

PKourseUtopia——一款基于上下文学习、多准则决策与主成分分析的课程推荐系统

赵俊涵, 徐艺峰, 刘鲁阳, 满子庆

北京大学

摘要

在当前大学教育环境下, 选课已成为每位学生必需的工作。尽管大多数高校都配备了先进的选课系统, 但它们通常缺乏针对学生个性化需求的课程推荐服务。以北京大学为例, 学生们往往需要依靠校内论坛或其他平台来搜索和评估特定课程, 以决定是否选择它们。针对这一现状, 我们设计了一款名为 PKourseUtopia 的个性化课程推荐系统, 该系统综合运用了上下文学习、多准则决策和主成分分析技术, 以满足用户的具体需求。此外, 我们还应用了 Kmeans 算法来对课程进行基于学生评价的分类。这一分类机制不仅增强了课程推荐的准确性, 而且为用户提供了与所推荐课程类似的其他选项。为了验证系统的有效性, 我们选取了北京大学 80 门通选课程作为测试样本。实验结果表明, PKourseUtopia 能够有效地根据用户的个性化需求提供推荐, 并且成功地从课程评价的角度对这些课程进行了全面的分类。

1. 问题背景

选课一直是大学生生活中的热门话题。以北京大学为例, 学生们通常通过以下几种方式获取选课信息:

- i) 在“北大树洞”“非官方课程测评”等网站上搜索特定课程;
- ii) 与同学交流, 分享彼此的课程体验;
- iii) 通过微信公众号等社交媒体平台阅读关于某门课程的推广文章。

尽管这些方法能够一定程度上提高选课的质量, 但它们各自存在局限性。例如, 特定课程的信息搜索既耗时又难以确保信息的准确性。更严重的是, 由于时间成本高昂, 学生们往往无法对所有感兴趣的课程进行深入搜索, 从而可能错过高质量的课程。此外, 通过同学和社交媒体获取的信息往往局限性较大。

实际上, 学生们常常面临这样的困惑: “我想找一门不用考试, 以论文结课, 且成绩给分宽松的通选课。”(图 1) 然而, 传统的信息获取渠道很难在短时间内提供准确的课程推荐。因此, 我们开发了名为 PKourseUtopia 的课程推荐系统, 该系统能够根据用户的个性化需求(包括是否有小组作业、是否有论文、是否有考试以及社交程度四项)快速提供课程建议。经实验验证, PKourseUtopia 推荐的课程集合 C 中的课程都能够满足用户的特定需求(如无小组作业、包含期末考试等), 且在“非官方课程评价”等网站上均获得了较优的评价。

2. 相关工作

为了实现 PKourseUtopia, 我们使用了一些已知的技术与算法, 这些技术与算法来自前人的相关工作:

上下文学习 (In-context Learning): 在自然语言处理领域, 上下文学习是指模型根据提供的文本上下文(如之前的对话内容、相关文档段落等)来理解和生成语言。这种学习方式的关键特征包括模型的自适应性, 能够根据上下文自动调整其回应, 以及零样本学习能力, 即在没有针对特定任务的训练样本的情况下执行任务。此外, 它还包括维持对话或文本序列的连续性, 以理解先前信息与当前输入之间的关联。注意到大语言模型超乎寻常的理解与推理能力, 我们运用大语言模型, 在具体实验中采用了 *GPT-turbo-0301*, 并通过上下文学习的方法, 指导它们通过阅读自然语言写就的文字评价, 生成该课程在是否有考试、是否有论文、是否有小组作业、课堂有趣程度以及社交程度五个维度上的得分(见附录 5.1)。

多准则决策 (Multi-Criteria Decision Making, MCDM): 多准则决策是一种分析方法, 用于在多个相互竞争的标准或属性之间做出选择。这种方法特别适合于那些涉及多个决策因素和目标的复杂决策问



图 1: 上半部分展示的是一类在树洞中经常可以看到的问题，下半部分是树洞中的一条课程评价，可以看到，这种类型的课程评价被非常多人关注（图中有 63 位关注者）。

题。在我们的应用中，为了得到用户的个性化需求，我们设计了两种不同的方案，这在第三部分“方法”中会有更详细的介绍。特别是在第一种方案中，我们使用了多准则决策：用户只需要在“小组作业”“考试”“论文”和“多社交”这些维度的两两比较中表达自己的倾向，从而帮助系统理解并满足他们的个人偏好和需求。这种方法不仅提高了决策的效率，也增强了决策的精确度和用户满意度。

