

数据挖掘课程报告

|  |  |
| --- | --- |
| 报告名称 | 基于Embedding的方法的yelp餐厅推荐 |
| 学 院 | 电子与信息工程学院 |
| 专 业 | 计算机科学与技术 |
| 姓 名 |  |
| 学 号 |  |
| 日 期 | 2019年6月19日 |

# 基于embedding方法的yelp餐厅推荐

## 相关工作综述

### 传统推荐算法

推荐系统一直是人工智能研究的热点问题之一。近年来，对个性化推荐系统的研究迅速增加。特别是成立于 2007 年的ACM Recommender Systems (RecSys)会议，现在成为在推荐技术的研究与应用中最重要的年度盛会。在第二十五届国际万维网大会(www)中，用户行为分析以及个性化成为会议研究的热点。此外，在数据库，信息系统和自适应系统这些传统会议中，同样致力于推荐系统的研究。在国内，对于推荐系统的研究正在快速发展中，例如中国计算机协会举办的关于推荐系统的会议，阿里巴巴在电商网站中使用个性化推荐技术等。这些进一步推动了推荐系统的发展。

在传统的推荐研究方面，主要分为四种推荐算法[1]：协同过滤推荐、基于内容的推荐、混合方法推荐和流行度推荐。

协同过滤推荐通过寻找用户一系列行为中特定的模式以给用户特定的推荐。该算法根据类型分为两种。一种为基于邻域的协同过滤（基于用户和基于项目的协同过滤），使用用户对已有项目的评分预测对新项目的评分。一种为基于模型的协同过滤，使用历史数据训练出的预测模型对新项目进行评分。协同过滤算法依赖于用户的数据，包括用户偏好，评级等。它的优势在于对所需数据的字段数量不高，用户和项目的特征没有要求，大多数情况下有良好的效果。然而它的缺点也很明显，比如冷启动，需要标准化数据和要求很高的用户项目比（1:10）[2][3]。

基于内容的过滤通过用户过去喜欢的商品的相似内容比如元数据，主题等推荐给用户。该算法通过信息检索（tf-idf）或者机器学习（朴素贝叶 斯,支持向量机,决策树），计算用户或者商品内容相似度做出推荐。它的优点是没有冷启动问题，不需要惯性的数据；缺点为项目内容需要有意义，缺少多样性，难以整合多个项目特征值[4][5]。

混合推荐是通过加权等方式结合至少两种以上推荐算法 来做推荐，并且减小各自的缺点。对数据源而言，需要同时有用户和项目的特征值和惯性数据。通过加权、切换、混合呈现、特征组合、串联、特征、扩充、元层次混合来实现混合方法的推荐算法。该算法优点为没有冷启动问题，可以推荐罕见的项目，不存在流行度偏见，具有多样性。缺点是需要做大量的工作均衡两种算法[6][7]。

流行度推荐是给用户推荐流行项（现在最多，观看最多，影响最大）的方法。需要通过惯性数据以及项目的内容来评测流行项。它的优点为有助于冷启动问题[8]，易实现；缺点是推荐列表不会改变太大，基本不会推荐新项。

### 基于深度学习的推荐算法

目前，随着深度学习方法在各个领域取得了突破性的成果，基于深度学习的推荐，基于张量分解的上下文感知推荐和基于社交的推荐逐渐发展，在推荐领域占有重要地位。这些算法采用渐进的方式，有利于维持系统最终性能。基于深度学习的推荐是将深度学习应用到传统推荐中。它将基于协同过滤的概率框架作为更复杂的目标，可以从内容，获取的相似性和用户或者商品之间的隐性关联提取出有效的特征表示。在传统推荐系统中使用深度学习可以 进一步提高性能[6][9][10]。基于张量分解的上下文感知推荐可将传统的用户-项目低维评分模型扩展到包含多种上下文信息的多维评分模型，并且将上下文感知推荐融入协同过滤或基于内容过滤的推荐系统中有助于提高推荐的精确度、多样性和鲁棒性等[11][12]。基于社交的推荐综合运用偏好分析、推荐信任分析、社会关系分析模块以及个性化的决策模块，以构建一个更全面的、个性化的电子商务化产品推荐系统。偏好分析模块基于用户产品购买记录来测量用户之间的偏好相似性。推荐信任分析模块根据用户对产品的评价记录来计算客户的产品推荐成功率。社会关系分析模块根据社交网络中的隐式交互记录或显式亲密度来分析用户之间的关系亲密度。个性化产品推荐模块基于用户对不同产品类别的评价计算产品的个性化产品因子权重。社会推荐系统在电商零售方面提供了一个强大的工具，它提高了服务质量，加强客户之间的关系以促进了产品的成功[13][14]。

