Adaptive Weighted Multi-View Clustering

Shuo Shuo Liu SHUOSHUO.LIU@PSU.EDU
Department of Statistics, The Pennsylvania State University, USA

Lin Lin LYNN.LIN@DUKE.EDU
Department of Biostatistics and Bioinformatics, Duke University, USA

Abstract

Learning multi-view data is an emerging problem in machine learning research, and nonnegative matrix factorization (NMF) is a popular dimensionality-reduction method for integrating information from multiple views. These views often provide not only consensus but also complementary information. However, most multi-view NMF algorithms assign equal weight to each view or tune the weight via line search empirically, which can be infeasible without any prior knowledge of the views or computationally expensive. In this paper, we propose a weighted multi-view NMF (WM-NMF) algorithm. In particular, we aim to address the critical technical gap, which is to learn both view-specific weight and observation-specific reconstruction weight to quantify each view's information content. The introduced weighting scheme can alleviate unnecessary views' adverse effects and enlarge the positive effects of the important views by assigning smaller and larger weights, respectively. Experimental results confirm the effectiveness and advantages of the proposed algorithm in terms of achieving better clustering performance and dealing with the noisy data compared to the existing algorithms.

Data and Code Availability In this study, we experiment with one image dataset and one clinicalmulti-omics dataset that are publicly available. The data description and processing details are in Appendix C. Codes are available at https://github.com/shuoshuoliu/WM-NMF.

1. Introduction

Learning multi-view data is an emerging problem in machine learning research, as multi-view data become more and more common in many real-world applications. For example, the multi-omics data are now ubiquitous where different biological layers such as genomics, epigenomics, transcriptomics, and proteomics can be obtained from the same set of objects (Hasin et al., 2017; Bhattacharya et al., 2021). In those scenarios, the same set of objects has different views collected from different measuring methods or modalities, where any particular single-view data may be inadequate to comprehensively describe the information of all the objects. Hence, one major goal of multi-view unsupervised learning is to search for a consensus clustering across views so that similar objects are grouped into the same cluster and dissimilar objects are separated into different clusters. In the literature, such a learning problem is called multi-view clustering (Bickel and Scheffer, 2004).

There are mainly two groups of approaches in the existing literature: generative (model-based) and discriminative (similarity-based and dimension reduction-based) (Rappoport and Shamir, 2018). For the generative approach, we typically use the mixture model and regression-based matrix factorization. The idea is to model each data view's probabilistic distribution and obtain a common clustering result by either allowing all views to share the same priors or derived from a shared latent factors (Lashkari and Golland, 2008; Shen et al., 2009; Tzortzis and Likas, 2009, 2010; Savage et al., 2010; Lock and Dunson, 2013; Gabasova et al., 2017). An advantage of the generative approach is that it provides a nice interpretation of what the cluster is built on, but this approach is more computationally expensive in the context of multi-view learning. The discriminative approach focuses on the objective function that optimizes the average similarities within clusters and dissimilarities between clusters. Different objective functions result in different methods, such as multi-view spectral clustering (Wang et al., 2013; Kumar and Daumé, 2011; Kumar et al., 2011), nonnegative matrix factorization for multi-view clustering (Liu et al., 2013; Kalayeh et al., 2014; Yang and Michailidis, 2015; Huang et al., 2014; Zhang et al., 2012),

and canonical correlation analysis (Chaudhuri et al., 2009; Klami et al., 2013; Lai and Fyfe, 2000; Witten and Tibshirani, 2009; Chen et al., 2013). The discriminative approach generally involves non-convex objective functions and it might be hard to find good solutions.

Nonnegative matrix factorization (NMF) is a well-known algorithm for dimension reduction and feature extraction for nonnegative data. Unlike other matrix factorization techniques (Golub and Reinsch, 1970; Abdi and Williams, 2010; Zhao et al., 2015), NMF provides a more intuitive and interpretable understanding through the *parts-based representation*: a data point can be represented by only a few activated basis elements (Turk and Pentland, 1991; Lee and Seung, 1999). NMF has been shown the advantages of extracting sparse and meaningful information from high-dimensional data (Lee and Seung, 1999). The theoretical analysis further reveals the equivalence of NMF and spectral clustering and K-means clustering (Ding et al., 2005). Thus, NMF can also be viewed as a clustering method. The multi-view NMF (MultiNMF) (Liu et al., 2013) is an extension of NMF problem to integrate multiple nonnegative data matrices obtained from a common set of data points. The framework of MultiNMF attempts to approximate each view with some constraints in order to obtain both consensus and view-specific information. Existing related methods tackle this problem with different objective functions motivated by different applications (Zhang et al., 2012; Li et al., 2012; Jin and Lee, 2015; Yang and Michailidis, 2015). However, most existing MultiNMF related methods either assume that all views are equally important or the view-specific weights are known a priori in deriving the consensus clustering. In practice, such an assumption may not be valid as we often have noisy datasets.

The aim of this paper is to design an effective multi-view NMF algorithm that not only can perform multi-view clustering but also quantify each view's weight and each observation's reconstruction weight by learning the corresponding relative values across all views. We expect this weighting mechanism to improve the clustering performance over traditional multi-view clustering algorithms.

Our major contributions include: (1) The proposed method extends and improves the existing MultiNMF method by automatically computing both the view-specific and observation-specific reconstruction weights without requiring the use of prior knowledge. The two types of weights provide two different resolutions in understanding the effects of different views. Thus, the consensus matrix can be obtained by weighting different views, which efficiently extracts different information qualities from each view. (2) We study the properties of these two weighting schemes and provide guidance on choosing the tuning parameters.

The rest of the paper is organized as follows. In Section 2, we introduce notations and overview existing algorithms most relevant to our proposed methods. In Sections 3 and 4, we present our proposed weighted multi-view NMF (WM-NMF) algorithm and study the optimization procedures. In Section 5, experimental results are reported for the handwritten digit data and multi-omics biological data. Comparisons are made with some competing models and popular methods. We conclude with discussions in Section 6.

2. Preliminary

Denote a nonnegative data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}_+^{M\times N}$, where $\mathbf{x}_i = (x_{1i}, ..., x_{Mi})^\top \in \mathbb{R}_+^M$ is the i-th data point of $\mathbf{X}$ containing M features. NMF factorizes $\mathbf{X}$ into a product of two lower-dimensional nonnegative matrices: $\mathbf{X} \approx \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}_+^{M\times K}$, $\mathbf{V} \in \mathbb{R}_+^{N\times K}$, and $K < \min(M, N)$ is a positive integer. The NMF problem $\min_{\mathbf{U},\mathbf{V}\geq 0} \|\mathbf{X} - \mathbf{U}\mathbf{V}^\top\|_F^2$ with Frobenius norm is in general nonconvex and NP-hard, but can be solved with iterative updates that work well in many applications (Lee and Seung, 1999, 2001). Different from other matrix factorization techniques, NMF provides a more intuitive and interpretable understanding through the *parts-based representation*: a data point can be represented by only a few activated basis elements. Further, $\mathbf{V}$ directly translates to data clustering by simply assigning each data point to the basis element on which it has the highest loading; that is, data point i is placed in cluster j if $\mathbf{V}_{i,j}$ is the largest entry in row i. Ding et al. (2005) further shows the equivalence between NMF and K-means and spectral clustering.

The multi-view NMF (MultiNMF) (Liu et al., 2013) is an extension of NMF problem to integrate multiple nonnegative data matrices obtained from a common set of data points and conducts clustering based on the low-rank representations. Let $\{\mathbf{X}^{(1)}, ..., \mathbf{X}^{(n_v)}\}$ be a set of n_v views of data points, with $\mathbf{X}^{(s)} \in \mathbb{R}_+^{M_s\times N}$. Without loss of generality,

we assume all the data matrices are pre-processed and transformed when necessary. The framework of MultiNMF attempts to approximate each view $\mathbf{X}^{(s)} \approx \mathbf{U}^{(s)}\mathbf{V}^{(s)\top}$ with some constraints in order to obtain both consensus and view-specific information. More specifically, the MultiNMF minimizes the following objective function:

$$\sum_{s=1}^{n_v} \left\| \mathbf{X}^{(s)} - \mathbf{U}^{(s)}\mathbf{V}^{(s)\top} \right\|_F^2 + \sum_{s=1}^{n_v} \alpha_s \left\| \mathbf{V}^{(s)}\mathbf{Q}^{(s)} - \mathbf{V}^{*} \right\|_F^2$$

with respect to $\mathbf{U}^{(s)}, \mathbf{V}^{(s)}, \mathbf{V}^{*} \geq 0$. The first part of the objective function performs NMF analysis independently on each view. The second part plays a key role in sharing information across views, and it regularizes the learned coefficient matrices $\mathbf{V}^{(s)}$'s towards a common $\mathbf{V}^{*}$. We take $\mathbf{V}^{*}$ as some latent data structure shared by all views. The amount of information for each view contributing to $\mathbf{V}^{*}$ is regularized by α_s. Thus, α_s is the parameter that tunes the relative weight among views. α_s's have the constraints that $\sum_{s=1}^{n_v} \alpha_s = 1$ and $0 \leq \alpha_s \leq 1$, $s = 1 : n_v$. α_s's are crucial in determining the quality of the consensus matrix $\mathbf{V}^{*}$. The MultiNMF degenerates to the single-view learning when α_s's are binary values with only one component being 1. The resulting consensus clustering is essentially determined by the view that provides the best approximation to the original data.

Most existing MultiNMF related methods tackle different problems with slightly different objective functions motivated by different applications (Zhang et al., 2012; Li et al., 2012; Jin and Lee, 2015; Yang and Michailidis, 2015). However, most of them assume that the weight vector is determined either by prior knowledge (which may be impractical when such knowledge is missing) or assigned to be equal. In practice, such an assumption may not be valid as we often have noisy datasets. In Section 3, we provide an alternative solution to allow a more interpretable and transparent understanding of how to derive the consensus clustering among views.

