Multiple Kernel Concept Factorization for Data Clustering

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

Abstract—The abstract goes here.

Index Terms—Computer Society, IEEEtran, journal, LaTeX, paper, template.

1 Introduction

THIS demo file is intended to serve as a "starter file" for IEEE Computer Society journal papers produced under LaTeX using IEEEtran.cls version 1.8a and later. I wish you the best of success.

mds
September 17, 2014

1.1 Subsection Heading Here

Subsection text here.

1.1.1 Subsubsection Heading Here

Subsubsection text here.

2 Related Work

2.1 NMF and CF

Non-negative Matrix Factorization (NMF) [xxx] is a matrix factorization algorithm that focuses on the analysis of data matrices whose elements are nonnegative. Given a data matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathcal{R}^{d \times n}$, each column of $\mathbf{X}$ is a sample vector. NMF aims to find two non-negative matrices $\mathbf{U} \in \mathcal{R}^{d \times n}$ and $\mathbf{U} \in \mathcal{R}^{d \times n}$ whose product can well approximate the original matrix $\mathbf{X}$. The optimal value of $\mathbf{U}$ and $\mathbf{V}$ can be found by solving the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}} \quad ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||^2, \quad \text{s.t.} \quad \mathbf{U} \geq 0, \mathbf{V} \geq 0. \tag{1}$$

It can be seen that each data vector $\mathbf{x}_i$ is approximated by a linear combination of the columns of $\mathbf{U}$, weighted by the components of $\mathbf{V}$, i.e. $\mathbf{x}_i = \sum_k \mathbf{u}_k v_{ik}$. Thus, $\mathbf{U}$ can be regarded as a set of basis and $\mathbf{V}$ can be regarded as the new representation of each data point in the new basis $\mathbf{U}$.

NMF can only be performed in the original feature space of the data points. In the case that the data are highly nonlinear distributed, it is desirable that we can kernelize NMF and apply the powerful idea of the kernel method. To achieve this goal, Xu and Gong [xxx] proposed an extension of NMF which is called Concept Factorization. In CF, each basis $\mathbf{u}_k$ is required to be a nonnegative linear combination of the sample vectors $\mathbf{x}$

$$\min_{\mathbf{U},\mathbf{V}} \quad ||\mathbf{X} - \mathbf{X}\mathbf{U}\mathbf{V}^T||^2, \quad \text{s.t.} \quad \mathbf{U} \geq 0, \mathbf{V} \geq 0. \tag{2}$$

It can be seen that the above concept factorization model can be easily kernelized by solving the following problem

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \quad & \mathrm{tr}(\mathbf{K}) - 2\mathrm{tr}(\mathbf{V}^T\mathbf{K}\mathbf{U}) + \mathrm{tr}(\mathbf{U}^T\mathbf{K}\mathbf{U}\mathbf{V}^T\mathbf{V}) \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{V} \geq 0. \end{aligned} \tag{3}$$

It has been shown that the optimal value of $\mathbf{U}$ and $\mathbf{V}$ in the kernel concept factorization model can be obtained by the following multiplicative update rules:

$$\mathbf{U}_{ij} = \mathbf{U}_{ij} \frac{(\mathbf{K}\mathbf{V})_{ij}}{(\mathbf{K}\mathbf{U}\mathbf{V}^T\mathbf{V})_{ij}} \tag{4}$$

$$\mathbf{V}_{ij} = \mathbf{V}_{ij} \frac{(\mathbf{K}\mathbf{U})_{ij}}{(\mathbf{V}\mathbf{U}^T\mathbf{K}\mathbf{U})_{ij}} \tag{5}$$

For the kernel matrix with negative entries, the multiplicative update rules become

$$\mathbf{U}_{ij} = \mathbf{U}_{ij} \frac{(\mathbf{K}\mathbf{V})_{ij} + \sqrt{(\mathbf{K}\mathbf{V})_{ij}^2 + 4\mathbf{P}_{ij}^+\mathbf{P}_{ij}^-}}{2\mathbf{P}_{ij}^+} \tag{6}$$

$$\mathbf{V}_{ij} = \mathbf{V}_{ij} \frac{(\mathbf{K}\mathbf{U})_{ij} + \sqrt{(\mathbf{K}\mathbf{U})_{ij}^2 + 4\mathbf{Q}_{ij}^+\mathbf{Q}_{ij}^-}}{2\mathbf{Q}_{ij}^+} \tag{7}$$

where

$$\mathbf{K}^+ = \frac{|\mathbf{K}| + \mathbf{K}}{2} \tag{8}$$

$$\mathbf{K}^- = \frac{|\mathbf{K}| - \mathbf{K}}{2} \tag{9}$$

$$\mathbf{P}^+ = \mathbf{K}^+\mathbf{U}\mathbf{V}^T\mathbf{V} \tag{10}$$

$$\mathbf{P}^- = \mathbf{K}^-\mathbf{U}\mathbf{V}^T\mathbf{V} \tag{11}$$

$$\mathbf{Q}^+ = \mathbf{V}\mathbf{U}^T\mathbf{K}^+\mathbf{U} \tag{12}$$

$$\mathbf{Q}^- = \mathbf{V}\mathbf{U}^T\mathbf{K}^-\mathbf{U} \tag{13}$$

- *M. Shell is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332.*
E-mail: see http://www.michaelshell.org/contact.html
- *J. Doe and J. Doe are with Anonymous University.*

Manuscript received April 19, 2005; revised September 17, 2014.

2.2 Multiple Kernel Clustering

3 Globalized Multiple Kernel Concept Factorization

3.1 Motivation and Formulation

The proposed method above only works for single kernel data clustering. However, one of the central problems with kernel methods in general is that it is often unclear which kernel is the most suitable for a particular task. [more diffcult for unsupervised MKL].

In this section, we extend kernel concept factorization to automatically learn an appropriate kernel from the convex linear combination of several pre-computed kernel matrices within the multiple kernel learning framework [?].

