中国人民大學
RENMIN UNIVERSITY OF CHINA

如何以初学者角度写好一篇国际（顶级）学术论文

汇 报 人：赵鑫
单　　位：中国人民大学

本次报告的定位与声明

- 所使用例子均来自信息检索、数据挖掘、自然语言处理领域
- 所使用的例子为讲者之前的论文，全部为个人观点，只是个人的经验分享
- 介绍基本的论文写作思路以及规范
- 强调避免一些基本的写作错误
- “提速”的一些方法
- 推荐阅读更好的报告
  - 清华大学刘洋老师的《机器翻译学术论文文写作方方法和技巧》
    - http://nlp.csai.tsinghua.edu.cn/%7Ely/talks/cwmt14_tut.pdf

示例论文简介

- 例子1

Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation

Jingyuan Wang*, Ning Wu*
Beijing Advanced Innovation Center for BDBC, School of Computer Science and Engineering, Beihang University, Beijing, China
{jywang,wuning}@buaa.edu.cn

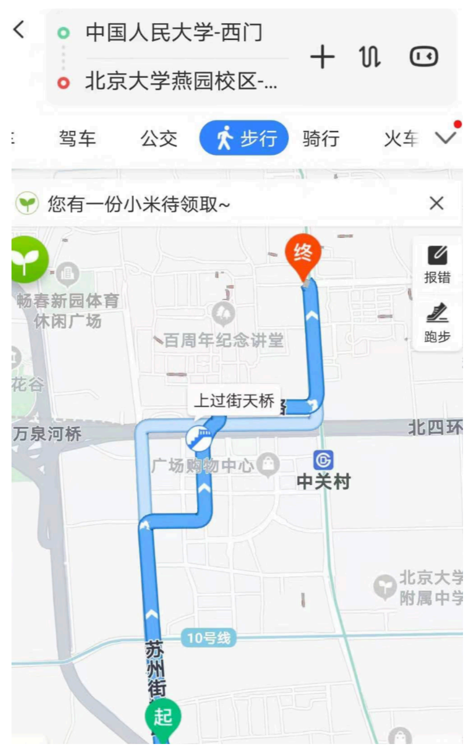
Wayne Xin Zhao†
School of Information, Renmin University of China
Beijing, China
batmanfly@gmail.com

Fanzhang Peng, Xin Lin
MOE Engineering Research Center of ACAT, School of Computer Science and Engineering, Beihang University
Beijing, China
{pengfanzhang,sweeneylin}@buaa.edu.cn

[SIGKDD'19] Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation.

示例论文简介

- 基于背景信息的智能化路径搜索算法



任务：给定路网结构，生成一个用户从A点到B点的个性化路线规划
简化为生成一个路段ID的序列预测任务

[SIGKDD'19] Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation.

示例论文简介

- 基于背景信息的智能化路径搜索算法

Review of A^* Algorithm. In the literature [10], A^* search algorithm is widely used in pathfinding and graph traversal due to its performance and accuracy. Starting from a source node of a graph, it aims to find a path to the given destination node resulting in the smallest *cost*. It maintains a tree of paths originating at the source node and extending those paths one edge at a time until its termination criterion is satisfied. At each extension, A^* evaluates a candidate node n based on a *cost function* $f(n)$

$$f(n) = g(n) + h(n), \tag{2}$$

where $g(n)$ is the cost of the path from the source to n (we call it *observable cost* since the path is observable), and $h(n)$ is an estimate of the cost required to extend the future path to the goal (we call it *estimated cost* since the actual optimal path is unknown). The key part of A^* is the setting of the heuristic function $h(\cdot)$, which has an important impact on the final performance.

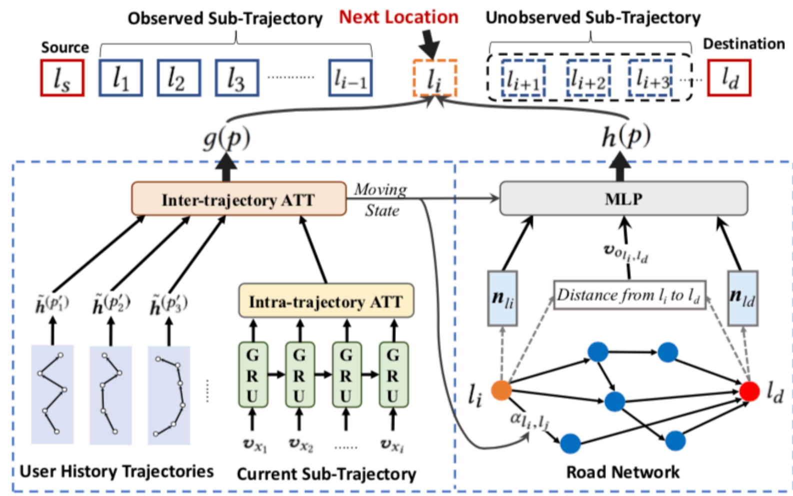


Figure 1: The overall architecture of the NASR model. $g(\cdot)$ learns the cost from the source to a candidate location, called *observable cost*; $h(\cdot)$ predicts the estimated cost from a candidate location to the destination, called *estimated cost*.

A*算法：分解启发式搜索值为两个部分
g计算历史消耗，h预估未来消耗

[SIGKDD'19] Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation.

示例论文简介

- 基于背景信息的智能化路径搜索算法

Review of A^* Algorithm. In the literature [10], A^* search algorithm is widely used in pathfinding and graph traversal due to its performance and accuracy. Starting from a source node of a graph, it aims to find a path to the given destination node resulting in the smallest *cost*. It maintains a tree of paths originating at the source node and extending those paths one edge at a time until its termination criterion is satisfied. At each extension, A^* evaluates a candidate node n based on a *cost function* $f(n)$

$$f(n) = g(n) + h(n), \tag{2}$$

where $g(n)$ is the cost of the path from the source to n (we call it *observable cost* since the path is observable), and $h(n)$ is an estimate of the cost required to extend the future path to the goal (we call it *estimated cost* since the actual optimal path is unknown). The key part of A^* is the setting of the heuristic function $h(\cdot)$, which has an important impact on the final performance.

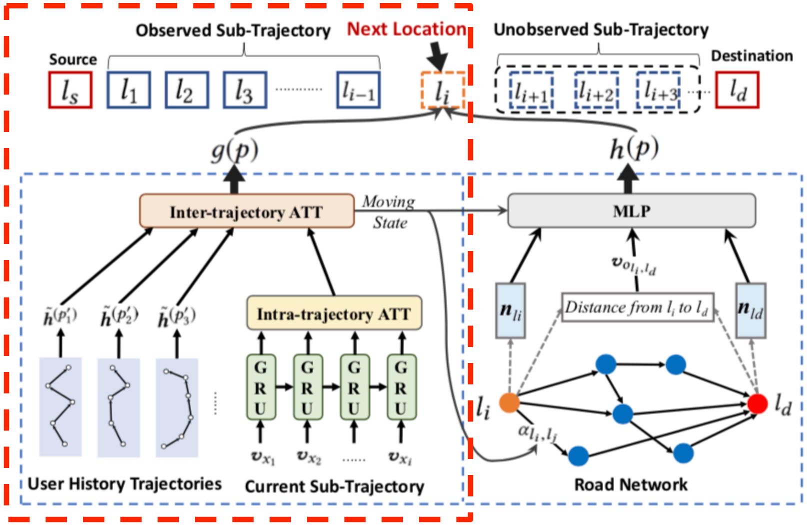


Figure 1: The overall architecture of the NASR model. $g(\cdot)$ **learns the cost from the source to a candidate location, called** ***observable cost***; $h(\cdot)$ **predicts the estimated cost from a candidate location to the destination, called** ***estimated cost.***

g函数实现：使用基于注意力机制的GRU网络刻画用户运动状态，
路径生成概率作为g函数的数值

[SIGKDD'19] Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation.

示例论文简介

- 基于背景信息的智能化路径搜索算法

Review of A^* Algorithm. In the literature [10], A^* search algorithm is widely used in pathfinding and graph traversal due to its performance and accuracy. Starting from a source node of a graph, it aims to find a path to the given destination node resulting in the smallest *cost*. It maintains a tree of paths originating at the source node and extending those paths one edge at a time until its termination criterion is satisfied. At each extension, A^* evaluates a candidate node n based on a *cost function* $f(n)$

$$f(n) = g(n) + h(n), \tag{2}$$

where $g(n)$ is the cost of the path from the source to n (we call it *observable cost* since the path is observable), and $h(n)$ is an estimate of the cost required to extend the future path to the goal (we call it *estimated cost* since the actual optimal path is unknown). The key part of A^* is the setting of the heuristic function $h(\cdot)$, which has an important impact on the final performance.

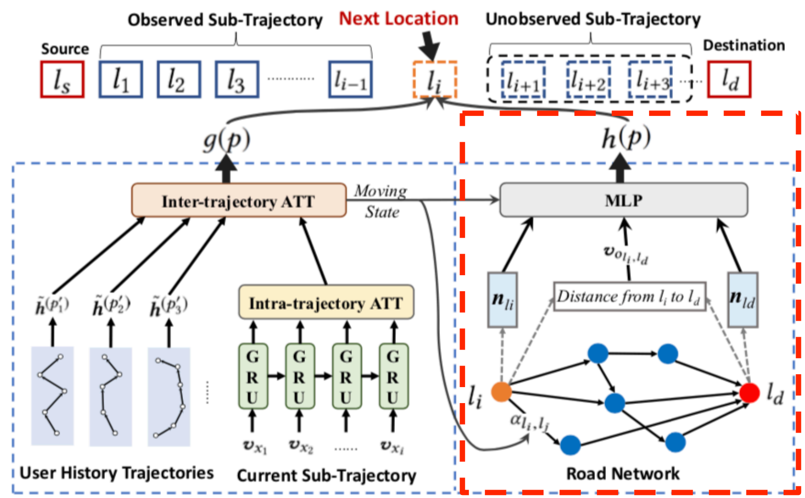


Figure 1: The overall architecture of the NASR model. $g(\cdot)$ learns the cost from the source to a candidate location, called *observable cost*; $h(\cdot)$ predicts the estimated cost from a candidate location to the destination, called *estimated cost*.

h函数实现：使用图神经网络基于路网学习节点表示

搭建估值网络，使用强化学习算法进行训练

[SIGKDD'19] Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation.

