Cost analysis

Luca Blum

May 2020

1 Introduction

We define the cost measure as:

$C(n) = C_{add} * N_{add} + C_{mult} * N_{mult} + C_{div} * N_{div} + C_{log} * N_{log} + C_{cmp\_lt} * N_{cmp\_lt} + C_{cmp\_eq} * N_{cmp\_eq}$

The timing consists following steps:

- forward step
- backward step
- update step
- finishing criteria

We evaluated the cost for the default baum-welch algorithm and for a stable baum-welch algorithm. We define N as the number of hidden states, T as the number of time steps or equivalent as the number of observations made and K as the number of observables.

2 Default baum-welch algorithm

2.1 forward step

- computing $\alpha[0]$

$$C_1(n) = N * C_{mult}$$

- for T timesteps:
  - compute $\alpha[s, t]$

$$C_2(n) = N * N * C_{add} + (N * N + N) * C_{mult}$$

Overall

$$C_{forward}(n) = T * N * N * C_{add} + (N + T * N * N + T * N) * C_{mult}$$

2.2 backward step

- for T-1 timesteps:
  - compute $\beta[s,t]$

$$C_1(n) = N * N * C_{add} + 2 * N * N * C_{mult}$$

Overall

$$C_{backward}(n) = (T-1) * N * N * C_{add} + (T-1) * 2 * N * N * C_{mult}$$

2.3 update step

- compute evidence

$$C_1(n) = T * N * C_{add} + T * N * C_{mult}$$

- compute gamma

$$C_2(n) = T * N * C_{mult} + T * N * C_{div}$$

- compute xi

$$C_3(n) = 3 * T * N * N * C_{mult} + T * N * N * C_{div}$$

- compute gamma sum denominator

$$C_4(n) = T * N * N * C_{add}$$

- new transition matrix

$$C_5(n) = N * N * C_{div}$$

- compute gamma sum numerator

$$C_6(n) = T * K * N * C_{add} + T * K * N * C_{comp\_eq}$$

- new emission matrix

$$C_7(n) = N * K * C_{div}$$

Overall

$$\begin{aligned} C_{update} =& (T * N + T * N * N + T * K * N) * C_{add} + (T * N + T * N + 3 * T * N * N) * C_{mult} \\ &+ (T * N + T * N * N + N * N + N * K) * C_{div} + T * K * N * C_{comp\_eq} \\ =& (T * N + T * N * N + T * K * N) * C_{add} + (2 * T * N + 3 * T * N * N) * C_{mult} \\ &+ (T * N + T * N * N + N * N + N * K) * C_{div} + T * K * N * C_{comp\_eq} \end{aligned}$$

2.4 finishing criteria

- for every timestep
$$C_1(n) = T * C_{add}$$
- return value
$$C_2(n) = 1 * C_{add} + 1 * C_{cmp\_lt}$$

Overall
$$C_{finished}(n) = (T + 1) * C_{add} + 1 * C_{cmp\_lt}$$

2.5 Total

$$\begin{aligned}
C_{stable}(n) =& C_{forward}(n) + C_{backward}(n) + C_{update}(n) + C_{finished}(n) \\
C_{stable}(n) =& T * N * N * C_{add} + (N + T * N * N + T * N) * C_{mult} \\
&+ (T - 1) * N * N * C_{add} + (T - 1) * 2 * N * N * C_{mult} \\
&+ (T * N + T * N * N + T * K * N) * C_{add} + (2 * T * N + 3 * T * N * N) * C_{mult} \\
&+ (T * N + T * N * N + N * N + N * K) * C_{div} + T * K * N * C_{comp_eq} \\
&+ (T + 1) * C_{add} + 1 * C_{cmp_lt} \\
C_{stable}(n) =& (T * N * N + (T - 1) * N * N + T * N + T * N * N + T * K * N + (T + 1)) * C_{add} \\
&+ (N + T * N * N + T * N + (T - 1) * 2 * N * N + 2 * T * N + 3 * T * N * N) * C_{mult} \\
&+ (T * N + T * N * N + N * N + N * K) * C_{div} \\
&+ T * K * N * C_{comp_eq} \\
&+ 1 * C_{cmp_lt} \\
=& (2 * T * N * N + (T - 1) * N * N + T * N + T * K * N + (T + 1)) * C_{add} \\
&+ (N + 3 * T * N + (T - 1) * 2 * N * N + 4 * T * N * N) * C_{mult} \\
&+ (T * N + T * N * N + N * N + N * K) * C_{div} \\
&+ T * K * N * C_{comp_eq} \\
&+ 1 * C_{cmp_lt}
\end{aligned}$$

