mathematics

MDPI

Article

Multi-Target Feature Selection with Adaptive Graph Learning and Target Correlations

Yujing Zhou † and Dubo He *,†

Department of Management Engineering and Equipment Economics, Naval University of Engineering, Wuhan 430033, China; 1920191045@nue.edu.cn
* Correspondence: 21000801@nue.edu.cn
† These authors contributed equally to this work.

Abstract: In this paper, we present a novel multi-target feature selection algorithm that incorporates adaptive graph learning and target correlations. Specifically, our proposed approach introduces the low-rank constraint on the regression matrix, allowing us to model both inter-target and input–output relationships within a unified framework. To preserve the similarity structure of the samples and mitigate the influence of noise and outliers, we learn a graph matrix that captures the induced sample similarity. Furthermore, we introduce a manifold regularizer to maintain the global target correlations, ensuring the preservation of the overall target relationship during subsequent learning processes. To solve the final objective function, we also propose an optimization algorithm. Through extensive experiments on eight real-world datasets, we demonstrate that our proposed method outperforms state-of-the-art multi-target feature selection techniques.

Keywords: feature selection; multi-target regression; graph learning

MSC: 68T09

Citation: Zhou, Y.; He, D. Multi-Target Feature Selection with Adaptive Graph Learning and Target Correlations. *Mathematics* **2024**, *12*, 372. https://doi.org/10.3390/math12030372

Academic Editors: Florin Leon, Mircea Hulea and Marius Gavrilescu

Received: 27 December 2023
Revised: 17 January 2024
Accepted: 19 January 2024
Published: 24 January 2024

1. Introduction

Multi-target regression (MTR) aims to predict multiple target (response) variables by a common set of features. Unlike the Multi-Label Classification (MLC), where the multivariate outputs are all binary variables, the multi-outputs in MTR are all real-valued variables. Recently, MTR is enjoying increasing popularity in machine-learning community because of its ability to predict multiple outputs simultaneously and better generalization performance. Moreover, due to its superior ability, MTR has been widely employed in solving challenging problems in numerous applications such as data mining [1–4], computer vision [5], medical diagnosis [6], stock price prediction [7], load forecasting [8]. MTR takes into account the relationship between features and targets and the underlying correlation among targets, ensuring a better representation and interpretability of real-world problems. Another advantage of MTR is that it can generate cleaner models with better computational efficiency.

In order to obtain desirable and reliable predictions for multiple target variables, many potentially relevant variables are typically involved in the formulation of high-dimensional data which would represent and explain the target variables. However, high dimensional input features not only induce a complex correlation structure between features and targets but also result in the problem of the "curse of dimensionality". In addition, unrelated and redundant features adversely affect the effectiveness of the modeling and reduce the generalization performance. As an efficient dimensionality reduction technique to choose a subset of features from the primitive high-dimensional data, feature selection contributes to prevent the "curse of dimensionality" and enables the selection of an optimal subset from the primitive feature space with specific criterion. As feature selection does not modify the

primitive semantics of the original variables, it makes the model more interpretable with reduced training time and space requirements [9].

The Multi-Target Feature Selection (MTFS) methods generally fall into one of three categories [10]: filter [11,12], wrapper [13,14] and embedded approaches [15,16]. The filter approaches use specific evaluation metrics such as mutual information [11] and Laplacian score [12] to measure the importance of features and select the most relevant features to form a subset. The family of filter methods is independent of the algorithm, which makes them computationally efficient. They can effectively remove irrelevant features from a dataset. However, one limitation of filter methods is that they may include redundant features in the selected subset since they ignore the correlation between features. On the other hand, wrapper methods select a subset of features by inputting them into a specific model for training. This process continues until satisfactory performance is achieved. Wrapper methods take into account the correlation between features and consider their impact on the model performance. Wrapper methods can be computationally expensive since the performance of the selected subset needs to be verified after each feature selection. To balance the trade-off between filter and wrapper methods, embedded methods treat feature selection as an optimization problem. Embedded methods can select the most informative features with a relatively low computational cost compared to wrapper methods. By embedding feature selection within the model building process, embedded methods are able to take into account the correlation between features while also minimizing computational costs. These methods weigh the importance of each feature and select the most relevant ones by optimizing the model performance. As a result, embedded methods often lead to better model performance compared to filter methods, while still being more computationally efficient than wrapper methods. Therefore, embedded methods are increasingly drawing attention due to their superior performance.

Closely related to MTR, multi-label learning is generally viewed as a particular case of MTR in statistics analysis [17]. Inspired by the intimate relationship between multi-label classification (MLC) and MTR, Various MTR models have been proposed based on the thought of handling label relevance in the context of MLC, such as the ensemble of regressor chains (ERC), stacked single-target (SST), Random Linear Target Combinations (RLC) [18,19]. Spyromitros-Xioufis et al. discrete the output space by product quantization and thus convert the MTR problem into a MLC problem [11]. It is evident that there are favorable similarities between MLC and MTR, and various methods of MLC have been transferred to handle MTR problems with excellent performance. However, there are a few approaches to solving the feature selection problem in MTR by exploiting various feature selection strategies in MLC. Indeed, various supervised, semi-supervised and unsupervised feature selection methods in MLC can also be transferred to feature selection tasks in MTR scenarios, such as incorporating local and global correlation structures of labels, features or samples into the learning process to improve the feature selection performance, which is inspiring for MTFS [20–22].

The significant challenges of MTR arise from jointly addressing input–output and inter-target correlations [23]. By exploring the correlation information between the targets accurately and effectively, the MTR model can obtain improved performance compared to the single-target model. Therefore, most existing MTR models focus on exploring target correlations. The general technique imposes various sparse regularizer or low-rank constraints on the regression matrix [6,23,24]. However, the above methods do not consider the structure information of features or samples. Both the global and local structures of features and samples have been previously demonstrated in the literature to provide complementary information for reinforcing the performance of feature selection [20,22,25]. Specifically, preserving the geometric structure of samples can strengthen the feature selection performance since the effects of noises and outliers could be mitigated [21,22]. Moreover, in MTR scenarios, the intrinsic inter-target relationships can also provide discriminate information to feature selection and discover the essential features that are highly correlated to the

relationships between targets. Incorrect inter-target relationships could also deteriorate the generalization capability of feature selection model.

To address the above-mentioned issues, we design a novel MTFS method by integrating an adaptive graph structure learning and manifold learning of global target correlations into a general multi-target sparse regression model. The key contributions of this paper are highlighted below:

- A novel MTFS method with low-rank constraint is designed to generate low redundancy yet informative feature subset for MTR by imposing a low-rank constraint on the regression matrix, to conduct subspace learning and thus decouple the inter-input as well as the inter-target relationships, which can reduce the influence of redundant or irrelevant features.
- Based on the nearest neighbors of the samples, the similarity-induced graph matrix is learned adaptively, and the local geometric structure of the data can be preserved during the feature selection process, thus mitigating the effects of noise and outliers.
- A manifold regularizer based on target correlation is designed by considering the statistical correlation information between multiple targets over the training set, which is beneficial to discover informative features that are associated with inter-target relationships.
- The alternative optimization algorithm is proposed to solve the proposed objective function, and the convergence of the algorithm is proved theoretically. Extensive experiments are conducted on a benchmark data sets to validate the feasibility and effectiveness of the proposed method.

The rest of this paper is organized as follows. In Section 2, some related works on multi-objective feature selection and multi-label classification feature selection methods are briefly reviewed. The proposed multi-objective feature selection method is described in detail in Section 3, followed by the proposed optimization algorithm in Section 4. Section 5 proves the convergence of the proposed algorithm and analyzes the corresponding time complexity. In Section 6, experimental results are reported and analyzed to demonstrate the effectiveness of the proposed method. Finally, a brief conclusion is summarized in Section 7.

2. Related Work

To date, different MTFS methods have been proposed. Hashemi et al. [26] proposed a feature selection method incorporating the VIKOR algorithm to rank the features in the MTR problem. Sechidis et al. [11] proposed a feature selection method for both MLC and MTR. The method considers correlation, redundancy and complementarity between features by calculating the interaction among targets, thus ensuring that the acquired subset of features can have less redundancy and higher correlation. Petkovic et al. [27] proposed a feature-ranking method based on predictive clustering tree integration and RReliefF method extensions, and the optimal feature ranking is determined by integrating the feature scores of these two groups of methods. Masmoudi et al. [28] presented a multi-target feature ranking method based on regression chain ensemble and random forest; the final feature ranking is obtained by combining the feature importance information from both methods.

Recently, different embedded approaches have also been proposed. Yuan et al. [29] proposed an embedded Sparse Structural Feature Selection (SSFS) model based on a multi-layer multi-output framework. This model achieves improved feature selection performance by simultaneously applying sparsity constraints on the objective function, regression coefficients, and structure matrix. Similarly, Zhu et al. [30] utilized low-rank constraint to identify correlations between output variables and impose $\ell_{2,1}$-norm regularization on regression matrix to achieve feature selection. The above-mentioned methods impose sparsity or low rank on the loss function or parameter matrix to achieve the feature selection. However, these embedded methods either consider the similarity structure of samples or the statistical correlations between different targets, which may constrain the performance of feature selection.

In fact, the feature selection method in MLC tasks can also be deployed in MTR tasks when the model can handle continuous output variables. Fan et al. [31] proposed a feature selection method based on both label correlations and feature redundancies; the label correlations are explored through low-dimensional embedding, which maintains the global and local structure of the original label space. Xu et al. [32] proposed to perform feature extraction by maximizing feature variance and feature–label dependence to achieve better performance in MLC problems. Zhu et al. [21] proposed a robust unsupervised spectral feature selection method that maintains the local structure of features by exploiting the self-representation of features and maintains the global structure of samples as features via imposing low-rank constraints on the weight matrix. Mahsa et al. [33] proposed a low-redundant unsupervised feature selection method based on data structure learning and feature orthogonalization. Obviously, the above method introduces other information such as the local and global structure of the labels, the structure of the data and the relationship between the features by considering not only the relationship between the features and the labels in the feature selection process.

Recently, graph-based methods, such as spectral clustering, graph learning and hypergraph learning, have played an important role in machine learning due to their ability to encode similarity relationships among data. Ma et al. [34] proposed a feature selection method named discriminative multi-label feature selection with adaptive graph diffusion, and the graph embedding learning framework is constructed with adaptive graph diffusion to uncover a latent subspace that preserves the higher-order structure information. Zhang et al. [35] proposed a novel unsupervised feature selection via adaptive graph learning and constraint. Zhu et al. [36] proposed an unsupervised spectral feature selection method with dynamic hypergraph Learning. You et al. [37] proposed an unsupervised feature selection method via Neural Networks (NN) and self-expression with adaptive graph constraint. Deepak et al. [38] extended the feature selection algorithm presented in via Gumbel softmax to Graph Neural Networks (GNN). It can be seen that graph learning can effectively mine the similarity or structural relationship between data, and thus improve the performance of feature selection.

From the above research, it is evident that maintaining the various structural information contained in the original data, such as the geometric or similar structure of the samples, the structural information among the features and different outputs, can provide supplementary information for feature selection in different perspectives, thereby improve the feature selection performance. However, existing MTFS methods rarely consider the above information simultaneously.

3. The Proposed Approaches

3.1. Notations

For a $n \times m$ matrix $\mathbf{A} = [a_{i,j}] \in \mathbb{R}^{n \times m}$, and $a_{i,j}$ denotes the (i,j)-th entry of $\mathbf{A}$. $\mathbf{A}^T$ denotes its transpose. $tr(\mathbf{A})$ is $\mathbf{A}$'s trace. The Frobenius norm of $\mathbf{A}$ is defined as $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{i,j}^2}$, and the $\ell_{p,q}$-norm of matrix $\mathbf{A}$ is defined as

$$\|\mathbf{A}\|_{p,q} = \left[\sum_{i=1}^{n} \left(\sum_{j=1}^{m} |a_{i,j}|^p \right)^{\frac{q}{p}} \right]^{\frac{1}{q}} \tag{1}$$

and hence the $\ell_{2,1}$-norm of $\mathbf{A}$ is defined as

$$\|\mathbf{A}\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} a_{i,j}^2} \tag{2}$$

For a n-dimensional vector $\mathbf{c} \in \mathbb{R}^n$, $\|\mathbf{c}\|_2 = \sqrt{\sum_{i=1}^{n} c_i^2}$ is its ℓ_2-norm, $\mathbf{I}$ denotes an identity matrix, and let $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ denote the center matrix, where $\mathbf{1}_n \in \mathbb{R}^n$ and the value of each element is 1.

