

离散特征处理

王树森

ShusenWang@xiaohongshu.com

<http://wangshusen.github.io/>

小红书

离散特征

- 性别：男、女两种类别。
- 国籍：中国、美国、印度等200个国家。
- 英文单词：常见的英文单词有几万个。
- 物品ID：小红书有几亿篇笔记，每篇笔记有一个ID。
- 用户ID：小红书有几亿个用户，每个用户有一个ID。

离散特征处理

1. 建立字典：把类别映射成序号。

- 中国 \rightarrow 1
- 美国 \rightarrow 2
- 印度 \rightarrow 3

2. 向量化：把序号映射成向量。

- One-hot编码：把序号映射成高维稀疏向量。
- Embedding：把序号映射成低维稠密向量。

One-Hot 编码

例1：性别特征

- 性别：男、女两种类别。
- 字典：男 \rightarrow 1，女 \rightarrow 2。
- One-hot编码：用 2 维向量表示性别。
 - 未知 \rightarrow 0 \rightarrow [0, 0]
 - 男 \rightarrow 1 \rightarrow [1, 0]
 - 女 \rightarrow 2 \rightarrow [0, 1]

例2: 国籍特征

- 国籍：中国、美国、印度等 200 种类别。
- 字典：中国 \rightarrow 1，美国 \rightarrow 2，印度 \rightarrow 3， ...
- One-hot编码：用 200 维稀疏向量表示国籍。
 - 未知 \rightarrow 0 \rightarrow $[0, 0, 0, 0, \dots, 0]$
 - 中国 \rightarrow 1 \rightarrow $[\textcolor{red}{1}, 0, 0, 0, \dots, 0]$
 - 美国 \rightarrow 2 \rightarrow $[0, \textcolor{red}{1}, 0, 0, \dots, 0]$
 - 印度 \rightarrow 3 \rightarrow $[0, 0, \textcolor{red}{1}, 0, \dots, 0]$

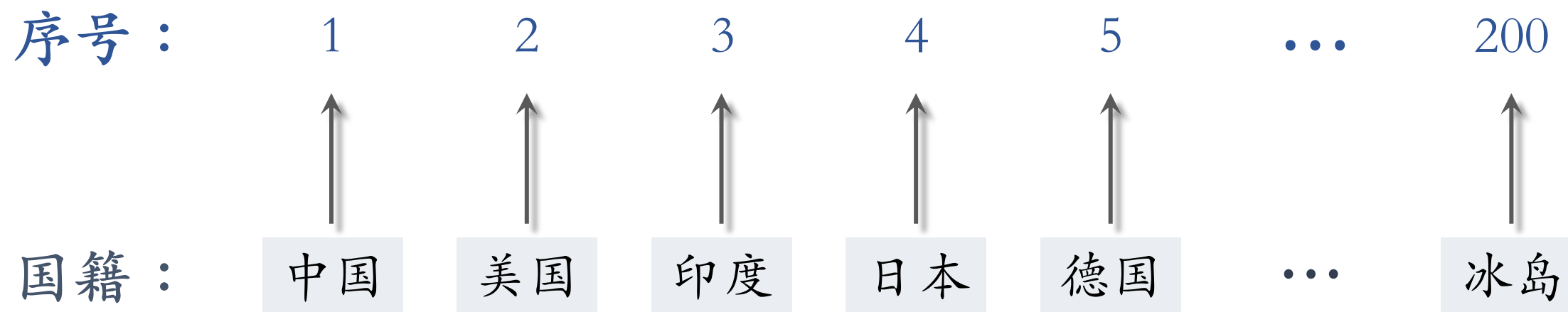
One-Hot编码的局限

- 例1：自然语言处理中，对单词做编码。
 - 英文有几万个常见单词。
 - 那么one-hot向量的维度是几万。
- 例2：推荐系统中，对物品ID做编码。
 - 小红书有几亿篇笔记。
 - 那么one-hot向量的维度是几亿。

类别数量太大时，通常不用 one-hot 编码。

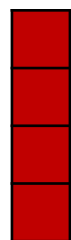
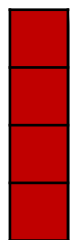
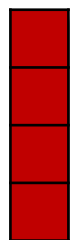
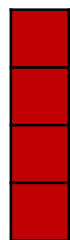
Embedding (嵌入)

例1: 国籍的Embedding



例1: 国籍的Embedding

向量：



...



序号：

1

2

3

4

5

...

200

国籍：

中国

美国

印度

日本

德国

...

