

Removing Redundancies from Classification Dataset: Fisher Discriminant Analysis



Ronald Fisher

Song Liu (song.liu@Bristol.ac.uk)

Objectives

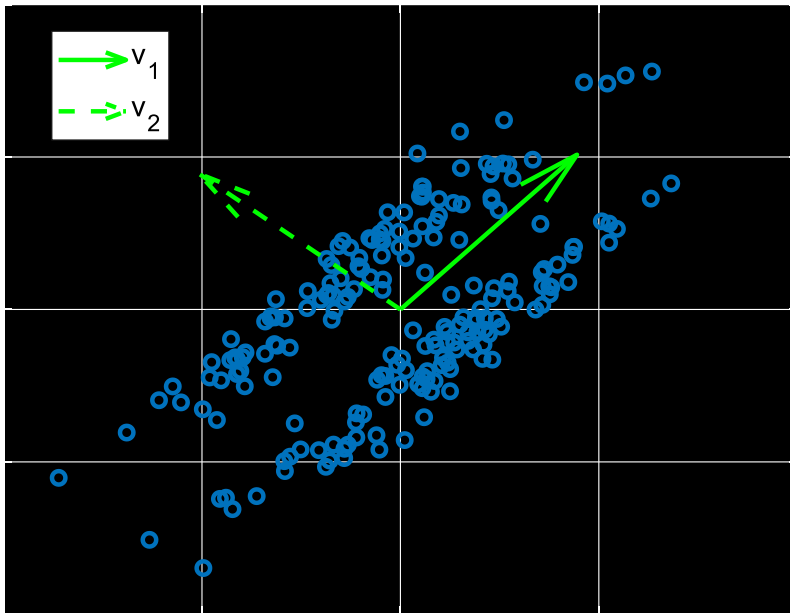
- Understand how to preserve and highlight class information when reducing dimensionality of dataset.
 - Good embedding strategies for classification tasks
- Know how to perform Fisher Discriminant Analysis (FDA)
 - Difference between FDA and PCA

Principle Component Analysis

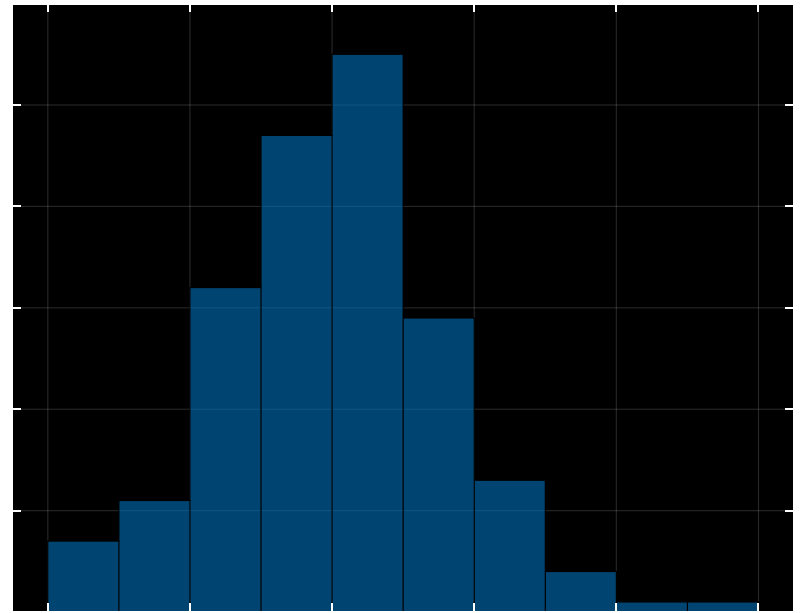
- PCA embed data points onto a lower dimensional surface, where they **spread out the most.**
 - By a trace maximization problem.
- PCA is performed by looking at eigenvectors corresp. to largest eigenvalues.

Problem of PCA

- PCA ignores class/cluster information in the dataset!