示例论文简介

- 例子2

Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks

Jin Huang
School of Information, Renmin University of China
jin.huang@ruc.edu.cn

Wayne Xin Zhao*
School of Information, Renmin University of China
batmanfly@gmail.com

Hongjian Dou
School of Information, Renmin University of China
hongjiandou@ruc.edu.cn

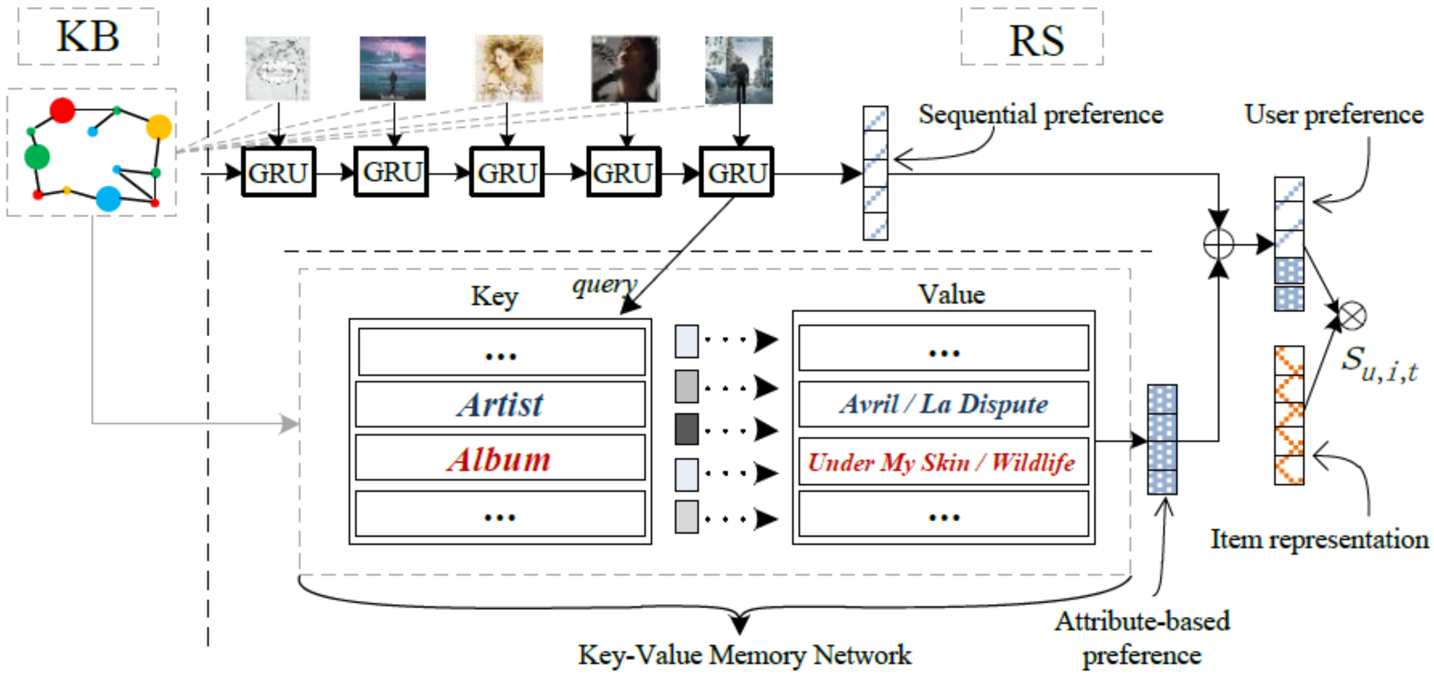
Ji-Rong Wen
Beijing Key Laboratory of Big Data Management and Analysis Methods
jrwen@ruc.edu.cn

Edward Y. Chang
Research & Innovation, HTC
eyuchang@gmail.com

[SIGIR'18] Improving Sequential Recommendation with Knowledge-enhanced Memory Networks.

示例论文简介

- 基于知识属性的序列化推荐算法



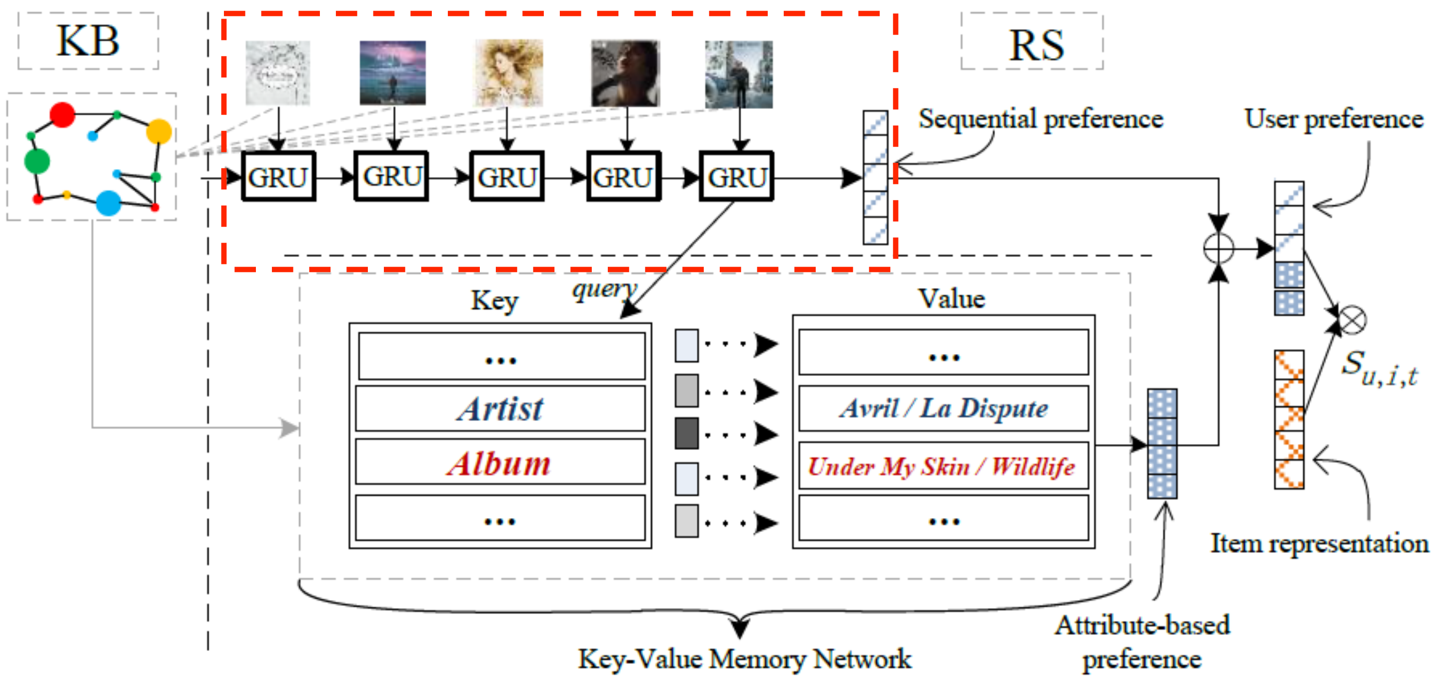
基本思路：使用记忆网络扩展数据存储

知识存于记忆网络，隐状态做结构化解码

[SIGIR'18] Improving Sequential Recommendation with Knowledge-enhanced Memory Networks.

示例论文简介

- 基于知识属性的序列化推荐算法

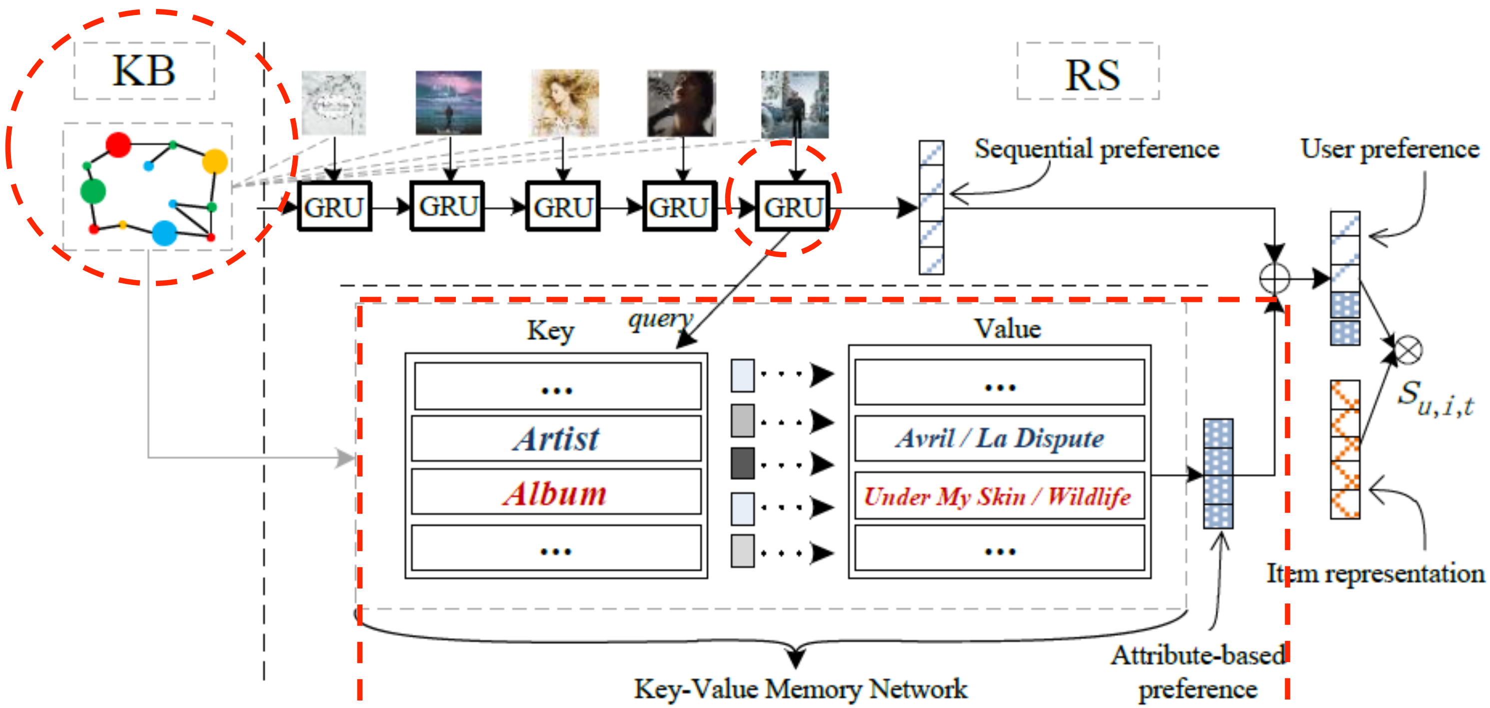


底层模型：传统的GRU序列推荐算法
推荐任务抽象为一个预测序列ID的任务

[SIGIR'18] Improving Sequential Recommendation with Knowledge-enhanced Memory Networks.

示例论文简介

- 基于知识属性的序列化推荐算法



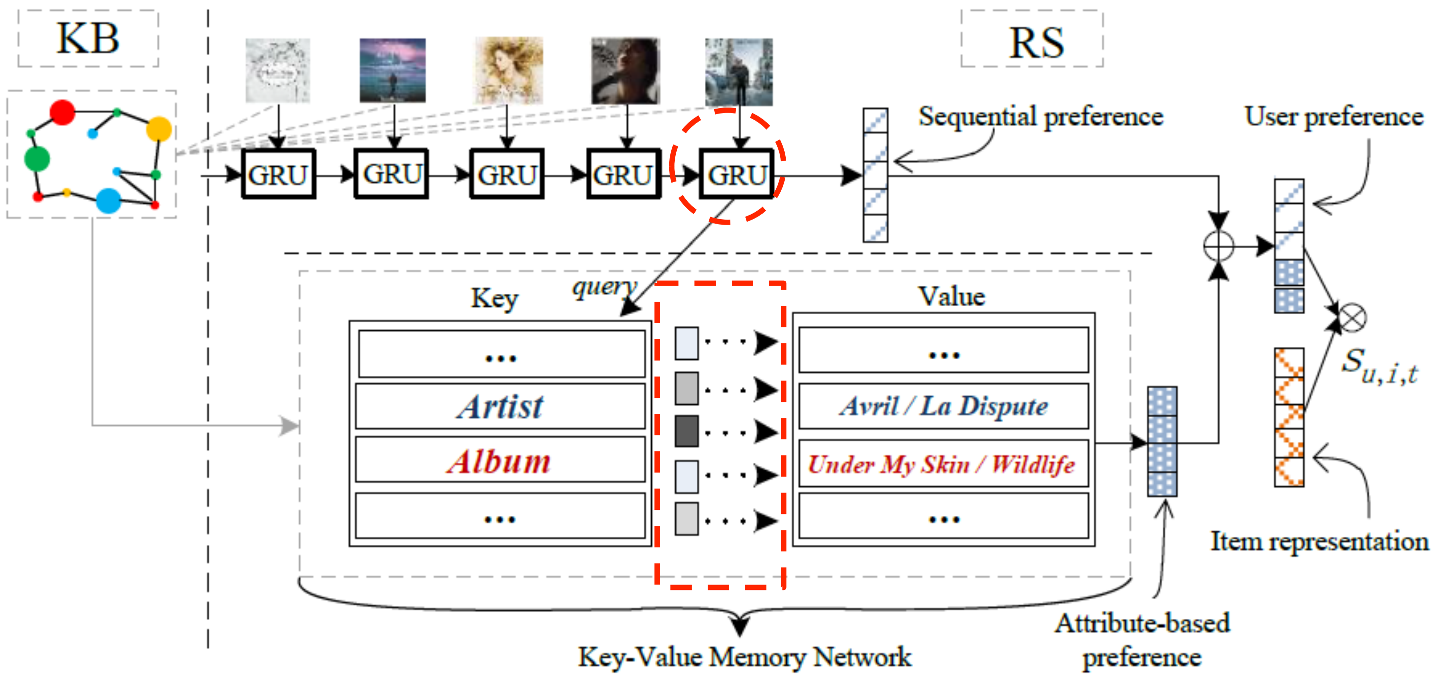
状态解码：传入键-值记忆网络

键是属性的表示、值是用户对应属性的偏好表示

[SIGIR'18] Improving Sequential Recommendation with Knowledge-enhanced Memory Networks.

