theta) \)
Problem 1: Decoding

Find the most likely parse of a sequence

Decoding

GIVEN \( x = x_1 x_2 \ldots x_N \)
Find \( \pi = \pi_1, \ldots, \pi_N \)
to maximize \( P[x, \pi] \)

\[ \pi^* = \text{argmax}_\pi P[x, \pi] \]

Maximizes 

\[ a_{\pi_1} e_{\pi_1}(x_1) a_{\pi_1 \pi_2} \ldots a_{\pi_{N-1} \pi_N} e_{\pi_N}(x_N) \]

*Dynamic Programming!*

\[ V_k(i) = \max_{\pi_1 \ldots \pi_{i-1}} P[x_1 \ldots x_i, \pi_1, \ldots, \pi_{i-1}, x_i, \pi_i = k] \]

= Prob. of most likely sequence of states ending at state \( \pi_i = k \)
Decoding – main idea

**Inductive assumption:** Given that for all states $k$, and for a fixed position $i$,

$$V_k(i) = \max_{\pi_1, \pi_{i-1}} P[x_1 ... x_{i-1}, \pi_1, ..., \pi_{i-1}, x_i, \pi_i = k]$$

What is $V_l(i+1)$?

From definition,

$$V_l(i+1) = \max_{\pi_1, \pi_i} P[x_1 ... x_i, \pi_1, ..., \pi_i, x_{i+1}, \pi_{i+1} = l]$$

$$= \max_{\pi_1, \pi_i} P(x_{i+1}, \pi_{i+1} = l \mid x_1 ... x_i, \pi_1, ..., \pi_i) P[x_1 ... x_i, \pi_1, ..., \pi_i]$$

$$= \max_{\pi_1, \pi_i} P(x_{i+1}, \pi_{i+1} = l \mid \pi_i) P[x_1 ... x_{i-1}, \pi_1, ..., \pi_{i-1}, x_i, \pi_i]$$

$$= \max_k \left[ P(x_{i+1} \mid \pi_{i+1} = l) \max_{\pi_1, \pi_{i-1}} P[x_1 ... x_{i-1}, \pi_1, ..., \pi_{i-1}, x_i, \pi_i = k] \right]$$

$$= \max_k \left[ P(x_{i+1} \mid \pi_{i+1} = l) P(\pi_{i+1} = l \mid \pi_i = k) V_k(i) \right]$$

$$= e_l(x_{i+1}) \max_k a_{ki} V_k(i)$$

**The Viterbi Algorithm**

Input: $x = x_1 ... x_n$

**Initialization:**

$$V_k(0) = 1 \quad (0 \text{ is the imaginary first position})$$

$$V_k(0) = 0, \text{ for all } k > 0$$

**Iteration:**

$$V_l(i) = e_i(x_i) \times \max_k a_{ki} V_k(i-1)$$

$$\text{Ptr}_l(i) = \arg \max_k a_{ki} V_k(i-1)$$

**Termination:**

$$P(x, \pi^*) = \max_k V_k(N)$$

**Traceback:**

$$\pi_n^* = \arg \max_k V_k(N)$$

$$\pi_{i-1}^* = \text{Ptr}_{i-1}(i)$$
The Viterbi Algorithm

Similar to "aligning" a set of states to a sequence

**Time:**
O(K^2N)

**Space:**
O(KN)

Viterbi Algorithm – a practical detail

Underflows are a significant problem

\[ P[x_1, ..., x_n, \pi_1, ..., \pi_n] = a_{01} a_{11} a_{12} ... a_{ni} e_{x1}(x_1) ... e_{xn}(x_n) \]

These numbers become extremely small – underflow

**Solution:** Take the logs of all values

\[ V_i(i) = \log e_{x_i}(x_i) + \max_k [ V_{i-1}(i) + \log a_{ki} ] \]
Example

Let $x$ be a long sequence with a portion of $\sim 1/6$ 6's, followed by a portion of $\sim 1/2$ 6's...

$$x = 123456123456...123456626364656...1626364656$$

Then, it is not hard to show that optimal parse is (exercise):

$$FFF...F LLL...L$$

6 characters “123456” parsed as $F$, contribute $0.95^5 \times (1/6)^6 = 1.6 \times 10^{-5}$

parsed as $L$, contribute $0.95^5 \times (1/2)^3 \times (1/10)^3 = 0.4 \times 10^{-5}$

“162636” parsed as $F$, contribute $0.95^5 \times (1/6)^6 = 1.6 \times 10^{-5}$

parsed as $L$, contribute $0.95^5 \times (1/2)^3 \times (1/10)^3 = 9.0 \times 10^{-5}$

A model for CGH data

K states

S1 S2 S3 S4

Homoygous Deletion (copy = 0)
Heterozygous Deletion (copy = 1)
Normal (copy = 2)
Duplication (copy > 2)

Emissions: Gaussians

Use Viterbi algorithm to derive segmentation.
Forward/backward to compute state of a single probe.
Sources

- http://ai.stanford.edu/~serafim/CS262_2006/ (HMM slides)