冰岛

例1: 国籍的Embedding

- 参数数量：向量维度 \times 类别数量。
 - 设 embedding 得到的向量都是 4 维的。
 - 一共有 200 个国籍。
 - 参数数量 $= 4 \times 200 = 800$ 。

例1： 国籍的Embedding

- **参数数量**：向量维度 \times 类别数量。
 - 设 embedding 得到的向量都是 4 维的。
 - 一共有 200 个国籍。
 - 参数数量 $= 4 \times 200 = 800$ 。
- **编程实现**：TensorFlow、PyTorch 提供 embedding 层。
 - 参数以矩阵的形式保存，矩阵大小是 **向量维度 \times 类别数量**。
 - 输入是序号，比如“美国”的序号是 2。
 - 输出是向量，比如“美国”对应参数矩阵的第 2 列。

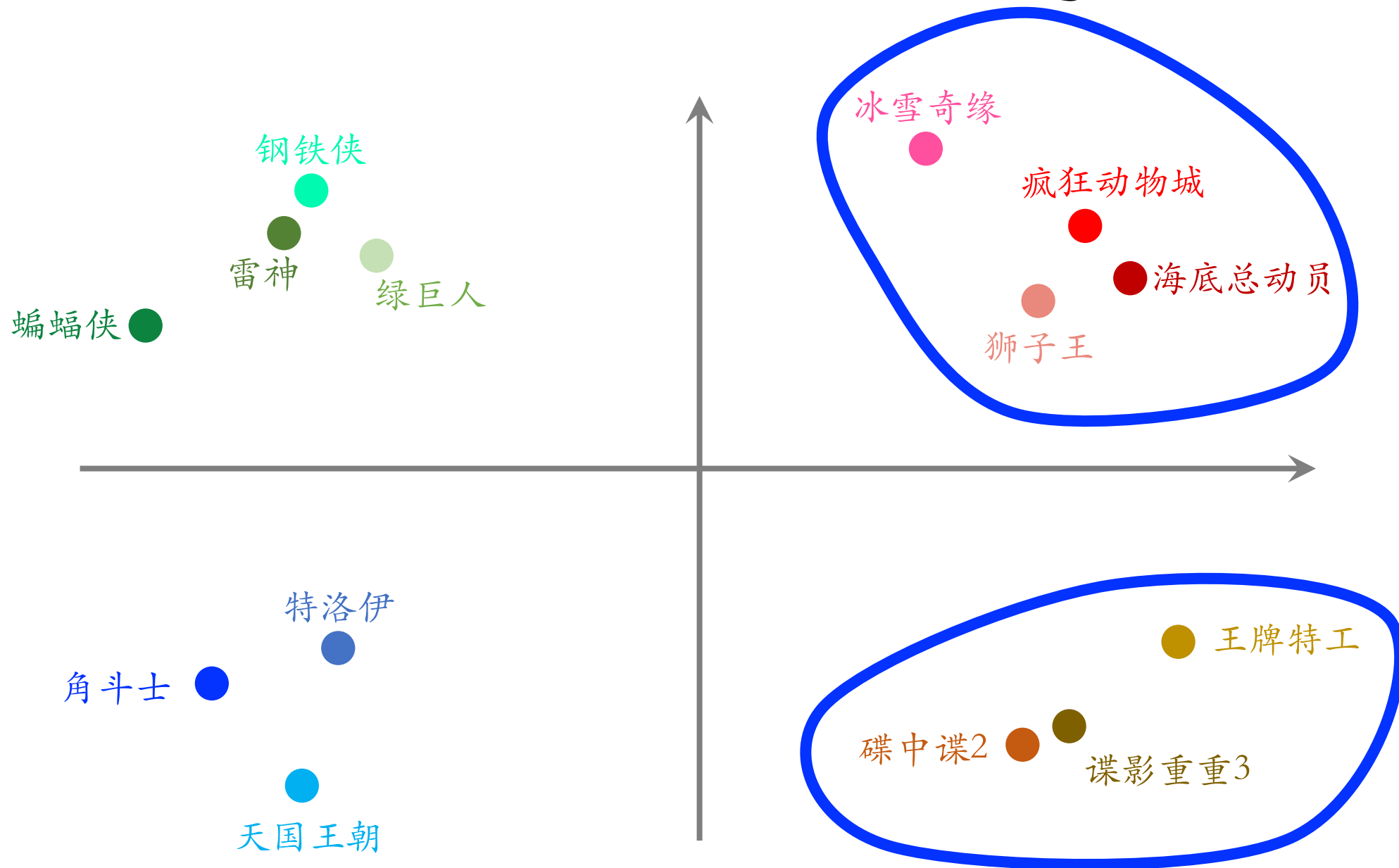
