A Generalized Geometric Theory of Centroid Discriminant Analysis for Linear Classification of Multi-dimensional Data

Yue Wu , Jialin Zhao , Carlo Vittorio Cannistraci *

Posted Date: 27 May 2025

doi: 10.20944/preprints202502.0789.v2

Keywords: linear classification; efficient and scalable algorithm; geometric discriminant analysis; centroid discriminant analysis; nonlinear classification via kernel method

 doi:10.20944/preprints202502.0789.v2

Article

A Generalized Geometric Theory of Centroid Discriminant Analysis for Linear Classification of Multi-dimensional Data

Yue Wu [1,3], Jialin Zhao [1,2] and Carlo Vittorio Cannistraci [1,2,3,*]

[1] Center for Complex Network Intelligence (CCNI), Tsinghua Laboratory of Brain and Intelligence (THBI), Department of Psychological and Cognitive Sciences, Tsinghua University, Beijing, China
[2] Department of Computer Science, Tsinghua University, Beijing, China
[3] Department of Biomedical Engineering, Tsinghua University, Beijing, China
* Correspondence: kalokagathos.agon@gmail.com

Abstract: With the advent of the neural network era, traditional machine learning methods have increasingly been overshadowed. Nevertheless, continuing to research about the role of geometry for learning in data science is crucial to envision and understand new principles behind the design of efficient machine learning. Nonlinear classifiers often build upon linear ones by leveraging shared underlying properties, with their performance largely dependent on the effectiveness of the foundational linear model. Linear classifiers are favored in certain tasks due to their reduced susceptibility to overfitting and their ability to provide interpretable decision boundaries. In biomedical data science, employing an efficient linear classifier is often the first step in assessing the intrinsic complexity of a dataset. However, achieving both scalability and high predictive performance in linear classification remains a persistent challenge. Here, we propose a theoretical framework named geometric discriminant analysis (GDA). GDA includes the family of linear classifiers that can be expressed as function of a centroid discriminant basis (CDB0) - the connection line between two centroids - adjusted by geometric corrections under different constraints. We demonstrate that linear discriminant analysis (LDA) is a subcase of the GDA theory, and we show its convergence to CDB0 under certain conditions. Then, based on the GDA framework, we propose an efficient linear classifier named centroid discriminant analysis (CDA) which is defined as a special case of GDA under a two-dimensional (2D) plane geometric constraint. CDA training is initialized starting from CDB0 and involves the iterative calculation of new adjusted centroid discriminant lines whose optimal rotations on the associated 2D planes are searched via Bayesian optimization. CDA has good scalability (quadratic time complexity) which is lower than LDA and support vectors machine (SVM) (cubic complexity). Results on 27 real datasets across classification tasks of standard images, medical images and chemical properties, offer empirical evidence that CDA outperforms other linear methods such as LDA, SVM and fast SVM in terms of scalability, performance and stability. Furthermore, we show that linear CDA can be generalized to nonlinear CDA via kernel method, demonstrating improvements on the linear version with tests on two challenging datasets in tasks such as classifications of images and chemical data. GDA theory may inspire the design of new linear and nonlinear classifiers under the definition of different geometric constraints. GDA general validity as a new theory for designing machine learning can pave the way towards more deeper understanding of the role of geometry in learning from data.

Keywords: linear classification; efficient and scalable algorithm; geometric discriminant analysis; centroid discriminant analysis; nonlinear classification via kernel method

1. Introduction

Linear classifiers are often favored over nonlinear models, such as neural networks, for certain tasks due to their comparable performance in high-dimensional data spaces, faster training speeds, reduced tendency to overfit, and greater interpretability in decision-making ([1,2]). Notably, linear

 doi:10.20944/preprints202502.0789.v2

classifiers have demonstrated performance on par with convolutional neural networks (CNNs) in medical classification tasks, such as predicting Alzheimer's disease from structural or functional brain MRI images ([3,4]).

These linear classifiers can be categorized into several types based on the principles they use to define the decision boundary or classification discriminant, as described below, where N is the number of samples, and M is the number of features:

- The nearest mean classifier (NMC) ([5]) which is a prototype-based classifier that assigns points according to the perpendicular bisector boundary between the centroids of two groups. This classifier has $O(NM)$ training time complexity.
- Fisher's linear discriminant analysis (LDA, specifically refer to Fisher's LDA in this study) is a variance-based classifier which can be trained in cubic time complexity $O(NM^2 + M^3)$. While faster implementations like spectral regression discriminant analysis (SRDA) ([6]) claim lower training time complexity, their efficiency depends on specific conditions, such as a sufficiently small iterative term and sparsity in the data. These constraints limit SRDA's applicability in real-world classification tasks.
- Support vector machine (SVM) ([7]) with a linear kernel is a maximum-margin classifier, which has a training time complexity of $O(N^3)$. A fast implementation LIBLINEAR ([8]) (denoted as fast SVM) reduces to $O(kNM)$ by using dual coordinate descent optimization. The iteration count k that depends on the optimization path leads to a quasi-quadratic overall complexity.
- Perceptron ([9]) is a misclassification-triggered ruled-based classifier. Its training time complexity is $O(kNM)$.
- Logistic regression (LR) ([10]) is a statistics-based classifier. It can be trained using either maximum likelihood estimation (MLE) or iteratively reweighted least squares, with time complexity of $O(NM^2 + M^3)$ and $O(NM^2)$ respectively.

Among linear classifiers, NMC offers the lowest training time complexity but suffers from limited performance due to its overly simplified decision boundary. Widely used methods such as LDA and SVM are often favored for their strong predictive capabilities. However, these methods can be computationally expensive, particularly for large-scale datasets. Hence, achieving both high scalability and strong predictive performance remains a challenging tradeoff, highlighting the need for new approaches that balance these competing demands.

To address this challenge, this paper makes the following **3 key contributions**:

- **A geometric theory framework for classifiers:** This study introduces a geometric framework, geometric discriminant analysis (GDA), to unify certain linear classifiers under a common theoretical model. GDA leverages a special type of centroid discriminant basis (CDB0), a vector connecting the centroids of two classes, which serves as the foundation for constructing classifier decision boundaries. The GDA framework adjusts the CDB0 through geometric corrections under various constraints, enabling the derivation of classifiers with desirable properties. Notably, we show that: NMC is a special case of GDA, where geometric corrections are not applied to CDB0; linear discriminant analysis (LDA) is a special case of GDA, where the CDB0 is corrected by maximizing the projection variance ratio.
- **A high-performance and scalable linear geometric classifier:** Building on the GDA framework, we propose centroid discriminant analysis (CDA), a novel geometric classifier that iteratively adjusts the CDB through performance-dependent rotations on two-dimensional planes. These rotations are optimized via Bayesian optimization, enhancing the decision boundary's adaptability while maintaining scalability. CDA exhibits quadratic training time complexity, outperforming LDA and SVM in terms of scalability and efficiency. Experimental evaluations on 27 real-world datasets, including standard image classification, medical imaging, and chemical property prediction, reveal that CDA consistently outperforms traditional linear methods such as LDA, SVM, and fast SVM in predictive performance, scalability, and stability.

 doi:10.20944/preprints202502.0789.v2

- **Nonlinear geometric classification via kernel method:** For complex data where linear models are not enough, CDA supports nonlinear classification via kernel method. We demonstrated with challenging image and chemical datasets that kernel CDA improved over linear CDA and outperformed kernel SVM. Nonetheless, while kernel CDA offers greater expressiveness and improved capability, linear CDA remains highly valuable for real-world tasks due to its superior training efficiency, interpretability, and reduced risk of overfitting.

More importantly, we emphasize that CDA not only achieves robust predictive performance but also offers superior computational efficiency. Unlike traditional methods such as LDA and SVM, which typically exhibit cubic time complexity, CDA operates with quadratic complexity in the worst case, resulting in significantly faster runtimes in practice. These advantages make CDA particularly attractive for real-world applications, where scalability, interpretability, and efficiency are essential. As linear classifiers remain widely used across scientific domains for their transparency and speed, CDA represents a valuable advancement for practitioners seeking reliable and computationally lightweight solutions. Lastly, the GDA theory, from which CDA is derived, may inspire new classifiers under the definition of different geometric constraints.

2. Geometric Discriminant Analysis (GDA)

In this study, we propose a generalized geometric theory for centroid-based linear classifiers. In geometry, the generalized definition of centroid is the weighted average of points. For binary classification problem, training a linear classifier involves finding a discriminant (perpendicular to the decision boundary) and a bias. In GDA, we focus on the centroid discriminant basis (CDB) which is defined as the directional vector from the centroid of negative class to that of positive class. Specifically, we focus on a particular discriminant termed as CDB0, which is constructed from centroids with uniform sample weights. GDA theory incorporates all the classifiers whose classification discriminant is CDB0 adjusted by geometric corrections on CDB0, which is described in details in the following using an instance with LDA.

Without loss of generality, assume a two-dimensional space (see Appendix. D for any-dimensional proofs). We derive the GDA theory as a generalization of LDA and includes it under a certain geometric constraint. Indeed, Fisher's LDA is a type of LDA without assuming normal distribution or equal class covariance for variables. In Fisher's LDA, the linear discriminant (LD) is derived as the maximization of between-class variance to within-class variance:

$$S = \frac{\sigma_{\mathrm{b}}^2}{\sigma_{\mathrm{w}}^2} = \frac{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0)^2}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}} = \frac{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0)^2}{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)\boldsymbol{w}} \tag{1}$$

where $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ are the means of negative and positive class, $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ are their covariance matrices, $\boldsymbol{w}$ is the projection coefficient. The maximum is obtained when $N\boldsymbol{\Sigma}\boldsymbol{w}_{\mathrm{LD}} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ ([11]):

where $\boldsymbol{w}_{\mathrm{LD}}$ is the solution of LDA and N is the total number of samples. $\boldsymbol{\Sigma}$ is the sum of within-class covariance matrices of each class $\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1$:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma 0_{xx}^2 & \sigma 0_{xy}^2 \\ \sigma 0_{yx}^2 & \sigma 0_{yy}^2 \end{bmatrix} + \begin{bmatrix} \sigma 1_{xx}^2 & \sigma 1_{xy}^2 \\ \sigma 1_{yx}^2 & \sigma 1_{yy}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 \end{bmatrix} \tag{2}$$

Suppose $\boldsymbol{\Sigma}$ has full rank, then $\boldsymbol{w}_{\mathrm{LD}} = \frac{1}{N}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \propto \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$. The inverse of the covariance matrix $\boldsymbol{\Sigma}$ is:

$$\boldsymbol{\Sigma}^{-1} = \frac{\mathrm{adj}(\boldsymbol{\Sigma})}{|\boldsymbol{\Sigma}|} \propto \begin{bmatrix} \sigma_{yy}^2 & -\sigma_{xy}^2 \\ -\sigma_{yx}^2 & \sigma_{xx}^2 \end{bmatrix} \propto \begin{bmatrix} 1 & -\frac{\sigma_{xy}^2}{\sigma_{yy}^2} \\ -\frac{\sigma_{yx}^2}{\sigma_{yy}^2} & \frac{\sigma_{xx}^2}{\sigma_{yy}^2} \end{bmatrix} \tag{3}$$

 doi:10.20944/preprints202502.0789.v2

Where adj($\mathbf{\Sigma}$) is the adjugate of $\mathbf{\Sigma}$. Considering that $|\mathbf{\Sigma}|$ is a scalar, we can neglect it and account only for the matrix part. Let $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0 = [\Delta\mu_x, \Delta\mu_y]^{\mathrm{T}}$, then:

$$\boldsymbol{w}_{\mathrm{LD}} \propto \begin{bmatrix} 1 & -\frac{\sigma_{xy}^2}{\sigma_{yy}^2} \\ -\frac{\sigma_{yx}^2}{\sigma_{yy}^2} & \frac{\sigma_{xx}^2}{\sigma_{yy}^2} \end{bmatrix} \begin{bmatrix} \Delta\mu_x \\ \Delta\mu_y \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -\frac{\sigma_{xy}^2}{\sigma_{yy}^2} \\ -\frac{\sigma_{yx}^2}{\sigma_{yy}^2} & \frac{\sigma_{xx}^2}{\sigma_{yy}^2} - 1 \end{bmatrix} \right) \begin{bmatrix} \Delta\mu_x \\ \Delta\mu_y \end{bmatrix}$$

$$= \boldsymbol{w}_{\mathrm{LD}} \propto [\Delta\mu_x, \Delta\mu_y]^{\mathrm{T}} + \boldsymbol{C}_{\mathrm{correction}} [\Delta\mu_x, \Delta\mu_y]^{\mathrm{T}} \tag{4}$$

where $\boldsymbol{C}_{\mathrm{correction}}$ is a correction matrix that acts as the second-order term associated with the sum of covariance matrices. Since $\boldsymbol{w}_{\mathrm{CDB0}}$ is constructed from centroids with uniform sample weights (i.e., arithmetic means), it can be written as $\boldsymbol{w}_{\mathrm{CDB0}} = [\Delta\mu_x, \Delta\mu_y]^{\mathrm{T}} / \|[\Delta\mu_x, \Delta\mu_y]\|$. As in the covariance matrix, $\sigma_{xy}^2 = \sigma_{yx}^2$, and let $c_{xy} = \sigma_{xy}^2/\sigma_{yy}^2$, $c_{xx/yy} = \sigma_{xx}^2/\sigma_{yy}^2 - 1$, the linear discriminant in Eq. 4 can be compressed into the following general form (Fig. A1a-c, general case):

$$\boldsymbol{w}_{\mathrm{GD}} = \boldsymbol{w}_{\mathrm{LD}} \propto \begin{bmatrix} \Delta\mu_x \\ \Delta\mu_y \end{bmatrix} + \begin{bmatrix} 0 & -c_{xy} \\ -c_{xy} & c_{xx/yy} \end{bmatrix} \begin{bmatrix} \Delta\mu_x \\ \Delta\mu_y \end{bmatrix} \propto \boldsymbol{w}_{\mathrm{CDB0}} + C_{\mathrm{correction}} \boldsymbol{w}_{\mathrm{CDB0}}. \tag{5}$$

Since from Eq. (4), $\boldsymbol{w}_{\mathrm{LD}}$ can be decomposed into the basis $\boldsymbol{w}_{\mathrm{CDB0}}$ and a correction on the basis, we write out $\boldsymbol{w}_{\mathrm{GD}}$ to indicate that $\boldsymbol{w}_{\mathrm{LD}}$ is a geometrical discriminant (GD), a discriminant geometrically modified from $\boldsymbol{w}_{\mathrm{CDB0}}$. This geometrical modification can be intuitively interpreted as performing rotations on $\boldsymbol{w}_{\mathrm{CDB0}}$.

