# 模式识别

特征归一化

Fisher线性判别分析

张振宇

智能科学与技术学院

2025

# 目标

- ✓ 掌握并能应用常见的特征归一化方法
- ✓ 能应用FLD，并能掌握其推导过程
- ✓ 能将PCA和FLD应用到人脸识别当中去

- ✓ 提高目标
  - 进一步能将本章方法应用到实际研究问题中去
  - 对线性判别在不同条件下的变化，有兴趣的可以进一步阅读

# 特征归一化

Feature normalization

# 1. 每维度归一

- ✓ per-dimension normalization
  - 虚拟的例子（判别性别）
    - 假设用两个特征：身高和体重
    - 如果1. 身高单位毫米，体重单位吨，那么？
    - 如果2. 身高单位公里，体重单位克，那么？
  - 很多时候，不同的维度需要统一到同样的取值范围！
- ✓ 训练集：$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$，　　$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$
  - 对每一维$j$，其数据为$x_{1j}, x_{2j}, \ldots, x_{nj}$
  - 取其最小值 $x_{min,j}$和最大值$x_{max,j}$
  - 对这一维的任何数据　$x_{ij} \leftarrow \frac{x_{ij} - x_{min,j}}{x_{max,j} - x_{min,j}}$

# 1. 稀疏数据

- ✓ 新数据的范围是？各维度统一了吗？
  - [0 1] （训练集中的情况）
  - 若某一维$x_{max,j} = x_{min,j}$？
  - 也可以统一到[-1 1]

$$x_{ij} \leftarrow 2 \times \left( \frac{x_{ij} - x_{min,j}}{x_{max,j} - x_{min,j}} - 0.5 \right)$$

- ✓ 稀疏数据sparse data：数据中很多维度值为0
  - 如果所有数据≥ 0，在两种归一化中，原来是0的会变成什么？

# 1. 适用场景

- ✓ 何时应该使用min-max归一化？
  - 特征量纲差异大：消除量纲影响
  - 模型对于输入范围敏感
  - 数据分布有明确的最大/最小值（什么时候fail？）
  - 对数值解释有直观需求

# 2.$\ell_2$或$\ell_1$归一化

- ✓ 若各维度取值范围的不同是有意义的，但是不同数据点之间的大小（如向量长度norm）应保持一致
  - 对每个数据$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$

$$x_{ij} \leftarrow \frac{x_{ij}}{\|\boldsymbol{x}_i\|_{\ell_2}} \qquad \|\boldsymbol{x}_i\|_{\ell_2} = \sqrt{\boldsymbol{x}_i^T \boldsymbol{x}_i}$$

- ✓ $\ell_1$归一化
  - 适用于非负的特征，即$x_{ij} \geq 0$总成立
  - 若数据$\boldsymbol{x}_i$是直方图(histogram)时，经常是最佳的

$$x_{ij} \leftarrow \frac{x_{ij}}{\|\boldsymbol{x}_i\|_{\ell_1}} \qquad \|\boldsymbol{x}_i\|_{\ell_1} = \sum_{j=1}^{d} |x_{ij}|$$

# 3.zero-score标准化

- ✓ 有时候有理由相信每一个维度是服从高斯分布的
  - 希望每一个维度归一化到$N(0,1)$
- ✓ 对每一维$j$，其数据为$x_{1j}, x_{2j}, \ldots, x_{nj}$
  - 计算其均值$\hat{\mu}_j$和方差$\hat{\sigma}_j^2$
  - 对每一个特征值

$$x_{ij} \leftarrow \frac{x_{ij} - \hat{\mu}_j}{\hat{\sigma}_j}$$

# 3.Robust Scaling

- ✓ 针对含离群值数据的归一化方法
  - $x_{ij} \leftarrow \frac{x_{ij} - \text{median}}{IQR}$
  - IQR: 四分位距
  - 将一组有序数据分为四等分的三个点：
    - 第一四分位数Q1：第25%位置的值
    - 第二四分位数Q2：第50%位置的值，即中位数
    - 第三四分位数Q3：第75%位置的值
    - IQR = Q3 – Q1

