# Semi-Supervised Discriminant Analysis with Relative Distance: Integration with a MOO Approach

Rakesh Kumar Sanodiya
*Computer Science and Engineering*
*Indian Institute of Technology Patna*
Patna, India
rakesh.pcs16@iitp.ac.in

Sriparna Saha
*Computer Science and Engineering*
*Indian Institute of Technology Patna*
Patna, India
sriparna@iitp.ac.in

Jimson Mathew
*Computer Science and Engineering*
*Indian Institute of Technology Patna*
Patna, India
jimson@iitp.ac.in

Michelle Davies Thalakottur
*Computer Engineering*
*MKSSS Cummins College of Engineering*
Pune, India
michelledaviest@gmail.com

Utkarshinee Aadya
*Mathematics*
*Birla Institute of Technology Mesra*
Ranchi, India
utkarshinee.aadya@gmail.com

***Abstract*—In many real-world applications, most of the data such as videos, genetic information, etc. resides in a high dimensional space. Before performing classification, we require to project the data from the high dimensional space to a lower dimensional space without losing too much information. Linear discriminant analysis (LDA) is one of the most widely used methods for dimensionality reduction, that maximizes the ratio of the between-class scatter and total data scatter in the projected space using the labelled information. However, in the real world, labelled information is hardly ever available in large quantities, but an abundant amount of unlabelled data is available. In this paper, we propose a Semi-Supervised Discriminant Analysis method called SSDARD, which considers the unlabelled information in the form of a k-NN graph. Different from the existing semi-supervised dimensionality reduction algorithms, our algorithm is more consistent in propagating the label information from labelled data to unlabelled data due to the use of relative distance function instead of normal Euclidean distance function to generate the k-NN graph. To find an appropriate relative distance function, we use pairwise constraints generated from labelled data and satisfy them using Bregman projection. Since the projection is not orthogonal, we require an appropriate subset of constraints. In order to select such subset of constraints, we have further developed a framework called MO-SSDARD, which uses an evolutionary algorithm while optimizing various cluster validity indices simultaneously. The experimental results on various datasets show that our proposed method is superior to various methods concerning various validity indices.**

***Index Terms*—Semi-supervised classification, Dimensionality reduction, Metric learning, Bregman projection, Evolutionary algorithm**

## I. INTRODUCTION

In many fields of machine learning applications, such as object detection, face recognition, image classification and activity recognition, we are often confronted with high dimensional data (1). In order to process such high dimensional data, we need to first project the data into lower dimensions while conveying the same information, to boost the performance of the system. Dimensionality Reduction (DR) is one of the most widely used methods for projecting the data from a high dimensional space to lower dimensional one while ensuring that it conveys similar information (2). In the literature survey, many useful methods for dimensionality reduction have been introduced and can be classified into three categories: supervised, semi-supervised and unsupervised (3). Supervised discriminant analysis uses only labelled information, semi-supervised uses both labelled and unlabelled information and unsupervised learning purely depends on the unlabelled information for performing the projection.

In supervised learning, Linear Discriminant Analysis (LDA) (4), Locality Preserving Projection (LPP) (5) and Regularized Coplanar Discriminant Analysis (RCDA) (1) are the most popular methods. LDA, a supervised method, ensures maximum class separability by maximizing the ratio of between-class variance to within-class variance. Locality Preserving Projection (LPP) is an improvement on the Principal Component Analysis (PCA) (6). It forms a graph of the training data domain and with it, tries to capture the neighborhood information of the data to represent the data accurately. The Laplacian of this graph is used to project this data to a subspace while still preserving the fundamental similarities and dissimilarities of the data. Additionally, the structure of the manifold that the data exists in is also preserved through this graph. RCDA aims to seek such a linear projection matrix that minimizes the error of the within class linear representation while keeping the error of the between-class linear representation constant.

In real-world applications, the obtained data may not be labelled because labelling requires more effort and time. On the other hand, we have an abundant amount of unlabelled data. To cope with this problem, Semi-supervised Discriminant Analysis (SDA) (7) has been proposed, which utilizes both labelled and unlabelled data at the same time to perform the projection. While using the unlabelled data, SDA suffers from the problem that it ignores the relationships between classes and only exploits the local neighborhood information. Zhang et al. (8) proposed Semi-Supervised Discriminant Analysis using robust path based similarity (SSDAR), which considers the global structure of the data to define the neighborhood relationships of data points. Nie et al. (9) proposed novel Semi-supervised Orthogonal Discriminant Analysis via label propagation called SODA, which propagates the information from labelled data to unlabelled data. Huang et al. (10) presented globally-locality preserving projections for bio-metric data dimensionality reduction (GLPP) algorithm, which divides the manifold of the data into two sub-manifolds of dynamic factors and static factors to precisely capture the manifold structure among the noisy samples.

However, this Dimensionality Reduction method has its own limitations. One of the major problems is a failure to capture the intrinsic geometry of the manifold in which the data exists. Euclidean distance formula and Mahalanobis distance formula have been used in previous related works to represent this geometry. But Euclidean distance formula does not perform well with non-linear subspace and Mahalanobis distance formula is unable to capture the manifold completely. Another problem is that the Dimensionality Reduction algorithms do not perform well with data sets where only a little amount of labelled data is present. However, in reality, we are often encountered with such data sets. Also, some of these algorithms do not work well with non-linear data. Apart from these, there are also some semi-supervised metric learning methods which can be used for Dimensionality Reduction. The main goal of the metric learning methods is to project the data in such a space where the distance between similar data points is minimized while the distance between dis-similar data points is kept far apart. DML-dc (11) is a supervised metric learning algorithm for nearest neighbor classification that uses a difference of convex function (DC) programming to overcome the non-convexity generated by the ramp loss function. DMLMJ (12) aims at learning a distance metric in which the Jeffery Divergence between two Gaussian distributions derived from local constraints is maximized. u-LMNN (13), based on LMNN, finds a distance metric for Universum examples that are not required to have the same distribution as the training data. RDML (14), a supervised approach, uses the generalization error of metric learning for handling high dimensional data.