Starting from Eq. (4), we have the following special cases to consider, which represent different forms of geometrical modification applied to $\boldsymbol{w}_{\mathrm{CDB0}}$:

Special case 1. If we assume two variables have the same variance (In experiments, we relax the condition to have similar variance, i.e., $\sigma_{xx}^2 \approx \sigma_{yy}^2$), then $c_{xx/yy} = 0$. Eq. (4) becomes (Fig. A1, special case 1):

$$\boldsymbol{w}_{\mathrm{LD}} \propto \begin{bmatrix} \Delta\mu_x \\ \Delta\mu_y \end{bmatrix} + \begin{bmatrix} 0 & -c_{xy} \\ -c_{xy} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mu_x \\ \Delta\mu_y \end{bmatrix} \propto \boldsymbol{w}_{\mathrm{CDB0}} + \boldsymbol{C}_{\mathrm{correction}} \boldsymbol{w}_{\mathrm{CDB0}} \tag{6}$$

From Eq. (6), there are two special cases:

Special case 1.1. If two covariance matrices are similar ($\mathbf{\Sigma}_0 \approx \mathbf{\Sigma}_1$), then the following also holds: $c_{xy} = \frac{\sigma_{xy}^2}{\sigma_{yy}^2} \approx \frac{\mathbb{E}[(x-\mu_x)(y-\mu_y)]}{\sigma_x \sigma_y} = r_{xy}$ where r_{xy} is the Pearson correlation coefficient (PCC) between x and y of the samples in each class. Eq. (6) becomes (Fig. A1, special case 1.1):

$$\boldsymbol{w}_{\mathrm{LD}} \propto \begin{bmatrix} \Delta\mu_x \\ \Delta\mu_y \end{bmatrix} + \begin{bmatrix} 0 & -r_{xy} \\ -r_{xy} & 0 \end{bmatrix} \begin{bmatrix} \Delta\mu_x \\ \Delta\mu_y \end{bmatrix} \propto \boldsymbol{w}_{\mathrm{CDB0}} + \boldsymbol{C}_{\mathrm{correction}} \boldsymbol{w}_{\mathrm{CDB0}} \tag{7}$$

Special case 1.1.1. From Eq. (7), if there is no correlation between x and y variables (e.g., $r_{xy} \approx 0$), then the equation becomes (Fig. A1, special case 1.1.1), which is equivalent to NMC method except for the bias:

$$\boldsymbol{w}_{\mathrm{LD}} \propto [\Delta\mu_x, \Delta\mu_y]^{\mathrm{T}} \propto \boldsymbol{w}_{\mathrm{CDB0}} \tag{8}$$

Special case 1.2. From Eq. (6), if two classes are symmetric about one variable, i.e., $\sigma 0_{xy}^2 \approx -\sigma 1_{xy}^2 \neq 0$, then $c_{xy} = \frac{\sigma_{xy}^2}{\sigma_{yy}^2} \approx 0$. In this case, the obtained discriminant is close to the one in Special case 1.1.1 (Fig. A1, special case 1.2):

$$\boldsymbol{w}_{\mathrm{LD}} \propto [\Delta\mu_x, \Delta\mu_y]^{\mathrm{T}} \propto \boldsymbol{w}_{\mathrm{CDB0}} \tag{9}$$

Fig. A1c shows how LDA applies geometric modifications to $\boldsymbol{w}_{\mathrm{CDB0}}$ as a shape adjustment for different shapes of data. When two covariance matrices are similar (Special case 1.1), the correction term acts only when the variables x and y have a certain extent of correlation and increases with

 doi:10.20944/preprints202502.0789.v2

this correlation (Fig. A1, the third row). Interestingly, when further assumptions are made that two variables have no correlation (Special case 1.1.1), or when two classes are symmetric about one variable (Special case 1.2), we can see from Eq. (8)-(9) that $\boldsymbol{w}_{\text{LD}}$ will approach $\boldsymbol{w}_{\text{CDB0}}$, which is indeed what we observe in the last two rows of Fig. A1. Videos showing these special cases using 2D simulated data can be found from this link[1].

The demonstrations of all these cases for higher-dimensional space are in the Appendix. D.

From the above derivation, Eq. (5)-(9) show that the solution of LDA can be represented by $\boldsymbol{w}_{\text{CDB0}}$ superimposed with a geometric correction on $\boldsymbol{w}_{\text{CDB0}}$, and the geometric correction term is obtained under the constraint of Eq. (1), which solves the maximization of the projection variance ratio. Without loss of generality, the conclusion can be extended to other linear classifiers with different constraints imposed on the geometric correction term, for instance NMC, for which the corrections terms are all to zero.

Here, we propose the generalized GDA theory in which not only the geometric correction term can impose any constraint based on different principles, but also an arbitrary number of correction terms can act together on $\boldsymbol{w}_{\text{CDB0}}$ to create the classification discriminant, which is given by:

$$\boldsymbol{w}_{\text{GD}} = \boldsymbol{w}_{\text{CDB0}} + \boldsymbol{C}_1\boldsymbol{w}_{\text{CDB0}} + \boldsymbol{C}_2\boldsymbol{w}_{\text{CDB0}} + \cdots + \boldsymbol{C}_n\boldsymbol{w}_{\text{CDB0}} \tag{10}$$

Fig. 1 shows that LDA is derived by the general GD equation imposing one variance-based geometric correction in Fig. 1b, shown in Fig. 1a using an instance of 2d artificial data with specific covariance.

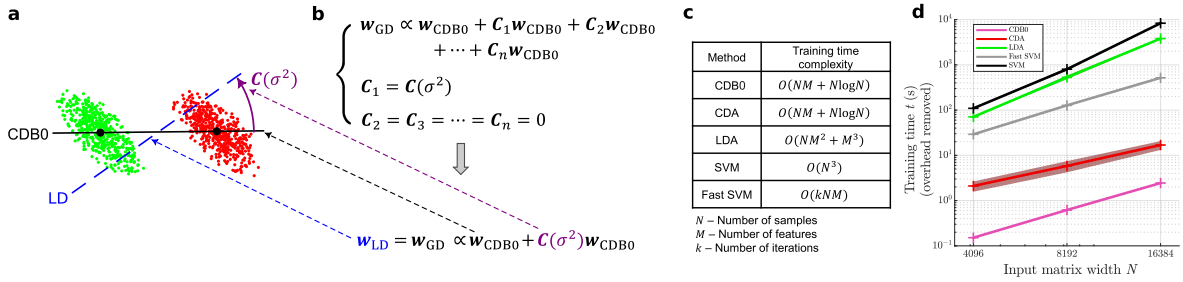

Figure 1. (a) A 2-dimensional instance showing that LD (blue dashed) can be constructed from CDB0 (black solid) using covariance correction which is a geometrical correction. (b) The evolution from CDB0 to LD. The first equation is the general form of geometrical discriminants (GD) that can be constructed by CDB0 with geometric corrections. If only one correction exists and this correction is a covariance-related matrix, LD can be derived from the general expression. (c) Theoretical training time complexity of linear classifiers. (d) The training time of linear classifiers with increasing input matrix sizes tested on CIFAR10. The number of features were set the same as number of samples N. Log-scale is used for both axes to reveal the scalability of algorithms. Shaded area represents standard error. CDB0: Centroid Discriminant Basis 0; LD: Linear Discriminant; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine

3. Centroid Discriminant Analysis (CDA)

Based on the GDA theory, in this study we propose an efficient geometric centroid-based linear classifier named centroid discriminant analysis (CDA), then introduce the extension to nonlinear classification via kernel method in section 5. As described by Eq. 10 in the GDA thoery, the model of a geometric classifier can be expressed by the basis CDB0 imposed by geometric corrections on this basis. We first give the conclusion that the final discriminant of CDA after n rotations is in the form $\boldsymbol{w}_{\text{CDA}}^{(n)} = \boldsymbol{w}_{\text{CDB0}} + \boldsymbol{C}_1\boldsymbol{w}_{\text{CDB0}} = \boldsymbol{w}_{\text{GD}}$, where $\boldsymbol{C}_1 = \boldsymbol{I} - \prod^n \boldsymbol{A}_{\text{cda}}$ is the geometric correction operator term, $\boldsymbol{I}$ is the identity operator, and $\boldsymbol{A}_{\text{cda}}$ is the operator of a single CDA rotation). The complete derivation of CDA in GDA theory can be found in Appendix. E.

[1] https://drive.google.com/drive/folders/1E3QqNzkBz7hdTWpBhEIA75pxlAY4xxw3?usp=sharing

 doi:10.20944/preprints202502.0789.v2

From the geometric point of view, CDA is built on the basis CDB0 in the GDA theory, then subjected to a series of geometric constraints guiding the rotations of the discriminant CDB in high-dimensional spaces. CDA follows a performance-dependent training that starting from CDB0 involves the iterative calculation of new adjusted centroid discriminant lines whose optimal rotations on the associated 2D planes are searched via Bayesian optimization. The following parts together with Figure. A2 describe the CDA workflow, mechanisms and principles in detail.

Performance-associated CDB Classifier: In GDA theory, CDB is defined as the unit vector pointing from the geometric centroid of the negative class to that of positive class. The geometric centroid is defined in a general sense which considers the weights of the data. The space of CDB consists of unit vectors obtained with all possible weights.

Apart from the discriminant line, a bias is further required to realize classification by offering a decision boundary for data projected onto the discriminant. We perform a search for this bias by checking the middle points of every two consecutive sorted projections. Thus, given N samples, there are $N-1$ candidates. We name the best candidate as optimal operating point (OOP) and select OOP according to a performance-score, defined as $(Fscore^{\text{pos}} + Fscore^{\text{neg}} + ACscore)/3$ (see Appendix. F for definitions), where pos and neg means evaluating the metric for each class. The performance score is a comprehensive metric that simultaneously considers sensitivity, recall, and specificity, providing a fair evaluation that accounts for biased models trained on imbalanced data, owing to the more conservative AC-score metric ([12]). With this OOP search strategy, any vector in the space of CDB is associated with a performance-score. The OOP search can be performed efficiently in $O(N\log N)$ time (see Alg. A3 for pseudocode).

CDA as Consecutive Geometric Rotations of CDB in 2D planes: From the function optimization perspective, binary classification problems can be regarded as maximizing certain performance metrics in multi-dimensional space. However, it is impractical to explicitly find the global optimum. Instead, a feasible technique is to find a path that gradually approaches the region close to the global optimum or suboptimums, and escapes from local minimums as much as possible. Based on this problem formulation, our idea is to start the optimization path from CDB0, continuously rotate the classification discriminant on 2D planes on which there is a high probability of having a classification discriminant with better performance .

To construct such a 2D plane with another vector, a key observation is that the samples, whose projections onto CDB are close to the decision boundary (i.e. OOP), should have more weights, because these samples are prone to overlapping with samples from the other class, causing misclassification. Thus, we compute another CDB using centroids with shifted sample weights toward OOP (see next part). On the plane formed by these two CDBs, the best rotation is found by Bayesian optimization. For clarity, we have the following definition: the first vector in each rotation is termed as CDB1, the second vector in each rotation is termed as CDB2, the CDB searched with the best performance is termed as CDA, and at the end of each rotation CDA becomes CDB1 for the next rotation. CDA rotation is used to refer to this rotation process. Figure A2a shows the diagram for the first CDA rotation, where CDB1 equals to CDB0 calculated using uniform sample weights. The CDA training stops when meeting any of the two criteria: (1) Reach the maximum 50 iterations. (2) the coefficient of variation (CV) of the last 10 performance-score is less than a threshold, indicating that the training has converged (see Alg. A1,A2 and A4 for pseudocode).

Sample Weights Update Strategy: The sample weights shift mechanism is described as follows. Given CDB1 and the associated OOP in each CDA rotation, the distance of all projections to OOP can be obtained by $d_i = |q_i - oop|,\ \forall i \in \{1, 2, \ldots, N\}$, which is then reversed by $\boldsymbol{d}_{\text{r}} = |\boldsymbol{d} - \min(\boldsymbol{d}) - \max(\boldsymbol{d})|$, in align with the purpose that points close to decision boundary should have larger weights. Since only the relative information of sample weights is important, they are L2-normalized and decay smoothly from previous sample weights by $\boldsymbol{\alpha} = \boldsymbol{\alpha} \odot \boldsymbol{d}_{\text{r}} / \|\boldsymbol{\alpha} \odot \boldsymbol{d}_{\text{r}}\|_2$, where $\odot$ indicates the Hadamard (element-wise) product. Specifically, in the first CDA rotation, the CDB1 is obtained with uniform sample weights, which corresponds to CDB0 in the GDA theory.

 doi:10.20944/preprints202502.0789.v2

Bayesian Optimization (BO): The CDA rotation aims to search for a unit vector CDB with the best performance-score, which can be efficiently realized by BO ([13]). BO is a statistical-based technique to estimate the global minimum of a function with as fewer evaluations as possible. CDA leverages this high-efficiency characteristic of BO to achieve fast training. BO itself has the time complexity of $O(Z^3)$ when it searches a single parameter, where Z is the number of sampling-and-evaluations. CDA employs a strategy that grows BO sampling times from 4 to the maximum 10 with CDA iterations, which are small numbers. (see Appendix C.1). Figure A2b shows an instance of BO working process. Alg. A5 shows the pseudocode for the CDA rotation as the black box function to optimize by BO.

Finalization: Lastly, CDA involves a finalization step, where on the best rotation plane it refines the discriminant line with a null-model statistical test using 100 random CDB lines drawn from the plane (see Appendix. C.2 and Alg. A6).

Multiclass Prediction: Error-correcting output codes (ECOC) ([14]) is a voting- (or penalizing-) based method to make multiclass predictions from trained binary classifiers. To predict a new sample, multiple scores can be obtained from the set of trained binary classifiers. These scores are interpreted as they either vote for or are against a particular class. Depending on the coding matrix and the loss function chosen, ECOC takes the class with the highest overall reward or lowest overall penalty as the predicted class. In this study, the coding matrix for one-versus-one training scheme is selected, as it creates more linearly separability ([15,16]). For the type of loss function, hinge-loss is chosen, as our internal tests suggested this loss leads to the best classification performance for all tested linear classifiers.

4. Experimental Evidences on Linear Classification of Real Data

In this section, we compared the proposed linear CDA with other linear classifiers. The selected linear classifiers include LDA, SVM and fast SVM, which are widely recognized as state-of-the-art, first-to-try linear classifiers for binary classification tasks. Additionally, we include the baseline method, CDB0 (equivalent to NMC with OOP bias), to quantify the improvements made by CDA over a simplistic centroid-based classification approach. Experiment details are in Appendix I.

4.1. Algorithm Scalability of Linear Classifiers

Training speed and scalability are the most important characteristics as they are the core advantages of linear classifiers. Therefore, we first performed a single-score speed test for linear classifiers using CIFAR10 dataset ([17]), with varying numbers of samples and features by random subsampling and oversampling. Original image data were used and flattened into 1D representations. Fig. 1c-d show respectively their theoretical training time complexities and running times with overhead removed (overheads were counted by training on a small 10×10 input data, see Fig. A4 for overhead kept). Tests at each data size were repeated five times. In Fig. 1d of running time (shaded area represents standard error), CDA was faster than all comparison approaches but the baseline CDB0, in align with their theoretical time complexities in Fig. 1c, highlighting CDA's efficiency that offers fast training times for smaller datasets and superior scalability for very large datasets.

In addition, there are real-world problems where the samples are at million-level in addition to plenty of features, posing a significant challenge to train classifiers in reasonable time. One of these problems is omics data classification such as single-cell sequencing data. We collected the 1.3-million-cell mouse brain dataset ([18]) and took the largest two classes to create a binary classification task (757,526 samples and 27,998 features), with varying number of samples by random subsampling till all samples were included. We compared CDA and fast SVM, which is one of the most efficient and scalable linear SVM, with regard to performance and speed . Original data of sequencing counts were used, and the training and test sets were created using a 4:1 split. The results show that with growing sample sizes, linear CDA outperforms fast SVM not only on classification performance AUROC (Figure 2a and Appendix Figure A3), but also on the single-core training speed (Figure 2b). These results indicate that linear CDA is even more efficient and scalable than the flagship SVM method in efficiency.

 doi:10.20944/preprints202502.0789.v2

Hence, CDA as a fast approach has the potential to drive large-scale real-world applications in fields such as biomedicine and autonomous driving.

