Regularization Functions in Subspace Learning-based Feature Selection: Tutorial

Amir Moslemi

Regularization Functions in Subspace Learning-based Feature Selection: Tutorial

Amir Moslemi[1,2,]

[1] Department of Physics, Toronto Metropolitan University, Ontario, Canada

[2] Physical Sciences, Sunnybrook Health Sciences Centre, Toronto, ON M4N 3M5, Canada

Email: Amir.moslemi@ryerson.ca

Abstract

This is a tutorial about regularization functions for feature selection using subspace learning. In this tutorial, sparse regularization, structure learning regularization, rank minimization, redundancy minimization, soft label learning, self-paced learning and contrastive learning were explained. For sparse regularization: $\ell_{2,1}$, $\ell_{2,p}$ $(0 < p < 1)$, $\ell_{2,1-2}$, $\ell_{2,0}$ and inner-product norms were explained. For structure learning; Laplacian graph, Hessian graph, dynamic graph learning and hyper graph were covered and explained. For rank minimization and low-rank constraint; nuclear norm and Schatten p-norm $(0 < p < 1)$ were explained. This tutorial is appropriate for researchers and students who are interested in dimensionality reduction and feature selection. Each of the regularization function are mathematically explained and derived.

Keywords: Sparse regularization, structure learning regularization, rank minimization, redundancy minimization, soft label learning, self-paced learning and contrastive learning.

1 Introduction:

The development of artificial intelligence (AI) and information technology (IT) has significantly increased the dimensionality of data. Moreover, the expansion of AI and IT has diversified data sources, further contributing to the growth in data dimensions. To address challenges such as computational complexity and overfitting, dimensionality reduction is an essential step in machine learning. Two primary techniques for reducing dimensionality are feature extraction and feature selection [1]. However, feature extraction methods, such as principal component analysis (PCA) [2], linear discriminant analysis (LDA) [3], and autoencoders [4], often suffer from a lack of interpretability. In contrast, feature selection offers a dimensionality reduction approach that retains interpretability.

Filter, wrapper, embedded and hybrid are feature selection strategies. In filter strategy, there is no connection between feature selection and machine learning algorithm and features are selected based on their correlation and dependency with other features. In wrapper strategy, the important features are selected based on the performance of machine learning algorithm. In embedding strategy, features selection is part of training phase such as decision tree and LASSO regression [5].

Feature selection techniques are categorized to supervised, unsupervised and semi-supervised in terms of label availability. In supervised, features are ranked and selected based on relevancy of features to the labels. In unsupervised, information is extracted based on structure of features without label information. In semi-supervised, information is extracted from both labeled and unlabeled data [5].

Computation metaheuristic-based techniques [6], information theory-based techniques [5,7], reinforcement learning-based techniques [8] and subspace learning-based techniques [9] are the mainstream of feature selection. In this tutorial, we focused on subspace learning and essential regularization to improve the feature selection.

Subspace learning techniques look for a mapping function (matrix) to map data from original space to feature space with lower dimension [10]. In addition to projecting into lower dimension,

subspace learning techniques can remove the noisy and redundant features using appropriate regularization [11].

The subtle point should be mentioned is that information can be missed during projection and noisy data can be remained after mapping into feature space. To address these challenges, different regularization functions have been introduced to preserve the information and remove the noisy data. Sparsity, structure learning (geometrical), rank minimization, correlation, label guided, self-paced learning and contrastive learning are the main regularization functions which are widely used for feature selection using subspace learning. The Taxonomy of the main regularization functions for subspace learning-based feature selection techniques, is shown in Figure 1.

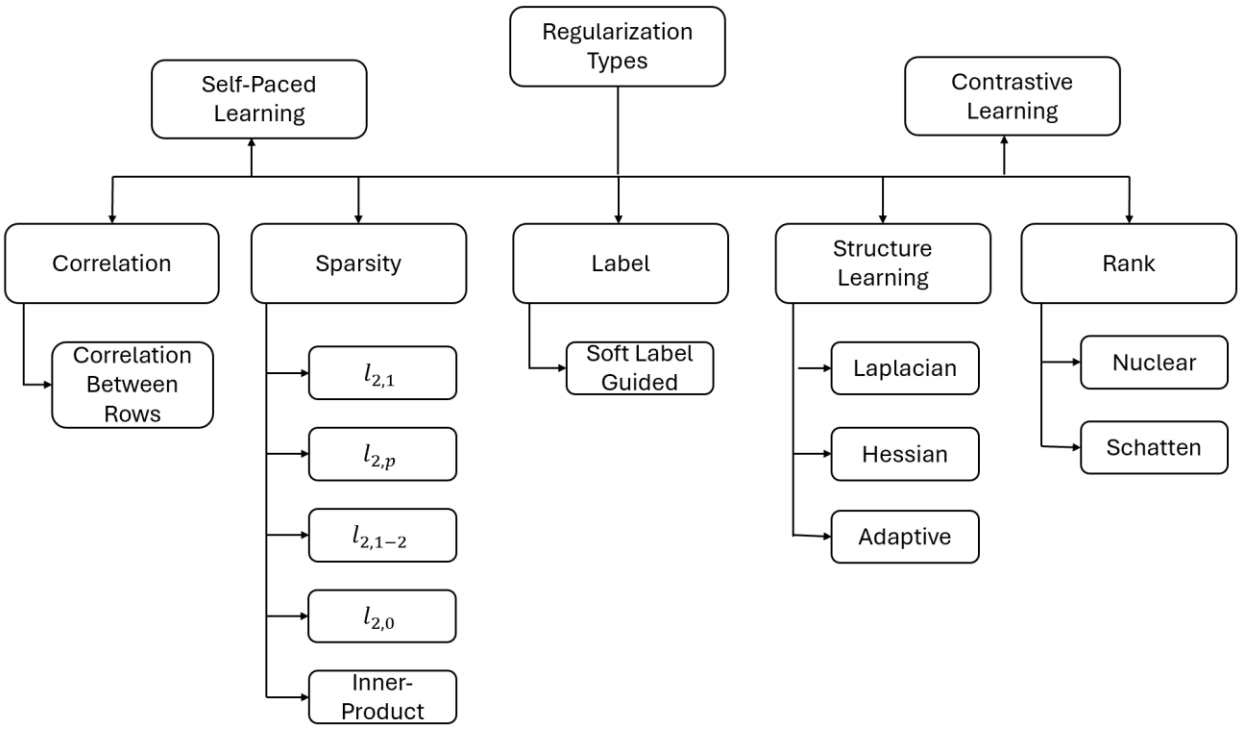

Figure 1. This figure shows the different types of regularization functions for subspace learning-based feature selection techniques.

2 Feature Selection using Subspace learning

The concept of feature selection through subspace learning involves identifying a subset of features that effectively represents the entire feature set. In other words, feature selection using subspace learning seeks to find a subset of features that can span or approximate the full set of features.

To explain more effectively, consider a data $X = \{f_1, f_2, \dots, f_d\}$, which has d-feature, and $X_k = \{f_1, f_2, \dots, f_k\}$, which is a subset of X such that $k < d$. Then, the matrix X_k is a submatrix of X ($X_I \subset X$).

The objective of feature selection using subspace learning is to approximate X_k as much as possible close to X. Consequently, feature selection using subspace learning is mathematically formulated as following:

$$arg\min_I \quad \vec{d}\,(span(X), span(X_k)) \tag{1}$$

Where $\vec{d}$ is a metric to evaluate distance between all features and k selected features, and k is the number of selected features.

In this tutorial we consider nonnegative matrix factorization (NMF) as subspace learning-based feature selection technique [10], and we solve feature selection with different regularization for NMF. NMF feature selection is described as follows:

$$\min \ \|X - XWH\|_F \ \ S.t \ \ W \geq 0 \ , \ H \geq 0, W^T W = I \tag{2}$$

Where $\|.\|_F$, X, W and H are Frobenius norm, data, feature weight matrix and coefficient matrix, respectively. $W \geq 0$, $H \geq 0$ and $W^T W = I$ are nonnegative constraint of feature weight matrix, nonnegative constraint of coefficient matrix and orthogonality constraint, respectively. The more frequent used notations are shown in Table 1.

Table 1. Notations frequently used in this tutorial.

Notation	Representation
$X \epsilon R^{n \times d}$	Data with *n* samples and *d* features
W	Feature weight matrix of NMF
H	Coefficient matrix of NMF
I	Identity matrix
$\|X\|_F$	Frobenius norm: $\|X\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d} x_{ij}^2}$
$\|X\|_{q,p}$	Mixed matrix norm: $\|X\|_{q,p} = (\sum_{i=1}^{d} \|x^i\|_q^p)^{\frac{1}{p}}$
†	pseudo inverse

2.1 Sparsity Regularization

The Euclidean norm of each row in the feature weight matrix W reflects the importance of each corresponding feature. Sparse regularization function is applied to preserve the global information of data and increase the robustness against outliers. Although $\ell_{2,0}$-norm is the ideal regularization to peak the exact k-top informative features, it is not convex and its problem is NP-hard [12].

