Today's Lecture

- Baum-Welch training
- More HMM examples

Baum-Welch training

- Special case of EM (‘expectation-maximization’) algorithm
- like Viterbi training, but
  - uses *all* paths, each weighted by its probability

    rather than just highest probability path.
- sometimes give significantly better results than Viterbi
  - e.g. for PFAM

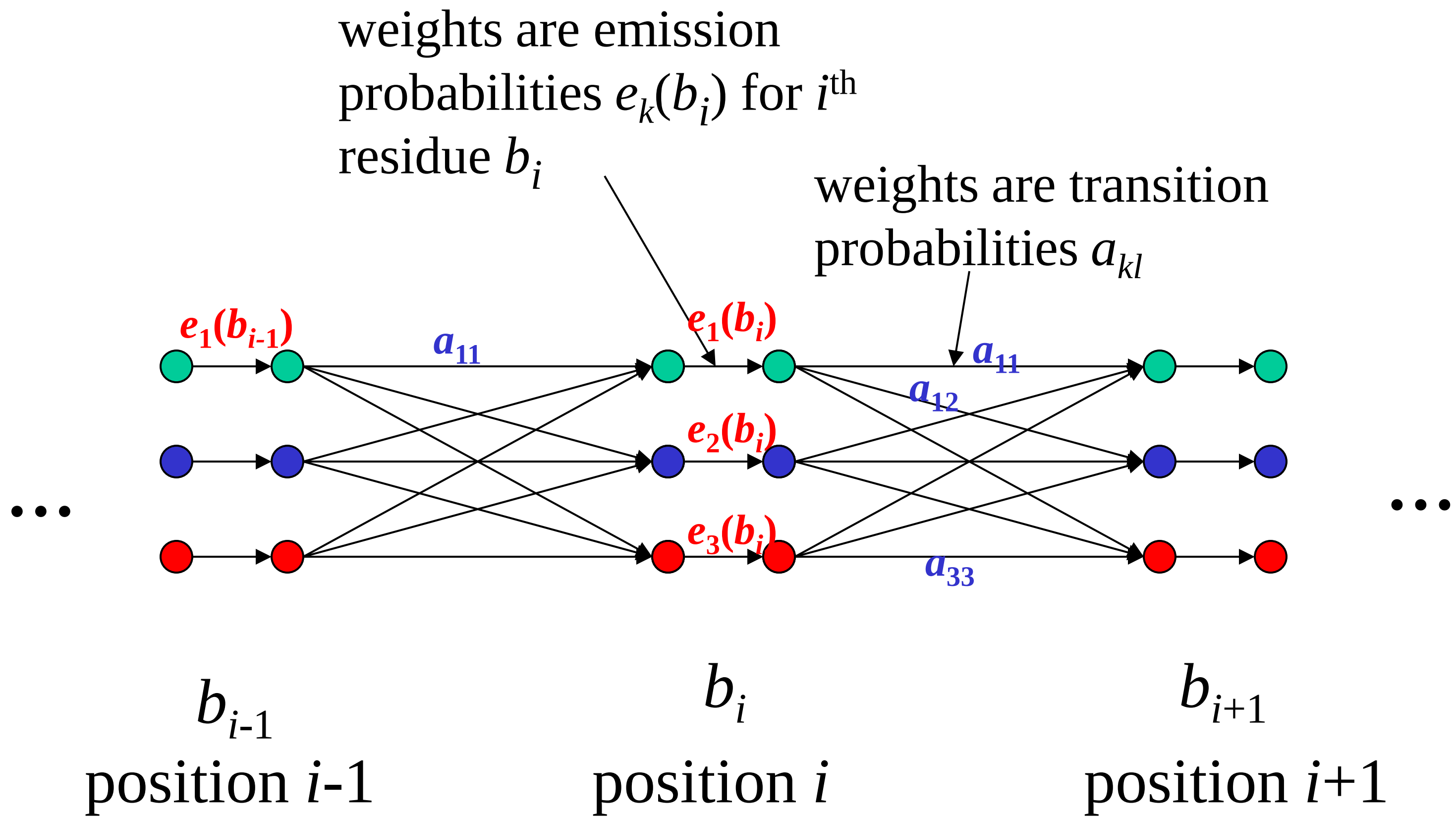
Implementing Baum-Welch

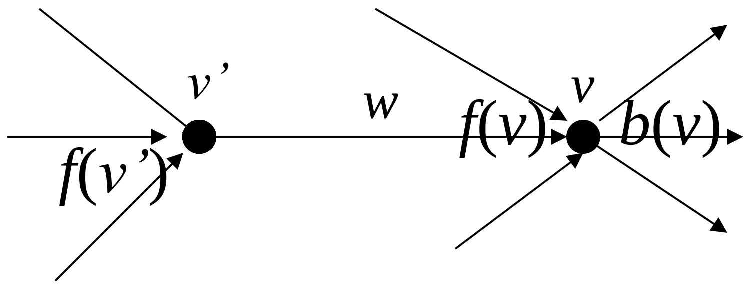
– An edge in the WDAG contributes *fractional* (or *weighted*) *counts* given by its posterior probability:

– (*): $(\sum_{\text{all paths } p \text{ through edge } e} \text{weight}(p)) / (\sum_{\text{all paths } p} \text{weight}(p))$

(Fractional counts are computed using forward-backward algorithm)

WDAG for 3-state HMM, length n sequence





$f(v)b(v)$ = sum of the path weights of all paths *through v*.

$f(v')wb(v)$ = sum of the path weights of all paths *through the edge (v',v)*

– Compute new param estimates

- $e_k(b)$^ = (frac. # times symbol b emitted by state k) / (frac. # times state k occurs)
- a_{kl} ^ = (frac. # times state k followed by state l) / (frac. # times state k occurs)
  - (In denom,, omit frac counts at last position of sequence)

where "frac. # times" is given by (*) for appropriate edge type (emission or transition)

- New Baum-Welch parameter estimates have higher likelihood
  - general property of EM algorithm
  - not true in general for Viterbi training

- Iterate: get series of estimates converging to a *local* maximum on likelihood surface

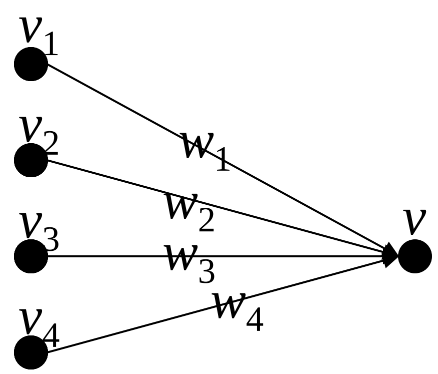
Search of parameter space

- ML estimates correspond by definition to *global* maximum;
- but there may be many *local* maxima, and EM or Viterbi search can get "trapped" in one
- remedies:
  - Consider multiple starts (multiple choices for starting parameters)
  - use "reasonable values" to start search (e.g. unlikely transitions should be given small initial probabilities)

- Allow search to “jump” out of local maxima:
  - Add “noise” to counts at each iteration; gradually reduce the amount of noise
  - Use Viterbi-like training, but
    - sample paths probabilistically
      - (in retracing Viterbi path, choose edge at random according to its prob, rather than taking highest prob parent);
    - use “temperature” T to adjust probabilities;
      - initially with large T making all probs approximately equal;
      - then gradually reduce T
    - similar to Gibbs sampler

Probabilistic Viterbi Backtracking

reset all weights w to $w^{1/T}$. For large T (>> 1), this makes distinct w's relatively close; for small T (<< 1), relatively far apart



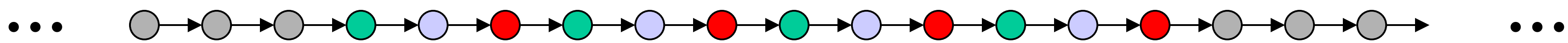
choose parent v_i with probability $w_i\, f(v_i) / f(v)$. For large T, all parents almost equally likely to be chosen; for small T, strongly favor largest (max) $w_i\, f(v_i)$

given choice of paths, re-estimate weights; iterate

HMM Examples (cont'd)

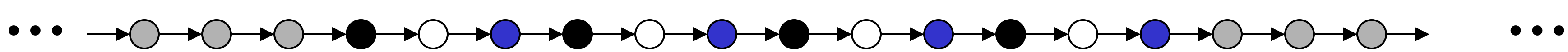
- Simple 7-state prokaryote genome model:
  - 1 state for intergenic regions
  - 3 states for codon positions in top-strand genes
  - 3 for codon positions in bottom-strand genes
- more complex models including sites (with states for each position in site) –
  - promoter elements
  - Shine-Dalgarno (translation start site)
  - (in eukaryotes) splice sites, polyadenylation sites etc.

7-state model for prokaryote genomes



- intergenic
- first codon position – top strand coding sequence
- second codon position – top strand coding sequence
- third codon position – top strand coding sequence
- first codon position – bottom strand coding sequence
- second codon position – bottom strand coding sequence
- third codon position – bottom strand coding sequence

a (very short!) ‘bottom-strand’ gene, in a different region of the genome:



- N.B. the emitted symbols are always ***top strand*** nucleotides!

Other HMM examples (see Durbin *et al.*)

- protein families (like site models – but important to allow insertions & deletions)
- Pair HMMs
- protein structure (symbols emitted are structural elements)