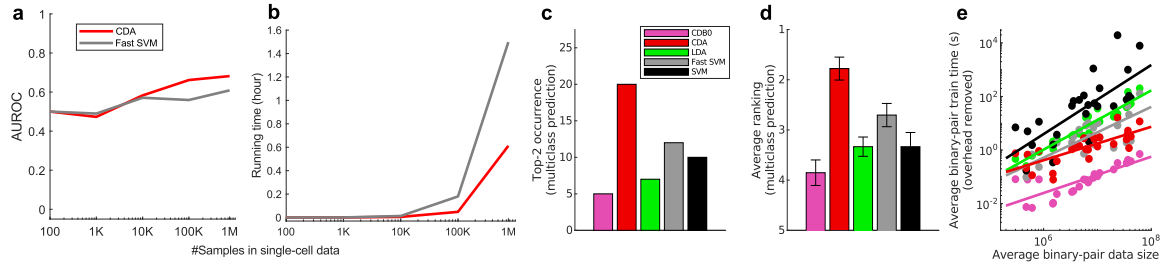

Figure 2. (a-b) (a) Test set performance AUROC and (b) training time on large-scale single-cell data with varying sizes of training samples by CDA and fast SVM. (c-e) Classification performance on 27 real datasets. (c) Top-2 occurrences and (d) average ranking of algorithms according to multiclass AUROC. Error bars represent standard errors. (e) Relationship between training time and dataset size averaged across binary-pairs in each dataset. Log-scale is used for both axes to reveal the scalability of algorithms; linear regression is applied to the points in the log-scale. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine; AUROC: Area under receiver operating characteristic curve.

4.2. Linear Classification Performance on Real Data

We assessed linear classification performance across 27 datasets, including standard image classification ([17,19–26]), medical image classification ([27]), and chemical property prediction tasks ([28]). These datasets represent a broad range of real-world applications and varying data sizes, enabling evaluations for both training speed and predictive performance. Image data were used in their original value flattened to 1d vectors; chemical formula were processed by simplified molecular input line entry system (SMILES) tokenized encoding (see Appendix Table A1). Each dataset was split into a 4:1 ratio for training and test sets. The final model for each method was an unweighted ensemble of the five cross-validated models on the training set.

Fig. 2c-d show the test set multiclass prediction performance. In Fig. 2c, CDA achieved a top-2 occurrence on 20 out of 27 datasets based on AUROC, indicating its stability and competitiveness. Fig. 2d shows that CDA achieved the highest average ranking which is 1.78, outperforming LDA and SVM, with statistically significant margins as indicated by standard error bars. See Appendix K-L for binary classification results, ranking results based on AUPR, Fscore and accuracy, and the complete performance tables.

Fig. 2e compares training speed across datasets. Train times per binary-pair were averaged within each dataset after deducting constant overheads unrelated to data size (same manner as above; see Fig. A5 for overhead kept). Linear regression of train times and data sizes shows that CDA and CDB0 exhibit the best scalability, followed by fast SVM, LDA, and SVM, consistent with their theoretical complexities. CDA not only improves significantly over CDB0 in performance but also retains similar scalability. This improvement is driven by three key components: generalized centroids with non-uniform weights, sample weight shift strategy, and rotations by Bayesian optimization. The weighted average number of iterations required by CDA per dataset was 29.33 - a small constant that does not contribute to time complexity. In contrast, fast SVM's iterations increase significantly with more challenging datasets to achieve a reasonable classification performance. Considering the diversity of tested datasets, CDA demonstrates itself as a generic classifier with strong performance and scalability, making it applicable to large-scale classification tasks across various domains.

5. Preliminary Test on Nonlinear Kernel CDA

In scenarios where data have abundant nonlinear relationship, linear classifiers often fall short in performance. To address this problem, models must exploit these nonlinear patterns to enhance expressiveness and improve classification performance. In fact, linear CDA can be naturally extended

 doi:10.20944/preprints202502.0789.v2

to a nonlinear variant using kernel method. This generalization requires only minor modifications to the algorithm (see Appendix C.4), meaning that the high computational efficiency of linear CDA is largely retained. The main computational bottleneck lies in the kernel matrix computation, which is a common cost shared by all kernel-based classifiers.

To compare the performance of nonlinear CDA and other nonlinear classifiers, we performed tests on two challenging datasets, image dataset SVHN and chemical property dataset ClinTox (see Appendix. A1 for data processing). We choose these datasets because they are difficult to classify with linear classifiers, and we are interested in to what extent can nonlinear classifiers improve over linear ones. We used a subset of SVHN samples (24,000 out of 99,289) due to the time limitation to test with kernel method. We compared linear CDA, nonlinear gaussian CDA (nCDA), linear SVM, nonlinear gaussian SVM (nSVM). For kernel methods, we performed hyperparameter search for the gaussian parameter σ. The data were divided into train, validation and test set, where on the validation set the hyperparameter was tuned. The training specifics can be found in Appendix I.2. The multiclass test set results (on ClinTox binary results since it is a binary task) in Table 1 show that nonlinear kernel CDA outperforms gaussian SVM and linear methods on both the SVHN subset and ClinTox on all classification metrics. Despite the fact that gaussian CDA improves substantially over linear CDA in tasks such as the SVHN image classification, we emphasize that linear CDA is still very useful and important, with high efficiency and low resource requirement in computation, among other merits.

Table 1. Test set classification performance on SVHN subset and ClinTox.

Dataset	Method	AUROC	AUPR	Fscore	ACscore
SVHN subset (image)	CDA	0.615±0.02	0.63±0.02	0.619±0.02	0.423±0.05
	nCDA	**0.777±0.01**	**0.782±0.01**	**0.78±0.01**	**0.731±0.02**
	SVM	0.555±0.01	0.568±0.007	0.551±0.006	0.273±0.05
	nSVM	0.736±0.02	0.776±0.009	0.756±0.008	0.654±0.03
ClinTox (chemical)	CDA	0.567	0.561	0.56	0.351
	nCDA	**0.625**	**0.627**	**0.627**	**0.46**
	SVM	0.565	0.578	0.575	0.294
	nSVM	0.500	0.481	0.480	0.000

6. Conclusions and Discussions

Linear classifiers, while inherently simpler, are favored in certain contexts due to their reduced tendency to overfit and their interpretability in decision-making. However, achieving both high scalability and robust predictive performance simultaneously remains a significant challenge.

In this study, the introduction of the Geometric Discriminant Analysis (GDA) framework marks a notable step forward in addressing this challenge. By leveraging the geometric properties of centroids – a fundamental concept in multiple disciplines – GDA provides a unifying framework for certain linear classifiers. The core innovation lies in a special type of Centroid Discriminant Basis (CDB0), which serves as the foundation for deriving discriminants. These discriminants, when augmented with geometric corrections under varying constraints, extend the theoretical flexibility of GDA. Notably, Nearest Mean Classifier (NMC) and Linear Discriminant Analysis (LDA) are shown to be a subset of this broader framework, demonstrating how they converge to CDB0 under specific conditions. This theoretical generalization not only validates the GDA framework but also sets the stage for novel classifier designs.

A key practical contribution of this work is the Centroid Discriminant Analysis (CDA), a specialized implementation of GDA. CDA employs geometric rotations of the CDB within planes defined by centroid-vectors with shifted sample weights. These rotations, combined with Bayesian optimization techniques, enhance the method's scalability, achieving quadratic time complexity $O(NM + N\log N)$. Across diverse datasets—including standard images, medical images, and chemical property classifications—CDA demonstrated superior performance in scalability, predictive performance, and stability. We emphasize that CDA not merely robustly outperforms existing linear classification methods, but crucially, it achieves this with superior computational efficiency. Specifically, CDA exhibits a quadratic

 doi:10.20944/preprints202502.0789.v2

time complexity in the worst-case scenario, compared to the cubic complexity typical of established methods such as LDA and SVM, with significantly shorter runtimes that highlight its practical advantage in real-world applications. These findings hold particular relevance for the broader machine learning community, as linear classifiers remain widely used across numerous scientific domains where interpretability, scalability, and computational efficiency are critical. In such contexts, practitioners often prefer models that are not only robust and fast but also transparent and easy to deploy in real-world decision-making scenarios.

In cases where the data exhibit complex nonlinear structures that linear classifiers cannot adequately capture, linear CDA can be extended to a nonlinear form via kernel methods. This generalization significantly broadens the applicability of CDA by enabling it to handle nonlinear patterns in the data. Together, linear and kernel-based CDA form a complementary toolkit for classification tasks. A practical strategy is to begin with linear CDA to see whether the predictions are satisfying, which offers fast and interpretable results. Our preliminary results also suggest that extending CDA with nonlinear kernels is a promising direction for future research, since the variety of available kernels broadens the opportunity to tailor the mapping of original data onto higher-dimensional spaces according to the specific structure of the data - where it may become more easily separable.

In conclusion, the GDA framework and its CDA implementation represent a paradigm shift in classifier design, combining the interpretability of geometric principles with state-of-the-art computational efficiency. This work not only advances the field of classification but also lays groundwork for innovative approaches to supervised learning.

Appendix A. GDA Demonstration with 2D Artificial Data

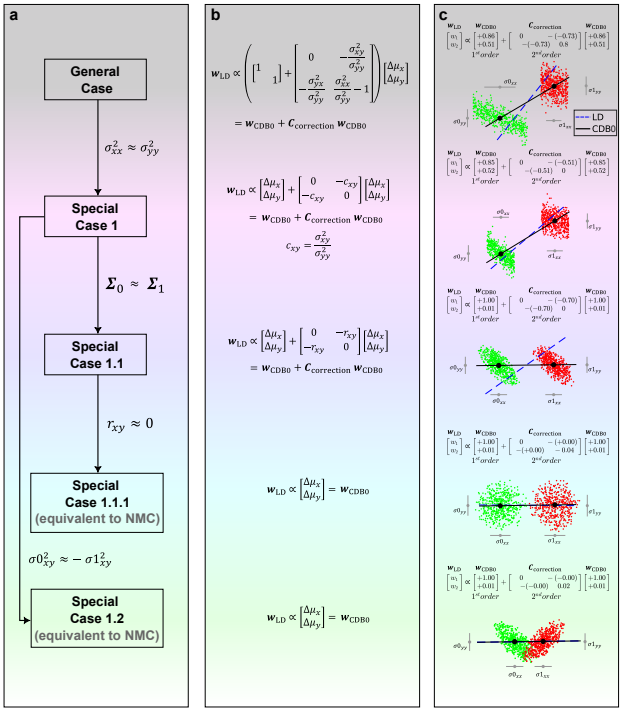

Figure A1. The GDA theory in 2 dimensions showing how the relation between LD and CDB0 evolves under specific conditions. (a) The relations between LD and CDB0 proceeding from the general case to different special cases under the conditions shown along each arrow. (b) The specific expressions of LD in terms of CDB0 and corrections corresponding to each case in column (a). (c) Binary classifications models of LD (blue dashed) and CDB0 (black solid) on different 2D data corresponding to each case in column (a). LD: Linear Discriminant; CDB0: Centroid Discriminant Basis 0.

 doi:10.20944/preprints202502.0789.v2

Appendix B. CDA Schematic Diagram

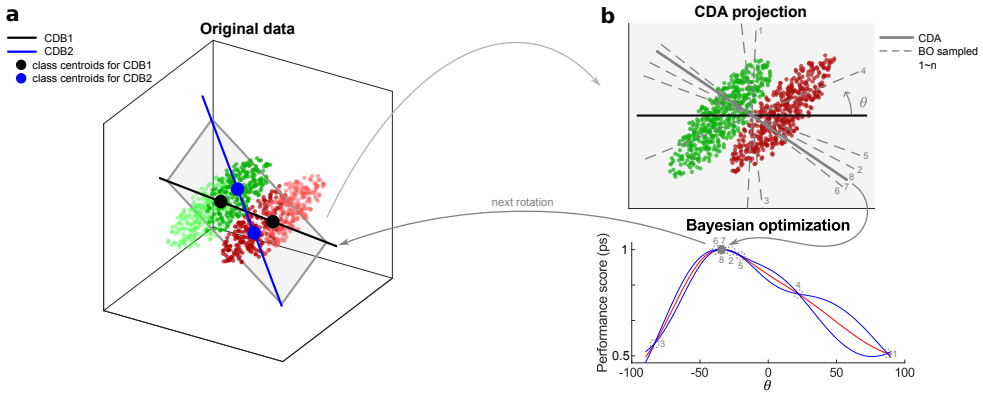

Figure A2. Diagram for the first CDA rotation. (a) CDB1 is obtained from specific sample weights. CDB2 is obtained from centroids with a shifted sample weights toward the decision boundary. Darker points represent larger sample weights. CDB1 and CDB2 form a 2d plane on which BO is performed to estimate the best classification discriminant. (b) The process of BO to estimate the best discriminant. The rotation angle θ from CDB1 is the only independent variable to search in BO. The estimated optimal line CDA servers as the new CDB1 in the next CDA rotation. CDB: Centroid Discriminant Basis; CDA: Centroid Discriminant Analysis.

Appendix C. The CDA Algorithm

Appendix C.1. Bayesian Optimization

One factor related to the effectiveness of CDA is the estimation with varying precision in BO. We set the number of BO sampling times to $\min(3 + rot, 10)$ as default parameter during the CDA rotation, where rot is the rot-th rotation. During the first few CDA rotations, the BO estimation has relatively less precision. This gives CDA enough randomness to first enter a large region in which there are more discriminants with high performance, under the hypothesis that regions close to global optimum or suboptimum have more high-performance solutions that are more likely to be randomly selected. Starting from this large region, BO precision is gradually enhanced to refine the search in or close to this region in terms of Euclidean distance, and force CDA to converge. The upper limit of 10 CDA rotations still creates a small level of imprecision, in order to make CDA escape from small-size local minimum, to acquire higher performance, and improve training efficiency.

Appendix C.2. Refining CDA with Statistical Examination on 2D Plane

In CDA, the plane associated with the best performance is selected, but it is uncertain whether there are higher-performance discriminants on this plane, since the precision might not be high enough with 10 BO samplings and even lower for less samplings. To determine whether the BO is precise enough, a statistical examination using p-value with respect to null-model is performed. On this best plane, we generate 100 random CDBs, run BO in turn for 10, 20 and to the maximum 30 BO samplings, until the p-value of training performance-score of the BO-estimated CDB with regard to the null model is 0, otherwise using the best random CDB. In this way, we are able to give a confidence level of the BO estimated discriminant in the best CDA rotation plane. Alg. A6 shows the pseudocode for the refining finalization step.

Appendix C.3. Linear CDA Pseudocode

This study deals with a supervised binary classification problem on labeled data $\{\boldsymbol{X}, \boldsymbol{y}\} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ with N samples and M features. The samples consist of positive and negative class data χ_1 and χ_2 with sizes N_1 and N_2, respectively, and the corresponding labels are tokenized as 1 and 0, respectively. The sample features are flattened to 1D if they have higher dimensions, such as image

 doi:10.20944/preprints202502.0789.v2

data. The following section introduces CDA in pseudo-code. CDA rotations, BO sampling and p-value computing involve constant numbers, thus, they do not contribute to the time complexity.