3 Stable baum-welch algorithm

3.1 forward step

- computing $\alpha[0]$
$$C_1(n) = N * C_{add} + N * C_{mult}$$
- computing scaling factor ct[0]
$$C_2(n) = 1 * C_{div}$$

- scale alpha[0]

$$C_3(n) = N * C_{mult}$$

- for T timesteps:
  - compute $\alpha[s,t]$

$$C_4(n) = N * N * C_{add} + (N * N + N) * C_{mult}$$

  - computing scaling factor ct[t]

$$C_5(n) = N * C_{add} + 1 * C_{div}$$

  - scaling alpha[s,t]

$$C_6(n) = N * C_{mult}$$

Overall

$$C_{forward}(n) = (N+T*N*N+T*N)*C_{add}+(N+N+T*(N*N+N)+T*N)*C_{mult}+(1+T)*C_{div}$$

$$C_{forward}(n) = N(1+T*(N+1))*C_{add}+N*(2+T*((N+1)+1))*C_{mult}+(1+T)*C_{div}$$

3.2 backward step

- for T-1 timesteps:
  - compute $\beta[s,t]$

$$C_1(n) = N * N * C_{add} + 2 * N * N * C_{mult}$$

  - scaling $\beta[s,t]$

$$C_2(n) = N * C_{mult}$$

Overall

$$C_{backward}(n) = (T-1) * N * N * C_{add} + ((T-1) * 2 * N * N + (T-1) * N) * C_{mult}$$

$$C_{backward}(n) = (T-1) * N * N * C_{add} + (T-1) * N * (2 * N + 1) * C_{mult}$$

3.3 update step

- compute gamma

$$C_1(n) = T * N * C_{mult}$$

- compute xi

$$C_2(n) = 3 * T * N * N * C_{mult}$$

- compute new pi

$$C_3(n) = N * C_{div}$$

- compute gamma sum denominator

$$C_4(n) = T * N * N * C_{add} + T * N * N * C_{div}$$

- new transition matrix

$$C_5(n) = N * N * C_{div}$$

- compute gamma sum numerator

$$C_6(n) = T * K * N * C_{add} + T * K * N * C_{div} + T * K * N * C_{comp\_eq}$$

- new emission matrix

$$C_7(n) = N * K * C_{div}$$

Overall

$$\begin{aligned} C_{update} =& (T * N * N + T * K * N) * C_{add} + (T * N + 3 * T * N * N) * C_{mult} \\ &+ (N + T * N * N + N * N + N * K + T * K * N) * C_{div} + T * K * N * C_{comp\_eq} \\ =& (T * N * N + T * K * N) C_{add} + T * N * (1 + 3 * N) * C_{mult} \\ &+ N * (1 + T * N + N + K + T * K) * C_{div} + T * K * N * C_{comp\_eq} \end{aligned}$$

3.4 finishing criteria

- for every timestep

$$C_1(n) = T * C_{add} + T * C_{log}$$

- return value

$$C_2(n) = 1 * C_{add} + 1 * C_{cmp\_lt}$$

Overall

$$C_{finished}(n) = (T + 1) * C_{add} + T * C_{log} + 1 * C_{cmp\_lt}$$

3.5 Total

$$
\begin{aligned}
C_{stable}(n) =& C_{forward}(n) + C_{backward}(n) + C_{update}(n) + C_{finished}(n)\\
C_{stable}(n) =& (N + T * N * N + T * N) * C_{add} + (N + N + T * N * N + T * N + T * N) * C_{mult}\\
& + (1 + T) * C_{div} + (T - 1) * N * N * C_{add}\\
& + ((T - 1) * 2 * N * N + (T - 1) * N) * C_{mult}\\
& + (T * N * N + T * K * N) * C_{add} + (T * N + 3 * T * N * N) * C_{mult}\\
& + (N + T * N * N + N * N + N * K + T * K * N) * C_{div} + T * K * N * C_{comp\_eq}\\
& + (T + 1) * C_{add} + T * C_{log} + 1 * C_{cmp\_lt}\\
C_{stable}(n) =& (N + T * N * N + T * N + (T - 1) * N * N + T * N * N + T * K * N + (T + 1)) * C_{add}\\
& + (N + N + T * N * N + T * N + T * N + (T - 1) * 2 * N * N\\
& + (T - 1) * N + T * N + 3 * T * N * N) * C_{mult}\\
& + (1 + T + N + T * N * N + N * N + N * K + T * K * N) * C_{div}\\
& + T * K * N * C_{comp\_eq}\\
& + T * C_{log}\\
& + 1 * C_{cmp\_lt}\\
=& (N + 2 * T * N * N + T * N + (T - 1) * N * N + (T + 1) + T * K * N) * C_{add}\\
& + (2 * N + 4 * T * N * N + 3 * T * N + (T - 1) * 2 * N * N + (T - 1) * N) * C_{mult}\\
& + (1 + T + N + T * N * N + N * N + N * K + T * K * N) * C_{div}\\
& + T * K * N * C_{comp\_eq}\\
& + T * C_{log}\\
& + 1 * C_{cmp\_lt}
\end{aligned}
$$

