1.Introduction:

1.1 MultiView Discriminant Analysis(MvDA):

In many machine learning applications, an object or class can be described from multiple views or styles, which leads to the need of multiView Discriminant Analysis.

There may be a large gap between views and the samples from different views and they might lie in different spaces.
Therefore, directly mapping the samples from different views is not acceptable.

Before MvDA, the available methods work well only in the scenario of two views. In case of multiple views, the pairwise (i.e., one-versus-one) strategy is generally exploited to convert one common space for v views problem to C(2,v) common spaces problem. However, such a pairwise manner is neither efficient nor optimal for classification across different views. What we need is a unified semantic common space, which should embody invariant features or attributes that can identify the underlying object, commonly shared by all the views rather than only two views.

MultiView Discriminant analysis seeks a nonlinear discriminant and view-invariant representation which is shared among multiple views.
MvDA employs a novel eigenvalue-based multi-view decomposition function to encapsulate as much discriminative variance as possible into all the available common feature dimensions.

MvDA produces representations possessing: a) low variance within the same class regardless of view discrepancy, b) high variance between different classes regardless of view discrepancy.

In the learned multi-view space, the obtained features are projected into a latent common space.

2. Mathematical Formulation:

2.1 Dataset representation

As shown in the figure below MvDA attempts to find v linear transforms $w_1,\ w_2,\ w_3, . . . w_v$ that can respectively project the samples from v views to one discriminant common space, Where we define $X^{(j)} = \{x_{ijk} \mid i\ = 1,\ . . .\ ,\ c;\ k\ = 1\ , . . . , n_{ij}\}$ as samples from jth view $(j = 1, . . . , v)$, where $x_{ijk} \in \mathbb{R}^{d_j}$ is the kth sample from the jth view of the ith class of d_j dimension, c is the number of classes and n_{ij} is the number of samples from the jth view of ith class.

The samples from v views are then projected to the same common space by using v view-specific linear transforms. We denote the projection results as $Y = \{y_{ij}\ =\ w^T{}_j\ x_{ijk} | \ i\ =\ 1, . . . , c,\ j\ =\ 1, . . . ,\ v,\ k = 1, . . . ,\ n_{ij}\}$ **.** In the common space, according to our goal, the between-class variation $S^y{}_B$ from all views should be maximized while the within-class variation $S^y{}_W$ from all views should be minimized. We formulated this objective as a generalized Rayleigh quotient.

$$(w^*{}_1, w^*{}_2, . . . , w^*{}_v)\ \ = arg\ max_{w_{1,...,}w_v} \frac{Tr\left(W^TDW\right)}{Tr(W^TSW)} \quad \text{.......................} (1)$$

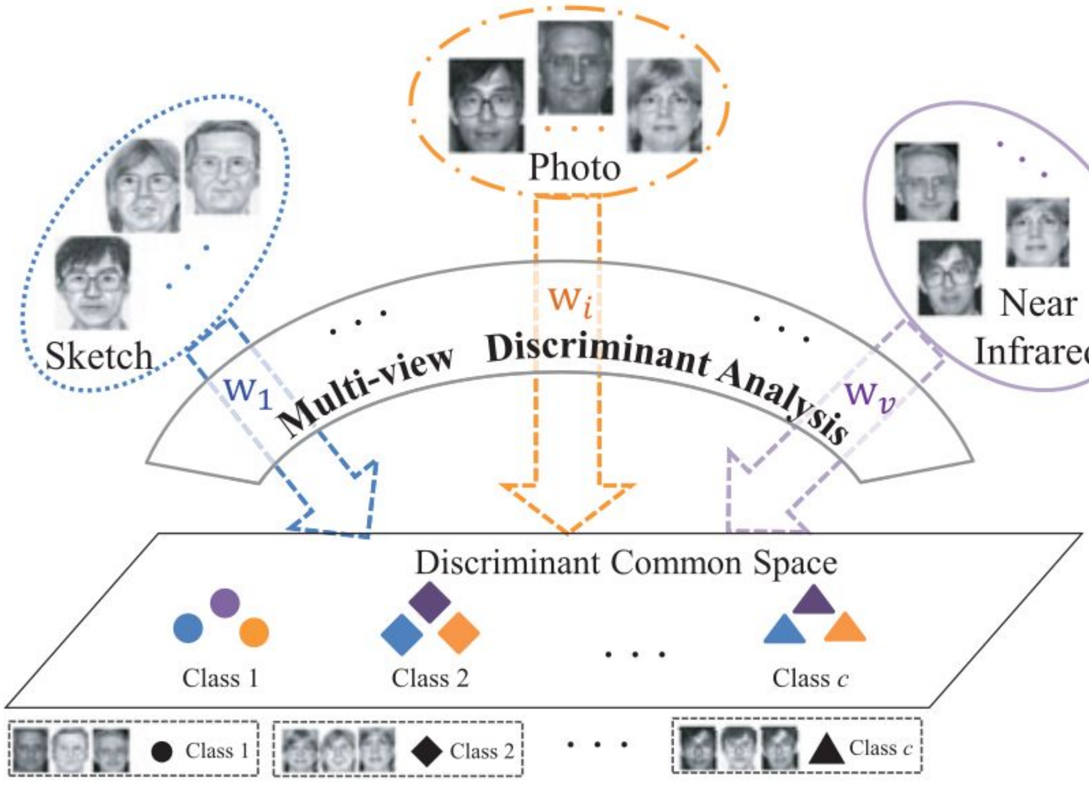
Here, the within-class scatter matrix $S^y{}_W$ and the between-class scatter matrix $S^y{}_B$ of the projected samples in the common space are computed as below:

$$S^y{}_W = \sum^c{}_{i=1}\sum^v{}_{j=1}\sum^{n_{ij}}{}_{k=1}(y_{ijk} - \mu_i)(y_{ijk} - \mu_i)^T \quad \text{.......................................} (2)$$

$$S^y{}_B = \ \sum^v{}_{i=1}n_i(\mu_i - \mu)(y_{ijk} - \mu_i)^T \quad \text{..} (3)$$

where $\mu_i\ = \frac{1}{n_i} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} y_{ijk}$ is the mean of all the samples of the ith class over all views in the common space, $\mu_i\ = \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{v} \sum_{k=1}^{n_{ij}} y_{ijk}$ is the mean of all samples over all views in the common space, $n_i\ = \sum_{j=1}^{v} n_{ij}$ the number of samples of the ith class in all views, and $n\ = \sum_{i=1}^{c} n_i$ is the number of samples from all classes and all views.

From Eq. (2) and Eq. (3), it is clear that the within-class and between-class variations are computed from the samples of all views, not only intra-view ones but also inter-view ones. In other words, not only the discriminant information from the intra-view but also that from the inter-view are considered. After obtaining $w_1, w_2, w_3, ... w_v$ from Eq. (1), the samples from v views can be compared after respectively projected to the discriminant common space.



2.2 Analytical Solution of MvDA:

Although Eq. (1) seems like LDA, it is much more complicated, as it needs to jointly optimize v distinct linear transforms. Fortunately, we work out an analytic solution by reformulating the trace ratio problem in Eq. (1) into a ratio trace problem.

Formally, the within-class scatter matrix in the common space in Eq. (2) can be reformulated as follows:

$$\mathbf{S}_W^y = \left[\mathbf{w}_1^T\ \mathbf{w}_2^T \cdots \mathbf{w}_v^T\right] \begin{pmatrix} \mathbf{S}_{11} & \cdots & \mathbf{S}_{1v} \\ \vdots & \vdots & \vdots \\ \mathbf{S}_{v1} & \cdots & \mathbf{S}_{vv} \end{pmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_v \end{bmatrix} = \mathbf{W}^T\mathbf{S}\mathbf{W},$$

.... (4)

with $W = [w_1{}^T, w_2{}^T, \dots, w_v{}^T]$ and S_{ij} is defined as below with $\mu_{ij}{}^{(x)} = \frac{1}{n_{ij}} \sum_{k=1}^{n_{ij}} x_{ijk}$:

$$
\begin{aligned}
&\mathit{if}\ (j == r):\\
&\qquad S_{jr} = \sum_{i=1}^{c} (\sum_{k=1}^{n_{ij}} x_{ij} x^{T}{}_{ijk} - \frac{n_{ij} n_{ij}}{n_i} \mu^{(x)}{}_{ij} \mu^{(x)T}{}_{ij}\\
&\mathit{else}:\\
&\qquad S_{jr} = \frac{n_{ij} n_{ir}}{n_i} \mu^{(x)}{}_{ij} \mu^{(x)T}{}_{ir}
\end{aligned}
\qquad \text{.. (5)}
$$

Similarly, the between-class scatter matrix in Eq. (3) can be further reformulated as follows:

$$
\mathbf{S}_B^y = \left[\mathbf{w}_1^T\ \mathbf{w}_2^T \cdots \mathbf{w}_v^T\right] \begin{pmatrix} \mathbf{D}_{11} & \cdots & \mathbf{D}_{1v} \\ \vdots & \vdots & \vdots \\ \mathbf{D}_{v1} & \cdots & \mathbf{D}_{vv} \end{pmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_v \end{bmatrix} = \mathbf{W}^T\mathbf{D}\mathbf{W} \qquad \text{........... (6)}
$$

where W is the same as above and D_{jr} is defined as:

$$
D_{jr} = \left(\sum_{i=1}^{c} \frac{n_{ij} n_{ir}}{n_i} \mu^{(x)}{}_{ij} \mu^{(x)T}{}_{ir} \right) - \frac{1}{n} \left(\sum_{i=1}^{c} n_{ij} \mu^{(x)}{}_{ij} \right) \left(\sum_{i=1}^{c} n_{ir} \mu^{(x)}{}_{ir} \right)^{T} \qquad \text{......... (7)}
$$

