Online Group-Structured Dictionary Learning
Supplementary Material

Zoltán Szabó[1] Barnabás Póczos[2] András Lőrincz[1]
[1]School of Computer Science, Eötvös Loránd University,
Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary
web: http://nipg.inf.elte.hu
email: szzoli@cs.elte.hu, andras.lorincz@elte.hu
[2]School of Computer Science, Carnegie Mellon University,
5000 Forbes Ave, 15213, Pittsburgh, PA, USA
web: http://www.autonlab.org
email: bapoczos@cs.cmu.edu

In this note we will derive the update equations for the statistics describing the minimum point of $\hat{f}_t$ (Section 2). During the derivation we will need an auxiliary lemma concerning the behavior of certain matrix series. We will introduce this lemma in Section 1. The pseudocode of our OSDL method is provided in Section 3.

1 The forgetting factor in matrix recursions

Let $\mathbf{N}_t \in \mathbb{R}^{L_1 \times L_2}$ $(t = 1, 2, \ldots)$ be a given matrix series, and let $\gamma_t = \left(1 - \frac{1}{t}\right)^\rho$, $\rho \geq 0$. Define the following matrix series with the help of these quantities:

$$\mathbf{M}_t = \gamma_t \mathbf{M}_{t-1} + \mathbf{N}_t \in \mathbb{R}^{L_1 \times L_2} \quad (t = 1, 2, \ldots), \tag{1}$$

$$\mathbf{M}'_t = \sum_{i=1}^{t} \left(\frac{i}{t}\right)^\rho \mathbf{N}_i \in \mathbb{R}^{L_1 \times L_2} \quad (t = 1, 2, \ldots). \tag{2}$$

Lemma 1. *If $\rho = 0$, then $\mathbf{M}_t = \mathbf{M}_0 + \mathbf{M}'_t$ $(\forall t \geq 1)$. When $\rho > 0$, then $\mathbf{M}_t = \mathbf{M}'_t$ $(\forall t \geq 1)$.*

Proof.

1. Case $\rho = 0$: Since $\gamma_t = 1$ $(\forall t \geq 1)$, thus $\mathbf{M}_t = \mathbf{M}_0 + \sum_{i=1}^{t} \mathbf{N}_i$. We also have that $\left(\frac{i}{t}\right)^0 = 1$ $(\forall i \geq 1)$, and therefore $\mathbf{M}'_t = \sum_{i=1}^{t} \mathbf{N}_i$, which completes the proof.
2. Case $\rho > 0$: The proof proceeds by induction.
   - $t = 1$: In this case $\gamma_1 = 0$, $\mathbf{M}_1 = 0 \times \mathbf{M}_0 + \mathbf{N}_1 = \mathbf{N}_1$ and $\mathbf{M}'_1 = \mathbf{N}_1$, which proves that $\mathbf{M}_1 = \mathbf{M}'_1$.
   - $t > 1$: Using the definitions of $\mathbf{M}_t$ and $\mathbf{M}'_t$, and exploiting the fact that $\mathbf{M}_{t-1} = \mathbf{M}'_{t-1}$ by induction, after some calculation we have that:

$$\mathbf{M}_t = \gamma_t \mathbf{M}_{t-1} + \mathbf{N}_t = \left(1 - \frac{1}{t}\right)^\rho \left[\sum_{i=1}^{t-1} \left(\frac{i}{t-1}\right)^\rho \mathbf{N}_i\right] + \mathbf{N}_t \tag{3}$$

$$= \left(\frac{t-1}{t}\right)^\rho \left[\sum_{i=1}^{t-1} \left(\frac{i}{t-1}\right)^\rho \mathbf{N}_i\right] + \left(\frac{t}{t}\right)^\rho \mathbf{N}_t = \sum_{i=1}^{t} \left(\frac{i}{t}\right)^\rho \mathbf{N}_i = \mathbf{M}'_t. \tag{4}$$