Suppose there are altogether m different kernel functions $\{\mathcal{K}^i\}_{i=1}^m$ available for the clustering task in hand. Accordingly, there are m different associated feature spaces denoted as $\{\mathcal{H}\}_i^m$. To combine these kernels and also ensure that the resulted kernel still satisfies Mercer condition, we consider a nonnegative combination of these feature maps, ϕ' , that is,

$$\phi'(\mathbf{x}) = \sum_{i=1}^m w_i \phi_i(\mathbf{x}) \quad \text{with } w_i \geq 0. \tag{14}$$

Unfortunately, as these implicit mappings do not necessarily have the same dimensionality, such a linear combination may be unrealistic. Hence, we construct an augmented Hilbert space $\tilde{\mathcal{H}} = \oplus_{i=1}^m \mathcal{H}^i$ by concatenating all feature spaces $\phi_{\mathbf{w}}(\mathbf{x}) = [w_1\phi_1(\mathbf{x}); w_2\phi_2(\mathbf{x}); \ldots; w_m\phi_m(\mathbf{x})]^T$ with different weight $w_i (w_i \geq 0)$, or equivalently the importance factor for kernel function $\mathcal{K}^i$. It can be verified that clustering in feature space $\tilde{\mathcal{H}}$ is equivalent to employing a combined kernel function [?]

$$\tilde{\mathcal{K}}(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^m w_i \mathcal{K}^i(\mathbf{x}, \mathbf{z}). \tag{15}$$

It is known that the convex combination, with $\mathbf{w}(w_i \geq 0)$, of the positive semi-definite kernel matrices $\{\mathbf{K}^i\}_{i=1}^m$ is still a positive semi-definite kernel matrix. By replacing the single kernel in Eq. (**??**) with the combined kernel, we propose a new Globalized Multiple Kernel Concept Factorization (GMKCF) method by solving:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{w}} \quad & \mathrm{tr}(\mathbf{K_w}) - 2\mathrm{tr}(\mathbf{V}^T\mathbf{K_w}\mathbf{U}) + \mathrm{tr}(\mathbf{U}^T\mathbf{K_w}\mathbf{U}\mathbf{V}^T\mathbf{V}) \\ & + \lambda||\mathbf{V}^T\mathbf{V} - \mathbf{I}||^2 \qquad (16) \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{U}\mathbf{1}_k = \mathbf{1}_n, \mathbf{V} \geq 0, \mathbf{w} \geq 0, \sum_{i=1}^m w_i^\gamma = 1. \end{aligned}$$

where $\lambda > 0$ is a parameter to control the orthogonality condition. Usually, λ should be large enough to insure the orthogonality satisfied and we fix it as 10^6 in our experiments.

The main purpose of imposing additional regularization on $\mathbf{V}$ is to guarantee the uniqueness of our solution. It has been pointed out that [?] the optimal solution obtained by the update rules in Eq. (**??**) for Eq. (**??**) is not unique: if $\{\mathbf{U}^*, \mathbf{V}^*\}$ is the optimal solution for Eq. (xxx), then $\{\mathbf{U}^*\mathbf{D}, \mathbf{V}^*\mathbf{D}^{-1}\}$ will also be a solution with the same objective function value for any positive diagnoal matrix $\mathbf{D}$. To eliminate this uncertainty, [?] proposed to use normalization on columns of $\mathbf{U}$ and $\mathbf{V}$ in each iteration during the optimization, i.e., $\mathbf{U} = \mathbf{U}[\mathrm{diag}(\mathbf{U}^T\mathbf{K}\mathbf{U})]^{-1/2}$ and $\mathbf{V} = \mathbf{V}[\mathrm{diag}(\mathbf{U}^T\mathbf{K}\mathbf{U})]^{1/2}$. However, this is not a principled way to solve the problem. It is necessary to explicitly include the orthonormal regularization in our framework to avoid such an adhoc step.

3.2 Algorithm

The optimization problem in Eq. (xx) is not convex in all variables together, but convex in them separately. In the following, we introduce an iterative algorithm based on block coordinate descent to solve it. We separately update the value of $\mathbf{w}$, $\mathbf{U}$ and $\mathbf{V}$, while holding the other variables as constant. Thus, a local minima can be expected by solving a sequence of convex optimization problems.

3.2.1 *Optimizing w.r.t.* $\mathbf{U}$ *when* $\mathbf{V}$ *and* $\mathbf{w}$ *are fixed*

The minimization of the objective function in Eq. (16) with respect to $\mathbf{U}$ can be decomposed into solving the following problem:

$$\begin{aligned} \min_{\mathbf{U}} \quad & \mathrm{tr}(\mathbf{U}^T\mathbf{K_w}\mathbf{U}\mathbf{V}^T\mathbf{V}) - 2\mathrm{tr}(\mathbf{U}^T\mathbf{K_w}\mathbf{V}) \qquad (17) \\ \text{s.t.} \quad & \mathbf{U} \geq 0. \end{aligned}$$

It can be seen that the above problem is the same as kernel concept factorization with the estimated kernel $\mathbf{K_w}$, thus, the optimal value can be obtained by using the multiplicative rule:

$$\mathbf{U}_{ij} = \mathbf{U}_{ij} \frac{(\mathbf{K}\mathbf{V})_{ij}}{(\mathbf{K}\mathbf{U}\mathbf{V}^T\mathbf{V})_{ij}} \tag{18}$$

For the kernel $\mathbf{K_w}$ with negative entries, denote $\mathbf{K_w^+} = \frac{|\mathbf{K_w}| + \mathbf{K_w}}{2}$, $\mathbf{K_w^-} = \frac{|\mathbf{K_w}| - \mathbf{K_w}}{2}$, we have

$$\mathbf{U}_{ij} = \mathbf{U}_{ij} \frac{(\mathbf{K_w}\mathbf{V})_{ij}^2 + \sqrt{(\mathbf{K_w}\mathbf{V})_{ij} + 4\mathbf{P}_{ij}^+\mathbf{P}_{ij}^-}}{2\mathbf{P}_{ij}^+} \tag{19}$$

where $\mathbf{P_w^+} = \mathbf{K_w^+}\mathbf{U}\mathbf{V}^T\mathbf{V}$, $\mathbf{P_w^-} = \mathbf{K_w^-}\mathbf{U}\mathbf{V}^T\mathbf{V}$.

3.2.2 *Optimizing w.r.t.* $\mathbf{V}$ *when* $\mathbf{U}$ *and* $\mathbf{w}$ *are fixed*

The minimization of the objective function in Eq. (16) with respect to $\mathbf{U}$ can be decomposed into solving the following problem:

$$\begin{aligned} \min_{\mathbf{V}} \quad & \mathrm{tr}(\mathbf{V}^T\mathbf{U}^T\mathbf{K_w}\mathbf{U}\mathbf{V}) - 2\mathrm{tr}(\mathbf{V}^T\mathbf{K_w}\mathbf{U}) + \lambda||\mathbf{V}^T\mathbf{V} - \mathbf{I}||^2 \\ \text{s.t.} \quad & \mathbf{V} \geq 0. \qquad (20) \end{aligned}$$

Defining $\mathbf{A}^+ = \mathbf{U}^T\mathbf{K_w^+}\mathbf{U}$, $\mathbf{A}^- = \mathbf{U}^T\mathbf{K_w^-}\mathbf{U}$, $\mathbf{B}^+ = \mathbf{K_w^+}\mathbf{U}$ and $\mathbf{B}^- = \mathbf{K_w^-}\mathbf{U}$, we have

$$\begin{aligned} \min_{\mathbf{V}} \quad & \mathrm{tr}(\mathbf{V}(\mathbf{A}^+ - \mathbf{A}^-)\mathbf{V}^T) - 2\mathrm{tr}(\mathbf{V}^T(\mathbf{B}^+ - \mathbf{B}^-)) \\ & + \lambda||\mathbf{V}^T\mathbf{V} - \mathbf{I}||^2 \qquad (21) \\ \text{s.t.} \quad & \mathbf{V} \geq 0. \end{aligned}$$

The objective function in Eq. (20) is a fourth-order nonconvex function with respect to the entries of $\mathbf{V}$, and has multiple local minima. For this type of problem, it is difficult to find a global minimum; thus a good convergence property we can expect is that every limit point is a stationary point [?]. We can directly apply standard gradient search algorithms, which lead to stationary point solutions; however, they suffer from either slow convergencerate or expensive computation cost. Thus, it is not a trivial problem to minimize Eq. (20) efficiently.