3. Weighted multi-view NMF

To take the advantage of the consensus matrix used in MultiNMF and learn the weight vector automatically, we adopt the idea of exponential parameter to automatically quantify each view's information content (Tzortzis and Likas, 2009; Xu et al., 2016). In addition, as demonstrated in the handwritten digit dataset and the multi-omics data for liver hepatocellular carcinoma from Section 5, the same data point across views is likely heterogeneous in determining the clustering structure. Thus, it is also important to determine the weight of each observation to describe the relative information content. This is achieved by quantifying the relative reconstruction errors for the same data point across all views. We refer to such weight as observation-specific reconstruction weight. For simplicity, we call it *reconstruction weight* throughout the paper. The strategy of weighting has also been studied in the literature but in a different approach, for example, Li and Ding (2008) weighs each input clustering.

With the abovementioned information, we propose a weighted multi-view NMF (WM-NMF) framework for a more interpretable data integration procedure, while it achieves the ability to automatically update view weight and reconstruction weight. More specifically, WM-NMF works on minimizing the following objective function:

$$\begin{aligned} \mathcal{O} = &\sum_{s=1}^{n_v} \left\| \left\{ \mathbf{X}^{(s)} - \mathbf{U}^{(s)}\mathbf{V}^{(s)\top} \right\} \mathrm{Diag}(\boldsymbol{w}^{(s)}) \right\|_F^2 + \\ &\sum_{s=1}^{n_v} \alpha_s^p \left\| \mathbf{V}^{(s)}\mathbf{Q}^{(s)} - \mathbf{V}^{*} \right\|_F^2 + \beta g(\mathbf{V}^{(1:n_v)}), \end{aligned} \tag{1}$$

where $g(\mathbf{V}^{(1:n_v)})$ is a regularization term on $\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(n_s)}$ which can be set for different purposes, such as sparse NMF (Hoyer, 2004), orthogonal NMF (Zhang et al., 2019; Liang et al., 2020), and graph NMF (Cai et al., 2010; Huang et al., 2014). $\beta > 0$ is the corresponding tuning parameter. We minimize $\mathcal{O}$ over $\mathbf{U}^{(s)}, \mathbf{V}^{(s)}, \boldsymbol{w}^{(s)}, \alpha_s$, and $\mathbf{V}^{*}$ under the constraints that $\mathbf{V}^{*} \geq 0, \mathbf{V}^{(s)} \geq 0, \mathbf{U}^{(s)} \geq 0, \sum_{s=1}^{n_v} \alpha_s = 1, \alpha_s \geq 0, \sum_{s=1}^{n_v} w_i^{(s)} = 1, w_i^{(s)} \geq 0, s = 1 : n_v, i = 1 : N$.

Here, $p \geq 1$ is the exponential parameter and it controls the sparsity of α_s. We provide a discussion about p in Section 4.3. The vector $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_{n_v})^\top$ represents the relative weight among different views and the agreement between $\mathbf{V}^{(s)}$ and $\mathbf{V}^{*}$. It reflects each view's contribution for reaching the consensus matrix $\mathbf{V}^{*}$. $\boldsymbol{w}^{(s)} = (w_1^{(s)}, \ldots, w_N^{(s)})^\top$, where $w_i^{(s)}$ is the reconstruction weight of data point i in view s. Its functionality is different from the view-specific weights, where α_s provides an overall measure to quantify the contribution from view s towards a consensus matrix. Intuitively, a relatively smaller value of $w_i^{(s)}$ implies that the low-dimensional representation $\mathbf{v}_i^{(s)}$ fails to reconstruct observation $\mathbf{x}_i^{(s)}$, compared to other views. The introduction

of $w_i^{(s)}$ provides the flexibility to allow one view to compensate for the shortcoming in another, and potentially prevents the spurious results from noisy or highly divergent views. Therefore, by constraining $\sum_{s=1}^{n_v} w_i^{(s)} = 1$ and $\sum_{s=1}^{n_v} \alpha_s = 1$, we show the feasibility to automatically update the weights as demonstrated in Section 4.

WM-NMF framework extends and improves the existing literature with several benefits. First, it automatically computes the weight vectors without relying on any prior knowledge. Second, it calculates the consensus matrix by weighting different coefficient matrices, which efficiently extracts different qualities of information from each view. Third, it can alleviate the negative effects of unimportant views and enlarge the positive effects of important views by assigning small and large weights on different views and observations, respectively. Lastly, additional regularization can be easily incorporated based on our WM-NMF framework. For example, a manifold regularization can be used to further improve the clustering results (Cai et al., 2010).

The idea of manifold regularization is based on the local invariance assumption such that the geometric structure of the original dataset is inherited in the low-rank representations (Belkin and Niyogi, 2001). To extend the existing manifold regularization for single-view NMF to accommodate our multi-view NMF, we first define an adjacency matrix $\mathbf{A}^{(s)}$ to measure the closeness between any two data points represented by view s. We adopt the Gaussian kernel, $a_{ij}^{(s)} = \exp\left(-\frac{\|\mathbf{x}_i^{(s)} - \mathbf{x}_j^{(s)}\|_2^2}{\sigma^2}\right)$ if $\mathbf{x}_j^{(s)} \in \mathcal{N}_i^{(s)}$ and 0 otherwise, where $\mathcal{N}_i^{(s)}$ denotes the neighbour for point i represented by view s. $\mathcal{N}_i^{(s)}$ is generated using K-nearest neighbour which utilizes the distance between two data points: $\|\mathbf{x}_i^{(s)} - \mathbf{x}_j^{(s)}\|_2^2$. The number of neighbours is set to be 5 and $\sigma^2 = 1$ as suggested in Cai et al. (2010).

Thus, together with the corresponding low-dimensional representation $\mathbf{v}_i^{(s)}$, the manifold regularization is defined as

$$S = \frac{1}{2}\sum_{i,j=1}^{N} \left\|\mathbf{v}_i^{(s)} - \mathbf{v}_j^{(s)}\right\|^2 a_{ij}^{(s)} = \mathrm{Tr}\left(\mathbf{V}^{(s)\top}\mathbf{L}^{(s)}\mathbf{V}^{(s)}\right),$$

where $\mathbf{L}^{(s)} = \mathbf{D}^{(s)} - \mathbf{A}^{(s)}$ is the graph Laplacian matrix and $\mathbf{D}$ is a diagonal matrix with the ith diagonal entry being $\sum_{j=1}^{N} a_{ij}^{(s)}$. $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix. By minimizing S, we expect that if $\mathbf{x}_i^{(s)}$ and $\mathbf{x}_j^{(s)}$ are close, i.e., $a_{ij}^{(s)}$ is large, the corresponding low-dimensional representations $\mathbf{v}_i^{(s)}$ and $\mathbf{v}_j^{(s)}$ are also close together.

Replacing $g(\mathbf{V}^{(1:n_v)})$ in Eq. (1) by the above manifold regularization, we can define the objective function of the manifold regularized WM-NMF as

$$\mathcal{O} = \sum_{s=1}^{n_v} \left\|\left\{\mathbf{X}^{(s)} - \mathbf{U}^{(s)}\mathbf{V}^{(s)\top}\right\}\mathrm{Diag}(\boldsymbol{w}^{(s)})\right\|_F^2 + \sum_{s=1}^{n_v} \alpha_s^p \left\|\mathbf{V}^{(s)}\mathbf{Q}^{(s)} - \mathbf{V}^*\right\|_F^2 + \beta \sum_{s=1}^{n_v} \mathrm{Tr}(\mathbf{V}^{(s)\top}\mathbf{L}^{(s)}\mathbf{V}^{(s)}). \quad (2)$$

We will discuss how to choose β in the experiment section. The scenario for WM-NMF without manifold regularization can be retrieved by setting $\beta = 0$. An illustration of the manifold regularized WM-NMF is shown in Figure 1. In Section 4, the optimization procedures are based on the objective function $\mathcal{O}$ in Eq. (2).

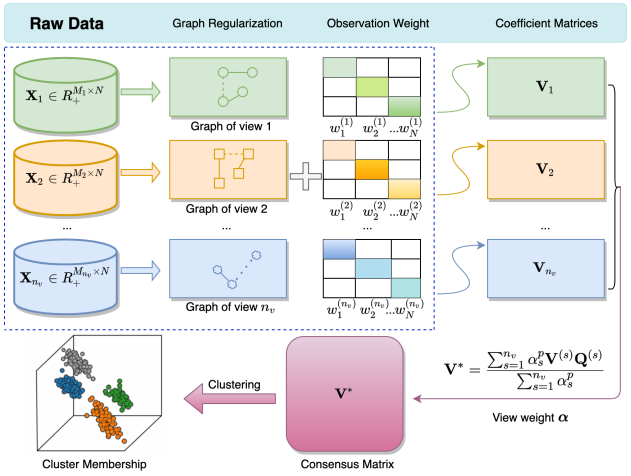

Figure 1: The illustration of the weighted multi-view NMF with manifold regularization for clustering.

4. Optimization

The joint optimization function in Eq. (2) is nonconvex over all variables. However, if we keep four of the five variables ($\mathbf{U}^{(s)}$, $\mathbf{V}^{(s)}$, $\mathbf{V}^*$, $\boldsymbol{\alpha}$, $\boldsymbol{w}^{(s)}$) fixed, and optimize over one of them, the problem is convex and can be solved efficiently. We thus consider the following iterative alternating minimization method until

convergence. At each iteration, we optimize over the five variables alternatively.

More specifically, we first update $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$ individually while keeping the others fixed. We call these procedures the *inner iteration*. Updating $\mathbf{U}^{(s)}$ or $\mathbf{V}^{(s)}$ only needs to solve the single-view objective function represented by

$$\begin{aligned}\mathcal{O}_0 = &\left\| \left\{ \mathbf{X}^{(s)} - \mathbf{U}^{(s)} \mathbf{V}^{(s)^\top} \right\} \mathrm{Diag}(\boldsymbol{w}^{(s)}) \right\|_F^2 + \\ &\alpha_s^p \left\| \mathbf{V}^{(s)} \mathbf{Q}^{(s)} - \mathbf{V}^* \right\|_F^2 + \beta \mathrm{Tr}(\mathbf{V}^{(s)^\top} \mathbf{L}^{(s)} \mathbf{V}^{(s)}).\end{aligned} \tag{3}$$

After obtaining $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$, we solve the exact solutions of $\boldsymbol{\alpha}$, $\boldsymbol{w}^{(s)}$ and $\mathbf{V}^*$. We call these procedures the *outer iteration*.