3.2. MTR Based on Low-Rank Constraint

Given a training set consisting of n instances $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$ represents feature or input matrix, where $\mathbf{x}_i = [x_{i,1}, \ldots, x_{i,q}]^T \in \mathbb{R}^d$, and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^T \in \mathbb{R}^{n \times q}$ represents target or output matrix, where $\mathbf{y}_i = [y_{i,1}, \ldots, y_{i,q}]^T \in \mathbb{R}^q$ is the multi-target output corresponding to $\mathbf{x}_i$. The traditional ridge regression can be extended to multi-dimension, and we reach the following objective function:

$$\min_{\mathbf{W},\mathbf{b}} \|\mathbf{X}\mathbf{W} + \mathbf{1}_n\mathbf{b}^T - \mathbf{Y}\|_F^2 + \alpha\|\mathbf{W}\|_F^2 \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{d \times q}$ is the regression coefficients, $\mathbf{b} \in \mathbb{R}^q$ is the bias, and $\alpha > 0$ is the regularization parameter. d and q are dimensions of features and targets. To select the features, the $\ell_{2,1}$-norm regularizer is imposed on regression matrix $\mathbf{W}$, and we have

$$\min_{\mathbf{W},\mathbf{b}} \|\mathbf{X}\mathbf{W} + \mathbf{1}_n\mathbf{b}^T - \mathbf{Y}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} \tag{4}$$

where the sparse learning of $\mathbf{W}$ based on $\ell_{2,1}$-norm encourages the row sparsity to unselect the irrelevant features in the original feature matrix $\mathbf{X}$. Evidently, Equation (4) does not take into account the correlation among targets, which leads to poor performance in MTFS. Therefore, we impose a low-rank constraint on $\mathbf{W}$, i.e., $\mathbf{W} = \mathbf{A}\mathbf{B}$, where $\mathbf{A} \in \mathbb{R}^{d \times r}$, $\mathbf{B} \in \mathbb{R}^{r \times q}$, $r \leq min(d, q)$. Hence, Equation (4) is modified to

$$\min_{\mathbf{A},\mathbf{B},\mathbf{b}} \|\mathbf{X}\mathbf{A}\mathbf{B} + \mathbf{1}_n\mathbf{b}^T - \mathbf{Y}\|_F^2 + \alpha\|\mathbf{A}\mathbf{B}\|_{2,1} \tag{5}$$

In Equation (5), the parameter matrix $\mathbf{A}$ can be viewed as transforming the original feature space $\mathbb{R}^d$ into an latent variable space $\mathbb{R}^r$ geometrically, and then parameter matrix $\mathbf{B}$ transforms $\mathbf{X}\mathbf{A}$ to the target space $\mathbb{R}^q$. Considering the correlation among q targets, $\mathbf{B}$ can be served to encode inter-target correlations explicitly. Thus, the low-rank constraint takes into account global target correlations to leverage subspace learning and enables the simultaneous modeling of input–output correlations as well as inter-target relationships. In addition, the effects of redundant features and anomalous variables can be mitigated by low-rank learning, resulting in the output of robust feature selection models [39,40].

3.3. Adaptive Graph-Learning Based on Local Sample Structure

So far, the majority of studies have shown that, in addition to characterizing the significance of features in the regression model through sparse learning, the local structural information of the sample can also contribute additional information to feature selection [20–22,25]. By preserving the nearest neighbour structure of instances, the distribution of samples in the learned low-dimensional space can maintain consistency with the original sample space [21,22]. Even for a MTR problem with a complex correlation structure, The output $\mathbf{Y}$ can be reasonably hypothesized to be a continuous and smooth function of the input $\mathbf{X}$. It is natural to expect close samples $\mathbf{x}_i$ and $\mathbf{x}_j$ to have close output values $\mathbf{y}_i$ and $\mathbf{y}_j$; thereby, the corresponding prediction outputs $\hat{\mathbf{y}}_i$ and $\hat{\mathbf{y}}_j$ should also be adjacent to each other [41]. Based on the hypothesis, the geometric structure information of different instances in the feature space is leveraged to ensure that the predicted output of the model also maintains a similar geometric structure.

The existing literature obtains the local distribution structure and information of samples by learning the graph matrix $\mathbf{S}$ between samples, and given the input matrix $\mathbf{X}$ and the corresponding weight coefficients $\mathbf{W}$, according to the literature [42], we have:

$$\min_{\mathbf{W}} \sum_{i,j=1}^{n} \|\mathbf{x}_i^T\mathbf{W} - \mathbf{x}_j^T\mathbf{W}\|_2^2 s_{i,j} \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{d \times q}$ and $\mathbf{S} = [s_{i,j}] \in \mathbb{R}^{n \times n}$, and $s_{i,j}$ represents the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Traditional methods are often based on heat kernel functions to calculate the similarity

between nearest neighbors samples, the similarity between nearest neighbor samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as

$$s_{i,j} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right) \tag{7}$$

otherwise $s_{i,j} = 0$. Although Equation (7) has been widely applied, the similarity matrix is highly sensitive to the existence of noise and outliers in the original data [21,22]. To deal with this, we learn the similarity matrix of the target space adaptively to mitigate the effect of noise and outliers. The hypothesis in manifold learning is that if two samples are close in the dimension reduction space, then their corresponding multivariate prediction outputs should also be closed in target space, which gives rise to

$$\begin{aligned} \min_{\mathbf{S},\mathbf{A},\mathbf{B}} & \sum_{i,j=1}^{n} \left(\|\mathbf{x}_i^T\mathbf{W} - \mathbf{x}_j^T\mathbf{W}\|_2^2 s_{i,j} + \gamma\|\mathbf{s}_i\|_2^2\right) \\ & s.t.\ \forall i, \mathbf{1}^T\mathbf{s}_i = 1, s_{i,i} = 0, \\ & s_{i,j} \geq 0 \text{ if } j \in \mathcal{N}(i), \text{otherwise } 0. \end{aligned} \tag{8}$$

where γ is a tuning parameter, The second item in (8) deals with avoiding trivial solutions. $\mathcal{N}(i)$ represents the nearest neighbours set of the ith sample, and $\mathbf{1}^T\mathbf{s}_i = 1$ has been proved to reinforce the robustness for noises and outliers in [43], where $\mathbf{s}_i$ is the ith column of matrix $\mathbf{S}$. Combining the low-rank constraint and Equation (8), which leads to

$$\begin{aligned} \min_{\mathbf{S},\mathbf{A},\mathbf{B}} & \sum_{i,j=1}^{n} \left(\|\mathbf{x}_i^T\mathbf{A}\mathbf{B} - \mathbf{x}_j^T\mathbf{A}\mathbf{B}\|_2^2 s_{i,j} + \gamma\|\mathbf{s}_i\|_2^2\right) \\ & s.t.\ \forall i, \mathbf{1}^T\mathbf{s}_i = 1, s_{i,i} = 0, \\ & s_{i,j} \geq 0 \text{ if } j \in \mathcal{N}(i), \text{otherwise } 0. \end{aligned} \tag{9}$$

Based on Equation (9), we can ensure that the nearest neighbour relationship in the predicted output is consistent with the original data, which benefits the subsequent learning of different output correlation structures. Moreover, preserving the nearest neighbour relationship between samples is beneficial to lessen the impact of redundant or irrelevant features to improve the performance of feature selection.

3.4. Manifold Regularization of Global Target Correlations

Since different target correlation structures can also affect the performance of MTFS, we propose a manifold regularization term for global target correlations, which automatically exacts the correlations from the target matrix. By incorporating the target manifold regularization via exploiting the correlation of the target variables to filter out the noises of target variables indirectly. First, we use the commonly used cosine similarity to measure the similarity between target variables, which is calculated as follows,

$$\tilde{s}_{i,j} = \frac{\langle \mathbf{y}_{:,i}, \mathbf{y}_{:,i} \rangle}{\|\mathbf{y}_{:,i}\|\|\mathbf{y}_{:,j}\|}, i, j = 1, \ldots, q \tag{10}$$

where $\mathbf{y}_{:,i}$ and $\mathbf{y}_{:,i}$ are the ith and jth column of $\mathbf{Y}$, respectively. We assume that for the coefficient matrix $\mathbf{B} \in \mathbb{R}^{r\times q}$, if the target output vectors $\mathbf{y}_{:,i}$ and $\mathbf{y}_{:,j}$ are similar to each other, their corresponding weight vectors $\mathbf{b}_i$ and $\mathbf{b}_j$ should also be close. Based on the assumptions, we have:

$$\min_{\mathbf{B}} \sum_{i,j=1}^{q} \|\mathbf{b}_i - \mathbf{b}_j\|_2^2 \tilde{s}_{i,j} \tag{11}$$

where $\mathbf{b}_i$ and $\mathbf{b}_j$ are the ith and jth column of $\mathbf{B}$. Equation (11) encourages the similarity of the weight vectors corresponding to similar target outputs. The advantage of Equation (11) is that it can use the similarity information among different target outputs, thus improving the feature selection performance in MTR problems.

3.5. Objective Function

By incorporating the model (9) and (11) into the generalized low-rank MTR model (5), we can obtain the final feature selection model based on adaptive graph learning and global target correlations for MTR, which is described as follows:

$$\begin{aligned}
&\min_{\mathbf{A},\mathbf{B},\mathbf{S},\mathbf{b}} \|\mathbf{XAB} + \mathbf{1}_n\mathbf{b}^T - \mathbf{Y}\|_F^2 + \alpha\|\mathbf{AB}\|_{2,1} \\
&+ \beta \sum_{i,j=1}^{n} \left(\|\mathbf{x}_i^T\mathbf{AB} - \mathbf{x}_j^T\mathbf{AB}\|_2^2 s_{i,j} + \gamma \sum_{i=1}^{n} \|\mathbf{s}_i\|_2^2 \right) \\
&+ \lambda \sum_{i,j=1}^{q} \|\mathbf{b}_i - \mathbf{b}_j\|_2^2 \tilde{s}_{i,j} \\
&s.t. \begin{cases} \forall i, \mathbf{1}^T\mathbf{s}_i = 1, s_{i,i} = 0, \\ s_{i,j} \geq 0 \text{ if } j \in \mathcal{N}(i), \text{otherwise } 0. \end{cases}
\end{aligned} \tag{12}$$

where α, β, γ and λ are tuning parameter. The proposed objective function (12) has the following important characteristics. On the one hand, the low-rank constraint on the regression matrix can decouple the input–target and inter-target correlations and enables robust learning of the correlation. On the other hand, by integrating the adaptive graph learning based on local sample structure and manifold regularization of global target correlations, we can consider both local sample structure and global target correlations. Moreover, the graph structure and regression parameter matrices learning could be iteratively updated by each other, and the global target correlations can be extracted from data automatically.

Consequently, given the optimal parameter matrix $\mathbf{A}$ and $\mathbf{B}$, we evaluate the importance of each feature based on the ℓ_2-norm of $(\mathbf{AB})_i$, and rank them in descending order, then the top-ranked subset of features can be obtained.

4. Optimization Algorithm

This section presents an alternating optimization algorithm to solve the problem (12), i.e., iteratively optimizing each variable while fixing the others until convergence.