Algorithm A1 CDA Main Algorithm (CDA) $O(NM + N\log N)$

Input: $N \times M$ data matrix $\boldsymbol{X}$, $N \times 1$ labels $\boldsymbol{y}$
Initialize $\boldsymbol{\alpha} = [1; 1; \ldots; 1]^{N\times 1}/\sqrt{N}$ $\{O(N)\}$
Compute $\boldsymbol{w}_{\text{CDB1}} = \sum_{c=1}^{2} (-1)^{c+1} \frac{1}{N_c} \sum_{\boldsymbol{x}_i \in \chi_c} \alpha_i \boldsymbol{x}_i$ $\{O(NM)\}$
Normalize $\boldsymbol{w}_{\text{CDB1}} = \boldsymbol{w}_{\text{CDB1}}/\|\boldsymbol{w}_{\text{CDB1}}\|_2$ $\{O(M)\}$
Initialize $ps^* = 0$ $\{O(1)\}$
for $i = 1$ **to** 50 **do**
  $\boldsymbol{\alpha} = \text{updateSampleWeights}(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{w}_{\text{CDB1}})$ $\{O(NM + N\log N)\}$
  Compute $\boldsymbol{w}_{\text{CDB2}} = \sum_{c=1}^{2} (-1)^{c+1} \frac{1}{N_c} \sum_{\boldsymbol{x}_i \in \chi_c} \alpha_i \boldsymbol{x}_i$ $\{O(NM)\}$
  Normalize $\boldsymbol{w}_{\text{CDB2}} = \boldsymbol{w}_{\text{CDB2}}/\|\boldsymbol{w}_{\text{CDB2}}\|_2$ $\{O(M)\}$
  Set $N_{\text{BO}} = \min(i + 3, 10)$ $\{O(1)$ Number of BO sampling$\}$
  $(\boldsymbol{w}_{\text{CDA}}, ps) = \text{CdaRotation}(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{w}_{\text{CDB2}}, \boldsymbol{X}, \boldsymbol{y}, N_{\text{BO}})$ $\{O(NM + N\log N)\}$
  if $ps > ps^*$ **then**
    $(\boldsymbol{w}^*_{\text{CDB1}}, \boldsymbol{w}^*_{\text{CDB2}}, ps^*) = (\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{w}_{\text{CDB2}}, ps)$ $\{O(M)\}$
  end if
  if coefficient of variation of the last 10 $ps < 0.001$ **then**
    break {Early stop convergence}
  end if
  Update $\boldsymbol{w}_{\text{CDB1}} = \boldsymbol{w}_{\text{CDA}}$ $\{O(M)\}$
end for
Compute $(\boldsymbol{w}_{\text{CDA}}, oop) = \text{refineOnBestPlane}(\boldsymbol{w}^*_{\text{CDB1}}, \boldsymbol{w}^*_{\text{CDB2}}, \boldsymbol{X}, \boldsymbol{y})$ $\{O(NM + N\log N)\}$
Output: $\boldsymbol{w}_{\text{CDA}}, oop$

Algorithm A2 Update Sample Weights (updateSampleWeights) $O(NM + N\log N)$

Input: $N \times M$ data matrix $\boldsymbol{X}$, $N \times 1$ labels $\boldsymbol{y}$, $N \times 1$ sample weights $\boldsymbol{\alpha}$, $1 \times M$ vector $\boldsymbol{w}_{\text{CDB1}}$
Initialize $oop = \text{searchOOP}(\boldsymbol{X}, \boldsymbol{w})$ $\{O(N\log N)\}$
Compute $\boldsymbol{q} = \boldsymbol{X}\boldsymbol{w}^{\text{T}}_{\text{CDB1}}$ $\{O(NM)\}$
Compute $d_i = |q_i - oop|$, $\forall i \in \{1, 2, \ldots, N\}$ $\{O(N)\}$
Compute $\boldsymbol{d}_r = |\boldsymbol{d} - \min(\boldsymbol{d}) - \max(\boldsymbol{d})|$ $\{O(N)\}$
Update $\boldsymbol{\alpha} = \boldsymbol{\alpha} \odot \boldsymbol{d}_r/\|\boldsymbol{\alpha} \odot \boldsymbol{d}_r\|_2$ $\{O(N), \odot$ is Hadamard product$\}$
Output: $\boldsymbol{\alpha}$

Algorithm A3 Search Optimal Operating Point (searchOOP) $O(N\log N)$

Input: $N \times 1$ projected points $\boldsymbol{q}$, $N \times 1$ labels $\boldsymbol{y}$
Sort $(\boldsymbol{q}, \boldsymbol{y})$ $\{O(N\log N)$, sort according to $\boldsymbol{q}\}$
Initialize $\boldsymbol{cm} = \text{evaluateMetrics}(\boldsymbol{y}, [0; 0; \ldots; 0]^{N\times 1})$ $\{O(N)$ initial confusion matrix$\}$
for $i = 1$ **to** $N - 1$ **do**
  Update $\boldsymbol{cm} = \text{updateCM}(y_i, \boldsymbol{cm})$ $\{O(1)$ scan each label and adjust cm$\}$
  Compute $ps_i = \text{evaluateMetrics}(\boldsymbol{cm})$ $\{O(1)\}$
end for
Compute $idx = \arg\max_i ps_i$ $\{O(N)\}$
Compute $ps = \max(\boldsymbol{ps})$ $\{O(N)\}$
Compute $oop = \frac{q_{idx} + q_{(idx+1)}}{2}$ $\{O(1)\}$
Output: oop, ps

 doi:10.20944/preprints202502.0789.v2

Algorithm A4 Approximate Optimal Line by BO (CdaRotation) $O(NM + N\log N)$

Input: $1 \times M$ lines $\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{w}_{\text{CDB2}}$, $N \times M$ data matrix $\boldsymbol{X}$, $N \times 1$ labels $\boldsymbol{y}$, total BO iteration N_{BO}
Compute $\boldsymbol{w}_{\text{orth}} = \boldsymbol{w}_{\text{CDB2}} - \boldsymbol{w}_{\text{CDB1}}^{\text{T}} \boldsymbol{w}_{\text{CDB2}} \boldsymbol{w}_{\text{CDB1}}$ $\{O(M)\}$
Normalize $\boldsymbol{w}_{\text{orth}} = \boldsymbol{w}_{\text{orth}} / \|\boldsymbol{w}_{\text{orth}}\|_2$ $\{O(M)\}$
Estimate $\hat{\theta}$ = BayesianOptimization(evaluateRotation$(\theta, \boldsymbol{w}_{\text{CDB1}}, \boldsymbol{w}_{\text{orth}}, \boldsymbol{X}, \boldsymbol{y}), N_{\text{BO}})$, $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ $\{O(NM + N\log N)\}$
Compute $\boldsymbol{w}_{\text{CDA}} = \boldsymbol{w}_{\text{CDB1}} \cos\hat{\theta} + \boldsymbol{w}_{\text{orth}} \sin\hat{\theta}$ $\{O(M)\}$
Compute $\boldsymbol{q} = \boldsymbol{X}\boldsymbol{w}_{\text{CDA}}^{T}$ $\{O(NM)\}$
Get the mean of (oop, ps) = searchOOP$(\boldsymbol{q}, \boldsymbol{y})$ with 5-fold cross-validation $\{O(N\log N)\}$
Output: $\boldsymbol{w}_{\text{CDA}}, oop, ps$

Algorithm A5 Evaluate the Line with Rotation Angle (evaluateRotation) $O(NM + N\log N)$

Input: Rotation angle θ, $1 \times M$ lines $\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{w}_{\text{orth}}$, $N \times M$ data matrix $\boldsymbol{X}$, $N \times 1$ labels $\boldsymbol{y}$
Compute $\boldsymbol{w}_{\text{CDA}} = \boldsymbol{w}_{\text{CDB1}} \cos\theta + \boldsymbol{w}_{\text{orth}} \sin\theta$ $\{O(M)\}$
Compute $\boldsymbol{q} = \boldsymbol{X}\boldsymbol{w}_{\text{CDA}}^{T}$ $\{O(NM)\}$
Get mean ps = searchOOP$(\boldsymbol{q}, \boldsymbol{y})$ with 5-fold cross-validation scheme $\{O(N\log N)\}$
Output: ps

Algorithm A6 Refine on the Best Model (refineOnBestPlane) $O(NM + N\log N)$

Input: $1 \times M$ $\boldsymbol{w}_{\text{CDB1}}$ and its orthogonal vector $\boldsymbol{w}_{\text{orth}}$, $N \times M$ data matrix $\boldsymbol{X}$, $N \times 1$ labels $\boldsymbol{y}$
for $i = 1$ **to** 100 **do**
  Randomly choose $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ $\{O(1)\}$
  Compute $\boldsymbol{w}_{\text{r}} = \boldsymbol{w}_{\text{CDB1}} \cos\theta + \boldsymbol{w}_{\text{orth}} \sin\theta$ $\{O(M)\}$
  Compute $\boldsymbol{q} = \boldsymbol{X}\boldsymbol{w}_{\text{r}}^{\text{T}}$ $\{O(NM)\}$
  Get mean $(oop_{\text{r}}, ps_{\text{r}})_i$ = searchOOP$(\boldsymbol{q}, \boldsymbol{y})$ with 5-fold cross-validation $\{O(N\log N)\}$
end for
Sort idx = sortIdx$([ps, ps_{\text{r}}])$ $\{O(1)$ sort returning position indices$\}$
Compute $p = 1 - \frac{(idx(1)-1)}{N_r}$ $\{O(1)\}$
Initialize $N_{\text{BO}} = 0$
while $p \neq 0$ and $N_{\text{BO}} < 30$ **do**
  Increment $N_{\text{BO}} = N_{\text{BO}} + 10$
  Compute $(\boldsymbol{w}_{\text{CDA}}, oop, ps)$ = CdaRotation$(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{w}_{\text{CDB2}}, \boldsymbol{X}, \boldsymbol{y}, N_{\text{BO}})$ $\{O(NM + N\log N)\}$
  Sort idx = sortIdx$([ps, \text{ps}_r])$ $\{O(1)$ sort returning position indices$\}$
  Compute $p = 1 - \frac{(idx(1)-1)}{N_r}$ $\{O(1)\}$
end while
if $p \neq 0$ **then**
  Compute $i = \arg\min_i ps_{\text{r},i}$ $\{O(1)\}$
  Set $(\boldsymbol{w}_{\text{CDA}}, oop, ps) = (\boldsymbol{w}_r, oop_{\text{r}}, ps_{\text{r}})_i$ $\{O(1)\}$
end if
Output: $\boldsymbol{w}_{\text{CDA}}, oop, ps$

Appendix C.4. Nonlinear kernel CDA

The CDA algorithm involves computing the CDBs using the equation provided in Appendix B, Algorithm 1:

$$\boldsymbol{w}_{\text{CDB}} = \sum_{c=1}^{2} (-1)^{c+1} \cdot \frac{1}{N_c} \sum_{\boldsymbol{x}_i \in \chi_c} \alpha_i \boldsymbol{x}_i \tag{A1}$$

Combining this with the normalization step for each CDB:

$$\boldsymbol{w}_{\text{CDB}} \leftarrow \boldsymbol{w}_{\text{CDB}} / \|\boldsymbol{w}_{\text{CDB}}\|_2, \tag{A2}$$

the equation can be rewritten as:

 doi:10.20944/preprints202502.0789.v2

$$\boldsymbol{w}_{\mathrm{CDB}} = (\boldsymbol{\alpha} \odot \boldsymbol{n} \odot \boldsymbol{y}_t \cdot k)^\top \boldsymbol{X} = \boldsymbol{\beta}^\top \boldsymbol{X}, \tag{A3}$$

where $\boldsymbol{\alpha}$ is the vector of sample weights; $\boldsymbol{n} = [1/N_{\mathrm{map}(i)}]$ for all $i \in \{1, 2, \ldots, N\}$ is the weight sum division for each class, and $\mathrm{map}(i)$ maps sample index i to class index c; $\boldsymbol{y}_t = (-1)^{c+1}$ are the tokenized labels ($+1$ or -1); k is the factor to normalize each CDB as a unit vector.

For the rotated CDBs used in Bayesian optimization (Appendix B, Algorithms 4–5), they can be expressed as linear combinations of CDB1 and orthogonalized CDB2, mixing their corresponding $\boldsymbol{\beta}$ vectors. Thus, the sample coefficient $\boldsymbol{\beta}$ is tracked throughout the training process for each CDB.

The data projection can be expressed as:

$$\boldsymbol{q} = \boldsymbol{X}\boldsymbol{w}_{\mathrm{CDB}}^\top = \boldsymbol{X}(\boldsymbol{\beta}^\top \boldsymbol{X})^\top = \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\beta} \tag{A4}$$

With this expression, the data projection can incorporate kernel methods to realize nonlinear classification, shown by the following equation.

$$\boldsymbol{q} = Ker(\boldsymbol{X}, \boldsymbol{X}^\top)\boldsymbol{\beta} \tag{A5}$$

In fact, kernel methods implicitly projects data into a high-dimensional space, then applying linear classifiers in the transformed feature space.

Appendix D. Deriving LDA in the GDA theory for m-dimensions (m>2)

In the GDA theory, we have demonstrated that linear discriminant (LD) can be expressed by the basis CDB0 with different geometric corrections on CDB0 under different conditions, as shown in Fig. A1. The conclusions are not confined only to the 2-dimensional case but also applicable to the m-dimensional case ($m > 2$). Here we derive the generalization of the LD coefficient to the m-dimensional case. The discriminant of LDA is given by $\boldsymbol{w}_{\mathrm{LD}} \propto \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \propto \boldsymbol{\Sigma}^{-1}\boldsymbol{w}_{\mathrm{CDB0}}$, where it is assumed that the within-class covariance matrix $\boldsymbol{\Sigma}$ is invertible, and $\boldsymbol{\Sigma} = \boldsymbol{\Sigma_0} + \boldsymbol{\Sigma_1}$ is the sum of covariance matrices of individual classes. Denote the elements in $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma_0}$, and $\boldsymbol{\Sigma_1}$ by σ_{ij}, $\sigma 0_{ij}$, and $\sigma 1_{ij}$ for all $i, j \in \{1, 2, \ldots, m\}$, respectively.

Special Case 1. Assume that pairwise features have the same covariance α, and all pairwise features have the same covariance $\alpha + \beta$ (In experiments, we relax the conditions to have similar variance and similar covariance). Then, $\boldsymbol{\Sigma} = \beta\mathbf{I} + \alpha\mathbf{1}$, where $\mathbf{I}$ is the identity matrix, and $\mathbf{1}$ is the all-one matrix. According to the Woodbury matrix identity, we have $\boldsymbol{\Sigma}^{-1} = \frac{1}{\beta}\mathbf{I} - \frac{\alpha}{\beta(m\alpha+\beta)}\mathbf{1} \propto \mathbf{I} - \frac{\alpha}{(m-1)\alpha+\beta}\mathbf{1}_N$, where $\mathbf{1}_N$ is the matrix with all ones except on the diagonal. Thus, $\boldsymbol{w}_{\mathrm{LD}} \propto \boldsymbol{w}_{\mathrm{CDB0}} - \frac{\alpha}{(m-1)\alpha+\beta}\mathbf{1}_N\boldsymbol{w}_{\mathrm{CDB0}}$, which corresponds to the row for special case 1 in Fig. A1.

Special Case 1.1. Based on the assumptions made in Special Case 1, we further assume that the two covariance matrices are similar, i.e., $\boldsymbol{\Sigma_0} \approx \boldsymbol{\Sigma_1}$, so that $\sigma_{0,ij} = \sigma_{1,ij}$ for all i, j. We further have $\frac{\sigma_{ij}}{\sigma_{ii}} = \frac{2\sigma 0_{ij}}{2\sigma 0_{ii}} = \frac{\sigma 0_{ij}}{\sqrt{\sigma 0_{ii}}\sqrt{\sigma 0_{jj}}} = r_{ij} \quad \forall i \neq j$, where r_{ij} is the Pearson correlation coefficient (PCC) between features i and j. Based on the assumptions made in this special case, all pairwise PCCs are the same, i.e., $r_{ij} = r$ for all $i \neq j$. Thus, $\boldsymbol{\Sigma}^{-1} \propto \mathbf{I} - \frac{r}{(m-2)r+1}\mathbf{1}_N$, which shows that the geometric correction part is related to feature correlations. Thus, $\boldsymbol{w}_{\mathrm{LD}} \propto \boldsymbol{w}_{\mathrm{CDB0}} - \frac{r}{(m-2)r+1}\mathbf{1}_N\boldsymbol{w}_{\mathrm{CDB0}}$, which corresponds to the row for special case 1.1 in Fig. A1.

Special Case 1.1.1. Based on Special Case 1.1, assume further that all pairwise features have no correlations (i.e., $r = 0$), then according to the equation, $\boldsymbol{\Sigma}^{-1} \propto \mathbf{I}$. Thus, $\boldsymbol{w}_{\mathrm{LD}} \propto \boldsymbol{w}_{\mathrm{CDB0}}$, which corresponds to the row for special case 1.1.1 in Fig. A1. By these derivations, we show how LDA converges to CDB0 under specific conditions of the data.