2.1.1 Sparsity Regularization-$\ell_{2,1}$-norm

$\ell_{2,1}$-norm is a good approximation of $\ell_{2,0}$-norm and it can provide sufficient sparse solutions. Therefore, NMF feature selection (NMFFS) with $\ell_{2,1}$-norm regularization is expressed as follows:

$$\min \ \|X - XWH\|_F + \lambda\|W\|_{2,1} \quad S.t \ W \geq 0 \ , \ H \geq 0, W^T W = I \tag{3}$$

Where λ is the regularization coefficient to set a trade-off between the reconstruction error and regularization function. By applying partial derivative with respect to W, we have following for $\ell_{2,1}$-norm regularization.

$$\frac{\partial \|W\|_{2,1}}{\partial W} = \frac{\partial \, Tr(W^T QW)}{\partial W} = 2QW \tag{4}$$

Where Q is defined as following:

$$Q = diag(\frac{1}{\|w^i\|_2 + \varepsilon}) \tag{5}$$

Where w^i is i-th row of feature weight matrix W and ε s a small value to stabilize the fraction. Therefore, matrix Q is a diagonal matrix whose diagonal elements are $\frac{1}{\|w^i\|_2 + \varepsilon}$.

2.1.2 Sparsity Regularization-$\ell_{2,p}$-norm

The sparsity can be increased using $\ell_{2,p}$-norm ($0 < p < 1$) rather than $\ell_{2,1}$-norm. $\|W\|_{2,p} = Tr(W^T QW)$ is like $\|W\|_{2,1}$, but the matrix Q is defined as follows:

$$Q = diag(\frac{p}{\|w^i\|_2^{2-p} + \varepsilon}) \tag{6}$$

Although [13] reported the error of classification is reduced using $\ell_{2,1/2}$ ($p = 1/2$) and $\ell_{2,1/2}$ outperforms $\ell_{2,1}$, $\ell_{2,p}$-norm is neither convex nor Lipschitz continuous. Additionally, nonconvex

regularization on matrices has considerable complexity [15], and Lipschitz discontinuity directly affects the derivative in each iteration.

2.1.3 Sparsity Regularization-$\ell_{2,1-2}$-norm

$\ell_{2,1-2}$-norm is a nonconvex but Lipschitz continuous matrix norm [17], which is a sparse regularization function for feature selection [16]. $\ell_{2,1-2}$-norm has higher sparsity than $\ell_{2,1}$-norm and it is Lipschitz continuous in contrast to $\ell_{2,p}$-norm ($0 < p < 1$).

$\ell_{2,1-2}$-norm is the difference of $\ell_{2,1}$ and Frobenius norm and defined as follows:

$$\|W\|_{2,1-2} = \|W\|_{2,1} - \|W\|_{2,2} \tag{7}$$

$\ell_{2,1-2}$-norm regularization function is not convex since it is subtraction of two convex functions. ConCave-Convex Procedure (CCCP) technique can be applied to handle this nonconvexity. To apply CCCP, consider minimization problem for subtraction of two convex functions as follows:

$$\min_w Z(w) \coloneqq F(w) - G(w) \tag{8}$$

Where $F(.): R^n \to R$ and $G(.): R^n \to R$ are two convex functions. $Z(w)$ is nonconvex and minimization (2) is a nonconvex problem. For example, $F = x^2 + y^2$ and $G = 4(x + y)$ are two convex functions. Function $Z = F - G$ is a nonconvex function and Figure 2 shows the function Z.

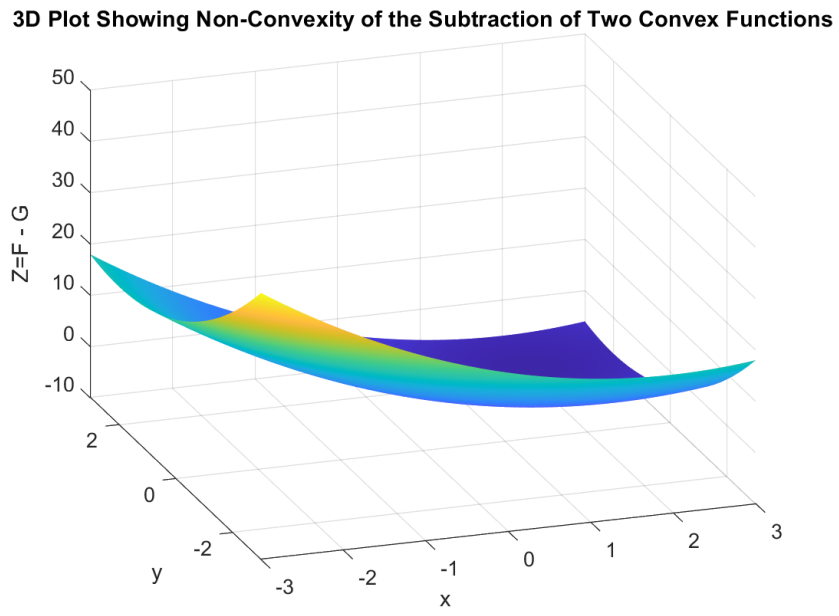

Figure 2. The plot of function $Z = F - G$, where $F = x^2 + y^2$ and $G = 4(x + y)$.

Problem (8) can be a convex function if and only if $G(w)$ is affine. CCCP can solve problem (8) by converting (8) into the series of convex subproblem. The main idea behind CCCP is the linearization of function $G(w)$ at each iteration in order to provide affine condition for second term. Therefore, CCCP solves (8) in the form of following iterations:

$$\begin{cases} \Delta^t \epsilon \frac{\partial G(w^t)}{\partial w} \\ w^{t+1} = arg\min_w F(w) - (G(w^t) + < \Delta^t, w - w^t >) \end{cases} \tag{9}$$

Where Δ^t the is linearized function of $G(w^t)$ at iteration t and $<,>$ represents the inner product. Therefore, $\|W\|_{2,1-2}$ norm can be written as convex function using CCCP as follows:

$$\|W\|_{2,1-2} = \|W\|_{2,1} - < W, \frac{\partial \|W\|_F}{\partial w} > \tag{10}$$

Where $\frac{\partial \|W\|_F}{\partial w}$ at iteration t is defined as follows:

$$\frac{\partial \|W\|_F}{\partial w} = \begin{cases} \frac{W^t}{\|W^t\|_F}, if\ W^t \neq 0 \\ 0, \quad Otherwise \end{cases} \tag{11}$$

The derivative of (10) with respect to W to update this variable is obtained as follows:

$$\frac{\partial \|W\|_{2,1-2}}{\partial w} = \frac{\partial}{\partial W} Tr(W^T QW) - \frac{\partial}{\partial W} < W, \frac{\partial \|W\|_F}{\partial w} > = QW - \frac{W^t}{\|W^t\|_F} \quad (12)$$

Where Q is defined in (5).

2.1.4 Sparsity Regularization-$\ell_{2,0}$-norm

As we mention in above, $\ell_{2,0}$-norm is exact sparse norm and is approximated by $\ell_{2,1}$-norm since it is not convex. $\ell_{2,1}$-norm and $\ell_{2,p}$-norm are not optimal and explicitly meaningful [18]. Additionally, $\ell_{2,1}$-norm and $\ell_{2,p}$-norm are the only slack versions of $\ell_{2,0}$-norm. $\ell_{2,1}$-norm and $\ell_{2,p}$-norm cannot select exact top-k features and only they can rank the features based on their score [18]. For example, we have four features F_1, F_2, F_3 and F_4 such that $F_1 > F_2 > F_3 > F_4$ in terms of score. We consider F_1 and F_2 If we want to pick two top features based on their score using $\ell_{2,1}$-norm and $\ell_{2,p}$-norm. Whereas the combination of F_1 and F_3 can provide better separability in subspace [19]. Figure 3 illustrates the difference between $\ell_{2,1}$-norm and $\ell_{2,0}$-norm.

