Feature Selection via ℓ_2-Distance to Centroids

YUXUAN XIA, WEI-YUAN SU, and SIJIE KONG

Feature selection plays a crucial role in mitigating the challenges posed by high dimensionality, improving model interpretability, reducing computational costs, and enhancing predictive performance. Traditional feature selection methods—including filter, wrapper, and embedded approaches—often suffer from high computational complexity and instability, especially when dealing with large-scale multi-category data. In this work, we explore sparse centroid-based feature selection methods, which aim to minimize the distance between data samples and their corresponding class centroids while enforcing sparsity constraints. We identify key limitations in existing approaches, such as non-convex optimization and sensitivity to noise, and propose an improved linear transformation framework that mitigates these issues through convex optimization techniques. Our method achieves efficient and stable feature selection while preserving classification accuracy, making it a promising approach for handling high-dimensional datasets in modern machine learning applications.

Additional Key Words and Phrases: Feature Selection; Convex Optimization; Machine Learning.

1 Introduction

With the rise of big data and AI supercomputing, high-dimensional datasets have become central to machine learning, offering new opportunities while introducing significant challenges. This is especially true in bioinformatics, where gene expression studies involve tens or even hundreds of thousands of features per sample. Meanwhile, deep learning and transformer-based models have revolutionized predictive analytics but often lack interpretability due to their complexity. Rather than prioritizing transparency, current research trends tend to embrace complexity.

Feature selection is crucial in high-dimensional data, as only a small subset of features typically drives classification performance. It enhances model interpretability, facilitates biomarker discovery in applications like drug therapy, and improves visualization, reduces storage needs, and accelerates training.

Feature selection methods fall into three main categories: filter, wrapper, and embedded methods.

- Filter methods rank features based on divergence or correlation and apply a threshold for selection.
- Wrapper methods iteratively select or remove features using an objective function, often tied to predictive performance.
- Embedded methods integrate feature selection into model training through regularization.

However, as the volume of data continues to grow in the big data era—along with increasing numbers of model categories and features—these methods inevitably suffer from exponential computational complexity. Additionally, a key challenge in feature selection is how to reduce the number of features while maintaining high model accuracy. Therefore, developing simple, efficient, and interpretable feature selection methods for handling high-dimensional, multi-category data has become a crucial direction in current research.

2 Previous Work

Sparsity-driven feature selection methods [1] have been a prominent research focus over the past decadem, particularly in high-dimensional data scenarios such as genomic analysis, text classification, and image processing. The key idea behind these methods is to approximate the original objective function with a sparsity-inducing regularizer: ℓ_0-norm (number of non-zero elements). However, the ℓ_0-norm is not a norm, thus non-convex and computationally intractable.

Authors' Contact Information: Yuxuan Xia, yuxuanxia@ucsb.edu; Wei-Yuan Su, wei-yuansu@ucsb.edu; Sijie Kong, sijie_kong@ucsb.edu.

To address this issue, several ℓ_0-norm-like regularizers have been proposed [2], such as the ℓ_1-norm (sum of absolute values) and the $\ell_{2,1}$-norm (sum of ℓ_2-norms) [3]. These regularizers are convex and can be efficiently optimized using convex optimization techniques.

Minimizing the distance to class centroid is a reasonable heuristic for feature selection [4]. However, in SLCE, the objective function is defined as follows: $\min_{A,B} \|C - AA^T(BX)\|_F^2 + \lambda|\text{diag}(B)|_1$. It is not convex with respect to A, which can be easily proved by taking the Hessian matrix of the objective function (not always be positive semi-definite). The non-convexity of the objective function may lead to suboptimal solutions and slow convergence.

3 Methods

3.1 Problem Statement

Sparse centroid-based methods aim to select features by minimizing the distance between data samples and their corresponding class centroids, under a sparsity constraint. While conceptually appealing, existing approaches such as the Sparse Linear Centroid-Encoder (SLCE) face two critical limitations that undermine their practical effectiveness.