Appendix E. Deriving CDA in the GDA Theory

In this section, we give the mathematical demonstration of how CDA is formulated in the GDA theory. Since the correction matrix for LDA in Section 2 is a special case of a linear operator, we

 doi:10.20944/preprints202502.0789.v2

extend to a more general correction term deriving CDA in the GDA theory. The construction of the geometric discriminant $\boldsymbol{w}_{\text{GD}}$ can be realized by continuously rotating CDB0 in planes that satisfy several constraints, shown by:

$$\begin{aligned}
\boldsymbol{w}_{\text{CDA}}^{(n)} &= \boldsymbol{A}_{\text{cda}}(\boldsymbol{w}_{\text{CDB1}}^{(n)}, \boldsymbol{\alpha}^{(n)}, \boldsymbol{X}, \boldsymbol{y}) \\
&= \boldsymbol{A}_{\text{cda}}(\boldsymbol{w}_{\text{CDA}}^{(n-1)}, \boldsymbol{\alpha}^{(n)}, \boldsymbol{X}, \boldsymbol{y}) \\
&= \boldsymbol{A}_{\text{cda}}(\boldsymbol{A}_{\text{cda}}(\boldsymbol{w}_{\text{CDB1}}^{(n-1)}, \boldsymbol{\alpha}^{(n-1)}, \boldsymbol{X}, \boldsymbol{y}), \boldsymbol{X}, \boldsymbol{y}) \\
&= \cdots = \left(\prod^{n} \boldsymbol{A}_{\text{cda}}\right)(\boldsymbol{w}_{\text{CDB1}}^{(1)}, \boldsymbol{\alpha}^{(1)}, \boldsymbol{X}, \boldsymbol{y}) \\
&= \left(\prod^{n} \boldsymbol{A}_{\text{cda}}\right)(\boldsymbol{w}_{\text{CDB0}}) \\
&= \boldsymbol{w}_{\text{CDB0}} + (\boldsymbol{I} - \prod^{n} \boldsymbol{A}_{\text{cda}})(\boldsymbol{w}_{\text{CDB0}}) \\
&= \boldsymbol{w}_{\text{CDB0}} + \boldsymbol{C}_1 \boldsymbol{w}_{\text{CDB0}} = \boldsymbol{w}_{\text{GD}}
\end{aligned} \tag{A6}$$

where $\boldsymbol{C}_1 = (\boldsymbol{I} - \prod^{n} \boldsymbol{A}_{\text{cda}})$, in which $\boldsymbol{I}$ is the identity operator. For a given dataset $\boldsymbol{X}, \boldsymbol{y}$ are fixed, thus it is simplified by:

$$(\boldsymbol{w}_{\text{CDA}}, \boldsymbol{\alpha}) = \boldsymbol{A}_{\text{cda}}(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{\alpha}, \boldsymbol{X}, \boldsymbol{y}) = \boldsymbol{A}_{\text{cda}}(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{\alpha}) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^n$$

which defines the linear operator that maps the line CDB1 to CDA in each CDA iteration and maps the sample weights $\boldsymbol{\alpha}$ to the updated sample weights.

The second equality holds because $\boldsymbol{w}_{\text{CDA}}$ obtained at the end of each rotation is used as $\boldsymbol{w}_{\text{CDB1}}$ in the next rotation. The fifth equality shows that the final CDA line can be written in a form that only depends on variables of the first iteration. This is because, in the first CDA rotation, the initial sample weights $\boldsymbol{\alpha}^{(1)}$ are uniform, and CDB1 is the same as CDB0 of the GDA theory. Thus, from the sixth equality, $\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\alpha}^{(1)}$ are omitted. The last three equalities show that the final CDA discriminant is a subcase of the generalized GDA theory.

The operator $\boldsymbol{A}_{\text{cda}}$ can be decomposed as two sequential operators:

$$\boldsymbol{A}_{\text{cda}} = \boldsymbol{A}_{\text{BO}}\Big(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{A}_{\text{cv}}(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{\alpha}, \boldsymbol{X}, \boldsymbol{y}), \quad \boldsymbol{\alpha}, \boldsymbol{X}, \boldsymbol{y}\Big) = \boldsymbol{A}_{\text{BO}}\Big(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{A}_{\text{sbc}}(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{\alpha}), \boldsymbol{\alpha}\Big) \tag{A7}$$

where $\boldsymbol{A}_{\text{cv}}(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{\alpha}) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ is the linear operator that maps CDB1 to CDB2. $\boldsymbol{A}_{\text{BO}}(\boldsymbol{w}_{\text{CDB1}}, \boldsymbol{A}_{\text{cv}}, \boldsymbol{\alpha}) : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^n$ is the linear operator that maps two vectors CDB1 and CDB2 to the BO-estimated discriminant on the plane spanned by CDB1 and CDB2, and updates samples weights.

The operator $\boldsymbol{A}_{\text{cv}}$ does the mapping through the following equation:

$$\tilde{\boldsymbol{w}}_{\text{CDB2}} = \sum_{c=0}^{1} (-1)^{c+1} \frac{\sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i \delta(y_i - c)}{\sum_{i=1}^{N} \alpha_i \delta(y_i - c)} \tag{A8}$$

$$\boldsymbol{w}_{\text{CDB2}} = \tilde{\boldsymbol{w}}_{\text{CDB2}} / \|\tilde{\boldsymbol{w}}_{\text{CDB2}}\|_2 \tag{A9}$$

The operator $\boldsymbol{A}_{\text{BO}}$ does the mapping through the following equation:

$$\tilde{\boldsymbol{w}}_{\text{orth}} = \boldsymbol{w}_{\text{CDB2}} - \boldsymbol{w}_{\text{CDB1}} \boldsymbol{w}_{\text{CDB2}}^T \boldsymbol{w}_{\text{CDB1}} \tag{A10}$$

$$\boldsymbol{w}_{\text{orth}} = \tilde{\boldsymbol{w}}_{\text{orth}} / \|\tilde{\boldsymbol{w}}_{\text{orth}}\|_2 \tag{A11}$$

$$\hat{\theta^*} \approx \arg\max_{\theta} \boldsymbol{A}_{\text{ps}}(\cos\theta \boldsymbol{w}_{\text{CDB1}} + \sin\theta \boldsymbol{w}_{\text{CDB2}}) \tag{A12}$$

 doi:10.20944/preprints202502.0789.v2

$$\boldsymbol{w} = (\cos\hat{\theta^*}\boldsymbol{w}_{\text{CDB1}} + \sin\hat{\theta^*}\boldsymbol{w}_{\text{CDB2}}) \tag{A13}$$

Where the first two equations find the unit orthogonal vector to CDB1 by the Gram-Schmidt process. The orthogonalization process ensures an efficient search in the space of CDBs during the rotation. The last two equations estimate the optimal rotation angle using Bayesian optimization and the corresponding discriminant.

$ps = \boldsymbol{A}_{\text{ps}}(\boldsymbol{w}_{\text{CDB}}) : \mathbb{R}^{1\times m} \to \mathbb{R}$ is the operator that outputs the performance score given a classification discriminant, which projects all data onto the discriminant and performs the OOP search to find the score.

Appendix F. Classification Performance Evaluation

The datasets used in this study span standard image classification, medical image classification, and chemical property classification, most of which involve multiclass data. To comprehensively assess classification performance across these diverse tasks, we employed four key metrics: AUROC, AUPR, F-score, and AC-score. Accuracy was excluded due to its limitations in reflecting true performance, especially on imbalanced data. Though data augmentation can make data balanced, it introduces uncertainty due to the augmentation strategy and the data quality.

Since we do not prioritize any specific class (as designing a dataset-specific metric is beyond this study's scope), we perform two evaluations—one assuming each class as positive—and take the average. This approach applies to AUPR and F-score, whereas AUROC and AC-score, being symmetric about class labels, require only a single evaluation.

For multiclass prediction, a C-dimensional confusion matrix is obtained by comparing predicted labels with true labels. This is converted to C binary confusion matrices by each time taking one class as positive and all the others as negative, and the final evaluation is the average of evaluations on these individual confusion matrices.

To interpret this evaluation scheme, consider the MNIST dataset and the AUPR metric as an example. In binary classification, for the pair of digits "1" and "2", it calculates the sensitivity-recall for detecting "1" and the sensitivity-recall for detecting "2", then takes the average. For other pairs, the calculation follows the same pattern. In the multiclass scenario, it considers the sensitivity-recall for detecting "1" and the sensitivity-recall for detecting "not 1", then takes the average. The same approach applies to other pairs.

Individual Metrics

(a) AUROC

Area Under the Receiver Operating Characteristic curve involves the true positive rate (TPR) and false positive rate (FPR) as follows:

$$AUROC = \int_{x=0}^{1} TPR \times FPR^{-1}(x)dx \tag{A14}$$

This measure indicates how well the model distinguishes between classes. A high score means that the model can identify most positive cases with few false positives.

(b) AUPR

Area Under the Precision-Recall curve considers two complementary indices: precision ($prec$) and recall (rec), defined as:

$$AUPR = \int_{x=0}^{1} prec \times rec^{-1}(x)dx \tag{A15}$$

AUPR measures whether the classifier retrieves most positive cases.

 doi:10.20944/preprints202502.0789.v2

(c) F-score

F-score (using F1-score in this study) is the harmonic mean of precision and recall which contribute equally:

$$F_{score} = 2 \times \frac{prec \times rec}{prec + rec} \tag{A16}$$

(d) AC-score

An accuracy-related metric AC-score ([12]) addresses the limitations of traditional accuracy for imbalanced data. It imposes a stronger penalty for deviations from optimal performance on imbalanced data and is defined as:

$$AC_{score} = \frac{2 \times TPR \times TNR}{TPR + TNR} \tag{A17}$$

This metric assigns equal importance to both classes and penalizes imbalanced performance by ensuring that if one class is poorly classified, the overall score remains low. Thus, AC-score provides a more conservative but reliable evaluation of classifier performance.

(e) Performance-score (ps)

The proposed CDA classifier employs performance-dependent learning, guided by a performance score (ps):

$$ps = (F^{\text{pos}}_{score} + F^{\text{neg}}_{score} + AC_{score})/3 \tag{A18}$$

This formulation ensures a balanced consideration of both F-score and AC-score.

Though CDA as a generic classifier utilizes a relatively balanced metric, it can be customized to specific applications as needed.

Appendix G. Supplemental Performances on Linear Classification of Large-Scale Data

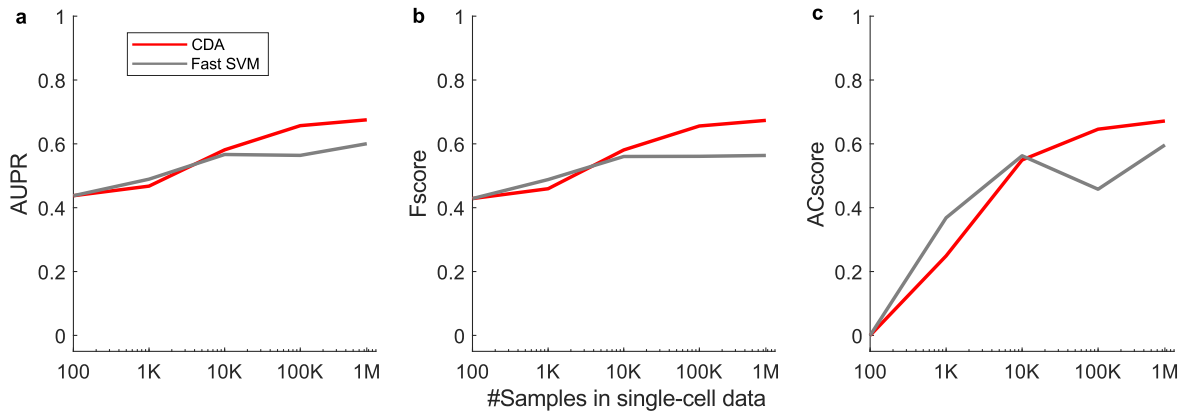

Figure A3. Test set performance with varying sizes of training samples in single-cell mouse brain data, evaluated by (a) AUPR, (b) Fscore and (c) ACscore. CDA: Centroid Discriminant Analysis; SVM: Support Vector Machine; AUPR: Area under precision-recall curve; ACscore: Accuracy score.

 doi:10.20944/preprints202502.0789.v2

Appendix H. Supplemental Training Speed Results

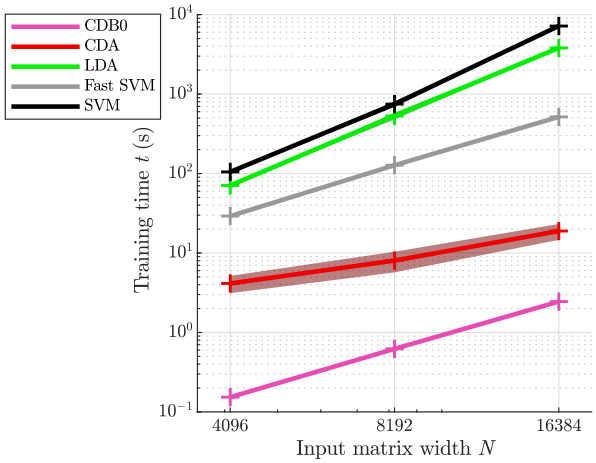

Figure A4. The training time with constant overhead kept. The training time of linear classifiers with increasing input matrix sizes. The number of features were set the same as number of samples N. Shaded area represents standard error. Log-scale is used for both axes to reveal the scalability of algorithms. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine

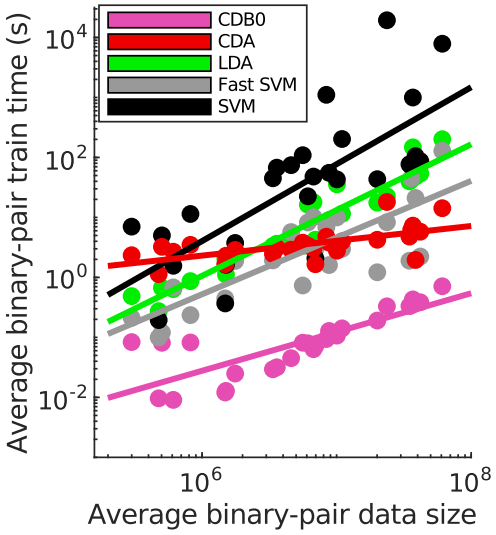

Figure A5. Average binary-pair training time on 27 real datasets with constant overhead kept. Both axes are presented on a logarithmic scale to highlight the scalability of the algorithms. Linear regression is applied to the logarithms of average binary-pair training time and that of average binary-pair data size. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine

Appendix I. Implementation Details

All data samples were converted into 1d representations without feature extraction or standardization throughout the paper, as the focus of this study is to compare the general performance of various classifiers on common tasks, rather than optimizing feature extraction for specific applications such as image classification.

 doi:10.20944/preprints202502.0789.v2

Appendix I.1. Linear Classifiers

Linear classifiers compared in this study include LDA, SVM, and fast SVM. Data were input to classifiers in their original values without standardization.

For CDA, the threshold for the coefficient of variation (CV) of performance-score was set to 0.001. 5-fold cross-validation is applied on the OOP search.

For LDA, the MATLAB toolbox function `fitcdiscr` was applied, with `pseudolinear` as the discriminant type, which solves the pseudo-inverse with SVD to adapt to poorly conditioned covariance matrices.

For SVM, the MATLAB toolbox function `fitcsvm` was applied, with sequential minimal optimization (SMO) as the optimizer. The cost function used the L2-regularized (margin) L1-loss (misclassification) form. The misclassification cost coefficient was set to $C = 1$ by default. The maximum number of iterations was linked to the number of training samples in the binary pair as $100 \times N_{\text{train}}$. The convergence criteria were set to the default values.