□

2 Online update equations for the minimum point of $\hat{f}_t$

Our goals are (i) to find the minimum of

$$\hat{f}_t(\mathbf{D}) = \frac{1}{\sum_{j=1}^{t}(j/t)^{\rho}} \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \left[\frac{1}{2}\left\|\mathbf{x}_{O_i} - \mathbf{D}_{O_i}\boldsymbol{\alpha}_i\right\|_2^2 + \kappa\Omega(\boldsymbol{\alpha}_i)\right] \tag{5}$$

in $\mathbf{d}_j$ while the other column vectors of $\mathbf{D}$ ($\mathbf{d}_i$ $(i \neq j)$) are being fixed, and (ii) to derive online update rules for the statistics of $\hat{f}_t$ describing this minimum point. $\hat{f}_t$ is quadratic in $\mathbf{d}_j$, hence in order to find its minimum, we simply have to solve the following equation:

$$\frac{\partial \hat{f}_t}{\partial \mathbf{d}_j}(\mathbf{u}_j) = \mathbf{0}, \tag{6}$$

where $\mathbf{u}_j$ denotes the optimal solution. We can treat the Ω, and the $\frac{1}{\sum_{j=1}^{t}(j/t)^{\rho}}$ terms in (5) as constants, since they do not depend on $\mathbf{d}_j$. Let $\mathbf{D}_{-j}$ denote the slightly modified version of matrix $\mathbf{D}$; its j^{th} column is set to zero. Similarly, let $\boldsymbol{\alpha}_{i,-j}$ denote the vector $\boldsymbol{\alpha}_i$ where its j^{th} coordinate is set to zero. Now, we have that

$$\mathbf{0} = \frac{\partial \hat{f}_t}{\partial \mathbf{d}_j} = \frac{\partial}{\partial \mathbf{d}_j}\left[\sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\left\|\boldsymbol{\Delta}_i(\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i)\right\|_2^2\right] \tag{7}$$

$$= \frac{\partial}{\partial \mathbf{d}_j}\left[\sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\left\|\boldsymbol{\Delta}_i[(\mathbf{x}_i - \mathbf{D}_{-j}\boldsymbol{\alpha}_{i,-j}) - \mathbf{d}_j\alpha_{i,j}]\right\|_2^2\right] \tag{8}$$

$$= \frac{\partial}{\partial \mathbf{d}_j}\left[\sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\left\|(\boldsymbol{\Delta}_i\alpha_{i,j})\mathbf{d}_j - \boldsymbol{\Delta}_i(\mathbf{x}_i - \mathbf{D}_{-j}\boldsymbol{\alpha}_{i,-j})\right\|_2^2\right] \tag{9}$$

$$= 2\sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\alpha_{i,j}[(\boldsymbol{\Delta}_i\alpha_{i,j})\mathbf{d}_j - \boldsymbol{\Delta}_i(\mathbf{x}_i - \mathbf{D}_{-j}\boldsymbol{\alpha}_{i,-j})] \tag{10}$$

$$= 2\sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\alpha_{i,j}^2\mathbf{d}_j - 2\sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\alpha_{i,j}(\mathbf{x}_i - \mathbf{D}_{-j}\boldsymbol{\alpha}_{i,-j}), \tag{11}$$

where we used the facts that

$$\mathbf{x}_{O_i} - \mathbf{D}_{O_i}\boldsymbol{\alpha}_i = \boldsymbol{\Delta}_i(\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i), \tag{12}$$

$$\frac{\partial\left\|\mathbf{A}\mathbf{y} - \mathbf{b}\right\|_2^2}{\partial \mathbf{y}} = 2\mathbf{A}^T(\mathbf{A}\mathbf{y} - \mathbf{b}), \tag{13}$$

$$\boldsymbol{\Delta}_i = \boldsymbol{\Delta}_i^T = (\boldsymbol{\Delta}_i)^2. \tag{14}$$

