Lecture 4

Modeling Biological Sequences using Hidden Markov Models
<table>
<thead>
<tr>
<th>Project</th>
<th>Sets</th>
<th>Week</th>
<th>Date</th>
<th>Topic</th>
<th>Lec</th>
<th>Topic</th>
<th>Read*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Describe your previous research, areas of interest in computational biology, type of project that best fits your interests. Post in a profile that lets your classmates know you &amp; find potential partners.</td>
<td>PS1 out onL1-L5</td>
<td>1</td>
<td>Thu, Sep 8</td>
<td>Introduction</td>
<td>L1</td>
<td>Intro: Biology, Algorithms, Machine Learning, Course Overview</td>
<td>1</td>
</tr>
<tr>
<td>Project profile due Tue 9/27</td>
<td></td>
<td></td>
<td>Fri, Sep 9</td>
<td></td>
<td>R1</td>
<td>Recitation 1: Biology and Probability Review</td>
<td></td>
</tr>
<tr>
<td>Identify previous project proposals, recent papers, and potential partners that match your areas of interest. List initial project ideas and partners.</td>
<td>PS2 out onL6-R4</td>
<td>4</td>
<td>Thu, Sep 27</td>
<td>Module I: Aligning and Modeling Genomes</td>
<td>L3</td>
<td>Alignment I: Dynamic Programming, Global and local alignment</td>
<td>2</td>
</tr>
<tr>
<td>Project team/ideas due Tue 10/4</td>
<td></td>
<td></td>
<td>Fri, Sep 30</td>
<td></td>
<td>R2</td>
<td>Recitation 2: Deriving Parameters of Alignment, Multiple Alignment</td>
<td></td>
</tr>
<tr>
<td>Form teams of two, specify project goals, division of work, milestones, datasets, challenges, Prepare slide presentation for the class and the mentors. Project proposal due Tue 10/18. Presented to mentor on Fri 10/21</td>
<td>PS3 out onL10-R6</td>
<td>6</td>
<td>Tue, Oct 11</td>
<td>Module II: Gene Expression and Networks</td>
<td>L8</td>
<td>Epigenomics: ChIP-Seq, Read mapping, Peak calling, IGR, Chromatin states</td>
<td>15, 16</td>
</tr>
<tr>
<td>Project planning: research areas, initial ideas, type of project, mentor matching, finding partners 32D-507</td>
<td></td>
<td></td>
<td>Thu, Oct 13</td>
<td></td>
<td>L9</td>
<td>Three-dimensional chromatin interactions: 3C, 5C, HiC, ChIP-Seq</td>
<td>19</td>
</tr>
<tr>
<td>Evaluate/discuss three peer proposals, NIH review format. Reviews due Fri 10/28</td>
<td>PS4 out onL13-R8</td>
<td>8</td>
<td>Tue, Oct 25</td>
<td>Module IV: Population and Disease Genetics</td>
<td>L10</td>
<td>Regulatory Motifs: Discovery, Representation, PBMs, Gibbs Sampling, EM</td>
<td>17</td>
</tr>
<tr>
<td>Reviews returned Tue 11/1</td>
<td></td>
<td></td>
<td>Thu, Oct 27</td>
<td></td>
<td>L11</td>
<td>Networks I: Neural Networks, Belief Networks, Deep learning</td>
<td>20, 21</td>
</tr>
<tr>
<td>Address peer evaluations, revise aims, scope, list of final deliverables / goals. Response due Thu 11/10</td>
<td>PS5 out onL17-R9</td>
<td>10</td>
<td>Tue, Nov 8</td>
<td>Module V: Comparative Genomics and Evolution</td>
<td>L17</td>
<td>Comparative genomics and evolutionary signatures</td>
<td>4</td>
</tr>
<tr>
<td>Midcourse report due Wed 11/23</td>
<td></td>
<td></td>
<td>Thu, Nov 10</td>
<td></td>
<td>L18</td>
<td>Genome Scale Evolution, Genome Duplication</td>
<td>5, 6</td>
</tr>
<tr>
<td>Complete your milestones, finalize progress report, conference publication format. As part of report, comment on your overall project experience. Written report due Sun 12/11</td>
<td>No more posts (work on your final project) (work on your final project)</td>
<td>12</td>
<td>Tue, Nov 22</td>
<td>Quiz</td>
<td>In Class Quiz (the only quiz - the class has no final exam) - covers L1-L20, R1-R9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conference format slide pres. Talks on Tue 12/13</td>
<td></td>
<td></td>
<td>Thu, Nov 24</td>
<td></td>
<td>No recitation, thanksgiving break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R11</td>
<td>Recitation 11: Presentation Tips - Intro, discussion, Slides, Presentation skills</td>
<td>30, 34, 35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R12</td>
<td>Recitation 10: Project Feedback, results, interpretation, directions</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R13</td>
<td>Recitation 9: Phylogenetic distance metrics, Coalescent Process</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
- **Module 1: Computational foundations**
  - Dynamic programming: exploring exponential spaces in poly-time
  - Linear-time string matching, Hashing, Content-based indexing
  - Hidden Markov Models: decoding, evaluation, parsing, learning
- **Last week: Sequence alignment / comparative genomics**
  - Local/global alignment: infer nucleotide-level evolutionary events
  - Database search: scan for regions that may have common ancestry
- **This week: Modeling genomes / exon / CpG island finding**
  - Modeling class of elements, recognizing members of a class
  - Application to gene finding, conservation islands, CpG islands
We have learned how to align sequences to other sequences