4.1. Update $\mathbf{U}^{(s)}$

To simplify the notation, we omit the view index (s) for the derivations of the inner iteration. Taking the derivative of $\mathcal{O}_0$ with respect to u_{ij} while taking into account of the nonnegative constraint on u_{ik}, and using the complementary slackness condition $\Psi_{ik} u_{ik} = 0$, where Ψ_{ik} is the Lagrange multiplier for the constraint $u_{ik} \geq 0$ leads to the multiplicative update rule of u_{ik}:

$$\begin{aligned}u_{ik} &\leftarrow u_{ik} \frac{\left[\mathbf{X}\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{j=1}^N v_{jk} v_{jk}^*}{\left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{l=1}^M u_{lk} \sum_{j=1}^N v_{jk}^2} \\ &= u_{ik} - \underbrace{\frac{u_{ik}}{\left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{l=1}^M u_{lk} \sum_{j=1}^N v_{jk}^2}}_{\text{step size}} \times \frac{\nabla_U \mathcal{O}_0}{2}\end{aligned} \tag{4}$$

where $\nabla_U \mathcal{O}_0 = \frac{\partial \mathcal{O}_0}{\partial u_{ik}}$. Details for the derivation is in Appendix A. The update can be viewed as an adaptive gradient descent algorithm, where the step size should be nonzero. Therefore, we should initialize a positive u_{ik}, otherwise u_{ik} equals to 0 for all subsequent iterations.

4.2. Update $\mathbf{V}^{(s)}$

Similarly, we omit the view index (s) for notation simplicity. Let Φ_{jk} be the Lagrange multiplier for the constraint $v_{jk} \geq 0$. Setting the derivative of $\mathcal{O}_0$ with respect to v_{jk} to be 0 while taking into consideration of the nonnegative constraint of v_{jk}, and using the complementary slackness condition $\Phi_{jk} v_{jk} = 0$, we have the multiplicative update rule of v_{jk}:

$$\begin{aligned}v_{jk} &\leftarrow v_{jk} \frac{\left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{X}^\top \mathbf{U}\right]_{jk} + \alpha_s^p \left[\mathbf{V}^* \mathbf{Q}^\top\right]_{jk} + \beta [\mathbf{A}\mathbf{V}]_{jk}}{\left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\mathbf{U}^\top \mathbf{U}\right]_{jk} + \alpha_s^p [\mathbf{V}\mathbf{Q}\mathbf{Q}^\top]_{jk} + \beta [\mathbf{D}\mathbf{V}]_{jk}} \\ &= v_{jk} - \underbrace{\frac{v_{jk}}{\left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\mathbf{U}^\top \mathbf{U}\right]_{jk} + \alpha_s^p [\mathbf{V}\mathbf{Q}\mathbf{Q}^\top]_{jk} + \beta [\mathbf{D}\mathbf{V}]_{jk}}}_{\text{step size}} \times \nabla_V \mathcal{O}_0,\end{aligned} \tag{5}$$

where $\nabla_V \mathcal{O}_0 = \frac{\partial \mathcal{O}_0}{\partial v_{jk}}$, and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the graph Laplacian matrix defined in Section 3. The update can be viewed as an adaptive gradient descent algorithm, where the step size should be nonzero. Again, we should make sure the initialization of v_{jk} is positive, otherwise $v_{jk} = 0$ at all subsequent iterations.

Proposition 1 below ensures that when we initialize positive u_{ik} and v_{jk}, the entries of $\mathbf{U}$ and $\mathbf{V}$ will always be updated as positive numbers, and the updated values will not get trapped in 0. We provide the proof in Appendix B.

Proposition 1 *If $u_{ik}^1 > 0$ and $v_{jk}^1 > 0$, $\forall i, j, k$, then $u_{ik}^t > 0, v_{jk}^t > 0$, $\forall i, j, k, \forall t \geq 1$, where t denotes the t-th update.*

4.3. Estimate α_s

This is equivalent to minimizing the following objective over α_s that

$$\min_{\alpha_s} \alpha_s^p \left\| \mathbf{V}^{(s)} \mathbf{Q}^{(s)} - \mathbf{V}^* \right\|_F^2, \text{ subject to } \sum_{s=1}^{n_v} \alpha_s = 1.$$

When $p = 1$, the optimal solution of α_s is

$$\hat{\alpha}_s = \begin{cases} 1, & s = \underset{s' \in 1,\ldots,n_v}{\arg\min} \|\mathbf{V}^{(s')} \mathbf{Q}^{(s')} - \mathbf{V}^*\|_F^2 \\ 0, & \text{otherwise.} \end{cases}$$

The above solution implies that $p = 1$ only offers a binary solution of α_s, i.e., the consensus matrix $\mathbf{V}^*$ depends on a single view. Such a solution is obviously too restrictive, as it prevents the partial information sharing among views. On the other hand, when $p > 1$, we obtain the optimal solution for $\boldsymbol{\alpha}$ as:

$$\hat{\alpha}_s = \frac{1}{\sum_{s'=1}^{n_v} \left(\frac{\|\mathbf{V}^{(s)} \mathbf{Q}^{(s)} - \mathbf{V}^*\|_F^2}{\|\mathbf{V}^{(s')} \mathbf{Q}^{(s')} - \mathbf{V}^*\|_F^2} \right)^{\frac{1}{p-1}}}. \tag{6}$$

The solution implies that when the s-th view's information content contributes more to the consensus matrix, i.e., $\|\mathbf{V}^{(s)} \mathbf{Q}^{(s)} - \mathbf{V}^*\|_F^2$ is smaller, $\hat{\alpha}_s$ becomes

larger. Therefore, the more important the view is, the larger the corresponding weight is.

Discussion about p: Denote $A^{(s)} = \|\mathbf{V}^{(s)}\mathbf{Q}^{(s)} - \mathbf{V}^*\|_F^2$ and $A^{(s')} = \|\mathbf{V}^{(s')}\mathbf{Q}^{(s')} - \mathbf{V}^*\|_F^2$. It is clear that as p goes to infinity, the denominator of Eq. (6) converges to n_v, which gives uniform weights to each view. Meanwhile, if the normalized s-th view $\mathbf{V}^{(s)}\mathbf{Q}^{(s)}$ contributes the most to the consensus matrix $\mathbf{V}^*$, i.e., $A^{(s)}/A^{(s')} < 1$ for $s' \neq s$, then $p \to 1^+$ implies $\alpha_s \to 1$. On the other hand, if the normalized s-th view $\mathbf{V}^{(s)}\mathbf{Q}^{(s)}$ contributes the least to the consensus matrix $\mathbf{V}^*$, i.e., $A^{(s)}/A^{(s')} > 1$ for $s' \neq s$, then $p \to 1^+$ implies $\alpha_s \to 0$. Hence, a smaller p results in a sparser weight vector $\boldsymbol{\alpha}$. Generally, a moderate size p should be used so that the relevant information from different views is preserved and the effect of consensus constraint is kept.

4.4. Estimate $w_i^{(s)}$

To optimize $w_i^{(s)}$, we only consider the terms involving $w_i^{(s)}$ in the objective that we consider

$$\min_{\boldsymbol{w}^{(s)} \geq 0, \sum_{s=1}^{n_v} w_i^{(s)} = 1} \left\| \left\{ \mathbf{X}^{(s)} - \mathbf{U}^{(s)} \mathbf{V}^{(s)\top} \right\} \mathrm{Diag}(\boldsymbol{w}^{(s)}) \right\|_F^2 = \sum_{i=1}^{N} {w_i^{(s)}}^2 \sum_{j=1}^{M_s} {\mathbf{Y}_{ji}^{(s)}}^2,$$

where $\mathbf{Y}^{(s)} = \mathbf{X}^{(s)} - \mathbf{U}^{(s)}\mathbf{V}^{(s)\top}$. Since we only optimize the weight for a single observation, it is equivalent to minimizing ${w_i^{(s)}}^2 \sum_{j=1}^{M_s} {\mathbf{Y}_{ji}^{(s)}}^2$ with the constraint $\sum_{s=1}^{n_v} w_i^{(s)} = 1$. We have that the optimal solution is

$$\hat{w}_i^{(s)} = \left(\sum_{s'=1}^{n_v} \frac{1}{\sum_{j=1}^{M_{s'}} (\mathbf{Y}_{ji}^{(s')})^2} \sum_{j=1}^{M_s} (\mathbf{Y}_{ji}^{(s)})^2 \right)^{-1}. \quad (7)$$

It is easy to find the solution is nonnegative. The above solution shows that the reconstruction weight is determined by the reconstruction error of the s-th view on the i-th observation across all the features. The smaller the error is compared with other views, the larger the weight is. Note that $w_i^{(s)}$ is the weight of an observation, so the algorithm may run slowly with a very large sample size.

4.5. Estimate $\mathbf{V}^*$

To optimize $\mathbf{V}^*$, we only consider the terms involving $\mathbf{V}^*$ in the objective $\mathcal{O} = \sum_{s=1}^{n_v} \alpha_s^p \left\| \mathbf{V}^{(s)}\mathbf{Q}^{(s)} - \mathbf{V}^* \right\|_F^2$. Setting the derivative of $\mathcal{O}_v$ with respect to $\mathbf{V}^*$ to 0, we have that the optimal solution is

$$\mathbf{V}^* = \frac{\sum_{s=1}^{n_v} \alpha_s^p \mathbf{V}^{(s)}\mathbf{Q}^{(s)}}{\sum_{s=1}^{n_v} \alpha_s^p}. \quad (8)$$

Since $\mathbf{V}^{(s)} \geq 0$, $\mathbf{Q}^{(s)} \geq 0$, and $\alpha_s > 0$, $\mathbf{V}^*$ is nonnegative. The underlying assumption of the multi-view clustering is that all the views can agree and reduce to a consensus matrix with different weights, so the cluster assignments can be determined according to the consensus matrix $\mathbf{V}^*$ by the maximum coefficient assignments. However, Welch et al. (2019) points out the spurious alignments in highly divergent datasets by the maximum coefficient assignments. In the experiment section, we use the default function `spectralcluster` in MATLAB on $\mathbf{V}^*$ to obtain the cluster membership.