Unsupervised algorithms like Principal Component Analysis (PCA) (6) works purely on unlabelled data, but its performance is not as good as supervised algorithms since it performs well only for data that exists on a linear subspace rather than on a manifold. Kernel PCA projects this non-linear data to a higher dimensional space, thereby making it linear. However, with Kernel PCA, there is the added problem of the curse of dimensionality that it requires more computational power and more data samples.

The major contributions of the current work are as follows:

- Our proposed algorithm, MO-SSDARD, exploits the underlying structure with a small number of features using relative distance constraints and is capable of performing well even if we have very little labelled data.
- There are two types of relative distance constraints, equality and inequality constraints. For generating the equality and inequality relative constraints, only a few labelled data points are required which makes our approach practically useful in real-life scenarios where the dimension of the data set is much higher than the amount of labelled data available.
- Since we use a Gaussian kernel to generate the initial distance matrix, our approach works for both linearly and non-linearly separable data.
- We have compared the performance of MO-SSDARD with other algorithms on different benchmark datasets and our results have shown that our proposed algorithm achieves greater performance and accuracy compared to other state-of-the-art algorithms.

## II. Problem Statement

There are many semi-supervised methods which exploit local neighborhood information using normalized Euclidean distance to generate the k-NN graph. However, the Euclidean distance formula may not be able to capture the global structure of the data and can lead to a decrease in the performance of the algorithm. In many real-world applications, the Euclidean distance may not be fit to capture the intrinsic dissimilarity and similarity between the data points. To understand the problem in more detail, let us have two classes of data: rectangle and circle. The rectangle class has three data points: $p$, $q$ and $r$ while the circle class has four data points: $a$, $b$, $c$ and $d$. We need to find the label of an unlabelled data points $x$ which actually belongs to the circle class. After constructing the k-NN graph by using the normalized Euclidean distance function, we can determine that unlabelled data point $x$ has 3-nearest neighbors, as shown in Fig. 1(A), $r$, $d$ and $q$, with weights $0.4$, $0.4$ and $0.2$, respectively. The total weight of the nearest neighbors belonging to the rectangle class is $0.6$ $(0.4 + 0.2)$ which is higher compared to the weight of the nearest neighbor belonging to the circle class, i.e., $0.4$. Thus, a label propagation algorithm such as SODA incorrectly assigns rectangle class to the unlabelled data point as shown in Fig. 1(B). Unfortunately, this data point does not belong to the rectangle class in reality and so the prediction of this algorithm is wrong. Thus, the performance of existing algorithms is affected by the use of Euclidean distance.

For solving the above-stated problem, we use pairwise relative constraints which could be equality or inequality constraints. Let we have three points, $i$-$p$, $k$-$q$ and $d$-$r$. Here $i$-$p$ denotes that point $i$ is denoted by '$p$'. Similarly, point $k$ is denoted by '$q$' and point $d$ is denoted by '$r$'. Here '$p$' and '$q$' belong to the same class but point '$r$' belongs to a different class. Thus corresponding inequality constraints are : $d(p, q) \leq d(p, r)$ and $d(q, p) \leq d(q, r)$. Another

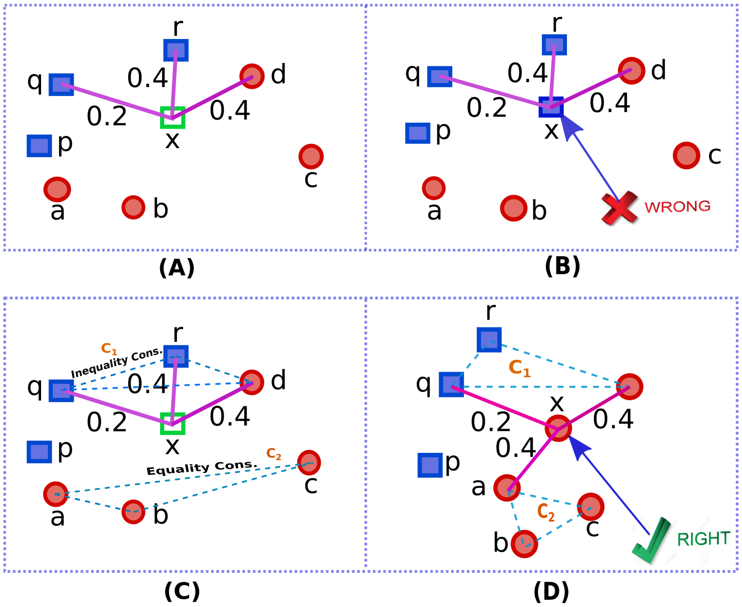

Fig. 1. (A) Data representation after application of k-NN with Euclidean distance function. (B) Wrong prediction of unlabelled data point x after application of existing methods like SODA. (C) Constraint representations of Euclidean data. (D) Right prediction of the unlabelled data point x after application of equality and inequality constraints using Bregman projection

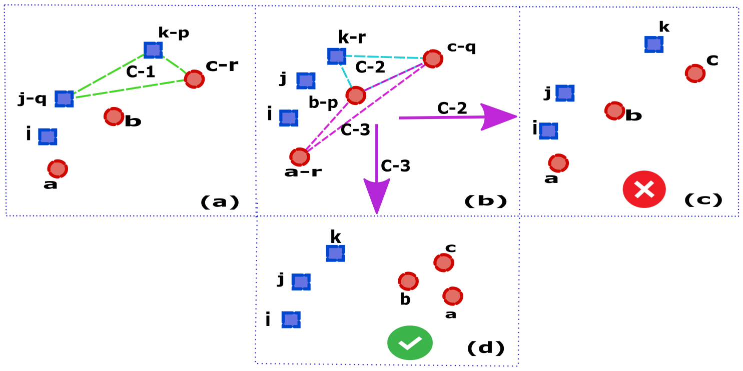

Fig. 2. (a) Data representation with inequality constraint C-1. (b) Data representation with inequality constraint C-2 and equality constraint C-3 after application of constraint C-1. (c) No change in data representation after application of C-1 and C-2 constraints, respectively (d) Desirable data representation after application of constraints C-1 and C-3, respectively.

point set is $\{a-p, b-q, c-r\}$. Here all three points, ’$p$’, ’$q$’ and ’$r$’ belong to the same class. Thus some equality constraints can be generated from these three points which are: $d(p,q) = d(p,r) = d(q,r)$. These constraints are also shown in Fig. 1(C).

