ENSAE Paris - 3rd year
Year 2020-2021

Compressed Sensing
Final report

Compressed Sensing - Final Report

Author : Mo LI, Zhonghao LI

This document consists the final report which should be submitted by 25 march 2021. This document contains only the theoretical part of our final work. For the practical implementation, please refer to the attached notebook.

1 Non-negative Matrix Factorization

In this section, we are going to focus on a previous study of non-negative matrix factorization technique. Specifically, we refer mainly to Le Thi Khanh Hien et al. (2020) [1] and it's references.

1.1 Introduction

As for a nonnegative matrix factorization(NMF) problem, given a nonnegative matrix data $V \in \mathbb{R}_+^{m+n}$ and a positive integer $r < min(m, n)$, is to find $W \in \mathbb{R}_+^{m\times r}$ and $H \in \mathbb{R}_+^{r\times n}$ such that $V \approx WH$. The quality of this approximation is measured by

$$D(V|WH) = \sum_{i=1}^{m}\sum_{j=1}^{n} d(V_{ij}|(WH)_{ij}),$$

and thus the NMF problem is to find a pair W, H to minimise $D(V|WH)$. Here, the scalar cost function $d(x|y)$ is given as:

$$d(x|y) = x\log\left(\frac{x}{y}\right) - x + y,$$

which is a special case of the β-divergence [2], and namely, the (generalized) Kullback-Leibler Divergence. In this case, the KL NMF problem can be discribed as the optimization problem:

$$\min_{W\in\mathbb{R}_+^{m\times r}, H\in\mathbb{R}_+^{r\times m}} D_{KL}(V|WH), \tag{1}$$

where

$$D_{KL}(V|WH) = \sum_{i=1}^{m}\sum_{j=1}^{n}((WH)_{ij} - V_{ij}\log(WH)_{ij}) + \sum_{i=1}^{m}\sum_{j=1}^{n}(V_{ij}\log V_{ij} - V_{ij}),$$

while we define additionally $0 \times \log 0 = 0$ and $a \times \log 0 = +\infty$ for $a > 0$.

The reason why the Frobenius norm, which corresponds to the β-divergence with $\beta = 2$, is not considered here is that it corresponds to the maximum likelihood estimator in the presence of additive i.i.d. Gaussian noise, but in this case we have a positive possibility to observe negative entries. Moreover, many nonnegative data sets are sparse in which case Gaussian noise is not appropriate.

However, a big advantage of adopting the Fro NMF is that the gradients of the objective with respect to W and H are Lipschitz-continuous over $W \in \mathbb{R}_+^{m\times r}, H \in \mathbb{R}_+^{r\times n}$. Although the objective of KL NMF is block-wise convex, but it is not differentiable at W or H when $(WH)_{ij} = 0$ for some i, j. Thus, proposing an efficient algorithm to solve the KL NMF is a more difficult job.

In the following, when we introduce the noise into our model, it is assumed that V_{ij} is a sample of variables $\tilde{V}_{ij}$ following the Poisson distribution of parameter $(WH)_{ij}$:

$$\mathbb{P}(\tilde{V}_{ij} = k) = \frac{(WH)_{ij}^k e^{-(WH)_{ij}}}{k!}, \qquad k = 0, 1, 2, \ldots$$

Meanwhile, since the objective function of problem 1 is not finite at every point of the constraint set, it is proposed to study the following version:

$$\begin{aligned} \min_{W,H} \quad & D(V|WH) \\ s.t. \quad & W_{ik} \geq \epsilon, H_{kj} \geq \epsilon, \qquad i \in [m], j \in [n], k \in [r], \end{aligned} \tag{2}$$

which is a truncated version of problem 1. In practice, we can fix ϵ above sufficiently small where the machine precision is recommended.

1.2 Properties of KL NMF and existing studies

Le Thi Khanh Hien et al. [1] presented several properties of the KL NMF problem. First of all, it is proved that the problem 2 has always at least a solution, although it may not be unique. A more interesting result is the following:

Proposition 1 *If (W^*, H^*) is a KKT point of 1, then W^*H^* preserves the row sums and the column sums of V, that is,*

$$Ve = (W^*H^*)e, \quad e^TV = e^T(W^*H^*),$$

where e denotes the vector of all ones with an appropriate dimension.