In the next, we use the auxiliary function approach **[?]** to derive an efficient updating rule for Eq. (20) and prove its convergence. The basic idea is to construct an auxiliary function which is a convex upper bound for the original objective function based on the solution obtained from the previous iteration. Then, a new solution to the current iteration is obtained by minimizing this upper bound. Here we first introduce the definition of auxiliary function.

Definition 3.2.3. *[?] g is an auxiliary function for f if the conditions*

$$g(x, x^t) \geq f(x) \textit{ and } g(x, x) \geq f(x)$$

are satisfied.

Lemma 3.2.4. *[?] If g is an auxiliary function for f, then f is non-increasing under the update*

$$x^{t+1} = \arg\min_x g(x, x^t).$$

Proof. $f(x^{t+1}) \leq g(x^{t+1}, x^t) \leq g(x^t, x^t) = f(x^t)$.

In the following Theorem 3.2.5, we propose an auxiliary function for the objective function in Eq. (20) and derive the multiplicative update rule to obtain its global minimum.

Theorem 3.2.5. *Let*

$$\begin{aligned} f(\mathbf{V}) =& \mathrm{tr}(\mathbf{V}\mathbf{A}^+\mathbf{V}^T) - \mathrm{tr}(\mathbf{V}\mathbf{A}^-\mathbf{V}^T) \\ & -2\mathrm{tr}(\mathbf{V}^T\mathbf{B}^+) + 2\mathrm{tr}(\mathbf{V}^T\mathbf{B}^-) + \lambda||\mathbf{V}^T\mathbf{V} - \mathbf{I}||^2 \end{aligned} \tag{22}$$

Then the following function

$$\begin{aligned} & g(\mathbf{V}, \mathbf{V}^t) \\ &= \frac{1}{2}\sum_{ij}(\mathbf{V}^t\mathbf{A}^+)_{ij}\mathbf{V}^t_{ij}(\frac{\mathbf{V}^4_{ij}}{(\mathbf{V}^t_{ij})^4} + 1) \\ &- \sum_{ijk}\mathbf{A}^-_{jk}\mathbf{V}^t_{ij}\mathbf{V}^t_{ik}(1 + \log\frac{\mathbf{V}_{ij}\mathbf{V}_{ik}}{\mathbf{V}^t_{ij}\mathbf{V}^t_{ik}}) \\ &- \sum_{ij}\mathbf{B}^+_{ij}\mathbf{V}^t_{ij}(1 + \log\frac{\mathbf{V}_{ij}}{\mathbf{V}^t_{ij}}) \\ &+ \frac{1}{4}\sum_{ij}\mathbf{B}^-_{ij}\mathbf{V}^t_{ij}\left(\frac{\mathbf{V}^4_{ij}}{(\mathbf{V}^t_{ij})^4} + 1\right) + \sum_{ij}\mathbf{B}^-_{ij}\frac{(\mathbf{V}^t_{ij})^2}{2\mathbf{V}^t_{ij}} \\ &+ k\lambda - 2\lambda\sum_{ijk}\mathbf{I}_{jk}\mathbf{V}^t_{ij}\mathbf{V}^t_{ik}(1 + \log\frac{\mathbf{V}_{ij}\mathbf{V}_{ik}}{\mathbf{V}^t_{ij}\mathbf{V}^t_{ik}}) \\ &+ \frac{\lambda}{2}\sum_{ijk}[(\mathbf{V}^t)^T(\mathbf{V}^t)]_{jk}\mathbf{V}^t_{ij}\mathbf{V}^t_{ik}\left(\frac{\mathbf{V}^4_{ij}}{(\mathbf{V}^t_{ij})^4} + \frac{\mathbf{V}^4_{ik}}{(\mathbf{V}^t_{ik})^4}\right) \end{aligned} \tag{23}$$

is an auxiliary function for $f(\mathbf{V})$. Furthermore, it is a convex function with respect to $\mathbf{V}$ and its global minimum is

$$\mathbf{V}_{ij} = \mathbf{V}^t_{ij}\left(\frac{[\mathbf{V}^t\mathbf{A}^- + \mathbf{B}^+ + 2\lambda\mathbf{V}^t]_{ij}}{[\mathbf{V}^t\mathbf{A}^+ + \mathbf{B}^- + 2\lambda\mathbf{V}^t(\mathbf{V}^t)^T\mathbf{V}^t]_{ij}}\right)^{\frac{1}{4}} \tag{24}$$

Proof. See Appendix A. □

Now, we prove the convergence of the update rule in Eq. (24) can be guaranteed by the following Theorem 3.2.6.

Theorem 3.2.6. *Updating $\mathbf{V}$ using Eq.* (24) *will monotonically decrease the value of the objective in Eq.* (20)*, hence it converges.*

Proof. By Lemma 3.2.4 and Theorem 3.2.5, we can get that $f(\mathbf{V}^0) \geq g(\mathbf{V}, \mathbf{V}^0) \geq f(\mathbf{V}^1) \geq \cdots$, So $f(\mathbf{V})$ is monotonically decreasing. Since $f(\mathbf{V})$ is obviously belower bounded, we prove this theorem. □

3.2.7 Optimizing w.r.t. $\mathbf{w}$ when $\mathbf{U}$ and $\mathbf{V}$ are fixed

By defining $\mathbf{e} \in \mathcal{R}^{m\times 1}$ with

$$e_i = \mathrm{tr}(\mathbf{K}^i) - 2\mathrm{tr}(\mathbf{V}^T\mathbf{K}^i\mathbf{U}) + \mathrm{tr}(\mathbf{U}^T\mathbf{K}^i\mathbf{U}\mathbf{V}^T\mathbf{V}), \tag{25}$$

the optimization of Eq. (xxx) with respect to $\mathbf{w}$ can be simplified as solving the following problem:

$$\min_{\mathbf{w}} \quad \sum_{i=1}^m w_i^2 e_i, \quad \text{s.t.} \quad \sum_{i=1}^m w_i = 1, w_i \geq 0. \tag{26}$$

The Lagrange function of Eq. (26) is $\mathcal{J}(\mathbf{w}) = \sum_{i=1}^m w_i^2 e_i + \lambda(1 - \sum_{i=1}^m w_i)$. By using the KKT condition $\frac{\partial\mathcal{J}(\mathbf{w})}{\partial w_i} = 0$ and the constraint $\sum_{i=1}^m w_i = 1$, the optimal solution of $\mathbf{w}$ can be obtained by

$$w_i = \frac{\frac{1}{e_i}}{\sum_{j=1}^m \frac{1}{e_j}}. \tag{27}$$

In summary, we present the iterative updating algorithm of optimizing Eq. (16) in Algorithm 1.