# 归一化测试数据

- ✓ 怎样归一化测试数据？
  - 从测试集寻找最大值、最小值、均值？
- ✓ 除了在测试的时候，永远不要使用测试数据！
  - 测试集和训练集应该使用相同的归一化方法
    - 还记得吗？训练和测试集应该从相同的$p(\boldsymbol{x})$取样
    - 同样的归一化会保持这个限定！
  - 这个原则同样适用于交叉验证！
- ✓ 那么，怎样做？
  - 保存从训练集上取得的归一化参数(parameter)
  - 使用同样的公式和保存的参数来归一化测试集

# 小结

- ✓ 归一化的方法应该是根据数据的特点来选择的
  - 在做任何机器学习之前，先搞清你的数据的特点
    - 稀疏？
    - 每一维有没有含义？
    - 每一维里面值的分布情况？Gauss？
    - 看你的数据！Do visualization!
- ✓ 归一化可能对准确度有极大的影响！
  - 在有些例子里，正确的归一化能大幅度提高accuracy
- ✓ 不同的归一化方法可以混合使用

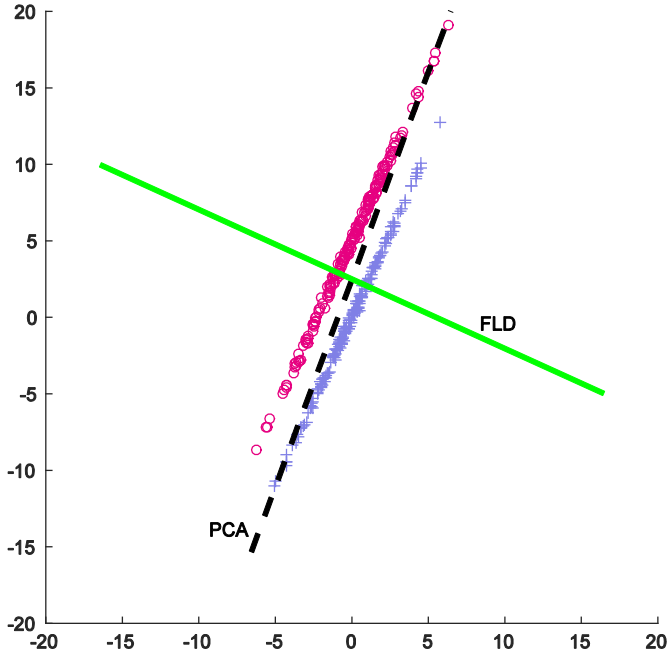
# Fisher线性判别分析

Fisher's Linear Discriminant analysis (FLD, 或有时候LDA)

# 为什么需要FLD？

- ✓ 理论上可以证明，PCA在数据是单个高斯分布时是最佳的
  - PCA有利于表示数据，但和分类无关
- ✓ 分类问题中，不同类别的分布$p(\boldsymbol{x}|y=i)$不能相同
- ✓ 如何提取特征(extract feature)，最有利于分类？
  - FLD是某些限制条件下最佳的线性特征提取方法 optimal linear feature extraction method under certain assumptions

# Idea: FLD的动机（motivation）




Bad linear feature (PCA)

Good linear feature (FLD)

# 用数学形式表示formalize

- ✓ 两个类别$y_i \in \{1,2\}$，数据$\boldsymbol{x}_i$，两类各有$N_1, N_2$个点
- ✓ 希望寻找一个投影方向projection direction，$u = \boldsymbol{w}^T \boldsymbol{x}$，使得两个类别的数据在投影以后容易被分开separate

- ✓ 两个类各自的均值为
  - $\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{y_i=1} \boldsymbol{x}_i$，　　$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{y_i=2} \boldsymbol{x}_i$
  - 投影以后的均值为$m_1 = \boldsymbol{w}^T \boldsymbol{\mu}_1$，　$m_2 = \boldsymbol{w}^T \boldsymbol{\mu}_2$

# Objective: Fisher's Criterion

- ✓ 怎样描述“分开”的程度(separation)?
- ✓ Maximize $(m_2 - m_1)^2$? 问题?
  - 这个值可以无限大。怎么解决?
  - 加限制条件$\boldsymbol{w}^T\boldsymbol{w} = 1$
  - 看前面的图，这个值不是越大越好。怎么解决?
- ✓ Fisher准则
  - 在要求$|m_2 - m_1|$尽量大的同时，要求两类在投影以后尽量集中，或者不分散。怎么度量分散程度?