示例论文简介

- 基于知识属性的序列化推荐算法

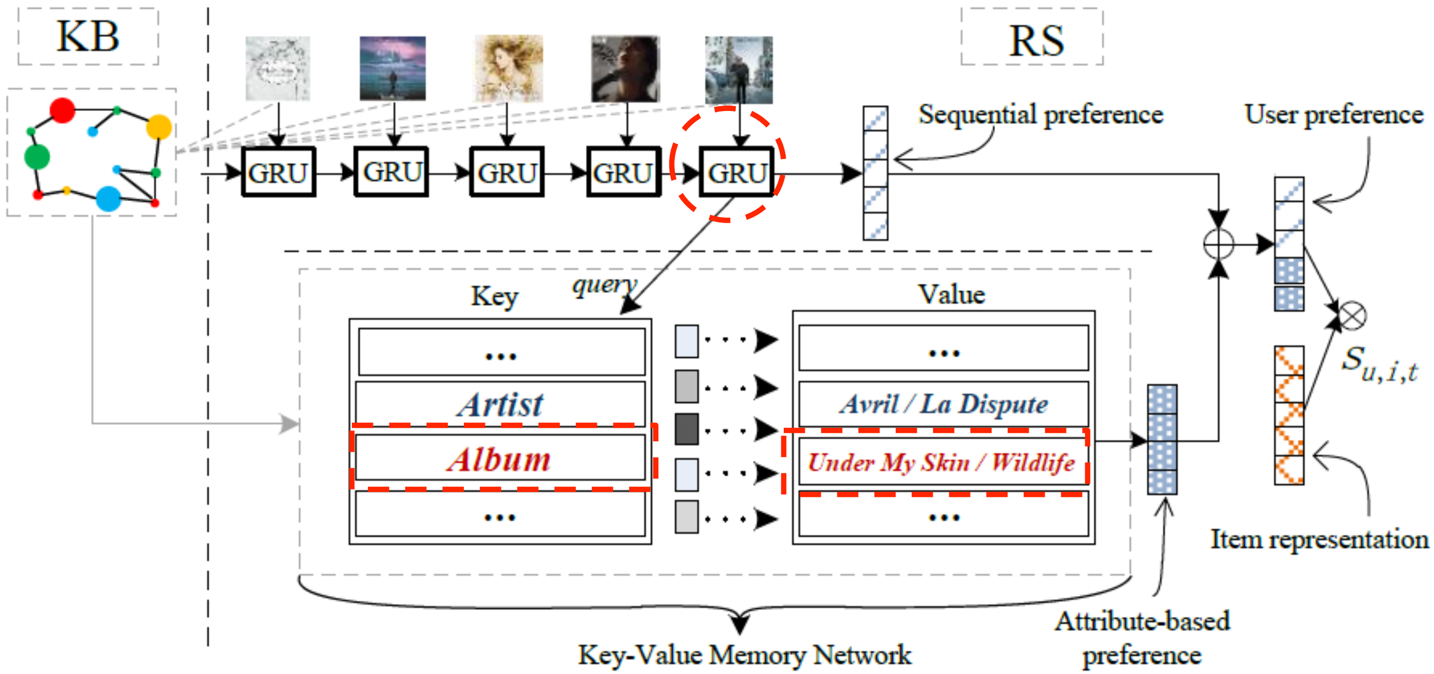


状态解码：获得细粒度属性上的用户偏好
灰度代表权重大小

[SIGIR'18] Improving Sequential Recommendation with Knowledge-enhanced Memory Networks.

示例论文简介

- 基于知识属性的序列化推荐算法



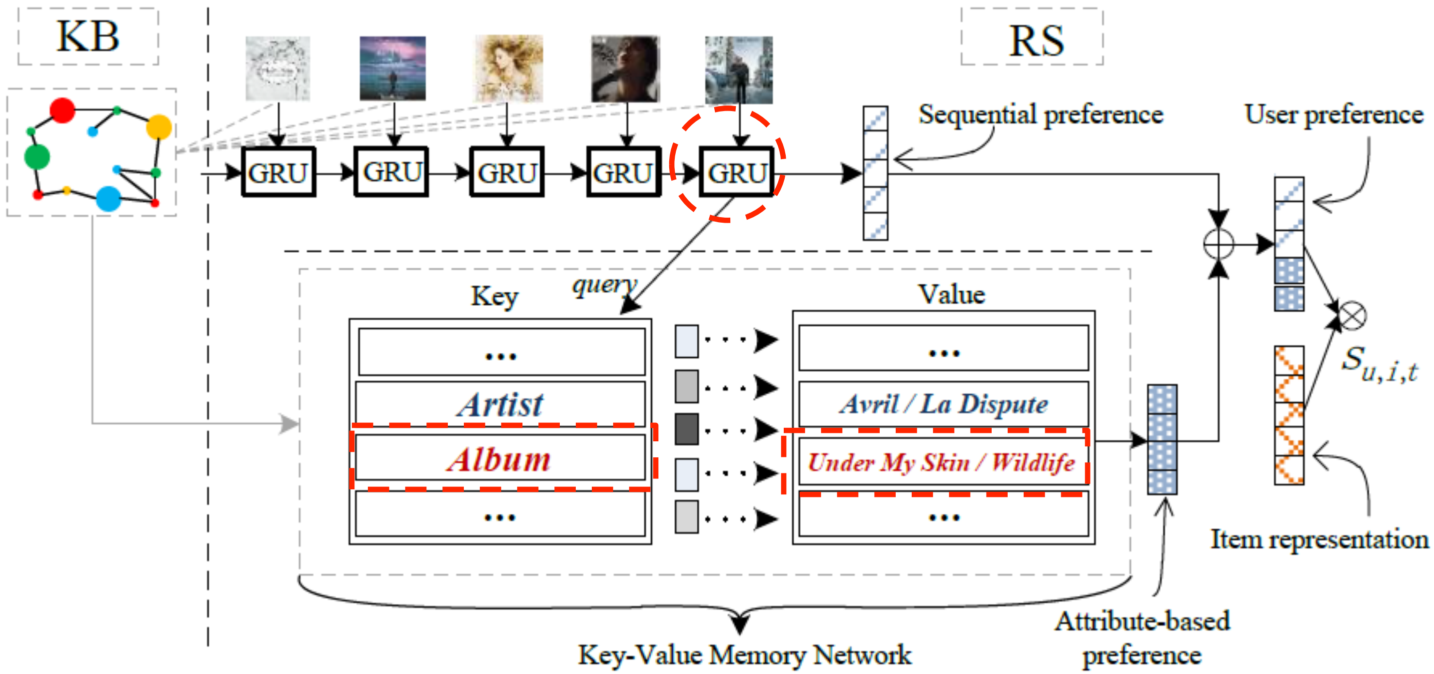
状态解码：获取与更新用户细粒度的偏好

对应value网络存储了用户过去该属性上的偏好

[SIGIR'18] Improving Sequential Recommendation with Knowledge-enhanced Memory Networks.

示例论文简介

- 基于知识属性的序列化推荐算法



模型优点：GRU的时序预测/推理能力+知识的结构化信息+记忆网络长期存储能力 → "预测+知识+记忆"

[SIGIR'18] Improving Sequential Recommendation with Knowledge-enhanced Memory Networks.

引文(Introduction)写法

- 论文的“门面”，必须写好
  - 引文一般包含的内容
    - 交代研究任务
    - 阐述研究现状并总结不足
    - 提出解决的新思路
    - 给出新方案的设计
    - 总结论文的贡献以及实验结论

引文(Introduction)写法

- 论文的“门面”，必须写好
  - 引文一般包含的内容
    - 交代研究任务
      - 让读者知道你要做的任务
    - 阐述研究现状并总结不足
      - 给读者一些研究背景的铺垫，并且带出挑战以及难点
        - 挑战如果非常重要，可以单独出一段
    - 提出解决的新思路
      - 引出论文的解决思路或者说idea
    - 给出新方案的设计
      - 较为详细地介绍idea的实现
    - 总结论文的贡献以及实验结论
      - 总结并且强调论文的贡献

引文(Introduction)写法

- 交代研究任务
  - 让读者知道你要做的任务

With the popularization of GPS-enabled mobile devices, a huge volume of trajectory data from users has become available in a variety of domains [27–29]. *Personalized Route Recommendation (PRR)* is one of the core functions in many online location-based applications, *e.g.*, online map. Given the road network, PRR aims to generate user-specific route suggestions on instant queries about the path planing from a source to a destination [6, 7]. It is challenging to perform effective pathfinding in a large and complex road network. For accurate route recommendation, it is necessary to consider rich context information, including personalized preference, spatial-temporal influence and road network constraint.

引文(Introduction)写法

- 阐述研究现状并总结不足
  - 给读者一些研究背景的铺垫

Early studies cast the route recommendation task as a pathfinding problem on graphs [17, 30]. These methods mainly focus on how to extend existing search algorithms (*e.g.*, Dijkstra shortest path algorithms and A^* search algorithm) for the studied task. With suitable heuristics, they can substantially reduce the search space and obtain high-quality responses. The key of heuristic search algorithms is to develop an effective cost function. Most of previous studies heuristically set the cost function, making their applicability highly limited. In addition, it is difficult to utilize various kinds of context information in the search process. To construct more flexible approaches, many studies have utilized machine learning methods for solving the PRR task [4, 32]. These methods are able to characterize the location dependencies or spatial-temporal information with principled models. While, most of them are shallow computational models, and may have difficulties in capturing complex trajectory patterns. With the revival of deep learning, it sheds light on the development of more effective PRR models using neural networks. Especially, sequential neural models, *i.e.*, Recurrent Neural Networks (RNN), have been widely used for modeling sequential trajectory data [1, 31, 34]. However, to our knowledge, these models mainly focus on one-step or short-term location prediction, which may not be suitable for the PRR task.

引文(Introduction)写法

- 提出解决的新思路
  - 引出论文的解决思路或者说idea，可以再介绍这一思路要实现的困难

Comparing the above approaches, we can see they have their own merits for the PRR task. On one hand, in terms of the problem setting, heuristic search algorithms are specially suitable for the PRR task, which can be considered as a pathfinding problem on graphs given the source and destination. They are able to generate high-quality approximate solutions using elaborate heuristics. On the other hand, as a newly emerging direction of machine learning, deep learning methods are effective to capture the complex data characteristics using learnable neural networks. They are able to learn effective mapping mechanisms from input to output or expressive feature representations from raw data in an automatic way. For developing a more effective PRR method, is there a principled way to combine the merits of both kinds of approaches?