Algorithm 1 The algorithm of GMKCF

```
Input: A set of kernels ${K^i}_{i=1}^m$, the desired number of cluster $c$.
  Initialize the indicator matrix $Z$ randomly, such that $Z$ satisfies $z_{ij} = \{0,1\}$ and $\sum_j z_{ij} = 1$;
  Initialize the kernel weight $w_i = 1/m$ for each kernel;
  Initialize the diagonal matrix $D = I_n$, where $I_n$ is the identity matrix;
  repeat
    Update the association matrix $\mathbf{U}$ by using (18) or (19);
    Update the projection matrix $\mathbf{V}$ by Eq. (24);
    Update the kernel weight $\mathbf{w}$ by Eq. (27);
    Update the estimated kernel $\mathbf{K_w}$ by Eq. (15);
  until Converges
Output: association matrix $\mathbf{U}$, projection matrix $\mathbf{V}$, and the kernel weights $\mathbf{w}$.
```

3.2.8 Convergence and Complexity Analysis

Note that the objective function in Eq. (16) is nonincreasing under the derived updating rules. Since the objective function is bounded below, the convergence of the algorithm is guaranteed.

In the following, we give the complexity analysis of the optimization algorithm. Initially, we need to compute m kernel matrices $\{\mathbf{K}^i\}_{i=1}^m$, whose cost is generally $\mathbf{O}(mn^2d)$, where n is the number of samples and d is the number of features. The cost of each iteration is given by

- Updating of $\mathbf{U}$: the computational cost of $\mathbf{P}^+$ and $\mathbf{P}^+$ is $\mathbf{O}(n^2k + k^2n)$, and the cost of evaluating Eq. (19) is $\mathbf{O}(n^2k)$.

- Updating of $\mathbf{V}$: the computational cost of $\mathbf{A}^+$ and $\mathbf{A}^-$ is $\mathbf{O}(n^2k + k^2n)$, the computational cost of $\mathbf{B}^+$ and $\mathbf{B}^-$ is $\mathbf{O}(n^2k)$, the cost of evaluating Eq. (24) is $\mathbf{O}(nk^2)$.
- Updating of $\mathbf{w}$: the computational cost of $\mathbf{e}$ is $\mathbf{O}(m(n^2k + k^2n))$.
- Updating of $\mathbf{K_w}$: the computational cost of $\mathbf{K_w}$ is n^2m.

Suppose the multiplicative updates stops after t iterations, the overall cost for GMKCF is $\mathbf{O}(mn^2d + n^2t(k+m))$have the same computational complexity by using the big $\mathbf{O}$ notation when dealing with the high-dimensional data.

4 Localized Multiple Kernel Concept Factorization

4.1 Motivation

4.2 Formulation

In our localized combination approach, the mapping function is represented as $\phi_{\mathbf{W}}(\mathbf{x}_i) = [W_{i1}\phi_1(\mathbf{x}_i); W_{i2}\phi_2(\mathbf{x}_i); \ldots; W_{im}\phi_m(\mathbf{x}_i)]^T$. Thus we get locally combined kernel function:

$$\begin{aligned} K_W(\mathbf{x}_i, \mathbf{x}_j) &= \langle \phi_{\mathbf{W}}(\mathbf{x}_i), \phi_{\mathbf{W}}(\mathbf{x}_j) \rangle \\ &= \sum_{t=1}^{m} \langle W_{it}\phi_t(\mathbf{x}_i), W_{jt}\phi_t(\mathbf{x}_j) \rangle \\ &= \sum_{t=1}^{m} W_{it}W_{jt}K_t(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \tag{28}$$

Theorem 4.2.1. *$K_W(\cdot,\cdot)$ defined in Eq.* (28) *is a positive semi-definite kernel function.*

Proof. To prove that $\mathbf{K_W}$ is a positive semi-definite kernel function, we introduce the following lemma:

Lemma 4.2.2. *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ be a symmetric function, the necessary and sufficient condition of that $K(x,z)$ is a positive semi-definite kernel function is that the Gram Matrix of $K(x,z)$ for any $x_i \in \mathcal{X}$: $\mathcal{K} = [K(x_i, x_j)]_{m\times m}$ is a positive semi-definite matrix.*

Proof. □

According to Lemma 4.2.2, we just need to prove that for any $x_1, ..., x_n$, the Gram Matrix of K_W is positive semi-definite. Let $\mathbf{K_W}$ be the Gram Matrix of $K_W(\cdot,\cdot)$ and $\mathbf{K_t}$ be the Gram Matrix of $K_t(\cdot,\cdot)$, we just need to prove that $(\mathbf{w}_t\mathbf{w}_t^T) \otimes \mathbf{K}_t$ is positive semi-definite, where $\mathbf{w}_t$ is the t-th column of $\mathbf{W}$ and $\otimes$ is the element-wise product operator.

Since $K_t(\cdot,\cdot)$ is positive semi-definite kernel function, according to Lemma 4.2.2, $\mathbf{K_t}$ is positive semi-definite. Thus all the leading principal minors of $\mathbf{K_t}$ are all positive. Now consider the j-th leading principal minor of $\mathbf{K_W}$:

$$\begin{aligned} &\begin{vmatrix} Kw(x_1,x_1) & \cdots & Kw(x_1,x_j) \\ \vdots & \vdots & \vdots \\ Kw(x_j,x_1) & \cdots & Kw(x_j,x_j) \end{vmatrix} \\ =&\left|\begin{pmatrix} W_{1t}W_{1t} & \cdots & W_{1t}W_{jt} \\ \vdots & \vdots & \vdots \\ W_{jt}W_{1t} & \cdots & W_{jt}W_{jt} \end{pmatrix} \otimes \begin{pmatrix} K_t(x_1,x_1) & \cdots & K_t(x_1,x_j) \\ \vdots & \vdots & \vdots \\ K_t(x_j,x_1) & \cdots & K_t(x_j,x_j) \end{pmatrix}\right| \\ =&W_{1t}^2W_{2t}^2\cdots W_{jt}^2 \begin{vmatrix} K_t(x_1,x_1) & \cdots & K_t(x_1,x_j) \\ \vdots & \vdots & \vdots \\ K_t(x_j,x_1) & \cdots & K_t(x_j,x_j) \end{vmatrix} \end{aligned} \tag{29}$$

where $|\cdot|$ is determinant of a matrix.