This gives us a brief description of how the solution of problem 1 looks like. And meanwhile, it gives us a hint on how to improve a current solution simply by multiplying a constant. If we say a pair (W, H) is scaled if the optimal solution of the problem $\min_{\alpha\in\mathbb{R}} D(V|\alpha WH)$ equales to 1, we may have the following proposition:

Proposition 2 *A pair (W, H) is scaled if and only if*

$$\sum_{i=1}^{m}\sum_{j=1}^{n} V_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} (WH)_{ij}.$$

Using this proposition, for any solution (W, H) to KL NMF, it can be improved by updating, for example, at each iteration:

$$W \leftarrow W\frac{e^TVe}{e^T(WH)e}.$$

In what follows, since the partial derivative of the objective function of KL NMF is not Lipschitz continous, a generalized version of the Lipschitz smoothness is proposed, namely relative smoothness, which is defined as below:

Definition 1 *Let $\kappa(\cdot)$ be any given differentiable convex function defined on a convex set Q. The function $g(\cdot)$ is L-smooth relative to $\kappa(\cdot)$ on Q if for any $x, y \in intQ$, there is a scalar L for which*

$$g(y) \leq g(x) + \langle \nabla g(x), y - x\rangle + L\mathcal{B}_\kappa(y, x),$$

where

$$\mathcal{B}_\kappa(y, x) = \kappa(y) - \kappa(x) - \langle \nabla\kappa(x), y - x\rangle, \qquad \forall x, y \in Q.$$

And this yields the following propositon:

Proposition 3 *Let $v \in \mathbb{R}_+^m$ and $W \in \mathbb{R}_+^{m\times r}$, and*

$$D(v|Wh) = \sum_{i=1}^{m} \left((Wh)_i - v_i \log(Wh)_i + v_i \log v_i - v_i\right).$$

Then the function $h \to D(v|Wh)$ is relative smooth to $\kappa(h) = -\sum_{j=1}^{r} \log h_j$ with the relative smooth constant $L = \|v\|_1$.

This proposition 3 is crucial to prove the monotonically decreasing property, as well as the bounded nature and globally convergence of the BMD method for KL NMF given by Hien et al. (2020) [1].

To give another monotone algorithm to solve problem 2, Hien et al. (2020) [1] also studied the self-concordant properties, which is defined in definition 2, for KL NMF problem.

Consider a closed convex function $g : \mathbb{E} \to \mathbb{R}$ with an open domain. Fixing a point $x \in \mathrm{dom}(g)$ and a direction $d \in \mathbb{E}$, let $\phi(\tau)_x = g(x+\tau d)$ and $\mathbf{D}g(x)[d] = \phi_x^{'}(0)$, $\mathbf{D}g(x)[d,d] = \phi_x^{''}(0)$ and $\mathbf{D}g(x)[d,d,d] = \phi_x^{'''}(0)$.

Definition 2 *We say a closed convex function with an open domain g belongs to the class $\mathcal{F}_M$ of self-concordant functions with parameter $M \geq 0$, if*

$$|\mathbf{D}^3 g(x)[d,d,d]| \leq 2M\|d\|^3_{\nabla^2 g(x)},$$

where $\|d\|^3_{\nabla^2 g(x)} = \langle \nabla^2 g(x)d, d\rangle = \phi_x^{''}(0)$. The function g is called standard self-concordant when $M = 1$.

With this definition, the following proposition has been proved:

Proposition 4 *Given $V \in \mathbb{R}_+^{m\times n}$, $W \in \mathbb{R}_+^{m\times r}$ and $H \in \mathbb{R}_+^{r\times n}$, the scalar function*

$$H_{kj} \to D(V|WH) = \sum_{i=1}^{m}\left(\sum_{a=1}^{n} W_{ia}H_{aj} - V_{ij}\log\sum_{a=1}^{n} W_{ia}H_{aj}\right) + constant$$

is a self-concordant function with constant $c_{H_{kj}} = \max_{\{i|V_{ij}>0\}}\left\{\frac{1}{\sqrt{V_{ij}}}\right\}$.