例2: 物品ID的Embedding

- 数据库里一共有 10,000 部电影。
- 任务是给用户推荐电影。
- 设 embedding 向量的维度是 16。

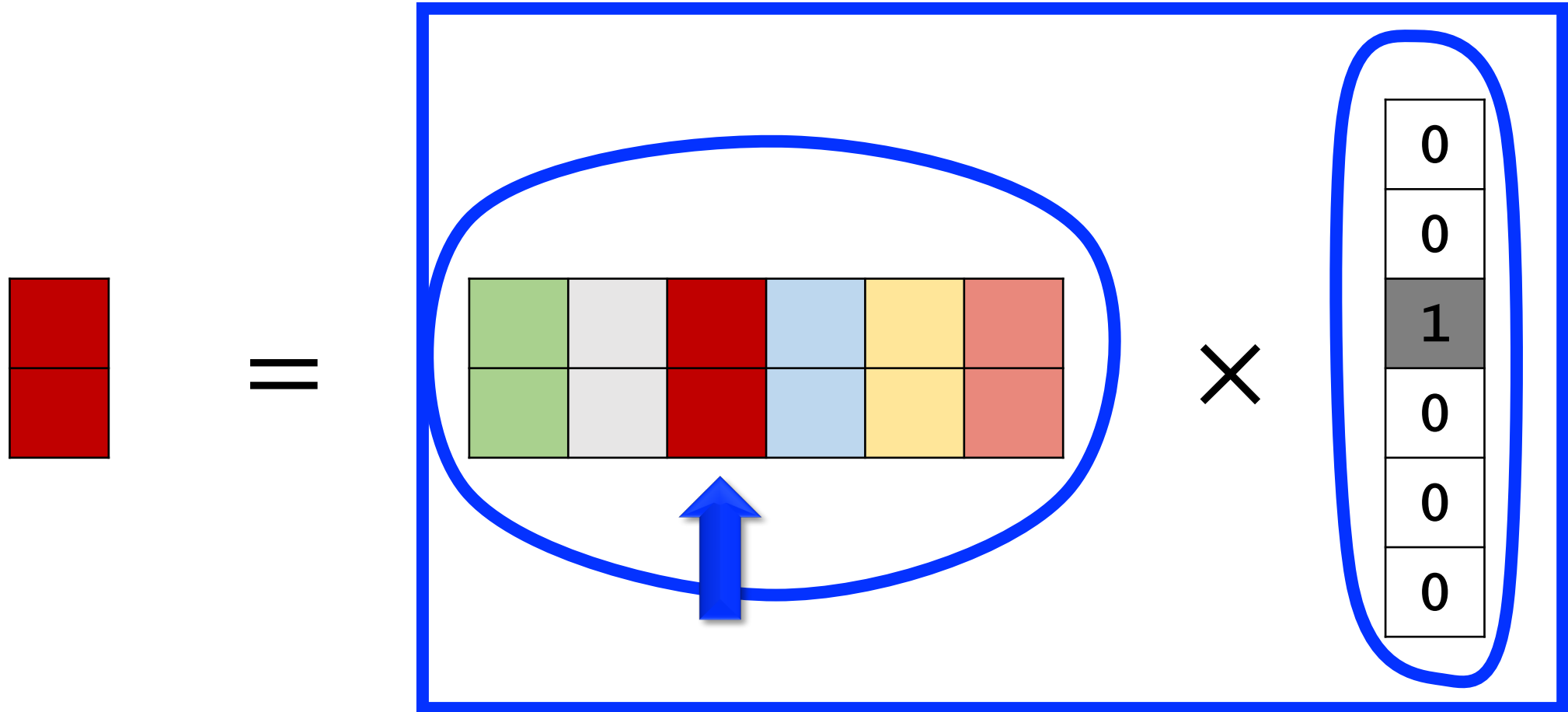
Embedding 层有多少参数？

- 参数数量 = 向量维度 \times 类别数量 = 160,000

例2: 物品ID的Embedding



Embedding = 参数矩阵 \times One-Hot向量



总结

- 离散特征处理：one-hot 编码、embedding。
- 类别数量很大时，用 embedding。
 - Word embedding。
 - 用户 ID embedding。
 - 物品 ID embedding。

Thank You!

<http://wangshusen.github.io/>