- **L2: Sequence alignment**
  - Dynamic programming, duality path ↔ alignment
  - Global / local alignment, general gap penalties
- **L3: Rapid string search**
  - Exact string match, semi-numerical matching
  - Database search: Hashing, BLAST, variations
- **L17: Comparative genomics: evolutionary signatures**
  - Tell me how you evolve, I’ll tell you what you are
  - Identifying conserved elements through evolution
- **L18: Whole-genome assembly/alignment/duplication:**
  - Finding all common substrings within/across species
  - Contigs/scaffolds, string graphs, glocal alignment paths
- **Problem set 1 due next Tues, project planning**
Today: apply these ideas to model DNA sequences

...GTACTCACCAGGTTACAGGATTATGGGTTACAGGTAACCGTT...

- What to do with a completely new piece of DNA
  - Align it to things we know about (database search)
  - Align it to things we don’t know about (assembly)
- Stare at it
  - Non-standard nucleotide composition?
  - Interesting k-mer frequencies?
  - Recurrent patterns?
- Model it
  - Make some hypotheses about it
  - Build a ‘generative model’ to describe it
  - Find sequences of similar type

How do we model DNA sequences?
Modeling biological sequences with HMMs
(a.k.a. What to do with big unlabelled chunks of DNA)

- **Ability to emit** DNA sequences of a certain *type*
  - Not exact alignment to previously known gene
  - Preserving ‘properties’ of *type*, not identical sequence
- **Ability to recognize** DNA sequences of a certain type (state)
  - What (hidden) state is most likely to have generated observations
  - Find set of states and transitions that generated a long sequence
- **Ability to learn** distinguishing characteristics of each state
  - Training our generative models on large datasets
  - Learn to classify unlabelled data
Why Probabilistic Sequence Modeling?

• Biological data is noisy

• Probability provides a calculus for manipulating models

• Not limited to yes/no answers – can provide “degrees of belief”

• Many common computational tools based on probabilistic models

• Our tools:
  – Markov Chains and Hidden Markov Models (HMMs)
Markov Chains and Hidden Markov Models
Andrey Markov (1856-1922)
Predicting tomorrow’s weather

- Markov Chain
- Hidden Markov Model

- What you see is what you get: next state only depends on current state (no memory)
- Hidden state of the world (e.g. storm system) determines emission probabilities
- State transitions governed by a Markov chain

