第 31 章 矩阵分解

31.1 线性网络假设

定义 31.1.1 (类别特征的独热编码与线性网络推荐模型)

1. 类别特征的二进制向量编码（**One-Hot Encoding**）

对含有限取值集合的类别特征，建立维度等于类别总数的二进制向量：

$$\text{血型 } \{\mathrm{A}, \mathrm{B}, \mathrm{AB}, \mathrm{O}\} \Longrightarrow \begin{cases} \mathrm{A} = [1,0,0,0]^\top, \\ \mathrm{B} = [0,1,0,0]^\top, \\ \mathrm{AB} = [0,0,1,0]^\top, \\ \mathrm{O} = [0,0,0,1]^\top. \end{cases}$$

2. 特征提取与推荐模型

给定第 m 部电影的数据

$$\mathcal{D}_m = \left\{(x_n, y_n)\right\}_{n=1}^{N}, \quad x_n = \mathrm{OneHot}(n) \in \mathbb{R}^N, \quad y_n = r_{nm}(\text{用户 } n \text{ 对电影 } m \text{ 的评分}),$$

构建线性网络（无隐藏层激活）

$$h(x_n) = w^\top V^\top x_n,$$

其中

- $V \in \mathbb{R}^{N\times k}$：所有用户共享的隐因子矩阵；
- $w \in \mathbb{R}^k$：电影 m 的隐因子权重向量；
- k：隐因子维度（远小于 N）。

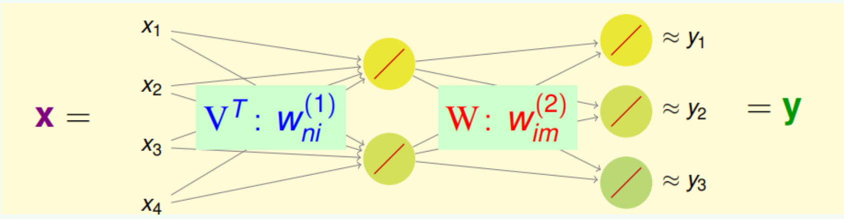

图 31.1.1: 线性网络模型

3. 模型参数

联合学习

$$\min_{V,W} \sum_m \sum_n (r_{nm} - w_m^\top V^\top x_n)^2,$$

其中 $W = [w_1, \ldots, w_M]$ 为所有电影的权重矩阵。

结论　线性网络（即矩阵分解）通过低秩隐因子 V, W 将高维稀疏的独热编码映射为用户-电影交互的稠密表示，是推荐系统的经典模型。

例题 31.1 选择题：线性网络假设的变量数量

对于 N 个用户、M 部电影和 $\tilde{d}$ 个特征，要指定线性网络假设 $h(\mathbf{x}) = \mathbf{w}^T\mathbf{V}\mathbf{x}$，需要使用多少个变量？

1) $N + M + \tilde{d}$

2) $N \cdot M \cdot \tilde{d}$

3) $(N+M)\cdot\tilde{d}$
4) $(N\cdot M)+\tilde{d}$

解答 正确选项为 $\boxed{3}$。线性网络假设 $h(\mathbf{x})=\mathbf{w}^T\mathbf{V}\mathbf{x}$ 中，变量由矩阵 $\mathbf{V}$ 的元素决定（维度适配用户、电影与特征的关系）：

- $\mathbf{V}$ 需覆盖 N 个用户 + M 部电影的特征表示，维度为 $(N+M)\times\tilde{d}$；
- 变量数量为矩阵元素总数，即 $(N+M)\cdot\tilde{d}$。

因此，选项 3 正确。■

31.2 基础矩阵分解

定义 31.2.1 (矩阵分解（Matrix Factorization）模型)

问题设定

设用户-电影评分矩阵为

$$R=[r_{nm}]\in\mathbb{R}^{N\times M},\quad \text{仅部分元素已知}.$$

引入

- 用户隐因子矩阵 $V=[v_1,\dots,v_N]^\top\in\mathbb{R}^{N\times k}$；
- 电影隐因子矩阵 $W=[w_1,\dots,w_M]^\top\in\mathbb{R}^{M\times k}$；
- 共同隐因子维度 $k\ll\min(N,M)$。