First, by setting the derivative of Equation (12) *w.r.t.* $\mathbf{b}$ to zero, we have

$$\mathbf{b}^T = \frac{1}{n}\left(\mathbf{1}_n^T\mathbf{Y} - \mathbf{1}_n^T\mathbf{XAB}\right) \tag{13}$$

Substituting the result of Equation (13) into (12), and the objective function can be rewritten as

$$\begin{aligned}
&\min_{\mathbf{A},\mathbf{B},\mathbf{S}} \|\mathbf{H}(\mathbf{XAB} - \mathbf{Y})\|_F^2 + \alpha\|\mathbf{AB}\|_{2,1} \\
&+ \beta \sum_{i,j=1}^{n} \left(\|\mathbf{x}_i^T\mathbf{AB} - \mathbf{x}_j^T\mathbf{AB}\|_2^2 s_{i,j} + \gamma \sum_{i=1}^{n} \|\mathbf{s}_i\|_2^2 \right) \\
&+ \lambda \sum_{i,j=1}^{q} \|\mathbf{b}_i - \mathbf{b}_j\|_2^2 \tilde{s}_{i,j} \\
&s.t. \begin{cases} \forall i, \mathbf{1}^T\mathbf{s}_i = 1, s_{i,i} = 0, \\ s_{i,j} \geq 0 \text{ if } j \in \mathcal{N}(i), \text{otherwise } 0. \end{cases}
\end{aligned} \tag{14}$$

where $\mathbf{H}$ is a symmetric center matrix. Since Equation (14) is convex for each parameter matrix while fixing others. Hence, the alternating optimization algorithm is introduced.

4.1. Fix $\mathbf{S}$ *Update* $\mathbf{A}$ *and* $\mathbf{B}$

With $\mathbf{S}$ is fixed, problem (14) can be rewritten as follows:

$$\begin{aligned}
&\min_{\mathbf{A},\mathbf{B}} \|\mathbf{H}(\mathbf{XAB} - \mathbf{Y})\|_F^2 + \alpha\|\mathbf{AB}\|_{2,1} \\
&+ \beta \sum_{i,j=1}^{n} \|\mathbf{x}_i^T\mathbf{AB} - \mathbf{x}_j^T\mathbf{AB}\|_2^2 s_{i,j} + \lambda \sum_{i,j=1}^{q} \|\mathbf{b}_i - \mathbf{b}_j\|_2^2 \tilde{s}_{i,j}
\end{aligned} \tag{15}$$

To prevent the non-differentiable problem in (15), we transform problem (15) as follows,

$$\begin{aligned}\min_{\mathbf{A},\mathbf{B}} \|\mathbf{H}(\mathbf{XAB}-\mathbf{Y})\|_F^2 + \alpha tr\left(\mathbf{B}^T\mathbf{A}^T\mathbf{DAB}\right)\\ + \beta tr\left(\mathbf{B}^T\mathbf{A}^T\mathbf{X}^T\mathbf{LXAB}\right) + \lambda tr(\mathbf{B}\widetilde{\mathbf{L}}\mathbf{B}^T)\end{aligned} \tag{16}$$

where $\mathbf{L}$ and $\widetilde{\mathbf{L}}$ are the Laplacian matrices corresponding to $s_{i,j}$ and $\tilde{s}_{i,j}$, respectively. $\mathbf{D} \in \mathbb{R}^{d\times d}$ is the diagonal matrix and

$$D_{i,i} = \frac{1}{2\|(\mathbf{AB})_i\|_2^2}, i = 1, 2, \ldots, d \tag{17}$$

where $(\mathbf{AB})_i$ is the ith row of matrix $\mathbf{AB}$. Similarly, by fixing $\mathbf{B}$, we set the derivative of Equation (16) with respect to $\mathbf{A}$ to zero and further to obtain

$$\mathbf{A}^* = \mathbf{P}^{-1}\mathbf{X}^T\mathbf{HYB}^T\left(\mathbf{BB}^T\right)^{-1} \tag{18}$$

where $\mathbf{P} = \mathbf{X}^T\mathbf{HX} + \alpha\mathbf{D} + \beta\mathbf{X}^T\mathbf{LX}$. In the same way, by fixing $\mathbf{A}$ we can obtain the following expression,

$$\min_{\mathbf{B}} tr\left(\mathbf{B}^T\mathbf{A}^T\mathbf{PAB} - 2\mathbf{B}^T\mathbf{A}^T\mathbf{X}^T\mathbf{HY}\right) + \lambda tr\left(\mathbf{B}\widetilde{\mathbf{L}}\mathbf{B}^T\right) \tag{19}$$

We set the derivative of Equation (19) *w.r.t.* $\mathbf{B}$ to zero and obtain

$$\mathbf{A}^T\mathbf{PAB} + \lambda\mathbf{B}\widetilde{\mathbf{L}} = \mathbf{A}^T\mathbf{X}^T\mathbf{HY} \tag{20}$$

Obviously, Equation (20) is a standard Sylvester equation $\mathcal{A}\Theta + \Theta\mathcal{B} = \mathcal{C}$, where Θ is the unknown corresponding to $\mathbf{B}$, $\mathcal{A} = \mathbf{A}^T\mathbf{PA}$, $\mathcal{B} = \lambda\widetilde{\mathbf{L}}$, and $\mathcal{C} = \mathbf{A}^T\mathbf{X}^T\mathbf{HY}$. Therefore, Equation (20) has a closed-form solution and can be solved analytically. The optimization of $\mathbf{A}$ and $\mathbf{B}$ is shown in Algorithm 1.

Algorithm 1 The procedure of optimizing $\mathbf{A}$ and $\mathbf{B}$

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{Y} \in \mathbb{R}^{n\times q}$, $\mathbf{L} \in \mathbb{R}^{n\times n}$, $\widetilde{\mathbf{L}} \in \mathbb{R}^{q\times q}$, α, β, λ, k and r;
Output: $\mathbf{A} \in \mathbb{R}^{d\times r}$, $\mathbf{B} \in \mathbb{R}^{r\times q}$;
1. Initialize $\mathbf{D} = \mathbf{I} \in \mathbb{R}^{d\times d}$;
2. Update the matrix $\mathbf{P}$;
3. **repeat**:
   3.1. Calculate $\mathbf{B}$ by Equation (18);
   3.2. Update $\mathbf{A}$ by Equation (20);
   3.3. Update $\mathbf{D}$ and $\mathbf{P}$ by Equation (17);
until *converge*;

4.2. *Fix* $\mathbf{A}$ *and* $\mathbf{B}$ *Update* $\mathbf{S}$

With fixed $\mathbf{A}$ and $\mathbf{B}$ we have:

$$\begin{aligned}\min_{\mathbf{S}} \sum_{i,j=1}^{n} \left(\|\mathbf{x}_i^T\mathbf{AB} - \mathbf{x}_j^T\mathbf{AB}\|_2^2 s_{i,j} + \gamma\|\mathbf{s}_i\|_2^2\right)\\ s.t.\ \forall i, \mathbf{1}^T\mathbf{s}_i = 1, s_{i,i} = 0,\\ s_{i,j} \geq 0 \text{ if } j \in \mathcal{N}(i), \text{otherwise } 0.\end{aligned} \tag{21}$$

Initially, we set the value of $s_{i,j} = 0$ if $j \notin \mathcal{N}(i)$, where $\mathcal{N}(i)$ is the k nearest neighbors of sample i. Otherwise, the $s_{i,j}$ value can be calculated by the following Equation (22). Since

different $\mathbf{s}_i$ $(i = 1, ..., n)$ are independent of each other, the solutions of $\mathbf{s}_i$ can be solved separately by parallel optimization. Therefore, rewrite Equation (21) as

$$\min_{\mathbf{1}^T\mathbf{s}_i=1, s_{i,i}=0, s_{i,j}\geq 0} \sum_{j=1}^{n} \left(\|\mathbf{x}_i^T\mathbf{AB} - \mathbf{x}_j^T\mathbf{AB}\|_2^2 s_{i,j} + \gamma s_{i,j}^2 \right) \tag{22}$$

By denoting $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_n] \in \mathbb{R}^{n\times n}$ where $g_{i,j} = \|\mathbf{x}_i\mathbf{AB} - \mathbf{x}_j\mathbf{AB}\|_2^2$, and rewrite Equation (23) as follows:

$$\min_{\mathbf{1}^T\mathbf{s}_i=1, s_{i,i}=0, s_{i,j}\geq 0} \frac{1}{2}\|\mathbf{s}_i + \frac{1}{2\gamma}\mathbf{g}_i\|_2^2 \tag{23}$$

Then we further derive the Lagrangian function of Equation (23) as

$$\begin{aligned} \mathcal{L}(\mathbf{s}_i, \zeta, \boldsymbol{\eta}) &= \frac{1}{2}\|\mathbf{s}_i + \frac{\mathbf{g}_i}{2\gamma}\|_2^2 - \zeta\left(\mathbf{1}^T\mathbf{s}_i - 1\right) - \boldsymbol{\eta}^T\mathbf{s}_i. \\ &= \frac{1}{2}\sum_{j=1}^{n}\left(s_{i,j} + \frac{g_{i,j}}{2\gamma}\right)^2 - \zeta\left(\sum_{j=1}^{n} s_{i,j} - 1\right) - \sum_{j=1}^{n}\eta_j s_{i,j} \end{aligned} \tag{24}$$

where ζ and $\boldsymbol{\eta}$ be the Lagrangian multipliers. By using the Karush–Kuhn–Tucker (KKT) conditions, we further achieve

$$\begin{cases} \forall j, s_{i,j} + \frac{g_{i,j}}{2\gamma} - \zeta - \eta_j = 0 \\ \forall j, s_{i,j} \geq 0 \\ \forall j, s_{i,j}\eta_j = 0 \\ \forall j, \eta_j \geq 0 \end{cases} \tag{25}$$

According to the KKT conditions, we can summarize the following three scenarios based on Equation (25):

$$\begin{cases} \text{scenario 1: } s_{i,j} > 0, \eta_j = 0 \Leftrightarrow s_{i,j} = -\frac{g_{i,j}}{2\gamma} + \zeta > 0 \\ \text{scenario 2: } s_{i,j} = 0, \eta_j > 0 \ \Leftrightarrow -\eta_j = -\frac{g_{i,j}}{2\gamma} + \zeta < 0 \\ \text{scenario 3: } s_{i,j} = \eta_j = 0 \Leftrightarrow -\frac{g_{i,j}}{2\gamma} + \zeta = 0 \end{cases} \tag{26}$$

Finally we have $s_{i,j} = \left(-\frac{g_{i,j}}{2\gamma} + \zeta\right)_+$. To ensure the sparsity of the similarity matrix and thus improve the model robustness, we only consider the k-nearest neighbours of each training sample. Without loss of generality, we suppose that $g_{i,1} \leq g_{i,2} \leq \ldots \leq g_{i,n}, \forall i$. For the vector $\mathbf{s}_i$ we have

$$\begin{cases} s_{i,k} > 0 \Rightarrow -\frac{g_{i,k}}{2\gamma} + \zeta > 0 \\ s_{i,k+1} \leq 0 \Rightarrow -\frac{g_{i,k+1}}{2\gamma} + \zeta \leq 0 \end{cases} \tag{27}$$

according to the constraint $\mathbf{1}^T\mathbf{s}_i = 1$, we have

$$\sum_{j=1}^{k}\left(-\frac{g_{i,j}}{2\gamma} + \zeta\right) = 1 \Rightarrow \zeta = \frac{1}{k} + \frac{1}{2k\gamma}\sum_{j=1}^{k} g_{i,j} \tag{28}$$

based on Equation (27) and (28), we can induce that

$$\frac{kg_{i,k} - \sum_{j=1}^{k} g_{i,j}}{2} < \gamma \leq \frac{kg_{i,k+1} - \sum_{j=1}^{k} g_{i,j}}{2} \tag{29}$$

let $\gamma = \frac{kg_{i,k+1} - \sum_j^k g_{i,j}}{2}$, the closed-form solution of $s_{i,j}$ can be yielded as

$$s_{i,j} = \begin{cases} \frac{g_{i,k+1} - g_{i,j}}{kg_{i,k+1} - \sum_{j=1}^k g_{i,j}}, & j \leq k, \\ 0, & j > k. \end{cases} \tag{30}$$

In summary, the overall pseudo-code of the proposed algorithm to solve the problem (14) is concluded in Algorithm 2.