- **Non-convex Optimization in Linear Models**
  In the linear setting, feature selection is often formulated as learning a transformation matrix that projects input features to a space where class centroids are well separated. However, SLCE formulates this as a non-convex optimization problem involving bilinear terms. The non-convexity not only increases the risk of convergence to suboptimal solutions but also results in high computational cost and instability, especially in cases where the objective's Hessian is not positive semi-definite.
- **Centroid Sensitivity and Noise in High-Dimensional Data**
  Centroid-based methods inherently rely on the assumption that class centroids accurately represent the typical pattern of each class. In real-world high-dimensional datasets, this assumption often breaks down due to two factors: 1) Sensitivity to outliers: Extreme values can significantly skew the computed centroids; 2) Irrelevant or redundant features: Many features in high-dimensional data are uninformative or even detrimental to the learning process, yet they may still influence centroid computation, leading to poor feature selection.

These challenges motivate two complementary problem perspectives:

- **Linear Transformation:**
  In the linear case, the focus is on designing an efficient and stable optimization framework for learning sparse projections while avoiding the pitfalls of non-convexity.
- **Non-linear Transformation:**
  In the non-linear case, the challenge lies in learning richer representations that can model complex patterns in data, while remaining robust to irrelevant features and centroid estimation errors.

3.2 Linear Transformation

3.2.1 Notations and Assumptions. Denote data samples $X = \left[x_1, x_2, \cdots, x_n\right] \in \mathbb{R}^{d\times n}$, A class centroid $c_k = \frac{1}{|C_k|}\sum_{y_i=k} x_i$, our goal is to minimize the average distance between each sample and its corresponding class centroid. The objective function is defined as follows: $\min_W \|C - X^T W\|_{2,1} + \lambda\|W\|_{2,1}$, where $C = \left[c_{y_1}, c_{y_2}, \cdots, c_{y_n}\right]^T \in \mathbb{R}^{n\times d}$, $W \in \mathbb{R}^{d\times d}$. Additionally, we assume that the feature dimension is much larger than the number of samples and the number of samples is much larger than the number of classes, i.e., $d \gg n \gg k$.

Algorithm 1 Iterative Update Algorithm

Require: Initial matrix U, convergence threshold ϵ
Ensure: Optimized matrix U
1: Initialize $D \leftarrow I$
2: **repeat**
3: Update U using the specified equation
4: Update $D \leftarrow \text{diag}\left(\frac{1}{2|U_{,1}|2}, \frac{1}{2|U,2|2}, \ldots, \frac{1}{2|U*,d|_2}\right)$
5: **until** convergence criterion is met

3.2.2 Algorithm. The original objective function can be rewritten as follows: $\min_W \frac{1}{\lambda}\|C - X^T W\|_{2,1} + \|W\|_{2,1}$. Substitute $\lambda(C - X^T W)$ with Z, we have:

$$\min_{W,Z} \|Z\|_{2,1} + \|W\|_{2,1}$$
$$\text{s.t. } X^T W + \lambda Z = C$$

Let $A = \begin{bmatrix} X^T & \lambda I \end{bmatrix}$ and $U = \begin{bmatrix} W \\ Z \end{bmatrix}$. The augmented Lagrangian function [5] is defined as follows:

$$\mathbf{L}(U, \Lambda) = \|Z\|_{2,1} - \text{tr}\left(\Lambda^T (AU - C)\right)$$

where Λ is the Lagrange multiplier. The partial derivative of $\mathbf{L}(U, \Lambda)$ with respect to U is:

$$\frac{\partial \mathbf{L}(U, \Lambda)}{\partial U} = 2\text{diag}\left(\frac{1}{2\|U_{*,1}\|_2}, \frac{1}{2\|U_{*,2}\|_2}, \cdots, \frac{1}{2\|U_{*,d}\|_2}\right) U - A^T \Lambda = 0$$

where $U_{*,i}$ denotes the i-th column of U. Left-multiplying both sides by $A\text{diag}\left(2\|U_{*,1}\|_2, 2\|U_{*,2}\|_2, \cdots, 2\|U_{*,d}\|_2\right)$, we have:

$$2C - A\text{diag}\left(2\|U_{*,1}\|_2, 2\|U_{*,2}\|_2, \cdots, 2\|U_{*,d}\|_2\right) A^T \Lambda = 0$$

and

$$\Lambda = 2\left(A\text{diag}\left(2\|U_{*,1}\|_2, 2\|U_{*,2}\|_2, \cdots, 2\|U_{*,d}\|_2\right) A^T\right)^{-1} C.$$

Substitute Λ back to the original equation, we have:

$$U = \text{diag}\left(2\|U_{*,1}\|_2, 2\|U_{*,2}\|_2, \cdots, 2\|U_{*,d}\|_2\right) A^T \left(A\text{diag}\left(2\|U_{*,1}\|_2, 2\|U_{*,2}\|_2, \cdots, 2\|U_{*,d}\|_2\right) A^T\right)^{-1} C.$$

This is an equation of U and can be solved by the following alternating optimization method [3] 1.