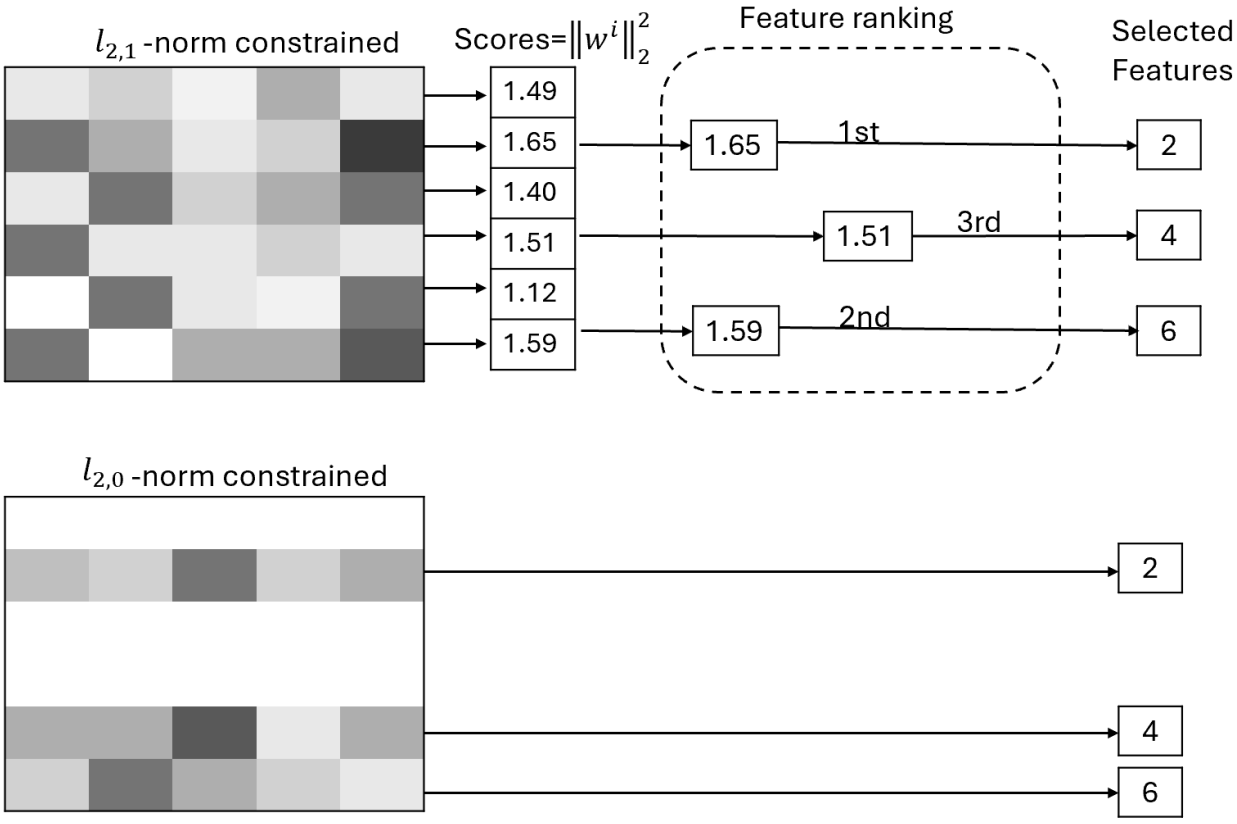

Figure 3. Different effect of sparsity $\ell_{2,1}$-norm and $\ell_{2,0}$-norm for feature selection. In this example, we set selected features equal to three (k=3).

As aforementioned, solving $\ell_{2,0}$-norm is difficult and leading to NP-hard problem. To circumvent $\ell_{2,0}$-norm challenge, a new optimization technique was proposed for feature selection using $l_{2,0}$-norm constraint directly [20]. To main challenge using $l_{2,0}$-norm constraint directly is that it cannot be applied for all types of feature selection objective function. To this end, consider following feature selection problem:

$$\max Tr(W^T XW) \quad s.t \quad W^T W = I, \|W\|_{2,0} = k \tag{13}$$

Where W is a projection matrix, X is positive semidefinite matrix (PSD) and k shows the exact selected top features.

If we consider the dimension of subspace equal to m, we have two scenarios for solving (13) including; $Rank(X) \leq m$ and $Rank(X) \geq m$.

A) $Rank(A) \leq m$*:* full rank decomposition W is considered as follows:

$$W = UV \tag{14}$$

Where U is indicator matrix (selection matrix) and matrix V is all nonzero rows of W. Therefore, $U \in \{0,1\}^{d\times k}$ and satisfy $U^T 1_d = 1_k$, and $V^T V = I$, where d is the dimension of features. We know $V^T V = VV^T = I$, then we can consider $Tr(W^T XW) = Tr(V^T U^T XUV) = Tr\left(V^T \tilde{X} V\right) = Tr\left(\tilde{X} VV^T\right)$. Therefore, the problem (13) is converted into following form:

$$\max Tr(\tilde{X}) \quad s.t \quad \tilde{X} \in \mathcal{M}_m(X) \tag{15}$$

Where $\mathcal{M}_m(X) = \{U^T XU : U \in \{0,1\}^{d\times k}, U^T 1_d = 1_k\}$. The solution of maximization problem (13) is top- k diagonal elements of X and this solution is global maximum [20, Algorithm 1] . To solve the maximization problem (15), we have three steps as follows:

1- Sorting diagonal elements of X .
2- Assigning the rows and columns corresponding to top- k element to obtain k-order principal submatrix $\tilde{X}$ of X ($\tilde{X} \in \mathcal{M}_m(X)$)
3- Applying eigen-decomposition on $\tilde{X}$ to obtain W.

B) $Rank(X) > m$: In this scenario, strategy $Rank(X) \leq m$ cannot be used for since $Tr\left(\tilde{X}\right) \neq \sum_i^m \lambda_i(\tilde{X})$. To use strategy $Rank(X) \leq m$, we first need to construct a low-rank proxy PSD matrix such that $Rank(P) \leq m$. Consequently, matrix P is defined as follows:

$$P = XW(W^T XW)^{\dagger} W^T X \tag{16}$$

Where † is pseudo inverse. Matrix P is the best approximation of matrix X. We can solve problem (16) for P same problem (13) for X [20, Algorithm 2].

$l_{2,0}$-norm constraint can be directly applied for feature selection using other objective function. [21] used Linear Discriminant Analysis (LDA) objective function for feature selection using $l_{2,0}$-norm constraint, and [22] used latent representation learning objective function.

2.1.5 Sparsity Regularization-Inner product-norm

$\ell_{2,1}$-norm regularization is lonely not appropriate to obtain both the high sparsity and low redundancy. Inner-product norm regularization was proposed to preserve the sparsity and to determine the independence of variables [23]. Concretely, inner-product norm is the combination of ℓ_1-norm for WW^T and ℓ_2-norm for W. Inner-product norm is formulated as follows:

$$\Omega(\mathrm{W}) := \sum_{i,j=1,i\neq j}^{d} < w^i, w^j > = \sum_{i,j=1,i\neq j}^{d} w^i w^{jT} \; = Tr(1_{d\times d} WW^T) - Tr(WW^T) \qquad (17)$$

The derivative of (17) with respect to W in order to update this variable is obtained as follows:

$$\frac{\partial \sum_{i,j=1,i\neq j}^{d} <w^i, w^j>}{\partial w} = 1_{d\times d} W - W \qquad (18)$$

2.2 Structure Learning Regularization-Manifold Information Preservation

2.2.1 Laplacian Graph Matrix

Structure (local) learning regularization is essential to preserve data geometry information (Topology of data) in subspace learning [24]. More precisely, structure learning aims to preserve the geometry of data after projection into subspace. The idea behind structure learning (manifold learning) is that if we have two samples x_i and x_j are close in original space of data (before mapping), the corresponding samples x'_i and x'_j in the subspace should be close (after mapping). For data $X \in R^{n\times d}$, a mapping function H is used to project X into subspace $X' \in R^{n\times d'}$ such that $d >> d'$. We considered $\|E\|_M^2 = \int \|\nabla_M E\|^2$ to measure the smoothness of E along the geodesic of data ($M \subset R^d$ is submanifold and ∇_ME is gradient of E with M). $\|E\|_M^2$ cannot be obtained in continuous form, since submanifold M is not known. To address this, $\|E\|_M^2$ must be discretely approximated as follows [25]:

$$\|E\|_M^2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|x'_i - x'_j\right\|_2^2 \mathrm{s}_{ij} = \frac{1}{2}\sum_{i=1}^{n} x_i x_i^T \sum_{j=1}^{n} \mathrm{s}_{ij} + \frac{1}{2}\sum_{j=1}^{n} x_j x_j^T \sum_{i=1}^{n} \mathrm{s}_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j^T \mathrm{s}_{ij} \qquad (19)$$

Where x' and $S \in R^{n \times n}$ are projected sample and symmetric affinity matrix (s_{ij} is element of matrix S), respectively. Degree matrix is $D_{ii} = \sum_{j=1}^{n} s_{ij}$ and equation (19) can be reformulated as follows:

$$\|E\|_M^2 = \frac{1}{2}\sum_{i=1}^{n} x'_i {x'_i}^T D_{ii} - \sum_{i=1}^{n}\sum_{j=1}^{n} x'_i {x'_j}^T s_{ij} = Tr(X'^T(D - S)X') = Tr(X'^T LX') \quad (20)$$

Where $X' = \sum_{i=1}^{n} x'_i$ is matrix of samples in subspace (projected matrix) and $L = D - S$ is graph Laplacian of matrix X. n is the number of samples. The affinity matrix is calculated using heat kernel as follows:

$$s_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma}\right) & if \;\; x_j \in N_k(x_i) \;\; or \;\; x_i \in N_k(x_j) \\ 0 \end{cases} \quad (21)$$

Where $N_k(x_i)$ is the set of K-nearest neighbours of sample $x_i \notin N_k(x_i)$. σ is the Gaussian scale parameter and it is recommended to be set as follows [26]:

$$\sigma = \sum_{i,j}^{n} \frac{\sqrt{\|x_i - x_j\|_2^2}}{n^2} \quad (22)$$

The structure learning (locally preserving) regularization term is $Tr(W^T X^T LXW)$ for NMFFS and derivative with respect to W to update this variable is obtained as follows:

$$\frac{\partial Tr(W^T X^T LXW)}{\partial w} = 2X^T LXW \quad (23)$$

2.2.2 Hessian Graph Matrix

Geodesic and null space consideration are two main limitations for Laplacian graph regularization. Laplacian graph regularization may face difficulty to preserve manifold of data after projection due to bias towards a constant geodesic function [27]. Hessian regularization is an alternative to circumvent Laplacian graph regularization limitations. Hessian regularization has rich null space and can preserve the geometrical information of data (manifold) stronger than Laplacian graph regularization.

For smooth manifold $M \subset R$, Eelles-energy is defined for mapping function $h: M \to R$ as follows:

$$E_{Elles}(h) = \int_M \|\nabla_a \nabla_b h\|^2_{T_x^* M \otimes T_x^* M} dV(x) \quad (24)$$

Where $T_x^* M$ is local tangent space for point $x \subset M$, mapping function and $\nabla_a \nabla_b h$ is the second covariant derivative of h. $dV(x)$ is the natural volume element on manifold M [28]. Normal coordinates system is needed to evaluate Hessian energy function on manifold M. Normal coordinates x^r centered at x is estimated as follows:

$$\|\nabla_a \nabla_b h\|^2_{T_x^*M \otimes T_x^*M} = \sum_{r,s=1}^{l} \left(\frac{\partial^2 h}{\partial x^r \partial x^s}\right)^2 \quad (25)$$

(25) shows that the norm of second order derivative is the Forbenius norm of the Hessian of h in normal coordinates at point x.