Algorithm 2 MTFS Method based on Alternating Optimization Algorithm

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{Y} \in \mathbb{R}^{n \times q}$, α, β and λ, k and r;
Output: $\mathbf{A} \in \mathbb{R}^{d \times r}$, $\mathbf{B} \in \mathbb{R}^{r \times q}$, $\mathbf{S} \in \mathbb{R}^{n \times n}$
1. Calculate k nearest neighbors of all training samples;
2. Initialize $\mathbf{S}$ by Equation (8) where $\mathbf{W}$ is an identity matrix;
3. Update the Laplacian matrix $\widetilde{\mathbf{L}}$;
4. **repeat**:
   4.1. Update $\mathbf{A}$ and $\mathbf{B}$ via Algorithm 1;
   4.2. Calculate $\mathbf{S}$ via Equation (27);
   4.3. Calculate the Laplacian matrix $\mathbf{L}$ corresponding to $\mathbf{S}$;
until *converge*;

5. Convergence and Complexity Analysis

To demonstrate the convergence of the proposed algorithm, a Lemma is first listed as follows [44]:

Lemma 1. *For any two non-zero vectors* $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$*, the following equation is always holds.*

$$\|\mathbf{u}\|_2 - \frac{\|\mathbf{u}\|_2^2}{2\|\mathbf{v}\|_2} \leq \|\mathbf{v}\|_2 - \frac{\|\mathbf{v}\|_2^2}{2\|\mathbf{v}\|_2} \tag{31}$$

5.1. Convergence Analysis of Algorithm 2

The convergence of Algorithm 2 is guaranteed by the following Theorem.

Theorem 1. *The value of objective function* (15) *is monotonically decreases until Algorithm 2 converges.*

Proof. Denote $\mathcal{J}\left(\mathbf{A}_{(t)}, \mathbf{B}_{(t)}\right)$ as the objective function of (15) in tth iteration. $\mathbf{W}_{(t)} = \mathbf{A}_{(t)}\mathbf{B}_{(t)}$, where $\mathbf{A}_{(t)}$ and $\mathbf{B}_{(t)}$ are the $\mathbf{A}$ and $\mathbf{B}$ in the tth iteration, respectively. After fixing $\mathbf{S}$, according to Algorithm 1, we can obtain

$$\begin{aligned} \left\langle \mathbf{A}_{(t)}, \mathbf{B}_{(t)} \right\rangle = & \arg\min_{\mathbf{A},\mathbf{B}} \|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t)} - \mathbf{Y}\right)\|_F^2 + \alpha tr\left(\mathbf{W}^T\mathbf{D}\mathbf{W}\right) \\ & + \beta tr\left(\mathbf{W}_{(t)}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}_{(t)}\right) + \lambda tr\left(\mathbf{B}_{(t)}\widetilde{\mathbf{L}}\mathbf{B}_{(t)}^T\right) \end{aligned} \tag{32}$$

Since $\|\mathbf{W}\|_{2,1} = \sum_{i=1}^d \|\mathbf{w}_i\|_2$, hence

$$\begin{aligned} & \|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t+1)} - \mathbf{Y}\right)\|_F^2 + \beta tr\left(\mathbf{W}_{(t+1)}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}_{(t+1)}\right) + \lambda tr\left(\mathbf{B}_{(t+1)}\widetilde{\mathbf{L}}\mathbf{B}_{(t+1)}^T\right) \\ & + \alpha\|\mathbf{W}_{(t+1)}\|_{2,1} + \alpha\sum_{i=1}^d \left(\frac{\|\mathbf{w}_{i(t+1)}\|_2^2}{2\|\mathbf{w}_{i(t)}\|_2} - \|\mathbf{w}_{i(t+1)}\|_2^2\right) \\ & \leq \|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t)} - \mathbf{Y}\right)\|_F^2 + \beta tr\left(\mathbf{W}_{(t)}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}_{(t)}\right) + \lambda tr\left(\mathbf{B}_{(t)}\widetilde{\mathbf{L}}\mathbf{B}_{(t)}^T\right) \\ & + \alpha\|\mathbf{W}_{(t)}\|_{2,1} + \alpha\sum_{i=1}^d \left(\frac{\|\mathbf{w}_{i(t)}\|_2^2}{2\|\mathbf{w}_{i(t)}\|_2} - \|\mathbf{w}_{i(t)}\|_2^2\right) \end{aligned} \tag{33}$$

where $\mathbf{w}_{i(t)}$ and $\mathbf{w}_{i(t+1)}$ denote the ith row of $\mathbf{W}_{(t)}$ and $\mathbf{W}_{(t+1)}$, respectively. According to Lemma 1, we have

$$\|\mathbf{w}_{i(t+1)}\|_2 - \frac{\|\mathbf{w}_{i(t+1)}\|_2^2}{2\|\mathbf{w}_{i(t)}\|_2} \leq \|\mathbf{w}_{i(t)}\|_2 - \frac{\|\mathbf{w}_{i(t)}\|_2^2}{2\|\mathbf{w}_{i(t)}\|_2} \tag{34}$$

By plugging Equation (34) into Equation (33), we have

$$\begin{aligned} &\|\mathbf{H}\Big(\mathbf{X}\mathbf{W}_{(t+1)} - \mathbf{Y}\Big)\|_F^2 + \beta tr\Big(\mathbf{W}_{(t+1)}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}_{(t+1)}\Big) \\ &+ \alpha\sum_{i=1}^{d}\|\mathbf{w}_{i(t+1)}\|_2^2 + \lambda tr\Big(\mathbf{B}_{(t+1)}\widetilde{\mathbf{L}}\mathbf{B}_{(t+1)}^T\Big) \\ &\leq \|\mathbf{H}\Big(\mathbf{X}\mathbf{W}_{(t)} - \mathbf{Y}\Big)\|_F^2 + \beta tr\Big(\mathbf{W}_{(t)}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}_{(t)}\Big) \\ &+ \alpha\sum_{i=1}^{d}\|\mathbf{w}_{i(t)}\|_2^2 + \beta tr\Big(\mathbf{B}_{(t)}\widetilde{\mathbf{L}}\mathbf{B}_{(t)}^T\Big) \end{aligned} \tag{35}$$

and further we have

$$\begin{aligned} &\|\mathbf{H}\Big(\mathbf{X}\mathbf{W}_{(t+1)} - \mathbf{Y}\Big)\|_F^2 + \beta tr\Big(\mathbf{W}_{(t+1)}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}_{(t+1)}\Big) \\ &+ \alpha\|\mathbf{W}_{i(t+1)}\|_{2,1} + \lambda tr\Big(\mathbf{B}_{(t+1)}\widetilde{\mathbf{L}}\mathbf{B}_{(t+1)}^T\Big) \\ &\leq \|\mathbf{H}\Big(\mathbf{X}\mathbf{W}_{(t)} - \mathbf{Y}\Big)\|_F^2 + \beta tr\Big(\mathbf{W}_{(t)}^T\mathbf{X}^T\mathbf{L}\mathbf{X}\mathbf{W}_{(t)}\Big) \\ &+ \alpha\|\mathbf{W}_{i(t)}\|_{2,1} + \lambda tr\Big(\mathbf{B}_{(t)}\widetilde{\mathbf{L}}\mathbf{B}_{(t)}^T\Big) \end{aligned} \tag{36}$$

Hence, we have the following inequality:

$$\mathcal{J}\Big(\mathbf{A}_{(t+1)}, \mathbf{B}_{(t+1)}\Big) \leq \mathcal{J}\Big(\mathbf{A}_{(t)}, \mathbf{B}_{(t)}\Big).$$

Therefore, $\mathcal{J}\Big(\mathbf{A}_{(t)}, \mathbf{B}_{(t)}\Big)$ is monotonically decreasing until convergence, and Theorem 1 proved. □

5.2. Convergence Analysis of Algorithm 1

Likewise, we also prove the convergence of Algorithm 1 according to the following Theorem 2.

Theorem 2. *The objective function* (21) *monotonically decreases with each optimization step until Algorithm 1 converges.*

Proof. According to Theorem 1, after the tth iteration, the optimal $\mathbf{A}_{(t)}$, $\mathbf{B}_{(t)}$ and $\mathbf{S}_{(t)}$ have obtained, we need to calculate $\mathbf{S}_{(t+1)}$ by fixing $\mathbf{A}_{(t)}$ and $\mathbf{B}_{(t)}$ in the $(t+1)$th iteration. Furthermore, the $\mathbf{S}_{(t+1)}$ can converge to the globally optimal solution according to Equation (30) since $s_{i,j}^{(t+1)}$ has the closed-form solution. Therefore, we have

$$
\begin{aligned}
&\|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t)}-\mathbf{Y}\right)\|_F^2+\alpha\|\mathbf{W}_{(t)}\|_{2,1}\\
&+\beta\left(\sum_{i,j=1}^{n}\|\mathbf{x}_i^T\mathbf{W}_{(t)}-\mathbf{x}_j^T\mathbf{W}_{(t)}\|_2^2 s_{i,j}^{(t+1)}+\gamma\sum_{i=1}^{n}\|\mathbf{s}_i^{(t+1)}\|_2^2\right)\\
&+\lambda\sum_{i,j=1}^{d}\|\mathbf{b}_i^{(t)}-\mathbf{b}_j^{(t)}\|_2^2\tilde{s}_{i,j}\\
&\leq\|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t)}-\mathbf{Y}\right)\|_F^2+\alpha\|\mathbf{W}_{(t)}\|_{2,1}\\
&+\beta\left(\sum_{i,j=1}^{n}\|\mathbf{x}_i^T\mathbf{W}_{(t)}-\mathbf{x}_j^T\mathbf{W}_{(t)}\|_2^2 s_{i,j}^{(t)}+\gamma\sum_{i=1}^{n}\|\mathbf{s}_i^{(t)}\|_2^2\right)\\
&+\lambda\sum_{i,j=1}^{d}\|\mathbf{b}_i^{(t)}-\mathbf{b}_j^{(t)}\|_2^2\tilde{s}_{i,j}
\end{aligned}
\tag{37}
$$

where $\mathbf{s}_i^{(t)}$ and $\mathbf{s}_i^{(t+1)}$ are the ith row of $\mathbf{S}_{(t)}$ and $\mathbf{S}_{(t+1)}$, respectively. When fixing $\mathbf{S}_{(t+1)}$ to update $\mathbf{A}_{(t+1)}$ and $\mathbf{B}_{(t+1)}$, we have the following inequality,

$$
\begin{aligned}
&\|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t+1)}-\mathbf{Y}\right)\|_F^2+\alpha\|\mathbf{W}_{(t+1)}\|_{2,1}+\lambda\sum_{i,j=1}^{d}\|\mathbf{b}_i^{(t+1)}-\mathbf{b}_j^{(t+1)}\|_2^2\tilde{s}_{i,j}\\
&+\beta\left(\sum_{i,j=1}^{n}\|\mathbf{x}_i^T\mathbf{W}_{(t+1)}-\mathbf{x}_j^T\mathbf{W}_{(t+1)}\|_2^2 s_{i,j}^{(t+1)}+\gamma\sum_{i=1}^{n}\|\mathbf{s}_i^{(t+1)}\|_2^2\right)\\
&\leq\|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t)}-\mathbf{Y}\right)\|_F^2+\alpha\|\mathbf{W}_{(t)}\|_{2,1}+\lambda\sum_{i,j=1}^{d}\|\mathbf{b}_i^{(t)}-\mathbf{b}_j^{(t)}\|_2^2\tilde{s}_{i,j}\\
&+\beta\left(\sum_{i,j=1}^{n}\|\mathbf{x}_i^T\mathbf{W}_{(t)}-\mathbf{x}_j^T\mathbf{W}_{(t)}\|_2^2 s_{i,j}^{(t+1)}+\gamma\sum_{i=1}^{n}\|\mathbf{s}_i^{(t+1)}\|_2^2\right)
\end{aligned}
\tag{38}
$$