Inspired by recent progress of deep learning in strategy-based games (*e.g.,* Go and Atari) [18, 23] , we propose to improve search algorithms with neural networks for solving the PRR task. Especially, we adopt the A^* algorithm [10] as the base search algorithm, since it has been widely used in pathfinding and graph traversal. Previous studies have also shown that A^* algorithm is a promising approach to solving the route recommendation task [11, 20, 30]. The main idea of our solution is to automatically learn the cost functions in A^* algorithms, which is the key of heuristic search algorithms. For this purpose, there are three important issues to consider. First, we need to define a suitable form for the cost in the PRR task. Different from traditional graph search problems, a simple heuristic form can not directly optimize the goal of our task [11, 30], *e.g.,* the route based on the shortest distance may not meet the personalized needs of a specific user. Second, we need to design effective models for implementing cost functions with different purposes, and unify different cost functions for deriving the final cost. The entire cost function $f(\cdot)$ of A^* can be decomposed into two parts, *i.e.,* $f(\cdot) = g(\cdot) + h(\cdot)$. The two parts compute the *observable cost* from the source node to the evaluation node and the *estimated cost* from the evaluation node to the destination node respectively. Intuitively, the two parts require different modeling methods, and need to jointly work to compute the entire cost. Third, we need to utilize rich context or constraint information for improving the task performance. For example, spatial-temporal influence and road networks are important to consider in modeling trajectory patterns, and should be utilized to develop the cost functions.

引文(Introduction)写法

- 给出新方案的设计
  - 详细地介绍idea的实现

To address these difficulties, we propose a novel neuralized A^* search algorithm for solving the PRR task. To define a suitable form for the search cost, we formulate the PRR task as a conditional probability ranking problem, and compute the cost by summing the negative log of conditional probabilities for each trajectory point in a candidate trajectory. We use this form of cost to instruct the learning of the two cost functions in A^* algorithm, namely $g(\cdot)$ and $h(\cdot)$. For implementing $g(\cdot)$, we propose to use attention-based RNNs to model the trajectory from the source location to the candidate location. We incorporate useful context information to better capture sequential trajectory behaviors, including spatial-temporal information, personalized preference and road network constraint. Instead of simply computing a single cost, our model also learns a time-varying vectorized representation for the moving state of a user. For learning $h(\cdot)$, we propose to use a value network for estimating the cost for unobserved part of a trajectory. In order to capture the complex characteristics of road networks, we build the value network on top of improved graph attention networks by incorporating useful context information. In these two different

引文(Introduction)写法

- 总结论文的贡献以及实验结论
  - 总结并且强调论文的贡献

To the best of our knowledge, we are the first to use neural networks for improving A^* algorithm in the PRR task. Our approach is able to automatically learn the cost functions without handcrafting heuristics. It is able to effectively utilize context information and characterize complex trajectory characteristics, which elegantly combines the merits of A^* search algorithms and deep learning. The two components are integrated in a joint model for deriving the evaluation cost. Extensive results on the three datasets have shown the effectiveness and robustness of the proposed model.

引文(Introduction)写法

- 引文中的灵魂="逻辑"
  - 语法只是皮囊
- 常见错误逻辑
  - 因为模型A好使，所以用A做某任务（蓄意的科研）
  - 因为任务B没有人做，所以我做了（暴力的科研）
  - 之前的人做了什么工作，我做了什么工作，我的好（缺乏解释、对比）
  - 这个任务很难，这篇论文我们这样解决了它（缺乏过渡、解释）
  - 夸大自己模型的贡献、忽略别人的工作（Reviewer最反感的写法）

引文(Introduction)写法

- 引文中的“度”
  - 包装需要，但是要适宜

- 常见不合适的“度”
  - 随意给出一些非常主观的意见
    - 例如：CNN model is perfectly good at modeling xxx data
      - 加引用可能能缓解
  - 随意使用一些特别general的词汇
    - 例如：“knowledge”、“context”、“information”等
  - 随意夸大自己的模型、放大自己的贡献
    - 例如：
      - 之前的工作这也不行那也不行，就我的方法行
        - » 例子：Our model significantly improves over all previous methods
      - 建议从头到尾检查一下所有副词、形容词，去掉带有太多主观色彩的词汇，也要慎重使用一些程度强烈的词汇（如significantly）
      - 加一些限定词
        - » “on xxx task in terms of xxx metrics”
        - » “all previous methods”---“all the comparison methods”

引文(Introduction)写法

- 引文中的“铺垫”
  - 引文的一个作用就是要让大家容易读懂全文

- 常见铺垫
  - 解释“主要术语”，避免拿来术语就用
    - 例如：knowledge-aware recommender systems
  - 解释清楚模型的主要设计思路、技术路线
    - 常见误区：面面俱到介绍所有细节（调参等），只需要写清楚最重要的部分，有些辅助组件甚至可以不提
    - 避免自己的盲目沉浸：避免在introduction里云里雾里，把模型说的特别高大上，让人摸不到头脑
    - 画图、举例子是好的解决方法
  - 强调主要创新之处
    - 需要写出一至两句非常中肯而又学术性的话来总结自己的创新点
      - 所有论文都有责任让自己的idea能够用1-3句话说清楚
    - 小tip：这样的话放在显眼的位置，如段首

引文(Introduction)写法

- 初学者写法-六句话扩充法
  - 第一句写任务介绍以及意义
  - 第二句概述研究现状以及主要的问题
  - 第三句写解决这些问题的研究挑战
  - 第四句写当前方法的主要出发点以及解决思路
  - 第五句写当前方法的主要技术方案
  - 第六句写总结、强调贡献

引文(Introduction)写法

- 过渡词的使用
  - Yet
  - although
  - Though
  - However
  - Still
  - while
  - Because
  - Since
  - and
  - but

引文(Introduction)写法

- 引文的写作是一门“讲故事”的艺术
  - 期望的引文样子（逐步递增)
    - 让人知道你在做什么
    - 让人觉得你的工作重要
    - 让人觉得你的解决思路清奇
    - 让人觉得你的工作不可或缺
    - 让人读起来如沐春风
  - 节奏感很重要
    - 交代背景（入）、铺垫包袱（难）、谜底揭晓（奇）、细节解析（懂）、重点总结（收）

相关工作(Related work)写法

- 主要要求
  - 尽量覆盖所有相关的相关工作
  - 分类整理
  - 突出相关的地方
  - 强调不同的地方

相关工作(Related work)写法

- 尽量覆盖所有相关的工作
  - 一些特定的排版style是有帮助的

2 RELATED WORK

Our work is related to the following research directions.

Route Recommendation. With the availability of user-generated trajectory information, route recommendation has received much attention from the research community [6, 7, 11], which aims to generate reachable paths between the source and destination locations. The task can be defined as either *personalized* [6, 7] or *non-personalized* [4, 11, 17, 35], and constructed based on different types of trajectory data, *e.g.*, GPS data [35] or POI check-in data [3, 22]. In the literature, various methods have been developed for route recommendation, including graph search algorithms [4, 15, 30], time-sensitive algorithms [17], A^* search algorithm [11], Go to page 9 bilistic POI transition/ranking models [3] and diver-direction based methods [35]. Overall, most of the studies focus on using search based algorithms or probabilistic models by considering additional constraints, *e.g.*, road networks or time. Our work is built on top of search based solutions, and the novelty lies in the automatic learning of the cost functions using neural networks. Our model is flexible to incorporate rich context or constraint information.

Deep Learning for Trajectory Data Mining. Recent years have witnessed the success of deep learning in modeling complex data relations or characteristics. In specific, Recurrent Neural Network (RNN) together with its variant Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been widely used for modeling sequential trajectory data. Typical works include hierarchical RNN [36], RNN with road network constraints [31], and multi-modal embedding RNN [9], spatial-temporal RNN [16] and space time feature-based RNN [1]. These studies mainly focus on short-term trajectory behaviors, *e.g.*, one-step location recommendation [16], which are not suitable for solving the current task.

Machine Learning for Heuristic Search. These studies in this direction aim to automatically improve or optimize the search algorithms with machine learning methods. Early works include the use of machine learning in creating effective, likely-admissible or improved heuristics [8, 13, 21]. More recently, deep learning has significantly pushed forward the research of this line. The main idea is

improving the tasks that require complicated solving strategies, including the Go game [23] and Atari games [18]. Our work is highly inspired by these pioneering works, but have a quite different focus on the studied task, *i.e.*, personalized route recommendation. Our task itself involves specific research challenges that make the reuse of previous works impossible.

相关工作(Related work)写法

- 突出相关的地方、并且强调不同的地方

Route Recommendation. With the availability of user-generated trajectory information, route recommendation has received much attention from the research community [6, 7, 11], which aims to generate reachable paths between the source and destination locations. The task can be defined as either *personalized* [6, 7] or *non-personalized* [4, 11, 17, 35], and constructed based on different types of trajectory data, *e.g.*, GPS data [35] or POI check-in data [3, 22]. In the literature, various methods have been developed for route recommendation, including graph search algorithms [4, 15, 30], time-sensitive algorithms [17], A^* search algorithm [11], probabilistic POI transition/ranking models [3] and diver-direction based methods [35]. Overall, most of the studies focus on using search based algorithms or probabilistic models by considering additional constraints, *e.g.*, road networks or time. Our work is built on top of search based solutions, and the novelty lies in the automatic learning of the cost functions using neural networks. Our model is flexible to incorporate rich context or constraint information.

相关工作(Related work)写法

- 长度的控制
  - 短写

Route Recommendation. With the availability of user-generated trajectory information, route recommendation has received much attention from the research community [6, 7, 11], which aims to generate reachable paths between the source and destination locations. The task can be defined as either *personalized* [6, 7] or *non-personalized* [4, 11, 17, 35], and constructed based on different types of trajectory data, *e.g.*, GPS data [35] or POI check-in data [3, 22]. In the literature, various methods have been developed for route recommendation, including graph search algorithms [4, 15, 30], time-sensitive algorithms [17], A^* search algorithm [11], probabilistic POI transition/ranking models [3] and diver-direction based methods [35]. Overall, most of the studies focus on using search based algorithms or probabilistic models by considering additional constraints, *e.g.*, road networks or time. Our work is built on top of search based solutions, and the novelty lies in the automatic learning of the cost functions using neural networks. Our model is flexible to incorporate rich context or constraint information.

相关工作(Related work)写法

- 长度的控制
  - 长写

Route Recommendation Algorithms. With the availability of user-generated trajectory information, route recommendation has received much attention from the research community [4], [5], [16], which aims to generate reachable paths between the source and destination locations. The task can be defined as either *personalized* [4], [5], [17] or *non-personalized* [7], [9], [16], [18], [19], and constructed based on different types of trajectory data, *e.g.*, GPS data [19] or POI check-in data [20], [21]. In the literature, various algorithms have been developed for route recommendation. Wei et al. [6], [9], [22] utilized graph search algorithms for identifying the path over the road network; Luo et al. [7] proposed time-sensitive algorithms to find the most frequent path in a specific time period; Kanoulas et al. [16] proposed a

相关工作(Related work)写法

- 常见错误
  - 简单罗列（平时读论文要做到分类总结）
    - 例如：A做了什么、B做了什么、C做了什么。。。
  - 没有说清楚区别和联系（平时读论文要做到分类总结）
  - 时态
    - 过去时
    - 现在完成时
    - 哪种都可以，但是不要混着来

相关工作(Related work)写法

- 常见错误
  - [13] constructed the first study
  - As shown in Zhao et al., 2018
  - (Zhao et al., 2018) studied the problem of

相关工作(Related work)写法

- 常见错误

  - In-text citations include the surname of the author and date, either both inside parentheses or with the author names in running text and the date in parentheses. For example:
    "Recently, Johnson (2014) has shown that" or
    "This has recently been shown (Johnson, 2014)"
  - If there are two authors, name both:
    "This method was developed by Johnson and Smith (2012)"
  - If there are more than two authors, use the et al. (*et alii*; "and others") convention:
    "This was based on a method introduced by Smith et al. (2002)"
  - If more than one references are cited at one location in the text, order them chronologically in the running text separated by a comma:
    "this was discussed by Smith et al. (2002), Johnson and Smith (2012), and Johnson (2014)"
    or order them between brackets separated using a semicolon:
    "...has widely been recognised (Smith et al., 2002; Johnson and Smith, 2012; Johnson, 2014)"
  - For citations of multiple works by the same authors in the same year, add lowercase letters (a, b, c, ...) after the year. The name or names of the authors do not needed to be repeated, for example:
    "This method has been extensively applied (e.g., Murphy and Wong, 2014a, 2014b; Wong, 2014)"
  - In the unlikely case, it was impossible to trace the original publication, refer to both the original work and the work it was cited in, for example:
    "According to Peterson (1873), cited by Vanderkeelen (1999)"

定义部分(Definition)写法

- 定义部分的写作
  - 介绍清楚所有术语
  - 给出所有符号的含义以及使用方式
  - 形式化地描述清楚任务

定义部分(Definition)写法

- 介绍清楚所有术语

In our task, we assume road network information is available for the pathfinding task, which is the foundation of the traffic communication for users.