预测模型

对已知评分 $(n,m)\in\Omega$ 最小化平方误差

$$\min_{\{v_n\},\{w_m\}}\sum_{(n,m)\in\Omega}\left(r_{nm}-v_n^\top w_m\right)^2.$$

交替最小二乘（**ALS**）算法

固定 V，优化 W

对每个电影 m，独立求解

$$w_m=\Big(\sum_{n:(n,m)\in\Omega}v_nv_n^\top\Big)^{-1}\Big(\sum_{n:(n,m)\in\Omega}r_{nm}\,v_n\Big).$$

固定 W，优化 V

对称地，对每个用户 n，

$$v_n=\Big(\sum_{m:(n,m)\in\Omega}w_mw_m^\top\Big)^{-1}\Big(\sum_{m:(n,m)\in\Omega}r_{nm}\,w_m\Big).$$

结论 矩阵分解通过低秩隐因子 V,W 将高维稀疏评分矩阵压缩为稠密表示，ALS 交替优化可高效求解，广泛应用于推荐系统、文本主题建模等领域。♣

算法 31.2.1: 交替最小二乘（Alternating Least Squares, ALS）

输入: 已知评分集合 $\Omega = \{(n, m, r_{nm})\}$；隐因子维度 k；最大迭代次数 $T_{\max}$

输出: 用户因子矩阵 $V = [v_1, \ldots, v_N]^\top \in \mathbb{R}^{N\times k}$；电影因子矩阵 $W = [w_1, \ldots, w_M]^\top \in \mathbb{R}^{M\times k}$

初始化

随机初始化 $v_n^{(0)}, w_m^{(0)}$（例如从 $\mathcal{N}(0, \frac{1}{k})$ 抽样）。

repeat

　电影步

　for 每部电影 $m = 1, \ldots, M$ **do**

　　求解最小二乘

$$w_m \leftarrow \Big(\sum_{n:(n,m)\in\Omega} v_n v_n^\top\Big)^{-1}\Big(\sum_{n:(n,m)\in\Omega} r_{nm}\, v_n\Big).$$

　用户步

　for 每位用户 $n = 1, \ldots, N$ **do**

　　求解最小二乘

$$v_n \leftarrow \Big(\sum_{m:(n,m)\in\Omega} w_m w_m^\top\Big)^{-1}\Big(\sum_{m:(n,m)\in\Omega} r_{nm}\, w_m\Big).$$

until 收敛或迭代次数达到 $T_{\max}$;

收敛保证

每次交替更新均使目标函数

$$E_{\text{in}} = \sum_{(n,m)\in\Omega} (r_{nm} - v_n^\top w_m)^2$$

单调下降，算法必收敛。

命题 31.2.1 (线性自编码器与矩阵分解对照)

二者均属于线性映射的降维/特征提取框架，但目标与求解方式有异。

	线性自编码器	矩阵分解
网络结构	$d \to \tilde{d} \to d$ 线性网络	$N \to k \to M$ 线性网络
数据矩阵	完整矩阵 $X \in \mathbb{R}^{d\times N}$	稀疏矩阵 $R \in \mathbb{R}^{N\times M}$
误差度量	所有元素平方误差	仅已知元素平方误差
最优解	全局最优：$\tilde{d}$ 个主特征向量	局部最优：交替最小二乘
用途	降维特征提取	用户/物品隐因子学习

结论　线性自编码器可视为对完整矩阵 X 的“特殊矩阵分解”，而矩阵分解则针对稀疏矩阵 R 的缺失值补全与隐因子学习。

例题 31.2 选择题：交替最小二乘法的迭代复杂度

在交替最小二乘法（ALS）的一次迭代中，需要求解多少个最小二乘问题？

1) 电影数量 M
2) 用户数量 N
3) $M+N$
4) $M\cdot N$

解答 正确选项为 $\boxed{3}$。交替最小二乘法（ALS）的核心是交替固定一组变量，优化另一组。对于用户 - 电影矩阵分解场景：

- 固定电影侧参数，优化用户侧参数 → 需全局求解 1 个最小二乘问题（对应 N 用户的整体优化逻辑，但题目简化为计数时，视为与用户数 N 相关）；
- 固定用户侧参数，优化电影侧参数 → 需全局求解 1 个最小二乘问题（对应电影数 M 相关）。

