Removing Redundancies from Labelled Data: Fisher Discriminant Analysis

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

Objectives

- Understand how to preserve and highlight class information when reducing dimensionality of dataset.
  - Good embedding for classification

- Know how to perform Fisher Discriminant Analysis (FDA)

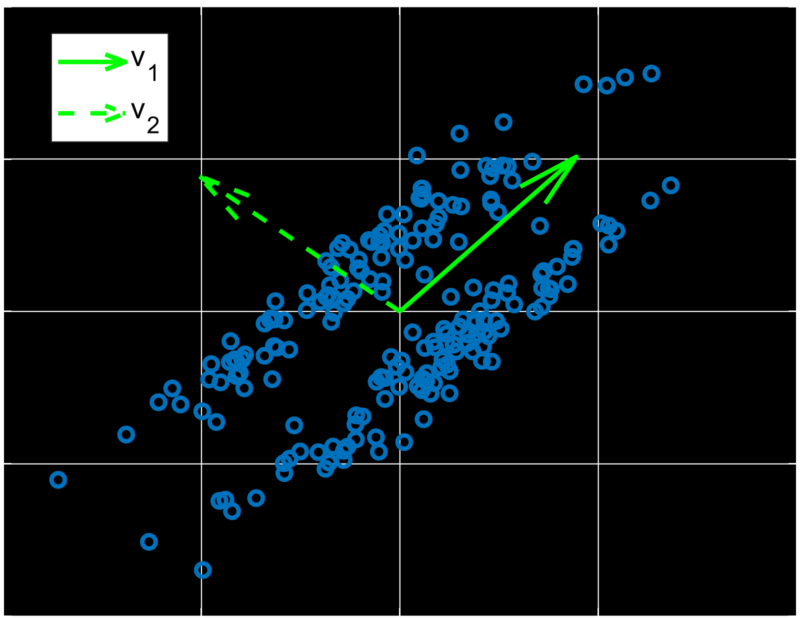
Principle Component Analysis

- PCA embed data points onto a lower dimensional surface, where they **spread out the most.**
    - By a trace maximization problem.

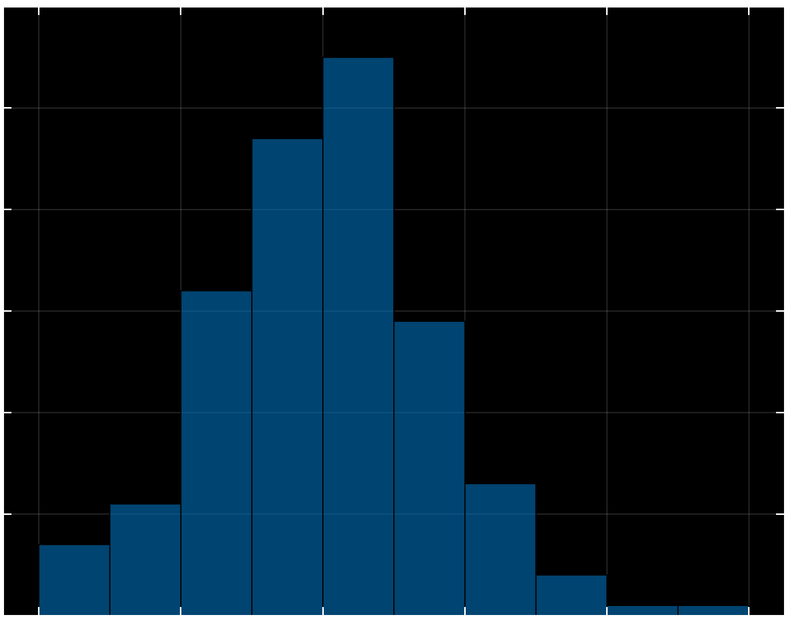
- PCA is performed by looking at eigenvectors corresp. to largest eigenvalues.

Problem of PCA

- PCA ignores class/cluster information in the dataset!



