Supplement for “Online Sparse Collapsed Hybrid Variational-Gibbs Algorithm for Hierarchical Dirichlet Process Topic Models”

Sophie Burkhardt and Stefan Kramer
Johannes Gutenberg-Universität Mainz
{burkhardt,kramer}@informatik.uni-mainz.de

1 Algorithm Description

In this section, we will describe the algorithm of our method in more detail. The model is trained using Alg. 2. For each minibatch we compute the dense distribution $q^{e,w}$ and the sum $Q_w$ for each word $w$ (line 3). Here, we assume that we use an empty pseudo document $e$ which enables to compute $q^e$ only once for the whole minibatch instead of once for each document. Now, we compute the alias tables $A_w$ for each word using distributions $q^{e,w}$ (line 4).

The computation of an alias table is given in algorithm 16. Here, $r$ is calculated as the average bucket size (line 1). First the distribution values are distributed to two sets $L$ and $H$ (lines 3–7). Each value (with the corresponding index) is added to $L$ if it is equal or smaller than $r$ or to $H$ if it is larger. Now, we calculate the alias table by repeatedly taking one value which is smaller or equal than $r$ and one which is larger from $L$ and $H$ respectively (lines 9–10). Then, we calculate how much probability is left after taking the smaller value and adding from the larger value to fill the bucket to be of size $r$ (line 12). The left-over probability is added back into $L$ or $H$ depending on whether it is larger or smaller than $r$ (lines 12–15).

After the alias table has been built, we can sample from it by generating one sample from a uniform distribution and selecting the corresponding entry from the alias table. Each entry in the alias table stores two topic indices and one probability $p_a$. To select one of the topics we draw another sample from a uniform distribution and check whether or not it is smaller than $p_a$. Thus, it is possible to draw from the distribution by generating two samples from a uniform distribution without having to iterate over the whole distribution.

After all alias tables have been computed, we start iterating over the documents of the current minibatch (Alg. 2, line 5). We do $S + B$ iterations over each document and save samples after the burn-in iterations $B$ (lines 10-11). For each token in the current document, we sample a topic and table indicator (line 9) using Alg. 3. Finally, we update the global distributions (line 12).

Alg. 3 describes the sampling of one topic and table indicator for word $w$ and corresponding alias table $A_w$. First we compute the sparse distribution $\tilde{p}$ by iterating over the topics that exist in the current document (line 1). At the same time we can calculate the discard mass $\Delta$ (line 3). We now check whether
to choose a topic from the sparse bucket or the dense bucket (lines 5–6). If we choose the dense bucket, we take a sample from the alias table as previously described (lines 12–14). We then check whether the sampled topic exists in the document and continue sampling until it is a new topic (line 15). If we choose the sparse bucket, we iterate over $\tilde{p}$ to determine the topic and table indicator (lines 6–11).

Algorithm 1: computeAliasTable($q$)

1. $r \leftarrow 1/q, \text{length}$
2. $L = H = A = \emptyset$
3. for $k = 1, \ldots, K$ do
   4. if $q(k) \leq r$ then
      5. $L \leftarrow L \cup (k, q(k))$
   6. else
      7. $H \leftarrow H \cup (k, q(k))$
8. while $L \neq \emptyset$ do
   9. $(l, pl) \leftarrow L.pop()$
10. $(h, ph) \leftarrow H.pop()$
11. $A \leftarrow A \cup (l, h, pl)$
12. if $ph + pl - r > r$ then
13.   $H \leftarrow H \cup (h, ph + pl - r)$
14. else
15.   $L \leftarrow H \cup (h, ph + pl - r)$
16. return $A$

Algorithm 2: Train Topic Model

Input: Dataset $D$
1. repeat
2. $M \leftarrow$ get minibatch from $D$
3. compute $q^e$ (Equations 23, 17, 18) and $Q = \sum q^e$
4. $A_w \leftarrow$ computeAliasTable($\frac{q^e}{Q_w}$) for each word $w$
5. for document $d \in M$ do
6.   $z_d \leftarrow$ initialize randomly
7.   for iteration $i = 1, \ldots, S + B$ do
8.     for token $n = 1, \ldots, N_d$ do
9.       $z_{dn}, u_{dn} \leftarrow$ Sample($A, w_{dn}$) (Algorithm 3)
10.      if $i > B$ then
11.         update $\tilde{\beta}$ and $\tilde{\gamma}$ (Equations 21 and 22)
12.   until convergence
Algorithm 3: Sample($A, w$)

1. compute $\tilde{p}$ (Equation 24)
2. compute $\tilde{P} = \sum \tilde{p}$
3. compute $\Delta$ (Equation 25)
4. initialize $i = -1, u \leftarrow 1$
5. sample $r \sim \text{Uniform}(0, \tilde{P} + \tilde{Q})$
6. if $r < \tilde{P}$ then
   7. while $r > 0$ do
      8. $i \leftarrow i + 1$
      9. $t \leftarrow i/2$
     10. $u \leftarrow 2 - (i \mod 2)$
     11. $r \leftarrow r - \tilde{p}_{j,w}(t,u)$
   12. else
      13. repeat
         14. $t \leftarrow \text{sample from Alias } A_w$
      15. until $t$ is new in document
16. return $u, t$