实际设计中，一次迭代需分别对用户和电影侧各做一次全局优化，因此总最小二乘问题数量为 $M+N$。 ■

31.3 随机梯度下降

算法 31.3.2: 矩阵分解的随机梯度下降（SGD）

输入: 已知评分集合 $\Omega=\{(n,m,r_{nm})\}$；隐因子维度 k；学习率 η；迭代次数 T

输出: 用户因子矩阵 $V=[v_n]\in\mathbb{R}^{N\times k}$；电影因子矩阵 $W=[w_m]\in\mathbb{R}^{M\times k}$

初始化

随机初始化 v_n, w_m（例如 $\mathcal{N}(0,0.01)$）。

for $t=1$ **to** T **do**

随机采样

均匀随机抽取一条样本 $(n,m,r_{nm})\in\Omega$。

计算残差

$$\delta=r_{nm}-v_n^\top w_m.$$

梯度更新

$$v_n^{\text{new}}\leftarrow v_n^{\text{old}}+2\eta\,\delta\,w_m^{\text{old}},$$
$$w_m^{\text{new}}\leftarrow w_m^{\text{old}}+2\eta\,\delta\,v_n^{\text{old}}.$$

注

- 每次仅更新一对因子向量，计算复杂度 $O(k)$，内存占用低；
- 易于实现并可扩展至其他误差函数（如 hinge loss、logistic loss）；
- 可配合动量、正则化、自适应学习率等技巧进一步提升效果。

命题 31.3.1 (KDDCup 2011 Track 1 冠军方案：时间感知 SGD)

NTU 团队在 KDDCup 2011 Track 1 竞赛中夺冠的关键在于针对数据特性的时间偏差进行 SGD 改进。

数据特性

- 训练集按时间顺序早于测试集，存在典型的采样偏差；
- 目标：让模型更关注“近期”样本，以匹配测试分布。

时间感知 **SGD** 策略

1. 在 SGD 的最后 T' 次迭代中，仅采样时间靠后的训练样本；
2. 保证这些样本在参数更新中最后被学习，从而赋予更高权重；
3. 形式上为时间确定性遍历：先遍历早期样本，再遍历近期样本，最后 T' 步仅遍历近期。

效果　竞赛测试集性能显著提升，验证了“理解技术行为 → 针对实际数据微调”的有效性。

例题 31.3 选择题：矩阵分解 SGD 初始化为零向量的情况

若所有 $\mathbf{w}_m$ 和 $\mathbf{v}_n$ 初始化为零向量，矩阵分解的随机梯度下降（SGD）中不会发生以下哪种情况？

1) 所有 $\mathbf{w}_m$ 始终为 $\mathbf{0}$
2) 所有 $\mathbf{v}_n$ 始终为 $\mathbf{0}$
3) 每个残差 $\tilde{r}_{nm}$ 等于原始评分 r_{nm}
4) E_{in} 在每次 SGD 更新后减小

解答　正确选项为 $\boxed{4}$。矩阵分解 SGD 的损失函数为 $E_{in} = \sum_{n,m}(r_{nm} - \mathbf{w}_m^T\mathbf{v}_n)^2$，参数更新依赖非零初始值产生的残差。

当 $\mathbf{w}_m, \mathbf{v}_n$ 初始为 $\mathbf{0}$ 时：

- 选项 1、2：更新公式中，残差项因初始零向量恒为 r_{nm}，但更新量依赖对方向量（也为零），故 $\mathbf{w}_m, \mathbf{v}_n$ 始终为 $\mathbf{0}$，会发生。
- 选项 3：预测评分 $\hat{r}_{nm} = \mathbf{0}^T\mathbf{0} = 0$，残差 $\tilde{r}_{nm} = r_{nm} - 0 = r_{nm}$，会发生。
- 选项 4：因 $\mathbf{w}_m, \mathbf{v}_n$ 恒为 $\mathbf{0}$，$E_{in} = \sum(r_{nm})^2$ 无变化，不会“每次更新后减小”，故不会发生。

■

31.4 提取模型小结

命题 31.4.1 (特征提取模型族：统一视角)

