SHUANGLN

工作汇报

屠晓涵

2016.11.11

本周工作

基于CNN的语境感知推荐系统——解决数据稀疏性问题

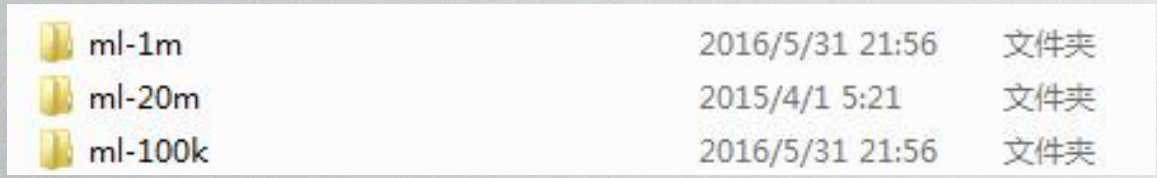
基于CNN的语境感知推荐系统

- ✓ 问题：推荐系统数据稀疏性问题

  数据非常稀疏，使得绝大部分算法（譬如协同过滤）效果都不好。

- ✓ 解决方法：利用文本数据（评论，摘要等）进行文档建模，基于语境感知，将卷积神经网络集成到推荐算法中，捕捉语境信息，提高准确度。

- ✓ 数据集：MovieLens数据集



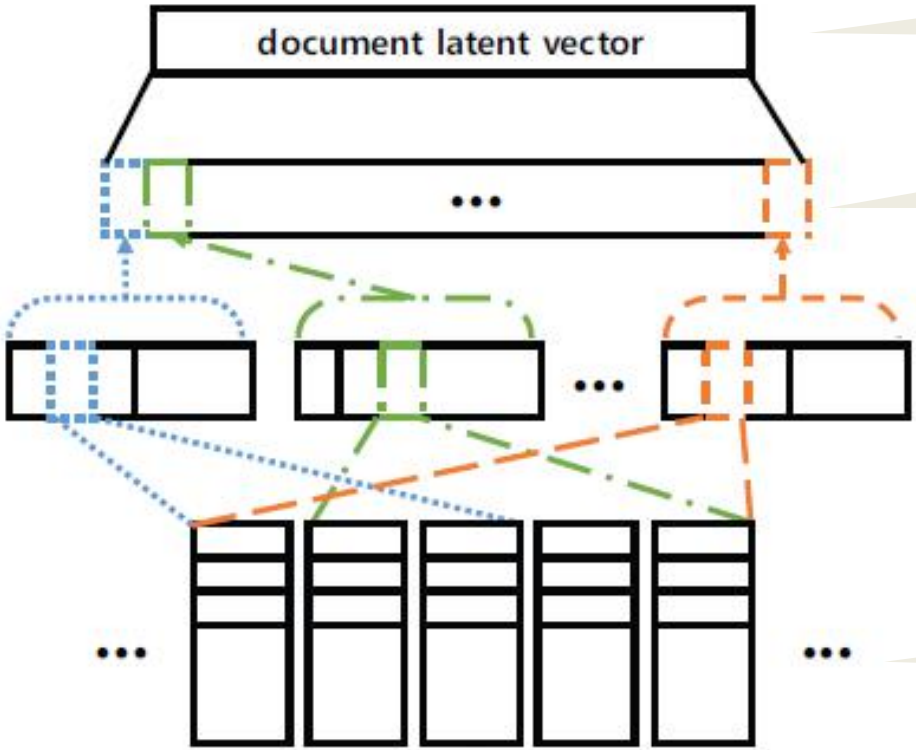
CNN——捕获语境信息

- ✓ 目的：利用物品描述文档，采用CNN，生成项目文档的文本特征向量，服务于推荐
- ✓ 卷积神经网络（Convolutional Neural Network, CNN）：前馈神经网络， 其模型被证明可有效的处理各种自然语言处理问题，如语义分析、搜索结果提取、句子建模 、分类、预测、和其他传统的NLP任务等。

- ✓ 平台：基于Theano和TensorFlow的深度学习框架Keras
- ✓ 选择原因：1，简易和快速的原型设计（keras具有高度模块化，极简，和可扩充特性）
  2，支持CNN和RNN，或二者的结合
  3，支持任意的链接方案（包括多输入和多输出训练）
  4，无缝CPU和GPU切换

CNN——捕获语境信息

✓ CNN语境感知结构如下：



输出层：输出文本特征向量

池化层：提取卷积层有代表性的特征向量

卷积层：提取语境特征

嵌入层：原文本矩阵转化为稠密数值矩阵

评分矩阵分解——进行推荐

利用SVD改进算法进行评分矩阵分解来获得推荐：

1，利用SVD将评分矩阵R分解为U、C、V；

2，将C简化为维数是K的矩阵，得到C_k；

3，相应的简化U、V得到矩阵U_k、V_k；

4，计算两个相关矩阵$U_r=U_k*\sqrt{C_k}$与$V_r=\sqrt{C_k}*V_k^T$;

5，计算用户u对未评分项目i的预测评分：$r_{ui}=U_r(u)V_r(i)$

实际推荐系统有些属性通常与用户、项目无关，因此,目标用户对指定项目的预测评分可表示为基准预测评分加上用户与项目之间相互影响因子，公式变为：

$$b_{ui}=\mu+b_u+b_i$$

$$\hat{r}_{ui}=b_{ui}+\sum_f^F p_{uf}q_{if}$$

$$=\mu+b_u+b_i+\sum_f^F p_{uf}q_{if}$$

评分矩阵分解——进行推荐

由上，损失函数为：

$$F(p,q,b_*)=\sum_{(u,i)\in T}(r_{ui}-\mu-b_u-b_i-\sum_f^F p_{uf}q_{if})^2+\lambda(\|p_u\|^2+\|q_i\|^2+b_u^2+b_i^2)$$

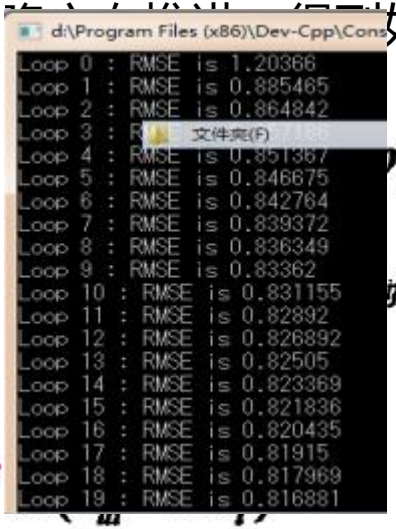
```
        ++recCount;
    }
    float MAE = sqrt(sumErr / recCount);
    return MAE;
}

int main()
{
    SVDPP svp;
    svp.buildModel();
    Evaulator evaul;
    printf("RMSE: %f\n", evaul.RMSEEvaluator(svp));

    return 0;

}
```

如下结果：



推荐系统效果评测

推荐效果评测：

1，离线实验：划分训练集和测试集，在训练集训练用户兴趣模型，在测试集预测

优点：快速方便

缺点：无法用真实的商业指标来衡量

2，用户调查：用抽样的方法找部分用户试验效果

优点：指标比较真实

缺点：规模受限，统计意义不够

3，在线实验：AB测试

优点：指标真实

缺点：测试时间长，设计复杂

下周任务

下周任务： 算法改进

功能模块实现

SHUANGLN

2016

谢谢您的观看