Supplement to Fast Large-scale Mixture Modeling with Component-specific Data Partitions

Bo Thiesson[*]
Microsoft Research

Chong Wang[*†]
Princeton University

Appendix: Efficient Variational E-step

As we have shown in the main paper, the optimal q_{B_k} satisfies

$$q_{B_k}(\lambda) = \exp\left(\frac{\sum_{l:B_l \subseteq B_k} \lambda_l}{|B_k|} - 1\right) \pi_k \exp\left(g_{B_k}\right). \tag{1}$$

Algorithm 1 (on p. 3) describes the general algorithm for finding the optimal solution, which first traverses $\mathcal{T}$ bottom up, level by level and gradually reduces the nested constraints to a single equation involving only one λ_l. After solving this equation the algorithm now traverses $\mathcal{T}$ top down, level by level and gradually resolves the remaining Lagrange multipliers $\lambda_l, l \in \mathcal{L}$ by back-substitution of previously resolved λ_l. The resolved Lagrange multipliers can now be used to find the desired variational distribution q. The next corollary is useful for the detailed description of the individual steps in Algorithm 1 that follows below.

Corollary 1. *Let* $i, i^* \in \{1, \ldots, I\}$. *If*

$$\exp\left(\frac{\lambda_i}{|B_i|} - 1\right) D_i = \exp\left(\frac{\lambda_{i^*}}{|B_{i^*}|} - 1\right) D_{i^*}$$

then

$$\lambda_i = |B_i| \left(\frac{\lambda_{i^*}}{|B_{i^*}|} + \log D_{i^*} - \log D_i\right).$$

Initialization. Let $\mathcal{V}$ denote the set of nodes in the MPT $\mathcal{T}$. When computing the solution for the variational distribution it is convenient to define three scalar values K_v, C_v, and D_v for each $v \in \mathcal{V}$ K_v. Initially,

$$\begin{aligned} K_v &= 0, \\ C_v &= \textstyle\sum_{k \in \mathcal{K}_v} \pi_k \exp(g_{B_k}), \\ D_v &= C_v, \end{aligned} \tag{2}$$

where $\mathcal{K}_v$ denote the set of mixture components for which its data partition has a data block associated with node v. For each $v \in V$, we define the *node distribution*

$$q_v = \textstyle\sum_{k \in \mathcal{K}_v} q_{B_k} = \exp\left(\frac{\sum_{l:B_l \subseteq B_v} \lambda_l}{|B_v|} - 1\right) C_v, \tag{3}$$

where the second equality follows from (1) and (2).

Collect-up. Collect-up traverses the tree $\mathcal{T}$ bottom up, level by level, where each step in this traversal considers a node v and its children $u \in ch(v)$. The crux of a collect-up step is a manipulation of the parent-node distribution q_v in such a way that the $\sum_{l:B_l \subseteq B_v} \lambda_l$ is replaced by a single $\lambda_l = \lambda_{l_v^*}$ of reference, where the index l_v^* emphasises that it is a reference-λ that has been selected at node v. In particular, as described below, the collect-up step will transform the representation in (3) into a representation

$$q_v = \exp\left(\frac{\lambda_{l_v^*}}{|B_{l_v^*}|} - 1\right) C_v \exp\left(\frac{K_v}{|B_v|}\right) \tag{4}$$

[*]Equal contributors. †Work done during internship at Microsoft Research.

by deriving the equality

$$\sum_{l:B_l \subseteq B_v} \lambda_l = |B_v| \frac{\lambda_{l_v^*}}{|B_{l_v^*}|} + K_v, \tag{5}$$

assuming a similar representation for each child-node distribution q_u, $u \in ch(v)$. Note that node distributions at leaf nodes only involve one λ and therefore trivially obey this condition, so traversal starts one level above. The transformation starts by the equality (recall that $u \in ch(v)$),

$$\sum_{l:B_l \subseteq B_v} \lambda_l = \sum_u \sum_{l:B_l \subseteq B_u} \lambda_l = \sum_u |B_u| \frac{\lambda_{l_u^*}}{|B_{l_u^*}|} + K_u, \tag{6}$$

which implies that the $\sum_{l:B_l \subseteq B_v} \lambda_l$ involved in the parent-node distribution q_v can be reduced to an expression involving just the $\lambda_{l_u^*}$ of reference for each of the $|ch(v)|$ children. In order to transform Eq. (6) into one that involves only the single $\lambda_{l_v^*}$ of reference, we apply the following procedure.