Definition 1. ***Road Network***. *A road network is a directed graph* $\mathcal{G} = (\mathcal{L}, \mathcal{E})$, *where* $\mathcal{L}$ *is a vertex set of locations and* $\mathcal{E} \subset \mathcal{L} \times \mathcal{L}$ *is an edge set of road segments. A vertex* $l_i \in \mathcal{L}$ (i.e., *a location) represents a road junction or a road end. An edge* $e_{l_i,l_j} = \langle l_i, l_j \rangle \in \mathcal{E}$ *represents a directed road segment from vertex* l_i *to vertex* l_j.

Definition 2. ***Route***. *A route* (a.k.a., *a path*) p *is an ordered sequence of locations connecting the source location* l_s *with the destination location* l_d *with* m *intermediate locations,* i.e., $p : l_s \rightarrow l_1 \rightarrow \ldots \rightarrow l_m \rightarrow l_d$, *where each pair of consecutive locations* $\langle l_i, l_{i+1} \rangle$ *corresponds to a road segment* $e_{l_i,l_{i+1}}$ *in the road network.*

The moving trajectory of a user on the road network can be recorded using GPS-enabled devices. Due to instrumental inaccuracies, the sampled trajectory points may not be well aligned with the locations in $\mathcal{L}$. Following [33], we can preform the procedure of *map matching* for aligning trajectory points with locations in $\mathcal{L}$.

Definition 3. ***Trajectory***. *A trajectory* t *is a time-ordered sequence of* m *locations (after map matching) generated by a user,* i.e., $t : \langle l_1, b_1 \rangle \rightarrow \langle l_2, b_2 \rangle \rightarrow \ldots \rightarrow \langle l_m, b_m \rangle$, *where* b_i *is the visit timestamp for location* l_i.

定义部分(Definition)写法

- 给出所有符号的含义以及应用方式
  - 变量的符号极其重要
    - 要好看
    - 要好记
    - 要成体系

定义部分(Definition)写法

- 一个Notation的重要参考
  - http://www.deeplearningbook.org/contents/notation.html

Numbers and Arrays

a	A scalar (integer or real)
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\mathsf{\mathbf{A}}$	A tensor
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context
$\boldsymbol{e}^{(i)}$	Standard basis vector $[0, \ldots, 0, 1, 0, \ldots, 0]$ with a 1 at position i
$\mathrm{diag}(\boldsymbol{a})$	A square, diagonal matrix with diagonal entries given by $\boldsymbol{a}$
a	A scalar random variable
$\mathbf{a}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable

定义部分(Definition)写法

- 一个Notation的重要参考
  - http://www.deeplearningbook.org/contents/notation.html

Sets and Graphs

$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
$\{0, 1, \dots, n\}$	The set of all integers between 0 and n
$[a, b]$	The real interval including a and b
$(a, b]$	The real interval excluding a but including b
$\mathbb{A} \backslash \mathbb{B}$	Set subtraction, i.e., the set containing the elements of $\mathbb{A}$ that are not in $\mathbb{B}$
$\mathcal{G}$	A graph
$Pa_{\mathcal{G}}(\mathrm{x}_i)$	The parents of x_i in $\mathcal{G}$

定义部分(Definition)写法

- 一个Notation的重要参考
  - http://www.deeplearningbook.org/contents/notation.html

Indexing

a_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
a_{-i}	All elements of vector $\boldsymbol{a}$ except for element i
$A_{i,j}$	Element i, j of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$
$\mathsf{A}_{i,j,k}$	Element (i, j, k) of a 3-D tensor $\boldsymbol{\mathsf{A}}$
$\boldsymbol{\mathsf{A}}_{:,:,i}$	2-D slice of a 3-D tensor
a_i	Element i of the random vector $\mathbf{a}$

定义部分(Definition)写法

- 一个Notation的重要参考
  - http://www.deeplearningbook.org/contents/notation.html

Linear Algebra Operations

$\boldsymbol{A}^\top$	Transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^+$	Moore-Penrose pseudoinverse of $\boldsymbol{A}$
$\boldsymbol{A} \odot \boldsymbol{B}$	Element-wise (Hadamard) product of $\boldsymbol{A}$ and $\boldsymbol{B}$
$\det(\boldsymbol{A})$	Determinant of $\boldsymbol{A}$

定义部分(Definition)写法

- 一个Notation的重要参考
  - http://www.deeplearningbook.org/contents/notation.html

Calculus

$\dfrac{dy}{dx}$	Derivative of y with respect to x
$\dfrac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	Matrix derivatives of y with respect to $\boldsymbol{X}$
$\nabla_{\mathsf{X}} y$	Tensor containing derivatives of y with respect to X
$\dfrac{\partial f}{\partial \boldsymbol{x}}$	Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$
$\nabla_{\boldsymbol{x}}^2 f(\boldsymbol{x})$ or $\boldsymbol{H}(f)(\boldsymbol{x})$	The Hessian matrix of f at input point $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathbb{S}$

定义部分(Definition)写法

- 一个Notation的重要参考
  - http://www.deeplearningbook.org/contents/notation.html

Probability and Information Theory

$\mathrm{a} \perp \mathrm{b}$	The random variables a and b are independent
$\mathrm{a} \perp \mathrm{b} \mid \mathrm{c}$	They are conditionally independent given c
$P(\mathrm{a})$	A probability distribution over a discrete variable
$p(\mathrm{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathrm{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{\mathrm{x}\sim P}[f(x)]$ or $\mathbb{E}f(x)$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$
$\mathrm{Var}(f(x))$	Variance of $f(x)$ under $P(\mathrm{x})$
$\mathrm{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $P(\mathrm{x})$
$H(\mathrm{x})$	Shannon entropy of the random variable x
$D_{\mathrm{KL}}(P\|Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

定义部分(Definition)写法

- 一个Notation的重要参考
  - http://www.deeplearningbook.org/contents/notation.html

Functions

$f : \mathbb{A} \rightarrow \mathbb{B}$	The function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$f \circ g$	Composition of the functions f and g
$f(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$. (Sometimes we write $f(\boldsymbol{x})$ and omit the argument $\boldsymbol{\theta}$ to lighten notation)
$\log x$	Natural logarithm of x
$\sigma(x)$	Logistic sigmoid, $\dfrac{1}{1+\exp(-x)}$
$\zeta(x)$	Softplus, $\log(1+\exp(x))$
$\|\|\boldsymbol{x}\|\|_p$	L^p norm of $\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|$	L^2 norm of $\boldsymbol{x}$
x^+	Positive part of x, i.e., $\max(0, x)$
$\mathbf{1}_{\text{condition}}$	is 1 if the condition is true, 0 otherwise

定义部分(Definition)写法

- 一个Notation的重要参考
  - http://www.deeplearningbook.org/contents/notation.html

Datasets and Distributions

p_{data}	The data generating distribution
$\hat{p}_{\text{data}}$	The empirical distribution defined by the training set
$\mathbb{X}$	A set of training examples
$\boldsymbol{x}^{(i)}$	The i-th example (input) from a dataset
$y^{(i)}$ or $\boldsymbol{y}^{(i)}$	The target associated with $\boldsymbol{x}^{(i)}$ for supervised learning
$\boldsymbol{X}$	The $m \times n$ matrix with input example $\boldsymbol{x}^{(i)}$ in row $\boldsymbol{X}_{i,:}$

定义部分(Definition)写法

- Notation的例子
  - 应该符合人们常用的思维

DEFINITION 1. ***Road Network***. *A road network is a directed graph* $\mathcal{G} = (\mathcal{L}, \mathcal{E})$, *where* $\mathcal{L}$ *is a vertex set of locations and* $\mathcal{E} \subset \mathcal{L} \times \mathcal{L}$ *is an edge set of road segments. A vertex* $l_i \in \mathcal{L}$ (i.e., *a location*) *represents a road junction or a road end. An edge* $e_{l_i, l_j} = \langle l_i, l_j \rangle \in \mathcal{E}$ *represents a directed road segment from vertex* l_i *to vertex* l_j.

DEFINITION 2. ***Route***. *A route* (a.k.a., *a path*) p *is an ordered sequence of locations connecting the source location* l_s *with the destination location* l_d *with* m *intermediate locations,* i.e., $p : l_s \rightarrow l_1 \rightarrow \ldots \rightarrow l_m \rightarrow l_d$, *where each pair of consecutive locations* $\langle l_i, l_{i+1} \rangle$ *corresponds to a road segment* $e_{l_i, l_{i+1}}$ *in the road network.*

定义部分(Definition)写法

- Notation的例子
  - 活用上下标

We first introduce the notations used throughout the paper. In a recommender system (RS), let $\mathcal{U}$ denote a set of users and $\mathcal{I}$ denote a set of items. Our task focuses on the recommendation scenario with implicit feedback [25, 26], where we only concern whether a user $u \in \mathcal{U}$ has interacted with an item $i \in \mathcal{I}$ at time t. By sorting the interaction records by time ascendingly, we can form the *interaction sequence* for user u, namely $\{i_1^{(u)}, \cdots, i_t^{(u)}, \cdots, i_{n_u}^{(u)}\}$, where $i_t^{(u)}$ is the item that u has interacted with at time t and n_u is the length of interaction records for user u. Following [26], we use the relative time index instead of absolute time index for numbering interaction records.

Given the interaction sequence $\{i_1, \cdots, i_t\}^1$ of user u, our GRU-based recommender computes the current hidden state vector $\boldsymbol{h}_t^u \in \mathbb{R}^{L_H}$ conditioned on previous hidden state vector $\boldsymbol{h}_{t-1}^u$ as below

$$\boldsymbol{h}_t^u = \mathrm{GRU}(\boldsymbol{h}_{t-1}^u, \boldsymbol{q}_{i_t}; \Theta), \tag{1}$$

where GRU($\cdot$) is the GRU unit [4], $\boldsymbol{q}_{i_t}$ is the embedding vector for item i_t, and Θ denotes all the related parameters of GRU networks. The embedding vector $\boldsymbol{q}_{i_t} \in \mathbb{R}^{L_H}$ is called *item embedding*, which can be fixed or learned. In this way, the predictor encodes the interaction sequence of u into a hidden vector $\boldsymbol{h}_t^u$, which models the sequential preference of u at time t. Hence, we call $\boldsymbol{h}_t^u$ *sequential preference representation* of user u.

定义部分(Definition)写法

- Notation的例子
  - 部分问题
    - 一个符号多次使用，又代表不同意思
    - 全部使用未加粗的notation表示集合、矩阵等
    - 符号使用不按照习惯使用
    - 频繁使用一些单词的缩写用于notation (src, dest)
    - 频繁使用上下角标都存在的符号
    - 符号的数量非常多
    - \log, \exp \min ···

模型部分(Method)写法

- 模型部分的写作
  - 逻辑很关键
  - 几种常见的逻辑
    - 总-分-式
    - 总-基础-增强式

模型部分(Method)写法

- 模型部分的写作
  - 总-分-式

5 THE NASR MODEL

In the section, we present the proposed *Neuralized A-Star based personalized route Recommendation (NASR)* model.