To construct local normal coordinate, K-nearest neighbors ($\aleph_k(x_i)$) is used and for the local tangent space T_x^*M estimation, principal component analysis (PCA) is used. Therefore, local tangent space T_x^*M is utilized to estimate normal coordinates x^r of a point $x_q \in \aleph_k(x_i)$, where $1 \leq q \leq k$. The Hessian of h at x_i is estimated as follows:

$$\frac{\partial^2 h}{\partial x^r \partial x^s}\Big|_{x_i \approx \sum_{q=1}^{k} Z_{rsq}^{(i)} h_q} \quad (26)$$

Where $h(x_q) = h_q$ and $Z_{rsq}^{(i)}$ is a local Hessian operator of sample x_i in the normal coordinates x^r of a point $x_q \in \aleph_k(x_i)$. A second-order polynomial $p(x)$ is fitted to calculate $Z_{rsq}^{(i)}$ in normal coordinates to $h(x_q)_{q=1}^{k}$ as follows:

$$p^i(x) = h(x_i) + \sum_{r=1}^{l} H_r x^r + \sum_{r=1}^{l} \sum_{s=r}^{l} N_{rs} x^r x^s \quad (27)$$

Where $h(x_i)$ is the zeroth-order term. (27) is the second-order Taylor expansion of h around the point x_i with zero neighborhood size. Then, H_r and N_{rs} are

$$H_r = \frac{\partial h}{\partial x^r}|_{x_i} \quad , \quad N_{rs} = \frac{1}{2}\frac{\partial^2 h}{\partial x^r \partial x^s}|x_i \quad (28)$$

with symmetric property $N_{rs} = N_{sr}$. By using standard linear squares for fitting the polynomial, we have following:

$$\text{argmin}_{\upsilon \in R^p} \sum_{q=1}^{k} \left(\left(f(x_q) - f(x_i)\right) - (\Phi \upsilon)\right)^2 \quad (29)$$

Where $\Phi \in R^{k \times p}$ is the design matrix with $p = n + \frac{n(n+1)}{2}$. The basis functions of Φ are the monomials of the normal coordinates (centered at x_i) of $x_q \in \aleph_k(x_i)$ up to second order. $\upsilon = \Phi^{\dagger} w$ is the solution of (29), where $w = (w_1, w_2, \dots, w_k)^T \in R^k$ and $w_q = f(x_q)$.

Assuming $h(x_\pi) = w_\pi$, Forbenius norm of the Hessian h at x_i is approximated as follows:

$$\|\nabla_a \nabla_b h\|^2 \approx \sum_{r,s=1}^{n} \left(\sum_{\pi=1}^{k} Z_{rs\pi}^{(i)} h_\pi\right)^2 = \sum_{\pi,\mu=1}^{k} h_\pi h_\mu O_{\pi\mu}^{(i)} \quad (30)$$

Where $O_{\pi\mu}^{(i)} = \sum_{r,s=1}^{n} O_{rs\pi}^{(i)} O_{rs\mu}^{(i)}$. Finally, the Hessian energy function is derived as follows:

$$S_H(h) = \sum_{i=1}^{n} \sum_{\pi \in \aleph_k(x_i)} \sum_{\mu \in \aleph_k(x_i)} h_\pi h_\mu O_{\pi\mu}^{(i)} = J^T O J \quad (31)$$

Where $O = \sum_{i=1}^{n} O^{(i)}$. The Hessian matrix is computed by following steps:

1- Use KNN (K-nearest neighbors) to construct a neighbour matrix, which can be called K-matrix.

2- Apply singular value decomposition (SVD) on k-matrix to obtain a tangential coordinate system.
3- Use least square method to estimate the Hessian energy.
4- Calculate Structure learning regularization using $\Omega(V) = \sum_{i=1}^{n} \sum_{\pi \in \aleph_k(x_i)} \sum_{\mu \in \aleph_k(x_i)} h_\pi h_\mu O_{\pi\mu}^{(i)} = J^T O J$. Where $\aleph_k$ is neighbor matrix, π shows the rows and μ shows the columns.

(The matlab code to construct Hessian matrix is found https://www.ml.uni-saarland.de/code/HessianSSR/HessianSSR.htm)

Hessian regularization can be added to objective function like Laplacian graph regularization. Therefore, Hessian regularization in the form of $Tr(W^T X^T O X W)$ is added to NMFFS, where O is Hessian matrix.

2.2.3 Dynamic Graph Learning (Adaptive affinity matrix)

The main limitation of Laplacian graph matrix with fixed affinity matrix is that input data matrix $X \in R^{n \times d}$ can be noisy and has outliers. Therefore, the constructed affinity matrix is affected by noisy data and outliers, and Laplacian graph matrix is subsequently destructed.

For affinity matrix $S \in R^{n \times n}$, each element s_{ij} is the probability of connectivity between two samples x_i and x_j. Two close samples have larger s than two far samples, which shows that higher probability to be neighbors. To have dynamic Laplacian graph, affinity matrix must be updated in each iteration. Consequently, the structure learning regularization with adaptive graph is formulated as follows [29]:

$$\min \sum_{i,j} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \alpha {s_{ij}}^2 \right) \quad s.t \qquad \|S\|_1 = \mathbf{1^d}, 0 \le s_{i,j} \le 1 \tag{32}$$

Where $\mathbf{1^d}$ is a vector with all elements one and $s_{i,j}$ is the probability of sample i and j are neighbours. W is feature weight matrix of NMFFS. α is an important coefficient to avoid the trivial solution, since without α the optimal solution is $s_{i,j} = 1$ if two samples are neighbors. To solve minimization (32) with respect to $s_{i,j}$, The problem (32) for ith sample can be expressed as follows:

$$min \sum_{j} \left(\left\| W^T x_i - W^T x_j \right\|_2^2 s_{ij} + \alpha {s_{ij}}^2 \right) \quad s.t \quad \|S\|_1 = \mathbf{1^d}, 0 \le s_{i,j} \le 1 \tag{33}$$

Let have $P \in R^{n \times n}$ with elements $p_{ij} = \left\| W^T x_i - W^T x_j \right\|_2^2$. Therefore, (33) is reformulated as follows:

$$min \left\| s_i + \frac{1}{2\alpha} p_j \right\|_2^2 \quad s.t. \; s_i^T \mathbf{1} = 1, \; 0 \le s_{i,j} \le 1 \tag{34}$$

Using augmented Lagrangian technique we have following:

$$L(s_i, \theta, \Phi_i) = \frac{1}{2} \left\| s_i + \frac{1}{2\alpha} p_j \right\|_2^2 - \theta(s_i^T \mathbf{1} - 1) - {\Phi_i}^T s_i \tag{35}$$

Where θ and Φ_i are Lagrangian multipliers and the optimal solution of s_i is achieved by Karush-Kuhn-Tucker conditions. Lagrangian multiplier θ is obtained as follows:

$$\theta = \frac{1}{k} + \frac{1}{2k\alpha_i}\sum_{j=1}^{k} p_{ij} \tag{36}$$

where k is the number of connected neighbors to ith sample [30]. The optimal solution of s_i is obtained as follows:

$$s_{ij} = \left(-\frac{1}{2\alpha_i}p_{ij} + \theta\right)_+ \tag{37}$$

Parameter α with k neighbors connected to ith sample is obtained as follows:

$$\alpha_i = \frac{k}{2}p_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k} p_{ij} \tag{38}$$

Each element of affinity matrix S is updated by s_{ij} (37) and thus Laplacian matrix L is updated at each iteration.

2.2.4 Dynamic Graph Learning-Rank Constrained Laplacian Graph

The ideal scenario is that affinity matrix S has exact c connected components (c is the number of sample categories), which is not possible for real world dataset. However, affinity matrix S can have exact c connected components by applying rank constraint on graph Laplacian matrix. Therefore, $rank(L_s) = n - c$ must be considered as a constraint for structure learning regularization [31]. It is proven that $rank(L_s)$ is equivalent to $\sum_{i=1}^{c}\sigma_i(LS) = 0$ where σ_i is ith small singular value of Laplacian matrix and c is the number of sample categories. The constraint $rank(L) = n - c$ is not convex. To this end, Ky Fan's theorem [32] is applied to reformulate $rank(L)$ constraint to following solvable problem.

$$\min \sum_{i=1}^{c}\sigma_i(L) = \min Tr(F^T LF) \quad s.t \quad F^T F = I\ , F \in R^{n\times c} \tag{39}$$

The optimization problem (39) is an eigen problem and the optimal solution for variable F is obtained by the c-eigenvector of L corresponding to the c smallest eigenvalues. F can preserve the structure of cluster. Rank constrained is applied to preserve the structure of clusters as following:

$$\begin{aligned} \min_{s.t} \quad & \sum_{i,j}\left(\left\|W^T x_i - W^T x_j\right\|_2^2 s_{ij} + \alpha {s_{ij}}^2\right) + \lambda Tr(F^T LF) \\ s.t \quad & \|S\|_1 = \mathbf{1^d}, 0 \le s_{i,j} \le 1, F^T F = I\ , F \in R^{n\times c} \end{aligned} \tag{40}$$

Where λ is the regularization coefficient and $Tr(F^T LF) = \sum_j \left\|f_i - f_j\right\|_2^2 s_{ij}$. Problem (40) is solved like (34). Let we introduce matrix $M = \left\|W^T x_i - W^T x_j\right\|_2^2$ and matrix $N \in R^{n\times n}$ with elements $n_{ij} = \left\|f_i - f_j\right\|_2^2$ and then we have vector $p_{ij} = m_{ij} + \lambda n_{ij}$. Therefore, we have

problem (34) here and we can update s_{ij} by (37). The steps of adaptive affinity matrix and dynamic Laplacian graph are as follows:

1- Initialize affinity matrix S.
2- Calculate Laplacian graph matrix L.
3- Obtain optimal matrix F by (39): the c-eigenvector of L corresponding to the c smallest eigenvalues.
4- Update Matrix S by (37), where $p_{ij} = m_{ij} + \lambda n_{ij}$, $n_{ij} = \left\|f_i - f_j\right\|_2^2$ and $M = \left\|W^T x_i - W^T x_j\right\|_2^2$.
5- Update Laplacian graph matrix L.