After rearranging the terms in (11), we have that

$$\left(\sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\alpha_{i,j}^2\right)\mathbf{u}_j = \sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\alpha_{i,j}(\mathbf{x}_i - \mathbf{D}_{-j}\boldsymbol{\alpha}_{i,-j}) \tag{15}$$

$$= \sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\mathbf{x}_i\alpha_{i,j} - \sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\mathbf{D}_{-j}\boldsymbol{\alpha}_{i,-j}\alpha_{i,j} \tag{16}$$

$$= \sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\mathbf{x}_i\alpha_{i,j} - \sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i(\mathbf{D}_{-j}\boldsymbol{\alpha}_{i,-j} + \mathbf{d}_j\alpha_{i,j} - \mathbf{d}_j\alpha_{i,j})\alpha_{i,j} \tag{17}$$

$$= \sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\mathbf{x}_i\alpha_{i,j} - \sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\mathbf{D}\boldsymbol{\alpha}_i\alpha_{i,j} + \left(\sum_{i=1}^{t}\left(\frac{i}{t}\right)^{\rho}\boldsymbol{\Delta}_i\alpha_{i,j}^2\right)\mathbf{d}_j. \tag{18}$$

We note that (16) is a system of linear equations, and its solution $\mathbf{u}_j$ does not depend on $\mathbf{d}_j$. We have introduced the '$\mathbf{d}_j\alpha_{ij} - \mathbf{d}_j\alpha_{ij}$' term only for one purpose; it can help us with deriving the recursive updates for $\mathbf{u}_j$ in a simple form. Define the following quantities

$$\mathbf{C}_{j,t} = \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \boldsymbol{\Delta}_i \alpha_{i,j}^2 \in \mathbb{R}^{d_x \times d_x} \quad (j = 1, \ldots, d_\alpha), \tag{19}$$

$$\mathbf{B}_t = \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \boldsymbol{\Delta}_i \mathbf{x}_i \boldsymbol{\alpha}_i^T = [\mathbf{b}_{1,t}, \ldots, \mathbf{b}_{d_\alpha,t}] \in \mathbb{R}^{d_x \times d_\alpha}, \tag{20}$$

$$\mathbf{e}_{j,t} = \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \boldsymbol{\Delta}_i \mathbf{D}\boldsymbol{\alpha}_i \alpha_{i,j} \in \mathbb{R}^{d_x} \quad (j = 1, \ldots, d_\alpha). \tag{21}$$

Here (i) $\mathbf{C}_{j,t}$s are diagonal matrices and (ii) the update rule of $\mathbf{B}_t$ contains the quantity $\boldsymbol{\Delta}_i \mathbf{x}_i$, which is $\mathbf{x}_{O_i}$ extended by zeros at the non-observable ($\{1, \ldots, d_x\} \setminus O_i$) coordinates. By using these notations and (18), we obtain that $\mathbf{u}_j$ satisfies the following equation:

$$\mathbf{C}_{j,t}\mathbf{u}_j = \mathbf{b}_{j,t} - \mathbf{e}_{j,t} + \mathbf{C}_{j,t}\mathbf{d}_j. \tag{22}$$

Now, according to Lemma 1, we can see that (i) when $\rho = 0$ and $\mathbf{C}_{j,0} = \mathbf{0}$, $\mathbf{B}_0 = \mathbf{0}$, or (ii) $\rho > 0$ and $\mathbf{C}_{j,0}$, $\mathbf{B}_0$ are arbitrary, then the $\mathbf{C}_{j,t}$ and $\mathbf{B}_t$ quantities can be updated online with the following recursions:

$$\mathbf{C}_{j,t} = \gamma_t \mathbf{C}_{j,t-1} + \boldsymbol{\Delta}_t \alpha_{t,j}^2, \tag{23}$$