5.1 Model Overview

Our model is developed based on the general A^* algorithm framework. For node evaluation, we decompose the entire cost function $f(\cdot)$ into two parts, namely *observable cost* and *estimated cost*, which

5.2 Modeling the Observable Cost with RNN

This part studies the learning of function $g(\cdot)$ for observable cost. Given an observed sub-route $l_s \rightarrow l_1 \rightarrow l_2 \cdots \rightarrow l_i$, as shown in Eq. (4), the problem becomes how to effectively learn the conditional transition probabilities $\Pr(l_{k+1}|l_s \rightarrow l_k, q, u)$. Simple

5.3 Modeling the Estimated Cost with Value Networks

Besides the observable cost, we need to learn the estimated cost from a candidate location to the destination. Specially, we introduce a value network to implement $h(\cdot)$. This part is more difficult to model since no explicit trajectory information is observed. In order to better utilize the road network information for estimation, we build the value network on top of an improved graph attention network with useful context information.

模型部分(Method)写法

- 模型部分的写作
  - 总-基础-增强式

4 THE PROPOSED APPROACH

In this section, we present the knowledge-enhanced sequential recommender. We start with a base sequential recommender using GRU networks, and then augment the base model with Key-Value Memory Networks using entity attribute information from KBs.

4.1 A GRU-based Sequential Recommender

Recurrent Neural Networks (RNN) have been shown effective in capturing and characterizing the temporal dependency in sequence data. A major problem of RNNs is that it suffers from the prob-

4.2 Augmenting Sequential Recommender with Knowledge-Enhanced Memory Networks

The GRU-based recommender encodes the user preference into a latent vector, which is less powerful to capture fine-grained preference over attribute or feature dimensions of items. Knowing detailed user interests in the attribute level is particularly useful to

模型部分(Method)写法

- 模型部分的写作
  - 建议配图

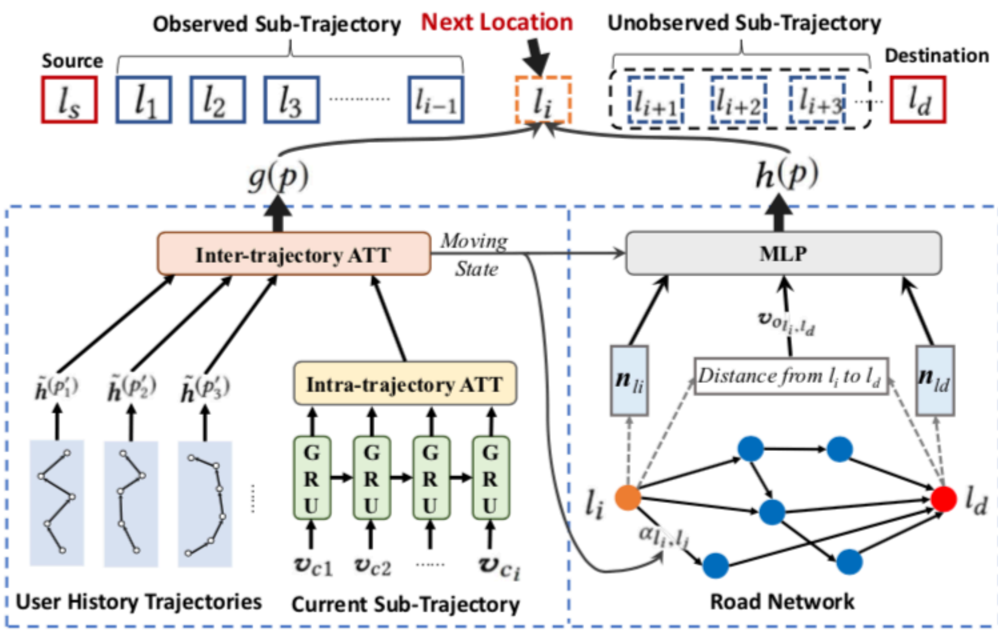


Figure 1: The overall architecture of the NASR model. $g(\cdot)$ **learns the cost from the source to a candidate location, called** ***observable cost***; $h(\cdot)$ **predicts the estimated cost from a candidate location to the destination, called** ***estimated cost*****.**

模型部分(Method)写法

- 模型部分的写作
  - 最后部分的讨论可能会给模型“添彩”

5.4 Model Analysis and Learning

Integrating the two components in Section 5.2 and 5.3, we obtain the complete NASR model for the PRR task. NASR follows the similar search procedure of A^* algorithm but uses the learned cost for node evaluation.Specially, it has fulfilled the cost functions of A^* algorithms with neural networks, namely $g(\cdot)$ and $h(\cdot)$. Given a candidate location, the first component utilizes RNNs to characterize the currently generated sub-trajectory for learning *observable cost*, while the second component incorporates a value network to predict the *estimated cost* to arrive at the destination. Finally, the two cost values are summed as the final evaluation cost of a candidate location.

Compared with traditional heuristic search algorithms, NASR has the following merits. First, it does not require to manually set functions with heuristics, but automatically learns the functions from data. Second, it can utilize various kinds of context information

模型部分(Method)写法

- 模型部分的写作
  - 最后部分的讨论可能会给模型“添彩”

We present a diagram sketch of our model in Fig. 2. We call our model *Knowledge-enhanced Sequential Recommender* (***KSR***). Our model has the following merits. First, the GRU network is able to effectively capture temporal dependency, yielding a sequential representation for user preference (*i.e.*, $\boldsymbol{h}_t^u$). Second, the KV-MN part is able to characterize the detailed user interests over item attributes, yielding an attribute-based representation for user preference (*i.e.*, $\boldsymbol{m}_t^u$). Third, the *hidden* sequential preference representation (*i.e.*, $\boldsymbol{h}_t^u$) is used to dynamically generate a set of attention weights (*i.e.*, $w_{t,u,a}$) over the *explicit* attributes, which provides the capacity of explaining the latent sequential preference in the attribute level. Putting all together, our model is endowed with the benefits from both GRU and KV-MN, and further enhanced with external structured knowledge information. Hence, our model is expected to be more powerful in sequential recommendation, effective and interpretable.

模型部分(Method)写法

- 模型部分的写作
  - 最后部分的讨论可能会给模型“添彩”
    - 正确性证明
    - 时间复杂度
    - 完整的算法流程
    - 参数汇总+学习算法
    - 与之前工作的区别（突出创新性）
    - 与之前工作的联系（增强泛化性）
    - 模型可扩展的地方（堵漏）

实验部分(Method)写法

- 一般流程
  - 数据集合、评测指标、评测流程
  - 对比方法
  - 主干实验分析
  - 模型细致分析
  - 定性实验

实验部分(Method)写法

- 数据集合

Construction of the Datasets. In our task, we need to prepare both KB and RS data. For KB data, we adopt the one-time FREEBASE [8] dump consisting of 63 million triples. For RS data, we use four datasets from different domains, namely LAST.FM music [28], MOVIELENS ml-20m [9], MOVIELENS ml-1m [9] and AMAZON book [11]. The LAST.FM music dataset is very large, and we take the subset from the last year; for the ml-20m dataset, we take the subset from year 2005 to 2015. Following [10, 26], we only keep the k-core dataset, and filter unpopular items and inactive users with fewer than k records, which is set to 3 in book dataset and 10 for the other datasets. Then, we link filtered items with FREEBASE entities. With an offline FREEBASE search API, we retrieve KB entities with item title (*e.g.,* song titles) as queries. Once mulitple entities are returned, we further incorporate at least one attribute as the filter to identify the only correct entity. We only keep the interactions related to the linked items in the final datasets. We group the interaction records by users, sort them according to the timestamps ascendingly, and form the interaction sequence for each user. To train TRANSE, we start with linked entities as seeds and expand the graph with one-step search. Not all the relations in KBs are useful, we remove unfrequent relations with fewer than 5,000 triples. We summarize the detailed statistics of the datasets in Table 1.

Table 1: Statistics of our datasets. #Entities indicates the number of entities that are extended by seed entities with one-step search in KBs for training TRANSE.

Datasets	#Interactions	#LinkedItems	#Users	#Entities	#Relations
Music	203,975	30,658	7,694	214,524	19
ml-20m	5,868,015	19,533	61,583	1,125,100	81
ml-1m	916,714	3,210	6,040	1,125,100	81
Book	828,560	69,975	65,125	313,956	49

一般流程：

原始数据集合->重要预处理步骤->最后的数据集合

Tip: 数字不要居中（靠右对齐），使用“逗号”表示法

实验部分(Method)写法

- 评测指标、评测流程

Evaluation Metrics. For the PRR task, we adopt a variety of evaluation metrics widely used in previous works [6, 12, 14]. Given an actual route p, we predict a possible route p' with the same source and destination. Following [6, 14], we use *Precision*, *Recall* and *F1-score* as evaluation metrics: $Precision = \frac{|p \cap p'|}{|p'|}$, $Recall = \frac{|p \cap p'|}{|p|}$ and $F1 = \frac{2*P*R}{P+R}$. *Precision* and *Recall* compute the ratios of overlapping locations *w.r.t.* the actual and predicted routes respectively. Besides, we use the *Edit distance* as a fourth measure [12], which is the minimum number of edit operations required to transform the predicted route into the actual route. Note the source and destination locations are excluded in computing evaluation metrics.

Task Setting. For each user, we divide her/his trajectories into three parts with a ratio of $7:1:2$, namely training set, validation set and test set. We train the model with training set, and optimize the model with validation set. Instead of reporting the overall performance on all test trajectories, we generate three types of queries *w.r.t.* the number of locations in the trajectories, namely *short* (10 to 20 locations), *medium* (20 to 30 locations) and *long* (more than 30 locations). In test set, given a trajectory, the first and last locations are treated as the source and destination respectively, and the

  - 新任务的评测指标要完全给出
  - 老任务的评测指标可以沿用，进而压缩空间

实验部分(Method)写法

- 对比方法

Methods to Compare. We consider the following comparisons:

•*RICK* [30]: It builds a routable graph from uncertain trajectories, and then answers a users online query (a sequence of point locations) by searching top-k routes on the graph.

•*MPR* [4]: It discovers the most popular route from a transfer network based on the popularity indicators in a breadth-first manner.

•*CTRR* [6]: It proposes collaborative travel route recommendation by considering user's personal travel preference.

•*STRNN* [16]: Based on RNNs, it models local temporal and spatial contexts in each layer with transition matrices for different time intervals and geographical distances.

•*DeepMove* [9]: It is a multi-modal embedding RNN that can capture the complicated sequential transitions by jointly embedding the multiple factors that govern the human mobility.

Among these baselines, RICK and MPR are heuristic search based methods, CTRR is a machine learning method, and STRNN and DeepMove are deep learning methods. The parameters in all the models have been optimized using the validation set.

介绍baseline，如果没有特殊实现，可以比较简略。重要参数要给出。

最后最好总结一下baseline，让读者有一个整体的了解

实验部分(Method)写法

- 对比方法

Our baselines have a comprehensive coverage of the related models. To summarize, we categorize the baselines into eight groups shown in Table 2, according to the *task orientation*, *with/without KB* and *with/without neural models*.

Table 2: The categorization of the comparison methods.

Tasks	KB	Neural (No)	Neural (Yes)
General	Yes	–	CKE
	No	BPR	NCF
Sequential	Yes	–	RUM, GRU_F, **KSR**
	No	FPMC	GRU, GRU++

表格是一个总结对比方法的途径

实验部分(Method)写法

- 主干实验
  - 要很清楚实验的目的
    - 对引文里面给出的贡献、发现或者结论的证明
    - 不要流水账一样介绍，要突出原因
    - 错误例子：A比B好、B比C好。。。
  - 有些异常结果要加以解释
    - 有些模型达不到原始论文的效果，要好好分析一下
  - 加上统计性显著检验
    - 确保提升是有效的
    - 有的时候是压死论文的最后一棵稻草

实验部分(Method)写法

- 主干实验

6.2 Results and Analysis

We present the results of all the comparison methods in Table 1. First, heuristic search methods, *i.e.*, RICK and MPR, perform very well, especially the RICK method. RICK fully characterizes the road network information and adopts the informed A^* algorithm. As a comparison, MPR mainly considers the modeling of transfer network and uses a relatively simple BFS search procedure. Second, the matrix factorization based method CTRR does not perform better than RICK and MPR. A possible reason is that CTRR can not well utilize the road network information. Besides, it has limited capacities in learning complicated trajectory characteristics. In our experiments, CTRR tends to generate short route recommendations, giving very bad recall results for medium and long queries. Third, deep learning method DeepMove performs very well among all the baselines, while STRNN gives a worse performance. Compared with STRNN, DeepMove considers more kinds of context information and designs more advanced sequential neural networks. Finally, the proposed model NASR is consistently better than all the baselines in all cases, yielding very good performance even on long queries.