Eigenvecs



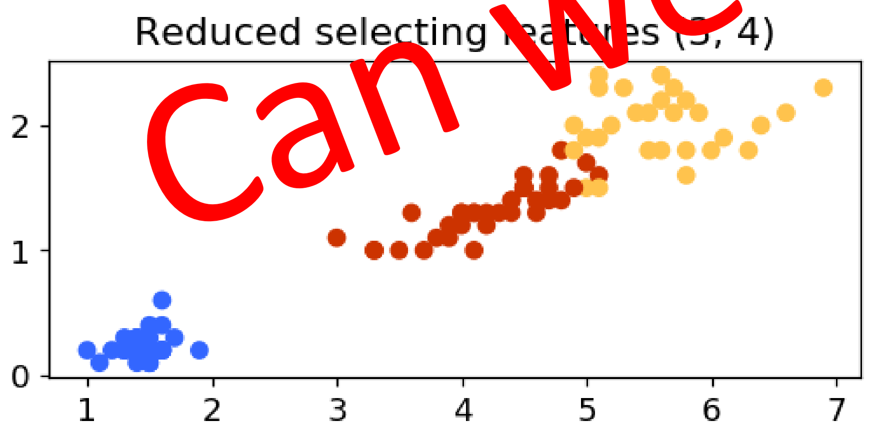
Embedding

Problem of PCA

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

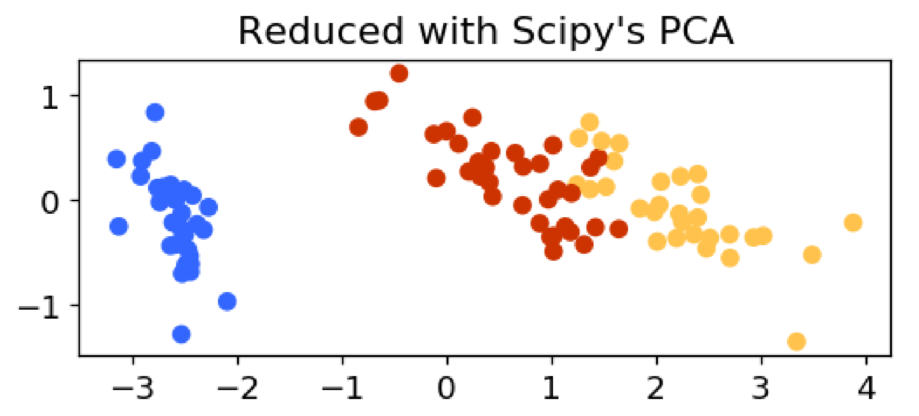
Can we do better?

Manual



Test Accuracy: 96%

PCA



Test Accuracy: 88%

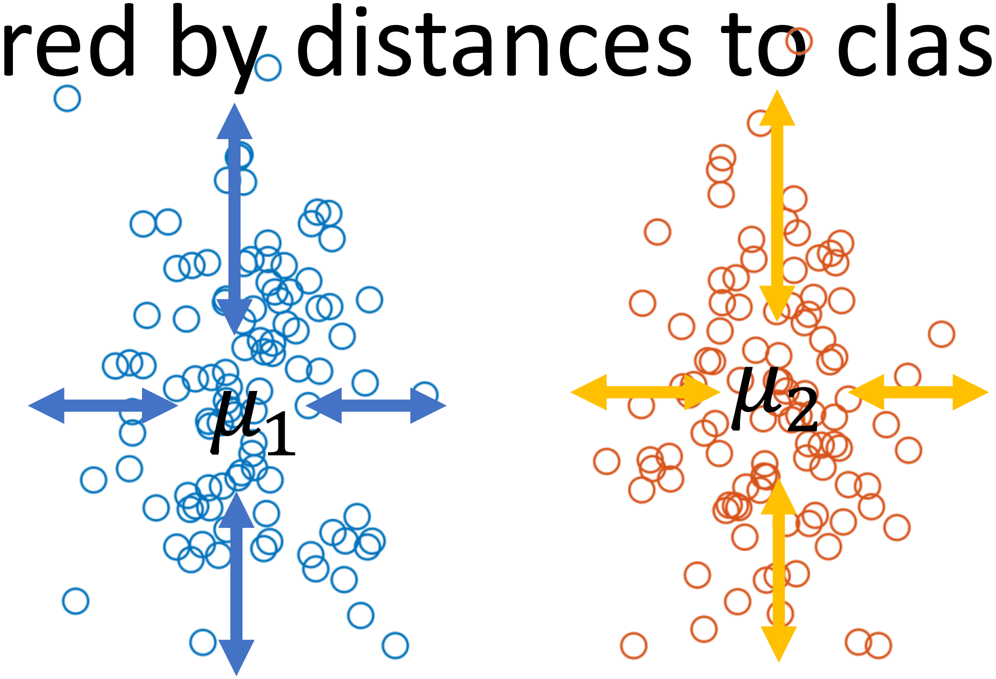
Problem Setting

- Consider a classification dataset:
- $D = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^{n}, \boldsymbol{x} \in R^d, y \in \{1 \dots k\}$.