Some existing algorithms have also been discussed in Hien et al. (2020) [1], whereas the most interesting one among them is the MU(Multiplicative Updates) method. As it was referred to Févotte et al. (2008) [3], the original MU method concerned a subproblem of KL NMF when W is fixed to be solved for each column of H, that is the linear regression problem:

$$\min_{h\in\mathbb{R}_+^R} D_{KL}(v, Wh),$$

which consists updating the matrix H column by column. More specifically, at each iteration,

$$H_{kj}^{+} = H_{kj} \cdot \left(\frac{\sum_{l=1}^{m} W_{lk}V_{lj}/(WH)_{lj}}{\sum_{l=1}^{m} W_{lj}}\right)$$

In the context of KL NMF, it is derived that one can update alternatively W and H, and the MU method can be generalized to the perturbed KL NMF problem 2 as:

$$H_{aj}^{k+1} = max\left(H_{aj}^{k}\frac{\sum_{i=1}^{n} W_{ia}^{k}V_{ij}^{k}/(W^kH^k)_{ij}}{\sum_{\mu=1}^{n} W_{\mu a}^{k}}, \epsilon\right),$$

and

$$W_{ia}^{k+1} = max\left(W_{ia}^{k}\frac{\sum_{j=1}^{m} H_{aj}^{k+1}V_{ij}^{k}/(W^{k}H^{k+1})_{ij}}{\sum_{\nu=1}^{m} W_{a\nu}^{k}}, \epsilon\right).$$

An important property of MU is presented in the following proposition:

Proposition 5 *Let $\epsilon = 0$ and denote H^+ (resp. W^+) the update of H (resp. W) after applying one MU of H (resp. W) on (W, H). Suppose the MU for H (resp. W) is well-defined, that is, $(WH)_{ij} > 0$ for all i, j and W has no zero column (resp. H has no zero row). Then the MU of H preserve the column sum of V, that is, $e^T V = e^T(WH^+)$, while the MU of W preserves the row sum of V, that is, $Ve = W^+He$.*

The reason why proposition 5 is interesting is that is shows that any iterate of the MU, except the first one, is scaled, as defined previously. Hence, MU can be used to scale the pair (W, H) within KL NMF algorithms to improve their performance.

Many other algorithms, including the ADMM method, Primal-dual approach and cyclic coordinate descent method, are also presented in Hien et al. (2020) [1], but they are not as interesting as the MU method.

1.3 New methods proposed

1.3.1 Block Mirror Descent Method

The first method presented, Block Mirror Descent(BMD) method, is appreciated because it is the first method in the literature that guarantee the non-increasingness of the objective function as well as the global convergence, but we can see later that in practice, this method doesn't usually get the best performance.

Using the L-relative smoothness defined in 1, one can generalise the classic update of the gradient descent (GD) scheme for a L-smooth problem:

$$x^{k+1} = \underset{x\in Q}{\operatorname{argmin}} \left\{ g(x^k) + \left\langle (x^k), x - x^k \right\rangle + \frac{L}{2}\|x - x^k\|_2^2 \right\}$$

into the following mirror descent step:

$$x^{k+1} = \underset{x\in Q}{\operatorname{argmin}} \{ g(x^k) + \left\langle \nabla g(x^k), x - x^k \right\rangle + L\mathcal{B}_\kappa(x, x^k) \}.$$