Since $\mathbf{K_t}$ is positive semi-definite, we have

$$\begin{vmatrix} K_t(x_1,x_1) & \cdots & K_t(x_1,x_j) \\ \vdots & \vdots & \vdots \\ K_t(x_j,x_1) & \cdots & K_t(x_j,x_j) \end{vmatrix} \geq 0 \tag{30}$$

Thus, we obtain:

$$\begin{vmatrix} K_W(x_1,x_1) & \cdots & K_W(x_1,x_j) \\ \vdots & \vdots & \vdots \\ K_W(x_j,x_1) & \cdots & K_W(x_j,x_j) \end{vmatrix} \geq 0 \tag{31}$$

Eq. (31) holds for any $1 \leq j \leq n$, thus K_W is positive semi-definite and $K_W(\cdot,\cdot)$ is a positive semi-definite kernel function. □

Taking Eq.(28) into Eq. (16), we get the formulation of localized multiple kernel concept factorization:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{W}} \quad & \mathrm{tr}(\mathbf{K_W}) - 2\mathrm{tr}(\mathbf{V}^T\mathbf{K_W}\mathbf{U}) + \mathrm{tr}(\mathbf{U}^T\mathbf{K_W}\mathbf{U}\mathbf{V}^T\mathbf{V}) \\ & + \lambda||\mathbf{V}^T\mathbf{V} - \mathbf{I}||^2 \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{V} \geq 0, \mathbf{W} \geq 0, \mathbf{W}\mathbf{1}_m = \mathbf{1}_n \end{aligned} \tag{32}$$

4.3 Algorithm

Similar to the optimization of Eq.(16), we also introduce a block coordinate descent algorithm to solve it.

4.3.1 Optimizing w.r.t. $\mathbf{U}$ when $\mathbf{V}$ and $\mathbf{W}$ are fixed

When fixing $\mathbf{V}$ and $\mathbf{W}$, it is the same to optimize $\mathbf{U}$ as Eq. (19). Substitute $\mathbf{K_w}$ in Eq. (19) with $\mathbf{K_W}$, we obtain:

$$\mathbf{U}_{ij} = \mathbf{U}_{ij}\frac{(\mathbf{K_W}\mathbf{V})_{ij}^2 + \sqrt{(\mathbf{K_W}\mathbf{V})_{ij} + 4\mathbf{P}_{ij}^+\mathbf{P}_{ij}^-}}{2\mathbf{P}_{ij}^+} \tag{33}$$

where $\mathbf{P}^+ = \mathbf{K}_\mathbf{W}^+\mathbf{U}\mathbf{V}^T\mathbf{V}$, $\mathbf{P}^- = \mathbf{K}_\mathbf{W}^-\mathbf{U}\mathbf{V}^T\mathbf{V}$, and $\mathbf{K}_\mathbf{W}^+ = \frac{|\mathbf{K_W}|+\mathbf{K_W}}{2}$, $\mathbf{K}_\mathbf{W}^- = \frac{|\mathbf{K_W}|-\mathbf{K_W}}{2}$.

4.3.2 Optimizing w.r.t. $\mathbf{V}$ when $\mathbf{U}$ and $\mathbf{W}$ are fixed

Similar to Eq. (24), we update $\mathbf{V}$ as follows:

$$\mathbf{V}_{ij} = \mathbf{V}_{ij}^t\left(\frac{[\mathbf{V}^t\mathbf{A}^- + \mathbf{B}^+ + 2\lambda \oslash \mathbf{V}^t]_{ij}}{[\mathbf{V}^t\mathbf{A}^+ + \mathbf{B}^- + 2\lambda\mathbf{V}^t(\mathbf{V}^t)^T\mathbf{V}^t]_{ij}}\right)^{\frac{1}{4}} \tag{34}$$

where $\mathbf{A}^+ = \mathbf{U}^T\mathbf{K}_\mathbf{W}^+\mathbf{U}$, $\mathbf{A}^- = \mathbf{U}^T\mathbf{K}_\mathbf{W}^-\mathbf{U}$, $\mathbf{B}^+ = \mathbf{K}_\mathbf{W}^+\mathbf{U}$ and $\mathbf{B}^- = \mathbf{K}_\mathbf{W}^-\mathbf{U}$.

4.3.3 *Optimizing w.r.t.* $\mathbf{W}$ *when* $\mathbf{U}$ *and* $\mathbf{V}$ *are fixed*

When fix $\mathbf{U}$ and $\mathbf{V}$, Eq.(32) can be simplified as:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \operatorname{tr}\left(\mathbf{K}_{\mathbf{W}}(\mathbf{I} - 2\mathbf{V}^T\mathbf{U} + \mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T)\right) \\ \text{s.t.} \quad & \mathbf{W} \geq 0, \mathbf{W}\mathbf{1}_m = \mathbf{1}_n \end{aligned} \tag{35}$$

Denote $\mathbf{G} = (\mathbf{I} - 2\mathbf{V}^T\mathbf{U} + \mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T)$, and substitute Eq.(28) in Eq.(35), we have:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{t=1}^{m} \mathbf{w}_t^T(\mathbf{K}^t \otimes \mathbf{G})\mathbf{w}_t \\ \text{s.t.} \quad & \mathbf{W} \geq 0, \mathbf{W}\mathbf{1}_m = \mathbf{1}_n. \end{aligned} \tag{36}$$

To optimize Eq. (36), we can apply Proximal Gradient Descent [?] to solve $\mathbf{W}$. More specifically, we denote $\mathbf{M}^t = \mathbf{K}_t \otimes \mathbf{G}$ and $f(\mathbf{W}) = \sum_{t=1}^{m} \mathbf{w}_t^T \mathbf{M} \mathbf{w}_t$, then linearize $f(\mathbf{W})$ at $\mathbf{W}^k$ and add a proximal term:

$$g(\mathbf{W}, \mathbf{W}^k) = f(\mathbf{W}^k) + \langle \nabla f(\mathbf{W}^k), \mathbf{W} - \mathbf{W}^k \rangle + \frac{\mu}{2}\|\mathbf{W} - \mathbf{W}^k\|_F^2 \tag{37}$$

where ∇f is the gradient of $f(\cdot)$, and $\mu > L(f)$ where $L(f)$ is Lipschitz constant of $f(\cdot)$.