2.2.5 Dynamic Graph Learning-Maximizing the Information Entropy of Similarity Matrix

While the central concept in graph theory is that high similarity between two samples suggests they belong to the same class, constructing a highly detailed similarity matrix can increase model complexity and the risk of overfitting. Maximizing the information entropy of similarity matrix is an approach to have adaptive affinity matrix [42]. Based on maximum entropy theorem, the similarity matrix can be optimized by : $\max_{\sum_{j=1}^{n} s_{ij}=1, s_{ij}>0} \sum_{i=1}^{n}\sum_{j=1}^{n} -s_{ij}\log s_{ij}$. To add dynamic graph learning to NFFS based on maximum entropy adaptive affinity matrix, $\min_{F^TF=I\ , F\in R^{n\times c}, \sum_{j=1}^{n} s_{ij}=1, s_{ij}>0} Tr(F^T LF) + \beta \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}\log s_{ij}$ must be added to objective function of NMFFS.

2.2.6 Hyper Graph Learning

The classical graph structure learning extract geometrical information based on data points and their neighbors [34]. Therefore, the high-order relationships among the points are omitted. Hyper-graph regularization is an approach to consider the high-order relationships for data points [33,34]. A hypergraph $G = (V, E, W_e)$ is created by edges $E = \{e_i|1,2,\dots,n\}$, vertex $V = \{v_i|1,2,\dots,n\}$, and hyper-edge weight W_e. Figure 4 shows an example of hypergraph [37].

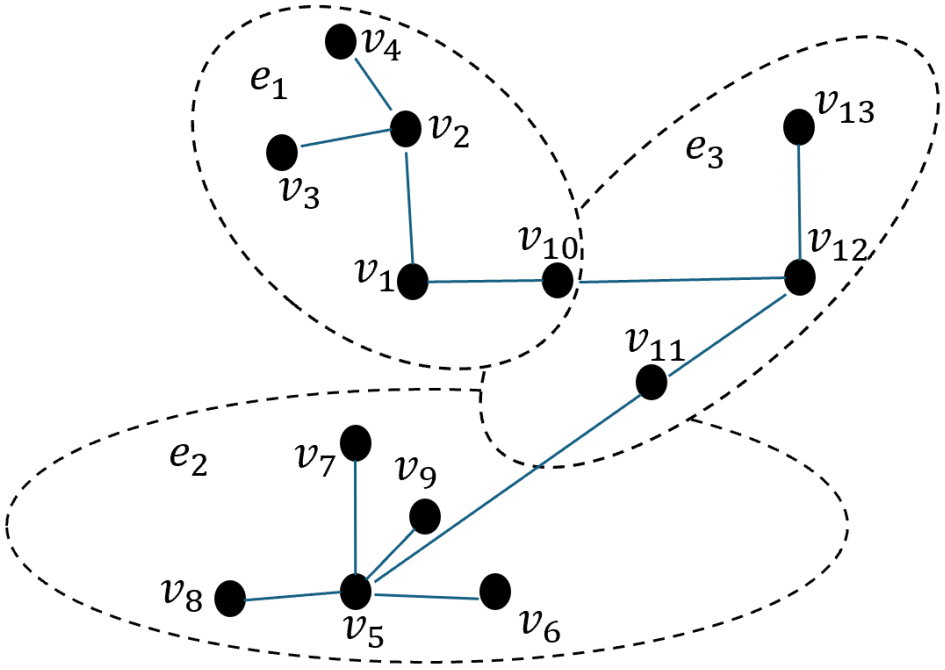

Figure 4. This figure shows a hypergraph with three edges and 13 vertices.

There are three sequential steps to construct a hypergraph for data points. In first step, the incidence matrix, which shows the binary vertex-edge relationship, must be constructed [35]. The incidence matrix is obtained as follows:

$$H(v_i, e_j) = \begin{cases} 1, & v_i \in e_j \\ 0, & otherwise \end{cases} \tag{41}$$

In second step, hyper-edge weight W_e, which measures the importance of hyperedges, must be obtained. In the last step, the normalized Laplacian matrix of the hyper-graph must be calculated [36].

Hyper edge e_i can be calculated by following formula:

$$e_i = \{v_j | sim(x_i, x_j) \leq 0.1\sigma_i\}, \quad i, j = 1,2, \dots, n \tag{42}$$

Where $sim(x_i, x_j)$ measures the similarity between two samples x_i and x_j (cosine similarity and Euclidean distance) and σ_i is the average distance between sample x_j and rest of the samples. Affinity matrix formula (21) can be used to initialize the W_e as follows:

$$W_e^i = \sum_{v_i \in e_j} s_{ij} \tag{43}$$

Where s_{ij} represents the element of affinity matrix.

The degree of each vertex and the degree of each hyper-edge are calculated as following:

$$d(v) = \sum_{e_i \epsilon E} w(e)\, H(v,e) \tag{44}$$

$$d(e) = \sum_{v_j \epsilon V} H(v,e) \tag{45}$$

Therefore D_V is a diagonal matrix whose elements are associated with vertex degrees and the D_e is a diagonal matrix of hyper-edge degree.

Unnormalized Laplacian hyper graph is obtained as follows:

$$L_{Hyper} = D_V - HW_e D_e^{-1} H^T \tag{46}$$

And normalized Laplacian hyper graph is obtained as follows:

$$L_{Hyper} = I - {D_V}^{-1/2} H W_e D_e^{-1} H^T {D_V}^{-1/2} \tag{47}$$

L_{Hyper} is embedded into $Tr(W^T X^T L_{Hyper} XW)$ and added to NMFFS objective function.

2.3 Rank Constraint-Low Rank Learning

2.3.1 Nuclear Norm-$\|.\|_*$

Imposing a rank constraint on an optimization problem reduces feature redundancy and facilitates the extraction of the true low-rank structure of the matrix. Solving an optimization problem with rank constraint is led to a nonconvex problem. To tackle this problem, rank constraint is approximated by nuclear norm [38]. For NMFFS with rank minimization we have following:

$$\min \ \|X - XWH\|_F \ + \alpha \|W\|_* \ S.t \ W \geq 0 \ , \ H \geq 0, W^T W = I \tag{48}$$

Where $\|.\|_*$ and α represent nuclear norm (sum of singular values) and regularization coefficient, respectively. Although $\|W\|_*$ is the main challenge to solve problem (48), derivative with respect to W can be directly applied and no auxiliary function needed.

Derivative of $\|W\|_*$ with respect to W:
Let consider singular value decomposition of W, $SVD(W) = U\Sigma V^T$, where U and V are orthonormal matrices ($U^T U$ =I, $V^T V$ =I) and Σ is diagonal matrix. Derivative of $\|W\|_*$ with respect to W is formulated as follows [47]:

$$\frac{\partial \|W\|_*}{\partial W} = U\Sigma^{\dagger}|\Sigma|V^T \tag{49}$$

Where ŧ is the Moore-Penrose pseudo-inverse. The nuclear norm can be defined as follows:

$$\|W\|_* = tr\left(\sqrt{W^T W}\right) = tr\left(\sqrt{\left(U\Sigma V^T\right)^T (U\Sigma V^T)}\right) = tr\left(\sqrt{V\Sigma U^T U\Sigma V^T}\right) = tr(\sqrt{V\Sigma^2 V^T}) \quad (50)$$

By applying the circularity property of trace, we obtain $tr\left(\sqrt{V\Sigma^2 V^T}\right) = tr\left(\sqrt{V^T V\Sigma^2}\right) = tr(|\Sigma|)$. Therefore, the subgradient problem is obtained as follows:

$$\frac{\partial\|W\|_*}{\partial W} = \frac{\partial tr(|\Sigma|)}{\partial W} = \frac{tr(\partial|\Sigma|)}{\partial W} \quad (51)$$

The subdifferential set of diagonal matrix $|\Sigma|$ is

$$\frac{\partial|\Sigma|}{\partial W} = \Sigma^{ŧ}|\Sigma|\frac{\partial\Sigma}{\partial W} \quad (52)$$

Substituting (52) into (51), we obtain:

$$\frac{tr(\partial|\Sigma|)}{\partial W} = \frac{tr(\Sigma^{ŧ}|\Sigma|\partial\Sigma)}{\partial W} \quad (53)$$

Addition, we know

$$\partial W = \partial U\Sigma V^T + U\partial\Sigma V^T + U\Sigma\partial V^T \quad (54)$$

Which can be expressed as follows:

$$U\partial\Sigma V^T = \partial W - \partial U\Sigma V^T - U\Sigma\partial V^T \quad (55)$$

By multiplying V from right-side and U^T form left side to (55) we obtain following:

$$\partial\Sigma = U^T\partial W V - U^T\partial U\Sigma - \Sigma\partial V^T V \quad (56)$$

In this step we need to have $tr(U^T\partial U\Sigma) = \text{tr}(\Sigma\partial V^T V) = 0$ to obtain $\frac{\partial\|W\|_*}{\partial W}$.