The BMD method uses the mirror descent step by updating the variable block by block. More specifically, if we consider the following problem:

$$\min_{x\in\mathcal{X}} f(x),$$

where f is continuously differentiable on $\mathcal{X}$, $x = (x_1, x_2, \ldots, x_s)$ and $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_s$ being closed convex sets. It is also assumed that for all $\bar{x} \in \mathcal{X}$, the function $x_i \to f(\bar{x}_1, \ldots, \bar{x}_{i-1}, x_i, \bar{x}_{i+1}, \ldots, \bar{x}_s)$ is relative smooth to $\kappa_i(x_i)$ with constant $L_i(\bar{x}_1, \ldots, \bar{x}_{i-1}, \bar{x}_{i+1}, \ldots, \bar{x}_s)$, which is bounded by $\underline{L}_i$ and $\bar{L}_i$. Concerning this problem, at each iteration and for each i, we only need to do the following update:

$$x_i^{k+1} = \underset{x_i\in\mathcal{X}_i}{\operatorname{argmin}} \{ f_i^k(x_i^k) + \left\langle \nabla f_i^k(x_i^k), x_i - x_i^k \right\rangle + L_i^k \mathcal{B}_{\kappa_i}(x_i, x_i^k) \}.$$

The proposition 3 ensures us to be capable to apply BMD to solve problem 2 where the blocks of variables are the columns of H and the rows of W, and we have the following proposition:

Proposition 6 *Using the notation of Proposition 3, the problem*

$$\min_{h>\epsilon} \left\{ D(v|Wh^k) + \left\langle \nabla_h D(v|Wh)[h^k], h - h^k \right\rangle + L\mathcal{B}_\kappa(h, h^k) \right\}$$

has a unique closed-form solution h^{k+1}: for $l \in [r]$:

$$(h^{k+1})_l = \max \left\{ \frac{(h^k)_l}{1 + \frac{1}{L}(h^k)_l \left(\sum_{i=1}^m W_{il} - \sum_{i=1}^m \frac{v_i W_{il}}{(Wh^k)_i} \right)}, \epsilon \right\}$$

It is also proved in Hien et al. (2020) that this BMD for KL NMF for problem 2 monotonically decreases the objective function for any $\epsilon > 0$ and the generated sequence by BMD is bounded and globally converges to a stationary point of (2).

1.3.2 A scalar Newton-type algorithm

Taking advantage of the concordant property of the objective function, a new scalar Newton-type (SN) algorithm is also proposed.

This algorithm referred to a study of Tran-Dinh et al. [4] which gives a Newton solution of the form:

$$\min_{x \in \mathbb{R}^n} \Psi(x) = \psi(x) + \phi(x),$$

where $\psi(x)$ is a standard self-concordant function and $\phi(x)$ is a proper, closed, convex but possibly non-smooth function. By applying this method into the KL NMF problem, the SN algorithm is proposed and the non-increasingness of the objective function is also proved. One can also see that this algorithm works well in practice.

1.3.3 SN-MU algorithm

As SN solutions for the KL NMF problem is not assured to be scaled, it is nature to try to adopt a MU step, which returns a scaled pair (W, H), after several SN steps. Hien et al. (2020) [1] recommended to do a MU step after every 10 SN steps. This provides us a powerful algorithm in practice.

1.4 Numerical experiment, conclusion and perspective

Hien et al. (2020) compared several compared several methods, including the three new methods above, using both simulated data and real data. Sparsity is involved in the simulation data, with density equals to $1, 0.9, 0.3$ respectively. Results show that among all methods, CCD, MU and SN-MU provide generally better results, while BMD, which enjoys really good properties in the literature, has an average performance.

We did thought about using those KL NMF methods presented into the recommendation system problem, since matrix factorization is a reliable solution to recommendation system models, and NMF problems always give us non-negative results, which is appealing for a recommendation system. However, after a simple implementation of the MU method with a simulated data, we found out that the result was far from satisfying. KL NMF somehow managed to maintain the sparsity of the original matrix, which makes it impossible to propose new items to be recommended to a certain user.

The reason is probably hidden behind the objective function. Recall that the objective function in the KL NMF problem is:

$$d(x|y) = x \log \frac{x}{y} - x + y,$$

which is positive even if $x = 0$, i.e. when we have no information about a user's rating of an item. In this case, it is relatively expensive to do the prediction, rather than maintaining the sparsity, which means that this model does not tend to fill the NAs in the original sparse matrix with predictions, contrary to the aim of the recommendation system.

