STAT243 Lecture 10.5 Eigendecomposition and SVD

1 Eigendecomposition

Logic

关于 Eigendecomposition 的详细论述, 见 DDA3005 Lecture 15
关于 Power method 的详细论述, 见 DDA3005 Lecture 16
关于 Inverse iteration, Rayleigh quotient iteration, QR iteration 的详细论述, 见 DDA3005 Lecture 17
关于对 QR iteration 的改进的详细论述, 见 DDA3005 Lecture 18

1.1 Eigendecomposition 的原理

- Eigendecomposition（谱分解）在算法收敛分析、PCA 等应用中非常重要
    - 将方差结构分解为一组正交方向（eigenvectors）与对应的变异量（eigenvalues）
- 对实对称矩阵 A：

$$A = \Gamma\Lambda\Gamma^\top$$

    - Γ: 正交矩阵（列为特征向量）
    - Λ: 对角矩阵（特征值，按降序排列）
    - 经典定义：

$$Ax = \lambda x$$

- 逆与平方根：

$$A^{-1} = \Gamma\Lambda^{-1}\Gamma^\top, \quad A^{1/2} = \Gamma\Lambda^{1/2}$$

- Spectral radius:

$$\rho(A) = \max|\lambda_i|$$

  对于对称矩阵, spectral radius 即为 L2 norm:

$$\rho(A) = ||A||_2$$

  会在许多迭代算法的 convergence rate 中出现

1.2 Eigendecomposition 的 Computation

- 常用方法：QR algorithm
    - 第一步：将 A 化为 upper Hessenberg（对称矩阵化为 tridiagonal）
    - 通过 Householder 或 Givens 完成初步化简
    - 之后反复做 QR 分解并 shift，对角元素收敛到 eigenvalues
    - eigenvectors 由所有变换矩阵相乘得到
- 只需最大（或前几个最大）特征值时：使用 power method
    - 通过反复乘以 A 并归一化得到最大特征向量
    - $$z^{(k)} = \frac{Az^{(k-1)}}{|Az^{(k-1)}|}$$
    - λ_1 可通过

$$\lambda_1 \approx \frac{(Az)_i}{z_i}$$

- 找第二大 eigenvalue：
  使用
  $$B = A - \lambda_1 vv^\top$$
  对 B 使用相同方法

1.3 Singular value decomposition

我们考虑一个 $n \times m$ 矩阵 A，假设 $n \geq m$。
矩阵总能分解为

$$A = UDV^\top$$

- U: 左奇异向量，列正交
- V: 右奇异向量，列正交
- D: 对角矩阵，奇异值非负
- 若 A 是方阵且对称正定，SVD 与 eigendecomposition 一致

1.3.1 Representations

- Rank-k 分解：
  $$A = \sum_{j=1}^{k} D_{jj} u_j v_j^\top$$
  每项是 rank-one 外积
- 关系：
  $$A^\top A = VD^2V^\top, \quad AA^\top = UD^2U^\top$$
  因此奇异值的平方等于 eigenvalues
- padding 版本（扩展为方阵）
  $$A = U_{n\times n} D_{n\times m} V_{m\times m}^\top$$

1.3.2 Uses

- 确定矩阵秩
- 求 pseudo-inverse：
  $$A^+ = VD^+U^\top$$
- 最佳 rank-p 近似：Truncated SVD
  $$\tilde{A} = \sum_{j=1}^{p} D_{jj} u_j v_j^\top$$
  根据 Eckart–Minsky–Young 定理，这是 Frobenius norm 下的最优 rank-p 逼近
- 在图像压缩、聚类、推荐系统中广泛使用（如 Netflix Prize）

1.3.3 Interpretation as basis transformation

- 作用 $UDV^\top x$ 可理解为：
  1. $V^\top x$：在 V 的基底中表达 x
  2. D：按奇异值缩放
  3. U：将结果映射到 U 的列空间
- 是两个正交基之间的线性变换

1.4 Computation

1.4.1 Golub–Reinsch algorithm（经典 SVD）

- 通过一系列 Householder 变换将 A 化为 upper bidiagonal 矩阵 $A^{(0)}$
    - 右乘去掉上三角
    - 左乘去掉下三角
- 之后通过一系列 Givens rotations 把 $A^{(j)}$ 迭代逼近对角矩阵 D

  #### $$

  A^{(j+1)} R^\top A^{(j)} T

- U 是所有 pre-multiplication 的积
- V 是所有 post-multiplication 的积
- 最后调整符号（保证奇异值非负）并按降序排序

1.5 Computation for large tall-skinny matrices

- 若 X 是 $n \times p$ 且 $n \gg p$ 可以先做 QR：

$$X = QR$$

- 然后对较小的 $p \times p$ 的 R 做 SVD：

$$R = UDV^\top$$

- 从而

$$X = QUDV^\top = U^*DV^\top$$

- 优点
    - 最耗时的部分变成 tall-skinny QR，可高效并行
    - SVD 部分仅对 $p \times p$ 矩阵，计算量大幅减少
    - 适合大规模数据矩阵（如文本、图像、高维特征）