By combining Equation (37) and (38), we obtain

$$
\begin{aligned}
&\|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t+1)}-\mathbf{Y}\right)\|_F^2+\alpha\|\mathbf{W}_{(t+1)}\|_{2,1}\\
&+\beta\left(\sum_{i,j=1}^{n}\|\mathbf{x}_i^T\mathbf{W}_{(t+1)}-\mathbf{x}_j^T\mathbf{W}_{(t+1)}\|_2^2 s_{i,j}^{(t+1)}+\gamma\sum_{i=1}^{n}\|\mathbf{s}_i^{(t+1)}\|_2^2\right)\\
&+\lambda\sum_{i,j=1}^{d}\|\mathbf{b}_i^{(t+1)}-\mathbf{b}_j^{(t+1)}\|_2^2\tilde{s}_{i,j}\\
&\leq\|\mathbf{H}\left(\mathbf{X}\mathbf{W}_{(t)}-\mathbf{Y}\right)\|_F^2+\alpha\|\mathbf{W}_{(t)}\|_{2,1}\\
&+\beta\left(\sum_{i,j=1}^{n}\|\mathbf{x}_i^T\mathbf{W}_{(t)}-\mathbf{x}_j^T\mathbf{W}_{(t)}\|_2^2 s_{i,j}^{(t)}+\gamma\sum_{i=1}^{n}\|\mathbf{s}_i^{(t)}\|_2^2\right)\\
&+\lambda\sum_{i,j=1}^{d}\|\mathbf{b}_i^{(t)}-\mathbf{b}_j^{(t)}\|_2^2\tilde{s}_{i,j}
\end{aligned}
\tag{39}
$$

According to Equation (38), the value of objective function monotonically decreases after each iteration of Algorithm 1, Theorem 2 is proved. □

5.3. Complexity Analysis

We further analyze the computational complexity of the proposed algorithm. In each iteration, the computation cost of Algorithm 1 focuses on calculating $\mathbf{P}^{-1}\mathbf{X}^T\mathbf{H}\mathbf{Y}\mathbf{B}^T\left(\mathbf{B}\mathbf{B}^T\right)^{-1}$ and solving the Sylvester function, the corresponding complexity are $\max\{\mathcal{O}(r^3),\mathcal{O}(d^3),\mathcal{O}(ndq),\mathcal{O}(dqr)\}$ and $\mathcal{O}(q^3)$, respectively. The complexity of Algo-

rithm 2 stems from calculating the matrix $\mathbf{G}$, the computation cost is $\max\{\mathcal{O}(n^2d), \mathcal{O}(n^2q)\}$. Since $r \leq \min(d, q)$, $n, d \gg r, q$, and it is experimentally observed that Algorithm 1 can converge within 30 iterations on different data sets. Hence, the computational complexity of the proposed method is approximate $\mathcal{O}(td^3 + tnd^2)$, where t $(n, d \gg t)$ is the iteration of the whole alternating optimization.

6. Experiments

6.1. Datasets

We test the proposed approach on eight high-dimensional datasets (http://mulan.sourceforge.net/datasets-mtr.html, accessed on 18 January 2024), which are all from the public website Mulan [45]. All selected datasets are commonly used benchmark datasets for measuring MTR modeling performance. The detailed statistics of these datasets are shown in Table 1. We follow the strategies in [18] to impute the datasets with missing values, i.e., RF1 and RF2, which are replaced with sample means in the datasets.

Table 1. Characters of the datasets.

Datasets	Instances	Features	Targets	#-Fold	Domains
ATP1d	337	411	6	10	Price prediction
ATP7d	296	411	6	10	Price prediction
OES10	403	298	16	10	Artificial
OES97	334	263	16	10	Artificial
RF1	9125	64	8	2	Environment
RF2	9125	576	8	2	Environment
SCM1d	9803	280	16	2	Environment
SCM20d	8966	61	16	2	Environment

6.2. Compared Methods

In this paper, different MTFS methods are selected to compare the performance with the proposed approach.

- **MTFS** [44]: The row sparsity constraint is imposed on the weight matrix by $\ell_{2,1}$-norm regularization,
$$\min_{\mathbf{W}} \|\mathbf{XW} - \mathbf{Y}\|_F^2 + \lambda \|\mathbf{W}\|_{2,1} \tag{40}$$
where λ is the tuning parameter, we set the parameters to range as $\{10^{-3}, 10^{-2}, \ldots, 10^3\}$ empirically.
- **RFS** [46]: By jointly imposing $\ell_{2,1}$-norm regularization on the loss function and the weight matrix, the objective function of RFS is:
$$\min_{\mathbf{W}} \|\mathbf{XW} - \mathbf{Y}\|_{2,1} + \lambda \|\mathbf{W}\|_{2,1} \tag{41}$$
where the parameter λ range as $\{10^{-3}, 10^{-2}, \ldots, 10^3\}$.
- **SSFS** [29]: The multi-layer regression structure is constructed by low-dimensional embedding, and the loss function, weight matrix and structure matrix are joint $\ell_{2,1}$-norm regularized, and the objective function is:
$$\min_{\mathbf{W},\mathbf{U}} \|\mathbf{ZU} - \mathbf{Y}\|_{2,1} + \lambda \|\mathbf{W}\|_{2,1} + \beta \|\mathbf{U}\|_{2,1} \tag{42}$$
where $\mathbf{Z} = \mathbf{XW}$, λ and β are tuning parameters. All tuning parameters' range as $10^{[-3:1:3]}$.

- **HLMR-FS** [47]: The method introduces a hyper-graph Laplacian regularization to maintain the correlation structure between samples and find the hidden correlation structure among different target variables via the low-rank constraint.

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} & \|\mathbf{Y}-\mathbf{XAB}\|_F^2+\alpha\|\mathbf{AB}\|_{2,p}+\beta tr\left(\mathbf{B}^T\mathbf{A}^T\mathbf{X}^T\mathbf{L}_H\mathbf{XAB}\right) \\ & s.t.\ \mathbf{A}^T\mathbf{A}=\mathbf{I} \end{aligned} \tag{43}$$

  where $\mathbf{L}_H$ is the graph Laplacian matrix between the predicted output vectors of different training samples. α and β searched in the grid $10^{[-3:1:3]}$, and p searched in the grid $\{0.1,\ldots,1.9\}$.
- **LFR-FS** [30]: The method captures the correlation between different objectives through low-rank constraint, and by designing $\ell_{2,p}$-norm regularization on the loss function and the regression matrix, the learning of the orthogonal subspace enables multiple outputs to share the same low-rank data structure to obtain the corresponding feature selection results.

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B}} & \|\mathbf{Y}-\mathbf{XAB}\|_{2,p}+\alpha\|\mathbf{A}\|_{2,p} \\ & s.t.\ \mathbf{A}^T\mathbf{A}=\mathbf{I} \end{aligned} \tag{44}$$

  where α searched in the grid $10^{[-3:1:3]}$, and p varied in $\{0.1,\ldots,1.9\}$.
- **VMFS** [26]: VMFS ranks each feature in MTR via the famous Multi-Criteria Decision-Making (MCDM) method called VIKOR.
- **RSSFS** [48]: RSSFS uses the mixed convex and non-convex $\ell_{2,p}$-norm minimization on both regularization and loss function for joint sparse feature selection, and the objective function is:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{H},\mathbf{Q}} & \left\|\mathbf{X}^T\mathbf{W}-\mathbf{Y}\right\|_{2,p}^p+\alpha\|\mathbf{W}\|_{2,p}^p+\beta\|\mathbf{W}-\mathbf{QH}\|_F^2 \\ & s.t.\mathbf{Q}^T\mathbf{Q}=\mathbf{I} \end{aligned} \tag{45}$$

  In the experiments, the regularization parameter α and β were set in $10^{[-3:1:3]}$, and p varied in $\{0.1,\ldots,0.9\}$.

In addition to choosing the above-compared methods, we also perform regressions by using the original data without feature selection as a **Baseline** to test and validate the effectiveness of the proposed method. We adopt the Multi-output Kernel Ridge Regression (mKRR) [49] to obtain the regression result corresponding to feature subsets obtained by different MTFS methods. In mKKR, Radial Basis Function (RBF) is utilized as the kernel function, and the kernel parameter and the regularization parameter range as $10^{[-3:1:3]}$ on the training data [29]. For different data sets, 70% of the samples are selected as the training set and the rest as the test set. As is shown in Table 1, we use two-fold cross-validation for RF1/RF2 and SCM1d/SCM20d and five-fold cross-validation on the training data for the rest of the datasets to conduct model selection.

6.3. Evaluation Metrics

Two evaluation metrics are employed in experiment, including average Correlation Coefficient (aCC) and average Relative Root Mean Squared Error (aRRMSE) [47]. The definition of aCC is as follows,

$$aCC = \frac{1}{q}\sum_{i=1}^{q}\frac{\sum_{j=1}^{N_{test}}\left(y_i^{(j)}-\bar{y}_i\right)\left(\hat{y}_i^{(j)}-\tilde{y}_i\right)}{\sqrt{\sum_{j=1}^{N_{test}}\left(y_i^{(j)}-\bar{y}_i\right)^2\sum_{j=1}^{N_{test}}\left(\hat{y}_i^{(j)}-\tilde{y}_i\right)^2}} \tag{46}$$

where $y_i^{(j)}$ and $\hat{y}_i^{(j)}$ are the real and predicted values of the jth sample on the target i, $\bar{y}_i$ and $\tilde{y}_i$ are the mean of true value and the predicted value on target i over the test set, respectively. Likewise, the formula for aRRMSE is given:

$$aRRMSE = \frac{1}{q}\sum_{i=1}^{q}\sqrt{\frac{\sum_{j=1}^{N_{test}}\left(y_i^{(j)}-\hat{y}_i^{(j)}\right)^2}{\sum_{j=1}^{N_{test}}\left(y_i^{(j)}-\mathbf{y}_i\right)^2}} \tag{47}$$

where $\mathbf{y}_i$ is the average value of the training samples on the ith target.

6.4. Results on the Data Sets

Figures 1 and 2 show the aRRMSE and aCC values for different MTFS methods on different data sets, respectively. For ATP1d and ATP7d, we choose 60, 70, 80, 90, 100, 110 features. For OES10, RF2 and SCM1d, we choose 60, 70, 80, 90, 100 and 110 features. For OES97, we choose 40, 60, 80, 100, 120 and 140 features. For RF1, we choose 10, 15, 20, 25, 30 and 35 features. For SCM20d, we choose 20, 25, 30, 35, 40 and 45 features.

Meanwhile, the best aCC and aRRMSE values of compared MTFS methods on various datasets are ranked, and the average rank of different methods on all datasets is calculated. The *Friedman* test [50] with the significant level $\alpha = 0.05$ is employed, and we utilize *Bonferroni-Dunn* test [50] as the post hoc test to further analysis of the comparison. The critical difference (CD) is calculated to measure the difference between the proposed method and other algorithms. The calculation of CD is as follows:

$$CD = q_\alpha\sqrt{\frac{n(n+1)}{6T}}. \tag{48}$$

where n is the number of algorithms compared, and T is the number of datasets. At significance level $\alpha = 0.05$, the corresponding $q_\alpha = 3.73$, thus we have CD = 2.41 ($n = 9, T = 8$). Figures 3 and 4 show the average ranks of different feature selection methods based on aRRMSE and aCC metrics.

Obviously, from Figures 1 and 2, we can observe that for different data sets, selecting the correct number of feature subsets can achieve better results than the baseline, which indicates that for MTR problems, a practical feature selection method can not only improve the computational efficiency of the model but also improve the comprehensive performance of the model on different targets. Furthermore, the regression performance does not necessarily improve as the size of the selected features increases. On the contrary, in most cases, such as OES97, RF1, SCM20d, etc., the performance decreases as the number of selected features increases, indicating the presence of redundant or irrelevant features in the original feature set may significantly reduce the performance of regression.