By summarizing these results, we can see heuristic search methods are competitive to solve the PRR task, especially when suitable heuristics are used and context information is utilized. Besides, deep learning is also able to improve the performance by leveraging the powerful modeling capacity. Our proposed model NASR is able to combine both the benefits of heuristic search and neural networks, and hence it performs best among the comparison methods.

实验部分(Method)写法

- 细致性分析实验
  - 检查contribution的来源
    - Ablation study
  - 组件内部调节
  - 参数调节
  - 数据调节

实验部分(Method)写法

- 细致性分析实验

Effect of the RNN Component. We first examine the effect of the RNN component with different variants. We have incorporated two kinds of attentions, namely inter- and intra-trajectory attention in Section 5.2. Here, we consider three variants of the attention mechanism for implementing $g(\cdot)$: *without attention (NA)*, *using only intra-trajectory attention (IA)* and *using both intra- and inter-trajectory attention (BA)*. Recall our RNN component is also able to learn a vectorized representation for the moving state of users. We further prepare a variant for verifying the effect of the learned

Effect of the Value Network. Predicting the estimated cost (*i.e.*, $h(\cdot)$) of a candidate location is especially important for our task We use a value network for implementing $h(\cdot)$, which replaces the traditional heuristics. We now examine the performance of different variants for the value network. In this part, we fix the RNN component as its optimal setting. Then we prepare four variants for the value network as comparisons, including (1) *ED* using Euclid distance as heuristics, (2) *SP* using the scalar product between the embeddings of the candidate and destination locations, (3) *o-GAT* using the original implementation of graph attention networks, and (4) *i-GAT* using our improved GAT by incorporating context information. Both variants (3) and (4) are trained using the same TD learning method. In Fig. 2(b), it can be observed that the performance rank is as follows: *ED* < *SP* < *o-GAT* < *i-GAT*. We can see that the simplest spatial distance baseline *ED* gives the worst performance, which indicates simple heuristics may not work well in our task. Graph attention networks are more effective to capture structural characteristics from graphs. When incorporating context information, our value network is able to outperform the variant using original implementation.

Effect of Temporal Difference Learning Method. To learn our model, an important technique we apply is the Temporal Difference (TD) method. For verifying the effectiveness of the n-step TD method, we consider four variants for comparison, including (1) *SL* which directly learns the actual distance between the candidate location and the destination in a supervised way, (2) *MC* which applies Monte Carlo method to generate sampled sequences and trains the model with the cost of these sampled sequences [25], (3) *n-TD* which uses a TD step number of 5. From Fig. 2(c), we can see that the simplest supervised learning method performs worst. Since the prediction involves multi-step moving process, it is not easy to directly fit the distance using traditional supervised learning methods. Compared with all the methods, we can see that the 5-TD learning method is the most effective in our task. In our experiments, we find that using a step number of 5 produces the optimal performance.

实验部分(Method)写法

- 定性的例子
  - 用例子解释为什么你的想法会work
  - 用例子解释哪些情况，你的模型会有比较显著的提升
  - 这里是对于Introduction的最强呼应
    - 用例子论证introduction的motivation

实验部分(Method)写法

- 一般的流程
  - 避免探索式描写，要有核心驱动进行描写
  - 画一个好图：一图胜千言
  - 用好图的标题：各种符号、颜色以及整体场景的设置
    - 不要让文本描述和图的标题大部分一样，双方各有分工
  - 写一段清楚的描述
    - 首先写清楚目的
    - 接着写清楚当前例子的整体故事
    - 然后分解进入关键部分
    - 最后总结发现

实验部分(Method)写法

- 定性的例子

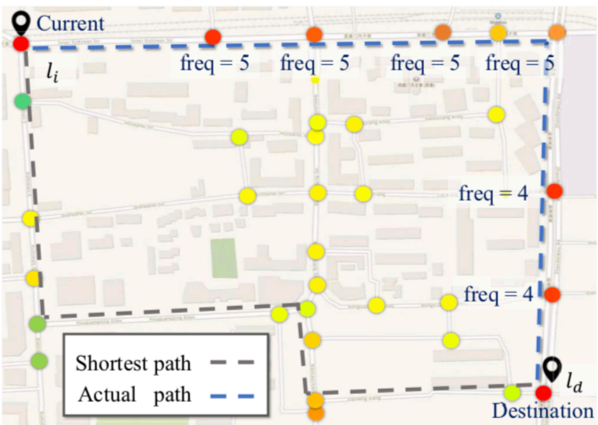


Figure 3: Visualization of the learned association scores using improved graph attention networks. The colored circles denote locations in the road network. A darker color indicates a larger importance degree *w.r.t.* current location l_i and destination l_d. "*freq*" denotes the visit frequency by the user in historical trajectories.

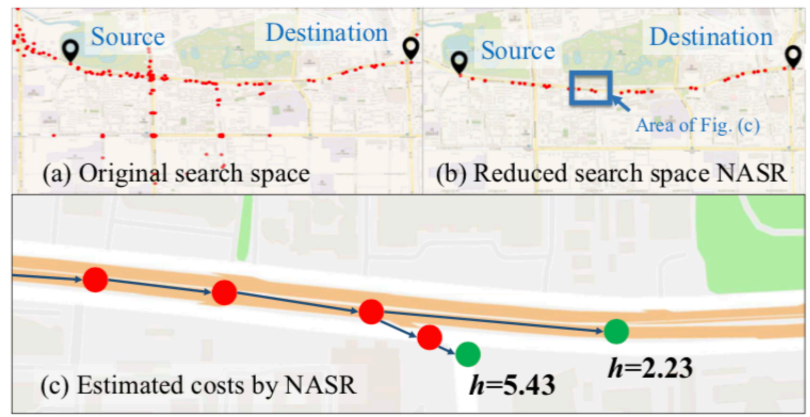


Figure 4: Visualization of the search procedure with the estimated costs by the NASR model. In (c), red points have been already explored and green points are candidate locations to extend in A^* search algorithm.

实验部分(Method)写法

- 定性的例子

In NASR, the improved graph attention network is the core component for modeling road network information. It can generate informative node representations for encoding structural characteristics. To see this, we present an illustrative example in Fig. 3. A user is currently located at l_i and moving towards the destination l_d. For a candidate location l_j, we compute a simple scoring formula: $\boldsymbol{n}_{l_j}^{\top} \cdot \boldsymbol{n}_{l_i} + \boldsymbol{n}_{l_j}^{\top} \cdot \boldsymbol{n}_{l_d}$, where $\boldsymbol{n}_{(\cdot)}$s are the node representations learned in Eq. (15). This formula measures the association degree of l_j with both current location and destination. For comparison, we plot both the actual and shortest route. As we can see, the locations on the actual route has a larger association weight than those on the shortest route. By inspecting into the dataset, we find the shortest route contains several side road segments that are possibly in traffic congestion at the visit time. Another interesting observation is that the user indeed visits the locations in the actual route more times in historical trajectories. These observations indicate that our model is able to learn effective node representations for identifying more important locations to explore for the PRR task.

Next, we continue to study how the learned cost function helps the search procedure in NASR. Figure 4 presents a sample trajectory from a specific user. Given the source and destination, we need to predict the actual route. By comparing Fig. 4(a) (the original search space) and Fig. 4(b) (the reduced search space by NASR), it can be seen that our model is able to effectively reduce the search space. When zooming into a subsequence of this route, we further compare the estimated cost values for two candidate locations (green points) in Fig 4(c). Although the second location has a longer distance with the explored locations, it is located on the main road that is likely to lead to a better traffic condition. Our model is able to predict a lower cost for the second location by effectively learning such trajectory characteristics from road network and historical data.

实验部分(Method)写法

- 定性的例子

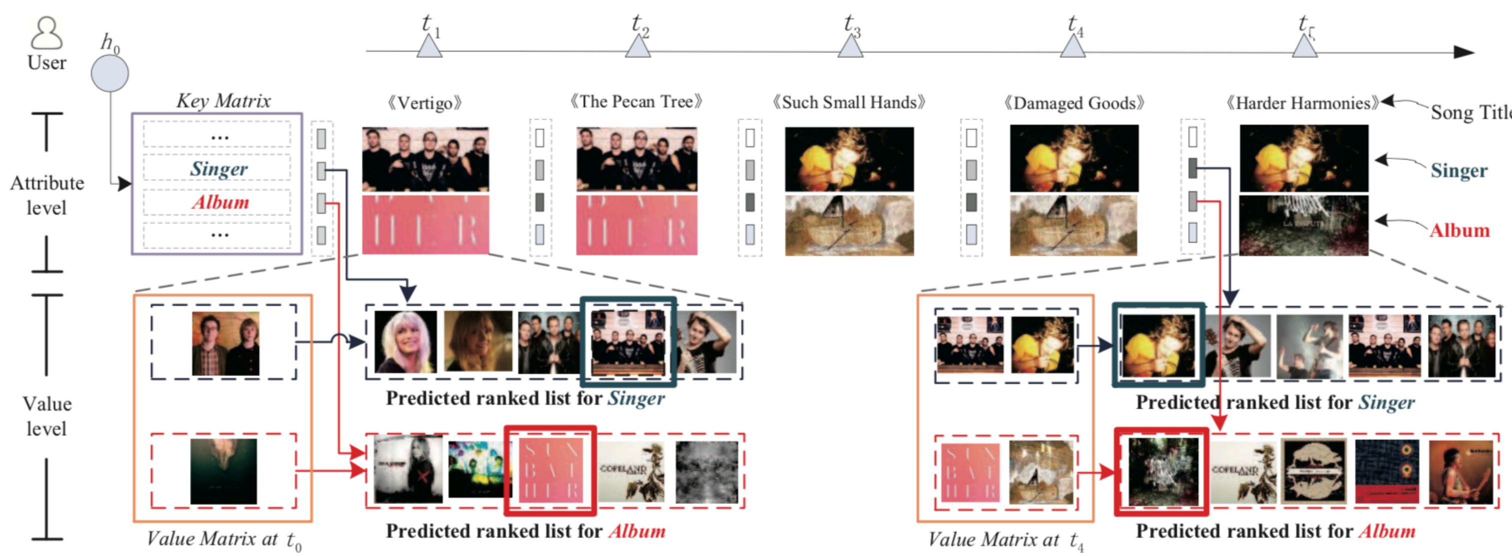


Figure 5: An interaction sequence from a sample user in music dataset. We use dark blue and red to indicate attributes of *album* and *singer* respectively. We present the predictions of our model KSR on attribute weights and value entities. For attention weights (top of the figure), we use color darkness to indicate the value of attention weights: darker is larger. For value entities (bottom of the figure), we show the predicted ranklist of candidate entities for both attributes at time t_1 and t_5 (using value matrices of KV-MN at t_0 and t_4). We highlight the correct entities in predicted ranklists with colored boxes.

实验部分(Method)写法

- 定性的例子

Attribute-level Interpretability. Fig. 5 presents an interaction sequence of five records from a sample user. Each record consists of two parts: the left part corresponds to the learned attention weights and the right part corresponds to ground-truth information of a song, including title, singer and album. First, the user started with two songs from the same album, which followed the way of listening by *album*. Then, she listened to two more songs from another album. For the fifth song, the user switched to a third album. The interesting point is that its singer is the same as that of previous two songs. Hence, for the last three songs, the user essentially followed a mixture of listening by *album* and listening by *singer*. It is clear that our model has predicted a larger weight on the attribute of *album* till the fourth record, and a larger weight on the attribute of *singer* on the fifth song. This example indicates the user preference is likely to be dynamic and evolving, and our model is able to capture evolving preference over the attributes.