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

# 分散程度的度量

- ✓ 对一维数据，自然的度量是方差或散度 ($k = 1,2$)

$$s_k^2 = \sum_{y_i=k} (u_i - m_k)^2$$

  - 称为类内散度within class scatter
- ✓ $s_1^2 + s_2^2$：总的类内散度
  - total within-class scatter
- ✓ $s_k^2 = \sum_{y_i=k} (u_i - m_k)^2 = \sum_{y_i=k} \left(\boldsymbol{w}^T (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right)^2 = \boldsymbol{w}^T \sum_{y_i=k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{w}$
- ✓ $(m_2 - m_1)^2 = \boldsymbol{w}^T (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \boldsymbol{w}$

# 散度矩阵

- ✓ $S_k = \sum_{y_i=k}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$ 是什么？
- ✓ 类内散度矩阵within-class scatter matrix

$$S_W = S_1 + S_2$$

- ✓ 类间散度矩阵between-class scatter matrix

$$S_B = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T$$

- ✓ Fisher准则的矩阵形式
  - $\max J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}},\ s.t. \boldsymbol{w}^T \boldsymbol{w} = 1$
  - 这种形式称为广义瑞利商generalized Rayleigh quotient

# Optimization: 如何求解？

- ✓ (Simplification/transformation)：用拉格朗日乘子法，证明（记得查表）最优时必须满足

$$S_B \boldsymbol{w} = \lambda S_W \boldsymbol{w}$$

- ✓ 该问题称为广义特征值generalized eigenvalue问题
  - 得到“$S_B$ 和$S_W$”的广义特征值和广义特征向量
  - Generalized eigenvalue (eigenvector) of $S_B$ and $S_W$
- ✓ 但是我们不用去解这个问题
  - $S_B \boldsymbol{w} = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \boldsymbol{w} \propto (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$
  - $(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) = \lambda S_W \boldsymbol{w}$!

# FLD的步骤

1. 计算$\boldsymbol{\mu}_2, \boldsymbol{\mu}_1$
2. 计算$S_W$
3. 计算$\boldsymbol{w} = S_W^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$
4. 归一化：

$$\boldsymbol{w} \leftarrow \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

# 如果不可逆怎么办？

- ✓ 如果数据很少或者维度很高， $S_W$很可能不可逆
  - 广义逆矩阵generalized inverse matrix
- ✓ $S_W$是实对称的，而且至少是半正定的
  - $S_W = E\Lambda E^{\mathrm{T}},\ \lambda_{ii} \geq 0$
- ✓ Moore–Penrose伪逆pseudoinverse
  - 若$\lambda_{ii} > 0$, 定义$\lambda_{ii}^{+} = 1/\lambda_{ii}$, 否则定义$\lambda_{ii}^{+} = 0$
  - $\Lambda$的M-P伪逆为：$\Lambda^{+} = \mathrm{diag}(\lambda_{11}^{+}, \lambda_{22}^{+}, \ldots, \lambda_{dd}^{+})$
  - $S_W$的伪逆为

$$S_W^{+} = E\Lambda^{+}E^{T}$$

# 如果大于2类怎么办？

- ✓ $C$类问题
  - $\boldsymbol{\mu}_i, N_i, m_i, S_i$和2类问题中一样定义
  - $S_W = \sum_{i=1}^{C} S_i$, 很容易从2类问题推广
  - 定义$N = \sum_{i=1}^{C} N_i$
  - 定义总均值$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{C} N_i \boldsymbol{\mu}_i = \frac{1}{N}\sum_{\boldsymbol{x}} \boldsymbol{x}$
- ✓ $S_B$没有定义，无法直接从2类问题推广
  - 总散度矩阵total scatter matrix, $S_T = \sum_{\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T$
  - $S_T = S_W + \sum_{i=1}^{C} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T = S_W + S_B$
  - 定义多类的 $S_B = \sum_{i=1}^{C} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$
  - 证明，当$C = 2$时， 有$S_T = S_W + S_B$