If we count all of these as one operation we get

$$
\begin{aligned}
C_{total}(n) =& N + 2 * T * N * N + T * N + (T - 1) * N * N + T + 1 + T * K * N\\
& + 2 * N + 4 * T * N * N + 3 * T * N + (T - 1) * 2 * N * N + (T - 1) * N\\
& + 1 + T + N + T * N * N + N * N + N * K + T * K * N + T * K * N + T + 1\\
=& 3KNT + KN + 10N^2T - 2N^2 + 5NT + 3N + 3T + 3
\end{aligned}
$$

Asymptotically:

$$C_{total} = 3KNT + 10N^2T$$

4 Reordered

4.1 Pre-Forward

- add remaining parts

$$C_1(n) = 2N * C_{add} + 2N * C_{div}$$

- new emission matrix

$$C_2(n) = KN * C_{mult}$$

$$C_{pre} = C_1 + C_2 = 2N * C_{add} + KN * C_{div} + KN * C_{mult}$$

4.2 Forward

- alpha(0)

$$C_1(n) = N * C_{add} + N * C_{mult}$$

- scale alpha(0)

$$C_2(n) = N * C_{mult}$$

- alpha(1) inner loop

$$C_3(n) = N^2 * C_{add} + N^2 * C_{mult}$$

- alpha(1) outer loop

$$C_4(n) = N * C_{add} + N * C_{mult}$$

- gamma Sum to zero

$$C_{4.5}(n) = N * C_{add} + 2N * C_{mult}$$

- scale alpha(t)

$$C_5(n) = N * C_{mult}$$

- T, outer loop

$$C_6(n) = (T - 2) * C_{div}$$

- T, scale loop

$$C_7(n) = (T - 2) * N * C_{mult}$$

- T, most inner loop

$$C_8(n) = (T - 2) * N^2 * C_{add} + (T - 2) * N^2 * C_{mult}$$

- T, write alpha loop

$$C_9(n) = (T - 2) * N * C_{add} + (T - 2) * N * C_{mult}$$

- alpha(T-1), outer loop

$$C_{10}(n) = N * C_{add} + N * C_{mult}$$

- alpha(T-1), inner loop

$$C_{11}(n) = N^2 * C_{add} + N^2 * C_{mult}$$

- scale alpha(t)

$$C_{12}(n) = N * C_{mult}$$

$$C_{forward} = \Sigma_i C_i = ((N^2T + NT + 2N) * C_{add} + (T - 2) * Cdiv + (N^2T + 2NT + 4N) * C_{mult}$$

4.3 Fused Backward and Update

- Init Beta and Gamma Sum

$$C_1(n) = 0$$

- T, outer loop

$$C_2(n) = 0$$

- pre-compute ab

$$C_4(n) = KN^2 * C_{mult}$$

- T, middle loop

$$C_3(n) = 2(T-1)N * C_{add} + 2(T-1) * N * C_{mult}$$

- T, inner loop

$$C_4(n) = 2(T-1)N^2 * C_{add} + 2(T-1)N^2 * C_{mult}$$

$$C_{fused} = \Sigma_i C_i = (2N^2T - 2N^2 + 2NT - 2N) * C_{add} + (KN^2 + 2N^2T - 2N^2 + 2NT - 2N) * C_{mult}$$

4.4 Finishing

$$C_{finishing} = T * C_{add} + T * C_{log2}$$

4.5 Total

$$C_{total} = C_{pre} + C_{forward} + C_{fused} + C_{finishing}$$

$$C_{total_add} = (3N^2T - 2N^2 + 3NT + 2N + T) * C_{add}$$

$$C_{total_{div}} = (T - 2 + KN) * C_{div}$$

$$C_{total_log2} = T * C_{log2}$$

$$C_{total_mult} = (KN^2 + KN + 3N^2T - 2N^2 + 4NT + 2N) * C_{mult}$$

Asymptotically (assuming all operations cost the same):

$$C_{total} = (KN^2 + 2KN + 6N^2T - 4N^2 + 7NT + 4N + 3T - 2) * C_{operation}$$

$$C_{total} \approx (KN^2 + 6N^2T) * C_{operation}$$