### 词嵌入

Embedding简单来说就是用一个低维向量表示一个物体等，能够在使用低维向量对物体进行编码的同时还能保留其含义特点，这使得它非常适合深度学习。

在自然语言处理领域，最为著名的Embedding方法之一是词嵌入（Word Embedding）。如果使用one-hot编码词向量，整个向量维度会因为语料库的体积而十分庞大，造成维度灾难。使用Word Embedding方法可以将one-hot编码的向量映射到低维的实数向量。

词嵌入（Word Embedding）算法经常被应用于自然语言处理、句法分析和实体识别等领域，并且得到了越来越广泛的应用。在机器学习中，解决自然语言处理问题的前提是将文本转化为向量化。常用的方法包括 One-Hot 编码、词袋模型（BoW）、词频－逆文档频率（TF-IDF）以及词嵌入方法。其中词嵌入方法中最常用的是Word2Vec，它是由谷歌提出的一个Embedding框架，一种能够将词性以及语义信息影射到为向量的方法。该框架可以在百万数量级的词典和上亿的数据集上进行高效的训练，而且训练得到的词向量可以很好地度量词与词之间的相似性。

Word2Vec 的处理流程大致分为三个阶段，首先会将语句切分，找出文本中句子的边界。第二阶段是分词，识别出具体的词汇。第三阶段为每个词赋予词性与语义信息。Word2Vec 的应用不止于解析自然语言处理，还可以用于基因组，社交媒体点赞等其他符号序列。其原因是这些数据为离散状态，而该算法的目的是获取这些离散状态的转移概率，即数据共同出现的可能性。 Word2Vec 可在向量空间内按词的相似性将词进行分组，该算法生成的向量，以分布式的数值形式来表示词的特征。只要有足够的数据、用法和上下文，Word2Vec就能够对词进行高度准确地预测。这些结果包含了词与词之间的关系，可用于对文档进行主题分类等，在科研、电商等领域具有广泛的应用价值。通过使用余弦相似度能够识别出数学上的相似性，无相似性表示为 90 度角，而相似度为 1 的完全相似则表示为 0 度角，即完全重合。Word2Vec 是根据语料相邻的词来对每个词进行定型训练，而不是像受限玻尔茨曼机（RBM）那样重构输入的词语来定型。该算法有两种模型，一种模型是将词窗口中目标词的上下文作为输入，目标词作为输出，来训练词向量。这种模型使用的方法是连续词袋模型（CBOW）；而另一种模型使用的方法和上述方法相反，将目标词作为输入，上下文作为输出，来训练词向量。这种模型使用的方法是 Skip-Gram 方法。除此之外，Word2Vec 能够将词之间的语义关系映射到训练后的词向量中。因此，通过Word2Vec学到的内容多余人工作业得到的学习成果，而且是以灵活自动的方式进行学习，使得词具有更为连续且更全面的意义。

## 基于Embedding方法构建对象表征

### Word2Vec算法

很多人误以为word2vec是一种深度学习算法，其实它的背后是一个浅层的神经网络，分为两种模型：CBOW模型和Skip-gram模型。

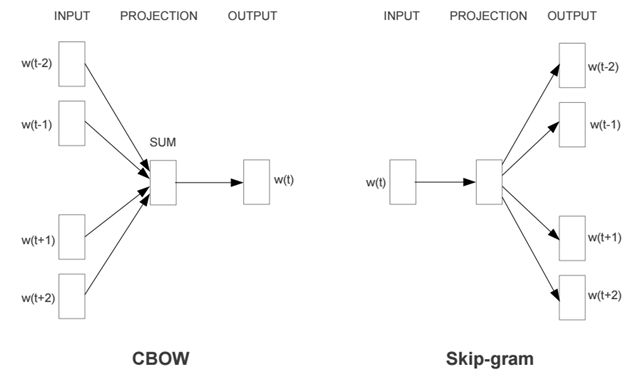


图2.1 CBOW模型（左）和Skip-gram模型（右）

图2.1给出了两种模型的基本框架，可以看出，两种模型的思路其实是刚好相反的。CBOW模型输入的是某一个中心词的上下文相关的词对应的词向量，输出是该中心词的词向量；而Skip-gram模型输入时一个中心词对应的词向量，输出是该中心词上下文相关的词对应的词向量。CBOW模型适合小型数据，而Skip-gram模型在大型语料库中表现更好。