# 更多的投影方向

$$\max J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}$$

- ✓ 求解广义特征值问题

$$S_B \boldsymbol{w}_i = \lambda_i S_W \boldsymbol{w}_i$$

- ✓ 最多能得到$C-1$个有效的投影方向
  - 为什么？

# FLD的主要变体

- ✓ 正则化FLD
  - $S'_W = S_W + \epsilon \mathrm{I}$
  - 提高数值稳定性，适用于样本数少于特征数的场景
  - 需要手动调整正则化参数

- ✓ Kernel FLD
  - 通过核函数将数据映射到高维特征空间，处理非线性可分数据
  - 需要选择核函数调参，计算复杂度高

# FLD的主要变体

- ✓ 2D-FLD
  - 每个样本图像为$\mathbf{X}_k$找到行投影矩阵$\mathbf{U}$和列投影矩阵$\mathbf{V}$
  - 类内散度矩阵

$$\mathbf{S}_w^{\text{row}} = \sum_{c=1}^{C} \sum_{\mathbf{X}_k \in c} (\mathbf{X}_k - \mathbf{M}_c)(\mathbf{X}_k - \mathbf{M}_c)^\top \qquad \mathbf{S}_w^{\text{col}} = \sum_{c=1}^{C} \sum_{\mathbf{X}_k \in c} (\mathbf{X}_k - \mathbf{M}_c)^\top (\mathbf{X}_k - \mathbf{M}_c)$$

  - 类间散度矩阵

$$\mathbf{S}_b^{\text{row}} = \sum_{c=1}^{C} N_c (\mathbf{M}_c - \mathbf{M})(\mathbf{M}_c - \mathbf{M})^\top \qquad \mathbf{S}_b^{\text{col}} = \sum_{c=1}^{C} N_c (\mathbf{M}_c - \mathbf{M})^\top (\mathbf{M}_c - \mathbf{M})$$

  - 最大化投影后数据的 Fisher准则

$$J(\mathbf{U}, \mathbf{V}) = \frac{\text{tr}\left(\mathbf{U}^\top \mathbf{S}_b^{\text{row}} \mathbf{U}\right)}{\text{tr}\left(\mathbf{U}^\top \mathbf{S}_w^{\text{row}} \mathbf{U}\right)} \cdot \frac{\text{tr}\left(\mathbf{V}^\top \mathbf{S}_b^{\text{col}} \mathbf{V}\right)}{\text{tr}\left(\mathbf{V}^\top \mathbf{S}_w^{\text{col}} \mathbf{V}\right)}$$

# FLD的主要变体

- ✓ 由于同时优化 $\mathbf{U}$ 和 $\mathbf{V}$ 是耦合的非凸问题，采用交替迭代法：

1. **初始化**：随机生成 $\mathbf{U}^{(0)}$ 和 $\mathbf{V}^{(0)}$，或使用PCA初始化。
2. **交替优化**：
   - **步骤1**：固定 $\mathbf{V}^{(t)}$，通过广义特征值分解更新 $\mathbf{U}^{(t+1)}$。
   - **步骤2**：固定 $\mathbf{U}^{(t+1)}$，通过广义特征值分解更新 $\mathbf{V}^{(t+1)}$。
3. **收敛判断**：若 $\left|J^{(t+1)} - J^{(t)}\right| < \epsilon$ 或 $t \geq T$，停止迭代；否则 $t = t + 1$，返回步骤1。
4. **输出**：投影矩阵 $\mathbf{U}$ 和 $\mathbf{V}$，以及低维特征 $\mathbf{Y}_k = \mathbf{U}^\top \mathbf{X}_k \mathbf{V}$。