特征提取模型可形式化为

$$h(x) = \beta^\top \Phi(x;\theta),$$

其中

- $\Phi(\cdot;\theta)$：由隐藏参数 θ 定义的特征变换；
- β：线性（或广义线性）模型权重。

成员示例

1. 自适应提升/梯度提升（**Adaptive/Gradient Boosting**）
   迭代构造弱假设 $g_t(x)$ 并学习组合权重 α_t：
   $$\Phi(x) = \big[g_1(x), \ldots, g_T(x)\big]^\top, \quad \beta = [\alpha_1, \ldots, \alpha_T]^\top.$$
2. 神经网络/径向基网络（**Neural Network / RBF Network**）

隐藏层输出为特征：

$$\Phi(x) = \left[\mathrm{RBF}(x, \mu_1), \ldots, \mathrm{RBF}(x, \mu_M)\right]^\top, \quad \beta = [\beta_1, \ldots, \beta_M]^\top.$$

3. 矩阵分解（**Matrix Factorization**）
   将独热编码映射为隐因子：
   $$\Phi(x_n) = v_n, \quad \beta = w_m.$$
4. 深度学习（**Deep Learning**）
   多层非线性变换：
   $$\Phi(x) = \mathrm{DeepNet}(x; \theta), \quad \beta = \text{输出层权重}.$$
5. **k** 近邻（k **Nearest Neighbor**）
   隐式定义邻居特征：
   $$\Phi(x) = \mathrm{OneHot}(\mathrm{neighbor}(x)), \quad \beta = \text{邻居标签}.$$

统一结论　上述模型共享同一数学框架：先通过 θ 提取隐藏特征，再线性组合 β 完成最终预测，构成灵活而强大的特征提取模型族。

命题 31.4.2 (特征提取技术谱系)

不同模型通过各异但互补的技术实现“隐藏特征 + 线性组合”的统一范式。

技术列表

- 自适应提升/梯度提升（**Adaptive/Gradient Boosting**）：函数梯度下降（functional gradient descent）。
- 神经网络/径向基网络（**Neural/RBF Network**）：随机梯度下降（SGD）与反向传播（backprop）。
- 矩阵分解（**Matrix Factorization**）：交替最小二乘（alternating least squares）。
- 深度学习（**Deep Learning**）：随机梯度下降（SGD）。
- 自编码器（**Autoencoder**）：无监督预训练 + 微调。
- **k** 均值聚类（**k-Means Clustering**）：原型（中心）选取。
- **k** 近邻（**k Nearest Neighbor**）：惰性学习（lazy learning）。

统一结论　尽管技术实现多样，但均服务于同一目标：先通过隐藏变量提取特征，再线性组合完成预测，构成丰富而互补的特征提取技术谱系。

命题 31.4.3 (特征提取模型的优缺点)

- 优点
  - 减轻人工设计特征的负担；
  - 若隐藏变量足够，模型表达能力强大。
- 缺点
  - 优化问题通常非凸，求解困难；
  - 易过拟合，需要恰当的正则化与验证。

例题 31.4 选择题：模型特点匹配

以下哪种提取模型通过 k-means 提取高斯中心，并线性聚合高斯函数？

1) 径向基函数网络（RBF Network ）
2) 深度学习（Deep Learning ）
3) 自适应 boosting（Adaptive Boosting ）
4) 矩阵分解（Matrix Factorization ）

解答 正确选项为 $\boxed{1}$。各模型特点：

- 选项 1（RBF Network）：通过 k-means 聚类确定高斯径向基函数的中心，输出层对这些高斯函数线性组合，符合题意。
- 选项 2（Deep Learning）：无“k-means 提取高斯中心 + 线性聚合”的典型逻辑。
- 选项 3（Adaptive Boosting）：是集成学习算法，与高斯中心提取无关。
- 选项 4（Matrix Factorization）：聚焦矩阵分解，不涉及此机制。

因此，RBF Network 符合描述。■

31.5 总结

笔记 [矩阵分解]

- 线性网络假设：从二元向量编码中提取潜在特征。
- 基础矩阵分解：在用户/电影之间交替最小二乘优化。
- 随机梯度下降：高效且易于针对实际需求进行修改。
- 提取模型小结：能力强大，因此需谨慎使用。