Proof $tr(U^T\partial U\Sigma) = \text{tr}(\Sigma\partial V^T V) = 0$:

Based on trace property ($Tr(W) = Tr(W^T)$) we can express $Tr(U^T\partial U\Sigma)$ as follows:

$$Tr(U^T\partial U\Sigma) = Tr((U^T\partial U\Sigma)^T) = Tr(\Sigma^T\partial U^T U) \quad (57)$$

Based matrix analysis, we know that $\partial I = 0$ and $U^T U = I$ and therefore we have following expression:

$$0 = \partial I = \partial(U^T U) = \partial U^T U + U^T\partial U \quad (58)$$

And following conclusion is achieved:

$$\partial U^T U = -\partial U U^T \quad (59)$$

Consequently, (57) can be reformulated as follows:

$$Tr(\Sigma^T\partial U^T U) = -Tr(\Sigma^T U^T\partial U) = -Tr(U^T\partial U\Sigma) \quad (60)$$

We found that $Tr(U^T\partial U\Sigma) = -Tr(U^T\partial U\Sigma)$ which means that $Tr(U^T\partial U\Sigma)$ =0. therefore, we proved that $tr(U^T\partial U\Sigma) = \text{tr}(\Sigma\partial V^T V) = 0$. We can conclude that $\partial\Sigma = U^T\partial Y V$ and $\frac{\partial\|W\|_*}{\partial W}$ is as follows:

$$\frac{\partial\|W\|_*}{\partial W} = \frac{tr(\partial|\Sigma|)}{\partial W} = \frac{tr(\Sigma^{ŧ}|\Sigma|\partial\Sigma)}{\partial W} = \frac{tr(\Sigma^{ŧ}|\Sigma|U^T\partial W V)}{\partial W} = \frac{tr(V\Sigma^{ŧ}|\Sigma|U^T\partial W)}{\partial W} = (V\Sigma^{ŧ}|\Sigma|U^T)^T \quad (61)$$

The partial derivative $\|W\|_*$ with respect to W is $(V\Sigma^{\dagger}|\Sigma|U^T)^T$, where $SVD(W) = U\Sigma V^T$.

2.3.2 Schatten p-norm ($0 < p < 1$)- $\|.\|_{Sp}^p$

Schatten p-norm ($0 < p < 1$) can be applied as rank constraint and studies showed that it has better rank approximation than nuclear norm [39,40]. Schatten p-norm can improve the abilty of feature selection algorithm to extract the inherent low-rank property of data [40]. Schatten p-norm is defined as follows:

$$\|W\|_{Sp}^p = \sum_{i=1}^{\min\{n,d\}} \sigma_i^p = Tr(W^T W)^{p/2} \tag{62}$$

For NMFFS with Schatten p-norm regularization, we have following:

$$\min \ \|X - XWH\|_F \ + \alpha \|W\|_{sp}^p \ \ S.t \ \ W \geq 0 \ , \ H \geq 0, W^T W = I \tag{63}$$

To solve (63), we need to consider auxiliary variable Q. Therefore, NMFFS with rank constraint is reformulated as follows [42]:

$$\min \ \|X - XWH\|_F \ + \alpha \|Q\|_{sp}^p \ \ S.t \ \ W \geq 0 \ , \ H \geq 0, W^T W = I, W - Q = 0 \tag{64}$$

Using Lagrange augmented technique, we have following:

$$L(W,H,Q,Y) = \frac{1}{2}\|X - XHW\|_F^{\ 2} + \alpha \|Q\|_{sp}^p + \frac{\beta}{4}\|W^T W - I\|_F^2 + \frac{\lambda}{2}\left\|W - Q + \frac{1}{\lambda_6}Y\right\|_F^{\ 2} - \frac{1}{2\lambda}\|Y\|_F^{\ 2} + Tr(AW^T) + Tr(BH^T) \tag{65}$$

Where $A \in R^{d\times m}$, $B \in R^{m\times d}$ and $Y \in R^{d\times m}$ are Lagrange multipliers to guarantee that $W \geq 0$, $H \geq 0$ and $W - Q = 0$, respectively.

Feature weight matrix W is affected by Q and Y in each update.

To update variable Q, let consider $\Omega = W + \frac{1}{\lambda}Y$ and then we have following problem:

$$\min_Q \alpha\|Q\|_{sp}^p + \frac{\lambda}{2}\|Q - \Omega\|_F^{\ 2} \tag{66}$$

$\|Q\|_{sp}^p$ is a non-convex relaxation term and must be converted to convex version. To this end, let have $SVD(Q) = U\Sigma V^T$, $SVD(\Omega) = P\Delta Z^T$ and $\|Q\|_{sp}^p$ =Tr (Σ^p). By applying trace inequality of John von Neumann theorem [41] we have following:

$$L(Q) = \frac{\alpha}{\lambda}\text{Tr}\ (\Sigma^p) + \frac{1}{2}\|Q - \Omega\|_F^2 = \frac{\alpha}{\lambda}\text{Tr}\ (\Sigma^p) + \frac{1}{2}(tr(\Sigma^T\Sigma) - 2Tr(\mathrm{Q}^T\Omega) + Tr(\Delta^T\Delta))$$
$$\geq \frac{\alpha}{\lambda}\text{Tr}\ (\Sigma^p) + \frac{1}{2}\big(tr(\Sigma^T\Sigma) - 2Tr(\Sigma^T\Delta) + Tr(\Delta^T\Delta)\big) = \frac{\alpha}{\lambda}\text{Tr}\ (\Sigma^p) + \frac{1}{2}\|\Sigma - \Delta\|_F^2 \tag{67}$$

The equality in problem (67) can only be held if and only if $U = P$ and $V = Z$. As a result, the optimal solution of Q is obtained by finding σ_i(singular value of matrix Q) using following:

$$\text{argmin}_{\sigma_1,\sigma_1,\dots,\sigma_d \geq 0} \sum_{i=1}^{d} \frac{1}{2}(\sigma_i - \delta_i)^2 + \frac{\alpha}{\lambda}{\sigma_i}^p \, , i = 1,2, \dots, d \tag{68}$$

Where $\Sigma = \text{diag}(\sigma_i)$ and $\Delta = \text{diag}(\delta_i)$. To solve problem (40) for σ_i we have following:

$$\text{argmin f}(\sigma_i) = \frac{1}{2}(\sigma_i - \delta_i)^2 + \frac{\alpha}{\lambda}{\sigma_i}^p \; s.t. \sigma_i \geq 0, i = 1,2, \dots, d \tag{69}$$

The second derivative f(σ_i) is $f''(\sigma_i) = p(p-1)\frac{\alpha}{\lambda}{\sigma_i}^{p-2} + 1$ and $\mu = (\frac{\lambda}{\alpha p(1-p)})^{\frac{1}{p-2}}$ is the inflection point (μ can be obtained by $f''(\sigma_i) = 0$). Obviously, the minimum value of (69) occurred when $\sigma_i = \mu$. Therefore, the optimal solution can be obtained by following:

$$\sigma_i = \begin{cases} 0, & f(\mu) \geq 0 \\ \text{argmin}_{\sigma_i \in \{0,v\}} f(\sigma_i), & f(\mu) < 0 \end{cases} \tag{70}$$

Where v is the stationary point and can be calculated by $f'(\sigma_i) = 0$.Consequently, the optimal solution of $Q = U\Sigma V^T$ by setting $\Sigma = diag(\sigma_i)$ using (70).

Additionally, in each iteration, Lagrange multiplier Y is updated by following:

$$Y = Y + \lambda(W - Q) \tag{71}$$

And coefficient λ is updated by following:

$$\lambda = \min(\rho\lambda, \lambda_{\max}) \tag{72}$$

Where ρ is the parameter to increase λ in each iteration ($\lambda_{\max}$ and ρ are coefficients).

Nuclear norm is Schatten p-norm for $p = 1$.

2.4 Minimum Redundancy-Correlation Minimization

To guarantee minimum redundancy between the features, correlation between the rows of W can be model as regularization and added to NMFFS objective function [11]. The correlation of features for a data $X \epsilon R^{n \times d}$ (n samples and d features) can be formulated as follows:

$$corr(\{f_1, \dots, f_d\}) = \frac{1}{d^2}\sum_{i,j=1}^{d} < f_i, f_j > = \frac{1}{d^2}(1_d^T X_d^T X_d 1_d) = \frac{1}{d^2}(1_d^T W^T X^T X W 1_d) \tag{73}$$

Where 1_d is a vector, whose elements are one.

To update feature weight matrix W, the partial derivative of (73) with respect to W is $\frac{\partial corr}{\partial W} = X^T X W 1_{d \times d}$.