- ✓ 在ORL数据集上识别率从传统FLD的89%提升至96%，计算时间减少60%。

# FLD的主要变体

- ✓ 稀疏FLD
  - $\max J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}},\ s.t.\ \|\boldsymbol{w}\|_1 \leq \boldsymbol{\tau}$
  - 提高模型可解释性，适用于高维数据特征筛选
  - 优化过程复杂，可能损失部分判别信息

$$\max_{\mathbf{w}} \left\{ \frac{\mathbf{w}^\top \mathbf{S}_b \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_w \mathbf{w}} - \lambda \|\mathbf{w}\|_1 \right\}$$

  - 需要利用近端梯度下降迭代求解

# FLD的主要变体

- ✓ 鲁棒FLD
  - $\boldsymbol{\mu}_i \rightarrow \text{Median}(\mathbf{X}_i), S_W \rightarrow \text{Robust Covariance}$
  - 对噪声和离群点鲁棒，提升模型稳定性
  - 计算复杂度高，鲁棒协方差估计在小样本下不可靠

- ✓ 鲁棒的协方差矩阵计算方法
  - 最小协方差行列式（Minimum Covariance Determinant, MCD）
  - 寻找一个包含 h 个样本的子集（h>n/2），使得其协方差矩阵的行列式最小，从而排除离群点。

# FLD的主要变体

- ✓ 增量FLD
  - $\boldsymbol{\mu}_{new} = \frac{N\boldsymbol{\mu}_{old}+\boldsymbol{x}_{new}}{N+1}$
  - 适用于实时数据或大规模流式数据（监控，网络）
  - 长期增量更新会导致累计误差

- ✓ 其核心目标是在新数据到达时，无需重新计算全部历史数据，而是通过增量更新类间散度矩阵和类内散度矩阵

# PCA + FLD

- ✓ 利用PCA解决FLD的数值稳定性问题
  - 当n<d， $S_W$通常不可逆。
  - 通过PCA先将数据降至k<n维，确保后续的$S_W$可逆

- ✓ Fisherfaces
  - 人脸图像维度高100x100，样本数量可能远小于维度
  - PCA阶段：保留前k=n-C个主成分，确保$S_W$可逆
  - FLD阶段：在低维空间求解
  - 在ORL数据集上，Fisherfaces相比纯PCA（Eigenfaces）分类准确率显著提升。

# 应用：人脸识别

Application: face recognition

# 人脸

- ✓ 为什么人脸数据特别适合PCA和FLD？
- ✓ 用什么分类器？

- ✓ ORL人脸数据集：
  http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
- ✓ OpenCV人脸识别tutorial
  http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html
- ✓ 准备作业：首先需要在windows/linux/mac下安装OpenCV

# 张量Tensor:深度学习的基石

- ✓ 标量(scalar, 纯量)：$x \in \mathbb{R}$
- ✓ 向量(vector): $\boldsymbol{x} \in \mathbb{R}^d$
- ✓ 矩阵(matrix): $X \in \mathbb{R}^d \times \mathbb{R}^d$
- ✓ 进一步推广？
  - 如果$\mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$
  - 称为张量tensor，上例是3阶张量
  - 标量、向量、矩阵分别是0、1、2阶张量
- ✓ 张量的操作，最基本的是向量化vectorize
  - 将矩阵的各行堆积stack起来

# 进一步的阅读

- ✓ 不同条件或要求下的线性特征抽取
  - 如http://cs.nju.edu.cn/wujx/paper/icml2005.pdf
- ✓ 张量和多线性特征抽取
  - Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data
    http://www.crcpress.com/product/isbn/9781439857243
- ✓ 关于特征值和特征向量
  - Golub & van Loan, Matrix Computation, 3rd ed.
    http://www.cs.cornell.edu/courses/cs621/Books/GVL/index.htm