All observed

Transitions

Emissions
HMM nomenclature for this course

- Vector $\mathbf{x} = \text{Sequence of observations}$
- Vector $\mathbf{\pi} = \text{Hidden path (sequence of hidden states)}$
- Transition matrix $\mathbf{A} = a_{kl} = \text{probability of } k \rightarrow l \text{ state transition}$
- Emission vector $\mathbf{E} = e_k(x_i) = \text{prob. of observing } x_i \text{ from state } k$
- Bayes’s rule: Use $P(x_i|\pi_i=k)$ to estimate $P(\pi_i=k|x_i)$

Transitions: $a_{kl} = P(\pi_i=l|\pi_{i-1}=k)$
Transition probability from state $k$ to state $l$

Emissions: $e_k(x_i) = P(x_i|p_i=k)$
Emission probability of symbol $x_i$ from state $k$
Components of a Markov Chain

Definition: A *Markov chain* is a triplet $(Q, \rho, A)$, where:

- $Q$ is a finite set of states. Each state corresponds to a symbol in the alphabet $\Sigma$
- $\rho$ is the initial state probabilities.
- $A$ is the state transition probabilities, denoted by $a_{st}$ for each $s, t$ in $Q$.
- For each $s, t$ in $Q$ the transition probability is: $a_{st} \equiv P(x_i = t | x_{i-1} = s)$

Output: The output of the model is the set of states at each instant time => the set of states are observable

Property: The probability of each symbol $x_i$ depends only on the value of the preceding symbol $x_{i-1}$: $P(x_i | x_{i-1}, ..., x_1) = P(x_i | x_{i-1})$

Formula: The probability of the sequence: $P(x) = P(x_L, x_{L-1}, ..., x_1) = P(x_L | x_{L-1}) P(x_{L-1} | x_{L-2}) ... P(x_2 | x_1) P(x_1)$

Slide credit: Serafim Batzoglou
Components of an HMM (Hidden Markov Model)

Definition: An HMM is a 5-tuple \((Q, V, p, A, E)\), where:

- \(Q\) is a finite set of states, \(|Q|=N\)
- \(V\) is a finite set of observation symbols per state, \(|V|=M\)
- \(p\) is the initial state probabilities.
- \(A\) is the state transition probabilities, denoted by \(a_{st}\) for each \(s, t\) in \(Q\).
  - For each \(s, t\) in \(Q\) the transition probability is: \(a_{st} \equiv P(x_i = t | x_{i-1} = s)\)
- \(E\) is a probability emission matrix, \(e_{sk} \equiv P(v_k at time t | q_t = s)\)

Output: Only emitted symbols are observable by the system but not the underlying random walk between states \(\rightarrow \) “hidden”

Property: Emissions and transitions are dependent on the current state only and not on the past.

Slide credit: Serafim Batzoglou
# The six algorithmic settings for HMMs

## One path

1. Scoring $x$, one path
   \[ P(x, \pi) \]
   Prob of a path, emissions

2. Scoring $x$, all paths
   \[ P(x) = \sum_{\pi} P(x, \pi) \]
   Prob of emissions, over all paths

3. Viterbi decoding
   \[ \pi^* = \arg\max_{\pi} P(x, \pi) \]
   Most likely path

## All paths

4. Posterior decoding
   \[ \pi^\Lambda = \{\pi_i | \pi_i = \arg\max_k \sum_{\pi} P(\pi_i = k | x)\} \]
   Path containing the most likely state at any time point.

## Learning

5. Supervised learning, given $\pi$
   \[ \Lambda^* = \arg\max_{\Lambda} P(x, \pi | \Lambda) \]
   Viterbi training, best path

6. Unsupervised learning
   \[ \Lambda^* = \arg\max_{\Lambda} \sum_{\pi} P(x, \pi | \Lambda) \]
   Baum-Welch training, over all paths
Examples of HMMs

The dishonest casino
The dishonest genome
... and many more
A casino has two dice:

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

- **Loaded die**
  \[ P(1) = P(2) = P(3) = P(4) = P(5) = 1/10 \]
  \[ P(6) = 1/2 \]