Then we update $\mathbf{W}$ by solving:

$$\mathbf{W}^{k+1} = \arg \min_{\mathbf{W} \geq 0, \mathbf{W}\mathbf{1}_m = \mathbf{1}_n} \|\mathbf{W} - \left(\mathbf{W}^k - \frac{1}{\mu}\nabla f(\mathbf{W}^k)\right)\|_F^2 \tag{38}$$

Let $\mathbf{H} = \mathbf{W}^k - \frac{1}{\mu}\nabla f(\mathbf{W}^k)$, to get $\mathbf{W}^{k+1}$, we need to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{W} - \mathbf{H}\|_F^2 \\ \text{s.t.} \quad & \mathbf{W} \geq 0, \mathbf{W}\mathbf{1}_m = \mathbf{1}_n \end{aligned} \tag{39}$$

Eq. (39) is row-decoupled and can be decomposed into n subproblem. Each subproblem is a well-known Euclidean Projection onto Simplex problem and can be efficiently solved by root finding algorithm [?]. Algorithm 2 shows the process.

Algorithm 2 The optimization algorithm of Euclidean Projection onto Simplex

```
Input: $\mathbf{h}$
  sort $\mathbf{h}$ into $\mathbf{b}$ where $b_1 \geq b_2 \geq, ..., b_n$
  find $\rho = \max\{1 \leq j \leq n : b_j + \frac{1}{j}(1 - \sum_{i=1}^{j} b_i) > 0\}$
  define $z = \frac{1}{\rho}(1 - \sum_{i=1}^{\rho} b_i)$
Output: $\mathbf{w}$ with $w_j = \max\{h_j + z, 0\}, j = 1, ..., n$
```

According to [?], Algorithm 2 provides the global optima of (39). We use the result of Algorithm 2 to update $\mathbf{W}$.

Although Proximal Gradient Descent can be used to solve Eq. (36), the converge rate is slow, i.e., $O(\frac{1}{\epsilon})$ []. Here to achieve more efficient optimization, we apply Nesterov's method [] to accelerate the proximal gradient descent, which owns the convergence rate as $O(\frac{1}{\sqrt{\epsilon}})$. We construct a linear combination of $\mathbf{W}^k$ and $\mathbf{W}^{k+1}$ to update $\mathbf{Y}^{k+1}$ as follows:

$$\mathbf{Y}^{k+1} = \mathbf{W}^k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\mathbf{W}^{k+1} - \mathbf{W}^k) \tag{40}$$

Then we substitute $\mathbf{W}^k$ in Eq. (38) with $\mathbf{Y}^k$,

$$\mathbf{W}^{k+1} = \underset{\mathbf{W} \geq 0, \mathbf{W}\mathbf{1}_m = \mathbf{1}_n}{\arg\min} \|\mathbf{W} - \left(\mathbf{Y}^k - \frac{1}{\mu}\nabla f(\mathbf{Y}^k)\right)\|_F^2 \tag{41}$$

Eq. (41) can be solved by Algorithm 2 as discussed before. Algorithm 3 shows the process of the accelerated Proximal Gradient Descent.

Algorithm 3 The accelerated Proximal Gradient Descent algorithm

```
Input: The initial constant $L_0$, $a_1 = 1$, $\mathbf{Z}_1 = \mathbf{W}^0$.
  Set $t = 0$, $\bar{L}_{\text{candi}} = L_0$
  repeat
    Set $\bar{L}_{\text{candi}} = L_{\text{old}}$;
    Update $\bar{\mathbf{Z}}_{\text{candi}} = p_{\bar{L}_{\text{candi}}}(\mathbf{W}_{\text{old}})$ using Algorithm xxx;
    While $f(\bar{\mathbf{Z}}_{\text{candi}}) > g_{\bar{L}}(\bar{\mathbf{Z}}_{\text{candi}}, \mathbf{Z}_t)$, do
      Set $\bar{L}_{\text{candi}} = \gamma\bar{L}_{\text{candi}}$;
      Update $\bar{\mathbf{W}}_{\text{candi}} = p_{\bar{L}_{\text{candi}}}(\mathbf{Z}_t)$ using Algorithm xxx;
    end while
    Set $L_t = \bar{L}_{\text{candi}}$;
    Set $\mathbf{W}_t = p_{L_t}(\mathbf{Z}_t)$;
    Set $a_{t+1} = \frac{1+\sqrt{1+4a_t^2}}{2}$;
    Set $\mathbf{Z}_{t+1} = \mathbf{W}_t + (\frac{a_t - 1}{a_{t+1}})(\mathbf{W}_t - \mathbf{W}_{t-1})$;
  until Converges
Output: $\mathbf{U}_t$.
```

The convergence of this algorithm is stated in the following theorem.

Theorem 4.3.4. *[] Let $\mathbf{W}^k$ be the sequence generated by Algorithm 3, then for any $k \geq 1$, we have*

$$f(\mathbf{W}^k) - f(\mathbf{W}^*) \leq \frac{2\gamma L\|\mathbf{W}^1 - \mathbf{W}^*\|_F^2}{(k+1)^2}, \tag{42}$$

where L is the Lipschitz constant of the gradient of $f(\mathbf{W})$, and $\mathbf{W}^* = \arg\min_{\mathbf{W}} f(\mathbf{W})$.

It is easy to verify that $f(\mathbf{W})$ is Lipschitz continuous. Thus Theorem 4.3.4 shows that the convergence rate of the accelerated proximal gradient descent method is $O(\frac{1}{\sqrt{\epsilon}})$.

The whole algorithm of localized multiple kernel concept factorization is summarized in Algorithm 4.

Algorithm 4 The algorithm of LMKCF

```
Input: A set of kernels $\{K^i\}_{i=1}^m$, the desired number of
  cluster $c$.
  Initialize the kernel weight $w_{ij} = 1/m$ for each kernel;
  Initialize the diagonal matrix $D = I_n$, where $I_n$ is the
  identity matrix;
  repeat
    Update the association matrix $\mathbf{U}$ by using (33);
    Update the projection matrix $\mathbf{V}$ by Eq. (34);
    Update the kernel weight $\mathbf{W}$ by Algorithm 3;
    Update the estimated kernel $\mathbf{K}_{\mathbf{W}}$ by Eq. (28);
  until Converges
Output: association matrix $\mathbf{U}$, projection matrix $\mathbf{V}$, and the
  kernel weights $\mathbf{W}$.
```

4.3.5 *Convergence and Complexity*

The objective function in Eq. (32) is nonincreasing under the derived updating rules. Since the objective function is bounded below, the convergence of the algorithm is guaranteed.