4.6. Summary of the algorithm

We summarize the pseudocode of the WM-NMF algorithm below. The algorithm stops when the maximum number of iterations is reached or it converges, i.e., when the difference between the two consecutive iterations is less than the threshold 9×10^{-8}. The algorithm converges to a local minima since the objective function is nonconvex.

Since the objective function is nonconvex, the solution may depend on the initialization. We initialize $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$ by the Graph Regularized Nonnegative Matrix Factorization (GNMF) (Cai et al., 2010).

Theorem 2 *The objective function $\mathcal{O}$ converges to a local minima under Algorithm 1.*

The proof is given in Appendix B. We verify Theorem 2 through different datasets and provide complexity analysis in Appendix C. Besides, we include the complexity analysis (operation counts) in Appendix D.

Discussion of the tuning parameters: A larger value of K better approximates the original data, but on the other hand, it raises the risk of overfitting. Many approaches have been developed to select the number of basis elements K for the NMF problem, such as Bi-cross validation (Owen et al., 2009), Stein's unbiased risk estimator (Ulfarsson and Solo, 2013), minimum description length (MDL) (Squires et al., 2017), and missing data imputation (Lin and Boutros, 2020).

Algorithm 1: Weighted Multi-View NMF (WM-NMF)

Input: Dataset $\{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n_v)}\}$; rank K; exponential parameter p; manifold parameter β.
Output: Basis matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(n_v)}$; Coefficient matrices $\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(n_v)}$; Consensus matrix $\mathbf{V}^*$; View weight vector α_s for each view; Reconstruction weight $w_i^{(s)}$ for each view.
Normalize each view $\mathbf{X}^{(s)}$ such that $\|\mathbf{X}^{(s)}\|_1 = 1$.
Initialize $\mathbf{U}^{(s)}$, $\mathbf{V}^{(s)}$, $\mathbf{V}^*$, set equal view weight $\alpha_s = 1/n_v$ and reconstruction weight $w_i^{(s)} = 1/N$.
repeat ← (outer iteration)
  for $s = 1 : n_v$ **do** ← (inner iteration)
    repeat
      Fixing $\boldsymbol{w}^{(s)}$, $\boldsymbol{\alpha}$, $\mathbf{V}^*$ and $\mathbf{V}^{(s)}$, update $\mathbf{U}^{(s)}$ by Eq. (4);
      Fixing $\boldsymbol{w}^{(s)}$, $\boldsymbol{\alpha}$, $\mathbf{V}^*$ and $\mathbf{U}^{(s)}$, update $\mathbf{V}^{(s)}$ by Eq. (5);
    until $\mathcal{O}_0$ converges or the maximum number of iteration is reached.
  end for
  Fixing $\boldsymbol{w}^{(s)}$, $\mathbf{U}^{(s)}$, $\mathbf{V}^{(s)}$, $s = 1, ..., n_v$, and $\mathbf{V}^*$, update $\boldsymbol{\alpha}$ by Eq. (6);
  Fixing $\boldsymbol{\alpha}$, $\mathbf{U}^{(s)}$, $\mathbf{V}^{(s)}$, and $\mathbf{V}^*$, update $\boldsymbol{w}$ by Eq. (7);
  Fixing $\boldsymbol{w}^{(s)}$, $\mathbf{U}^{(s)}$, $\mathbf{V}^{(s)}$, $s = 1, ..., n_v$, and $\boldsymbol{\alpha}$, update $\mathbf{V}^*$ by Eq. (8);
until the maximum number of iteration is reached or the algorithm converges.
Perform clustering analysis based on $\mathbf{V}^*$.

Overall, all these methods show the capacity of selecting K in certain datasets empirically. In this paper, we assume prior information of K is given. In Appendix C, we show that WM-NMF works considerately well within a range of K in terms of the clustering accuracy. Thus, it is quite robust to the choice of K. Empirical results on other tuning parameters are also included in Appendix C.

5. Experiments

In this section, we present experimental results on one handwritten digit dataset and one multi-omics dataset. For each dataset, we use six metrics to evaluate the clustering performance: accuracy (ACC), normalized mutual information (NMI), Precision, Recall, F-score, Adjusted Rand index (Adj-RI). For all these metrics, higher values indicate better clustering performance. Details and formulas of them are available in Manning et al. (2008). Empirical studies on the tuning parameters are given in Appendix C.

In addition, we compare WM-NMF with several competing multi-view clustering algorithms described below:

1. K-means: The default `kmeans` function in MATLAB is implemented to obtain the results. There are two strategies: (1) Apply K-means independently on each single view, and select the best performance of K-means as the final results. We denote this strategy as BSV-kmeans. (2) Apply K-means on the data where all the views are concatenated. We denote this strategy as ConcatK.
2. Spectral clustering: The classical spectral clustering algorithm is applied to the datasets. The default function `spectralcluster` in MATLAB is implemented to obtain the results. Similar to the above K-means, we denote BSV-Spectral as the best performance of spectral clustering over each single view, and ConcatSpectral represents the result of spectral clustering on the data with all views concatenated.
3. MultiNMF: Multi-view nonnegative matrix factorization with equal weight. MultiNMF1 is implemented with equal weights summing to 1, i.e., $\alpha_s = 1/n_v$ for $s = 1, ..., n_v$. MultiNMF2 is implemented with equal weight such that $\alpha_s = 0.01$, which is shown to have the best performance in Liu et al. (2013). The clustering is performed based on $\mathbf{V}^*$ by K-means.
4. MLRSSC: Multi-View low-rank sparse subspace clustering, which is shown to be very competitive in (Brbić and Kopriva, 2018). More specifically, we implement four variants: P-MLRSSC, C-MLRSSC, P-KMLRSSC and C-KMLRSSC which represent pairwise, centroid-based, pairwise kernel, and centroid-based kernel multi-view low-rank sparse subspace clustering, respectively.
5. NMF-W1 and NMF-W2: They are the simplified version of WM-NMF. We denote NMF-W1 as the case when $\boldsymbol{w}$ is fixed a priori in WM-NMF. Comparing NMF-W1 to WM-NMF, we emphasize the importance of the reconstruction weight. Likewise, we denote NMF-W2 as the situation when $\boldsymbol{w}$ is fixed and β is set to 0, such that there is no manifold regularization in WM-NMF. Comparing NMF-W2 to WM-NMF, we empha-

size the importance of the reconstruction weight and the manifold regularization.

For all the experiments, we set $p = 5$ and $\beta = 0.01$ for WM-NMF and we use the default settings for all the other algorithms.

5.1. Data description

We summarize the key information of all datasets in Table 1 and provide the data descriptions in Appendix C.

Table 1: Detailed information about the experiment datasets.

Dataset	view	observations	features	clusters
Synthetic	$\mathbf{X}^{(1)}$	5000	100	10
	$\mathbf{X}^{(2)}$	5000	150	10
	$\mathbf{X}^{(3)}$	5000	50	10
	$\mathbf{X}^{(4)}$	5000	200	10
	$\mathbf{X}^{(5)}$	5000	100	-
	$\mathbf{X}^{(6)}$	5000	50	-
Handwritten	fou	2000	76	10
	pix	2000	240	10
	zer	2000	47	10
	fac	2000	216	10
LIHC	GE	404	15397	2
	CNA	404	16384	2
	DNAm	404	16384	2

5.2. Results on the handwritten digit dataset

Table 2 compares the result of WM-NMF to the other algorithms based on the handwritten digit dataset. It is worth noting that WM-NMF obtains the highest scores for all six evaluation metrics. All the other competing methods show clustering performance significantly worse than WM-NMF. Now, we analyze the effectiveness of the manifold regularization, view-specific weight and reconstruction weight, respectively. First, we observe that NMF-W1 performs better than NMF-W2, which implies the importance of the manifold regularization for clustering analysis. Second, WM-NMF, NMF-W1, and NMF-W2 all outperform MultiNMF, this shows the ineffectiveness of using equal weight for $\boldsymbol{\alpha}$. Third, we find that WM-NMF hits higher scores than NMF-W1. This demonstrates the advantage of using reconstruction weight and the ability of WM-NMF on integrating heterogeneous data.

5.3. Results on the multi-omics LIHC dataset

Seal et al. (2020) uses both CNV and DNAm to predict the sample status, either tumor or normal sample, so it is treated as a classification problem and the accuracy is 95.1%. In this paper, we treat this data as unsupervised problem and integrate all the three omics data to conduct clustering analysis.

Figure 2 presents the analysis results. As it shows, both WM-NMF and NMF-W2 outperform the other algorithms in all the evaluation metrics while WM-NMF behaves much better than NMF-W2. The average scores of the proposed WM-NMF for the 6 evaluation metrics are 0.97, 0.70, 0.99, 0.95, 0.97, and 0.83, respectively. Note that we do not plot the results of MLRSSC algorithms due to their code constraints. Instead, we report the highest six metric scores with standard deviations among the four algorithms: 0.57 (0.04), 0.18 (0.00), 0.88 (0.00), 0.54 (0.00), 0.67 (0.00) and 0.10 (0.00). Besides, the proposed WM-NMF algorithm has lower standard deviations compared to all the other algorithms. It is worth noting that the clustering algorithm WM-NMF achieves slightly higher accuracy than the neural network approach for classification in Seal et al. (2020).

6. Discussion

We develop a weighted multi-view NMF (WM-NMF) algorithm, with the goal of learning multi-view data for integrative clustering analysis. One key feature of WM-NMF is the ability to learn both view-specific and reconstruction weights to quantify each view's information content. Thus, the unnecessary views' adverse effects can be alleviated and the positive effects of the important views are enlarged, making WM-NMF robust to the potentially heterogeneous multi-view data. Such ability enables WM-NMF to deal with heterogeneous and noisy data. Technically, our proposed weighting scheme can be integrated into other methods such as model-based approaches. Therefore, we may combine the benefits of the model-based approaches with the weighting scheme to study the theoretical properties.