Eigenvecs



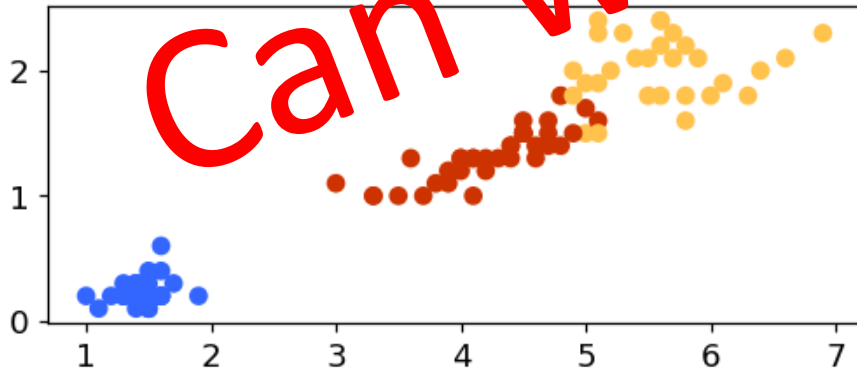
Embedding

Problem of PCA

- Although, by maximizing the spread, PCA still does an respectable job.

Manual

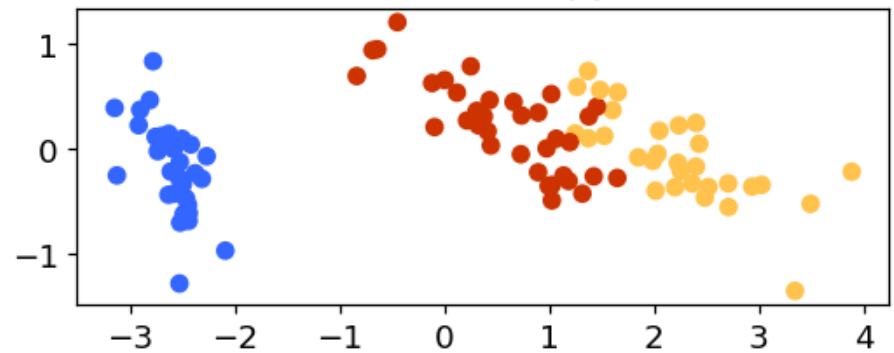
Reduced selecting features (5, 4)



Test Accuracy: 96%

PCA

Reduced with Scipy's PCA



Test Accuracy: 88%

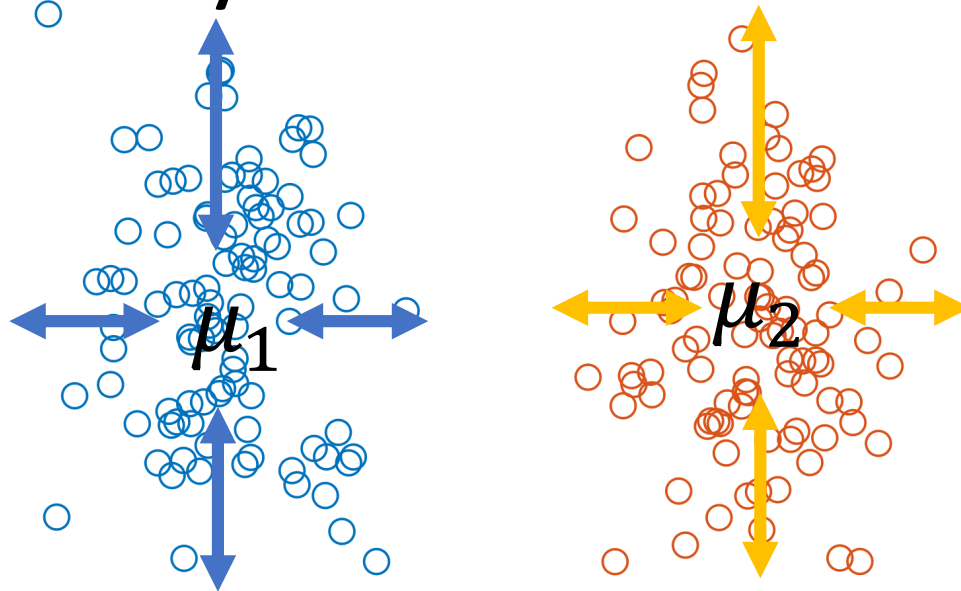
Can we do better?