For most cases, SSFS, HLMR-FS and the proposed method can obtain a lower aRRMSE and higher aCC than MTFS, RFS and VMFS. It shows that the performance of MTFS can be improved via a low-rank constraint. The proposed method not only considers the structural information of different samples in feature space but also uses the intrinsic correlation information between targets to improve the performance of MTFS. Furthermore, the proposed method can outperform the baseline in most cases, regardless of the number of features. It indicates that the proposed method can effectively alleviate the influence of redundant features, thereby maintaining outstanding performance on the selected subset even if some redundant features are included.

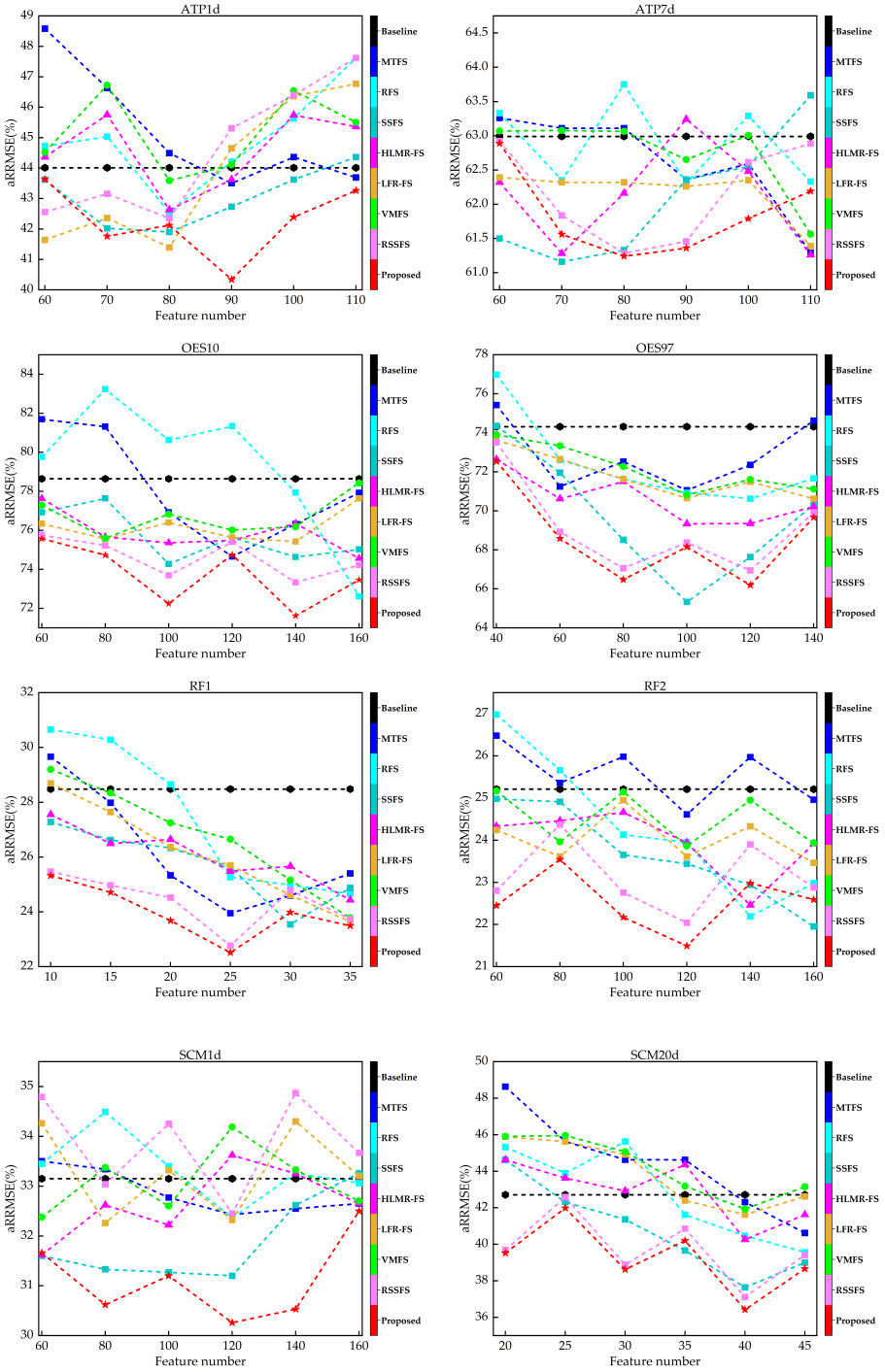

Figure 1. aRRMSE results compared with compared methods under different number of selected features.

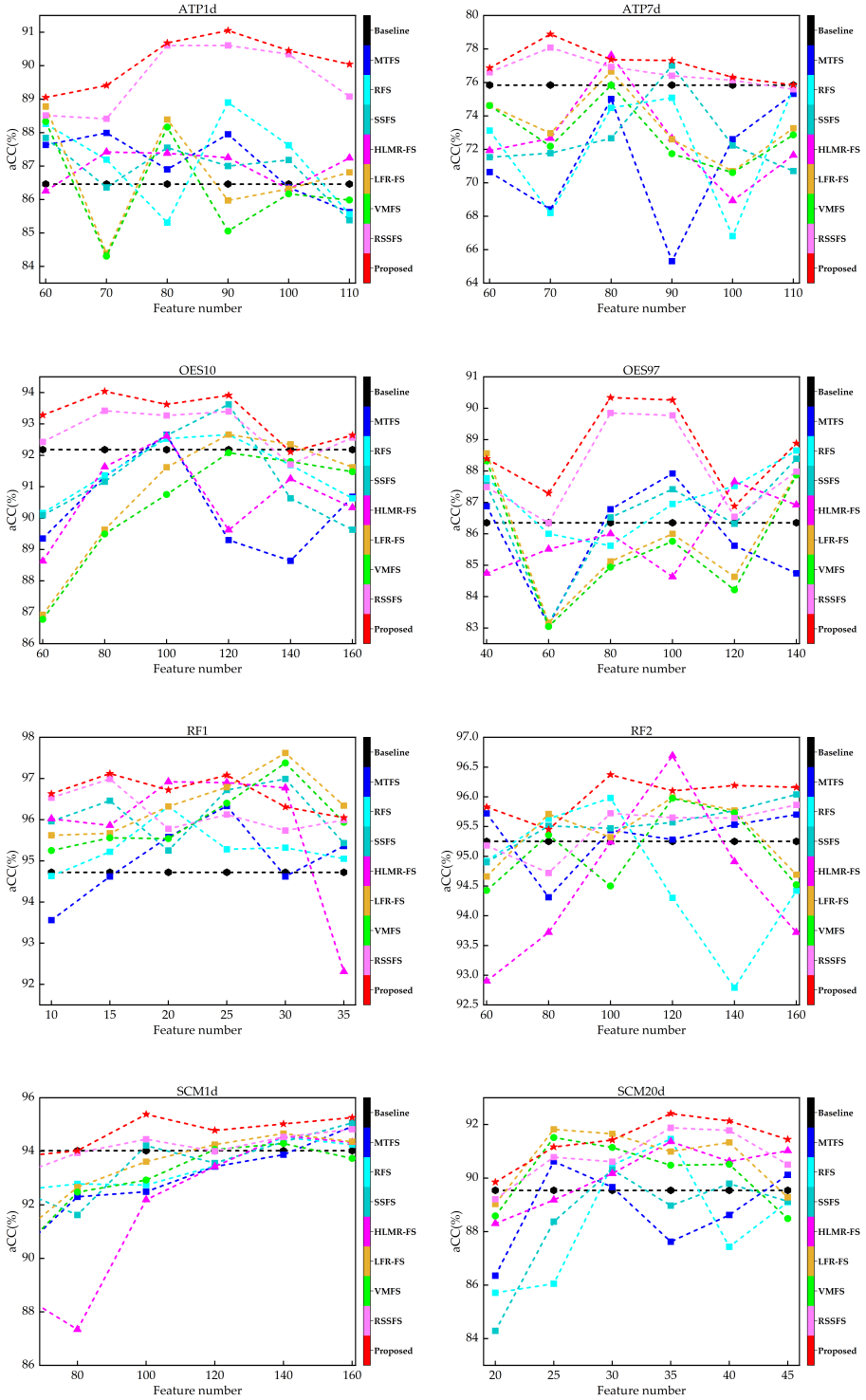

Figure 2. aCC results compared with state-of-the-art methods under different number of selected features.

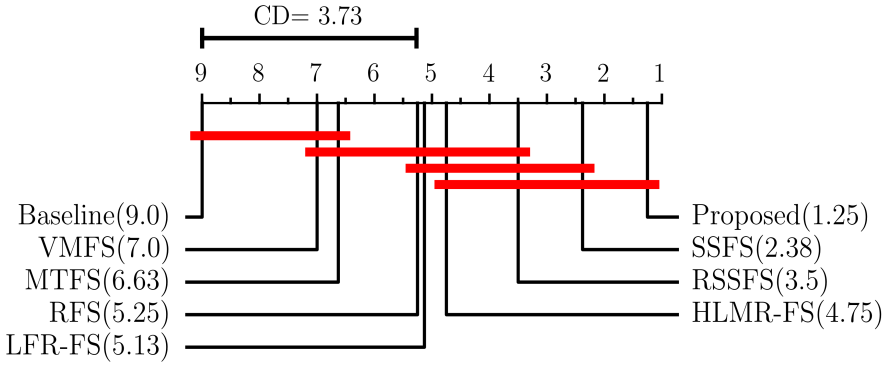

Figure 3. Average rank of different feature selection methods based on aRRMSE under Bonferroni–Dunn test.

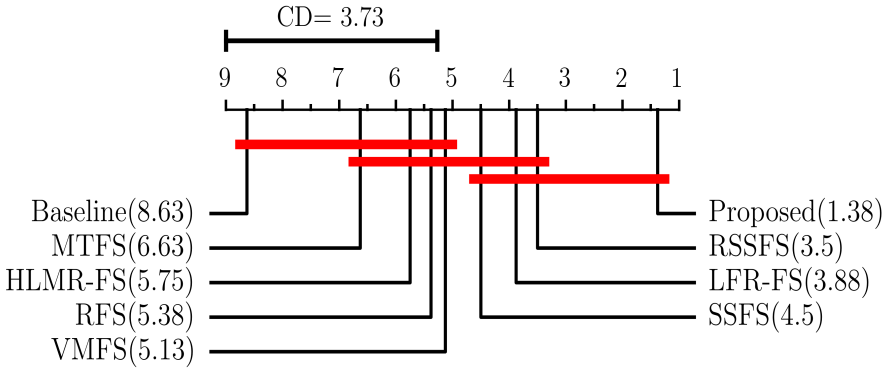

Figure 4. Average rank of feature selection methods based on aCC under Bonferroni–Dunn test.

6.5. Effect of Low-Rank Constraint

We also investigate the influence of different ranks over different data sets, set $r = 1, 2, \ldots, q$. The performance when $r = q$ is taken as the performance of the algorithm at full rank, on account of the condition $r \leq \min\{d, q\}$. The number of input features d in the adopted data set is much larger than q, so the corresponding rank value of the regression matrix at full rank is q. We set $r = \{1, 2, \ldots, 6\}$ in the ATP1d; $r = \{1, 2, \ldots, 16\}$ in the OES10; $r = \{1, 2, \ldots, 8\}$ in the RF1; $r = \{1, 2, \ldots, 16\}$ in the SCM1d. By setting different values of r to impose low-rank constraints on $\mathbf{A}$ and $\mathbf{B}$. The fluctuations of aRRMSE and aCC values of the algorithm with α fixed are shown in Figure 5.