Institutional Review Board (IRB) This study has no human-subject research and only uses publicly available and de-identified data, which does not need an IRB approval.

Table 2: Comparisons of clustering performance between WM-NMF and other competing methods for handwritten digit dataset. Numbers in the bracket represent standard deviations of the corresponding scores, which is obtained based on 20 replications for each algorithm.

Algorithm	ACC	NMI	Precision	Recall	F-score	Adj-RI
BSV-kmeans	0.69 (0.07)	0.70 (0.03)	0.61 (0.05)	0.67 (0.04)	0.63 (0.05)	0.59 (0.05)
ConcatK	0.63 (0.07)	0.62 (0.03)	0.51 (0.05)	0.59 (0.03)	0.55 (0.04)	0.49 (0.04)
BSV-Spectral	0.68 (0.00)	0.71 (0.00)	0.58 (0.00)	0.68 (0.00)	0.62 (0.00)	0.58 (0.00)
ConcatSpectral	0.12 (0.00)	0.01 (0.00)	0.10 (0.00)	0.41 (0.04)	0.16 (0.00)	0.00 (0.00)
MultiNMF1	0.64 (0.03)	0.58 (0.02)	0.51 (0.03)	0.54 (0.03)	0.52 (0.03)	0.47 (0.03)
MultiNMF2	0.79 (0.04)	0.72 (0.02)	0.66 (0.03)	0.69 (0.03)	0.68 (0.03)	0.64 (0.03)
P-MLRSSC	0.75 (0.07)	0.77 (0.04)	0.68 (0.07)	0.75 (0.05)	0.71 (0.06)	0.68 (0.07)
C-MLRSSC	0.75 (0.06)	0.77 (0.04)	0.68 (0.06)	0.74 (0.05)	0.71 (0.06)	0.67 (0.06)
P-KMLRSSC	0.77 (0.06)	0.72 (0.02)	0.66 (0.05)	0.68 (0.05)	0.67 (0.04)	0.63 (0.05)
C-KMLRSSC	0.76 (0.07)	0.72 (0.03)	0.65 (0.06)	0.68 (0.05)	0.67 (0.05)	0.63 (0.06)
NMF-W1	0.92 (0.03)	0.88 (0.03)	0.85 (0.06)	0.88 (0.03)	0.86 (0.05)	0.84 (0.05)
NMF-W2	0.81 (0.08)	0.77 (0.05)	0.69 (0.10)	0.76 (0.06)	0.72 (0.08)	0.69 (0.09)
WM-NMF	0.96 (0.02)	0.93 (0.01)	0.93 (0.04)	0.94 (0.01)	0.93 (0.03)	0.93 (0.03)

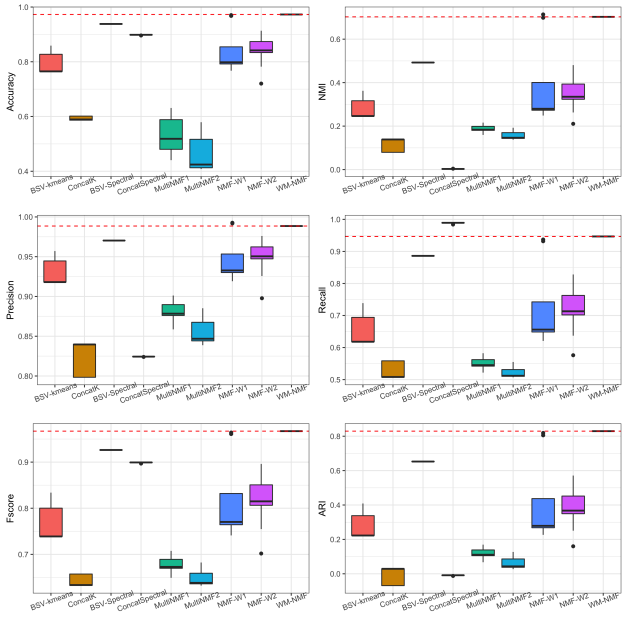

Figure 2: Boxplots representing clustering results for LIHC dataset based on 20 replications for each algorithm. The dashed red line is the average score of WM-NMF.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, 2001.

Arjun Bhattacharya, Yun Li, and Michael I. Love. Mostwas: Multi-omic strategies for transcriptome-wide association studies. *PLOS Genetics*, 17(3): 1–30, 03 2021.

Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *ICDM*, volume 4, pages 19–26, 2004.

Maria Brbić and Ivica Kopriva. Multi-view low-rank sparse subspace clustering. *Pattern Recognition*, 73:247–258, 2018.

Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

Kamalika Chaudhuri, Sham M. Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clus-

tering via canonical correlation analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning*, page 129–136, New York, NY, USA, 2009. Association for Computing Machinery.

Jun Chen, Frederic D Bushman, James D Lewis, Gary D Wu, and Hongzhe Li. Structure-constrained sparse canonical correlation analysis with an application to microbiome data analysis. *Biostatistics*, 14(2):244–258, 2013.

Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 606–610. SIAM, 2005.

Evelina Gabasova, John Reid, and Lorenz Wernisch. Clusternomics: Integrative context-dependent clustering for heterogeneous datasets. *PLoS Computational Biology*, 13(10):e1005781, 2017.

G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, Apr 1970.

Yehudit Hasin, Marcus Seldin, and Aldons Lusis. Multi-omics approaches to disease. *Genome Biology*, 18(1):83, May 2017.

Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(9), 2004.

Jin Huang, Feiping Nie, Heng Huang, and Chris Ding. Robust manifold nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3):1–21, 2014.

Daeyong Jin and Hyunju Lee. A computational approach to identifying gene-microrna modules in cancer. *PLOS Computational Biology*, 11(1):1–33, 01 2015.

M. M. Kalayeh, H. Idrees, and M. Shah. Nmf-knn: Image annotation using weighted multi-view non-negative matrix factorization. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 184–191, 2014.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14:965–1003, 2013.

Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 393–400, 2011.

Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems*, pages 1413–1421, 2011.

Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377, 2000.

Danial Lashkari and Polina Golland. Convex clustering with exemplar-based models. In *Advances in neural information processing systems*, pages 825–832, 2008.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

Tao Li and Chris Ding. Weighted consensus clustering. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 798–809. SIAM, 2008.

Wenyuan Li, Shihua Zhang, Chun-Chi Liu, and Xianghong Jasmine Zhou. Identifying multi-layer gene regulatory modules from multi-dimensional genomic data. *Bioinformatics*, 28(19):2458–2466, 08 2012.

Youwei Liang, Dong Huang, Chang-Dong Wang, and Philip S Yu. Multi-view graph learning by joint modeling of consistency and inconsistency. *arXiv preprint arXiv:2008.10208*, 2020.

Chih-Jen Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 18(6):1589–1596, 2007.

Xihui Lin and Paul C Boutros. Optimization and expansion of non-negative matrix factorization. *BMC bioinformatics*, 21(1):1–10, 2020.

Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix

factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 252–260. SIAM, 2013.

Eric F Lock and David B Dunson. Bayesian consensus clustering. *Bioinformatics*, 29(20):2610–2616, 2013.

Christopher D Manning, Hinrich Schütze, and Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge university press, 2008.

Art B Owen, Patrick O Perry, et al. Bi-cross-validation of the svd and the nonnegative matrix factorization. *The annals of applied statistics*, 3(2): 564–594, 2009.

Nimrod Rappoport and Ron Shamir. Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic Acids Research*, 46(20): 10546–10562, 2018.

Richard S Savage, Zoubin Ghahramani, Jim E Griffin, Bernard J De la Cruz, and David L Wild. Discovering transcriptional modules by bayesian data integration. *Bioinformatics*, 26(12):i158–i167, 2010.

Dibyendu Bikash Seal, Vivek Das, Saptarsi Goswami, and Rajat K De. Estimating gene expression from dna methylation and copy number variation: A deep learning regression model for multi-omics integration. *Genomics*, 2020.

Ronglai Shen, Adam B Olshen, and Marc Ladanyi. Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, 25(22):2906–2912, 2009.

Steven Squires, Adam Prügel-Bennett, and Mahesan Niranjan. Rank selection in nonnegative matrix factorization using minimum description length. *Neural computation*, 29(8):2164–2176, 2017.

Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3 (1):71–86, January 1991. doi: 10.1162/jocn.1991.3.1.71.

Grigorios Tzortzis and Aristidis Likas. Convex mixture models for multi-view clustering. In *International Conference on Artificial Neural Networks*, pages 205–214. Springer, 2009.

Grigorios F Tzortzis and Aristidis C Likas. Multiple view clustering using a weighted combination of exemplar-based mixture models. *IEEE Transactions on neural networks*, 21(12):1925–1938, 2010.

Magnus O Ulfarsson and Victor Solo. Tuning parameter selection for nonnegative matrix factorization. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6590–6594. IEEE, 2013.

Xiang Wang, Buyue Qian, Jieping Ye, and Ian Davidson. Multi-objective multi-view spectral clustering via pareto optimization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 234–242. SIAM, 2013.

Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

Daniela M Witten and Robert J Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genetics and Molecular Biology*, 8(1), 2009.

Yu-Meng Xu, Chang-Dong Wang, and Jian-Huang Lai. Weighted multi-view clustering with feature selection. *Pattern Recognition*, 53:25–35, 2016.

Zi Yang and George Michailidis. A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics*, 32(1):1–8, 09 2015.

Kai Zhang, Sheng Zhang, Jun Liu, Jun Wang, and Jie Zhang. Greedy orthogonal pivoting algorithm for non-negative matrix factorization. In *International Conference on Machine Learning*, pages 7493–7501. PMLR, 2019.

Shihua Zhang, Chun-Chi Liu, Wenyuan Li, Hui Shen, Peter W. Laird, and Xianghong Jasmine Zhou. Discovery of multi-dimensional modules by integrative analysis of cancer genomic data. *Nucleic Acids Research*, 40(19):9379–9391, 08 2012.

Tuo Zhao, Zhaoran Wang, and Han Liu. A nonconvex optimization framework for low rank matrix estimation. *Advances in neural information processing systems*, 28:559–567, 2015.

Appendix A

In this section, we provide the derivations for updates of $\mathbf{U}^{(s)}$, $\mathbf{V}^{(s)}$, and α_s.