Problem Setting

- Consider a classification dataset:
- $D = \{(y_i, \mathbf{x}_i)\}_{i=1}^n, \mathbf{x} \in R^d, y \in \{1 \dots k\}$.
- Find feature transform function $f(\mathbf{x}) \in R^m$ to reduce dimensionality of dataset.
 - while preserving distinct **class separation**.

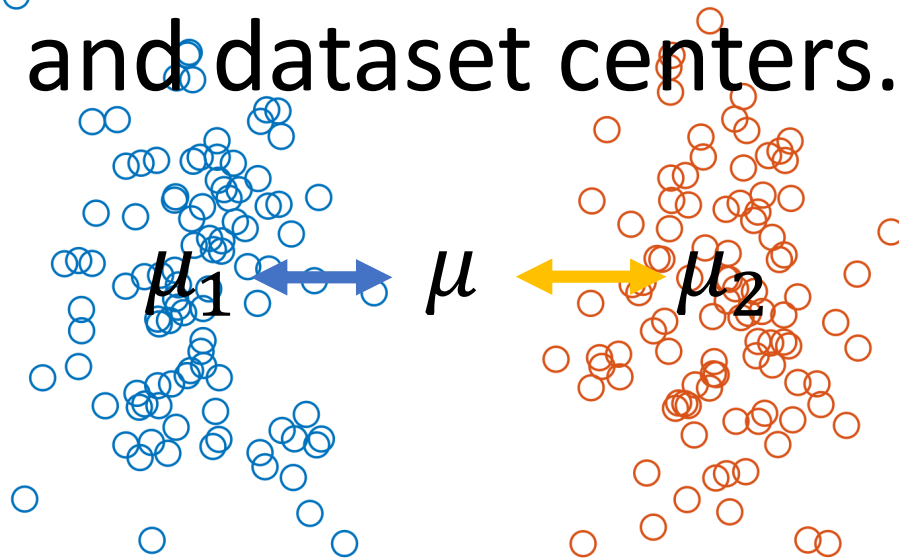
What is a Good Embedding for a Classification Dataset?

- Points **within** the same class are close to each other.
- Within classes **scatterness** can be measured by distances to class center.



What is a Good Embedding for a Classification Dataset?

- Points **between** different classes are far apart from each other.
- Between classes **scatterness** can be measured by distances between class centers and dataset centers.



Within-class Scatterness

- Embedding is $\mathbf{B}\mathbf{x}^\top$.
- Embedded center for class k :