Casino player switches between fair and loaded die on average 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
The dishonest casino model

- **Hidden (model)**
  - Fair
  - Loaded

- **Observed (world)**
  - \( P(1|\text{Fair}) = 1/6 \)
  - \( P(2|\text{Fair}) = 1/6 \)
  - \( P(3|\text{Fair}) = 1/6 \)
  - \( P(4|\text{Fair}) = 1/6 \)
  - \( P(5|\text{Fair}) = 1/6 \)
  - \( P(6|\text{Fair}) = 1/6 \)
  - \( P(1|\text{L}) = 1/10 \)
  - \( P(2|\text{L}) = 1/10 \)
  - \( P(3|\text{L}) = 1/10 \)
  - \( P(4|\text{L}) = 1/10 \)
  - \( P(5|\text{L}) = 1/10 \)
  - \( P(6|\text{L}) = 1/2 \)

Slide credit: Serafim Batzoglou
The dishonest genome model

Virus

"Self"

P(A|Virus) = 1/6
P(T|Virus) = 1/6
P(C|Virus) = 1/3
P(G|Virus) = 1/3

P(A|Self) = 1/4
P(T|Self) = 1/4
P(C|Self) = 1/4
P(G|Self) = 1/4

Observed (world)

Hidden (model)

Slide credit: Serafim Batzoglou
## Examples of HMMs for genome annotation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topology / Transitions</strong></td>
<td>2 states, different nucleotide composition</td>
<td>2 states, different conservation levels</td>
<td>2 states, different trinucleotide composition</td>
<td>2 states, different evolutionary signatures</td>
<td>~20 states, different composition/conservation, specific structure</td>
<td>40 states, different chromatin mark combinations</td>
</tr>
<tr>
<td><strong>Hidden States / Annotation</strong></td>
<td>GC-rich / AT-rich</td>
<td>Conserved / non-conserved</td>
<td>Coding exon / non-coding (intron or intergenic)</td>
<td>Coding exon / non-coding (intron or intergenic)</td>
<td>First/last/middle coding exon, UTRs, intron1/2/3, intergenic, <em>(+/- strand)</em></td>
<td>Enhancer / promoter / transcribed / repressed / repetitive</td>
</tr>
<tr>
<td><strong>Emissions / Observations</strong></td>
<td>Nucleotides</td>
<td>Level of conservation</td>
<td>Triplets of nucleotides</td>
<td>Nucleotide triplets, conservation levels</td>
<td>Codons, nucleotides, splice sites, start/stop codons</td>
<td>Vector of chromatin mark frequencies</td>
</tr>
</tbody>
</table>
Running the model: Probability of a sequence

What is the joint probability of observing $x$ and a specific path $\pi$:

$$\pi = \text{Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair, Fair}$$

and rolls

$$x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4$$

Joined probability $P(x, \pi) = P(x|\pi)P(\pi) = P(\text{emissions}|\text{path}) \times P(\text{path})$

$$p = \frac{1}{2} \times P(1 | \text{Fair}) \times P(\text{Fair}_{i+1} | \text{Fair}_i) \times P(2 | \text{Fair}) \times P(\text{Fair} | \text{Fair}) \times \ldots \times P(4 | \text{Fair})$$

$$= \frac{1}{2} \times (1/6)^{10} \times (0.95)^9$$

$$= 5.2 \times 10^{-9}$$

Why is $p$ so small?