Consider the update function of U: $U = D^{-1}A^T(AD^{-1}A^T)^{-1}C$, C can be decomposed as follows: $C = \begin{bmatrix} c_{y_1}, c_{y_2}, \cdots, c_{y_n} \end{bmatrix}^T = B\tilde{C}$, where B is the one-hot encoding matrix: $B = \begin{bmatrix} b_1, b_2, \cdots, b_n \end{bmatrix}^T \in \mathbb{R}^{n \times k}$, b_i is the one-hot encoding vector of y_i and $\tilde{C} = \begin{bmatrix} c_1, c_2, \cdots, c_k \end{bmatrix}^T \in \mathbb{R}^{k \times d}$. Substitute C with $B\tilde{C}$, we have:

$$U = D^{-1}A^T(AD^{-1}A^T)^{-1}B\tilde{C}.$$

Consider $E = (AD^{-1}A^T)^{-1}B$, we can apply the conjugate gradient method [6] to solve the equation $(AD^{-1}A^T)E_{*,i} = B_{*,i}$ for $i = 1, 2, \cdots, k$. The process of this method is 2.

Algorithm 2 Iterative Conjugate Gradient-Like Algorithm

Require: Matrix A, diagonal matrix D, right-hand side B, iteration count k, convergence threshold ϵ
Ensure: Approximate solution E

Initialize $E \leftarrow B$
Initialize $R \leftarrow B - (AD^{-1}A^T)E$, $P \leftarrow R$
repeat
  for $i = 1$ to k **do**
    $\alpha \leftarrow \frac{R^T R}{P^T (AD^{-1}A^T)P}$
    $E \leftarrow E + \alpha P$
    $R_{\text{old}} \leftarrow R$
    $R \leftarrow R - \alpha(AD^{-1}A^T)P$
    $\beta \leftarrow \frac{R^T R}{R_{\text{old}}^T R_{\text{old}}}$
    $P \leftarrow R + \beta P$
  end for
until convergence criterion is met

3.3 Non-linear Transformation

To extend the linear approach to nonlinear settings, neural networks can capture more complex feature relationships. In non-linear setting, a multi-layer perceptron (MLP) is used to approximate the non-linear transformation of the feature space [7]. The hidden layers(A) of the MLP learn complex representations that better separate classes in high-dimensional space. To enforce sparsity in feature selection, a sparse layer(B) is introduced, which applies regularization techniques like L1 norm. We also tried 2 different method to improve the performance, Alternating Optimization(AO) and RSCE. The objective function is as follows:

$$\mathcal{L}_{sce}(\theta) = \frac{1}{2N} \sum_{j=1}^{M} \sum_{i \in I_j} \|c_j - f(x^i; \theta)\|_2^2 + \lambda \|\theta_{spl}\|_1$$

3.3.1 Alternating Optimization. In the original method, the hidden layers(A) and the sparse layer(B) were optimized separately only once. But this cannot guarantee the optimal solution. Optimizing them seperately limits the potential for jointly improving feature selection and classification performance. AO provides a more refined solution by iteratively optimizing both A and B, leading to better convergence and enhanced feature selection. The problem is divided into two subproblems:

1. Optimization of A: A neural network transformation learns a better feature representation.

2. Optimization of B: The sparse encoding matrix is refined to enforce sparsity constraints. By optimizing A and enforcing sparsity in B iteratively, the model achieves a more stable and effective feature selection mechanism.