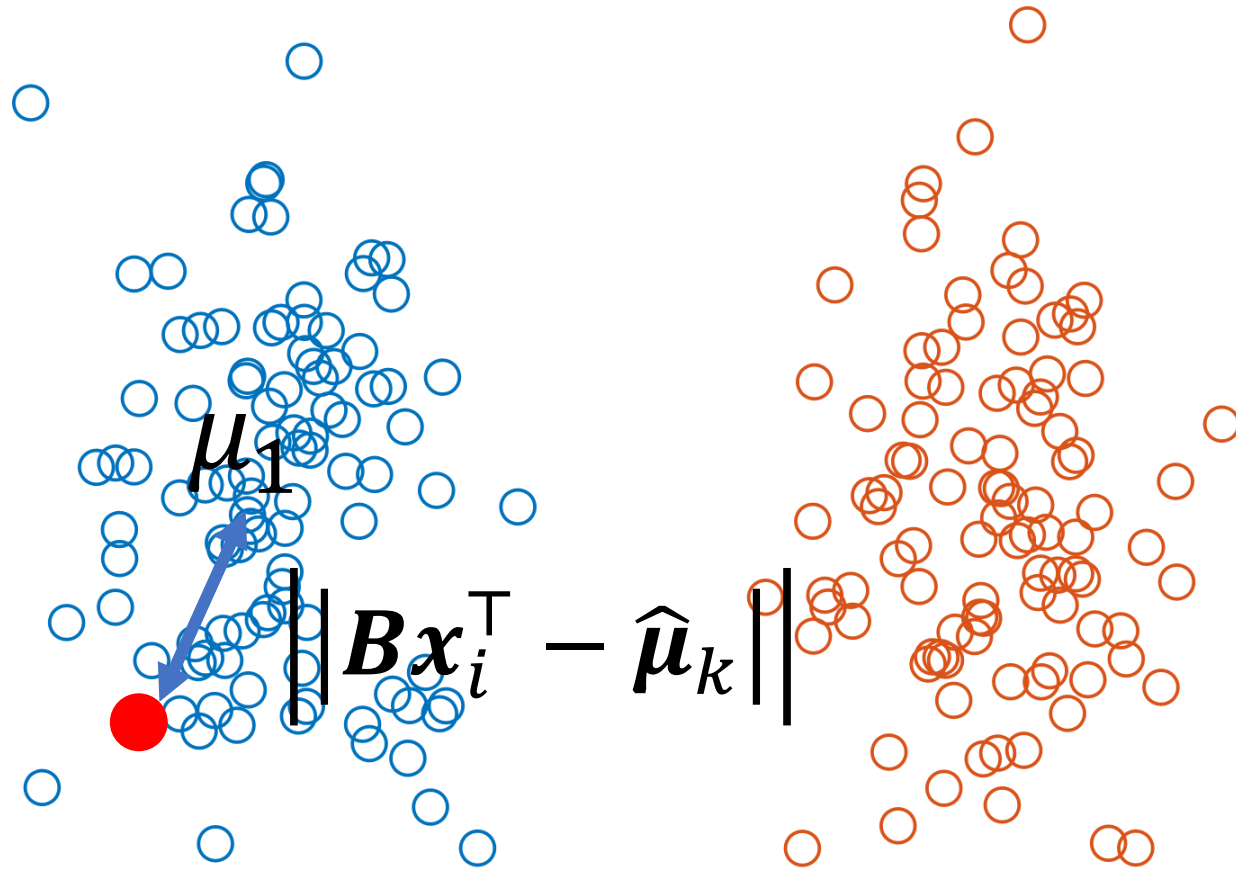
$$\bullet \hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i, y_i=k} \mathbf{B}\mathbf{x}_i^\top$$

- Within class scatterness of class k :

$$S_{W,k} = \sum_{i, y_i=k} \left\| \mathbf{B}\mathbf{x}_i^\top - \hat{\boldsymbol{\mu}}_k \right\|^2$$

- Sum over points in **individual** classes.

Within-class Scatterness



Between-class Scatterness

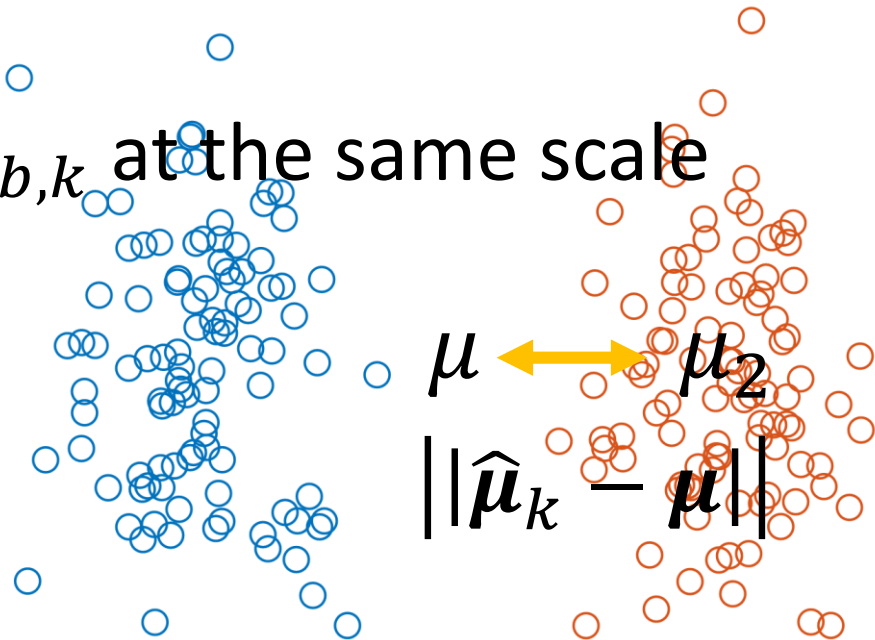
- Embedded dataset centroid:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^n \mathbf{B} \mathbf{x}_i^{\top}$$