In the following, we give the complexity analysis of the optimization algorithm. Initially, we need to compute m kernel matrices $\{\mathbf{K}^i\}_{i=1}^m$, whose cost is generally $\mathcal{O}(mn^2d)$, where n is the number of samples and d is the number of features. The cost of each iteration is given by

- Updating of $\mathbf{U}$: the computational cost of $\mathbf{P}^+$ and $\mathbf{P}^+$ is $\mathcal{O}(n^2k + k^2n)$and, the cost of evaluating Eq. (19) is $\mathcal{O}(n^2k)$.
- Updating of $\mathbf{V}$: the computational cost of $\mathbf{A}^+$ and $\mathbf{A}^-$ is $\mathcal{O}(n^2k + k^2n)$, the computational cost of $\mathbf{B}^+$ and $\mathbf{B}^-$ is $\mathcal{O}(n^2k)$, the cost of evaluating Eq. (24) is $\mathcal{O}(nk^2)$.
- Updating of $\mathbf{W}$: the computational cost of compute the gradient is $O(n^2m)$, the cost of Euclidean projection is $O(nmlogm)$.
- Updating of $\mathbf{K_w}$: the computational cost of $\mathbf{K_w}$ is n^2m.

5 Conclusion

The conclusion goes here.

6 Experiments

6.1 Evaluation Metrics

To evaluate their performance, we compare the generated clusters with the ground truth by computing the following two performance measures.

Clustering accuracy (ACC). The first performance measure is the clustering accuracy, which discovers the one-to-one relationship between clusters and classes. Given a point $\mathbf{x}_i$, let p_i and q_i be the clustering result and the ground truth label, respectively. The ACC is defined as follows:

$$\text{ACC} = \frac{1}{n}\sum_{i=1}^{n}\delta(q_i, map(p_i)), \tag{43}$$

where n is the total number of samples and $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise, and $map(\cdot)$ is the permutation mapping function that maps each cluster index to a true class label. The best mapping can be found by using the Kuhn-Munkres algorithm [**?**]. The greater clustering accuracy means the better clustering performance.

Normalized mutual information (NMI). Another evaluation metric that we adopt here is the normalized mutual information, which is widely used for determining the quality of clustering. Let $\mathcal{C}$ be the set of clusters from the ground truth and $\mathcal{C}'$ obtained from a clustering algorithm. Their mutual information $MI(\mathcal{C}, \mathcal{C}')$ is defined as follows:

$$MI(\mathcal{C}, \mathcal{C}') = \sum_{c_i\in\mathcal{C}, c'_j\in\mathcal{C}'} p(c_i, c'_j)\log\frac{p(c_i, c'_j)}{p(c_i)p(c'_j)}, \tag{44}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a data point arbitrarily selected from the data set belongs to the cluster c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected data point belongs to the cluster c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information as follows:

$$NMI(\mathcal{C}, \mathcal{C}') = \frac{MI(\mathcal{C}, \mathcal{C}')}{\max(H(\mathcal{C}), H(\mathcal{C}'))}, \tag{45}$$

where $H(\mathcal{C})$ and $H(\mathcal{C}')$ are the entropies of $\mathcal{C}$ and $\mathcal{C}'$, respectively. Again, a larger NMI indicates a better performance.

Appendix A

Lemma A.0.1. *For any nonnegative matrices* $\mathbf{F} \in \mathcal{R}^{n\times n}$, $\mathbf{G} \in \mathcal{R}^{k\times k}$, $\mathbf{H} \in \mathcal{R}^{n\times k}$, $\mathbf{H}^{'} \in \mathcal{R}^{n\times k}$, *and* $\mathbf{F}$, $\mathbf{G}$ *are symmetric, then the following inequality holds*

$$\operatorname{tr}(\mathbf{H}^T\mathbf{F}\mathbf{H}\mathbf{G}) \leq \sum_{i=1}^{n}\sum_{j=1}^{k}\frac{(\mathbf{F}\mathbf{H}^{'}\mathbf{G})_{ij}\mathbf{H}_{ij}^2}{\mathbf{H}_{ij}^{'}}. \tag{46}$$

Proof. By appling Lemma A.0.1 and the inequality $a \leq \frac{a^2+b^2}{2b}, \forall a, b > 0$, we have

$$\operatorname{tr}(\mathbf{V}\mathbf{A}^+\mathbf{V}^T) \leq \sum_{ij}\frac{(\mathbf{V}^t\mathbf{A}^+)_{ij}\mathbf{V}_{ij}^2}{\mathbf{V}_{ij}^t} = \sum_{ij}(\mathbf{V}^t\mathbf{A}^+)_{ij}\mathbf{V}_{ij}^t\frac{\mathbf{V}_{ij}^2}{(\mathbf{V}_{ij}^t)^2}$$

$$\leq \frac{1}{2}\sum_{ij}(\mathbf{V}^t\mathbf{A}^+)_{ij}\mathbf{V}_{ij}^t(\frac{\mathbf{V}_{ij}^4}{(\mathbf{V}_{ij}^t)^4} + 1) \tag{47}$$

By using the inequality $\mathbf{z} \geq 1 + \log z, \forall z > 0$, we have

$$\operatorname{tr}(\mathbf{V}\mathbf{A}^-\mathbf{V}^T) \geq \sum_{ijk}\mathbf{A}_{jk}^-\mathbf{V}_{ij}^t\mathbf{V}_{ik}^t(1 + \log\frac{\mathbf{V}_{ij}\mathbf{V}_{ik}}{\mathbf{V}_{ij}^t\mathbf{V}_{ik}^t}) \tag{48}$$

$$\operatorname{tr}(\mathbf{V}^T\mathbf{B}^+) \geq \sum_{ij}\mathbf{B}_{ij}^+\mathbf{V}_{ij}^t(1 + \log\frac{\mathbf{V}_{ij}}{\mathbf{V}_{ij}^t}) \tag{49}$$

By using the inequality $a \leq \frac{a^2+b^2}{2b}, \forall a, b > 0$, we have

$$\operatorname{tr}(\mathbf{V}^T\mathbf{B}^-) \leq \sum_{ij}\mathbf{B}_{ij}^-\frac{\mathbf{V}_{ij}^2 + (\mathbf{V}_{ij}^t)^2}{2\mathbf{V}_{ij}^t}$$

$$= \frac{1}{2}\sum_{ij}\mathbf{B}_{ij}^-\mathbf{V}_{ij}^t\frac{\mathbf{V}_{ij}^2}{(\mathbf{V}_{ij}^t)^2} + \sum_{ij}\mathbf{B}_{ij}^-\frac{(\mathbf{V}_{ij}^t)^2}{2\mathbf{V}_{ij}^t}$$

$$\leq \frac{1}{4}\sum_{ij}\mathbf{B}_{ij}^-\mathbf{V}_{ij}^t\left(\frac{\mathbf{V}_{ij}^4}{(\mathbf{V}_{ij}^t)^4} + 1\right) + \sum_{ij}\mathbf{B}_{ij}^-\frac{(\mathbf{V}_{ij}^t)^2}{2\mathbf{V}_{ij}^t} \tag{50}$$