- Find feature transform function $\boldsymbol{f}(\boldsymbol{x}) \in R^m$ to reduce dimensionality of dataset.
  - while preserving distinct **class separation**.

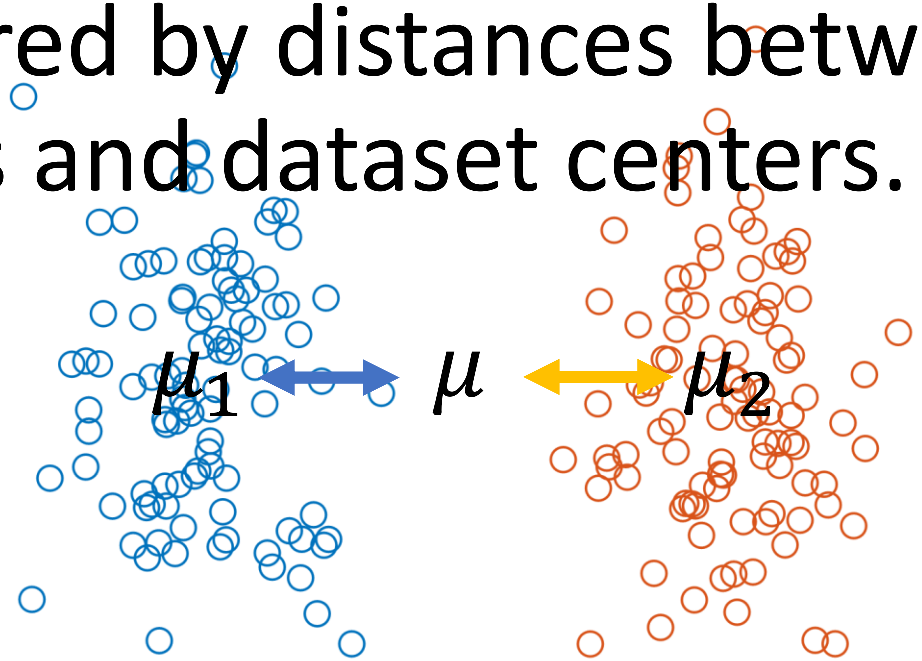
What is a Good Embedding for a Classification Dataset?

- 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 $\boldsymbol{B}\boldsymbol{x}^{\top}$.
- Embedded center for class k:
  - $\widehat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i, y_i = k} \boldsymbol{B}\boldsymbol{x}_i^{\top}$
- Within class scatterness of class k:

$$s_{w,k} = \sum_{i, y_i = k} \left|\left|\boldsymbol{B}\boldsymbol{x}_i^{\top} - \widehat{\boldsymbol{\mu}}_k\right|\right|^2$$

Between-class Scatterness

- Embedded dataset centroid:
  - $\widehat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{B}\boldsymbol{x}_i^{\top}$
- Between-class scatterness
  - $s_{b,k} = n_k \left|\left|\widehat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}\right|\right|^2$

Objective

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

- $\max_{\boldsymbol{B}} \boxed{\sum_k s_{b,k}} - \boxed{\sum_k s_{w,k}}$
- $\sum_k s_{b,k} = \mathrm{tr}\{\boldsymbol{B}[\sum_k n_k(\widehat{\boldsymbol{\mu}}_k - \widehat{\boldsymbol{\mu}})^\top(\widehat{\boldsymbol{\mu}}_k - \widehat{\boldsymbol{\mu}})]\boldsymbol{B}^\top\}$
- $\sum_k s_{w,k} = \mathrm{tr}\{\boldsymbol{B}[\sum_k \sum_i(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)]\boldsymbol{B}^\top\}$

- Live demonstration

Objective

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

- $\max\limits_{\boldsymbol{B}} \sum_k s_{b,k} - \sum_k s_{w,k}$
$$= \max_{\boldsymbol{B}} \mathrm{tr}[\boldsymbol{B}\boldsymbol{S}_{\boldsymbol{b}}\boldsymbol{B}^\top] - \mathrm{tr}[\boldsymbol{B}\boldsymbol{S}_{\boldsymbol{w}}\boldsymbol{B}^\top]$$

Objective

- However, the above problem is **very hard to solve!**
  - Like PCA, we make the problem easier by introducing a constraint on $\boldsymbol{B}$.