- Between-class scatterness

$$s_{b,k} = n_k \left| \hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu} \right|^2$$

- n_k is needed to make $s_{b,k}$ at the same scale with $s_{w,k}$.



Objective

- **Maximizing** between class scatterness \forall_k .
- **Minimize** within class scatterness \forall_k .

- $\max_B \sum_k S_{b,k} - \sum_k S_{w,k}$

- $\sum_k S_{b,k} = \text{tr}\{\mathbf{B}[\sum_k n_k (\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})^\top (\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})]\mathbf{B}^\top\}$

- $\sum_k S_{w,k} = \text{tr}\{\mathbf{B}[\sum_k \sum_i (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^\top (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)]\mathbf{B}^\top\}$

- Live demonstration

Objective

- Let $\mathbf{S}_w := \sum_k \sum_i (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^\top (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)$

- Let $\mathbf{S}_b := \sum_k n_k (\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})^\top (\hat{\boldsymbol{\mu}}_k - \hat{\boldsymbol{\mu}})$

- $\max_B \sum_k S_{b,k} - \sum_k S_{w,k}$
 $= \max_B \text{tr}[\mathbf{B}\mathbf{S}_b\mathbf{B}^\top] - \text{tr}[\mathbf{B}\mathbf{S}_w\mathbf{B}^\top]$

Objective

- However, the above problem is **very hard to solve!**

- Like PCA, we make the problem easier by introducing a constraint on \mathbf{B} .

- Constrained Objective:

- $\max_{\mathbf{B}, \mathbf{B}^T \mathbf{S}_w \mathbf{B} = \mathbf{I}} \text{tr}[\mathbf{B} \mathbf{S}_b \mathbf{B}^T] - \text{tr}[\mathbf{B} \mathbf{S}_w \mathbf{B}^T]$

- $\max_{\mathbf{B}, \mathbf{B}^T \mathbf{S}_w \mathbf{B} = \mathbf{I}} \text{tr}[\mathbf{B} \mathbf{S}_b \mathbf{B}^T]$

Solution

- Eigenvalue/eigenvectors of A
 - $A\mathbf{v}_i = \lambda_i\mathbf{v}_i$
- Generalized eigenvalue/eigenvectors of A and B
 - $A\mathbf{v}_i = \lambda_i B\mathbf{v}_i$
 - MATLAB: `[V,LAMBDA] = eig(A,B)`
 - Python: `scipy.linalg.eig(A,B)`