Slide credit: Serafim Batzoglou
Running the model: Probability of a sequence

What is the likelihood of

\[ \pi = \text{Load, Load, Load, Load, Load, Load, Load, Load, Load, Load, Loaded} \]

and rolls

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

emission  transition  emission  transition  emission

\[ p = \frac{1}{2} \times P(1 \mid \text{Load}) \times P(\text{Load}_{i+1} \mid \text{Load}_i) \times P(2 \mid \text{Load}) \times P(\text{Load} \mid \text{Load}) \times \ldots \times P(4 \mid \text{Fair}) \]

\[ = \frac{1}{2} \times \left(\frac{1}{10}\right)^8 \times \left(\frac{1}{2}\right)^2 \times (0.95)^9 \]

\[ = 7.9 \times 10^{-10} \]

Compare the two!
Comparing the two paths

Two sequence paths:
- \( P( x, \text{all-Fair} ) = 5.2 \times 10^{-9} \) (very small)
- \( P( x, \text{all-Loaded} ) = 7.9 \times 10^{-10} \) (very very small)

Likelihood ratio:
- \( P( x, \text{all-Fair} ) \) is 6.59 times more likely than \( P( x, \text{all-Loaded} ) \)

It is 6.59 times more likely that the die is fair all the way, than loaded all the way.

Slide credit: Serafim Batzoglou
What about partial runs and die switching

What is the likelihood of

$$\pi = \text{Fair, Fair, Fair, Fair, Load, Load, Load, Load, Fair, Fair}$$

and rolls

$$x = 1, 2, 1, 5, 6, 2, 1, 6, 2, 4$$

$$p = \frac{1}{2} \times P(1 \mid \text{Fair}) P(\text{Fair}_{i+1} \mid \text{Fair}_i) P(2 \mid \text{Fair}) P(\text{Fair} \mid \text{Fair}) \ldots P(4 \mid \text{Fair})$$

$$= \frac{1}{2} \times (1/10)^2 \times (1/2)^2 \times (1/6)^5 \times (0.95)^7 \times (0.05)^2$$

$$= 2.8 \times 10^{-10}$$

Much less likely, due to high cost of transitions
Let the sequence of rolls be:
\[ x = 1, 6, 6, 5, 6, 2, 6, 6, 3, 6 \]

Now, what is the likelihood \( \pi = F, F, \ldots, F \)?
\[
\frac{1}{2} \times \left(\frac{1}{6}\right)^{10} \times (0.95)^9 = 0.5 \times 10^{-9}, \text{ same as before}
\]

What is the likelihood \( \pi = L, L, \ldots, L \)?
\[
\frac{1}{2} \times \left(\frac{1}{10}\right)^{4} \times \left(\frac{1}{2}\right)^{6} \times (0.95)^9 = 0.5 \times 10^{-7}
\]

So, it is 100 times more likely the die is loaded
<table>
<thead>
<tr>
<th>Scoring</th>
<th>Decoding</th>
<th>Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Scoring ( x ), one path</td>
<td>3. Viterbi decoding</td>
<td>5. Supervised learning, given ( \pi )</td>
</tr>
<tr>
<td>( P(x, \pi) )</td>
<td>( \pi^* = \arg\max_\pi P(x, \pi) )</td>
<td>( \Lambda^* = \arg\max_\Lambda P(x, \pi</td>
</tr>
<tr>
<td>Prob of a path, emissions</td>
<td>Most likely path</td>
<td>Viterbi training, best path</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Scoring ( x ), all paths</td>
<td>4. Posterior decoding</td>
<td>6. Unsupervised learning</td>
</tr>
<tr>
<td>( P(x) = \sum_\pi P(x, \pi) )</td>
<td>( \pi^\wedge = { \pi_i \mid \pi_i = \arg\max_k \sum_\pi P(\pi_i = k</td>
<td>x) } )</td>
</tr>
<tr>
<td>Prob of emissions, over all paths</td>
<td>Path containing the most likely state at any time point.</td>
<td>Baum-Welch training, over all paths</td>
</tr>
</tbody>
</table>

**The six algorithmic settings for HMMs**

- **One path**
  - Scoring
    - \( P(x, \pi) \)
      - Prob of a path, emissions
  - Decoding
    - Viterbi decoding
      - \( \pi^* = \arg\max_\pi P(x, \pi) \)
      - Most likely path
  - Learning
    - Supervised learning
      - \( \Lambda^* = \arg\max_\Lambda P(x, \pi | \Lambda) \)
    - Viterbi training, best path

- **All paths**
  - Scoring
    - \( P(x) = \sum_\pi P(x, \pi) \)
      - Prob of emissions, over all paths
  - Decoding
    - Posterior decoding
      - \( \pi^\wedge = \{ \pi_i \mid \pi_i = \arg\max_k \sum_\pi P(\pi_i = k | x) \} \)
      - Path containing the most likely state at any time point.
  - Learning
    - Unsupervised learning
      - \( \Lambda^* = \arg\max_\Lambda \sum_\pi P(x, \pi | \Lambda) \)
      - Baum-Welch training, over all paths
3. DECODING:
What was the sequence of hidden states?