3.3.2 Relief and RSCE. Centroid-based methods are usually based on the assumption that centroid can correctly reflect the common pattern of a certain class. However, centroids may not always represent meaningful values due to: **1. Extreme Values:** In real-life datasets, some outliers may appear due to measurement errors and other reasons. Extreme outliers will cause the centroid of a certain class to deviate from the correct pattern, which will lead to lower accuracy of the selected features in the model. **2. Useless Features:** It is found that the sparse centroid method sometimes retains redundant features, which do not contribute to the classification of the model or even have side effects, and in subsequent classification models, the accuracy after feature selection will be reduced. Irrelevant features may interfere with optimization and affect feature selection quality. To mitigate these issues, the filter method can initially screen

features before the sparse centroid method. Specifically, we used the Relief-F algorithm [8]. The results show that the accuracy of the RSCE method is significantly higher than the sparse-centroid method and the separate Relief-F method. The importance matrix is as follows:

$$\delta^j = \sum_i -\text{diff}(x_i^j, x_{i,nh}^j)^2 + \sum_{l \neq k} \left(p_l \times \text{diff}(x_i^j, x_{i,im}^j)^2 \right)$$

We explore two methods for feature selection and classification: Relief and RSCE. Relief is a filter-based algorithm that ranks features according to their ability to distinguish between instances of different classes. It does so by randomly sampling training instances and comparing each one with its nearest neighbors from the same and opposite classes. Feature relevance scores are updated based on how effectively a feature separates similar and dissimilar samples. The top-ranked features are then used to train a neural network classifier, and classification performance is evaluated accordingly.

In contrast, RSCE (Relief followed by Sparse Coding-based Embedding) is a two-stage hybrid approach. Initially, Relief is applied as a coarse filtering step to retain the top 1000 potentially informative features. These features are subsequently refined using a Sparse Coding-based Embedding (SCE) model, which integrates an L1-regularized sparse layer into a shallow neural network. This model undergoes both pretraining and posttraining phases, enabling it to learn both a compact embedding and a sparse set of discriminative features. The final feature importance scores are extracted from the trained model, and the most relevant features are selected using the elbow method. These features are then evaluated using the same classifier, and accuracy is reported across varying feature counts.

Algorithm 3 Original SCE

Require: Training data $\mathbf{X} \in \mathbb{R}^{n \times d}$, labels $\mathbf{y} \in \{1, \ldots, C\}^n$, network config, hyperparameters, number of selected features k

Ensure: Indices of top-k features

Pre-Training Initialization
Standardize data: $\tilde{\mathbf{X}} = \dfrac{\mathbf{X} - \boldsymbol{\mu}}{\boldsymbol{\sigma}}$, where $\mu_j = \dfrac{1}{n}\sum_{i=1}^{n} X_{ij}$, $\sigma_j^2 = \dfrac{1}{n}\sum_{i=1}^{n}(X_{ij} - \mu_j)^2$
Construct centroid targets: $\mathbf{T}_{i,:} = \dfrac{1}{|C_c|}\sum_{j \in C_c} \tilde{\mathbf{X}}_j$ for $i \in C_c$, where $C_c = \{i : y_i = c\}$
Layer-Wise Supervised Pre-Training
for $d = 1$ to L (number of hidden layers) **do**
    Freeze layers 1 to $d - 1$
    for $e = 1$ to $N_{\text{epochs}}^{\text{pre}}$ **do**
        for all mini-batches $(\tilde{\mathbf{X}}_b, \mathbf{T}_b)$ **do**
            Forward pass: $\hat{\mathbf{T}}_b = f^{(d)} \circ \cdots \circ f^{(1)}(\tilde{\mathbf{X}}_b)$
            Loss: $\mathcal{L}_{\text{pre}} = \dfrac{1}{|b|}\sum_{i \in b} \|\hat{\mathbf{T}}_i - \mathbf{T}_i\|_2^2$
            Update weights of layer d: $\mathbf{W}^{(d)} \leftarrow \mathbf{W}^{(d)} - \eta \nabla_{\mathbf{W}^{(d)}} \mathcal{L}_{\text{pre}}$
        end for
    end for
end for
Post-Training Initialization
Insert sparse layer: $\mathbf{Z} = \tilde{\mathbf{X}} \cdot \mathbf{W}_{\text{SPL}}, \quad \mathbf{W}_{\text{SPL}} = \text{diag}(1, \ldots, 1)$
Full model: $\hat{\mathbf{T}} = f(\tilde{\mathbf{X}} \cdot \mathbf{W}_{\text{SPL}})$, using pretrained weights in f
Train all layers jointly (without L1 regularization) for stabilization
Post-Training with Feature Selection
for $e = 1$ to $N_{\text{epochs}}^{\text{post}}$ **do**
    for all mini-batches $(\tilde{\mathbf{X}}_b, \mathbf{T}_b)$ **do**
        Forward: $\hat{\mathbf{T}}_b = f(\tilde{\mathbf{X}}_b \cdot \mathbf{W}_{\text{SPL}})$
        Loss with L1: $\mathcal{L}_{\text{post}} = \dfrac{1}{|b|}\sum_{i \in b} \|\hat{\mathbf{T}}_i - \mathbf{T}_i\|_2^2 + \lambda \|\mathbf{W}_{\text{SPL}}\|_1$
        Update: $\mathbf{W}_{\text{SPL}} \leftarrow \mathbf{W}_{\text{SPL}} - \eta \nabla_{\mathbf{W}_{\text{SPL}}} \mathcal{L}_{\text{post}}$
    end for
end for
Feature Ranking
Compute feature importance: $\text{importance}_j = |W_{\text{SPL},jj}|$
return Indices of top-k features

