

## 1. Overfitting 和 underfitting

$\lambda$  小  $\rightarrow$  容易 overfit;  $\lambda$  大  $\rightarrow$  容易 underfit;  $K$  小  $\rightarrow$  模型简单, 欠拟合  $K$  大  $\rightarrow$  容易过拟合

1. $K$  大  $\rightarrow$  低 bias, 高 variance  $\rightarrow$  overfit;  $K$  小  $\rightarrow$  高 bias, 低 variance  $\rightarrow$  underfit

2. $\lambda$  大  $\rightarrow$  variance 低, bias 高;  $\lambda$  小  $\rightarrow$  variance 高, bias 低

overfitting 特征: 训练误差低, 验证误差高 Train RMSE  $\approx 0.8$  Val RMSE  $\approx 1.8$

$\rightarrow$  差距巨大  $\rightarrow$  明显过拟合

原因: 1.模型太复杂 2.参数学得自由

解决方法 (按有效程度排序):

增大  $\lambda$  (最有效的方法)

$\lambda = 0.0001 \rightarrow$  必过拟合

$\lambda = 0.01 \rightarrow$  常用正常值

$\lambda = 0.05, 0.1 \rightarrow$  更强正则化

减小  $K$  (其次有效)

大  $K \rightarrow$  模型容量高  $\rightarrow$  更容易过拟合

若  $K=50 \rightarrow$  改成  $K=10$  或 5

提早停止 (early stopping)

当验证集不改善时停止训练

Underfitting 特征: 训练误差依然高 Train RMSE  $\approx 1.5$  Val RMSE  $\approx 1.6$

$\rightarrow$  模型太弱, 根本没学到有效信息。

原因: 1.模型太简单 2.正则化太强 3. $K$  太小

解决方法:

增大  $K$  (最有效)

$K=3 \rightarrow$  换成  $K=10/20$

为用户/电影提供更丰富的 latent factors 表示能力

减小  $\lambda$  (其次)

$\lambda=0.1 \rightarrow$  改成 0.01

让模型自由一些, 能更好拟合训练数据

增加迭代次数

$iter=10 \rightarrow$  改成 20

(但效果通常不如  $K$  和  $\lambda$  明显)

## 2. Dense 和 Categorical

★ Dense Feature (连续特征 / 数值特征)

数值具有大小关系 / 距离意义, 可以做加减、求均值、标准化。

★ Categorical Feature (类别特征 / 离散特征)

数值只是 类别标签, 没有大小意义, 不能直接做加减或求平均。

✓ Dense (连续) : age/price/rating\_mean/count\_user/time\_diff (例如 last\_login\_difference)/number\_of\_interactions

✓ Categorical (类别) : user\_id (绝对不是 dense) /item\_id/contentType (text/video/article)

/language (en / cn / fr) /region (US / HK / JP) /author\_id

Dense 处理方式: from sklearn.preprocessing import StandardScaler

```
df[dense_cols] = StandardScaler().fit_transform(df[dense_cols])
Categorical 处理方式: from sklearn.preprocessing import LabelEncoder
df[col] = LabelEncoder().fit_transform(df[col])
embedding
```

### 3. Parameter vs Hyperparameter

#### ★ Parameter (模型参数)

模型从训练数据中学习到的数值。模型内部的、会被训练改变的值。

模型参数 (learned):

线性模型的权重 w、偏置 b

神经网络每层的 weights 和 bias

SVD 的 P、Q 矩阵 (latent factors)

决策树叶节点的预测值

特点:

训练过程中不断更新，是模型最终学习到的知识

越多参数 → 容量越大、越容易 overfit

#### ★ Hyperparameter (超参数)

在训练开始前人为设定、不会通过训练自动学习的参数。

超参数例子: 1. 学习率 (learning rate) 2. 正则化系数  $\lambda$  3. SVD 中的 latent dimension K

隐藏层数量: 1. batch size 2. epoch 数 3. dropout rate

特点: 1. 不能从数据中直接学到 2. 必须人为调参 (grid search, random search, bayesian optimization) 3. 影响泛化能力 (overfit / underfit)