

Multimodal Model Homework 4

2023141460251 孙浩翔

问题 1: MvDA 的主要思想

多视图判别分析 (Multi-view Discriminant Analysis, MvDA) 是一种监督学习下的多模态降维方法, 其核心思想是:

通过为每个模态学习各自的线性投影, 使得同一类别样本在所有视图下的低维表示相互靠近, 不同类别尽可能分开, 同时保持各视图之间的嵌入一致性。

1.1 方法动机

MvDA 的设计动机源于多模态表示学习中的三个基本挑战: 异构模态空间的对齐、判别能力的增强、以及模态间一致性的保证。它融合了 LDA 的类内/类间结构建模能力, 并扩展到多视图情形。

1.2 方法本质

MvDA 本质是一个多投影判别方法。每个模态 v 学得一个线性映射 $\mathbf{W}^{(v)}$, 目标是在共享子空间中实现类别聚合、类别分离和模态对齐。

1.3 应用场景

MvDA 常用于跨模态检索、多模态分类、视频分析等领域, 尤其适合监督信息充足的情形。

问题 2: MvDA 的优化目标

2.1 投影定义

设第 v 个模态样本为 $\mathbf{x}_i^{(v)} \in \mathbb{R}^{d_v}$, 线性映射后:

$$\mathbf{z}_i^{(v)} = \mathbf{W}^{(v)\top} \mathbf{x}_i^{(v)} \in \mathbb{R}^d.$$

2.2 优化目标

MvDA 的目标函数为：

$$J(\{\mathbf{W}^{(v)}\}) = \frac{\text{tr}(\mathbf{S}_B)}{\text{tr}(\mathbf{S}_W + \lambda \mathbf{S}_C)},$$

其中：

- \mathbf{S}_B ：类间散度，衡量不同类别之间的分离度；
- \mathbf{S}_W ：类内散度，衡量同类样本在各模态下的聚合度；
- \mathbf{S}_C ：视图一致性损失，衡量不同模态间的协同表示程度；
- $\lambda > 0$ ：调节一致性对判别性的影响。

2.3 矩阵定义

- 类间散度：

$$\mathbf{S}_B = \sum_{k=1}^C n_k (\bar{\mathbf{z}}_k - \bar{\mathbf{z}})(\bar{\mathbf{z}}_k - \bar{\mathbf{z}})^\top.$$

- 类内散度：

$$\mathbf{S}_W = \sum_{k=1}^C \sum_{i \in C_k} \sum_{v=1}^V (\mathbf{z}_i^{(v)} - \bar{\mathbf{z}}_k)(\mathbf{z}_i^{(v)} - \bar{\mathbf{z}}_k)^\top.$$

- 视图一致性：

$$\mathbf{S}_C = \sum_{i=1}^N \sum_{v < v'} \left(\mathbf{z}_i^{(v)} - \mathbf{z}_i^{(v')} \right) \left(\mathbf{z}_i^{(v)} - \mathbf{z}_i^{(v')} \right)^\top.$$

2.4 解法

最终优化等价于广义特征值问题：

$$\mathbf{S}_B \mathbf{w} = \mu (\mathbf{S}_W + \lambda \mathbf{S}_C) \mathbf{w},$$

对每个模态提取前 d 个特征向量作为投影基。

问题 3：MvDA 与 LDA 的联系与区别

3.1 相似之处

- 两者都基于最大化类间散度、最小化类内散度的准则；
- 都通过广义特征值分解求解；
- 都是线性判别投影方法。

3.2 区别分析

- **数据输入维度不同**：LDA 处理单模态，MvDA 处理多模态；
 - **投影学习形式不同**：LDA 仅学习一个 \mathbf{W} ，MvDA 学多个 $\mathbf{W}^{(v)}$ ；
 - **优化目标不同**：MvDA 引入模态一致性项 \mathbf{S}_C ，增强跨模态对齐；
 - **适用范围不同**：LDA 仅考虑判别性，MvDA 同时兼顾多模态结构。
-

问题 4：MvDA 通过什么消除的视图/模态之间的差异？

4.1 一致性定义

MvDA 通过一致性损失项强制不同模态对同一样本的投影结果相似：

$$\mathbf{S}_C = \sum_{i=1}^N \sum_{v < v'} \left\| \mathbf{W}^{(v)\top} \mathbf{x}_i^{(v)} - \mathbf{W}^{(v')\top} \mathbf{x}_i^{(v')} \right\|^2.$$

4.2 减少模态差异

该项在优化过程中：

- 强制不同模态学习到的子空间在语义上接近；
 - 避免不同模态各自独立训练导致的语义漂移；
 - 增强跨模态检索与统一分类能力。
-

问题 5：MvDA 与 GMA 的区别

5.1 MvDA 特点

- 明确使用类标签（强监督）；
- 目标函数以判别性为主；
- 适合有明确分类任务的场景。

5.2 GMA 特点

- Generalized Multiview Analysis 是一类统一框架；
- 可融合 LDA、CCA、PCA 等；
- 可用于无监督、弱监督与核扩展；
- 更具通用性与灵活性。

5.3 总结

MvDA 是 GMA 框架下一个具代表性的特例，强调判别性与一致性的结合，适合分类任务。

问题 6：视图一致性的定义与效果

6.1 定义

视图一致性要求：对于同一个样本，其不同模态的投影结果应尽量接近：

$$\left\| \mathbf{W}^{(v)\top} \mathbf{x}_i^{(v)} - \mathbf{W}^{(v')\top} \mathbf{x}_i^{(v')} \right\|^2 \rightarrow 0.$$

6.2 效果分析

- 提高不同模态之间的语义对齐程度；
- 降低因模态差异导致的判别性下降；
- 增强多模态嵌入的一致性与可泛化性；
- 对抗模态不对称信息（如图像更丰富，文本稀疏）。

问题 7：MvDA 算法伪代码

```
def mvda(X_views, y, d, lambda_):
    # X_views: list of view matrices [N x Dv]
    # y: labels
    # d: output dimension
    # lambda_: weight for view consistency

    W = [init_proj(X.shape[1], d) for X in X_views]

    while not converged:
        Z = [X @ w for X, w in zip(X_views, W)] # projection

        S_B, S_W = compute_discriminant(Z, y)
        S_C = compute_consistency(Z)

        A = S_B
        B = S_W + lambda_ * S_C
        eigvals, eigvecs = eigh(A, B)
```

```
    for v in range(len(X_views)):
        W[v] = update_proj(X_views[v], eigvecs[:, -d:])

    return W
```