In case of inequality constraints, the distance between points belonging to the same class must be less than the distance between points which belong to different classes. For example points: $i$-$p$ and $k$-$q$, must have smaller distances compared to the pair of points: $k$-$q$ and $d$-$r$ which belong to different classes. Similarly, in case of equality constraints, the distances between all the point pairs, $a$-$p$, $b$-$q$ and $c$-$r$ which belong to the same class must be almost equal and as close as possible. For satisfying both types of constraints onto the initial distance matrix, we use the Bregman projection technique. After satisfying two kinds of constraints, the resultant representation of the matrix or data is shown in Fig 1(D). Now if this learned distance matrix is given to our proposed SSDARD algorithm, then it will predict circle class for unlabelled data point $x$ which is the correct class because now the distance between $C_2$ and $x$ is less than that of $C_1$ and $x$.

Since the projection of constraints is not orthogonal, while satisfying the current set of constraints, previously satisfied constraints may get unsatisfied. Fig. 2(a) shows 6 data points and $S_1$, $S_2$ are the two subsets of constraints. C-1 ($k$-$p$, $j$-$q$ and $c$-$r$), C-2 ($b$-$p$, $c$-$q$ and $k$-$r$) are the constraints of $S_1$ and C-1 ($k$-$p$, $j$-$q$ and $c$-$r$), C-3 ($a$-$r$, $b$-$p$ and $c$-$q$) are the constraints of $S_2$. Fig. 2(a) shows the resultant data points after the constraints of $S_1$ are satisfied. The data points positions are depicted in fig. 2 (b) and 2 (c), respectively. The positions

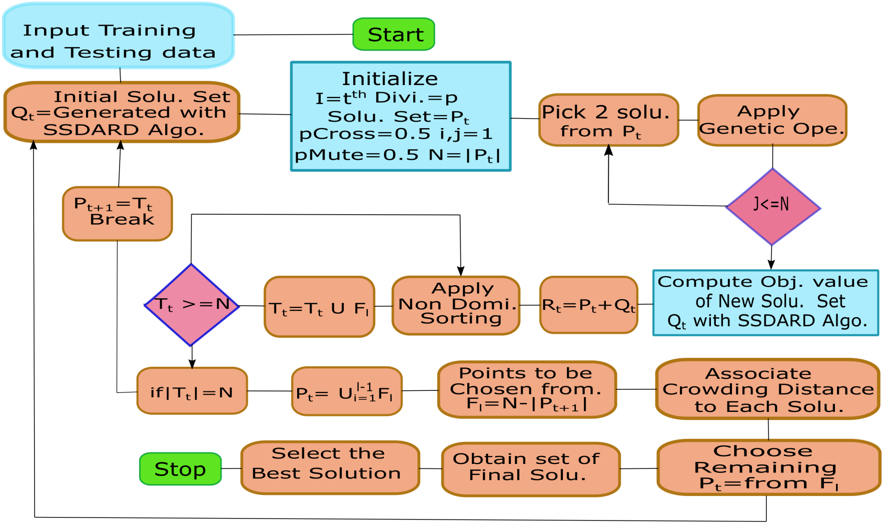
Input Training and Testing data
Start
Initial Solu. Set $Q_t$=Generated with SSDARD Algo.
Initialize $I=t^{th}$ Divi.=p Solu. Set=$P_t$ pCross=0.5 i,j=1 pMute=0.5 N=$|P_t|$
Pick 2 solu. from $P_t$
Apply Genetic Ope.
J<=N
$P_{t+1}=T_t$ Break
$T_t >= N$
$T_t=T_t \cup F_l$
Apply Non Domi. Sorting
$R_t=P_t+Q_t$
Compute Obj. value of New Solu. Set $Q_t$ with SSDARD Algo.
if$|T_t|$=N
$P_t= \cup_{i=1}^{l-1} F_i$
Points to be Chosen from. $F_l=N-|P_{t+1}|$
Associate Crowding Distance to Each Solu.
Stop
Select the Best Solution
Obtain set of Final Solu.
Choose Remaining $P_t$=from $F_l$

Fig. 3. Flow chart of proposed MO-SSDARD Algorithm

of data points do not change even after both types of S1 constraints are satisfied. Fig 2 (d) shows the position of data points if the constraints pair, C-1 and C-2 of $S_2$ are chosen. Now, the classes are well separated. So, our task is to select an appropriate subset of constraints for the projection of data. Moreover, for better labeling of the data, simultaneous optimization of various cluster validity indices, such as Accuracy and F-measure as external validity indices, is required. In order to satisfy both the objectives, 1) selection of an appropriate subset of constraints and 2) optimization of several quality measures, many objective optimization framework, namely NSGA-III is utilized in the current study.

We need to select only that subset of constraints which helps in identifying the optimal partitioning of the data. So the generated constraints that are redundant as well as conflicting to each other are ignored. The optimality of the partitioning can be checked with respect to different validity indices, namely, Accuracy (15) and F-measure (16). The selected constraints are utilized to transform the initial kernel distance matrix into an optimal kernel distance matrix.

Thus the problem statement of the current work is as follows:

$$\begin{aligned} \underset{\mathcal{K} \preceq 0}{\text{minimize}} \quad & D_{ld}(\mathcal{K}, \mathcal{K}_0) \\ \text{subject to} \quad & \bar{c} \subseteq C \\ & \forall_j^m obj_j \in O. \end{aligned} \tag{1}$$

where $\mathcal{K}_0$ is initial distance matrix , $\bar{c}$ is optimal subset of constraints and O=$\{obj_1, obj_2, obj_3, ..., obj_m\}$ is a set of some internal and external cluster validity indices. These objective functions are simultaneously optimized using the search capability of MOO-based technique. Thus, we have identified four problems to be solved; firstly, the above stated Euclidean distance problem; secondly, better utilization of unlabelled data to improve the performance of LDA, thirdly, the problem of nonlinear data and fourth, to select an appropriate subset of constraints.