与普通的神经网络相比，word2vec的隐藏层后不跟激活函数，而是在输出层加一个softmax函数，隐藏层的神经元个数即embedding后的向量维度。

使用神经网络实现两种模型的计算量是非常大的，每一次迭代都需要更新整个权重矩阵。为了加速计算，Google提出了两种加速方法：分层softmax和负采样。

1. 分层softmax：如图2.2所示，该方法实际上是采用哈夫曼树代替神经网络中隐藏层到输出层的映射，这样可以避免计算所有词的softmax概率，其本质是把N分类问题变成log(N)次二分类问题。具体每一次二分类问题采用二元逻辑回归的方法，判别正负类别使用sigmoid函数，目标是找到适合所有节点的词向量和内部节点使得训练样本达到最大似然。由于哈夫曼树本身的特点，高频词所在的层非常浅而生僻词所在的层非常深，这就导致它对生僻词的效率比较低，因此适合小样本。

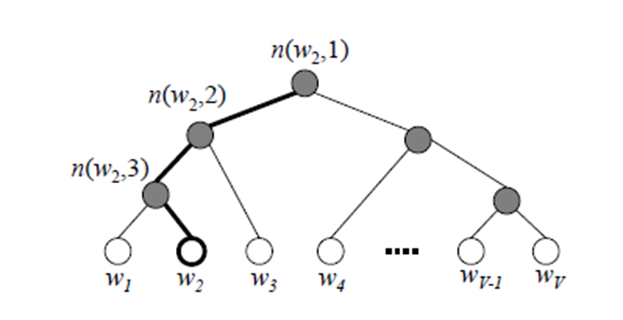


图2.2 分层softmax

2）负采样方法：对于一个中心词为的训练样本，上下文为 ，这是个真实的正例。通过负采样，得到 个和 不同的中心词 ，这样 和 组成了 个并不真实存在的负例，进行二元逻辑回归，得到负采样对应的每个词对应的模型参数 和每个词的词向量。负采样方法更适合大型语料库的训练。

### 融合文本信息的词向量生成

对于餐厅的推荐，应该考虑用户的历史行为以及对于新餐厅的覆盖度。由于用户对某一口味或者餐厅环境等因素的偏好，极有可能是一家餐厅的常客，因此考虑个性化推荐时，应该包含就餐历史记录信息。同时，为了提高对于新鲜感的覆盖度，考虑餐厅之间类型的相似度，为用户推荐新的近似类型的餐厅也是十分必要的。

### 用户就餐记录提取

Yelp数据集中没有直接的用户就餐历史信息。为了获得用户的最近就餐历史，我们从tip文件中的评论记录提取。首先根据时间的先后顺序将用户评论进行排序，选取最近就餐历史记录前N个，得到用户最近就餐历史列表（由餐厅id组成）。餐厅id列表可以作为Embedding模型的输入，以获得其向量表示，通过对其进行Embedding，计算用户就餐历史记录与待推荐餐厅列表的相似度，我们得到对于短期就餐偏好的反映。

### 餐厅类型文本构建

用户就餐历史Embedding得到的是用户的短期偏好。目的是展示用户曾去过的餐厅与推荐餐厅列表的相似关系。对于一个用户，如果其就餐位置发生了改变，对于个性推荐更重要的是餐厅类型的相似度。餐厅类型可以反映不同餐厅之间的相似性，这是一种长期就餐类型选择偏好，并且通过匹配相似类型的餐厅，可以给用户推荐新的就餐地点，提高对新鲜感的覆盖度。给定某一餐厅的元数据，例如：评论数量、是否设置露天位置，是否设置免费wifi等信息，我们根据数据特征的分布情况制定映射规则，将餐厅信息转化为可进行Embedding的句子。

具体规则如表2-1所示：

表 2- 1 business type 映射规则

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
| stars | <1 | 1-2 | 2-3 | 3-4 | 4-5 |
| review\_count | 0-12 | 13-35 | 36-85 | 86-246 | 247-8350 |
| OutdoorSeating | False | True |  |  |  |
| BusinessAcceptsCreditCards | False | True |  |  |  |
| RestaurantsReservations | False | True |  |  |  |
| WiFi | NO | Yes |  |  |  |
| Alcohol | Full\_Bar | Beer&Wine | NO |  |  |