For fast SVM, the LIBLINEAR MATLAB `mex` function was applied. The cost function used the L2-regularized (margin) L1-loss (misclassification) form, and was optimized by dual coordinate descent (DCD). The misclassification cost coefficient was set to $C = 1$ by default. A bias term was added, and correspondingly, the data had an additional dimension with a value of 1. The convergence criteria were set to the default values. Training epochs were set to 300 in general tasks and 100 in large-scale single-cell test.

Hyperparameter tuning for LDA, SVM, and fast SVM was disabled, as the goal of this study was to compare linear classifiers fairly in a parameter-free condition. CDA, as well as CDB0, do not require parameter tuning.

Speed tests and performance test were from different computational conditions. To obtain classification performance in a reasonable time span, GPU was enabled for SVM, and multicore was enabled for CDA and LDA. To compare speed fairly, all linear classifiers used single-core computing mode.

Appendix I.2. Nonlinear Classifiers

For the experiments on the nonlinear kernel-based approaches, on SVHN subset and chemical Clintox, the data were first divided into train+validation and test set with 5:1 and 4:1 ratio respectively, then the train+validation set was further divided into train set and validation set with a 4:1 ratio.

The gaussian kernel parameter sigma was tuned on the validation set by Bayesian optimization with 30 sampling. For the tuned gaussian parameter, on SVHN subset gaussian CDA has $coeff = 0.2$ and gaussian SVM has $coeff = 0.134$; on ClinTox gaussian CDA has $coeff = 1.34$ and gaussian SVM has $coeff = 68.9$, where in gaussian kernel $\sigma^2 = coeff * \text{median}(\text{pairwiseSquareEuclideanDistance}(X_{\text{train}}))$.

For CDA, the threshold for the coefficient of variation (CV) of performance-score was set to 0.001. For SVM, libSVM was implemented with L2-regularized (margin) L1-loss (misclassification) cost function and with default misclassification cost coefficient $C = 1$.

Appendix J. Dataset Description

The datasets tested in this study encompass standard image classification, medical image classification, and chemical property prediction, described in details in Table. 1.

 doi:10.20944/preprints202502.0789.v2

Table A1. Dataset description

Dataset	#Samples	#Features	#Classes	Balancedness	Modality/source	Classification task
Standard images						
MNIST	70000	400	10	imbalanced	image	digits
USPS	9298	256	10	imbalanced	image	digits
EMNIST	145600	784	26	balanced	image	letters
CIFAR10	60000	3072	10	balanced	image	objects
SVHN	99289	3072	10	imbalanced	image	house numbers
flower	3670	1200	5	imbalanced	image	flowers
GTSRB	26635	1200	43	imbalanced	image	traffic signs
STL10	13000	2352	10	balanced	image	objects
FMNIST	70000	784	10	balanced	image	fashion objects
Medical images						
dermamnist	10015	2352	7	imbalanced	dermatoscope	dermal diseases
pneumoniamnist	5856	784	2	imbalanced	chest X-Ray	pneumonia
retinamnist	1600	2352	5	imbalanced	fundus camera	diabetic retinopathy
breastmnist	780	784	2	imbalanced	breast ultrasound	breast diseases
bloodmnist	17092	2352	8	imbalanced	blood cell microscope	blood diseases
organamnist	58830	784	11	imbalanced	abdominal CT	human organs
organcmnist	23583	784	11	imbalanced	abdominal CT	human organs
organsmnist	25211	784	11	imbalanced	abdominal CT	human organs
organmnist3d	1472	21952	11	imbalanced	abdominal CT	human organs
nodulemnist3d	1633	21952	2	imbalanced	chest CT	nodule malignancy
fracturemnist3d	1370	21952	3	imbalanced	chest CT	fracture types
adrenalmnist3d	1584	21952	2	imbalanced	shape from abdominal CT	adrenal gland mass
vesselmnist3d	1908	21952	2	imbalanced	shape from brain MRA	aneurysm
synapsemnist3d	1759	21952	2	imbalanced	electron microscope	excitatory/inhibitory
Chemical formula						
bace	1513	198	2	imbalanced	chemical formula	BACE1 enzyme
BBBP	2050	400	2	imbalanced	chemical formula	blood-brain barrier permeability
clintox	1484	339	2	imbalanced	chemical formula	clinical toxicity
HIV	41127	575	2	imbalanced	chemical formula	HIV drug activity
Large-scale single-cell sequencing data						
Mouse brain	1306127	27998	10	imbalanced	single-cell sequencing	cell type

Data processing: Image data were used in their original value flattened to 1d vectors; chemical formula were processed by simplified molecular input line entry system (SMILES) tokenized encoding. SMILES is a line notation for describing the structure of chemical entities using short ASCII strings. Chemical formula were stacked and aligned from the left since they have different SMILES lengths. The strings were mapped to natural numbers using a predefined dictionary, and those missing string positions were filled with 0. On these converted data we performed different classifiers.

 doi:10.20944/preprints202502.0789.v2

Appendix K. Supplemental Performances on Real Datasets of linear CDA

Appendix K.1. Binary Classification

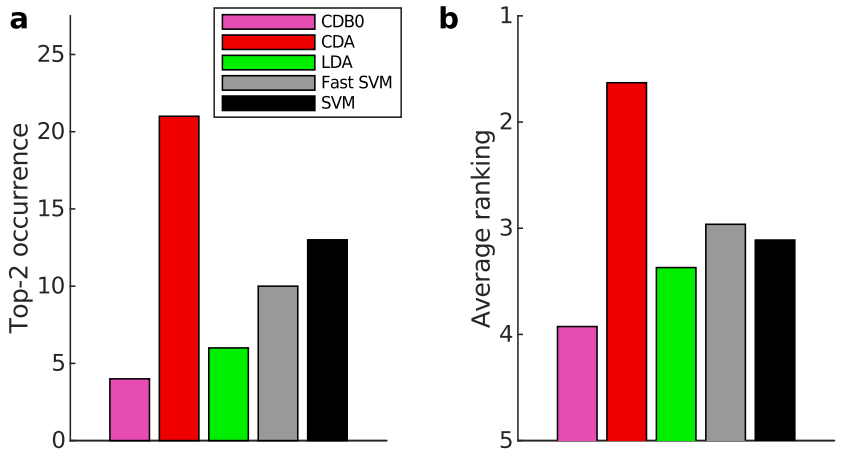

Figure A6. Binary classification performance on 27 real datasets according to AUROC. (a) Top-2 occurrences of algorithms. (b) Average ranking of algorithms. Error bars represent standard errors. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine.

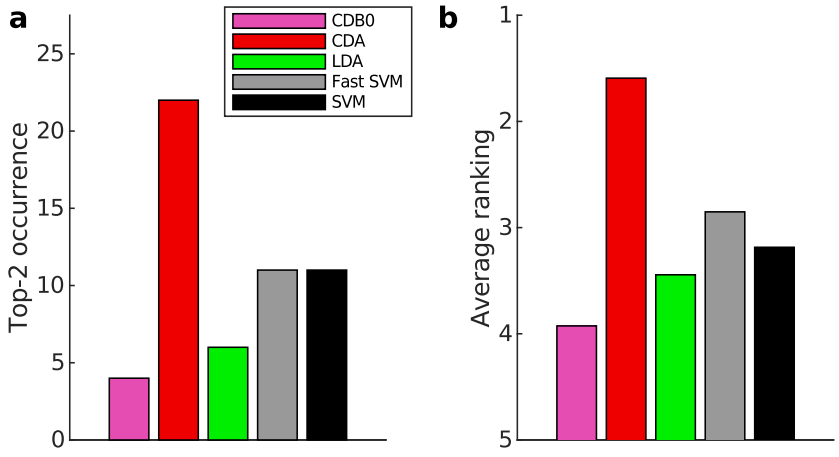

Figure A7. Binary classification performance on 27 real datasets according to AUPR. (a) Top-2 occurrences of algorithms. (b) Average ranking of algorithms. Error bars represent standard errors. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine.

 doi:10.20944/preprints202502.0789.v2

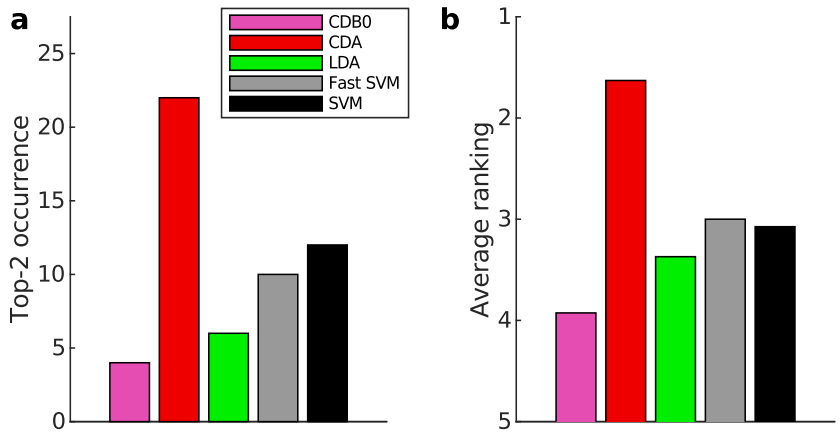

Figure A8. Binary classification performance on 27 real datasets according to F-score. (a) Top-2 occurrences of algorithms. (b) Average ranking of algorithms. Error bars represent standard errors. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine.

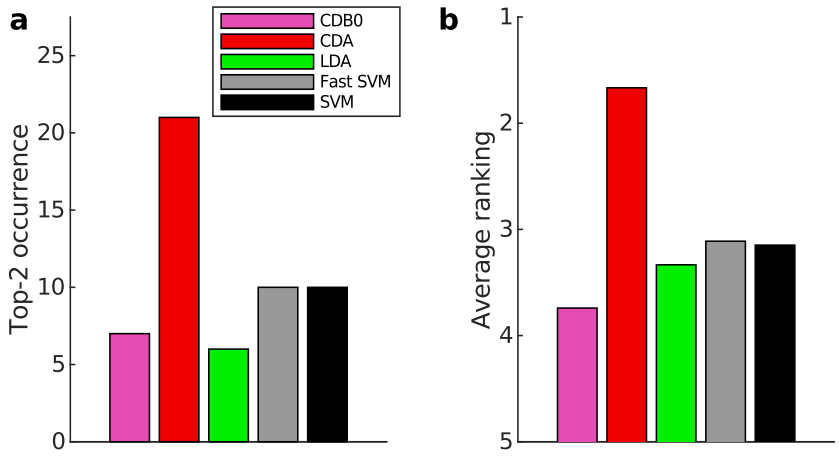

Figure A9. Binary classification performance on 27 real datasets according to AC-score. (a) Top-2 occurrences of algorithms. (b) Average ranking of algorithms. Error bars represent standard errors. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine.

 doi:10.20944/preprints202502.0789.v2

Appendix K.2. Multiclass Prediction

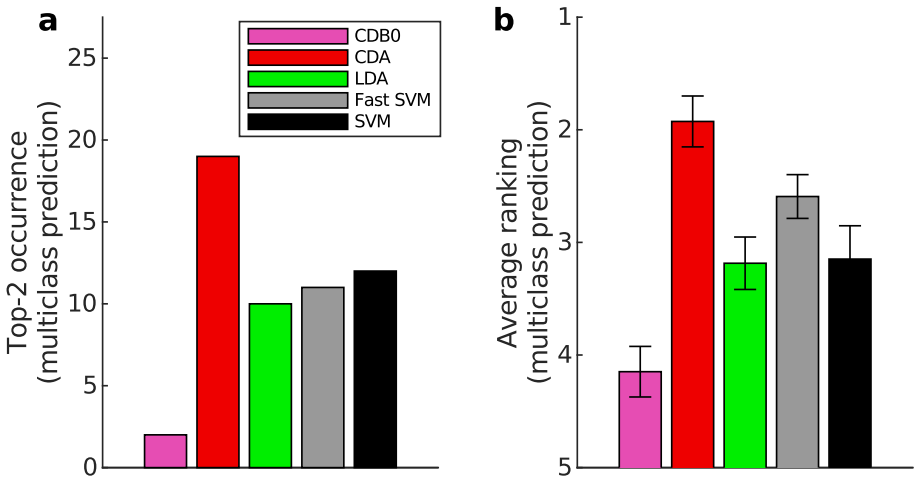

Figure A10. Multiclass prediction performance on 27 real datasets according to AUPR. (a) Top-2 occurrences of algorithms. (b) Average ranking of algorithms. Error bars represent standard errors. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine.

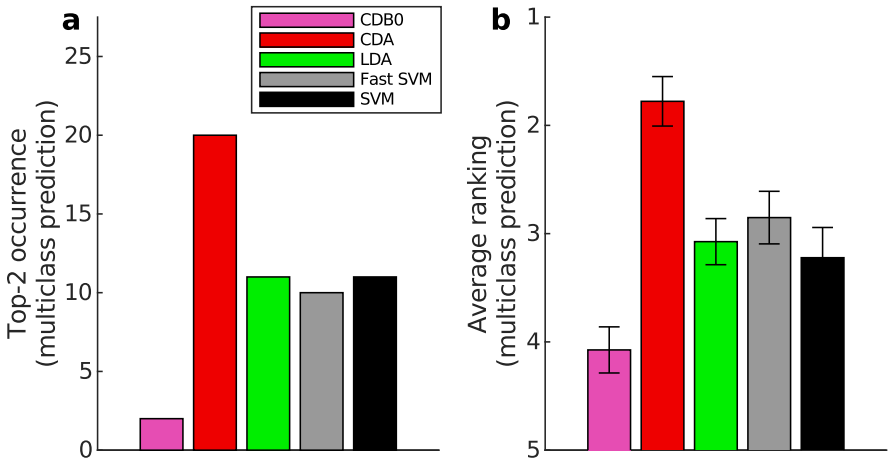

Figure A11. Multiclass prediction performance on 27 real datasets according to F-score. (a) Top-2 occurrences of algorithms. (b) Average ranking of algorithms. Error bars represent standard errors. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine.

 doi:10.20944/preprints202502.0789.v2

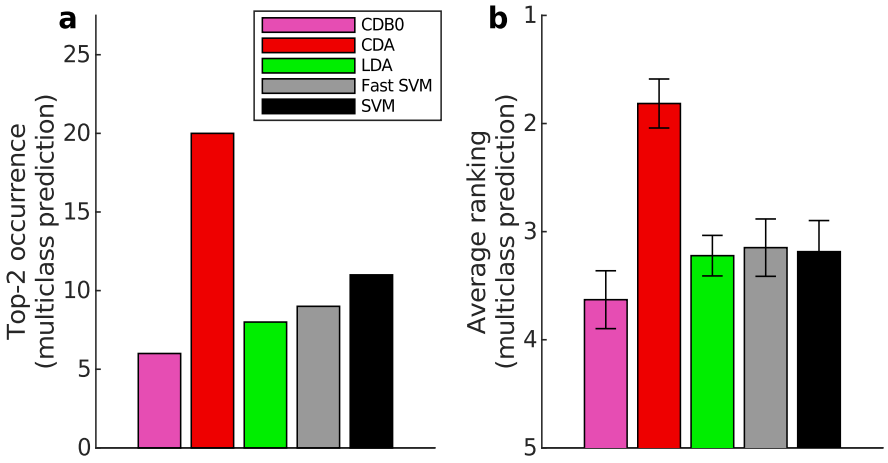

Figure A12. Multiclass prediction performance on 27 real datasets according to AC-score. (a) Top-2 occurrences of algorithms. (b) Average ranking of algorithms. Error bars represent standard errors. CDB0: Centroid Discriminant Basis 0; CDA: Centroid Discriminant Analysis; LDA: Linear Discriminant Analysis; SVM: Support Vector Machine.