## III. Semi-Supervised Discriminant Analysis with Relative Distance via Multi-objective Optimization (MO-SSDARD)

In order to solve the above-stated problems as discussed in Section II, we propose a method namely, Semi-Supervised Discriminant Analysis with Relative Distance in multi-objective optimization framework called MO-SSDARD. To extract the information from unlabelled data for dimensionality reduction application, first, we propose a method SSDAR as discussed in Subsection III-D, which uses k-NN graph. The k-NN graph uses the normalized Euclidean distance function for finding the appropriate neighbors of each data point. However, the Euclidean distance function is not capable of representing the appropriate relationships between instances and of capturing the geometry of the manifold. In our proposed SSDARD, we use relative distance based function for measuring the same. These constraints are satisfied by using the Bregman projection. However, the Bregman Projection is not orthogonal. Therefore, we need to identify an appropriate subset of constraints to find out a distance function. This distance function will generate an appropriate k-NN graph. Later, this graph is given to the SSDARD algorithm to classify the objects in lower dimensional space accurately. In order to select a proper subset of constraints to optimize initial distance matrix as discussed in Subsection III-A, a popular multi-objective optimization technique called NSGA-III is used.

### A. Determining Initial Distance Matrix $K_0$ (k-NN Graph)

Let us consider a training dataset $X = \{x_i \in R^d | i = 1, 2, 3, ..., n\}$ in some multidimensional Euclidean space. We construct an un-directed graph $G = (V, E)$ where the edge set $E$, a subset of $V \times V$, shows the distance relationship between each pair of points in the vertex set $V$ and all the data points of $X$ are contained in $V = \{1, 2, 3, ..., n\}$ . The pairwise similarity between any two data points $x_i$ and $x_j$ is determined as:

$$v_{ij} = \begin{cases} \text{exp} - \frac{\|x_i - x_j\|^2}{2\sigma} & \text{for} \quad i \text{ not equal to } j \\ 0 & \text{for} \quad i \text{ equal to } j \end{cases} \tag{2}$$

where $\sigma$ denotes the distance between point $x_i$ and its $k^{th}$ nearest neighbor that controls the degree of decrease of $v_{ij}$. We call this resultant $n \times n$ graph matrix as the initial graph matrix, $k_0$, which represents the similarity between all the points in the dataset, $X$.

### *B. Generate Relative Distance Constraints $C_{neq}$ and $C_{eq}$*

The pairwise similarity or dis-similarity, $v_{ij}$, available in the above initial distance matrix, $K_0$, is calculated by the Euclidean distance between any pair of points, $(x_i, x_j)$. However, this Euclidean distance function does not quantify whether these two data points come from the same class or not.

For this, we consider relative distance constraints, $C$, that express information about distances between points in $X$ in the form of distance function, $\gamma$. Our relative distance function, $\gamma$, considers relative positions from the labelled data that are available in addition to the Euclidean distances. We use the few available labelled data to generate the pairwise constraint set, $C$ and each element in this set $C$ represents the relative positions of three points in $X$. This constraint set $C$ is partitioned into two subsets: the inequality set, $(C_{neq})$ and the equality set, $(C_{eq})$.

The inequality constraint set, $(C_{neq})$, contains those constraints where any two points come from the same class and the other point belongs to a different class. Now, suppose we have three points $p$, $q$ and $r$ out of which $p$, $q$ belong to the same class whereas $r$ comes from a different class. So, ideally the distance between points $p$ and $q$ should be less than or equal to the distance between $p$ and $r$ and vice versa. So, each constraint $(p, q|r)$ in $(C_{neq})$ implies:

$$pq|r : \delta\gamma(p, q) \leq \gamma(p, r) \text{ and} \tag{3}$$

$$qp|k : \delta\gamma(q, p) \leq \gamma(q, r) \tag{4}$$

where $\delta$ is some constant value.

Similarly, for the constraints belonging to the equality set, $(C_{eq})$, the pairwise distance for given points, $p$, $q$ and $r$, must be equal. So, each constraint $(p, q|r)$ in $(C_{eq})$ satisfies the following relation:

$$\gamma(p, q) = \gamma(q, r) = \gamma(p, r) \tag{5}$$

### *C. Determination of Learned Distance Matrix $K$*

The Bregman divergence (17) between any two positive semidefinite matrices, $K$ and $K_0$, is defined as follows:

$$D_\Delta(K, K_0) = \Delta(K) - \Delta(K_0) - \text{tr}(\bigtriangledown\Delta(K_0)^T(K - K_0)) \tag{6}$$

where $\bigtriangledown$ denotes the gradient operator and $\Delta$ is strict convex function. Similar to (18) and (19), we can keep $\Delta(K) = -\text{lot det}(K)$, the resulting log determinant divergence is as follows:

$$D_d(K, K_0) = \text{tr}(KK_0^{-1}) - \text{log det}(KK_0^{-1}) - \text{Constant} \tag{7}$$

In order to find out learned matrix, $K$, from initial distance matrix, $K_0$, while satisfying all constraints, $C$, the optimization problem can be formulated as follows:

$$\begin{aligned} \min_K \quad & D_d(K, K_0) \\ \text{s.t.} \quad & \text{tr}(KC) \leq 0 \\ & K \preceq 0 \end{aligned} \tag{8}$$

With a Lagrange multiplier $\beta \geq 0$, this problem can be written like:

$$min_K = \min_K D_d(K, K_0) + \beta\text{tr}(KC) \tag{9}$$

Now substituting the value of $D_d(K, K_0)$ from the Equation 7, the above equation becomes:

$$min_K = \text{tr}(KK_0^{-1}) - \text{log det}(KK_0^{-1}) - \text{Constant} \tag{10}$$

$$+\beta\text{tr}(KC) \tag{11}$$

To find out the value of $K$, we calculate the partial derivative of Eq.10 with respect to $K$ and equate it to zero.