Update of $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$

With the notations in the main text, we have

$$\frac{\partial \mathcal{O}_1}{\partial u_{ik}} = 2\left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} - 2\left[\mathbf{X}\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p P_{ik} + \Psi_{ik},$$

where $P_{ik} = 2\left(\sum_{l=1}^M u_{lk}\sum_{j=1}^N v_{jk}^2 - \sum_{j=1}^N v_{jk}v_{jk}^*\right)$ and Ψ_{ik} is the Lagrange multiplier for the constraint $u_{ik} \geq 0$. Using the complementary slackness condition $\Psi_{ik}U_{ik} = 0$, plugging the expression P_{ik} into Eq. (6) and setting it to 0, we have

$$\left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})V\right]_{ik} u_{ik} - \left[\mathbf{X}\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} u_{ik} + \alpha_s^p \left(\sum_{l=1}^M u_{lk}\sum_{j=1}^N v_{jk}^2 - \sum_{j=1}^N v_{jk}v_{jk}^*\right) u_{ik} = 0$$

$$\Leftrightarrow \left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})V\right]_{ik} u_{ik} + \alpha_s^p \sum_{l=1}^M u_{lk}\sum_{j=1}^N v_{jk}^2 u_{ik} = \left[\mathbf{X}\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} u_{ik} + \alpha_s^p \sum_{j=1}^N v_{jk}v_{jk}^* u_{ik}.$$

Similarly for $\mathbf{V}^{(s)}$, we have

$$\frac{\partial \mathcal{O}_2}{\partial v_{jk}} = \left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\mathbf{U}^\top\mathbf{U}\right]_{jk} - \left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{X}^\top\mathbf{U}\right]_{jk} + \alpha_s^p\left[\mathbf{V}\mathbf{Q}\mathbf{Q}^\top - \mathbf{V}^*\mathbf{Q}^\top\right]_{jk} + \beta\left[\mathbf{L}\mathbf{V}\right]_{jk} + \Phi_{jk} = 0$$

$$\Downarrow\Downarrow$$

$$\left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\mathbf{U}^\top\mathbf{U}\right]_{jk} + \alpha_s^p\left[\mathbf{V}\mathbf{Q}\mathbf{Q}^\top\right]_{jk} + \beta\left[D\mathbf{V}\right]_{jk} + \Phi_{jk} = \left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{X}^\top\mathbf{U}\right]_{jk} + \alpha_s^p\left[\mathbf{V}^*\mathbf{Q}^\top\right]_{jk} + \beta\left[\mathbf{A}\mathbf{V}\right]_{jk},$$

Multiplying the above equation by v_{jk} and using the complementary slackness condition $\Phi_{jk}V_{jk} = 0$ gives the result.

Update of α_s

When $p > 1$, setting the derivative of $\mathcal{O}$ that only contains α_s with respect to α_s to 0, we get

$$p\alpha_s^{(p-1)}A + \lambda_1 = 0 \quad \Rightarrow \quad \alpha_s = \left(-\frac{\lambda_1}{pA}\right)^{\frac{1}{p-1}},$$

where we assume $A^{(s)} = \|\mathbf{V}^{(s)}\mathbf{Q}^{(s)} - \mathbf{V}^*\|_F^2 > 0$. Given the constraint that $\sum_{s'=1}^{n_v}\alpha_{s'} = 1$, we have

$$\sum_{s'=1}^{n_v}\left(-\frac{\lambda_1}{pA^{(s')}}\right)^{\frac{1}{p-1}} = 1 \quad \Rightarrow \quad (-\lambda_1)^{\frac{1}{p-1}} = \frac{1}{\sum_{s'=1}^{n_v}\left(\frac{1}{pA^{(s')}}\right)^{\frac{1}{p-1}}},$$

Finally, we obtain the solution of α_s

$$\hat{\alpha}_s = \frac{1}{\sum_{s'=1}^{n_v}\left(\frac{A^{(s)}}{A^{(s')}}\right)^{\frac{1}{p-1}}} = \frac{1}{\sum_{s'=1}^{n_v}\left(\frac{\|\mathbf{V}^{(s)}\mathbf{Q}^{(s)}-\mathbf{V}^*\|_F^2}{\|\mathbf{V}^{(s')}\mathbf{Q}^{(s')}-\mathbf{V}^*\|_F^2}\right)^{\frac{1}{p-1}}}.$$

Appendix B

In this section, we provide the proofs for Proposition 1 and Theorem 2.

Proof of Proposition 1

First we assume that the denominators in Eq. (4) and (5) are always well-defined. The update rules by Lee and Seung (2001) are not well-defined if the denominators are 0. This may happen in a very rare case when all the terms on the denominators are 0. In such case, a small positive number can be added to avoid 0 (Lin, 2007). When it is added, the analyses keep the same, so we stick to the situations without the small positive number in this paper.

When $t = 1$, Theorem 1 holds by the assumption of this theorem. For $t > 1$, we prove by induction. We first prove for the case of $\mathbf{U}$. Assuming the results are true at tth iteration, we note that from t to $t+1$, the step size for updating u_{ik} in Eq. (4) is positive:

$$\frac{u_{ik}}{\left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{l=1}^M u_{lk} \sum_{j=1}^N v_{jk}^2} > 0.$$

We now consider two situations for the derivative $\nabla_U \mathcal{O}_0$:
Case 1: When $\nabla_U \mathcal{O}_0 = 0$, $u_{ik}^{t+1} = u_{ik}^t$ and it converges as the complementary slackness condition suggests.
Case 2: When $\nabla_U \mathcal{O}_0 \neq 0$,

$$\begin{aligned}
u_{ik} &\leftarrow u_{ik} - \frac{u_{ik}}{\left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{l=1}^M u_{lk} \sum_{j=1}^N v_{jk}^2} \times \frac{1}{2}\nabla_U \mathcal{O}_0 \\
&= u_{ik} - u_{ik} \frac{\frac{1}{2}\nabla_U \mathcal{O}_0}{\left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{l=1}^M u_{lk} \sum_{j=1}^N v_{jk}^2} \\
&= u_{ik} - u_{ik} \left(1 - \frac{\left[\mathbf{X}\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{j=1}^N v_{jk} v_{jk}^*}{\left[\mathbf{U}\mathbf{V}^\top \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{l=1}^M u_{lk} \sum_{j=1}^N v_{jk}^2}\right) \\
&> 0.
\end{aligned}$$

The inequality follows the definition of $\nabla_U \mathcal{O}_0$. When $\nabla_U \mathcal{O}_0 > 0$, the term in the bracket is between 0 and 1. When $\nabla_U \mathcal{O}_0 < 0$, the term in the bracket is negative. Both imply that $u_{ik}^{t+1} > 0$. The proof for $v_{jk}^t, \forall t \geq 1$ is the same as the proof for $\mathbf{U}$ and we omit it here.

Prior to the details of proving Theorem 2, we introduce a lemma that is essential to the proof.

Lemma 3 (Lee and Seung (2001)) *If $G(h, h')$ is an auxiliary function of $J(h)$, then $J(h)$ is nonincreasing under the update rule*

$$h^{(t+1)} = \underset{h}{\mathrm{argmin}} G(h, h^{(t)}). \tag{9}$$

Proof of Theorem 2

The updates for $\mathbf{V}^*$ and α_s give exact solutions for the minimization of $\mathcal{O}$ when others are fixed. Therefore, we only need to prove that $\mathcal{O}$ is nonincreasing under the update rules of $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$, $s = 1, ..., n_v$. Again to ease the notation without confusion, we drop (s) from the notations, and we simply write $\mathbf{V}$ and $\mathbf{U}$ to refer to a specific view.

The proof is established by defining an auxiliary function and showing the Taylor-expansion of the objective function is less than or equal to the auxiliary function. The update rules are element-wise, and we only need to show L_{ik} and J_{jk} are nonincreasing for Equations (4) and (5), where L_{ik} and J_{jk} denote the part of $\mathcal{O}$ relative to u_{ik} and v_{jk} only, respectively. They are the same as $\mathcal{O}_1$ and $\mathcal{O}_2$ as defined above. For u_{ik}, if we define the function

$$\begin{aligned}
&G\left(u_{ik}, u_{ik}^t\right) = L_{ik}\left(u_{ik}^t\right) + L'_{ik}\left(u_{ik}^t\right)\left(u_{ik} - u_{ik}^t\right) \\
&+ \left\{ \frac{\left(\mathbf{U}^t \mathbf{V}^T \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right)_{ik}}{u_{ik}^t} + \frac{\alpha_s^p \sum_{i=1}^M u_{ik}^t \sum_{j=1}^N v_{jk}^2}{u_{ik}^t} \right\} \left(u_{ik} - u_{ik}^t\right)^2,
\end{aligned}$$

then we have $G(u_{ik}, u_{ik}) = L_{ik}(u_{ik})$.