A solution to this problem may be modifying the objective function. In future studies, one may try using the standard KL divergence, i.e. $d(x|y) = x \log \frac{x}{y}$ or a variation of the generalized KL divergence $d(x|y) = x \log \frac{x}{y} - x^2 + xy$ to ensure that the value of the objective function is 0 whenever $x = 0$. However, we are not ensured that these modifications do not affect the solvability.

Other than trying to implement the Nonnegative Matrix Factorization into the recommendation system problems, we decided to adopt some of the others classical methods, which are introduced in the following sections.

2 Vanilla Matrix Factorization

2.1 Principle analysis

A straightforward matrix factorization model maps both users and items to a joint latent factor space of dimensionality D — such that user-item interactions are modeled as inner products in that space.

1. Accordingly, each item i is associated with a vector q_i, and each user u is associated with a vector p_u.
2. For a given item i, the elements of q_i measure the extent to which the item possesses those factors, positive or negative.
3. For a given user u, the elements of p_u measure the extent of interest the user has in items that are high on the corresponding factors, positive or negative.
4. The resulting dot product $(q_i \cdot p_u)$ captures the interaction between user u and item i, which is the user's overall interest in the item's characteristics.

Thus, we have the equation 3 as follow:

$$R_{ui} = q_i \cdot p_u \tag{3}$$

The big challenge is to compute the mapping of each item and user to factor vectors q_i and p_u. Matrix factorization does this by minimizing the regularized squared error on the set of known ratings, as seen in equation below:

$$\sum (R_{ui} - q_i \cdot p_u)^2 + Regularize(q_i + p_u)$$

The model is learned by fitting the previously observed ratings. However, the goal is to generalize those previous ratings in a way that predicts future/unknown ratings. Thus, we want to avoid overfitting the observed data by adding an L2 regularization penalty to each element and optimize the learned parameters simultaneously with stochastic gradient descent.

1. When we are using SGD to fit a model's parameters to the learning problem at hand, we take a step in the solution space towards the gradient of the loss function with respect to the network's parameters at each iteration of the algorithm. Since the user-item interaction matrix in our recommendations is very sparse, this method of learning might overfit to training data.

2. L2 is a specific way of regularizing a cost function with the addition of a complexity-representing term. The term is the squared Euclidean norm of the user and item latent factors. An additional parameter λ is added to allow control of the strength of the regularization.

3. Adding the L2 term usually results in much smaller parameters across the entire model.

2.2 Loss function

For the score prediction, we use the squared difference to construct the loss function:

$$Cost = \sum_{u,i \in R} (r_{ui} - \hat{r}_{ui})^2 = \sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik})^2$$

Add L2 regularization:

$$Cost = \sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik})^2 + \lambda (\sum_{U} {p_{uk}}^2 + \sum_{I} {q_{ik}}^2)$$

The partial derivative of the loss function:

$$\begin{aligned} \frac{\partial}{\partial p_{uk}} Cost &= \frac{\partial}{\partial p_{uk}} [\sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik})^2 + \lambda (\sum_{U} {p_{uk}}^2 + \sum_{I} {q_{ik}}^2)] \\ &= 2 \sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik})(-q_{ik}) + 2\lambda p_{uk} \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial q_{ik}} Cost &= \frac{\partial}{\partial q_{ik}} [\sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik})^2 + \lambda (\sum_{U} {p_{uk}}^2 + \sum_{I} {q_{ik}}^2)] \\ &= 2 \sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik})(-p_{uk}) + 2\lambda q_{ik} \end{aligned}$$

2.3 Stochastic gradient descent method optimization

Gradient descent update parameters p_{uk}, q_{ik}:

$$\begin{aligned} p_{uk} &:= p_{uk} - \alpha \frac{\partial}{\partial p_{uk}} Cost \\ &:= p_{uk} - \alpha [2 \sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik})(-q_{ik}) + 2\lambda p_{uk}] \\ &:= p_{uk} + \alpha [\sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik}) q_{ik} - \lambda p_{uk}] \\ q_{ik} &:= q_{ik} + \alpha [\sum_{u,i \in R} (r_{ui} - \sum_{k=1}^{k} p_{uk} q_{ik}) p_{uk} - \lambda q_{ik}] \end{aligned}$$

Stochastic gradient descent:

$$p_{uk} := p_{uk} + \alpha[(r_{ui} - \sum_{k=1}^{k} p_{uk}q_{ik})q_{ik} - \lambda_1 p_{uk}]$$
$$q_{ik} := q_{ik} + \alpha[(r_{ui} - \sum_{k=1}^{k} p_{uk}q_{ik})p_{uk} - \lambda_2 q_{ik}]$$

Since the matrices P and Q matrix are two different matrices, each typically take different regularization parameters, such as λ_1 and λ_2.