实验部分(Method)写法

- 定性的例子

Value-level Interpretability. Suppose it is already known some attribute (*e.g.*, album) plays the key role in determining the interaction behavior of a user, can we further predict how the user will select among a set of entities for that attribute (*e.g.*, the selection of the favorite album in candidate albums)? For convenience, we call the entities (also in KB) corresponding to the attribute value of a RS item *value entities*, *e.g.*, *Deafheaven* is the value entity of attribute *singer* for song *The Pecan Tree*. Recall we have a user-specific value matrix in KV-MNs, which maintains the preference characteristics of a user on some specific attribute. We expect a value vector can reflect user preference over value entities for some attribute. A value vector $\boldsymbol{v}_a^u$ corresponds to a key vector $\boldsymbol{k}_a$ on attribute a. Since the value matrix is updated with KB embeddings of items (Eq. 7 and 9), the learned value vectors $\boldsymbol{v}_a^u$ can be represented in the same space as KB embeddings. Given an attribute, we can directly compute L_1 distance between the embedding of a candidate value entity (*e.g.*, $\boldsymbol{e}_{Deafheaven}$) and the user-specific value vector (*e.g.*, $\boldsymbol{v}_{singer}^u$) from the previous timestamp. Then, we rank the candidate value entities according to the L_1 distance and form a predication ranklist. We present the illustration of value-level interpretation at the bottom of Fig. 5. At the beginning (t_1), the value matrix is not well learned. By training with more records, our value matrix is able to dynamically trace the user preference on some specific attribute. At the fifth record (t_5), it correctly predicts the candidate entities for both *singer* and *album* attributes at the first position.

摘要(Method)写法

- 讲清楚任务、方法和创新点、不要提实现细节

Personalized Route Recommendation (PRR) aims to generate user-specific route suggestions in response to users' route queries. Early studies cast the PRR task as a pathfinding problem on graphs, and adopt adapted search algorithms by integrating heuristic strategies. Although these methods are effective to some extent, they require setting the cost functions with heuristics. In addition, it is difficult to utilize useful context information in the search procedure. To address these issues, we propose using neural networks to automatically learn the cost functions of a classic heuristic algorithm, namely A^* algorithm, for the PRR task. Our model consists of two components. First, we employ attention-based Recurrent Neural Networks (RNN) to model the cost from the source to the candidate location by incorporating useful context information. Instead of learning a single cost value, the RNN component is able to learn a time-varying vectorized representation for the moving state of a user. Second, we propose to use a value network for estimating the cost from a candidate location to the destination. For capturing structural characteristics, the value network is built on top of improved graph attention networks by incorporating the moving state of a user and other context information. The two components are integrated in a principled way for deriving a more accurate cost of a candidate location. Extensive experiment results on three real-world datasets have shown the effectiveness and robustness of the proposed model.

摘要(Method)写法

- 讲清楚任务、方法和创新点、不要提细节

With the revival of neural networks, many studies try to adapt powerful sequential neural models, *i.e.*, Recurrent Neural Networks (RNN), to sequential recommendation. RNN-based networks encode historical interaction records into a hidden state vector. Although the state vector is able to encode sequential dependency, it still has limited representation power in capturing complicated user preference. It is difficult to capture fine-grained user preference from the interaction sequence. Furthermore, the latent vector representation is usually hard to understand and explain.

To address these issues, in this paper, we propose a novel knowledge enhanced sequential recommender. Our model integrates the RNN-based networks with Key-Value Memory Network (KV-MN). We further incorporate knowledge base (KB) information to enhance the semantic representation of KV-MN. RNN-based models are good at capturing sequential user preference, while knowledge-enhanced KV-MNs are good at capturing attribute-level user preference. By using a hybrid of RNNs and KV-MNs, it is expected to be endowed with both benefits from these two components. The sequential preference representation together with the attribute-level preference representation are combined as the final representation of user preference. With the incorporation of KB information, our model is also highly interpretable. To our knowledge, it is the first time that sequential recommender is integrated with external memories by leveraging large-scale KB information.

总结 (Method)写法

- 简要总结工作，不要再埋包袱、打伏笔
  - 具体任务-解决方案-实验结果-未来扩展

In this paper, we took the initiative to use neural networks to automatically learn the cost functions in A^* for the PRR task. We first presented a simple A^* solution for solving the PRR task, and formally defined the suitable form for the search cost. Then, we set up two components to learn the two costs respectively, *i.e.,* the RNN component for $g(\cdot)$ and the value network for $h(\cdot)$. The two components were integrated in a principled way for deriving a more accurate cost of a candidate location for search. We constructed extensive experiments for verifying the effectiveness and robustness of the proposed model.

Since road network information is not always available, as future work, we will consider extending our model to solve the PRR task without road networks. Currently, we focus on the PRR task. We will also study whether our solution can be generalized to solve other complex search tasks.

我喜欢的改（写）论文流程

- 先写Related work，梳理清楚已有工作的不足，第一遍
- 然后改问题定义、模型，确保了解模型细节，第一遍
- 然后写Introduction，第一遍
- 然后写实验，第一遍
- Repeat “问题定义、模型、实验、相关工作”
- Repeat “Introduction、模型”
- 全文定稿
- 最后，abstract, conclusion，related work

我喜欢的改（写）论文方式

- 将文件按照section切割，使用input命令合并，共享一个main.tex
- 如果有多人进行操作，不同人改不同的文件
  - 拒绝网络软件共享修改
- 快速出初稿，然后打印出来多次重写
- 对于画图要求较高，通常一个图要修改几十次
- 最后写abstract, conclusion

其他应该避免的问题

- 大段文字摆上来，没有切割，没有分段
- 一个section内部的subsection太多，且各个部分关系不强
- 句子无意义的过长
- 在一个Section前面铺垫太长，再进入Subsection
- 对于很多常用词汇，习惯了错误用法（an approach to do?），而且不断强化错误
- 画很多无意义的图占论文空间（比如CNN架构）
- 实验图字体太小、清晰度不够
- 细枝末节介绍很详细，但是最关键的技术、实验部分不够清楚
- 避免论文中出现无用的大段空白
- 尽量少使用vspace、hspace、scriptsize等压缩命令

用词

Avoid informal or spoken language in scientific texts. Instead, use formal alternatives. Examples of informal words and their formal alternatives are:

Informal	Formal
• a lot of • do (verb) • big • like • think • talk • look at • get • keep • climb • really • things	• much, many • perform, carry out, conduct • large • such as • consider • discuss • examine • obtain • retain, preserve • ascend • … (leave out) • … (be precise)

In addition:

- Avoid contractions ("do not" instead of "don't")
- Avoid clichés ("this site is excellent for…" instead of "this site is the cream of the crop for")
- Avoid "one" as pronoun, use passive voice instead

建议

- 学一手好的LaTex，建议模板化、流程化、标准化
- 学会一些固定转折、承接、突出、总结、代入、发现、介绍等固定语句
- 重学四级单词或者学术论文常用单词，了解这些词的用法
  - 可以写个程序，找出来最频繁的词汇，以及他们的搭配，对于自己使用的新搭配要小心
- 建议学会一套好的画图技术、做表技术
- 相关工作应该平时准备好，对于bib提前找好
- 拼写检查

建议

- 最快的学习方法
  - 写论文、找人改（珍惜每一次老师给你的修改）
  - 多写论文是提升写作的唯一标准
- 最稳的学习方法
  - 阅读简单文章、背诵并且默写（偏学术的文章）
- 最廉价的学习方法
  - 读论文时不看abstract，然后写abstract，然后对比

期刊写作建议

- 学会把论文写长、但又看起来也不冗余
  - 图表、公式
  - 组织更为重要
- 尽量把参考文献写全
- 尽量把实验做全
- 写好Response重要

期刊写作建议

We appreciate the generosity of the anonymous reviewers. They provided many insightful comments and helpful suggestions. We respond to each reviewer separately.

Separate response

REVIEWER #1:

(1) Section 4.2.: To my taste, the description of quad max array is too verbose, to the extent that one assumes that there is more to it than is actually the case. A more formal definition like quadmax[i] = max{input[4*i],input[4*i+1],input[4*i+2],input[r*i+3]} might be simpler to digest.

RESPONSE: We revised the manuscript accordingly.

最后的建议

- 多写note
  - 回头看自己的note，语病真多，但是都是回忆

Technical Note Series **Spring 201**

Note 1: Varitional Methods for Latent Dirichlet Allocation

Version 1.0 *Wayne Xin Zhao (batmanfly@gmail.com*

Disclaimer: *The focus of this note was to reorganize the content in the original Blei's paper and add n detailed derivations. For convenience, in some part, I fully copied Blei's content. I hope it can help beginners to vEM of LDA.*

First of all, let us make some claims about the parameters and variables in the model.

Let K be the number of topics, D be the number of documents and V be the number of terms in vocabulary. We use i to index a topic [1], d to index a document [2], n index a word [3] and w (or v) to de a word. In LDA, $\boldsymbol{\alpha}_{K\times 1}$ and $\boldsymbol{\beta}_{K\times V}$ are *model parameters*, while $\boldsymbol{\theta}_{D\times K}$ and $\boldsymbol{z}$ [4] are *hidden variables*.

As a variational distribution $q(\cdot)$, we use a fully factorized model, where all the variables are independe governed by a different distribution,

$$q(\boldsymbol{\theta}, \boldsymbol{z}|\gamma, \boldsymbol{\phi}) = q(\boldsymbol{\theta}|\gamma)q(\boldsymbol{z}|\boldsymbol{\phi}), \quad ($$

where Dirichlet parameter $\boldsymbol{\gamma}_{D\times K}$ and the multinomial parameters $\boldsymbol{\phi}$ [5] are *variational parameters*. The t assignments of words and the documents are exchangeable, i.e., conditionally independent on the parame (either model parameters or variational parameters). Note that all the variational distributions $q(\cdot)$ here conditional distributions and should be written as $q(\cdot|\boldsymbol{w})$, for simplicity we write it as $q(\cdot)$.

The main idea is that we use variational expectation-maximization (EM): In the E-step variational EM use the variational approximation to the posterior described in the previous section and find the optimal ues of variational parameters. In the M-step, we maximize the bound with respect to the model paramet In a more condense way, we perform variational inference for learning variational parameters in E-step w perform parameter estimation in M-step. These two steps alternate in a iteration. We will optimize the lo bound w.r.t variational parameters and model parameters one by one, and this is to perform optimiza using a coordinate ascent algorithm.

Note for pLSA and LDA-Version 1.1

Wayne Xin Zhao

March 2, 2011

1 Disclaimer

In this part of PLSA, I refer to [4, 5, 1]. In LDA part, I refer to [3, 2]. Due to the limit of my English ability, in some place, I just copy the words from original papers or notes. I'm sorry I don't have time to get the approvement from these authors. I would remove or rewrite all such parts later. This note is strictly for studying instead of any other activies. Please just keep by yourself and don't distribute it. This note is done in two days, I didn't have time for reviewing but straightforward releasing it for possible materials of studying topic models. I would keep updating this note and try to make the notations consistent and correct all the errors. If you find any problem in it, feel free to contact me via `batmanfly@gmail.com`.

2 pLSA

2.1 Introduction

pLSA is one kind of mixture model. We assume that there are a collection of D documents, denoted by $\mathcal{C}$, and there are totally K latent topics, which are represented by a multinomial distribution over vocabulary and denoted by variable $z_i \in \{z_1, .., z_K\}$. In this model, all the words $\mathbf{w}$ are assumed observed variables while $\{z_i\}_{i=1}^{K}$ are unobserved.

致谢

- 感谢在我写作过程中给予重大帮忙的老师们

谢谢大家

- 有交流意向可以邮件沟通
  - batmanfly@qq.com