Next, we need to show $G(u_{ik}, u_{ik}^t) \geq L_{ik}(u_{ik})$. The Taylor expansion of $L_{ik}(u_{ik})$ gives

$$L_{ik}(u_{ik}) = L_{ik}(u_{ik}^t) + L'_{ik}(u_{ik}^t)(u_{ik} - u_{ik}^t) + \frac{1}{2}L''_{ik}(u_{ik}^t)(u_{ik} - u_{ik}^t)^2,$$

with the second order derivative $L''_{ik}(u_{ik}) = 2\left[\mathbf{V}^T \mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{kk} + 2\alpha_s^p \sum_{j=1}^N v_{jk}^2$. Comparing the Taylor-expansion of $L_{ik}(u_{ik})$ to $G(u_{ik}, u_{ik}^t)$, we only need to show

$$\frac{\left\{\mathbf{U}^t\mathbf{V}^T\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right\}_{ik}}{u_{ik}^t} + \frac{\alpha_s^p \sum_{i=1}^M u_{ik}^t \sum_{j=1}^N v_{jk}^2}{u_{ik}^t} \geq \left\{\mathbf{V}^T\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right\}_{kk} + \alpha_s^p \sum_{j=1}^N v_{jk}^2.$$

This is easy to verify by comparing the first and second terms of the above inequality, respectively. We have, according to the nonnegative constraints on $\mathbf{U}$ and $\mathbf{V}$

$$\left\{\mathbf{U}^t\mathbf{V}^T\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right\}_{ik} = \sum_l^K u_{il}^t \left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{lk} \geq u_{ik}^t \left\{\mathbf{V}^T\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right\}_{kk},$$

$$\alpha_s^p \sum_{i=1}^M u_{ik}^t \sum_{j=1}^N v_{jk}^2 \geq \alpha_s^p u_{ik}^t \sum_{j=1}^N v_{jk}^2.$$

Thus, $G(u_{ik}, u_{ik}^t)$ is an auxiliary function of $L_{ik}(u_{ik})$. Replacing $G(h, h^t)$ in Eq. (9) by $G(u_{ik}, u_{ik}^t)$, we have

$$\begin{aligned} u_{ik}^{t+1} &= u_{ik}^t - u_{ik}^t \frac{L'_{ik}(u_{ik}^t)}{2\left[\mathbf{U}\mathbf{V}^\top\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + 2\alpha_s^p \sum_{l=1}^M u_{lk} \sum_{j=1}^N v_{jk}^2} \\ &= u_{ik}^t \frac{\left[X\mathrm{Diag}^2(\boldsymbol{w})v\right]_{ik} + \alpha_s^p \sum_{j=1}^N v_{jk} v_{jk}^*}{\left[\mathbf{U}\mathbf{V}^\top\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}\right]_{ik} + \alpha_s^p \sum_{l=1}^M u_{lk} \sum_{j=1}^N v_{jk}^2}. \end{aligned}$$

The result follows Lemma 1 in Lee and Seung (2001) that L_{ik} is nonincreasing under the iteration $h^{t+1} = \arg\min_h G(h, h^t)$. Since the objective function is bounded below by 0, the monotone convergence theorem implies the convergence.

Similar statements for the proof of v_{jk} can be established by defining the auxiliary function

$$\begin{aligned} &G(v_{jk}, v_{jk}^t) = J_{jk}(v_{jk}^t) + J'_{jk}(v_{jk}^t)(v_{jk} - v_{jk}^t) \\ &+ \left\{ \frac{\left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}^t\mathbf{U}^\top\mathbf{U}\right]_{jk} + \alpha_s^p \left[\mathbf{V}^t\mathbf{Q}\mathbf{Q}^\top\right]_{jk} + \beta\left[\mathbf{D}\mathbf{V}^t\right]_{jk}}{v_{jk}^t} \right\} (v_{jk} - v_{jk}^t)^2. \end{aligned}$$

It is easy to see $G(v_{jk}, v_{jk}) = J_{jk}(v_{jk})$ and the remaining part is to show $G(v_{jk}, v_{jk}^t) \geq J_{jk}(v_{jk})$. The Taylor-expansion of $J_{jk}(v_{jk})$ gives

$$J_{jk}(v_{jk}) = J_{jk}(v_{jk}^t) + J'_{jk}(v_{jk}^t)(v_{jk} - v_{jk}^t) + \frac{1}{2}J''_{jk}(v_{jk}^t)(v_{jk} - v_{jk}^t)^2,$$

with the second order derivative $J''_{jk}(v_{jk}) = 2\left[\mathrm{Diag}^2(\boldsymbol{w})\right]_{jj}\left[\mathbf{U}^\top\mathbf{U}\right]_{kk} + 2\alpha_s^p\mathbf{Q}\mathbf{Q}^\top + 2\beta\left[\mathbf{L}\right]_{jj}$ (note that $\mathbf{L}$ is the graph Laplacian matrix defined in section 3). Comparing the Taylor-expansion of $J_{jk}(v_{jk})$ to $G(v_{jk}, v_{jk}^t)$, we are left to show

$$\begin{aligned} &\left\{ \frac{\left[\mathrm{Diag}^2(\boldsymbol{w})\mathbf{V}^t\mathbf{U}^\top\mathbf{U}\right]_{jk} + \alpha_s^p \left[\mathbf{V}^t\mathbf{Q}\mathbf{Q}^\top\right]_{jk} + \beta\left[\mathbf{D}\mathbf{V}^t\right]_{jk}}{v_{jk}^t} \right\} \\ &\geq \left[\mathrm{Diag}^2(\boldsymbol{w})\right]_{jj}\left[\mathbf{U}^\top\mathbf{U}\right]_{kk} + \alpha_s^p\mathbf{Q}\mathbf{Q}^\top + \beta\left[L\right]_{jj}. \end{aligned}$$

This can be verified by comparing the first and third terms of the inequality, respectively. We have, according to the nonnegative constraints on $\mathbf{U}$ and $\mathbf{V}$,

$$\left[\text{Diag}^2(\boldsymbol{w})\mathbf{V}^t\mathbf{U}^\top\mathbf{U}\right]_{jk} = \sum_l^K v_{jl}^t \left[\text{Diag}^2(\boldsymbol{w})\right]_{jj} \left[\mathbf{U}^\top\mathbf{U}\right]_{lk} \geq v_{jk}^t \left[\text{Diag}^2(\boldsymbol{w})\right]_{jj} \left[\mathbf{U}^\top\mathbf{U}\right]_{kk},$$

$$\beta\left[\mathbf{D}\mathbf{V}^t\right]_{jk} = \beta\sum_{l=1}^M d_{jl}v_{lk}^t \geq \beta d_{jj}v_{jk}^t \geq \beta\left[\mathbf{D}-\mathbf{A}\right]_{jj} v_{jk}^t = \beta L_{jj}v_{jk}^t.$$

Therefore, $G(v_{jk}, u_{jk}^t)$ is an auxiliary function of J_{jk}. Replacing $G(h, h^t)$ in Eq. (9) by $G(v_{jk}, v_{jk}^t)$, we have

$$\begin{aligned} v_{jk}^{t+1} &= v_{jk}^t - v_{jk}^t \frac{J'_{jk}\left(v_{jk}^t\right)}{\left[\text{Diag}^2(\boldsymbol{w})\mathbf{V}\mathbf{U}^\top\mathbf{U}\right]_{jk} + \alpha_s^p\left[\mathbf{V}\mathbf{Q}\mathbf{Q}^\top\right]_{jk} + \beta\left[\mathbf{D}\mathbf{V}\right]_{jk}} \\ &= v_{jk}^t \frac{\left[\text{Diag}^2(\boldsymbol{w})\mathbf{X}^\top\mathbf{U}\right]_{jk} + \alpha_s^p\left[\mathbf{V}\mathbf{Q}^\top\right]_{jk} + \beta\left[\mathbf{A}\mathbf{V}\right]_{jk}}{\left[\text{Diag}^2(\boldsymbol{w})\mathbf{V}\mathbf{U}^\top\mathbf{U}\right]_{jk} + \alpha_s^p\left[\mathbf{V}\mathbf{Q}\mathbf{Q}^\top\right]_{jk} + \beta\left[\mathbf{D}\mathbf{V}\right]_{jk}}. \end{aligned}$$

The result follows Lemma 1 in Lee and Seung (2001) that J_{jk} is nonincreasing under the iteration $h^{t+1} = \arg\min_h G(h, h^t)$. Since the objective function is bounded below by 0, the monotone convergence theorem implies the convergence.

Appendix C

Data information

1. **Synthetic dataset**: This synthetic dataset is generated by a four-component Gaussian mixture model. The data contains six views, with the last two views being noisy. More specifically, we randomly generate the cluster centers, denoted by $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_4$, for views $\mathbf{X}^{(1)}$ to $\mathbf{X}^{(4)}$. Each element of $\boldsymbol{\mu}_j$, $j = 1, ..., 4$ is independently drawn from the normal distribution with mean randomly generated from a uniform distribution $\mathcal{U}[a, a+10]$ and variance 1. We set $a = 10, 20, 30, 40$ for these four views. To generate the covariance matrix for each view, we first generate a random number b from $\mathcal{U}[0.1, 1]$, then multiply a symmetric matrix of all ones by b. Lastly, we take element-wise power of this matrix by a symmetric Toeplitz matrix whose diagonals are all 0. The prior proportions of the 4 components are set to be equal and sum to 1. Further, we set $\mathbf{X}^{(5)}$ to be the same as $\mathbf{X}^{(1)}$ but with the first 300 observations added by random noises independently generated from $\mathcal{N}(0, 5)$. We also let $\mathbf{X}^{(6)}$ to be the same as $\mathbf{X}^{(3)}$ but with the first 1000 observations added by random noises independently generated from $\mathcal{N}(0, 10)$.

2. **Handwritten digit dataset** [1]: This dataset contains 2000 digits and 10 labels. Each digit can be decomposed into four views: Fourier coefficients of the character shapes (**fou**), pixel averages in 2×3 windows (**pix**), Zernike moments (**zer**) and profile correlations (**fac**).

3. **Liver hepatocellular carcinoma (LIHC)**: This is a multi-omics dataset used in the application in Seal et al. (2020). Each sample has three different types of measurements (views): gene expression (GE), copy number variation (CNV), and DNA methylation (DNAm). The processed dataset has 404 samples, and the three views have 15397, 16384, 16384 features, respectively. To further reduce the dimension, we select the top 100 most highly variable features for each view. In addition, these samples belong to either tumor or normal samples, where such class labels are known a priori.

1. https://archive.ics.uci.edu/ml/datasets/Multiple+Features

Empirical analysis on the tuning parameters

In this section, we show how to select p, β, and K using the synthetic dataset and the handwritten digit dataset. The default values are $p = 5$, $\beta = 0.01$, and $K = 10$. During the experiment, we change the target parameter and fix the remaining ones.

We first analyze the effect of p on the algorithm performance. Figure 3 shows how the metric scores and the distribution of weights change as p changes. For the experiments, we let $p \in \{2, 4, 5, 8, 11\}$. The left panel shows p controls the sparsity of the weight vector, i.e., the effect of different values of p on the distributions of the weight vector. As p decreases, the weight vector $\boldsymbol{\alpha}$ becomes sparser. The right panel shows that all the metric scores are close with $p \in \{4, 5, 8, 11\}$ (a moderate size).