4 Results and Analysis

In order to evaluate the performance of our method and compare it with other methods, e.g., RFS [3], SLCE [4]. Our evaluation performance is based on the both the computational time and the average distance to class centroid of the selected features. We will use the following datasets [9] for evaluation.

4.1 Datasets

4.1.1 ALLAML. (continuous, binary, 72 samples, 7129 features): The ALLAML dataset is a well-known gene expression dataset used for leukemia classification. It consists of microarray expression profiles from patients diagnosed with either acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML).

4.1.2 GLIOMA. (continuous, 4 classes, 50 samples, 22283 features): The GLIOMA dataset is a gene expression dataset used for the classification and analysis of gliomas, a type of brain tumor originating in glial cells. This dataset is commonly used in bioinformatics and machine learning research for tumor subtype classification, biomarker discovery, and survival analysis.

4.1.3 lung. (continuous, 5 classes, 203 samples, 3312 features): The lung dataset is a gene expression dataset used for the classification of lung cancer subtypes. It consists of microarray expression profiles from patients diagnosed with either adenocarcinoma or squamous cell carcinoma of the lung.

4.2 Linear Transformation

Our method reaches almost the same performance as RFS in terms of the average distance to class centroid of the selected features. However, our method is much faster than SLCE.

4.2.1 Computational Time. Table 1 presents the computational time (in seconds) for different datasets.

Table 1. Computational time (in seconds)

Dataset	RFS	SLCE	Ours
ALLAML	177.07	402.32	157.72
GLIOMA	68.59	312.45	61.94
lung	45.23	201.34	39.87

4.2.2 Average Distance to Class Centroid. Table 2 shows the average distance to the class centroid for the top 20 features.

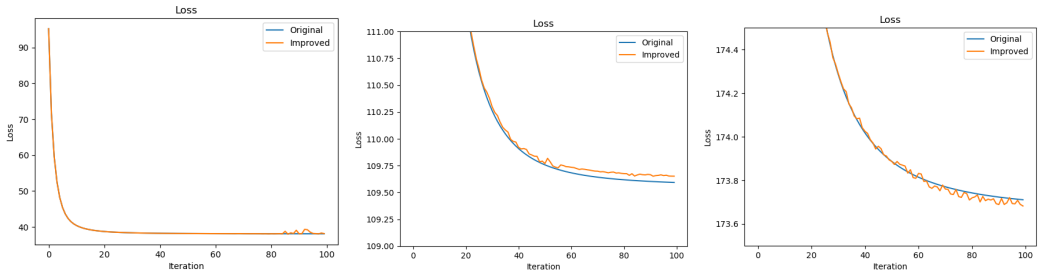
Table 2. Average distance to class centroid (top 20 features)

Dataset	RFS	SLCE	Ours
ALLAML	1.95	2.09	1.95
GLIOMA	17.67	30.12	17.69
lung	8.82	9.01	8.81