Given: Model parameters $e_i(\cdot), a_{ij}$
Given: Sequence of emissions $x$

Find: Sequence of hidden states $\pi$
Finding the optimal path

• We can now evaluate any path through hidden states, given the emitted sequences

• How do we find the best path?

• Optimal substructure! Best path through a given state is:
  – Best path to previous state
  – Best transition from previous state to this state
  – Best path to the end state

⇒ Viterbi algorithm
  – Define $V_k(i) = \text{Probability of the most likely path through state } \pi_i=k$
  – Compute $V_k(i+1)$ as a function of $\max_{k'} \{ V_{k'}(i) \}$

  – $V_k(i+1) = e_k(x_{i+1}) * \max_j a_{jk} V_j(i)$

⇒ Dynamic Programming
Andrew J. Viterbi, Massachusetts Beta '57, as a new teacher in 1963.
Finding the most likely path

- Find path $\pi^*$ that maximizes total joint probability $P[ x, \pi ]$

- $P(x, \pi) = a_{0\pi_1}^* \prod_i e_{\pi_i}(x_i) \times a_{\pi_i\pi_{i+1}}$

Slide credit: Serafim Batzoglou
Calculate maximum $P(x, \pi)$ recursively

- Assume we know $V_j$ for the previous time step $(i-1)$

- Calculate $V_k(i) = \max_j \left( e_k(x_i) \times \prod_{j=1}^{i-1} a_{jk} \times V_j(i-1) \right)$

Slide credit: Serafim Batzoglou
The Viterbi Algorithm

Input: $x = x_1 \ldots x_N$

**Initialization:**
\[ V_0(0) = 1, \quad V_k(0) = 0, \text{ for all } k > 0 \]

**Iteration:**
\[ V_k(i) = e^x_i \times \max_j a_{jk} \cdot V_j(i-1) \]

**Termination:**
\[ P(x, \pi^*) = \max_k V_k(N) \]

**Traceback:**
Follow max pointers back
Similar to aligning states to seq

**In practice:**
Use log scores for computation

**Running time and space:**
Time: $O(K^2N)$
Space: $O(KN)$

---

Slide credit: Serafim Batzoglou
<table>
<thead>
<tr>
<th>Scoring</th>
<th>Decoding</th>
<th>Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Scoring $x$, one path $P(x, \pi)$</td>
<td>3. Viterbi decoding $\pi^* = \arg\max_{\pi} P(x, \pi)$</td>
<td>5. Supervised learning, given $\pi$ $\Lambda^* = \arg\max_{\Lambda} P(x, \pi</td>
</tr>
<tr>
<td>Prob of a path, emissions</td>
<td>Most likely path</td>
<td>6. Unsupervised learning. $\Lambda^* = \arg\max_{\Lambda} \sum_\pi P(x, \pi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Viterbi training, best path</td>
</tr>
<tr>
<td>2. Scoring $x$, all paths $P(x) = \sum_\pi P(x, \pi)$</td>
<td>4. Posterior decoding $\pi^\land = {\pi_i</td>
<td>\pi_i = \arg\max_k \sum_\pi P(\pi_i=k</td>
</tr>
<tr>
<td>Prob of emissions, over all paths</td>
<td>Path containing the most likely state at any time point.</td>
<td>Baum-Welch training, over all paths</td>
</tr>
</tbody>
</table>

The six algorithmic settings for HMMs:

1. **One path**
   - Scoring: $P(x, \pi)$
   - Decoding: $\pi^* = \arg\max_{\pi} P(x, \pi)$
   - Learning: $\Lambda^* = \arg\max_{\Lambda} P(x, \pi | \Lambda)$

2. **All paths**
   - Scoring: $P(x) = \sum_\pi P(x, \pi)$
   - Decoding: $\pi^\land = \{\pi_i | \pi_i = \arg\max_k \sum_\pi P(\pi_i=k | x)\}$
   - Learning: $\Lambda^* = \arg\max_{\Lambda} \sum_\pi P(x, \pi | \Lambda)$

### Notes
- In supervised learning, the model parameters are given.
- Unsupervised learning methods estimate the model parameters without explicit labeled data.
- Viterbi training is a specific case of supervised learning.
- Baum-Welch training is a specific case of unsupervised learning.
2. EVALUATION
(how well does our model capture the world)

Given: Model parameters $e_i(\cdot)$, $a_{ij}$

Given: Sequence of emissions $x$

Find: $P(x|M)$, summed over all possible paths $\pi$
Simple: Given the model, generate some sequence x

Given a HMM, we can generate a sequence of length n as follows:

1. Start at state $\pi_1$ according to prob $a_{0\pi_1}$
2. Emit letter $x_1$ according to prob $e_{\pi_1}(x_1)$
3. Go to state $\pi_2$ according to prob $a_{\pi_1\pi_2}$
4. ... until emitting $x_n$

We have some sequence $x$ that can be emitted by $p$. Can calculate its likelihood. However, in general, many different paths may emit this same sequence $x$. How do we find the total probability of generating a given $x$, over any path?

Slide credit: Serafim Batzoglou
Complex: Given x, was it generated by the model?

Given a sequence x,
What is the probability that x was generated by the model (using any path)?

\[ P(x) = \sum_\pi P(x, \pi) = \sum_\pi P(x|\pi) P(\pi) \]

- (weighted average of conditional probability, summed over all paths, weighted by each path’s probability)

- Challenge: exponential number of paths
Calculate probability of emission over all paths

• Each path has associated probability
  – Some paths are likely, others unlikely: sum them all up
  → Return total probability that emissions are observed, summed over all paths
  – Viterbi path is the most likely one
    • How much ‘probability mass’ does it contain?

• (cheap) alternative:
  – Calculate probability over maximum (Viterbi) path $\pi^*$
  – Good approximation if Viterbi has highest density
  – BUT: incorrect

• (real) solution
  – Calculate the exact sum iteratively
    • $P(x) = \sum_{\pi} P(x,\pi)$
  – Can use dynamic programming
The Forward Algorithm – derivation

Define the forward probability:

\[ f_l(i) = P(x_1 \ldots x_i, \pi_i = l) \]

\[ = \sum_{\pi_1 \ldots \pi_{i-1}} P(x_1 \ldots x_{i-1}, \pi_1, \ldots, \pi_{i-2}, \pi_{i-1}, \pi_i = l) \cdot e_l(x_i) \]

\[ = \sum_k \sum_{\pi_1 \ldots \pi_{i-2}} P(x_1 \ldots x_{i-1}, \pi_1, \ldots, \pi_{i-2}, \pi_{i-1} = k) \cdot a_{kl} \cdot e_l(x_i) \]

\[ = \sum_k f_k(i-1) \cdot a_{kl} \cdot e_l(x_i) \]

\[ = e_l(x_i) \sum_k f_k(i-1) \cdot a_{kl} \]
Calculate total probability $\Sigma_{\pi} P(x, \pi)$ recursively

- Assume we know $f_j$ for the previous time step (i-1)