- Final Objective:
  - $\max_{\boldsymbol{B}, \boldsymbol{B}\boldsymbol{S}_w\boldsymbol{B}^\top = \boldsymbol{I}} \operatorname{tr}[\boldsymbol{B}\boldsymbol{S}_b\boldsymbol{B}^\top] - \operatorname{tr}[\boldsymbol{B}\boldsymbol{S}_w\boldsymbol{B}^\top]$
  - $\max_{\boldsymbol{B}, \boldsymbol{B}\boldsymbol{S}_w\boldsymbol{B}^\top = \boldsymbol{I}} \operatorname{tr}[\boldsymbol{B}\boldsymbol{S}_b\boldsymbol{B}^\top]$

Solution

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

Solution

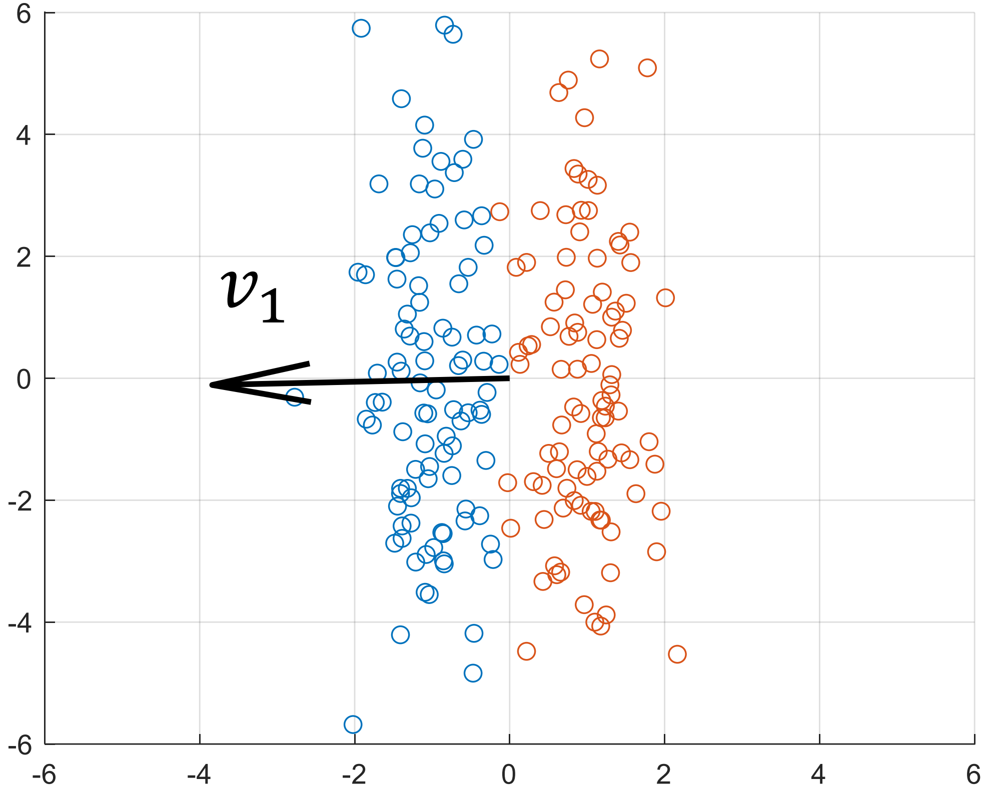
- $\max_{\boldsymbol{B}, \boldsymbol{B}\boldsymbol{S}_w\boldsymbol{B}^\top = \boldsymbol{I}} \operatorname{tr}[\boldsymbol{B}\boldsymbol{S}_b\boldsymbol{B}^\top]$
- The embedding matrix $\widehat{\boldsymbol{B}}$ can be constructed by
- $\widehat{\boldsymbol{B}} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_m]^\top$
    - $(\lambda_1, \boldsymbol{v}_{\mathbf{1}}), \ldots, (\lambda_m, \boldsymbol{v}_m)$ are m largest generalized eigenval. and eigenvec. of
    - $\boldsymbol{S}_b\boldsymbol{v}_i = \lambda_i\boldsymbol{S}_w\boldsymbol{v}_i$