4.2.3 Convergence Performance. Our method also converges almost the same as RFS and in some datasets, faster than it. The convergence curve is shown below:

4.3 Analysis for Linear Transformation

4.3.1 Convexity. The objective function $\|C - X^T W\|_{2,1} + \lambda \|W\|_{2,1}$ is convex with respect to W. $C - W^T X$ is an affine transformation of W, and $\|C - X^T W\|_{2,1}$ is the sum of the norms of the columns of $C - X^T W$. The norm of a vector is a convex function, and the sum of convex functions is also convex [5]. $\|W\|_{2,1}$ is the sum of the norms of the columns of W, which is also convex. Therefore, the objective function is convex.



(a) ALLAML (b) GLIOMA (c) lung

Fig. 1. Convergence Curves

4.3.2 Convergence. The convergence of the alternating optimization method can be found in [3]. The convergence of the conjugate gradient method is guaranteed if the matrix is symmetric positive semi-definite [6]. The matrix $AD^{-1}A^T$ is symmetric: $(AD^{-1}A^T)^T = A^T(D^{-1})^TA = AD^{-1}A^T$. It is also positive semi-definite: $x^TAD^{-1}A^Tx = (Ax)^TD^{-1}(Ax) \geq 0$. Therefore, $AD^{-1}A^T \in \mathbb{S}_+^n$ and the conjugate gradient method converges.

4.3.3 Complexity. We count the number of additions and multiplications in each iteration of our method. First, computing D^{-1} requires d multiplications, computing $AD^{-1}A^T$ requires $O(n^2(n+d))$ multiplications and $O(n^2(n+d))$ additions. Second, applying k times of the conjugate gradient method to solve $(AD^{-1}A^T)E_{*,i} = B_{*,i}$ requires $O(\sqrt{\kappa}n^2k)$ multiplications and $O(\sqrt{\kappa}n^2k)$ additions, where κ is the condition number $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ of the matrix $AD^{-1}A^T$. Therefore, the total complexity of our method is $O((n+d)(n^2+dk))$. If we compute the inverse of the matrix $AD^{-1}A^T$ directly and use C instead of $B\tilde{C}$, the complexity is $O((n+d)^3)$. Note that d is much larger than n and k and $d^3 \gg d^2k$.

4.4 Result for Non-linear Transformation

In Section *Non-linear Transformation*, we discussed the phenomenon where centroids fail to accurately reflect class patterns due to extreme values and irrelevant features. We now address these two issues by incorporating standard deviation and Relief-based weights.

4.4.1 Analysis on Dataset. We analyzed the variance of each feature across the two classes in the ALLAML dataset, as illustrated in Figure 2. The distribution of feature standard deviations in the ALLAML dataset exhibits significant heterogeneity, which may adversely affect the use of centroids to represent data patterns. As shown in the figure, the dataset comprises approximately 7,000 features, with the standard deviation of most features concentrated between 0 and 1. However, a small subset of features exhibits standard deviations approaching 2, indicating substantial numerical variability within both Class 0 and Class 1. Such elevated standard deviations suggest the presence of numerous extreme values, particularly in features with standard deviations nearing 2, where the numerical differences between samples can be highly pronounced. As the centroid represents the average of within-class samples, it is highly susceptible to the influence of these extreme values, causing it to deviate from the central tendency of the majority of the data and thus failing to accurately capture the overall class pattern. Furthermore, the standard deviation distributions of features in Class 0 and Class 1 are remarkably similar, indicating comparable levels of dispersion between the two classes. This similarity implies that the impact of extreme values is present in both classes, further complicating the centroid's ability

to distinguish inter-class patterns. Consequently, the prevalence of extreme values in the dataset may interfere with the centroid's capacity to effectively reflect the underlying data patterns.