3 Matrix Factorization with Biases

3.1 Principle analysis

One benefit of the matrix factorization approach to collaborative filtering is its flexibility in dealing with various data aspects and other application-specific requirements. Recall that equation 3 attempts to capture the interactions between users and items that produce different rating values. However, much of the observed variation in the rating values are due to effects associated with either users or items, known as biases, independent of any interactions. The intuition behind this is that some users give high ratings than others, and some items received high ratings than others systematically.

Thus, we can extend equation 3 to equation 4 as follows:

$$R_{ui} = q_i \cdot p_u + \mu + b_i + b_u \tag{4}$$

1. The bias involved in the overall average rating is denoted by μ.
2. The parameters b_i and b_u indicate the observed deviations of item i and user u from the average, respectively.
3. Note that the observed ratings are broken down into 4 components: (1) user-item interaction, (2) global average, (3) item bias, and (4) user bias.

The model is learned by minimizing a new squared error function, as seen in equation 4 below:

$$\sum (R_{ui} - q_i \cdot p_u - \mu - b_i - b_u)^2 + Regularize(q_i + p_u + b_i + b_u)$$

3.2 Loss function

For the score prediction, we use the squared difference to construct the loss function:

$$Cost = \sum_{u,i \in R} (r_{ui} - \hat{r}_{ui})^2 = \sum_{u,i \in R} (r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik})^2$$

Add L2 regularization:

$$Cost = \sum_{u,i \in R} (r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik})^2 + \lambda(\sum_{U} {b_u}^2 + \sum_{I} {b_i}^2 + \sum_{U} {p_{uk}}^2 + \sum_{I} {q_{ik}}^2)$$

3.3 Stochastic gradient descent method optimization

Gradient descent update parameters p_{uk}, q_{ik}, b_u, b_i:

$$p_{uk} := p_{uk} + \alpha[\sum_{u,i \in R}(r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik})q_{ik} - \lambda p_{uk}]$$
$$q_{ik} := q_{ik} + \alpha[\sum_{u,i \in R}(r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik})p_{uk} - \lambda q_{ik}]$$
$$b_u := b_u + \alpha[\sum_{u,i \in R}(r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik}) - \lambda b_u]$$
$$b_i := b_i + \alpha[\sum_{u,i \in R}(r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik}) - \lambda b_i]$$

Stochastic gradient descent:

$$p_{uk} := p_{uk} + \alpha[(r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik})q_{ik} - \lambda_1 p_{uk}]$$
$$q_{ik} := q_{ik} + \alpha[(r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik})p_{uk} - \lambda_2 q_{ik}]$$
$$b_u := b_u + \alpha[(r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik}) - \lambda_3 b_u]$$
$$b_i := b_i + \alpha[(r_{ui} - \mu - b_u - b_i - \sum_{k=1}^{k} p_{uk}q_{ik}) - \lambda_4 b_i]$$

Since the matrices P and Q matrix are two different matrices, each typically take different regularization parameters, such as λ_1 and λ_2.

References

[1] Le Hien and Nicolas Gillis. Algorithms for nonnegative matrix factorization with the kullback-leibler divergence, Oct 2020.

[2] C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the -divergence. Neural Computation, 23(9):2421–2456, 2011.

[3] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. Neural computation, 21:793–830, 10 2008.

[4] Quoc Tran-Dinh, Anastasios Kyrillidis, and Volkan Cevher. Composite self-concordant minimization. Journal of Machine Learning Research, 16(12):371–416, 2015.