2.5 Soft Label Learning- Pseudo Label

NMFFS is an unsupervised feature selection. The performance of NMFFS can be improved by learning the pseudo-label. With clustering approach, the labels can be modeled as the samples belong to clusters. The affinity matrix between each sample and the cluster centroids plays role as soft-label matrix. In contrast to hard-label, the soft-label is the probability of belonging the data point to cluster. Soft label learning not only considers sample point as a cluster, but also it considers the relationship with other clusters [43]. The objective function of soft-label learning is formulated as follows:

$$\min_{G,M} \sum_{i=1}^{n} \sum_{j=1}^{c} \left\| X_i - M_j \right\|_2^2 G_{ij} \;\; s.t \;\; \forall i, G_i^T 1_n = 1 \, , 0 \leq G_{ij} \leq 1 \tag{74}$$

Where G_{ij} and M_j represent the affinity degree of the i-th sample and the j-th cluster, and the cluster centroid of the jth cluster in the original feature space, respectively. Finally, soft label regression regularization is modeled as follows:

$$\min_G \|G + 1_n b^T - XWZ\|_F^2 \tag{75}$$

Where $b \epsilon \, R^c$ (c is the number of clusters) is the bias term and Z is learnable regression matrix. Biased term is $b = \frac{1}{n}(XWZ - G)^T 1_n$. Consequently, NMFFS with soft label regularization is described as follows:

$$\min_{W,H,G,Z,M_j} . \|X - XWH\|_F \; + \; \alpha Tr(G^T R) + \lambda \|(G - XWZ) C_n\|_F^2 \quad S.t \;\; W \geq 0 \, , \; H \geq 0 \, , \;\; W^T W = I, diag(G) = 0, G_i^T 1_n = 1 \, , 0 \leq G_{ij} \leq 1 \tag{76}$$

Where C_n is centering matrix $C_n = I_n - \frac{1}{n} 1_n 1_n^T$, $R_{ij} = \left\| X_i - M_j \right\|_2^2$, $diag(G) = 0$ is a sparse constraint on G and α is a balancing parameter between NMFFS and soft label learning. λ is a regularization coefficient.

2.6 Self-paced learning

Self-paced learning [44] is inspired by curriculum learning [45]. In curriculum learning, difficulty is gradually increased from easy to hard. In other words, the data is gradually added to training network from easy to hard, which leads to increased in the entropy of the training set. The main concern of curriculum learning is the metric of determining the priority of each sample. Self-paced learning aims to determine an optimal sequence of training samples to introduce during the learning process, with the goal of minimizing the impact of noise and improving overall model performance [46].

Self-paced learning objective function is described as follows:

$$\min_{W,H,\tau} \sum_{i=1}^{n} \tau_i \|x_i - x_i WH\|_F + \sum_{i=1}^{n} \frac{\gamma^2}{\tau_i + \frac{\gamma}{\mu}} \;\; S.t \;\; i = 1, \ldots, n, 0 < \tau_i < 1 W \geq 0 \, , \; H \geq 0, W^T W = I \tag{77}$$

Where τ_i represents the weight of the i-th sample x_i. $\frac{\gamma^2}{\tau_i+\frac{\gamma}{\mu}}$ is the mixture regularizer. γ controls the ''fuzzy interval'' between 0 and 1. The closer τ_i is to 1, x_i is to be the more likely selected if the closer τ_i is to 1, for the closer τ_i is to 0 vice-versa.

To update τ_i, we consider W and H fixed variables, and we have following objective function:

$$\tau_i D + \sum_{i=1}^{n} \frac{\gamma^2}{\tau_i+\frac{\gamma}{\mu}} \quad s.t \quad 0 < \tau_i < 1 \tag{78}$$

Where $D = \|x_i - x_i WH\|_F$. For $i = 1, \dots, n$, the problem (78) is decomposed into n independent sub-problems as follows:

$$\tau_i D_i + \frac{\gamma^2}{\tau_i+\frac{\gamma}{\mu}} \tag{79}$$

The closed form solution of τ_i is obtained as follows:

$$\tau_i = \begin{cases} 1, & if\ D_i \le \left(\frac{\gamma\mu}{\gamma+\mu}\right)^2 \\ 0, & if\ D_i > \mu^2 \\ \gamma\left(\frac{1}{\sqrt{D_i}} - \frac{1}{\mu}\right), & otherwise \end{cases} \tag{80}$$

Parameter μ is initialized and updated in each iteration by $\mu = \beta\mu$ ($\beta > 1$).

2.7 Contrastive Learning

Contrastive learning is a self-representation learning approach to maximize the similarity between positive pairs and minimize the similarity between negative pairs, which increase the discriminability of algorithm. Cross-entropy loss in contrastive learning is received great attention to provide maximum discrimination between positive pair and negative pair [48,49].

Graph contrastive learning is applied for feature selection as regularization function. This technique applies random corruption on nodes and edges to learn a correct node information. Then, contrastive regularizer is formulated as follows [50]:

$$\text{contrastive} = \sum_{i=1}^{n} \sum_{j \epsilon n_i} -\log \frac{\exp(s_{ij})}{\sum_{p \neq i}^{n} s_{ip}} \tag{81}$$

Where n, n_i are the total number of samples and the k-nearest neighbors of node i-th, respectively.

3 Example- NMFFS with Laplacian Graph and $\ell_{2,1}$-norm

The objective function of NMFFS with Laplacian Graph and $\ell_{2,1}$-norm is described as follows:

$$\min_{W,H} \frac{1}{2}\|X - XWH\|_F^2 + \frac{\alpha}{4}\|W^TW - I\|_F^2 + \frac{\rho}{2}Tr(W^TX^TLXW) + \frac{\gamma}{2}\|W\|_{2,1}$$

$$s.t \;\; W \geq 0\,,\; H \geq 0 \tag{82}$$

Where L is Laplacian matrix (details in section 2.2.1). ρ and γ are regularization coefficients and $\|W\|_{2,1}$ can be replaced by $Tr(W^TQW)$ such that $Q = diag\left(\frac{1}{4\|w^i+\varepsilon\|_2^2}\right)$is a diagonal matrix (details in section 2.1.1).

The augmented Lagrangian function of (82) is as follows:

$$\mathcal{L}(W,H,\beta,\lambda) = \tfrac{1}{2}\|X - XWH\|_F^2 + \tfrac{\alpha}{4}\|W^TW - I\|_F^2 + \tfrac{\rho}{2}Tr(W^TX^TLXW) + \tfrac{\gamma}{2}Tr(W^TQW)$$
$$+Tr(\beta W) + Tr(\lambda H^T) \tag{83}$$

Where β and λ are Lagrange multipliers for W and H, respectively.

To solve problem (83), the matrix H is fixed and W is updated as follows:

$$\frac{\partial \mathcal{L}}{\partial W} = \tfrac{1}{2}(-2X^TXH^T + 2X^TXWHH^T) + \alpha(WW^TW - W) + \beta + \rho X^TLXW + \gamma QW = 0 \tag{84}$$

then,

$$X^TXWHH^T + \alpha WW^TW + \beta\rho X^TLXW + \gamma QW = X^TXH^T + \alpha W \tag{85}$$

For H, the matrix W is fixed and H is updated as follows:

$$\frac{\partial \mathcal{L}}{\partial H} = \tfrac{1}{2}(-2W^TX^TX + 2W^TX^TXWH) + \lambda = 0 \tag{86}$$

$$W^TX^TXWH + \lambda = W^TX^TX \tag{87}$$

According to K.K.T conditions, $\beta_{ij}W_{ij} = 0$ and $\lambda_{ij}H_{ij} = 0$ for all $i \in [n]$ and $j \in [m]$.
Therefore, W and H are updating in each iteration as follows:

$$W_{ij} \leftarrow W_{ij}\frac{(X^TXH^T + \alpha W)_{ij}}{(X^TXWHH^T + \alpha WW^TW + \rho X^TLXW + \gamma QW)_{ij}} \tag{88}$$

$$H_{ij} \leftarrow H_{ij} \frac{(W^T X^T X)_{ij}}{(W^T X^T XWH)_{ij}} \tag{89}$$

Other regularization functions, which are discussed in this tutorial, can be applied to (82).

4 Conclusion

This tutorial paper covered the theory of different variants of regularization function for features selection using subspace learning. First, sparse regularization function was explained. Then, structure learning regularization was explained followed by rank minimization regularization. Minimum redundancy in the context correlation regularization was explained. In last, Self paced learning and contrastive learning were explained.

References:

[1] Dhal, Pradip, and Chandrashekhar Azad. "A comprehensive survey on feature selection in the various fields of machine learning." *Applied Intelligence* 52, no. 4 (2022): 4543-4581.

[2] Maćkiewicz, Andrzej, and Waldemar Ratajczak. "Principal components analysis (PCA)." *Computers & Geosciences* 19, no. 3 (1993): 303-342.

[3] Xanthopoulos, Petros, Panos M. Pardalos, Theodore B. Trafalis, Petros Xanthopoulos, Panos M. Pardalos, and Theodore B. Trafalis. "Linear discriminant analysis." *Robust data mining* (2013): 27-33.

[4] Moslemi, Amir, Aryan Safakish, Lakshmanan Sannchi, David Alberico, Schontal Halstead, and Greg Czarnota. "Predicting head and neck cancer treatment outcomes using textural feature level fusion of quantitative ultrasound spectroscopic and computed tomography: A machine learning approach." In *2023 IEEE International Ultrasonics Symposium (IUS)*, pp. 1-4. IEEE, 2023.

[5] Moslemi, Amir. "A tutorial-based survey on feature selection: Recent advancements on feature selection." *Engineering Applications of Artificial Intelligence* 126 (2023): 107136.

[6] Dokeroglu, Tansel, Ayça Deniz, and Hakan Ezgi Kiziloz. "A comprehensive survey on recent metaheuristics for feature selection." *Neurocomputing* 494 (2022): 269-296.