$$\mathbf{B}_t = \gamma_t \mathbf{B}_{t-1} + \boldsymbol{\Delta}_t \mathbf{x}_t \boldsymbol{\alpha}_t^T, \tag{24}$$

where $\gamma_t = \left(1 - \frac{1}{t}\right)^{\rho}$. We use the following online approximation for $\mathbf{e}_{j,t}$:

$$\mathbf{e}_{j,t} = \gamma_t \mathbf{e}_{j,t-1} + \boldsymbol{\Delta}_t \mathbf{D}\boldsymbol{\alpha}_t \alpha_{t,j}, \tag{25}$$

with initialization $\mathbf{e}_{j,0} = \mathbf{0}$ $(\forall j)$, and $\mathbf{D}$ is the *actual* estimation for the dictionary. This choice seems to be efficient according to our numerical experiences.

Note. *In the fully observable special case (*i.e., *when* $\boldsymbol{\Delta}_i = \mathbf{I}$, $\forall i$*) the* (19)-(21) *equations have the following simpler form:*

$$\mathbf{C}_{j,t} = \mathbf{I} \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \alpha_{i,j}^2, \tag{26}$$

$$\mathbf{B}_t = \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \mathbf{x}_i \boldsymbol{\alpha}_i^T, \tag{27}$$

$$\mathbf{e}_{j,t} = \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \mathbf{D}\boldsymbol{\alpha}_i \alpha_{i,j} = \mathbf{D} \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \boldsymbol{\alpha}_i \alpha_{i,j}. \tag{28}$$

Define the following term:

$$\mathbf{A}_t = \sum_{i=1}^{t} \left(\frac{i}{t}\right)^{\rho} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^T \in \mathbb{R}^{d_\alpha \times d_\alpha}, \tag{29}$$

and let $\mathbf{a}_{j,t}$ *denote the* j^{th} *column of* $\mathbf{A}_t$*. Now,* (28) *can be rewritten as*

$$\mathbf{e}_{j,t} = \mathbf{D}\mathbf{a}_{j,t}, \tag{30}$$

and thus (22) *has the following simpler form:*

$$(\mathbf{A}_t)_{j,j}\mathbf{u}_j = \mathbf{b}_{j,t} - \mathbf{D}\mathbf{a}_{j,t} + (\mathbf{A}_t)_{j,j}\mathbf{d}_j. \tag{31}$$

Algorithm (Online Group-Structured Dictionary Learning)

Input of the algorithm

$\mathbf{x}_{t,r} \sim p(\mathbf{x})$, (observation: $\mathbf{x}_{O_{t,r}}$, observed positions: $O_{t,r}$), $\mathbf{D}_0$ (initial dictionary),
T (number of mini-batches), R (size of the mini-batches), $\mathcal{G}$ (group structure),
ρ (≥ 0 forgetting factor), κ (> 0 tradeoff-), η ($\in (0, 1]$ regularization constant),
$\{\mathbf{d}^G\}_{G\in\mathcal{G}}$ ($\geq \mathbf{0}$, weights), $\boldsymbol{\mathcal{A}}$ (constraint set for $\boldsymbol{\alpha}$), $\boldsymbol{\mathcal{D}} = \times_{i=1}^{d_\alpha} \boldsymbol{\mathcal{D}}_i$ (constraint set for $\mathbf{D}$),
inner loop constants: ϵ (smoothing), T_α, T_D (number of iterations).

Initialization

$\mathbf{C}_{j,0} = \mathbf{0} \in \mathbb{R}^{d_x}$ $(j = 1, \ldots, d_\alpha)$, $\mathbf{B}_0 = \mathbf{0} \in \mathbb{R}^{d_x \times d_\alpha}$, $\mathbf{e}_{j,0} = \mathbf{0} \in \mathbb{R}^{d_x}$ $(j = 1, \ldots, d_\alpha)$.