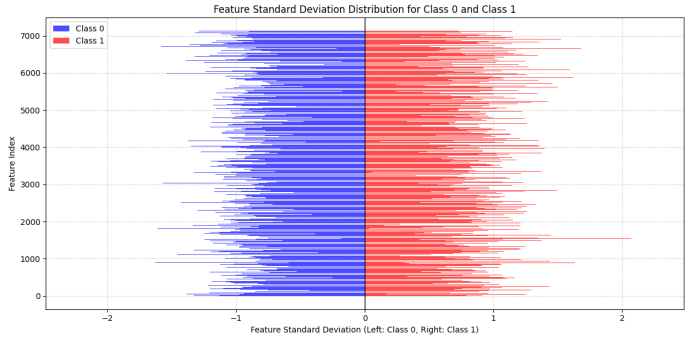


Fig. 2. Standard Variation of Features in Dataset

We applied the Relief algorithm to the two classes (ALL and AML) in the ALLAML dataset to analyze the association between features and inter-class differences, with the results presented in Figure 3. In the figure, the x-axis represents the feature index (ranging from 0 to approximately 7,000), while the y-axis denotes the weights (ranging from 0 to -8). Weights closer to 0 indicate that a feature is more significant in contributing to inter-class differences, whereas larger absolute negative values suggest that a feature is more likely to disrupt classification.

As observed in Figure 3, the majority of feature weights are concentrated near 0, indicating that many features play a crucial role in distinguishing between the ALL and AML classes. This reflects the high-dimensional nature of gene expression data. However, a subset of features exhibits weights as low as -8, with a relatively sparse distribution. These features are likely to represent noise or irrelevant attributes, potentially impairing classification performance. Specifically, features with weights closer to 0 are more effective for classification, yet the figure reveals that numerous features have large absolute weight values, signifying extremely low relevance. This observation aligns with the limitations of the centroid method that irrelevant features may obscure true class patterns. In the process of promoting sparsity through the centroid model, retaining features with minimal relevance while discarding those with greater relevance could reduce the alignment between the sparse selection outcomes and the classification task, ultimately compromising model performance.

4.4.2 Accuracy on Classification Task. We also evaluated the performance of features selected by SCE, AO-SCE, Relief, and RSCE in classification tasks, as illustrated in Figure 4. Initially, we applied various feature selection algorithms to identify the top 1 to 100 features. Subsequently, these selected features were used as input for classification tasks to assess the effectiveness of each algorithm. The datasets employed in this analysis include ALLAML (a straightforward binary classification dataset), GLIOMA (a multiclass dataset with an exceptionally large number of features), and Lung (a multiclass dataset with a particularly high number of samples).

As observed in Figure 4.2.3, for the relatively simple ALLAML dataset, RSCE demonstrates the best performance in terms of both convergence speed and accuracy, rapidly achieving a classification accuracy of approximately 95% with only 20 features and maintaining stability thereafter. AO-SCE follows as the second-best, with a slightly slower convergence speed and an accuracy that fluctuates between 85% and 90% after 20 features. In contrast, Relief and SCE

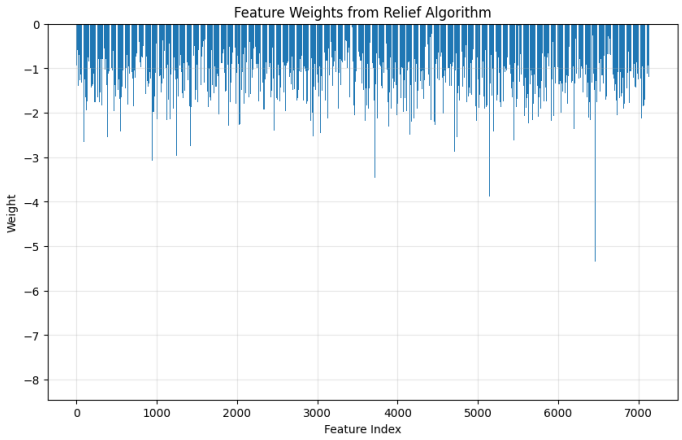
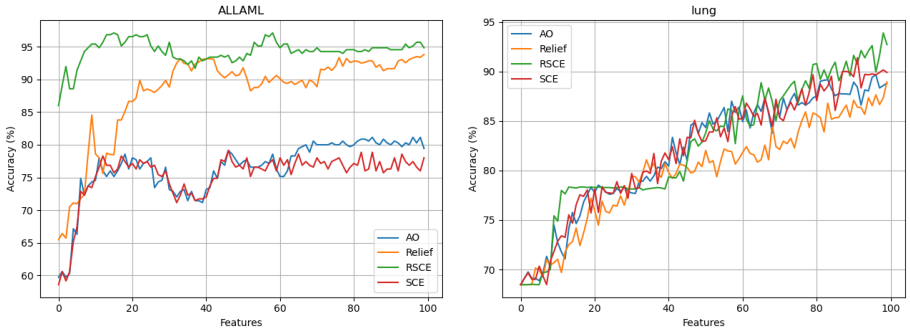


Fig. 3. Feature Weights from Relief Algorithm on ALLAML

exhibit poorer performance, with slow convergence speeds and accuracies that consistently fluctuate between 70% and 80%, struggling to improve.