From Figure 5, it is evident that performance of the proposed method can be effectively improved by choosing the appropriate rank value for different data sets. In addition, most of the rank values in different data sets are better than the performance at full rank, which indicates that the regression matrix can decouple the inter-features and inter-target correlation via embedding the latent space of different dimensions, and it is beneficial to improve the regression performance and robustness of the model.

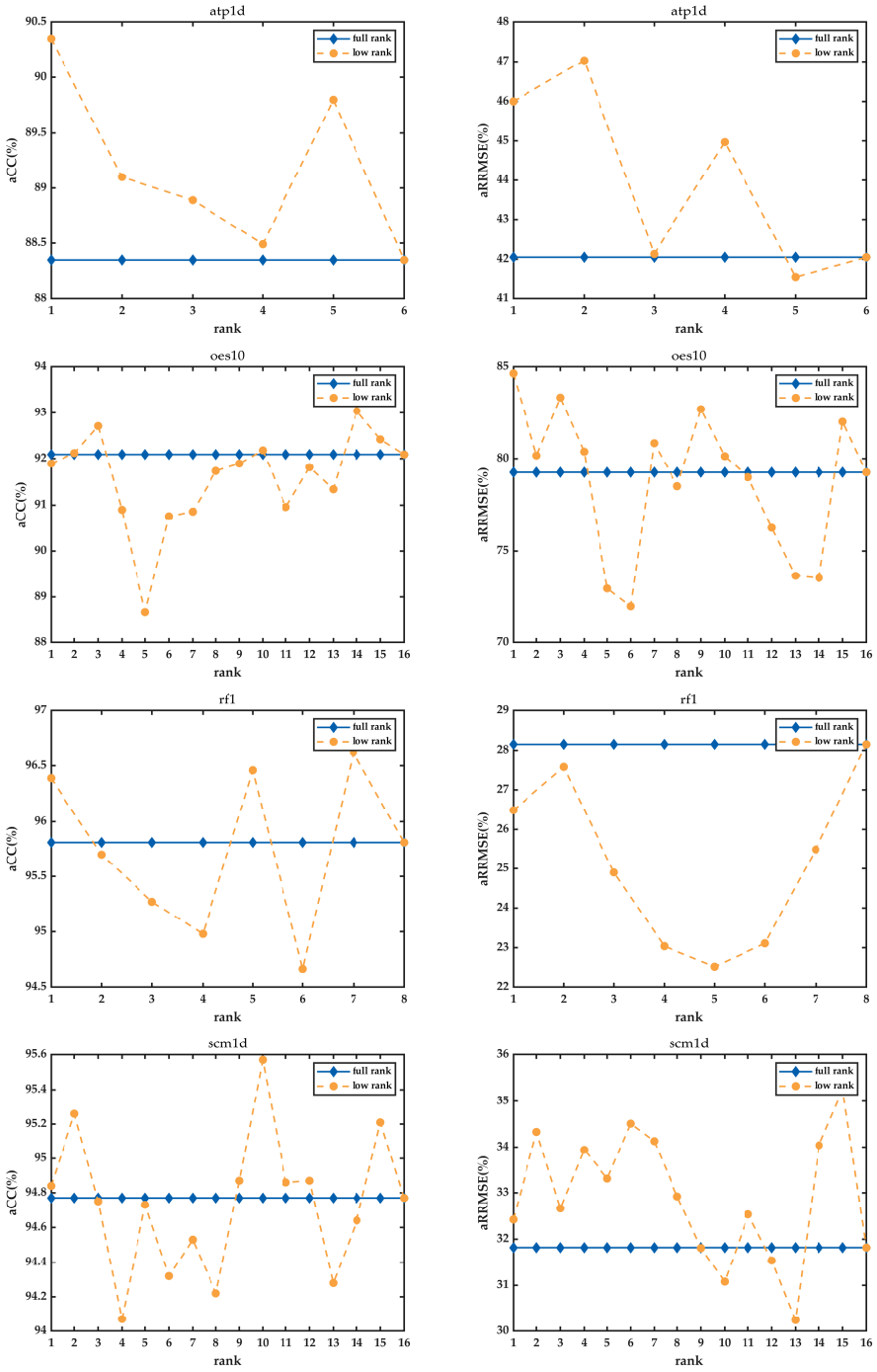

Figure 5. Performance of feature selection methods under different low-rank constraints.

6.6. Parameter Sensitivity

In this section, we further perform sensitivity analysis on different parameters in the proposed feature selection method. Since there is a closed-form solution for γ, we focus on sensitivity analysis for the regularization parameters α, λ and β. First of all, we tuned the parameter α within the range of $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ with $\lambda = 0.01$ and $\beta = 0.01$. Likewise, we tuned parameters λ and β in $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ with $\alpha = 0.1$, and the results are shown in Figures 6 and 7.

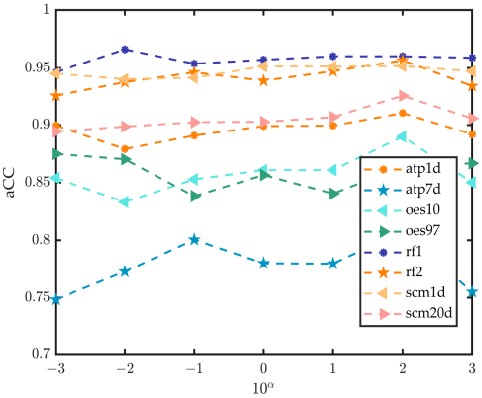

Figure 6. Sensitivity analysis of the parameter α with λ and β fixed.

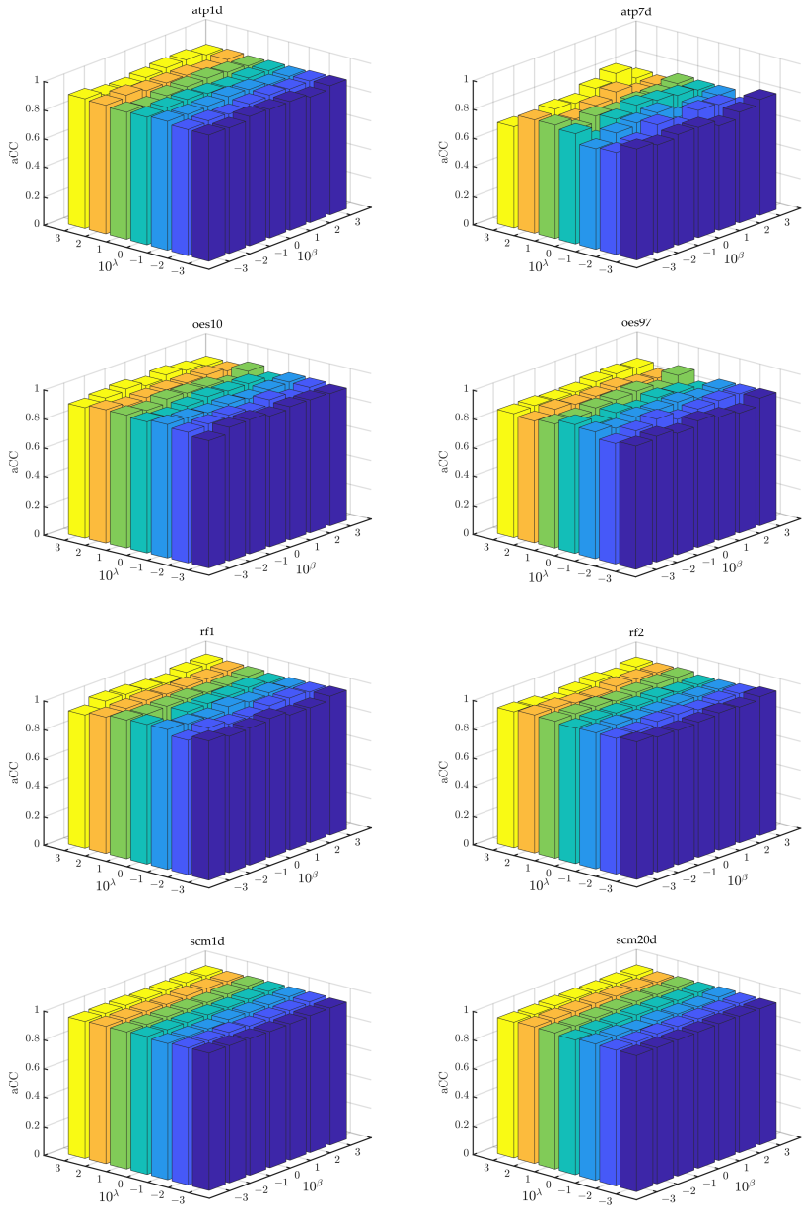

Figure 7. Sensitivity analysis of the parameter λ and β with α fixed.

In Figure 6, we can see that the variation of the parameter α will bring a certain degree of fluctuation in the model performance with λ and β fixed, which indicates that the proposed method is sensitive to α. Hence, parameter α is vital to determine the performance of the proposed method. From Figure 7, it can be seen that the changes in model performance after changes in parameters λ and β in ranges are not as significant as that of parameter α. However, properly tuning parameters λ and β can still improve the performance.

6.7. Convergence Study

We also plot the convergence curves of the objective function value of Equation (12) when the algorithm is updated iteratively on different data sets. As shown in Figure 8, it can be observed that ATP1d, ATP7d and RF1 can converge to the optimum within 20 iterations. The rest of the datasets can converge within 30 iterations, and the objective function converges quickly in the first few iterations. It indicates that the proposed alternating optimization algorithm can efficiently converge to the global optimum. Moreover, the monotone decrease of the objection function value demonstrates that the proposed problem can converge well. It confirms the effectiveness of the alternating optimization algorithm in addressing the proposed problem.

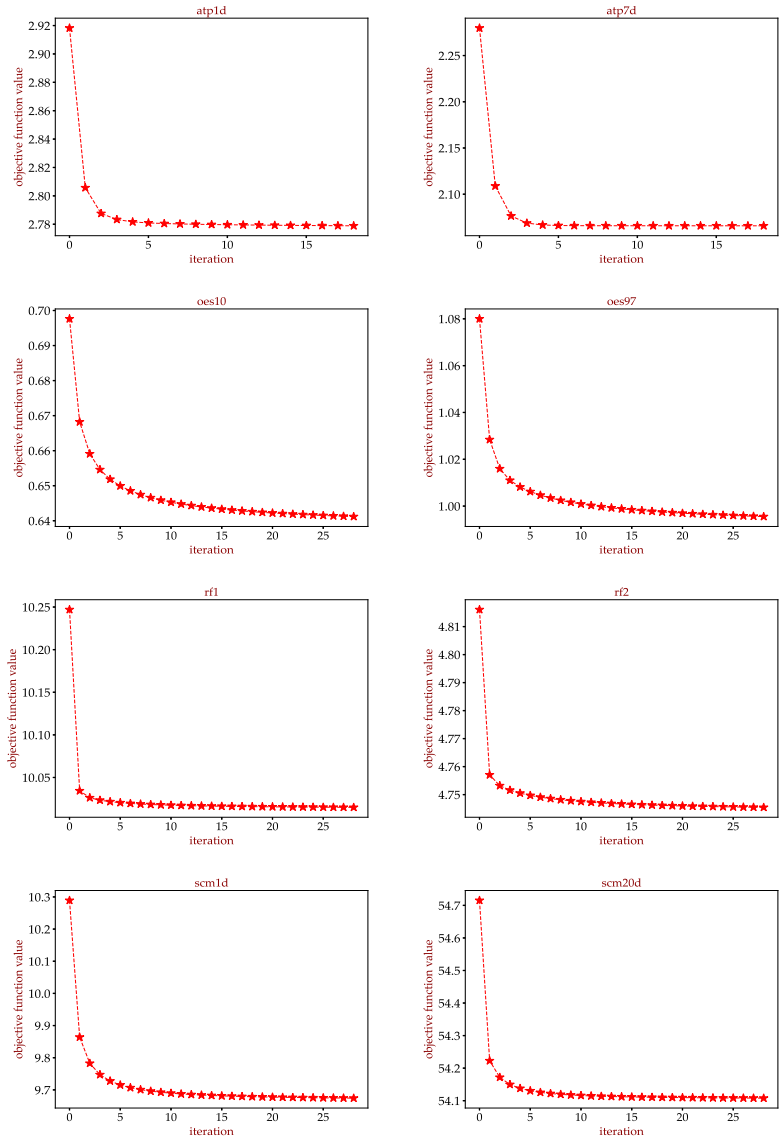

Figure 8. Convergence curves of the proposed method under different data sets.