Solution

- Unfortunately, $m < \mathrm{c} - 1$.
  - For a binary classification dataset, the embedding has to 1D.
  - $\mathrm{rank}(\boldsymbol{S}_b) = c - 1$

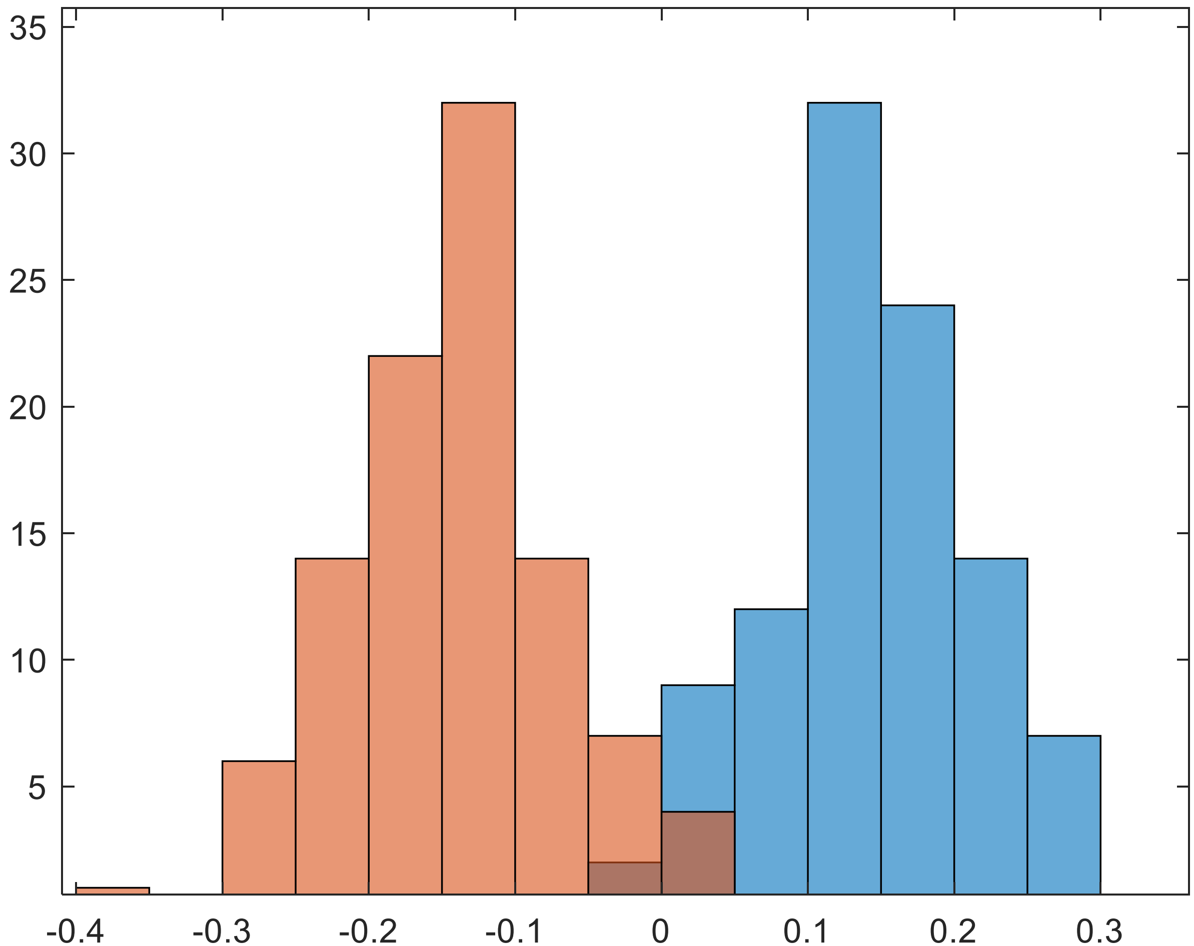
- The process of computing embedding using eigenvec. of $\boldsymbol{S}_b$ and $\boldsymbol{S}_w$ is called **Fisher Discriminant Analysis (FDA).**

Example: Binary Classification Dataset



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

Example: embedding, $\boldsymbol{v}_1^\top \boldsymbol{x}^\top$



Class separation is preserved after embedding.

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.