For the more complex Lung dataset, RSCE again converges the fastest, reaching an accuracy of approximately 90% with just 60 features and maintaining stability, highlighting its superior performance on multiclass datasets with large sample sizes. AO-SCE, SCE, and Relief, however, converge more slowly, requiring nearly 100 features to approach accuracies of 90% and 88%, with considerable fluctuations, indicating their lower efficiency in handling large sample sizes. The strong performance of RSCE underscores its greater adaptability in scenarios with large sample sizes and complex classification tasks.



(a) ALLAML (b) lung

Fig. 4. Classification Accuracy for Four Non-linear Feature Selection

The issue with *Relief* lies in the presence of redundant features, such as features A and B being equivalent to feature C, which *Relief* fails to mitigate through feature selection. Meanwhile, *SCE* is more sensitive to extreme values and irrelevant features, further limiting its effectiveness.

5 Conclusions and Limitations

We proposed an improved sparse centroid-based feature selection framework that addresses key challenges in high-dimensional data. By reformulating the objective using convex optimization, our method enhances stability, reduces sensitivity to noise, and preserves classification performance, all while enforcing sparsity constraints. This makes our approach particularly suitable for large-scale, multi-category datasets where interpretability and efficiency are crucial.

In the linear case, we demonstrated that our convex formulation achieves competitive centroid alignment while significantly reducing computation time compared to non-convex alternatives such as SLCE. In the nonlinear setting, we showed that integrating Relief-based filtering with sparse centroid-based embedding (RSCE) substantially improves feature selection robustness and downstream classification accuracy, particularly in complex datasets with extreme values or redundant features.

Despite these improvements, our method has several limitations. First, centroid estimation can still be unreliable in imbalanced datasets, which may affect selection quality. Second, performance is sensitive to the choice of regularization parameters, requiring careful tuning. Finally, while our approach handles linear and shallow nonlinear transformations effectively, its extension to deeper models or unsupervised tasks remains an open direction.

For future work, we can investigate adaptive centroid estimation under imbalance, extend our framework to semi-supervised and unsupervised learning, and explore integration with transformer-based architectures for feature selection in text and sequence data.

References

[1] Chandrashekar, Girish, and Ferat Sahin, *A survey on feature selection methods*, Computers & electrical engineering 40.1 (2014): 16-28.

[2] Bradley, Paul S., and Olvi L. Mangasarian, *Feature selection via concave minimization and support vector machines*, ICML. Vol. 98. 1998.

[3] Nie, Feiping, et al, *Efficient and robust feature selection via joint L2, 1-norms minimization*, Advances in neural information processing systems 23 (2010).

[4] Ghosh, Tomojit, Karim Karimov, and Michael Kirby, *Sparse linear centroid-encoder: A biomarker selection tool for high dimensional biological data*, 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2023.

[5] Boyd, Stephen P., and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[6] Shewchuk, Jonathan Richard, *An introduction to the conjugate gradient method without the agonizing pain*, (1994): 7.

[7] Makhzani, A., Frey, B. J., and Karklin, Y. 2013. Nonlinear feature selection using sparsity-promoted centroid-encoder. *In Proceedings of the 2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP'13)*. IEEE, 1–6. Available at: https://www.researchgate.net/publication/373296941_Nonlinear_feature_selection_using_sparsity-promoted_centroid-encoder.

[8] Koller, D. and Sahami, M. 1996. The feature selection problem: Traditional methods and a new algorithm. *In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96)*. AAAI Press, 129–134. Available at: https://cdn.aaai.org/AAAI/1992/AAAI92-020.pdf.

[9] https://jundongl.github.io/scikit-feature/datasets.html