7. Conclusions

This paper has proposed a novel MTFS method based on adaptive graph learning and global target correlations to perform feature selection in MTR problem. Considering the existence of feature redundancy and noise in the original data, adaptive graph learning based on the sample local structure is introduced. Meanwhile, a manifold regularizer based on the target correlations is constructed to explore the inter-target correlation, which enables the regression matrix to consider the correlation between targets in the sparse and low-rank learning process. Finally, an alternating optimization algorithm is proposed to solve the objective function of the MTFS problem, and the convergence of the algorithm is demonstrated both theoretically and empirically. Through extensive experiments, it is demonstrated that the proposed method has superior performance compared with other mainstream embedding MTFS algorithms. The proposed method can effectively select features for MTR data, and then improve the efficiency and accuracy of MTR modelling.

In the future, we will extend the proposed method to cope with the semi-supervised and unsupervised feature selection tasks in MTR scenarios, we will try to introduce more manifold constraints and low-rank structures to the feature selection problem of MTR and test its performance and we will also explore whether it can solve the feature selection problem in multi-task learning and MLC.

Author Contributions: Conceptualization, Y.Z. and D.H.; methodology, Y.Z.; software, D.H.; validation, Y.Z. and D.H.; formal analysis, Y.Z.; investigation, D.H.; resources, Y.Z.; data curation, D.H.; writing—original draft preparation, D.H.; writing—review and editing, D.H.; visualization, Y.Z.; supervision, Y.Z.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Social Science Foundation of China, grant number 18BGL287, 19CGL073.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank Zhang Kan for his fund support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li, H.; Zhang, W.; Chen, Y.; Guo, Y.; Li, G.-Z.; Zhu, X. A novel multi-target regression framework for time-series prediction of drug efficacy. *Sci. Rep.* **2017**, *7*, 40652. [CrossRef] [PubMed]
2. Kocev, D.; Džeroski, S.; White, M.D.; Newell, G.R.; Griffioen, P. Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecol. Model.* **2009**, *220*, 1159–1168. [CrossRef]
3. Sicki, D.M. Multi-target tracking using multiple passive bearings-only asynchronous sensors. *IEEE Trans. Aerosp. Electron. Syst.* **2008**, *44*, 1151–1160.
4. He, D.; Sun, S.; Xie, L. Multi-Target Regression Based on Multi-Layer Sparse Structure and Its Application in Warships Scheduled Maintenance Cost Prediction. *Appl. Sci.* **2023**, *13*, 435. [CrossRef]
5. Zhen, X.; Islam, A.; Bhaduri, M.; Chan, I.; Li, S. Descriptor Learning via Supervised Manifold Regularization for Multi-output Regression. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2035–2047. [PubMed]
6. Wang, X.; Zhen, X.; Li, Q.; Shen, D.; Huang, H. Cognitive Assessment Prediction in Alzheimer's Disease by Multi-Layer Multi-Target Regression. *Neuroinformatics* **2018**, *16*, 285–294. [CrossRef] [PubMed]
7. Ghosn, J.; Bengio, Y. Multi-task learning for stock selection. In Proceedings of the 9th Advances in Neural Information Processing Systems, Denver, CO, USA, 2–5 December 1996; pp. 946–952.
8. Chen, B.J.; Chang, M.W. Load forecasting using support vector Machines: A study on EUNITE competition 2001. *IEEE Trans. Power Syst.* **2004**, *19*, 1821–1830. [CrossRef]
9. Cai, J.; Luo, J.; Wang, S.; Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing* **2018**, *300*, 70–79. [CrossRef]
10. Dinov, I.D. Variable/feature selection. In *Data Science and Predictive Analytics: Biomedical and Health Applications Using R*; Springer International Publishing: Cham, Switzerland, 2018; pp. 557–572.
11. Sechidis, K.; Spyromitros-Xioufis, E.; Vlahavas, I. Information Theoretic Multi-Target Feature Selection via Output Space Quantization. *Entropy* **2019**, *21*, 855. [CrossRef]
12. He, X.; Deng, C.; Niyogi, P. Laplacian Score for Feature Selection. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005; Volume 18.

13. Sechidis, K.; Brown, G. Simple strategies for semi-supervised feature selection. *Mach. Learn.* **2018**, *107*, 357–395. [CrossRef]
14. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324. [CrossRef]
15. Tang, C.; Liu, X.; Li, M.; Wang, P.; Chen, J.; Wang, L.; Li, W. Robust unsupervised feature selection via dual self-representation and manifold regularization. *Knowl.-Based Syst.* **2018**, *145*, 109–120. [CrossRef]
16. Nouri-Moghaddam, B.; Ghazanfari, M.; Fathian, M. A novel multi-objective forest optimization algorithm for wrapper feature selection. *Expert Syst. Appl.* **2021**, *175*, 114737. [CrossRef]
17. Spyromitros-Xioufis, E.; Tsoumakas, G.; Groves, W.; Vlahavas, I. *Multi-Label Classification Methods for Multi-Target Regression*; Cornell University Library: Ithaca, NY, USA, 2014.
18. Spyromitros-Xioufis, E.; Tsoumakas, G.; Groves, W.; Vlahavas, I. Multi-target regression via input space expansion: treating targets as inputs. *Mach. Learn.* **2016**, *104*, 55–98. [CrossRef]
19. Tsoumakas, G.; Spyromitros-Xioufis, E.; Vrekou, A.; Vlahavas, I. *Multi-Target Regression via Random Linear Target Combinations*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 225–240.
20. Zhu, Y.; Kwok, J.T.; Zhou, Z.H. Multi-Label Learning with Global and Local Label Correlation. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1081–1094. [CrossRef]
21. Zhu, X.; Zhang, S.; Hu, R.; Zhu, Y.; Song, J. Local and Global Structure Preservation for Robust Unsupervised Spectral Feature Selection. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 517–529. [CrossRef]
22. Huang, Y.; Shen, Z.; Cai, F.; Li, T.; Lv, F. Adaptive graph-based generalized regression model for unsupervised feature selection. *Knowl.-Based Syst.* **2021**, *227*, 107156. [CrossRef]
23. Zhen, X.; Yu, M.; He, X.; Li, S. Multi-Target Regression via Robust Low-Rank Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 497–504. [CrossRef]
24. Zhen, X.; Yu, M.; Zheng, F.; Nachum, I.B.; Bhaduri, M.; Laidley, D.; Li, S. Multitarget Sparse Latent Regression. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1575–1586. [CrossRef]
25. Yang, J.; Zhang, D.; Yang, J.Y.; Niu, B. Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Applications to Face and Palm Biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 650–664. [CrossRef]
26. Hashemi, A.; Dowlatshahi, M.B.; Nezamabadi-pour, H. VMFS: A VIKOR-based multi-target feature selection. *Expert Syst. Appl.* **2021**, *182*, 115224. [CrossRef]
27. Petkovi, M.; Kocev, D.; Deroski, S. Feature ranking for multi-target regression. *Mach. Learn.* **2020**, *109*, 1179–1204. [CrossRef]
28. Masmoudi, S.; Elghazel, H.; Taieb, D.; Yazar, O.; Kallel, A. A machine-learning framework for predicting multiple air pollutants' concentrations via multi-target regression and feature selection. *Sci. Total. Environ.* **2020**, *715*, 136991. [CrossRef] [PubMed]
29. Yuan, H.; Zheng, J.; Lai, L.L.; Tang, Y.Y. Sparse structural feature selection for multitarget regression. *Knowl.-Based Syst.* **2018**, *160*, 200–209. [CrossRef]
30. Zhang, S.; Yang, L.; Li, Y.; Luo, Y.; Zhu, X. *Low-Rank Feature Reduction and Sample Selection for Multi-output Regression*; Springer International Publishing: Cham, Switzerland, 2016.
31. Fan, Y.; Chen, B.; Huang, W.; Liu, J.; Weng, W.; Lan, W. Multi-label feature selection based on label correlations and feature redundancy. *Knowl.-Based Syst.* **2022**, *241*, 108256. [CrossRef]
32. Xu, J.; Liu, J.; Yin, J.; Sun, C. A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously. *Knowl.-Based Syst.* **2016**, *98*, 172–184. [CrossRef]
33. Samareh-Jahani, M.; Saberi-Movahed, F.; Eftekhari, M.; Aghamollaei, G.; Tiwari, P. Low-Redundant Unsupervised Feature Selection based on Data Structure Learning and Feature Orthogonalization. *Expert Syst. Appl.* **2024**, *240*, 122556. [CrossRef]
34. Ma, J.; Xu, F.; Rong, X. Discriminative multi-label feature selection with adaptive graph diffusion. *Pattern Recognit.* **2024**, *148*, 110154. [CrossRef]
35. Zhang, R.; Zhang, Y.; Li, X. Unsupervised feature selection via adaptive graph learning and constraint. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 1355–1362. [CrossRef]
36. Zhu, X.; Zhang, S.; Zhu, Y.; Zhu, P.; Gao, Y. Unsupervised spectral feature selection with dynamic hyper-graph learning. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 3016–3028. [CrossRef]
37. You, M.; Yuan, A.; He, D.; Li, X. Unsupervised feature selection via neural networks and self-expression with adaptive graph constraint. *Pattern Recognit.* **2023**, *135*, 109173. [CrossRef]
38. Acharya, D.B.; Zhang, H. Feature Selection and Extraction for Graph Neural Networks. In Proceedings of the 2020 ACM Southeast Conference (ACM SE '20), Tampa, FL, USA, 2–4 April 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 252–255. [CrossRef]
39. Chen, L.; Huang, J.Z. Sparse Reduced-Rank Regression for Simultaneous Dimension Reduction and Variable Selection. *J. Am. Stat. Assoc.* **2012**, *107*, 1533–1545. [CrossRef]
40. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 171–184. [CrossRef] [PubMed]
41. Doquire, G.; Verleysen, M. A graph Laplacian based approach to semi-supervised feature selection for regression problems. *Neurocomputing* **2013**, *121*, 5–13. [CrossRef]
42. He, X.; Niyogi, P. Locality preserving projections. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, USA, 8–13 December 2003; Volume 16.

43. Wang, H.; Yang, Y.; Liu, B. GMC: Graph-Based Multi-View Clustering. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1116–1129. [CrossRef]
44. Liu, J.; Ji, S.; Ye, J. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 339–348.
45. Tsoumakas, G.; Spyromitros-Xioufis, E.; Vilcek, J. MULAN: A Java library for multi-label learning. *J. Mach. Learn. Res.* **2011**, *12*, 2411–2414.
46. Nie, F.; Huang, H.; Cai, X.; Ding, C. Efficient and Robust Feature Selection via Joint $\ell_{2,1}$-Norms Minimization. In Proceedings of the Neural Information Processing Systems (NIPS), Vancouver, BC, USA, 6–9 December 2010; pp. 1813–1821.
47. Borchani, H.; Varando, G.; Bielza, C.; Larrañaga, P. A survey on multi-output regression. *WIREs Data Min. Knowl. Discov.* **2015**, *5*, 216–233. [CrossRef]
48. Sheikhpour, R.; Gharaghani, S.; Nazarshodeh, E. Sparse feature selection in multi-target modeling of carbonic anhydrase isoforms by exploiting shared information among multiple targets. *Chemom. Intell. Lab. Syst.* **2020**, *200*, 104000. [CrossRef]
49. Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; Cambridge University Press: Cambridge, UK, 2004.
50. Demšar, J.; Schuurmans, D. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.