Optimization

for $t = 1 : T$

Draw samples for mini-batch from $p(\mathbf{x})$: $\{\mathbf{x}_{O_{t,1}}, \ldots, \mathbf{x}_{O_{t,R}}\}$.
Compute the $\{\boldsymbol{\alpha}_{t,1} \ldots, \boldsymbol{\alpha}_{t,R}\}$ representations:
$\boldsymbol{\alpha}_{t,r}$=Representation($\mathbf{x}_{O_{t,r}}, (\mathbf{D}_{t-1})_{O_{t,r}}, \mathcal{G}, \{\mathbf{d}^G\}_{G\in\mathcal{G}}, \kappa, \eta, \boldsymbol{\mathcal{A}}, \epsilon, T_\alpha$), $r = 1, \ldots, R$.
Update the statistics of the cost function:
$\gamma_t = \left(1 - \frac{1}{t}\right)^\rho$,
$\mathbf{C}_{j,t} = \gamma_t \mathbf{C}_{j,t-1} + \frac{1}{R}\sum_{r=1}^{R} \boldsymbol{\Delta}_{t,r} \alpha_{t,r,j}^2$, $j = 1, \ldots, d_\alpha$,
$\mathbf{B}_t = \gamma_t \mathbf{B}_{t-1} + \frac{1}{R}\sum_{r=1}^{R} \boldsymbol{\Delta}_{t,r} \mathbf{x}_{t,r} \boldsymbol{\alpha}_{t,r}^T$,
$\mathbf{e}_{j,t} = \gamma_t \mathbf{e}_{j,t-1}$, $j = 1, \ldots, d_\alpha$. %(part-1)
Compute $\mathbf{D}_t$ using BCD:
$\mathbf{D}_t$=Dictionary($\{\mathbf{C}_{j,t}\}_{j=1}^{d_\alpha}, \mathbf{B}_t, \{\mathbf{e}_{j,t}\}_{j=1}^{d_\alpha}, \boldsymbol{\mathcal{D}}, T_D, \{O_{t,r}\}_{r=1}^{R}, \{\boldsymbol{\alpha}_{t,r}\}_{r=1}^{R}$).
Finish the update of $\{\mathbf{e}_{j,t}\}_{j=1}^{d_\alpha}$-s: %(part-2)
$\mathbf{e}_{j,t} = \mathbf{e}_{j,t} + \frac{1}{R}\sum_{r=1}^{R} \boldsymbol{\Delta}_{t,r} \mathbf{D}_t \boldsymbol{\alpha}_{t,r} \alpha_{t,r,j}$, $j = 1, \ldots, d_\alpha$.

end

Output of the algorithm

$\mathbf{D}_T$ (learned dictionary).

Table 1: Pseudocode: Online Group-Structured Dictionary Learning.

Here $(\cdot)_{j,j}$ stands for the $(j,j)^{th}$ entry of its argument. By applying again Lemma 1 for (29)*, we have that when (i) $\rho = 0$ and $\mathbf{A}_0 = \mathbf{0}$, or (ii) $\rho > 0$ and $\mathbf{A}_0$ is arbitrary, then $\mathbf{A}_t$ can be updated online with the following recursion:*

$$\mathbf{A}_t = \gamma_t \mathbf{A}_{t-1} + \boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T. \tag{32}$$

We also note that in the fully observable case (24) *reduces to*

$$\mathbf{B}_t = \gamma_t \mathbf{B}_{t-1} + \mathbf{x}_t \boldsymbol{\alpha}_t^T, \tag{33}$$

and thus [1] is indeed a special case of our model:

- *We calculate $\mathbf{u}_j$ by* (31)*.*
- *To optimize $\hat{f}_t$, it is enough to keep track of $\mathbf{A}_t$ and $\mathbf{B}_t$ instead of $\{\mathbf{C}_{j,t}\}_{j=1}^{d_\alpha}, \mathbf{B}_t, \{\mathbf{e}_{j,t}\}_{j=1}^{d_\alpha}$.*
- *The quantities $\mathbf{A}_t$ and $\mathbf{B}_t$ can be updated online by* (32) *and* (33)*.*

3 Pseudocode

The pseudocode of the OSDL method with mini-batches is presented in Table 1-3. Table 2 calculates the representation for a fixed dictionary, and Table 3 learns the dictionary using fixed representations. Table 1 invokes both of these subroutines.