By using the Jensen inequality, we have

$$||\mathbf{V}^T\mathbf{V} - \mathbf{I}||^2 = \sum_{jk}(\sum_i\mathbf{V}_{ij}\mathbf{V}_{ik} - \mathbf{I}_{jk})^2 \tag{51}$$

$$\leq \sum_{jk}\left[\sum_i\frac{\mathbf{V}_{ij}^t\mathbf{V}_{ik}^t}{[(\mathbf{V}^t)^T(\mathbf{V}^t)]_{jk}}\left(\mathbf{I}_{jk} - \frac{[(\mathbf{V}^t)^T(\mathbf{V}^t)]_{jk}}{\mathbf{V}_{ij}^t\mathbf{V}_{ik}^t}\mathbf{V}_{ij}\mathbf{V}_{ik}\right)^2\right]$$

$$= \sum_{jk}\left(\mathbf{I}_{jk} - 2\sum_i\mathbf{I}_{jk}\mathbf{V}_{ij}\mathbf{V}_{ik} + \sum_i\frac{[(\mathbf{V}^t)^T(\mathbf{V}^t)]_{jk}}{\mathbf{V}_{ij}^t\mathbf{V}_{ik}^t}\mathbf{V}_{ij}^2\mathbf{V}_{ik}^2\right)$$

Similarly, we futher have

$$\sum_{ijk}\mathbf{I}_{jk}\mathbf{V}_{ij}\mathbf{V}_{ik} = \operatorname{tr}(\mathbf{V}\mathbf{V}^T)$$

$$\geq \sum_{ijk}\mathbf{I}_{jk}\mathbf{V}_{ij}^t\mathbf{V}_{ik}^t(1 + \log\frac{\mathbf{V}_{ij}\mathbf{V}_{ik}}{\mathbf{V}_{ij}^t\mathbf{V}_{ik}^t}) \tag{52}$$

$$\begin{aligned}&\sum_{ijk}[(\mathbf{V}^t)^T(\mathbf{V}^t)]_{jk}\mathbf{V}^t_{ij}\mathbf{V}^t_{ik}\frac{\mathbf{V}^2_{ij}\mathbf{V}^2_{ik}}{(\mathbf{V}^t_{ij})^2(\mathbf{V}^t_{ik})^2}\\ \le&\frac{1}{2}\sum_{ijk}[(\mathbf{V}^t)^T(\mathbf{V}^t)]_{jk}\mathbf{V}^t_{ij}\mathbf{V}^t_{ik}\left(\frac{\mathbf{V}^4_{ij}}{(\mathbf{V}^t_{ij})^4}+\frac{\mathbf{V}^4_{ik}}{(\mathbf{V}^t_{ik})^4}\right) \qquad (53)\end{aligned}$$

By summing over all the bounds, we can get $g(\mathbf{V},\mathbf{V}^t)$ in Eq. (23), which obviously satisfies the condition in Lemma 3.2.4, that is, (1) $g(\mathbf{V},\mathbf{V}^t)\ge f(\mathbf{V})$; (2)$g(\mathbf{V},\mathbf{V})=f(\mathbf{V})$. To find the minimum of $g(\mathbf{V},\mathbf{V}^t)$, we have

$$\begin{aligned}\frac{\partial g(\mathbf{V},\mathbf{V}^t)}{\partial\mathbf{V}_{ij}}&=2(\mathbf{V}^t\mathbf{A}^+)_{ij}\frac{\mathbf{V}^3_{ij}}{(\mathbf{V}^t_{ij})^3}-2(\mathbf{V}^t\mathbf{A}^-)_{ij}\frac{\mathbf{V}^t_{ij}}{\mathbf{V}_{ij}}\\ &-2\mathbf{B}^+_{ij}\frac{\mathbf{V}^t_{ij}}{\mathbf{V}_{ij}}+2\mathbf{B}^-_{ij}\frac{\mathbf{V}^3_{ij}}{(\mathbf{V}^t_{ij})^3}-4\lambda\mathbf{V}^t_{ij}\frac{\mathbf{V}^t_{ij}}{\mathbf{V}_{ij}}\\ &+4\lambda\left(\mathbf{V}^t(\mathbf{V}^t)^T\mathbf{V}^t\right)_{ij}\frac{\mathbf{V}^3_{ij}}{(\mathbf{V}^t_{ij})^3} \qquad (54)\end{aligned}$$

and the Hessian matrix of $g(\mathbf{V},\mathbf{V}^t)$

$$\begin{aligned}\frac{\partial^2 g(\mathbf{V},\mathbf{V}^t)}{\partial\mathbf{V}_{ij}\mathbf{V}_{ab}}&=\delta_{ia}\delta_{jb}\Bigg(6(\mathbf{V}^t\mathbf{A}^+)_{ij}\frac{\mathbf{V}^2_{ij}}{(\mathbf{V}^t_{ij})^3}+2(\mathbf{V}^t\mathbf{A}^-)_{ij}\frac{\mathbf{V}^t_{ij}}{\mathbf{V}^2_{ij}}\\ &+2\mathbf{B}^+_{ij}\frac{\mathbf{V}^t_{ij}}{\mathbf{V}^2_{ij}}+6\mathbf{B}^-_{ij}\frac{\mathbf{V}^2_{ij}}{(\mathbf{V}^t_{ij})^3}+4\lambda\mathbf{V}^t_{ij}\frac{\mathbf{V}^t_{ij}}{\mathbf{V}^2_{ij}}\\ &+12\lambda\left(\mathbf{V}^t(\mathbf{V}^t)^T\mathbf{V}^t\right)_{ij}\frac{\mathbf{V}^2_{ij}}{(\mathbf{V}^t_{ij})^3}\Bigg) \qquad (55)\end{aligned}$$

is a diagonal matrix with positive diagonal elements. Thus $g(\mathbf{V},\mathbf{V}^t)$ is a convex function of $\mathbf{V}$. Therefore, we can obtain the global minimum of $g(\mathbf{V},\mathbf{V}^t)$ by setting $\frac{\partial g(\mathbf{V},\mathbf{V}^t)}{\partial\mathbf{V}_{ij}}=0$ and solving for $\mathbf{V}$, from which we can get Eq. (**??**). □

APPENDIX B

Appendix two text goes here.

ACKNOWLEDGMENTS

The authors would like to thank...

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.