CSE 584A Class 26

Jeremy Buhler

April 27, 2016

1 Adding a Phylogenetic Tree

What if you know something about your sequences' evolutionary history?

- Knowing the history means being given a *phylogenetic tree* on input seqs
- Example: species tree for human, mouse, rat

- Tree tells you which sequences diverged more recently
- Implies existence of (unknown) *ancestral seqs* at internal nodes
- Knowing correct ancestral seqs would make alignment easier – instead of full multiple alignment, align each seq to its immediate ancestor to estimate correspondence on just one branch
- Build up full multiple alignment of (modern) leaves progressively
- *Example*:

Problem: ancestral seqs are (of course) unknown!

- We could guess them... but how?
- *Idea*: guess *parsimoniously*

- A pairwise alignment between seqs at endpts of an edge postulates a set of changes occuring along that edge
- Choose the ancestral seqs to minimize total number of changes inferred!

More formally...

- Let $D(s, t)$ be an edit distance on strings s, t
- Given a tree τ with seqs $s_1 \dots s_n$ at its leaves
- A *labeling* of tree τ is a mapping $\lambda : V(\tau) \to \Sigma^*$ that assigns strings to nodes v of τ
- Labeling of leaves follows input; i.e., if s_i belongs at leaf ℓ_i, then $\lambda(\ell_i) = s_i$.
- Labeling of internal nodes can be any strings.
- Let the distance of a labeling $D(\lambda)$ be the total distance of optimal alignments between the endpts of its edges:

$$D(\lambda) = \sum_{(u,v) \in E(\tau)} D(\lambda(u), \lambda(v))$$

- **Problem**: given τ and its leaf seqs, find a labeling λ^* for ancestors that minimizes $D(\lambda^*)$

Call this problem "Minimum-Distance Ancestor Reconstruction (MDAR)"

2 Approach to MDAR Problem

Bad News: the MDAR problem is NP-hard.

- Once again, do we give up, or do we try to deal?
- Once again, assume that the seq edit distance D is a metric.
- Will show that a simple heuristic gets us pretty close to optimal (min-distance) labeling

Here's the heuristic:

- Consider a tree τ with seqs $s_1 \dots s_n$ labeling its leaves.
- **Defn**: a *lifted labeling* of τ is a labeling λ in which, for each non-leaf node v, there exists a child w of v such that $\lambda(v) = \lambda(w)$.
- (I.e., we get label of each node by "lifting" label from one of its children)

- For a tree with two leaves, lifted labeling of single ancestor is clearly optimal!

- Indeed, for any hypothesized ancestral seq α, $D(s_i, \alpha) + D(\alpha, s_j) \geq D(s_i, s_j)$ by triangle ineq.
- In particular, setting $\alpha = s_i$ gives total distance $D(s_i, s_i) + D(s_i, s_j) = 0 + D(s_i, s_j)$, which is the optimum.
- Unfortunately, optimality of lifting is not guaranteed for more general tree structures on 3 or more seqs. However...

3 [Non-Constructive!] Proof of Approximation Ratio

Claim: Consider a tree τ with seqs $s_1 \dots s_n$ at leaves. Let $\lambda^*(\tau)$ be an optimal labeling of τ. Then there exists a lifted labeling $\lambda^L(\tau)$ such that

$$\frac{D(\lambda^L(\tau))}{D(\lambda^*(\tau))} \leq 2.$$

- **Pf**: two parts. First, exhibit a particular lifted labeling λ^L. Then, show that it achieves approximation ratio.
- **Not constructive!** Will assume that we *know* λ^* for purposes of constructing λ^L.
- (We'll construct a labeling later that will be at least as good as the one exhibited in the proof.)
- To derive λ^L from λ^*, proceed bottom-up in tree
- Consider an internal node v with children $w_1 \dots w_k$.
- Let $\lambda^L(w_1) \dots \lambda^L(w_k)$ be strings labeling $w_1 \dots w_k$ in lifted labeling.
- Choose i for which $D(\lambda^*(v), \lambda^L(w_i))$ is minimal.
- Set $\lambda^L(v) \leftarrow \lambda^L(w_i)$.

Why is λ^L a good labeling?

- First, consider local effect of setting $\lambda^L(v) \leftarrow \lambda^L(w_i)$ above.

- We have that

$$\begin{aligned} D(\lambda^L(w_j), \lambda^L(v)) &= D(\lambda^L(w_j), \lambda^L(w_i)) \\ &\leq D(\lambda^L(w_j), \lambda^*(v)) + D(\lambda^*(v), \lambda^L(w_i)) \\ &\leq 2D(\lambda^L(w_j), \lambda^*(v)). \end{aligned}$$

  Second step follows from triangle inequality; third follows because w_i is closer to v than is w_j.