$$\begin{aligned} & \frac{\partial\text{tr}(KK_0^{-1})}{\partial K} \\ -\frac{\partial\text{log det}(KK_0^{-1})}{\partial K} - \frac{\partial\text{Constant}}{\partial K} + \beta\frac{\partial\text{tr}(KC)}{\partial K} &= 0 \\ K_0^{-1} - \frac{1}{det(KK_0^{-1})}K_0^{-1} - 0 + \beta C &= 0 \\ K_0^{-1} + \beta C &= \frac{1}{K} \end{aligned}$$

then finally the value of learned $K$ is as follows:

$$K = \frac{1}{K_0^{-1} + \beta C} \tag{12}$$

### *D. SSDA with Relative Distance Matrix, K*

Suppose we have dataset, $X = \{x_1, x_2, ..., x_m, x_{m+1}, ..., x_n\} \in R^{d*n}$, where $x_i|_{i=1}^{m}$ and $x_i|_{i=m+1}^{n}$ represent labelled data points and unlabelled data points, respectively. For labelled data $x_i = \{x_1, x_2, ..., x_m\}$, the corresponding labels are given as $y_i \in \{1, 2, 3, ..., c\}$, where $c$ is the number of classes. To avoid the above discussed labelled data deficiency problem, we make use of unlabelled data for improving the performance of LDA. Similar to (8), we can derive new discriminant function as follows:

$$Q = \max_{Q} \frac{Q^T M_b Q}{Q^T M_w Q + I(Q)} \tag{13}$$

Where, $I(Q)$ is a regularization term that needs to be learned from the generated distance matrix, $K$.

$$I(Q) = J_{ll} + J_{lu} + J_{uu} \tag{14}$$

Where $J_{ll}$, $J_{lu}$ and $J_{uu}$ represent the similarity between labelled data points to labelled data points, labelled data points to unlabelled data points and unlabelled data points to unlabelled data points, respectively.

We can eliminate $J_{ll}$ from Eq. 14 because ground truth information about labelled data is already available.

For determining $J_{lu}$, if an unlabelled data point is similar to the mean of the labelled data points of $i^{th}$ class, then it is expected that the unlabelled data point will belong to $i^{th}$ class. Let $S$ be a similarity matrix between class mean and unlabelled data points, then $J_{lu}$ is calculated as follows:

$$\begin{aligned} J_{lu} &= \sum_{i=m+1}^{n} \sum_{j=1}^{c} (Q^T x_i - Q^T \mu_j)^2 S_{ij} \\ J_{lu} &= Q^T \left[ \sum_{i=m+1}^{n} (\sum_{j=1}^{c} S_{ij}) x_i x_i^T + \sum_{j=1}^{c} (\sum_{i=m+1}^{n} S_{ij}) \mu_j \mu_j^T \right. \\ &\quad -2 \sum_{i=m+1}^{n} \sum_{j=1}^{c} S_{ij} x_i \mu_j^T Q \end{aligned} \tag{15}$$

$$J_{lu} = Q^T (X_u V_1 X_u^T + U V_2 U^T - 2 X_u S U^T) Q \tag{16}$$

where $X_u$ is unlabelled data points, $\mu_j$ is the mean value of $j^{th}$ class, $U = [\mu_1, \mu_2, ..., \mu_c]$ and $V_1$ is a diagonal matrix where each element is row-wise sum of matrix $S$ and $V_2$ is also a diagonal matrix whose each elements is column-wise sum of matrix $S$.

Similarly, for $J_{uu}$, if two unlabelled data points are more similar, then it is assumed that both data points are close to each other. Thus, $J_{uu}$ can be determined as follows:

$$\begin{aligned} J_{uu} &= \sum_{i,j=m+1}^{n} (Q^T x_i - Q^T x_j)^2 S_{ij} \\ J_{uu} &= Q^T \sum_{i,j=m+1}^{n} (x_i x_i^T + x_j x_j^T - 2 x_i x_j^T)^2 S_{ij} Q \\ J_{uu} &= 2Q^T \left[ \sum_{i=m+1}^{n} ( \sum_{j=m+1}^{n} S_{ij}) x_i x_i^T - \sum_{i,j=m+1}^{n} S_{ij} x_i x_j^T Q \right] \end{aligned}$$

$$J_{uu} = 2Q^T X_u (L^{uu} - K^{uu}) X_u^T Q \tag{17}$$

where $S_{ij}$ is $(i, j)$th element of $K_{ij}, L^{uu}$ is diagonal matrix whose diagonal elements are the column sums of $K^{uu}$.

### *E. Optimize K with NSGA-III Framework*

As the proposed approach aims to optimize the initial distance matrix, K, with a set of external validity measures simultaneously while selecting the appropriate subset of constraints, a popular genetic algorithm based MOO technique, NSGA-III (Non-dominated sorting genetic algorithm-III) (20), is utilized as the underlying optimization strategy. The flow chart of the proposed approach is shown in Fig.3. To understand the approach in more detail, let us consider $i^{th}$ generation in which population set P is having N solutions. For each solution, we first generate the initial distance matrix $K_0$ (as discussed in Section III-A) and a set of constraints represented in the form of a chromosome (as discussed in Section III-F). After this, the initial distance matrix is updated by using Bregman projection in such a way that all the constraints in the constraint set are satisfied. Later we evaluate all objective functions as discussed in Section III-G corresponding to a particular solution. Thus each solution is composed of a subset of constraints and the corresponding objective functions. Next, two solutions are randomly selected from the set P and crossover operator is applied on them as discussed in Section III-H to generate two new solutions or chromosomes. After that, each chromosome undergoes through the mutation operation as discussed in Section III-H2. Application of crossover and mutation operators helps in generating a new population Q of size $N$. Next the old population $P$ and the new population, $Q$ are combined to generate a merged population (R) of size $2 \times N$. Non-Dominated sorting algorithm is applied to sort $R$ in different non-domination levels $(F_1, F_2, \ldots, F_m)$. Then, one by one each non-dominated level is selected to construct the new population $T$, starting from $F_1$, until the size of $|T| \geq N$. If the last included level is L, then all the solutions from level (L+1) onward are rejected from R. If $|T| = N$ then no further operation is needed and set $T$ is considered for next generation. If $|T| \leq N$, then members from fronts 1 to $L-1$ are selected and the remaining solutions are chosen from the last front, $L$, by using the crowding distance concept (20). In the end, the proposed approach finally provides a set of non-dominated solutions on the final Pareto front. All the solutions available on the final Pareto front are non-dominated to each other. Thus selecting one of such solutions is very difficult for the user. The best solution among all solutions in the Pareto front is selected based on an external cluster validity index.