[7] Solorio-Fernández, Saúl, J. Ariel Carrasco-Ochoa, and José Fco Martínez-Trinidad. "A review of unsupervised feature selection methods." *Artificial Intelligence Review* 53, no. 2 (2020): 907-948.

[8] Xu, R., Li, M., Yang, Z., Yang, L., Qiao, K., Shang, Z., 2021. Dynamic feature selection algorithm based on Q-learning mechanism. Appl. Intell. 51 (10), 7233–7244.

[9] Moslemi, Amir, and Arash Ahmadian. "Subspace learning for feature selection via rank revealing QR factorization: Fast feature selection." *Expert Systems with Applications* (2024): 124919.

[10] Wang, W. Pedrycz, Zhu, Q., Zhu, W., 2015a. Subspace learning for unsupervised feature selection via matrix factorization. Pattern Recogn. 48 (1), 10–19

[11] Saberi-Movahed, Farid, Rostami, Mehrdad, Berahmand, Kamal, Karami, Saeed, Tiwari, Prayag, Oussalah, Mourad, Shahab, S., 2022a. Band. Dual regularized unsupervised feature selection based on matrix factorization and minimum redundancy with application in gene selection. Knowl. Base Syst. 256.

[12] Cai, X., Nie, F., & Huang, H. (2013). Exact Top-k Feature Selection via l2,0-Norm Constraint. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, (pp. 1240-1246). https://dl.acm.org/doi/10.5555/2540128.2540307.

[13] Shang, K. Xu, Jiao, L., 2020a. Subspace learning for unsupervised feature selection via adaptive structure learning and rank approximation. Neurocomputing 413, 72–84.

[14] Zhang, Miao, Chris Ding, Ya Zhang, and Feiping Nie. "Feature selection at the discrete limit." In Proceedings of the AAAI conference on artificial intelligence, vol. 28, no. 1. 2014.

[15] Du, Xingzhong, Yan Yan, Pingbo Pan, Guodong Long, and Lei Zhao. "Multiple graph unsupervised feature selection." *Signal Processing* 120 (2016): 754-760.

[16] Yong Shi, Jianyu Miao, Zhengyu Wang, Peng Zhang, Lingfeng Niu, Feature selection with l2,1− 2 regularization, IEEE Transact. Neural Networks Learn. Syst. 29 (10) (2018) 4967–4982.

[17] Amir Moslemi, Sparse representation learning using ℓ1− 2 compressed sensing and rank-revealing QR factorization, Eng. Appl. Artif. Intell. 125 (2023), 106663.

[18] Nie F, Dong X, Tian L, Wang R, Li X (2020) Unsupervised featureselection with constrained ℓ2,0-norm and optimized graph. IEEETrans Neural Netw Learn Syst 33(4):1702–1713

[19] Zhu P, Hou X, Tang K, Liu Y, Zhao Y-P, Wang Z (2023) Unsu-pervised feature selection through combining graph learning andℓ2,0-norm constraint. Inf Sci 622:68–82

[20] Nie, Feiping, Xia Dong, Lai Tian, Rong Wang, and Xuelong Li. "Unsupervised Feature Selection With Constrained $\ell_{2,0}$ -Norm and Optimized Graph." *IEEE transactions on neural networks and learning systems* 33, no. 4 (2020): 1702-1713.

[21] Zhu, Peican, Xin Hou, Keke Tang, Yang Liu, Yin-Ping Zhao, and Zhen Wang. "Unsupervised feature selection through combining graph learning and ℓ2, 0-norm constraint." *Information Sciences* 622 (2023): 68-82.

[22] Moslemi, Amir, and Afshin Shaygani. "Subspace learning via Hessian regularized latent representation learning with l 2, 0-norm constraint: unsupervised feature selection." *International Journal of Machine Learning and Cybernetics* (2024): 1-20.

[23] J.Q. Han, Z.G. Sun, H.W. Hao, Selecting feature subset with sparsity and low redundancy for unsupervised learning, Knowl. Based Syst. 86 (2015) 210–223.

[24] Liu, X., Wang, L., Zhang, J., Yin, J., & Liu, H. (2013). Global and local structure preservation for feature selection. *IEEE transactions on neural networks and learning systems*, *25*(6), 1083-1095.

[25] Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, *15*(6), 1373-1396

[26] Ren, Weiya, Guohui Li, Dan Tu, and Li Jia. "Nonnegative matrix factorization with regularizations." *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 4, no. 1 (2014): 153-164.

[27] K.I. Kim, F. Steinke, M. Hein, Semi-supervised Regression using Hessian Energy with an Application to Semi-supervised Dimensionality Reduction, in: Advances in Neural Information Processing Systems (NIPS). MPI for Biological Cybernetics, Germany, 2010: pp. 979–987.

[28] J. M. Lee. Riemannian Manifolds - An introduction to curvature. Springer, New York, 1997.

[29] F. Nie, W. Zhu, and X. Li, “Unsupervised feature selection with structured graph optimization,” in Proc. 30th AAAI Conf. Artif. Intell., 2016, pp. 1302–1308.
[30] Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 977–986

[31] Mohar, B.; Alavi, Y.; Chartrand, G.; and Oellermann, O. 1991. The laplacian spectrum of graphs. Graph theory, combinatorics, and applications 2:871–898.
[32] Fan, K. 1949. On a theorem of weyl concerning eigenvalues of linear transformations i. Proceedings of the National Academy of Sciences of the United States of America 35(11):652.
[33] Jiao, Cui-Na, Ying-Lian Gao, Na Yu, Jin-Xing Liu, and Lian-Yong Qi. "Hyper-graph regularized constrained NMF for selecting differentially expressed genes and tumor classification." *IEEE journal of biomedical and health informatics* 24, no. 10 (2020): 3002-3011.
[34] Zhu, Xiaofeng, Shichao Zhang, Yonghua Zhu, Pengfei Zhu, and Yue Gao. "Unsupervised spectral feature selection with dynamic hyper-graph learning." IEEE Transactions on Knowledge and Data Engineering 34, no. 6 (2020): 3016-3028.
[35] Z. Zhang, L. Bai, Y. Liang, and E. Hancock, “Joint hypergraph learning and sparse regression for feature selection,” Pattern Recognit., vol. 63, pp. 291–309, 2016.
[36] D. Zhou, J. Huang, and B. Scholkopf, “Learning with hyper- € graphs: Clustering, classification, and embedding,” in Proc. 19th Int. Conf. Neural Inf. Process. Syst., 2006, vol. 19, pp. 1633–1640.
[37] Zhu, Xiaofeng, Shichao Zhang, Yonghua Zhu, Pengfei Zhu, and Yue Gao. "Unsupervised spectral feature selection with dynamic hyper-graph learning." *IEEE Transactions on Knowledge and Data Engineering* 34, no. 6 (2020): 3016-3028.
[38] Recht, Benjamin, Maryam Fazel, and Pablo A. Parrilo. "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization." *SIAM review* 52, no. 3 (2010): 471-501.
[39] Liu, L., Huang, W., & Chen, D. R. (2014). Exact minimum rank approximation via Schatten p-norm minimization. Journal of Computational and Applied Mathematics, 267, 218–227.

[40] Wang, Weigang, et al. "Low-rank sparse feature selection for image classification." *Expert Systems with Applications* 189 (2022): 115685.

[41] [20] Mirsky, L. (1975). A trace inequality of John von Neumann. Monatshefte für mathematik, 79(4), 303–306. https://doi.org/10.1007/BF01647331
[42] Moslemi, Amir, and Arash Ahmadian. "Dual regularized subspace learning using adaptive graph learning and rank constraint: Unsupervised feature selection on gene expression microarray datasets." *Computers in Biology and Medicine* 167 (2023): 107659.
[42] Li, Xuelong, Han Zhang, Rui Zhang, Yun Liu, and Feiping Nie. "Generalized uncorrelated regression with adaptive graph for unsupervised feature selection." *IEEE transactions on neural networks and learning systems* 30, no. 5 (2018): 1587-1595.
[43] Zhou, Shixuan, Peng Song, Zihao Song, and Liang Ji. "Soft-label guided non-negative matrix factorization for unsupervised feature selection." *Expert Systems with Applications* 216 (2023): 119468.
[44] M. Kumar, B. Packer, D. Koller, Self-Paced Learning for Latent Variable Models, 2010, pp. 1189–1197.
[45] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, 2009.
[46] Li, Weiyi, Hongmei Chen, Tianrui Li, Jihong Wan, and Binbin Sang. "Unsupervised feature selection via self-paced learning and low-redundant regularization." *Knowledge-Based Systems* 240 (2022): 108150.
[47] Zhen, Xiantong, Mengyang Yu, Xiaofei He, and Shuo Li. "Multi-target regression via robust low-rank learning." *IEEE transactions on pattern analysis and machine intelligence* 40, no. 2 (2017): 497-504.
[48] Li, Yuecheng, Jialong Chen, Chuan Chen, Lei Yang, and Zibin Zheng. "Contrastive deep nonnegative matrix factorization for community detection." In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6725-6729. IEEE, 2024.
[49] Zhou, Qian, Qianqian Wang, Quanxue Gao, Ming Yang, and Xinbo Gao. "Unsupervised Discriminative Feature Selection via Contrastive Graph Learning." *IEEE Transactions on Image Processing* (2024).
[50] Pan, Erlin, and Zhao Kang. "Multi-view contrastive graph clustering." *Advances in neural information processing systems* 34 (2021): 2148-2159.