Practical Correlated Topic Modeling and Analysis via the Rectified Anchor Word Algorithm (Supplementary Material)

Moontae Lee[1] **Sungjun Cho**[2] **David Bindel**[2] **David Mimno**[2]
[1]University of Illinois at Chicago, Microsoft Research at Redmond
[2]Cornell University
moontae@uic.edu, {sc782,bindel,mimno}@cornell.edu

A Implementation

For reproducibility, we provide pseudocode for algorithms integrated in different parts of JSMF.

A.1 Rectification

Algorithm 1 Alternating Projection

def RECTIFY-C_AP$(\boldsymbol{C}, K)$
$\boldsymbol{C}_{\mathcal{NN}} \leftarrow \boldsymbol{C}$
repeat
  $(\boldsymbol{U}, \boldsymbol{\Lambda}_K) = \text{TRUNCATED-EIG}(\boldsymbol{C}_{\mathcal{NN}}, K)$
  $\boldsymbol{\Lambda}_K^{+} \leftarrow \text{diag}(\max\{\text{diag}(\boldsymbol{\Lambda}_K),\ 0\})$
  $\boldsymbol{C}_{\mathcal{PSD}} \leftarrow \boldsymbol{U}\boldsymbol{\Lambda}_K^{+}\boldsymbol{U}^T$
  $\boldsymbol{C}_{\mathcal{NOR}} \leftarrow \boldsymbol{C}_{\mathcal{PSD}} + \frac{1-\sum_{i,j}\boldsymbol{C}_{\mathcal{PSD}}(i,j)}{N^2}\boldsymbol{1}\boldsymbol{1}^T$
  $\boldsymbol{C}_{\mathcal{NN}} \leftarrow \max\{\boldsymbol{C}_{\mathcal{NOR}},\ 0\}$
until the convergence of $\boldsymbol{C}_{\mathcal{NN}}$
return $\boldsymbol{C} \leftarrow \boldsymbol{C}_{\mathcal{NN}}/(\sum_{i,j}\boldsymbol{C}_{\mathcal{NN}}(i,j))$

diag$(\cdot)$ is the Matlab-style operation that maps the input vector into the diagonal matrix or extracts the diagonal vector from the input matrix.

Algorithm 2 Cyclic Douglas-Rachford Iteration

def RECTIFY-C_DR$(\boldsymbol{C}, K)$
$\boldsymbol{C}_3 \leftarrow \boldsymbol{C}$
repeat
  $\boldsymbol{C}_1 \leftarrow \frac{\boldsymbol{I}+\boldsymbol{R}_{\mathcal{NOR}}\boldsymbol{R}_{\mathcal{PSD}}}{2}\boldsymbol{C}_3$
  $\boldsymbol{C}_2 \leftarrow \frac{\boldsymbol{I}+\boldsymbol{R}_{\mathcal{NN}}\boldsymbol{R}_{\mathcal{NOR}}}{2}\boldsymbol{C}_1$
  $\boldsymbol{C}_3 \leftarrow \frac{\boldsymbol{I}+\boldsymbol{R}_{\mathcal{PSD}}\boldsymbol{R}_{\mathcal{NN}}}{2}\boldsymbol{C}_2$
until the convergence of $\boldsymbol{C}_3$
return $\boldsymbol{C} \leftarrow \boldsymbol{C}_3/(\sum_{i,j}\boldsymbol{C}_3(i,j))$

$\boldsymbol{I}$ is the identity mapping, $\boldsymbol{R}_{\mathcal{NOR}}\boldsymbol{R}_{\mathcal{PSD}}$ denotes the composition of two reflection operators: the matrix is first reflected onto the $\mathcal{PSD}$ space, then onto the $\mathcal{NOR}$ space. $\boldsymbol{R}_{\mathcal{PSD}}\boldsymbol{C} = 2\boldsymbol{C}_{\mathcal{PSD}} - \boldsymbol{C}$, and similarly for $\mathcal{NOR}$ and $\mathcal{NN}$.