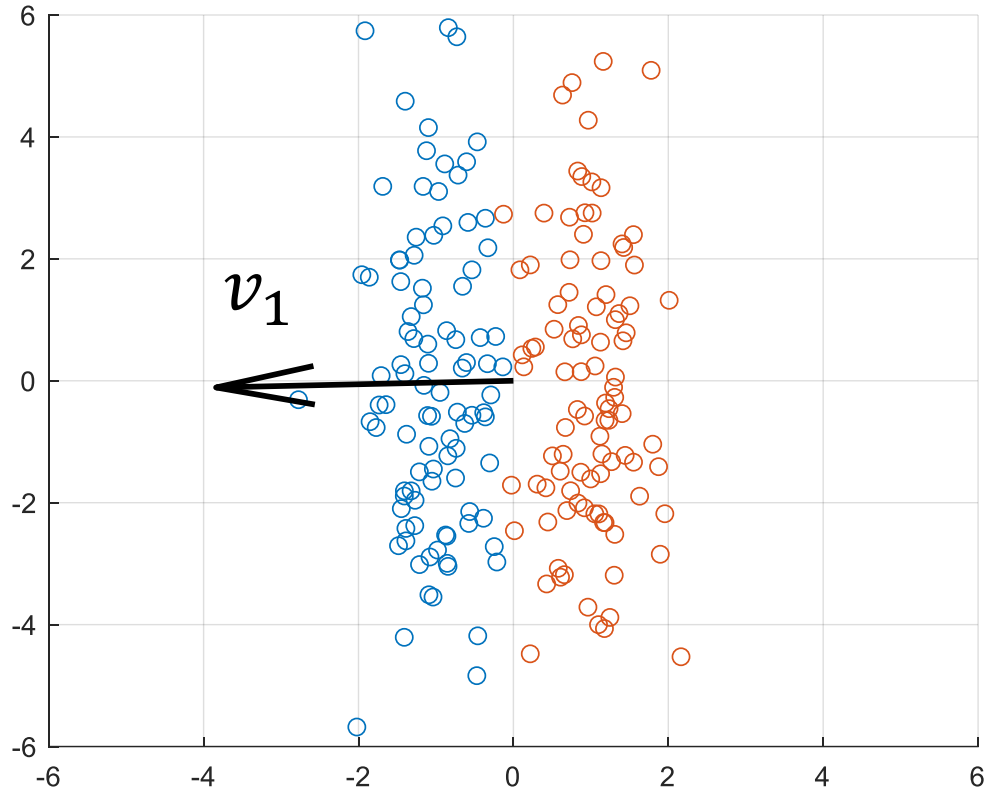
Solution

- $\max_{\mathbf{B}, \mathbf{B} \mathbf{S}_w \mathbf{B}^\top = \mathbf{I}} \text{tr}[\mathbf{B} \mathbf{S}_b \mathbf{B}^\top]$
- The embedding matrix $\hat{\mathbf{B}}$ can be constructed by
- $\hat{\mathbf{B}} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]^\top$
 - $(\lambda_1, \mathbf{v}_1), \dots, (\lambda_m, \mathbf{v}_m)$ are m largest generalized eigenval. and eigenvec. of
 - $\mathbf{S}_b \mathbf{v}_i = \lambda_i \mathbf{S}_w \mathbf{v}_i$