- Calculate $f_k(i) = e_k(x_i) * \sum_j (f_j(i-1) \times a_{jk})$

Slide credit: Serafim Batzoglou
The Forward Algorithm

Input: $x = x_1 \ldots x_N$

**Initialization:**
- $f_0(0) = 1$, $f_k(0) = 0$, for all $k > 0$

**Iteration:**
- $f_k(i) = e^{K(x_i)} \times \sum_j a_{jk} f_j(i-1)$

**Termination:**
- $P(x, \pi^*) = \sum_k f_k(N)$

**In practice:**
- Sum of log scores is difficult
  - approximate $\exp(1+p+q)$
  - scaling of probabilities

**Running time and space:**
- Time: $O(K^2N)$
- Space: $O(KN)$

Slide credit: Serafim Batzoglou
<table>
<thead>
<tr>
<th><strong>The six algorithmic settings for HMMs</strong></th>
<th><strong>One path</strong></th>
<th><strong>All paths</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scoring</strong> &lt;br&gt;1. Scoring $x$, one path</td>
<td>$P(x, \pi)$</td>
<td>$P(x) = \sum_\pi P(x, \pi)$</td>
</tr>
<tr>
<td></td>
<td>Prob of a path, emissions</td>
<td>Prob of emissions, over all paths</td>
</tr>
<tr>
<td><strong>Decoding</strong> &lt;br&gt;3. Viterbi decoding</td>
<td>$\pi^* = \arg\max_\pi P(x, \pi)$</td>
<td>$\pi^\wedge = {\pi_i</td>
</tr>
<tr>
<td></td>
<td>Most likely path</td>
<td>Path containing the most likely state at any time point.</td>
</tr>
<tr>
<td><strong>Learning</strong> &lt;br&gt;5. Supervised learning, given $\pi$</td>
<td>$\Lambda^* = \arg\max_\lambda P(x, \pi</td>
<td>\Lambda)$</td>
</tr>
<tr>
<td></td>
<td>$\Lambda^* = \arg\max_\lambda \max_\pi P(x, \pi</td>
<td>\Lambda)$</td>
</tr>
<tr>
<td></td>
<td>Viterbi training, best path</td>
<td></td>
</tr>
</tbody>
</table>
### Examples of HMMs for genome annotation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topology / Transitions</strong></td>
<td>2 states, different nucleotide composition</td>
<td>2 states, different conservation levels</td>
<td>2 states, different trinucleotide composition</td>
<td>2 states, different evolutionary signatures</td>
<td>~20 states, different composition/conservation, specific structure</td>
<td>40 states, different chromatin mark combination</td>
</tr>
<tr>
<td><strong>Hidden States / Annotation</strong></td>
<td>GC-rich / AT-rich</td>
<td>Conserved / non-conserved</td>
<td>Coding exon / non-coding (intron or intergenic)</td>
<td>Coding exon / non-coding (intron or intergenic)</td>
<td>First/last/middle coding exon, UTRs, intron 1/2/3, intergenic, <em>(+/- strand)</em></td>
<td>Enhancer / promoter / transcribed / repressed / repetitive</td>
</tr>
<tr>
<td><strong>Emissions / Observations</strong></td>
<td>Nucleotides</td>
<td>Level of conservation</td>
<td>Triplets of nucleotides</td>
<td>64x64 matrix of codon substitution frequencies</td>
<td>Codons, nucleotides, splice sites, start/stop codons</td>
<td>Vector of chromatin mark frequencies</td>
</tr>
</tbody>
</table>
What have we learned?

- **Modeling sequential data**
  - Recognize a *type* of sequence, genomic, oral, verbal, visual, etc…

- **Definitions**
  - Markov Chains
  - Hidden Markov Models (HMMs)

- **Examples of HMMs**
  - Recognizing GC-rich regions, preferentially-conserved elements, coding exons, protein-coding gene structures, chromatin states

- **Our first computations**
  - Running the model: know model $\rightarrow$ generate sequence of a ‘type’
  - Evaluation: know model, emissions, states $\rightarrow$ $p$?
  - Viterbi: know model, emissions $\rightarrow$ find optimal path
  - Forward: know model, emissions $\rightarrow$ total $p$ over all paths

- **Next time:**
  - Posterior decoding
  - Supervised learning
  - Unsupervised learning: Baum-Welch, Viterbi training