### *F. Chromosome Representation*

A constraint matrix that consists of all the constraints satisfying Equations 3,4, and 5 is encoded in the form of a chromosome. There are two types of constraints: equality constraint ($C_{eq}$) and inequality constraint ($C_{neq}$) as described in the above section. Let $p$, $q$ and $r$ be the data points belonging to any dataset then

- if $p$ and $q$ belong to the same class and $r$ belongs to the other class then it is called inequality constraint.
- if $p$, $q$ and $r$ all belong to the same class then it is called equality constraint.

Let we have three classes $a$, $b$ and $c$, then total number of permutations possible while satisfying these two conditions is shown in Fig. 5, the 6 triples [($a$, $a$, and $b$), ($a$, $a$, and $c$), ($b$, $b$, and $a$), ($b$, $b$, and $c$), ($c$, $c$, and $a$) and ($c$, $c$, and $b$)] should belong to inequality class ($C_{neq}$) and 3 triples [($a$, $a$, and $a$), ($b$, $b$, and $b$) and ($c$, $c$, and $c$)] should belong to equality class ($C_{eq}$).

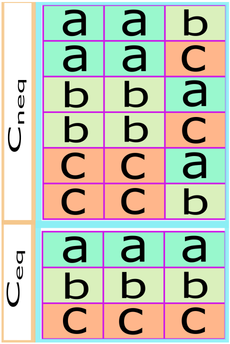

Fig. 4. Chromosome representation with different permutations of constraints

### *G. Computing Objective Functions*

As in Equation 1, the initial distance metric $K$ is optimized while choosing the appropriate subset of constraints and the set of objective functions. Here the objective functions ensure the better labeling of the testing data. Therefore, we have considered accuracy and F-measure as external validation indices.

*1) Accuracy:* Accuracy shows how close the predicted class label is to the actual (true) class label. The 4 parameters used for evaluating accuracy include: $T_p$(True positive), $T_n$(True negative) are observations which are correctly predicted and $F_p$(False positive), $F_n$(False negative) are false observations where the actual events were negative and positive, respectively.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{18}$$

*2) F-measure:* Another external validation index, f-measure is the harmonic mean of precision and recall that is evaluated as:

$$F - measure = \frac{2 * P * R}{P + R} \tag{19}$$

where P is the precision value and R is the recall value.

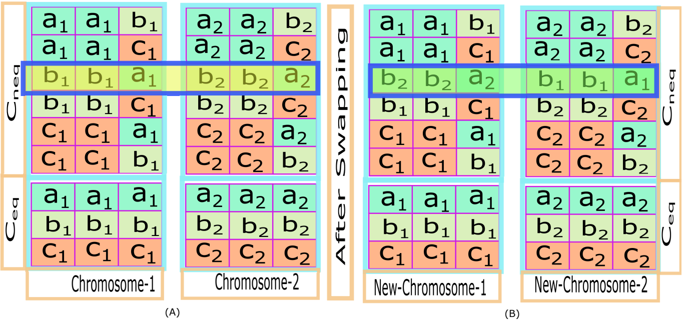

Fig. 5. Chromosome representation with different permutations of constraints

### *H. Genetic Operators*

In this section we have discussed about different genetic operators used in the proposed evolutionary framework.

*1) Crossover:* Crossover is a genetic operator widely used in the genetic algorithm. It is analogous to reproduction and biological crossover. In this, two random solutions from the population are selected and two off-springs are produced using the genetic material of these parents. Let us assume that we have two chromosomes (chromosome-1 and chromosome-2) as shown in Fig.4 and each chromosome consists of $p$ permutations of 3 classes while satisfying the conditions given in Section III-B. While performing the crossover operation, a random number between 1 to $p$ is generated. If the random number value is three then 3 permutations of both chromosome-1 and chromosome-2 are swapped with each other. The generated new chromosomes, chromosome-1 and chromosome-2 are shown in Fig.4.

*2) Mutation:* Mutation is a genetic operator and is used to maintain genetic diversity from one generation of a population to the next, of a particular genetic algorithm. It is analogous to biological mutation. Using this operator, we generate a random number between 1 to $p$ but for swapping we have chosen one solution from current population and another newly generated chromosome.

## IV. Experimental Results

In this section, we have compared our proposed method, MO-SSDARD, with several other dimensionality reduction methods, including, LDA (4), LLP (5), SODA (9), SSDAR (8), GLPP (10), and RCDA (1) and Metric learning methods like DML-DC (11), U-LMNN (13) and RDML (14).The source codes and respective parameter settings of all other algorithms are directly taken from their respective papers. After projecting the data onto a new embedding space that consists of $c - 1$ vectors where $c$ is the number of classes in each dataset, we apply the k-NN classifier to perform classification.