Algorithm (Representation)

Input of the algorithm

$\mathbf{x}$ (observation), $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_{d_\alpha}]$ (dictionary),
$\mathcal{G}$ (group structure), $\{\mathbf{d}^G\}_{G\in\mathcal{G}}$ (weights), κ (tradeoff-), η (regularization constant),
$\mathcal{A}$ (constraint set for $\boldsymbol{\alpha}$), ϵ (smoothing), T_α (number of iterations).

Initialization

$\boldsymbol{\alpha} \in \mathbb{R}^{d_\alpha}$.

Optimization

for $t = 1 : T_\alpha$

Compute $\mathbf{z}$: $z^G = \max\left(\left\|\mathbf{d}^G \circ \boldsymbol{\alpha}\right\|_2^{2-\eta} \left\|\left(\left\|\mathbf{d}^G \circ \boldsymbol{\alpha}\right\|_2\right)_{G\in\mathcal{G}}\right\|_\eta^{\eta-1}, \epsilon\right), G \in \mathcal{G}.$

Compute $\boldsymbol{\alpha}$:

compute $\boldsymbol{\zeta}$: $\zeta_j = \sum_{G\in\mathcal{G}, G\ni j} \frac{\left(d_j^G\right)^2}{z^G}, j = 1, \ldots, d_\alpha,$

$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha}\in\mathcal{A}}{\operatorname{argmin}} \left[\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \kappa\boldsymbol{\alpha}^T diag(\boldsymbol{\zeta})\boldsymbol{\alpha}\right].$

end

Output of the algorithm

$\boldsymbol{\alpha}$ (estimated representation).

Table 2: Pseudocode for *representation* estimation using fixed dictionary.

Algorithm (Dictionary)

Input of the algorithm

$\{\mathbf{C}_j\}_{j=1}^{d_\alpha}, \mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_{d_\alpha}], \{\mathbf{e}_j\}_{j=1}^{d_\alpha}$ (statistics of the cost function),
$\mathcal{D} = \times_{i=1}^{d_\alpha} \mathcal{D}_i$ (constraint set for $\mathbf{D}$), T_D (number of $\mathbf{D}$ iterations),
$\{O_r\}_{r=1}^R$(equivalent to $\{\boldsymbol{\Delta}_r\}_{r=1}^R$), $\{\boldsymbol{\alpha}_r\}_{r=1}^R$ (observed positions, estimated representations).

Initialization

$\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_{d_\alpha}]$.

Optimization

for $t = 1 : T_D$

for $j = 1 : d_\alpha$ %update the j^{th} column of $\mathbf{D}$

Compute '$\{\mathbf{e}_j\}_{j=1}^{d_\alpha}$'-s:

$\mathbf{e}_j^{temp} = \mathbf{e}_j + \frac{1}{R}\sum_{r=1}^R \boldsymbol{\Delta}_r \mathbf{D}\boldsymbol{\alpha}_r \alpha_{r,j}.$

Compute $\mathbf{u}_j$ solving the linear equation system:

$\mathbf{C}_j\mathbf{u}_j = \mathbf{b}_j - \mathbf{e}_j^{temp} + \mathbf{C}_j\mathbf{d}_j.$

Project $\mathbf{u}_j$ to the constraint set:

$\mathbf{d}_j = \Pi_{\mathcal{D}_j}(\mathbf{u}_j).$

end

end

Output of the algorithm

$\mathbf{D}$ (estimated dictionary).

Table 3: Pseudocode for *dictionary* estimation using fixed representations.

References

[1] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:10–60, 2010.