A.2 Topic Inference

Algorithm 3 Sparse Implicit Column-pivoted QR

def FIND-S$(\overline{\boldsymbol{C}}, K)$
$(\boldsymbol{P},\ \boldsymbol{Q},\ \boldsymbol{S},\ \boldsymbol{r}) \leftarrow (\overline{\boldsymbol{C}}^T,\ \boldsymbol{0}^{N\times K}, \varnothing,\ \boldsymbol{0}^K)$
$\boldsymbol{u} \leftarrow (\|\boldsymbol{p}_1\|_2^2, ..., \|\boldsymbol{p}_N\|_2^2) \in \mathbb{R}^{1\times N}$
for $k = 1$ to K **do**
  $n \leftarrow \text{argmax}_{1\le i\le N}\ u_i$
  $(\boldsymbol{S},\ \boldsymbol{q}_k,\ r_k) \leftarrow (\boldsymbol{S}\cup\{n\},\ \boldsymbol{p}_n,\ \sqrt{u_n})$
  $\boldsymbol{q}_k \leftarrow (\boldsymbol{q}_k - \sum_{l=1}^{k-1}\langle \boldsymbol{q}_l, \boldsymbol{p}_n\rangle \boldsymbol{q}_l)/r_k$
  $\boldsymbol{u} \leftarrow \boldsymbol{u} - (\boldsymbol{q}_k^T\boldsymbol{P}) \circ (\boldsymbol{q}_k^T\boldsymbol{P})$
end for
return $(\boldsymbol{S}, \boldsymbol{r})$

$\circ : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ denotes entry-wise multiplication of two vectors.

Algorithm 4 ADMM

def RECOVER-B$(\overline{\boldsymbol{C}}, \boldsymbol{c}, \boldsymbol{S}, \lambda, \gamma)$
$(\boldsymbol{U},\ \breve{\boldsymbol{B}},\ \boldsymbol{B}) \leftarrow ((\overline{\boldsymbol{C}}_{S*})^T,\ \boldsymbol{0}^{K\times N},\ \boldsymbol{0}^{N\times K})$
$\breve{\boldsymbol{B}}_{*S} \leftarrow \boldsymbol{I}_K$ $(\boldsymbol{I}_K = K\times K$ identity matrix)
$\boldsymbol{F} \leftarrow (\gamma\boldsymbol{U}^T\boldsymbol{U} + \boldsymbol{I}_K)^{-1}$
for each $i \in \{1, ..., N\} \setminus \boldsymbol{S}$ (in parallel) **do**
  $(\boldsymbol{v},\ \boldsymbol{f}) \leftarrow ((\overline{\boldsymbol{C}}_{i*})^T,\ \gamma\boldsymbol{U}^T\boldsymbol{v})$
  $\boldsymbol{y}^{(0)} \leftarrow \Pi_{\Delta^{K-1}}\big((\boldsymbol{U}^T\boldsymbol{U})^{-1}(\boldsymbol{f}/\gamma)\big)$
  $\boldsymbol{q}^{(0)} \leftarrow \boldsymbol{y}^{(0)}$
  repeat
    $\boldsymbol{p}^{(t)} \leftarrow \boldsymbol{F}(2\boldsymbol{y}^{(t-1)} - \boldsymbol{q}^{(t-1)} + \boldsymbol{f})$
    $\boldsymbol{q}^{(t)} \leftarrow \boldsymbol{q}^{(t-1)} + \lambda(\boldsymbol{p}^{(t)} - \boldsymbol{y}^{(t-1)})$
    $\boldsymbol{y}^{(t)} \leftarrow \Pi_{\Delta^{K-1}}(\boldsymbol{q}^{(t)})$
  until the convergence of $\boldsymbol{y}^{(t)}$
  $\breve{\boldsymbol{B}}_{*i} \leftarrow \boldsymbol{y}^{(t)}$
end for
for $(i,\ k) \in \{1, ..., N\} \times \{1, ..., K\}$ **do**
  $\boldsymbol{B}_{ik} \leftarrow (\breve{\boldsymbol{B}}_{ki}\boldsymbol{c}_i)/(\sum_{i'=1}^{N}\breve{\boldsymbol{B}}_{ki'}\boldsymbol{c}_{i'})$
end for
return $\boldsymbol{B}$

$\Pi_{\Delta^{K-1}}(\cdot)$ is the orthogonal projection to the $K-1$ simplex.

Algorithm 5 Diagonal Recovery

def RECOVER-A($\boldsymbol{C}, \boldsymbol{B}, \boldsymbol{S}$)

1: $(\boldsymbol{C}_{SS},\ \boldsymbol{D}) \leftarrow (\boldsymbol{C}(\boldsymbol{S}, \boldsymbol{S}), \boldsymbol{B}(\boldsymbol{S}, *))$
2: $\boldsymbol{A} \leftarrow \boldsymbol{D}^{-1}\boldsymbol{C}_{SS}\boldsymbol{D}^{-1}$
3: **return** $\boldsymbol{A}$

Set indexing extracts a principle submatrix whose rows and columns correspond to the arguments. Since $\boldsymbol{B}(\boldsymbol{S}, *)$ is diagonal, we use the element-wise reciprocal of $\boldsymbol{D}$ as $\boldsymbol{D}^{-1}$.