Appendix L. Full Classification Performance on Real Datasets of Linear CDA

In the below tables we give full classification performance on 27 real datasets based on 4 metrics with binary classification (Table. 1-4) and multiclass prediction (Table. 5-8). In multiclass prediction, the performances on datasets with only two classes were filled with the one of binary classification.

 doi:10.20944/preprints202502.0789.v2

Appendix L.1. Binary Classification

Table A2. AUROC (Binary Classification Performance)

Dataset	CDB0	CDA	LDA	Fast SVM	SVM
Standard images					
MNIST	0.957±0.004	0.985±0.002	0.981±0.002	0.985±0.002	**0.986±0.002**
USPS	0.966±0.005	0.989±0.001	0.982±0.002	0.99±0.002	**0.99±0.002**
EMNIST	0.928±0.003	**0.972±0.001**	0.964±0.001	0.97±0.001	0.97±0.001
CIFAR10	0.696±0.01	**0.797±0.01**	0.741±0.01	0.754±0.01	0.787±0.01
SVHN	0.528±0.003	**0.667±0.005**	0.555±0.003	0.55±0.004	0.591±0.004
flower	0.703±0.02	**0.739±0.02**	0.571±0.01	0.71±0.03	0.71±0.03
GTSRB	0.767±0.003	0.972±0.001	0.942±0.002	0.995±0.0004	**0.995±0.0004**
STL10	0.723±0.02	**0.781±0.02**	0.667±0.01	0.758±0.02	0.761±0.02
FMNIST	0.937±0.01	0.975±0.006	0.973±0.006	0.976±0.006	**0.976±0.006**
Medical images					
dermamnist	0.682±0.01	**0.753±0.02**	0.684±0.02	0.676±0.02	0.703±0.02
pneumoniamnist	0.837±0	0.933±0	0.912±0	**0.941±0**	0.872±0
retinamnist	0.63±0.03	**0.662±0.04**	0.616±0.02	0.631±0.03	0.61±0.02
breastmnist	0.66±0	**0.763±0**	0.703±0	0.726±0	0.71±0
bloodmnist	0.89±0.02	0.947±0.01	0.898±0.02	**0.951±0.01**	0.947±0.01
organamnist	0.897±0.009	0.948±0.008	**0.95±0.008**	0.928±0.01	0.894±0.02
organcmnist	0.89±0.01	**0.925±0.01**	0.908±0.01	0.895±0.01	0.886±0.02
organsmnist	0.831±0.01	**0.886±0.01**	0.866±0.01	0.842±0.02	0.814±0.02
organmnist3d	0.924±0.01	0.957±0.008	0.953±0.008	**0.965±0.007**	0.7±0.02
nodulemnist3d	0.715±0	**0.781±0**	0.732±0	0.687±0	0.654±0
fracturemnist3d	**0.671±0.06**	0.556±0.03	0.525±0.04	0.576±0.007	0.553±0.02
adrenalmnist3d	0.653±0	**0.756±0**	0.692±0	0.697±0	0.727±0
vesselmnist3d	0.605±0	**0.685±0**	0.681±0	0.61±0	0.612±0
synapsemnist3d	0.539±0	**0.544±0**	0.508±0	0.518±0	NaN
Chemical formula					
bace	0.621±0	**0.705±0**	0.684±0	0.618±0	0.61±0
BBBP	0.711±0	**0.743±0**	0.693±0	0.667±0	0.705±0
clintox	**0.65±0**	0.575±0	0.543±0	0.517±0	0.508±0
HIV	0.6±0	**0.616±0**	0.537±0	0.51±0	0.512±0

 doi:10.20944/preprints202502.0789.v2

Table A3. AUPR (Binary Classification Performance)

Dataset	CDB0	CDA	LDA	Fast SVM	SVM
Standard images					
MNIST	0.957±0.004	0.985±0.002	0.981±0.002	0.985±0.002	**0.986±0.002**
USPS	0.966±0.005	0.989±0.001	0.983±0.002	0.990±0.002	**0.990±0.002**
EMNIST	0.928±0.003	**0.972±0.001**	0.964±0.001	0.970±0.001	0.970±0.001
CIFAR10	0.697±0.01	**0.797±0.01**	0.741±0.01	0.762±0.01	0.787±0.01
SVHN	0.528±0.003	**0.682±0.006**	0.559±0.003	0.577±0.005	0.634±0.006
flower	0.704±0.02	**0.741±0.02**	0.571±0.01	0.712±0.03	0.712±0.03
GTSRB	0.757±0.003	0.973±0.001	0.934±0.002	0.995±0.0003	**0.995±0.0003**
STL10	0.724±0.02	**0.782±0.02**	0.667±0.01	0.759±0.02	0.762±0.02
FMNIST	0.937±0.01	0.975±0.006	0.973±0.006	0.976±0.006	**0.976±0.006**
Medical images					
dermamnist	0.653±0.01	**0.743±0.02**	0.681±0.02	0.729±0.02	0.695±0.02
pneumoniamnist	0.817±0	0.931±0	0.922±0	**0.937±0**	0.871±0
retinamnist	0.614±0.03	**0.649±0.03**	0.612±0.02	0.641±0.02	0.607±0.02
breastmnist	0.653±0	**0.759±0**	0.690±0	0.743±0	0.706±0
bloodmnist	0.889±0.02	0.947±0.01	0.895±0.02	**0.953±0.01**	0.947±0.01
organamnist	0.902±0.009	0.950±0.007	**0.951±0.008**	0.929±0.01	0.895±0.02
organcmnist	0.901±0.009	**0.931±0.009**	0.908±0.01	0.893±0.01	0.883±0.02
organsmnist	0.839±0.01	**0.892±0.01**	0.867±0.01	0.840±0.02	0.813±0.02
organmnist3d	0.924±0.009	0.958±0.008	0.954±0.008	**0.965±0.007**	0.723±0.02
nodulemnist3d	0.7±0	**0.771±0**	0.745±0	0.695±0	0.637±0
fracturemnist3d	**0.663±0.05**	0.566±0.02	0.531±0.05	0.583±0.003	0.555±0.02
adrenalmnist3d	0.65±0	**0.774±0**	0.705±0	0.708±0	0.728±0
vesselmnist3d	0.582±0	0.671±0	**0.694±0**	0.627±0	0.659±0
synapsemnist3d	0.537±0	0.542±0	**0.544±0**	0.533±0	NaN
Chemical formula					
bace	0.620±0	**0.704±0**	0.685±0	0.643±0	0.615±0
BBBP	0.701±0	**0.747±0**	0.734±0	0.712±0	0.714±0
clintox	**0.602±0**	0.570±0	0.548±0	0.553±0	0.514±0
HIV	0.565±0	0.583±0	0.558±0	0.612±0	**0.632±0**

 doi:10.20944/preprints202502.0789.v2

Table A4. F-score (Binary Classification Performance)

Dataset	CDB0	CDA	LDA	Fast SVM	SVM
Standard images					
MNIST	0.957±0.004	0.985±0.002	0.981±0.002	0.985±0.002	**0.986±0.002**
USPS	0.966±0.005	0.989±0.001	0.983±0.002	0.99±0.002	**0.99±0.002**
EMNIST	0.928±0.003	**0.972±0.001**	0.964±0.001	0.97±0.001	0.97±0.001
CIFAR10	0.696±0.01	**0.797±0.01**	0.741±0.01	0.747±0.01	0.787±0.01
SVHN	0.523±0.003	**0.664±0.005**	0.555±0.003	0.51±0.01	0.577±0.006
flower	0.701±0.02	**0.738±0.02**	0.57±0.01	0.709±0.03	0.709±0.03
GTSRB	0.743±0.003	0.972±0.001	0.931±0.002	0.995±0.0003	**0.995±0.0003**
STL10	0.722±0.02	**0.781±0.02**	0.666±0.01	0.756±0.02	0.761±0.02
FMNIST	0.937±0.01	0.975±0.006	0.973±0.006	0.976±0.006	**0.976±0.006**
Medical images					
dermamnist	0.621±0.02	**0.736±0.02**	0.677±0.02	0.682±0.03	0.692±0.02
pneumoniamnist	0.812±0	0.931±0	0.921±0	**0.937±0**	0.871±0
retinamnist	0.594±0.02	**0.639±0.03**	0.61±0.02	0.611±0.04	0.606±0.02
breastmnist	0.651±0	**0.759±0**	0.684±0	0.739±0	0.705±0
bloodmnist	0.888±0.02	0.947±0.01	0.894±0.02	**0.952±0.01**	0.946±0.01
organamnist	0.899±0.01	0.949±0.008	**0.951±0.008**	0.923±0.02	0.893±0.02
organcmnist	0.896±0.01	**0.929±0.009**	0.908±0.01	0.892±0.02	0.882±0.02
organsmnist	0.834±0.01	**0.89±0.01**	0.866±0.01	0.834±0.02	0.811±0.02
organmnist3d	0.92±0.01	0.957±0.008	0.953±0.008	**0.964±0.007**	0.68±0.02
nodulemnist3d	0.695±0	**0.769±0**	0.743±0	0.694±0	0.622±0
fracturemnist3d	**0.651±0.05**	0.523±0.03	0.514±0.05	0.577±0.007	0.55±0.02
adrenalmnist3d	0.65±0	**0.771±0**	0.703±0	0.707±0	0.728±0
vesselmnist3d	0.56±0	0.669±0	**0.693±0**	0.623±0	0.638±0
synapsemnist3d	0.534±0	**0.539±0**	0.45±0	0.493±0	NaN
Chemical formula					
bace	0.619±0	**0.704±0**	0.684±0	0.569±0	0.607±0
BBBP	0.699±0	**0.747±0**	0.718±0	0.691±0	0.713±0
clintox	0.54±0	**0.569±0**	0.547±0	0.517±0	0.506±0
HIV	0.531±0	**0.562±0**	0.548±0	0.511±0	0.514±0

 doi:10.20944/preprints202502.0789.v2

Table A5. AC-score (Binary Classification Performance)

Dataset	CDB0	CDA	LDA	Fast SVM	SVM
Standard images					
MNIST	0.957±0.004	0.985±0.002	0.981±0.002	0.985±0.002	**0.986±0.002**
USPS	0.966±0.005	0.989±0.001	0.982±0.002	0.99±0.002	**0.99±0.002**
EMNIST	0.927±0.003	**0.972±0.001**	0.964±0.001	0.97±0.001	0.97±0.001
CIFAR10	0.694±0.01	**0.796±0.01**	0.74±0.01	0.725±0.02	0.787±0.01
SVHN	0.524±0.002	**0.614±0.004**	0.499±0.008	0.352±0.03	0.438±0.02
flower	0.698±0.02	**0.733±0.02**	0.567±0.01	0.701±0.03	0.7±0.03
GTSRB	0.753±0.003	0.97±0.002	0.941±0.002	0.995±0.0004	**0.995±0.0004**
STL10	0.72±0.01	**0.779±0.02**	0.664±0.01	0.75±0.02	0.76±0.02
FMNIST	0.936±0.01	0.975±0.006	0.973±0.006	0.975±0.006	**0.976±0.006**
Medical images					
dermamnist	0.658±0.02	**0.72±0.02**	0.608±0.04	0.535±0.05	0.654±0.03
pneumoniamnist	0.837±0	0.932±0	0.908±0	**0.94±0**	0.868±0
retinamnist	0.62±0.03	**0.639±0.05**	0.567±0.03	0.513±0.07	0.561±0.03
breastmnist	0.641±0	**0.751±0**	0.698±0	0.682±0	0.691±0
bloodmnist	0.889±0.02	0.946±0.01	0.897±0.02	**0.949±0.01**	0.947±0.01
organamnist	0.892±0.01	0.946±0.008	**0.949±0.008**	0.919±0.02	0.882±0.02
organcmnist	0.881±0.01	**0.92±0.01**	0.907±0.01	0.893±0.02	0.886±0.02
organsmnist	0.822±0.01	**0.88±0.01**	0.862±0.01	0.833±0.02	0.801±0.02
organmnist3d	0.918±0.01	0.956±0.008	0.952±0.008	**0.964±0.007**	0.602±0.03
nodulemnist3d	0.707±0	**0.773±0**	0.693±0	0.636±0	0.651±0
fracturemnist3d	**0.668±0.06**	0.38±0.1	0.351±0.1	0.491±0.06	0.45±0.08
adrenalmnist3d	0.602±0	**0.718±0**	0.631±0	0.642±0	0.694±0
vesselmnist3d	0.547±0	**0.615±0**	0.58±0	0.43±0	0.405±0
synapsemnist3d	0.498±0	**0.506±0**	0.0612±0	0.189±0	NaN
Chemical formula					
bace	0.62±0	**0.704±0**	0.678±0	0.483±0	0.575±0
BBBP	0.693±0	**0.715±0**	0.603±0	0.553±0	0.66±0
clintox	**0.634±0**	0.365±0	0.238±0	0.0869±0	0.0868±0
HIV	**0.471±0**	0.465±0	0.159±0	0.0407±0	0.0473±0

 doi:10.20944/preprints202502.0789.v2

Appendix L.2. Multiclass Prediction

Table A6. AUROC (Multiclass Prediction Performance)

Dataset	CDB0	CDA	LDA	Fast SVM	SVM
Standard images					
MNIST	0.897±0.01	0.963±0.005	0.958±0.006	0.965±0.005	**0.966±0.005**
USPS	0.914±0.01	0.971±0.004	0.969±0.006	**0.974±0.005**	0.973±0.005
EMNIST	0.773±0.01	**0.896±0.008**	0.879±0.009	0.891±0.008	0.892±0.008
CIFAR10	0.599±0.02	**0.671±0.02**	0.627±0.01	0.641±0.02	0.663±0.02
SVHN	0.522±0.006	**0.638±0.01**	0.531±0.007	0.536±0.01	0.558±0.01
flower	0.61±0.03	**0.666±0.03**	0.554±0.02	0.632±0.03	0.632±0.03
GTSRB	0.589±0.01	0.878±0.01	0.821±0.02	0.982±0.003	**0.983±0.003**
STL10	0.607±0.02	**0.655±0.02**	0.596±0.02	0.648±0.02	0.653±0.02
FMNIST	0.836±0.03	0.917±0.02	0.92±0.02	0.924±0.02	**0.924±0.02**
Medical images					
dermamnist	0.614±0.03	**0.658±0.03**	0.588±0.02	0.595±0.03	0.622±0.02
pneumoniamnist	0.837±0	0.933±0	0.912±0	**0.941±0**	0.872±0
retinamnist	0.575±0.04	**0.622±0.04**	0.592±0.03	0.596±0.04	0.578±0.03
breastmnist	0.66±0	**0.763±0**	0.703±0	0.726±0	0.71±0
bloodmnist	0.789±0.04	0.88±0.03	0.817±0.03	**0.882±0.03**	0.881±0.03
organamnist	0.815±0.03	**0.888±0.02**	0.885±0.03	0.85±0.04	0.795±0.05
organcmnist	0.809±0.03	**0.869±0.03**	0.833±0.03	0.811±0.04	0.795±0.04
organsmnist	0.694±0.03	**0.761±0.03**	0.735±0.03	0.7±0.03	0.681±0.04
organmnist3d	0.867±0.03	0.913±0.02	0.903±0.03	**0.924±0.02**	0.636±0.03
nodulemnist3d	0.715±0	**0.781±0**	0.732±0	0.687±0	0.654±0
fracturemnist3d	**0.622±0.04**	0.518±0.01	0.554±0.04	0.574±0.004	0.554±0.02
adrenalmnist3d	0.653±0	0.756±0	0.9±0	0.928±0	**0.947±0**
vesselmnist3d	0.605±0	**0.685±0**	0.681±0	0.61±0	0.612±0
synapsemnist3d	0.539±0	**0.544±0**	0.508±0	0.518±0	NaN
Chemical formula					
bace	0.621±0	**0.705±0**	0.684±0	0.618±0	0.61±0
BBBP	0.711±0	**0.743±0**	0.693±0	0.667±0	0.705±0
clintox	**0.65±0**	0.575±0	0.543±0	0.517±0	0.508±0
HIV	0.6±0	**0.616±0**	0.537±0	0.51±0	0.512±0

 doi:10.20944/preprints202502.0789.v2

Table A7. AUPR (Multiclass Prediction Performance)