Let $l \mapsto r$ denote the path of nodes $(l = w_1, w_2, \ldots, w_n = r)$ from leaf l to root r in $\mathcal{T}$, and $w_i \mapsto w_j, 1 \leq i, j \leq n$ denote a subpath of $l \mapsto r$. With the notation $q_{w_i \mapsto w_j} = \sum_{w \in w_i \mapsto w_j} q_w$, the sum-to-one constraints in the main paper can be written as

$$q_{l \mapsto r} = q_{l \mapsto u} + q_{v \mapsto r} = 1 \text{ for all } l \in \mathcal{L}, \tag{7}$$

where $u \in ch(v)$ and $v \in \mathcal{V}$ are both on the path $l \mapsto r$. Since all children share the same path $v \mapsto r$ from their parent to the root, the constraints in (7) imply equality of all $q_{l \mapsto u}$, $u \in ch(v)$. In particular, we can ensure that for each $u \in ch(v)$ there exists a path, where the reference-λs for all nodes on the path to u are the same. (E.g., we can always choose the reference-λ associated with the left-most child at each step in the traversal.) We can therefore construct

$$q_{l_u^* \mapsto u} = \exp\left(\frac{\lambda_{l_u^*}}{|B_{l_u^*}|} - 1\right) D_u, \tag{8}$$

where

$$D_u = \sum_{w \in l_u^* \mapsto u} C_w \exp\left(\frac{K_w}{|B_w|}\right). \tag{9}$$

Thus, the condition for Corollary 1 is satisfied for all $q_{l_u^* \mapsto u}, u \in ch(v)$, allowing us to select one of the $\lambda_{l_u^*}$ as the $\lambda_{l_v^*}$ and represent each $\lambda_{l_u^*}$ as

$$\lambda_{l_u^*} = |B_{l_u^*}| \left(\frac{\lambda_{l_v^*}}{|B_{l_v^*}|} + \log D_{u^*} - \log D_u\right). \tag{10}$$

where u^* denotes the child with the chosen $\lambda_{l_u^*} = \lambda_{l_v^*}$ of reference. Substituting (10) into (6), we have

$$\begin{aligned}
\sum_{l:B_l \subseteq B_v} \lambda_l &= \sum_u \left(|B_u| \left(\frac{\lambda_{l_v^*}}{|B_{l_v^*}|} + \log D_{u^*} - \log D_u\right) + K_u\right) \\
&= |B_v| \frac{\lambda_{l_v^*}}{|B_{l_v^*}|} + |B_v| \log D_{u^*} + \sum_u \left(-|B_u| \log D_u + K_u\right) \\
&\triangleq |B_v| \frac{\lambda_{l_v^*}}{|B_{l_v^*}|} + K_v,
\end{aligned} \tag{11}$$

which is the desired equality in (5) leading to the representation of q_v as in (4) by substituting into (3). We are now ready for the next collect-up step.

The collect-up step is summarized by the following updates

$$K_v \leftarrow |B_v| \log D_{u^*} + \sum_u (-|B_u| \log D_u + K_u), \tag{12}$$

$$D_v \leftarrow C_v \exp\left(\frac{K_v}{|B_v|}\right) + D_{u^*}, \tag{13}$$

which are induced from Eqs. (11) and (9).

Distribute-down. After a collect-up traversal, each node in $\mathcal{T}$ is associated with a particular reference-λ, and these reference-λs are related as in (10). Distribute-down traverses $\mathcal{T}$ top down, where each step uses the reference-λ in the parent node to resolve the reference-λs associated with its child nodes. Notice that the update is redundant for the particular child with reference-λ chosen as the reference-λ for the parent in the collect-up traversal, where we simply have, $\lambda_{l_u^*} = \lambda_{l_v^*}$. The root node starts the recursion and needs special treatment. The constraints in (7) imply that

$$p_{l_r^* \mapsto r} = \exp\left(\frac{\lambda_{l_r^*}}{|B_{l_r^*}|} - 1\right) D_r = 1,$$

which can be solved to obtain

$$\lambda_{l_r^*} = |B_{l_r^*}|(1 - \log D_r). \tag{14}$$

Finalize. With λ_l available for all $l \in \mathcal{L}$, we can determine all q_{B_k} by simply inserting the appropriate $\sum \lambda_l$ into (1). Alternatively, it is more efficient to update q_{B_k}s during the distribute-down step, since $\lambda_{l_v^*}$ is available at this point, and $\sum \lambda_l$ therefore can be computed using Eq (5).

Algorithm 1 Variational E-step

```
//Initialization
for all nodes v ∈ V in MPT T do
    Initialize K_v, C_v, and D_v as in (2)
end for

//Collect-up
for all v ∈ V; traversed bottom up, level by level do
    if v is a leaf l ∈ L then
        Set pointer: *λ_v = *λ_l
    else
        Select u ∈ ch(v) and set pointer: *λ_v = *λ_u
        Update K_v and D_v as in (12) and (13)
    end if
end for

//Distribute-down and Finalize
for all v ∈ V; traversed top down, level by level do
    if v is the root then
        Set value λ_v = λ_{l_v^*} as in (14)
    else
        Set value λ_v = λ_{l_v^*} as in (10)
        for all k ∈ K_v do
            Compute q_{B_k} as in (1) using (5).
        end for
    end if
end for
```