With this, Eq. (1) can be reformulated as:

$$
(w^*{}_1, w^*{}_2, \dots, w^*{}_v) = \mathit{arg\ max}_{w_1,\dots,w_v} \frac{Tr(W^T D W)}{Tr(W^T S W)} \qquad \text{............................. (8)}
$$

The objective in Eq. (8) is in the form of trace ratio, which implies the closed form solution does not exist. We therefore relax it into a more tractable one in the form of ratio trace:

$$
(w^*{}_1, w^*{}_2, \dots, w^*{}_v) = \mathit{arg\ max}_{w_1,\dots,w_v} Tr(\frac{W^T D W}{W^T S W}) \qquad \text{........................(9)}
$$

which can be solved analytically through generalized eigenvalue decomposition.

3. Algorithm:

3.1 MVDA algorithm:

Input: X, Y

X: list of feature vectors from multiview datasets.

Y: list of output classes or labels.

1. Convert the list of features from each view into a list of pytorch tensors.

2. Group together the tensors of the same class or label i.e the tensors belonging to class 0 will appear first , then the tensors belonging to class 1 and so on.

3. Find the dimension of tensors.
4. Find the mean of each class of tensors.
5. Calculate between class and with in class scatter matrix as shown in the equation.

$$if\,(j\ ==\ r):$$

$$S_{jr} = \sum_{i=1}^{c}(\sum_{k=1}^{n_{ij}} x_{ij}x^{T}{}_{ijk} - \frac{n_{ij}n_{ij}}{n_i}\mu^{(x)}{}_{ij}\mu^{(x)T}{}_{ij}$$

$$else:$$

$$S_{jr} = \frac{n_{ij}n_{ir}}{n_i}\mu^{(x)}{}_{ij}\mu^{(x)T}{}_{ir}$$

within class scatter matrix.

$$D_{jr} = \left(\sum_{i=1}^{c}\frac{n_{ij}n_{ir}}{n_i}\mu^{(x)}{}_{ij}\mu^{(x)T}{}_{ir}\right) - \frac{1}{n}\left(\sum_{i=1}^{c} n_{ij}\mu^{(x)}{}_{ij}\right)\left(\sum_{i=1}^{c} n_{ir}\mu^{(x)}{}_{ir}\right)^{T}$$

between class scatter matrix.

6. Calculate the eigenvectors using Sw and Sb.
Using eigen decomposition find the V linear transforms by maximising Sw and minimizing Sb.

4. Documentation of API:

4.1 cca.MVDA( )

4.2 Methods:

fit_transform(X, Y)

X: list of feature vectors from multiview datasets.
Y: list of output classes or labels.

Returns the list of V linear transforms by projecting X into a common latent space.

5. Example:

5.1 Example 1

```
# import our API

import mvda


from sklearn.datasets import make_blobs

# generating random data for multi views using sklearn’s make_blobs

def generate_random_multiview_dataset(seed=138):

       np.random.seed(seed)

       X_veiw1, y = make_blobs(n_samples=10,n_features=3,  centers=3)

       X_veiw2 = np.array([x + np.random.rand(len(x.shape)) * 3 for x in (X_v1 +
       np.random.randn(3) * 7)])

       X_veiw3 = np.array([x + np.random.rand(len(x.shape)) * 3 for x in (X_v1 +
       np.random.randn(3) * -7)])

       return [torch.tensor(X_v1).float(), torch.tensor(X_v2).float(),
       torch.tensor(X_v3).float()], y


  X,Y = generate_random_multiview_dataset()


  MvDA=mvda.MVDA()

 # how to use our MVDA() API’s fit_transform method

  vTransforms = MvDA.fit_transform(X,Y)
```

```
print(len(vTransforms))
print(vTransforms)
```

6. Learning Outcomes:

We learned:

- How to implement the mathematics of any research into a python program.

- Theory of Canonical Correlation Analysis and its variants, the limitations of previous discriminant analysis techniques and how MvDA overcome these limitations.
- Working and limitations of Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA).
- How to find a common latent space from multiview datasets.
- Working of Rayleigh Quotient and Trace ratio problem.

7.Appendix A

References

A. https://www.researchgate.net/profile/Feiping_Nie/publication/24217912_Trace_Ratio_Problem_Revisited/links/0c96052057a4e1ddbb000000/Trace-Ratio-Problem-Revisited.pdf

B. http://www.unige.ch/~dlu/publications/rbstrq_preprint.pdf

C. https://scihub.wikicn.top/10.1109/TIP.2019.2913511

D. https://scihub.wikicn.top/10.1109/TPAMI.2015.2435740