Solution

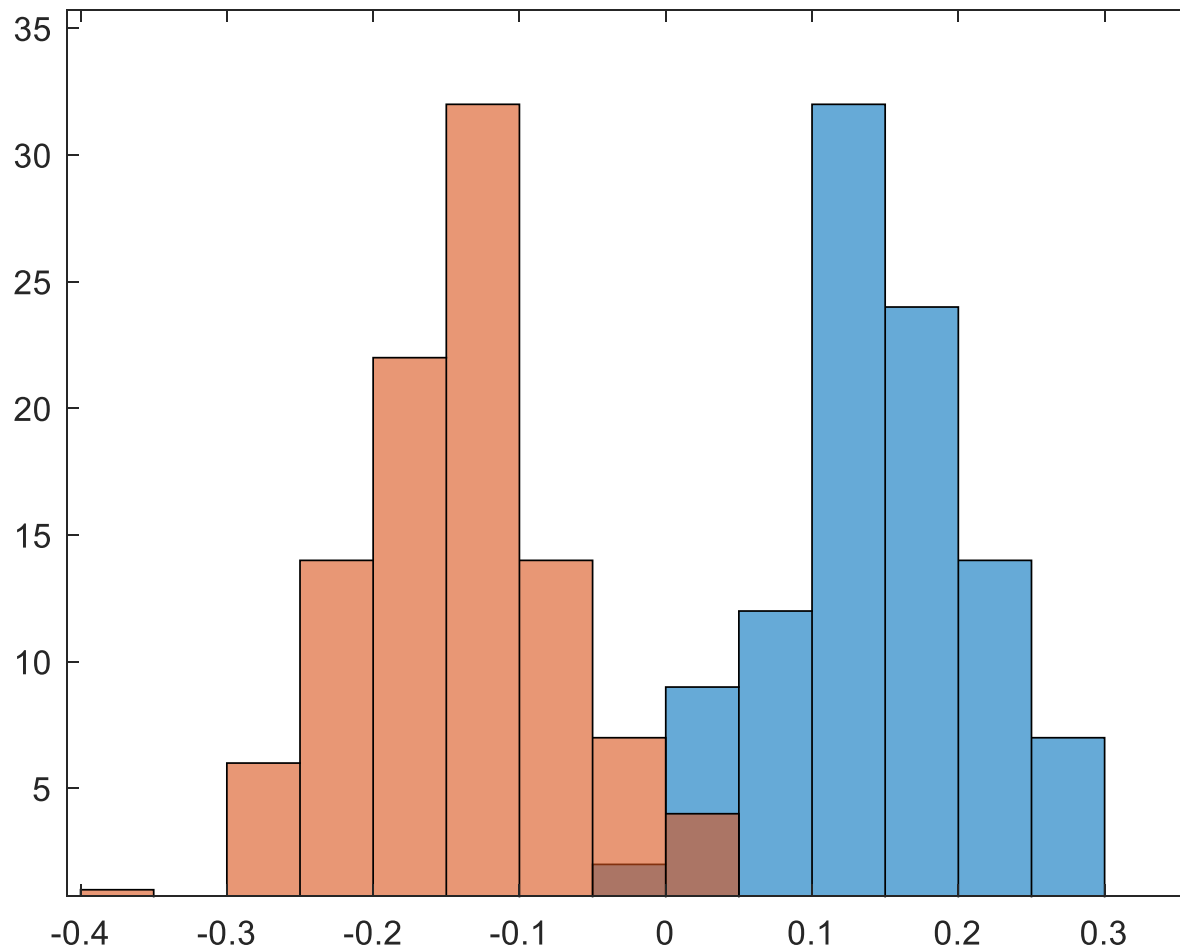
- Unfortunately, $m < c - 1$.
 - For a binary classification dataset, the embedding has to be 1D.
 - $\text{rank}(\mathbf{S}_b) = c - 1$
- The process of computing embedding using eigenvec. of \mathbf{S}_b and \mathbf{S}_w is called **Fisher Discriminant Analysis (FDA)**.

Example: Binary Classification Dataset



FDA embeds samples to a subspace that is the most **linearly** separable.

Example: embedding, $v_1^T x^T$



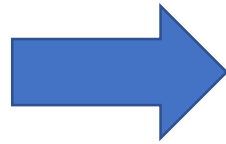
Class separation is preserved
after embedding.

Eigenfaces

$$X = \{\mathbf{x}'_i\}, \mathbf{x}'_i \in R^{d' \times d'}$$



\mathbf{x}'_i



$$\mathbf{x}_i \in R^{d' d'}$$



• **Perform PCA -> Eigenvectors**

• $[\mathbf{v}_1 \dots \mathbf{v}_m], \mathbf{v}_i \in R^{d' d' \times 1}$

Eigenfaces



$$\mathbf{v}_i \in R^{d' d'}$$



$$\mathbf{v}'_i \in R^{d' \times d'}$$

- Eigenfaces have huge applications in facial recognition.

Conclusion

- Good embedding of a classification dataset should have:
 - Small within class scatter
 - Large between class scatter
- FDA maximizes between class scatter and minimizes within class scatter
 - Preserves class separation on datasets.