TABLE I
SUMMARY OF THE DATASETS

| Datasets | #Samples | #Dimensions | #Classes | #labelled | #Unlabelled |
|---|---|---|---|---|---|
| Iris | 150 | 4 | 3 | 3 | 20 |
| Hayes_roth | 160 | 4 | 3 | 3 | 22 |
| Heart_satlog | 270 | 13 | 2 | 5 | 100 |
| Vehicle | 845 | 18 | 4 | 5 | 100 |
| Diabetes | 768 | 8 | 2 | 5 | 100 |
| Liver | 345 | 6 | 2 | 3 | 50 |
| COIL | 1500 | 241 | 6 | 5 | 100 |
| Thyroid | 215 | 5 | 3 | 3 | 20 |
| Pendigit | 3498 | 16 | 10 | 5 | 50 |
| BCI | 400 | 117 | 2 | 5 | 50 |
| Moon | 1000 | 2 | 2 | 3 | 50 |
| Ionosphere | 350 | 34 | 2 | 5 | 50 |

TABLE II
COMPARISON OF ACCURACY AND F-MEASURE VALUES ACHIEVED BY OUR PROPOSED APPROACH COMPARED TO VARIOUS STATE OF ART ALGORITHMS OVER DIFFERENT DATASETS

| Dataset | Validation Indices | MO-SSDARD | DML-DC | U-LMNN | RDML | RCDA | GLPP | SSDAR | SODA | LDA | LLP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Iris | Accuracy | **94.13** | 38.33 | 41.67 | 40.00 | 88.67 | 83.33 | 36.67 | **94.30** | 40.67 | **94.67** |
| | F-Measure | **92.04** | 35.09 | 33.96 | 30.77 | 85.47 | 79.67 | 29.63 | **92.74** | 28.80 | **92.24** |
| Hayes roth | Accuracy | **43.75** | 42.35 | 38.82 | 36.47 | 35.63 | 35.63 | 39.37 | 42.35 | 35.63 | 40.63 |
| | F-Measure | **45.78** | 32.88 | 36.59 | 27.03 | 29.93 | 33.55 | 34.90 | 43.68 | 27.97 | 33.57 |
| Heart satlog | Accuracy | **70.89** | 35.00 | 51.67 | 58.33 | 55.19 | 53.70 | 65.19 | 73.33 | 58.15 | 65.19 |
| | F-Measure | **72.37** | 40.00 | 63.29 | 65.75 | 53.99 | 55.52 | 70.44 | 82.61 | 72.37 | 70.62 |
| Vehicle | Accuracy | **60.66** | 23.29 | 24.47 | 25.18 | 46.39 | 34.79 | 51.83 | 44.24 | 31.95 | 41.30 |
| | F-Measure | **31.12** | 12.37 | 13.01 | 14.05 | 25.12 | 31.89 | 26.13 | 20.20 | 04.33 | 30.14 |
| Ionoshepre | Accuracy | **75.33** | 51.25 | 42.92 | 58.33 | 68.00 | 68.57 | 74.29 | 69.58 | 48.00 | 64.29 |
| | F-Measure | **80.26** | 61.64 | 46.69 | 71.26 | 75.55 | 79.17 | 79.64 | 77.68 | 33.58 | 70.31 |
| Liver | Accuracy | **60.59** | 45.19 | 41.84 | 50.21 | 50.43 | 58.84 | 51.01 | 45.19 | 51.01 | 57.39 |
| | F-Measure | **55.88** | 49.42 | 38.22 | 49.36 | 52.10 | 55.35 | 44.95 | 48.22 | 52.12 | 59.95 |
| Moon | Accuracy | **91.60** | 50.11 | 50.78 | 50.67 | 83.30 | 88.30 | 89.50 | 85.12 | 64.60 | 86.80 |
| | F-Measure | **88.27** | 44.53 | 45.81 | 43.39 | 82.80 | 87.83 | 89.21 | 84.15 | 71.27 | 86.05 |
| COIL | Accuracy | 52.93 | 15.98 | 17.36 | 18.62 | 51.20 | 18.00 | 49.53 | **55.29** | 25.73 | 51.00 |
| | F-Measure | 28.54 | 06.88 | 06.99 | 04.58 | 27.95 | 09.02 | 27.42 | **32.58** | 13.10 | 29.80 |
| Diabetes | ACC | **74.96** | 48.39 | 46.42 | 45.34 | 38.28 | 69.14 | 74.09 | 58.60 | 47.53 | 69.14 |
| | F-Measure | **64.77** | 36.84 | 37.32 | 38.13 | 49.68 | 57.90 | 59.80 | 41.52 | 32.27 | 57.90 |
| Thyroid | Accuracy | **93.26** | 60.27 | 52.05 | 60.96 | 93.95 | 89.30 | 62.79 | 79.45 | 64.65 | 93.95 |
| | F-Measure | 90.47 | 74.78 | 67.89 | 75.32 | **95.59** | 92.20 | 71.22 | 86.84 | 72.86 | 95.65 |
| BCI | Accuracy | **54.88** | 50.00 | 52.76 | 50.69 | 49.25 | 53.50 | 51.23 | 51.38 | 50.50 | 53.00 |
| | F-Measure | 50.43 | 58.45 | 53.87 | 53.42 | 44.69 | **59.91** | 47.23 | 37.89 | 45.60 | 56.48 |
| Pen digit | Accuracy | **88.68** | 09.84 | 10.45 | 10.21 | 87.34 | 68.50 | 7627 | 81.92 | 69.61 | 65.23 |
| | F-Measure | **53.70** | 01.34 | 03.01 | 01.56 | 52.62 | 26.14 | 3870 | 47.38 | 22.80 | 20.12 |

### *A. Experimental Setup*

We have evaluated the performance of these 9 state-of-the-art algorithms on 12 benchmark datasets including one image data set: COIL (21), a Brain-computer dataset, BCI (8) and 10 UCI datasets: Iris, Hayes-roth, Heart-statlog, Liver, Moon, Vehicle, Diabetes, Pen digit, Ionosphere and Thyroid, to compare their performance to that of our proposed dataset. These datasets have been widely used by researchers to evaluate the performance of their algorithms, as seen in the papers (8), (22) and (9). $x$ labelled data points and $y$ unlabelled data points are taken from each class of the dataset, to make up the training dataset while the rest of the data points are used for testing, in each dataset. Table I shows the partitioning of each dataset as well as its summary. With each dataset, we have randomly performed 20 partitions and have shown the average results over 20 trials. In our MO-SSDARD algorithm, we keep the parameter values as follows: $I$ (Number of iterations) = 10, $p$ (number of divisions) = 4, $P_t$ (Initial number of solutions) = 20, $pCross$ (Crossover probability) = 0.5 and $pMute$ (Mutation probability) = 0.5.