### 用户类型文本构建

考虑到个性化餐厅推荐不应该只利用用户的就餐历史记录，其口味偏好，个人特点以及对餐厅的评价标准等对最终推荐结果也有极大贡献。我们根据用户数据每一项特征的数据分布情况，制定了一定的规则，用来生成user\_type，生成规则如表2-2所示：

表 2- 2 user type 映射规则

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 |
| average\_stars | <1 | 1-2 | 2-3 | 3-4 | 4-5 |
| years | 1-6 | 7-8 | 9 | 10 | 11-14 |
| review | 9-27 | 28-58 | 59-118 | 119-260 | 261-10022 |
| useful | 0-28 | 29-72 | 73-183 | 184-619 | 620-74829 |
| cool | 0-9 | 10-26 | 27-76 | 77-333 | 334-63148 |
| funny | 0-7 | 8-21 | 22-60 | 61-243 | 244-49785 |
| friends | 0-574 | 575-1750 | 1751-3934 | 3935-8907 | 8908-229534 |

### Embedding

由于个性化推荐是一个动态问题，用户口味会随着时间或者个人生活习惯等因素的变化而发生变化。为了考虑到用户不断变化的偏好，我们将business\_type, user\_type和historical\_ids在相同的向量空间中学习嵌入。通过对用户类型、餐厅类型、用户就餐历史的Embedding，我们得到了可以用于推荐中计算对象相似度的嵌入向量。

## 实验与分析

### 3.1 概况

Yelp是美国最大的点评网站，囊括了各地餐馆、购物中心、酒店、旅游等领域的商户。用户可以在Yelp网站中给商户打分，提交评论，交流购物体验等。在Yelp搜索一个餐厅或者旅馆等，就能看到它的简要介绍与网友的点评，包含给的星级评分以及评论等。Yelp与国内的大众点评是同一类型的产品。

Yelp在2017年公开了其内部数据集。据官网介绍，这是一个通用的数据集，开放的主要目的是帮助学习。该数据集是Yelp涵盖的商户、点评和用户数据的一个子集，主要可以用于学习NLP、推荐等。

### 3.2 数据集组成

Yelp数据集有两种格式：JSON和SQL，包含470万条用户评价，15万条商户信息，20万张图片，12座城市。此外，数据集还涵盖110万用户的100万条tips，超过120万条商家属性，随着时间推移在每家商户签到的总用户数等。

从Yelp网站下载的数据集包含6个JSON文件：business.json（商户信息），checkin.json（用户签到信息），photo.json（图片信息），review.json（用户评论），tip.json（tips）以及user.json（用户信息）。对于餐厅推荐而言，我们需要的是餐厅信息、用户对餐厅的点评等。选择Yelp数据集的原因是它的数据量大、数据丰富，有很多东西可以挖掘。

### 3.3 数据清洗

鉴于简化实验的需要，我们只选择一个城市作为实验的地区。在众多城市中，拉斯维加斯拥有着最多数量的餐厅，也是一个旅游大城，因此我们最后选择了拉斯维加斯的餐厅用来做推荐。

总体来看，餐厅的推荐需要餐厅的一些基本信息，包括菜系、环境、平均评分等；还需要用户的信息，包括口味偏好、需求的餐厅类型、质量等，这些一部分可以从历史记录中得到。在本实验中，我们认为历史记录即用户评论过的餐厅。

数据清洗大致分为以下几个步骤：

1. 从原始数据中提取拉斯维加斯的餐厅数据；
2. 提取在拉斯维加斯用餐过的用户数据（即review和tips中的餐厅位于拉斯维加斯的）；
3. 去除冗余的数据列和无用的数据列；
4. 处理缺失值，规范化列的属性值（如餐厅WiFi的情况统一为No, Free和Paid三种）；
5. 将时间戳标准化为日期；
6. 进一步筛选餐厅和用户，为了提高推荐效果，剔除了review数目较少的用户，这样也能同时减少数据量。

数据清洗完成后，总计6450家餐厅以及7610个用户。

### 3.4 餐厅推荐

对于个性化推荐部分，我们首先根据Section 2 部分描述的算法，获得用户类型、用户就餐历史餐厅id列表以及餐厅类型的Embedding结果，如下图3-1所示。进行个性化推荐时，给定一个用户，计算嵌入向量之间的相似性，将前N个最相似的嵌入向量作为推荐餐厅。