Dataset	CDB0	CDA	LDA	Fast SVM	SVM
Standard images					
MNIST	0.897±0.01	0.963±0.004	0.958±0.005	0.965±0.004	**0.966±0.004**
USPS	0.918±0.01	0.971±0.005	0.969±0.005	0.974±0.004	**0.973±0.005**
EMNIST	0.774±0.01	**0.896±0.008**	0.88±0.009	0.891±0.008	0.892±0.008
CIFAR10	0.6±0.01	**0.67±0.01**	0.627±0.01	0.64±0.02	0.663±0.02
SVHN	0.528±0.009	**0.666±0.01**	0.533±0.006	0.546±0.005	0.597±0.005
flower	0.611±0.02	**0.665±0.02**	0.554±0.02	0.631±0.02	0.632±0.02
GTSRB	0.619±0.01	0.891±0.01	0.805±0.01	0.981±0.003	**0.981±0.003**
STL10	0.604±0.02	**0.653±0.02**	0.598±0.02	0.648±0.02	0.651±0.02
FMNIST	0.836±0.03	0.916±0.02	0.92±0.02	0.923±0.02	**0.924±0.02**
Medical images					
dermamnist	0.592±0.02	**0.645±0.02**	0.603±0.02	0.608±0.03	0.613±0.02
pneumoniamnist	0.817±0	0.931±0	0.922±0	**0.937±0**	0.871±0
retinamnist	0.568±0.03	**0.621±0.04**	0.594±0.03	0.589±0.04	0.577±0.03
breastmnist	0.653±0	**0.759±0**	0.69±0	0.743±0	0.706±0
bloodmnist	0.785±0.04	0.878±0.03	0.818±0.03	**0.89±0.03**	0.88±0.03
organamnist	0.82±0.03	0.892±0.02	**0.889±0.02**	0.851±0.03	0.803±0.05
organcmnist	0.818±0.03	**0.877±0.02**	0.838±0.03	0.811±0.04	0.792±0.04
organsmnist	0.697±0.02	**0.764±0.03**	0.741±0.03	0.703±0.03	0.688±0.04
organmnist3d	0.867±0.02	0.913±0.02	0.904±0.03	**0.924±0.02**	0.668±0.04
nodulemnist3d	0.7±0	**0.771±0**	0.745±0	0.695±0	0.637±0
fracturemnist3d	**0.617±0.04**	0.526±0.02	0.553±0.04	0.578±0.004	0.555±0.02
adrenalmnist3d	0.65±0	**0.774±0**	0.911±0	0.939±0	0.949±0
vesselmnist3d	0.582±0	0.671±0	**0.694±0**	0.627±0	0.659±0
synapsemnist3d	0.537±0	0.542±0	**0.544±0**	0.533±0	NaN
Chemical formula					
bace	0.62±0	**0.704±0**	0.685±0	0.643±0	0.615±0
BBBP	0.701±0	**0.747±0**	0.734±0	0.712±0	0.714±0
clintox	**0.602±0**	0.57±0	0.548±0	0.553±0	0.514±0
HIV	0.565±0	0.583±0	0.558±0	0.612±0	**0.632±0**

 doi:10.20944/preprints202502.0789.v2

Table A8. F-score (Multiclass Prediction Performance)

Dataset	CDB0	CDA	LDA	Fast SVM	SVM
Standard images					
MNIST	0.896±0.01	0.963±0.004	0.958±0.005	0.965±0.004	**0.966±0.004**
USPS	0.917±0.01	0.971±0.005	0.969±0.005	0.974±0.004	**0.973±0.005**
EMNIST	0.773±0.01	**0.896±0.008**	0.879±0.009	0.891±0.008	0.892±0.008
CIFAR10	0.59±0.01	**0.67±0.01**	0.627±0.01	0.634±0.02	0.663±0.02
SVHN	0.509±0.006	**0.649±0.01**	0.529±0.005	0.526±0.006	0.55±0.01
flower	0.599±0.02	**0.662±0.02**	0.553±0.02	0.629±0.02	0.63±0.02
GTSRB	0.586±0.01	0.886±0.01	0.782±0.01	0.98±0.003	**0.98±0.003**
STL10	0.601±0.02	**0.653±0.02**	0.597±0.02	0.646±0.02	0.651±0.02
FMNIST	0.835±0.03	0.916±0.02	0.919±0.02	0.923±0.02	**0.924±0.02**
Medical images					
dermamnist	0.571±0.02	**0.639±0.02**	0.599±0.02	0.602±0.03	0.61±0.02
pneumoniamnist	0.812±0	0.931±0	0.921±0	**0.937±0**	0.871±0
retinamnist	0.547±0.04	**0.615±0.04**	0.592±0.03	0.579±0.04	0.577±0.03
breastmnist	0.651±0	**0.759±0**	0.684±0	0.739±0	0.705±0
bloodmnist	0.779±0.04	0.877±0.03	0.817±0.03	**0.887±0.03**	0.88±0.03
organamnist	0.812±0.03	0.889±0.02	**0.889±0.02**	0.847±0.04	0.798±0.05
organcmnist	0.811±0.03	**0.874±0.02**	0.837±0.03	0.809±0.04	0.792±0.04
organsmnist	0.685±0.02	**0.76±0.03**	0.74±0.03	0.698±0.03	0.684±0.04
organmnist3d	0.862±0.02	0.913±0.02	0.903±0.03	**0.922±0.02**	0.627±0.04
nodulemnist3d	0.695±0	**0.769±0**	0.743±0	0.694±0	0.622±0
fracturemnist3d	**0.601±0.04**	0.486±0.02	0.547±0.04	0.575±0.002	0.552±0.02
adrenalmnist3d	0.65±0	**0.771±0**	0.911±0	0.939±0	0.949±0
vesselmnist3d	0.56±0	0.669±0	**0.693±0**	0.623±0	0.638±0
synapsemnist3d	0.534±0	**0.539±0**	0.45±0	0.493±0	NaN
Chemical formula					
bace	0.619±0	**0.704±0**	0.684±0	0.569±0	0.607±0
BBBP	0.699±0	**0.747±0**	0.718±0	0.691±0	0.713±0
clintox	0.54±0	**0.569±0**	0.547±0	0.517±0	0.506±0
HIV	0.531±0	**0.562±0**	0.548±0	0.511±0	0.514±0

 doi:10.20944/preprints202502.0789.v2

Table A9. AC-score (Multiclass Prediction Performance)

Dataset	CDB0	CDA	LDA	Fast SVM	SVM
Standard images					
MNIST	0.888±0.01	0.962±0.005	0.956±0.006	0.964±0.005	**0.965±0.005**
USPS	0.907±0.01	0.97±0.005	0.968±0.007	0.974±0.005	**0.973±0.005**
EMNIST	0.708±0.02	**0.884±0.01**	0.863±0.01	0.878±0.01	0.879±0.01
CIFAR10	0.404±0.05	**0.562±0.03**	0.48±0.03	0.491±0.05	0.548±0.03
SVHN	0.223±0.05	**0.478±0.03**	0.241±0.04	0.223±0.06	0.247±0.05
flower	0.492±0.07	**0.597±0.05**	0.417±0.05	0.545±0.05	0.546±0.05
GTSRB	0.296±0.03	0.853±0.02	0.762±0.03	0.981±0.004	**0.982±0.004**
STL10	0.425±0.05	**0.523±0.05**	0.413±0.04	0.512±0.05	0.521±0.05
FMNIST	0.801±0.05	0.909±0.02	0.912±0.02	0.916±0.02	**0.917±0.02**
Medical images					
dermamnist	0.439±0.09	**0.527±0.07**	0.352±0.07	0.348±0.1	0.462±0.06
pneumoniamnist	0.837±0	0.932±0	0.908±0	**0.94±0**	0.868±0
retinamnist	0.377±0.1	**0.506±0.07**	0.444±0.08	0.401±0.1	0.407±0.1
breastmnist	0.641±0	**0.751±0**	0.698±0	0.682±0	0.691±0
bloodmnist	0.742±0.05	0.864±0.04	0.784±0.04	0.865±0.03	**0.866±0.03**
organamnist	0.771±0.05	0.872±0.03	**0.868±0.03**	0.814±0.05	0.717±0.08
organcmnist	0.766±0.04	**0.848±0.03**	0.797±0.04	0.76±0.05	0.733±0.06
organsmnist	0.58±0.06	**0.69±0.05**	0.653±0.04	0.589±0.06	0.533±0.08
organmnist3d	0.845±0.04	0.902±0.03	0.89±0.03	**0.914±0.03**	0.434±0.09
nodulemnist3d	0.707±0	**0.773±0**	0.693±0	0.636±0	0.651±0
fracturemnist3d	**0.569±0.09**	0.279±0.05	0.425±0.2	0.5±0.07	0.451±0.1
adrenalmnist3d	0.602±0	**0.718±0**	0.894±0	0.924±0	0.946±0
vesselmnist3d	0.547±0	**0.615±0**	0.58±0	0.43±0	0.405±0
synapsemnist3d	0.498±0	**0.506±0**	0.0612±0	0.189±0	NaN
Chemical formula					
bace	0.62±0	**0.704±0**	0.678±0	0.483±0	0.575±0
BBBP	0.693±0	**0.715±0**	0.603±0	0.553±0	0.66±0
clintox	**0.634±0**	0.365±0	0.238±0	0.0869±0	0.0868±0
HIV	**0.471±0**	0.465±0	0.159±0	0.0407±0	0.0473±0

Appendix M. Computation Conditions

The simulations were performed on a Lenovo P620 workstation (a AMD Ryzen Threadripper PRO 3995WX, 64 CPU cores at maximum 4.308GHz, 256GB memory). The large-scale tests on single-cell data were performed on a server (two AMD EPYC 7H12 CPUs, each with 64 CPU cores at maximum 2.6GHz, 4096 GB memory), but in this task single-core computing was enabled. MATLAB 2023b was used as the platform to run the main algorithms.

Appendix N. Licensing

The code packages libSVM and liblinear are with BSD 3-Clause license. Matlab is with a paid proprietary license.

Appendix O. Code Availability

We released the Matlab and Python code for the CDA algorithm, on https://anonymous.4open.science/r/Centroid_discriminant_Analysis-D878

References

1. Varoquaux, G.; Raamana, P.R.; Engemann, D.A.; Hoyos-Idrobo, A.; Schwartz, Y.; Thirion, B. Assessing and tuning brain decoders: Cross-validation, caveats, and guidelines. *NeuroImage* **2017**, *145*. https://doi.org/10.1016/j.neuroimage.2016.10.038.
2. Yuan, G.X.; Ho, C.H.; Lin, C.J. Recent advances of large-scale linear classification. *Proceedings of the IEEE* **2012**, *100*. https://doi.org/10.1109/JPROC.2012.2188013.

 doi:10.20944/preprints202502.0789.v2

3. Schulz, M.A.; Yeo, B.T.; Vogelstein, J.T.; Mourao-Miranada, J.; Kather, J.N.; Kording, K.; Richards, B.; Bzdok, D. Different scaling of linear models and deep learning in UKBiobank brain images versus machine-learning datasets. *Nature Communications* **2020**, *11*. https://doi.org/10.1038/s41467-020-18037-z.
4. Varoquaux, G.; Cheplygina, V. Machine learning for medical imaging: methodological failures and recommendations for the future. *npj Digital Medicine* **2022**, *5*. https://doi.org/10.1038/s41746-022-00592-y.
5. Duda, R.O.; Hart, P.E.; Stork, D.G. Pattern classification, second edition. *NY: Wiley-Interscience* **2001**.
6. Cai, D.; He, X.; Han, J. Training linear discriminant analysis in linear time. In Proceedings of the Proceedings - International Conference on Data Engineering, 2008. https://doi.org/10.1109/ICDE.2008.4497429.
7. Cortes, C.; Vapnik, V. Support-Vector Networks. *Machine Learning* **1995**, *20*. https://doi.org/10.1023/A:1022627411411.
8. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* **2008**, *9*.
9. Minsky, M.L.; Papert, S. *Perceptrons, Expanded Edition An Introduction to Computational Geometry*; MIT Press, 1969.
10. Panda, N.R. A Review on Logistic Regression in Medical Research. *National Journal of Community Medicine* **2022**, *13*, 265–270. https://doi.org/10.55489/njcm.134202222.
11. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **1936**, *7*. https://doi.org/10.1111/j.1469-1809.1936.tb02137.x.
12. Wu, Y.; Cannistraci, C.V. Accuracy Score for Evaluation of Classification on Imbalanced Data, 2025. Preprints, https://doi.org/10.20944/preprints202501.2178.v1.
13. Frazier, P.I. Bayesian Optimization. In Proceedings of the INFORMS TutORials in Operations Research, 2018. https://doi.org/10.1287/educ.2018.0188.
14. Allwein, E.L.; Schapire, R.E.; Singer, Y. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research* **2001**, *1*.
15. Acevedo, A.; Duran, C.; Kuo, M.J.; Ciucci, S.; Schroeder, M.; Cannistraci, C.V. Measuring Group Separability in Geometrical Space for Evaluation of Pattern Recognition and Dimension Reduction Algorithms. *IEEE Access* **2022**, *10*, 22441–22471. https://doi.org/10.1109/access.2022.3152789.
16. Acevedo, A.; Wu, Y.; Traversa, F.L.; Cannistraci, C.V. Geometric separability of mesoscale patterns in embedding representation and visualization of multidimensional data and complex networks. *PLOS Complex Systems* **2024**, *1*, 1–28. https://doi.org/10.1371/journal.pcsy.0000012.
17. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. *Technical Report TR-2009, Computer Science Department, University of Toronto, Toronto* **2009**. https://doi.org/10.1.1.222.9220.
18. 10x Genomics. 1.3 Million Brain Cells from E18 Mice. https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3.0/1M_neurons, 2017.
19. Coates, A.; Lee, H.; Ng, A.Y. An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the Journal of Machine Learning Research, 2011, Vol. 15.
20. Cohen, G.; Afshar, S.; Tapson, J.; Schaik, A.V. EMNIST: Extending MNIST to handwritten letters. In Proceedings of the Proceedings of the International Joint Conference on Neural Networks, 2017, Vol. 2017-May. https://doi.org/10.1109/IJCNN.2017.7966217.
21. Hull, J.J. A Database for Handwritten Text Recognition Research. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1994**, *16*. https://doi.org/10.1109/34.291440.
22. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*. https://doi.org/10.1109/5.726791.
23. Netzer, Y.; Wang, T. Reading digits in natural images with unsupervised feature learning. *Nips* **2011**.
24. Nilsback, M.E.; Zisserman, A. Automated flower classification over a large number of classes. In Proceedings of the Proceedings - 6th Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2008, 2008. https://doi.org/10.1109/ICVGIP.2008.47.
25. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In Proceedings of the Proceedings of the International Joint Conference on Neural Networks, 2011. https://doi.org/10.1109/IJCNN.2011.6033395.
26. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017. arXiv.
27. Yang, J.; Shi, R.; Wei, D.; Liu, Z.; Zhao, L.; Ke, B.; Pfister, H.; Ni, B. MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data* **2023**, *10*. https://doi.org/10.1038/s41597-022-01721-8.

 doi:10.20944/preprints202502.0789.v2

28. Wu, Z.; Ramsundar, B.; Feinberg, E.N.; Gomes, J.; Geniesse, C.; Pappu, A.S.; Leswing, K.; Pande, V. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science* **2018**, *9*. https://doi.org/10.1039/c7sc02664a.