### *B. Results and Discussions*

From Table II, it can be concluded that our algorithm outperforms the other considered algorithms for the objectives, accuracy and F-measure, in case of Hayes Roth, Ionosphere, Diabetes and Pen Digit. For Vehicle, Liver, Moon and Pen digit datasets, accuracy improvement is very significant. But SODA performs better than our proposed algorithm for Iris, Heart Satlog, COIL, Pen digit datasets. LPP also outperforms our algorithm for Iris dataset. In case of Vehicle, Liver, Moon, Thyroid and BCI datasets, our algorithm falls behind other algorithms with respect to the F-measure value.

Bregman projection is not orthogonal which means that while satisfying the current constraint, the previously satisfied constraints may get unsatisfied. Hence, we require an appropriate subset of constraints for accurate prediction. Thus we can also improve the performance of our method for the datasets, COIL, liver and vehicle by choosing a proper subset of constraints.

### C. Statistical Significance Tests

We have also conducted a t-test (23) at 0.05 significance level to check whether the improvements attained by our proposed approaches are statistically significant or happened by chance. Results obtained using the t-test clearly show that the obtained performance improvements are statistically significant.

## V. Conclusion

In this paper, we have presented a new method for semi-supervised discriminant analysis with relative distance constraints by using a multi-objective optimization framework. The main contributions of this paper lie in the following two aspects: firstly, the proposed algorithm uses only a few labelled information in the form of equality and inequality constraints and a large quantity of the unlabelled data for training; secondly, in order to select the best subset of constraints, we use a MOO framework and these constraints will help in finding an appropriate distance matrix to generate the k-NN graph. Later, the generated graph is given to our SSDARD algorithm for better partitioning of the data. Comparative results presented in the Table show that the performance of our proposed approach is much better than the other one.

As we know, manually generating some labelled data is often time-consuming and expensive. Hence, there is a scarcity of labelled data in one domain often called the target task. However, often there exists some related existing domain often called source domain with sufficient amount of labelled data. Thus, in the future, we plan to extend our model by utilizing the existing well-established source domain knowledge to learn a well-performing model for the target domain.

## References

[1] K.-K. Huang, D.-Q. Dai, and C.-X. Ren, “Regularized coplanar discriminant analysis for dimensionality reduction,” *Pattern Recognition*, vol. 62, pp. 87–98, 2017.

[2] C. Yu, R. Nie, and D. Zhou, “A regularized locality projection-based sparsity discriminant analysis for face recognition,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 05, p. 1856006, 2018.

[3] Y. Sun, Q. Ye, R. Zhu, and G. Wen, “Cognitive gravity model based semi-supervised dimension reduction,” *Neural Processing Letters*, vol. 47, no. 1, pp. 253–276, 2018.

[4] P. Xanthopoulos, P. M. Pardalos, and T. B. Trafalis, “Linear discriminant analysis,” in *Robust data mining*. Springer, 2013, pp. 27–33.

[5] X. He and P. Niyogi, “Locality preserving projections,” in *Advances in neural information processing systems*, 2004, pp. 153–160.

[6] M. Agarwal, H. Agrawal, N. Jain, and M. Kumar, “Face recognition using principle component analysis, eigenface and neural network,” in *Signal Acquisition and Processing, 2010. ICSAP’10. International Conference on*. IEEE, 2010, pp. 310–314.

[7] D. Cai, X. He, and J. Han, “Semi-supervised discriminant analysis,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–7.

[8] Y. Zhang and D.-Y. Yeung, “Semi-supervised discriminant analysis using robust path-based similarity,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[9] F. Nie, S. Xiang, Y. Jia, and C. Zhang, “Semi-supervised orthogonal discriminant analysis via label propagation,” *Pattern Recognition*, vol. 42, no. 11, pp. 2615–2627, 2009.

[10] S. Huang, A. Elgammal, L. Huangfu, D. Yang, and X. Zhang, “Globality-locality preserving projections for biometric data dimensionality reduction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 15–20.

[11] B. Nguyen and B. De Baets, “An approach to supervised distance metric learning based on difference of convex functions programming,” *Pattern Recognition*, vol. 81, pp. 562–574, 2018.

[12] B. Nguyen, C. Morell, and B. De Baets, “Supervised distance metric learning through maximization of the jeffrey divergence,” *Pattern Recognition*, vol. 64, pp. 215–225, 2017.

[13] ——, “Distance metric learning with the universum,” *Pattern Recognition Letters*, vol. 100, pp. 37–43, 2017.

[14] R. Jin, S. Wang, and Y. Zhou, “Regularized distance metric learning: Theory and algorithm,” in *Advances in neural information processing systems*, 2009, pp. 862–870.

[15] J. S. Bernardes, F. R. J. Vieira, G. Zaverucha, and A. Carbone, “A multi-objective optimization approach accurately resolves protein domain architectures,” *Bioinformatics*, vol. 32, no. 3, pp. 345–353, 2015.

[16] R. O. Duda and P. E. Hart, “Pattern classification and scene analysis,” *A Wiley-Interscience Publication, New York: Wiley, 1973*, 1973.

[17] L. M. Bregman, “The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming,” *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967.

[18] S. Anand, S. Mittal, O. Tuzel, and P. Meer, “Semi-supervised kernel mean shift clustering,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1201–1215, 2014.

[19] E. Amid, A. Gionis, and A. Ukkonen, “Semi-supervised kernel metric learning using relative comparisons,” *arXiv preprint arXiv:1612.00086*, 2016.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[21] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in neural information processing systems*, 2001, pp. 556–562.

[22] S. Wang, J. Lu, X. Gu, H. Du, and J. Yang, “Semi-supervised linear discriminant analysis for dimension reduction and classification,” *Pattern Recognition*, vol. 57, pp. 179–189, 2016.

[23] P. Baldi and A. D. Long, “A bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes,” *Bioinformatics*, vol. 17, no. 6, pp. 509–519, 2001.