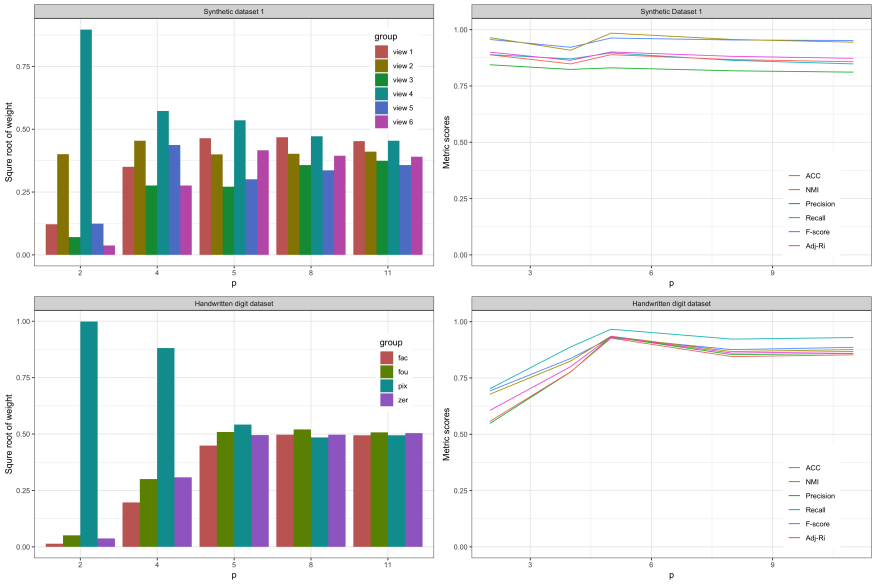

Figure 3: Distributions of view-specific weight $\boldsymbol{\alpha}$ (left) and the metric scores (right) under different values of p for the synthetic dataset and handwritten digit dataset.

Next we empirically illustrate how to choose β, the manifold regularizer. Before algorithm implementation, entries in $\mathbf{X}$ are scaled so that the value $\mathcal{O}_1$ is in general small. Consequently, we tend to choose a small β to balance the matrix factorization effect and the manifold regularization effect. As we can see from Figure 4, both datasets demonstrate robust results with different values of β. This implies that the clustering performance is robust to relatively small β values. Finally, we empirically show how to choose K. As we can see from Figure 5, both datasets demonstrate robust results when K lies in a neighbour of the ground truth. This means that the choice of K may not affect the clustering results even though it is overestimated or underestimated.

Complexity and convergence study

To study the computational complexity, we run a series of experiments on a server with 10 processors and each processor (2.2 GHz Intel Xeon) uses 20GB memory. We change N and n_v to investigate the corresponding effects. The default setting is 5000 data points ($N = 5000$), 4 views ($n_v = 4$) with 10 clusters ($K = 10$) and 100 features ($M = 100$). During the experiment, we change one aspect while keeping all others fixed. The values of N are set to 3k, 5k, 7k, 9k, and 11k, which is larger than M, so the theoretical complexity should be quadratic in N. We find the running time is overall linear in terms of smaller N and scales well for large N (e.g. $N > 9000$). Even though the theoretical result for $N > M$ indicates quadratic complexity in N, the

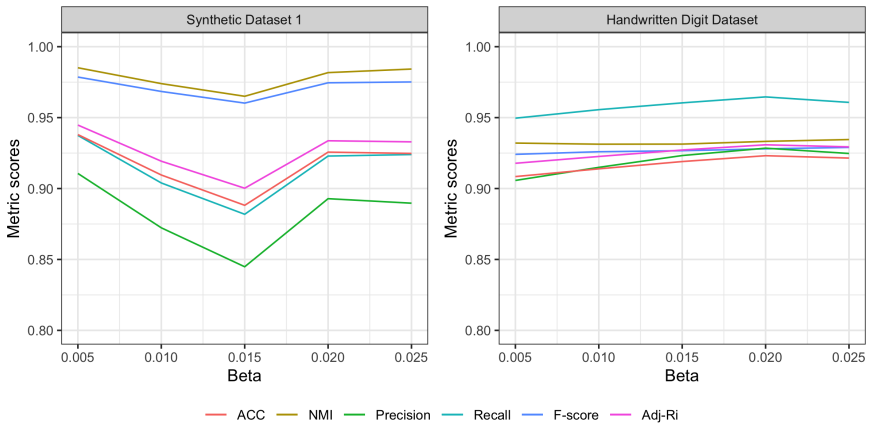

Figure 4: Metric scores under different values of β on the synthetic dataset (left) and handwritten digit dataset (right). The x-axis is the value of different β from 0.005 to 0.025. Note: we set the limits of the y-axis from 0.8 to 1.

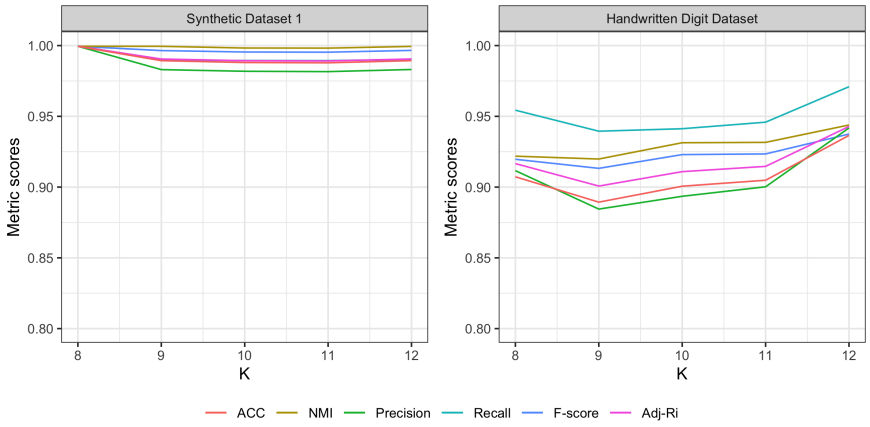

Figure 5: Metric scores under different values of K on the synthetic dataset and handwritten digit dataset. The x-axis shows the value of different K from 8 to 12. Note: we set the limits of the y-axis from 0.8 to 1.

running time is still acceptable. For n_v, we set its value from 2 to 6. Row 1 of Figure 6 shows the running time is linear in n_v.

The multiplicative update rule for minimizing the objective function $\mathcal{O}$ is iterative. Theorem 2 shows that the algorithm for updating $\mathbf{U}$ and $\mathbf{V}$ can converge to a local solution. Here we investigate how fast the convergence is empirically. Row 2 of Figure 6 demonstrates the convergence curves. The x-axis is the number of iterations and y-axis is the objective function value. We see that the algorithm converges very fast, usually within 50 iterations.

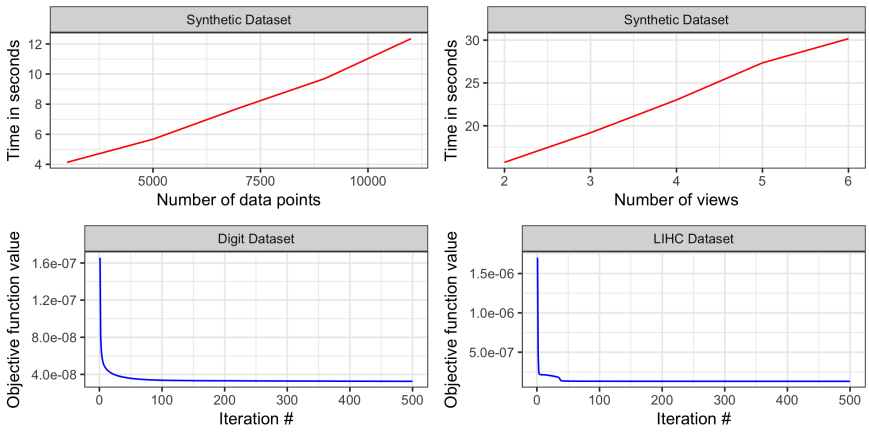

Figure 6: Running time of the WM-NMF algorithm on the synthetic dataset (row 1); convergence curves of WM-NMF algorithm on the handwritten digit and LIHC dataset (row 2).

Appendix D

Algorithm complexity analysis

We analyze the complexity for updating $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$ in the inner iteration. We divide the counts of iterations into multiplication, addition and division. The overall complexity for the inner iteration is $O(M_sNK + N^2K)$ (we provide the details in appendix A). We summarize the operation counts in Table 3.

Table 3: Computational operation counts for each iteration of $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$.

	multiplication	addition	division	overall
$\mathbf{U}^{(s)}$	$N + M_sN + M_sNK + (N+1)K +$ $N + KN + 2M_sNK + K(N+2)$	$KM_sN + NK +$ $2M_sNK + K(M_s + N)$	M_sK	$O(M_sNK)$
$\mathbf{V}^{(s)}$	$N + M_sN + M_sNK + 2NK + N^2K$ $N + KN + 2M_sNK + 3KN + K$	$KM_sN + N^2K +$ $2M_sNK + KN^2$	NK	$O(M_sNK + N^2K)$

Further, suppose there are t_1 iterations for updating $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$ for each view, then the complexity for all views is $O\left\{t_1 n_v (M_* NK + N^2K)\right\}$, where M_* denotes the maximum of $\{M_1, \ldots, M_{n_v}\}$. After the t_1 inner iterations of $\mathbf{U}^{(s)}$ and $\mathbf{V}^{(s)}$, we still need $O(n_v)$ for α_s, $O(n_vNK)$ for $\mathbf{V}^*$, and $O(n_vN)$ for Diag($\boldsymbol{w}$). Therefore, for each iteration of the whole procedure of Algoritm 1(lines 4-12), the total complexity is $O\left\{t_1 n_v (M_* NK + N^2K)\right\}$. Suppose t_2 outer iterations are taken for $\mathcal{O}$ to converge or reaching the maximum number of iteration, then the overall algorithm takes time $O\left\{t_1 t_2 n_v (M_* NK + N^2K)\right\}$ for $N > M_*$. When $N < M_*$, we have the overall complexity $O\left\{t_1 t_2 n_v M_* NK\right\}$.