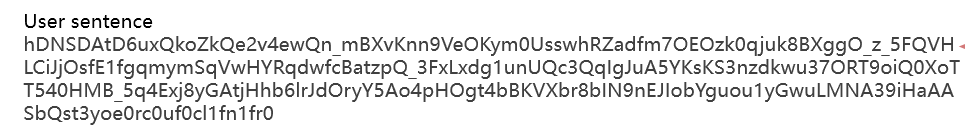


图 3- 1 嵌入结果

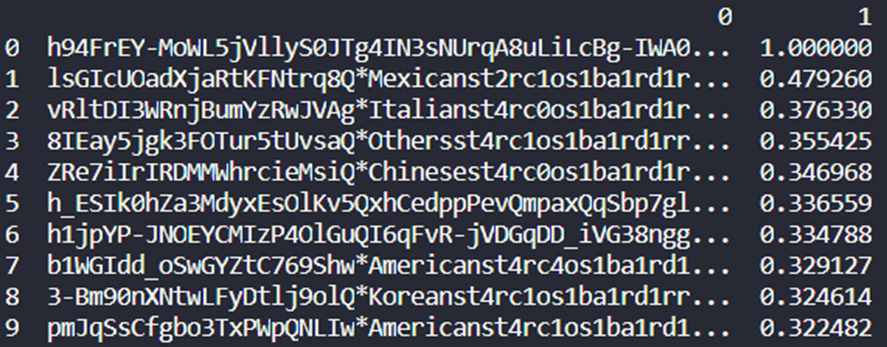


图 3- 2 给定一个用户的推荐结果

如图3-2所示，给定一个用户，计算得到前topN个最相似的餐厅进行推荐。由结果可以看出，排在第一位的为用户本身的嵌入向量。依次排在二、三位的是推荐餐厅，推荐餐厅的id为结果星号之前的字符串。

我们可以看到，推荐排名第一位的结果与用户本身之间只有47.93%的相似性，而第二位的结果只有37.63%的相似性，这些相似性数值并不高。原因是，我们采用的计算相似性的方法不是基于相同对象的，这是用户和餐厅之间的相似性匹配分数，会比计算相同对象之间的相似性分数低。而且我们所能够获得的用户个性化信息并不多，对于最终结果的贡献作用不会特别明显。但是这不能说明推荐的效果不好，因为我们综合考虑了用户的就餐历史、用户类型特征以及餐厅类型特征。

论文《Real-time Personalization using Embeddings for Search Ranking at Airbnb》为基于嵌入的个性化推荐提供了指导思路，通过对于实体的嵌入，可以获得具有一定信息量的原始实体的向量表示，对于进一步使用机器学习或是深度学习方法提供了便利。

## 参考文献

[1] Zhang F , Yuan N J , Lian D , et al. Collaborative Knowledge Base Embedding for Recommender Systems[C]// the 22nd ACM SIGKDD International Conference. ACM, 2016.

[2] Barkan O, Koenigstein N. Item2vec: neural item embedding for collaborative filtering[C]//2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2016: 1-6.

[3] Schafer J B, Frankowski D, Herlocker J, et al. Collaborative filtering recommender systems[M]//The adaptive web. Springer, Berlin, Heidelberg, 2007: 291-324.

[4] Florez O U. Deep learning of semantic word representations to implement a content-based recommender for the RecSys challenge’14[C]//Semantic Web Evaluation Challenge. Springer, Cham, 2014: 199-204.

[5] Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles[C]//Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 448-456.

[6] Wang X, Wang Y. Improving content-based and hybrid music recommendation using deep learning[C]//Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014: 627-636.

[7] Kouki P, Fakhraei S, Foulds J, et al. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems[C]//Proceedings of the 9th ACM Conference on Recommender Systems. ACM, 2015: 99-106.

[8] Aharon M, Anava O, Avigdor-Elgrabli N, et al. Excuseme: Asking users to help in item cold-start recommendations[C]//Proceedings of the 9th ACM Conference on Recommender Systems. ACM, 2015: 83-90.

[9] Wang H, Wang N, Yeung D Y. Collaborative deep learning for recommender systems[C]//Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2015: 1235-1244.

[10] Bowman S R. Can recursive neural tensor networks learn logical reasoning?[J]. arXiv preprint arXiv:1312.6192, 2013.

[11] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization[C]//Advances in neural information processing systems. 2014: 2177-2185.

[12] Chen W, Lin T, Tan Z, et al. Robust influence maximization[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016: 795-804.

[13] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014: 701-710.

[14] Tang J, Aggarwal C, Liu H. Recommendations in signed social networks[C]//Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2016: 31-40.