  But what does this have to do with total distance $D(\lambda^*)$ over all edges in the tree?

We will subdivide the tree into regions for which distance is easily computed.

- For each leaf s_i, lifted labeling assigns it to a *path* of zero or more internal nodes.
- (Path starts at leaf labeled with s_i and continues until it "loses" a lifting.)

- Let ℓ_i be leaf labeled with s_i, and let (u_i, v) be edge on which s_i finally loses to some other string, which becomes $\lambda^L(v)$.
- Let path $P(s_i) = \{\ell_i \dots v\}$ in tree be path of all edges for which s_i wins, plus final edge for which it loses.
- Observe that we can partition tree into paths $P(s_1) \dots P(s_n)$, since each input string is lifted to some point and then stops.

Let's evaluate distance along each path in lifted vs optimal labeling.

- Let $D^L(P)$ and $D^*(P)$ be total distance on path P under labelings λ^L and λ^*, respectively.

- In lifted labeling, both endpts of each edge on path $P(s_i)$ are labeled with s_i, except last edge (u_i, v).
- Total distance of all edges on $P(s_i)$ *except* last is 0!
- Hence, $D^L(P(s_i)) = D(s_i, \lambda^L(v))$.
- Moreover, by previous argument, we have that

$$D(s_i, \lambda^L(v)) \leq 2D(s_i, \lambda^*(v)).$$

- Now consider path $P(s_i)$ under *original* labeling λ^*.
- This path starts with s_i and ends with $\lambda^*(v)$, perhaps with other seqs in between.

- By triangle inequality on D, $D^*(P(s_i)) \geq D(s_i, \lambda^*(v))$.
- Conclude that

$$\begin{aligned} \frac{D^L(P(s_i))}{D^*(P(s_i))} &\leq \frac{D(s_i, \lambda^L(v))}{D(s_i, \lambda^*(v))} \\ &\leq \frac{2D(s_i, \lambda^*(v))}{D(s_i, \lambda^*(v))} \\ &= 2. \end{aligned}$$

- Because paths for each input string partition tree, conclude that

$$\begin{aligned} \frac{D(\lambda^L)}{D(\lambda^*)} &= \frac{\sum_i D^L(P(s_i))}{\sum_i D^*(P(s_i))} \\ &\leq \frac{\sum_i 2D^*(P(s_i))}{\sum_i D^*(P(s_i))} \\ &= 2. \end{aligned}$$

QED

4 Computing a Good Tree Labeling

Problem: previous proof showed that there *exists* a good lifted labeling on tree, but it was not constructive because it assumed knowledge of optimal solution.

- Will show how to find a labeling *at least as good* as best lifted labeling.
- Given inputs $s_1 \ldots s_n$, will find best labeling where each ancestor is labeled with one of $s_1 \ldots s_n$.
- (Algo considers a larger set of labelings that includes all lifted ones.)

- Dynamic programming approach; runs bottom-up on tree.
- For any tree τ rooted at node v, let $C(v, s)$ be total cost (i.e. total distance) of best labeling of tree among all those that set $\lambda(v) = s$.
- If leaves ℓ_i have fixed labels s_i, initialization is

$$C(\ell_i, s_j) = \begin{cases} 0 & \text{if } i = j \\ \infty & \text{otherwise.} \end{cases}$$

- For other nodes, labels are *not* fixed.
- In general case of node v with children $w_1 \dots w_m$,

$$C(v, s_j) = \sum_{k=1}^{m} \left[\min_{i=1}^{n} \left(D(s_j, s_i) + C(w_k, s_i) \right) \right].$$

- By computing this recurrence, can find optimal string from S for root.
- To derive labels for remaining nodes, go top-down, finding argmins for each min.

What does this cost?

- Assume n input seqs have common length m, and max degree of any node in tree is d.
- Assume we precompute distances $D(s_i, s_j)$, for $1 \le i, j \le n$.
- Cost of precomputation is $\Theta(n^2 m^2)$.
- At each node of tree, DP algo must consider each of n possible input strings.
- If node has at most d children, must minimize (independently) over n possible labels for each of d children, for each of n node labels.
- Total cost per node is $O(dn^2)$.
- Tree has $O(n)$ total nodes, since it has n leaves, and each node has degree ≥ 2.
- Conclude that total cost is $O(dn^3 + n^2 m^2)$.

This algo, restricted to single characters instead of arbitrary strings, is the *Fitch parsimony algorithm*. It and its relatives are basic tools in phylogeny.