主成分分析 (Principal Component Analysis, PCA): 主成分分析是一种统计方法，用于通过线性变换将一组可能相关的变量转换成一组线性无关的变量，这组新的变量被称为主成分。PCA 的主要目的是减少数据的维度，同时尽可能保留原始数据中的变异性。在对课程进行分类的任务中，我们运用了主成分分析，将原始的九维数据（前四维直接来自“非官方课程测评 @ 北京大学”，分别是“推荐程度”“课程内容”“工作量”和“考核”；后五维来自大预言模型的生成，分别是“有无小组作业”“有无考试”“有无论文”“课堂有趣程度”与“社交程度”）降维至二或三维。这样处理后的数据作为 K-means 聚类任务的基础，有助于更有效地识别和分类复杂的数据集，从而为用户提供更精确的课程推荐。

K-means 聚类算法 (K-means Clustering Algorithm): K-means 算法是一种常用的聚类算法，其核心思想是将数据点分组为 K 个集群，使得每个数据点都属于离它最近的均值（聚类中心）所代表的集群。这种方法以迭代的方式进行，旨在最小化每个点与其聚类中心之间的距离之和。在我们的应用中，我们使用了 K-means 聚类算法，利用主成分分析降维后的特征向量，实现将课程分类的目标。通过这种方式，我们能够有效地将课程分组到不同的类别中，根据它们在特征空间中的相似性，从而为用户提供更有针对性的课程推荐。

3. 方法

为实现 PKourseUtopia 的课程推荐功能，在实践中我们分上下文学习、数据处理、设计推荐系统与聚类系统四步完成。下面的部分将会对这四步分别进行方法层面的说明。更详细的代码解释可以参考附录 5.2

	A	B	C	D	E	F	G	H	I	J	K
1	大气概论	A	3.09090909	2.90909091	4.27272727	3.09090909	0	1	0	1.81818182	0.90909091
2	大学化学	A	4.68181818	4.68181818	3.81818182	4.90909091	1	1	0	3.13636364	1.61363636
3	地球环境与人类	F	3.03125	2.375	3.125	3.5	1	0	1	1.78125	2.21875
4	地史中的生命	A	4.1	4.1	4.1	3.9	0	1	0	3.3	2.1
5	地震概论	A	4.64150943	4.77358491	4.64150943	4.77358491	0	1	0	3.56603774	1.39622642

图 2: *data.xlsx* 前五行的情况，可以看到，每门课程生成了唯一的评分。

3.1 上下文学习

在本研究中，我们采用了上下文学习方法，利用先进的大型语言模型解析自然语言文本中的课程评价。具体地，模型根据给定的上下文指令与示例，自动生成课程的“有无小组作业”“有无论文”“课堂有趣程度”以及“社交程度”四个维度的分数。对于前两个维度，“1”表示“存在”，“0”表示“不存在”；而对于“课堂有趣程度”，评分从 1, 2, 3, 4, 5 中选取，其中“1”表示“极其无趣”，“5”表示“极其有趣”。同样的评分集合适用于“社交程度”，但在此维度中，“1”意味着“无需社交”，而“5”则意味着“社交活跃”。详细的上下文学习实例在附录 5.1 中有所展示，需要注意的是，附录中展示了更为先进的版本，实践中我们并未要求模型判断“是否开卷考试”，只需判断“是否考试”即可。

为了验证语言模型生成结果的可靠性，我们对其输出的结果进行了抽样检验，其中包括随机选取的 20 个实例进行人工评分，并将此评分与模型生成的结果对比分析。在诸如“是否有考试”这类二元问题上，模型展现出超过 80% 的准确度；而在“课堂有趣程度”和“社交程度”这两个评分维度上，准确度也超过了 50%。尽管在单个课程评价的打分中模型可能犯错，但由于每门课程都有多条评价，结合在“数据处理”小节中介绍的处理技术，最终能够确保每门课程的综合评分反映了真实情况，这一点是通过参考“北大树洞”和“非官方课程测评 @ 北京大学”上的用户评价得到的结论。

3.2 数据处理

在完成 3.1 之后，我们得到 *raw_data* 目录，目录下共有 80 个文件，均命名为 [课程名 *.xlsx*]，每个文件有若干行，每行代表对于该课程的评论；共 13 列，代表评论标题、评论正文、评论获得点赞的个数及 10 个特征指标。代码 *data_process.m* 对这 80 个文件进行了处理，生成最终的数据文件 *data.xlsx*，该文件共有 80 行 11 列，每列存放课程名及相应的 10 个特征指标。处理步骤如下：

1. 对于“是否有小组作业”“是否有考试”“是否有论文”这三个 0/1 二值的指标，由于可能有一部分评论没有提及相关信息，gpt 会将“有”判断为“无”，因此我们设定 0.4 至 0.6 之间的阈值，若判断为“有”的评论数占比超过阈值，则我们认为课程有该特征，将这一维设为 1，否则设为 0。

2. 对于其他 6 个以 1 至 5 打分的指标，以评论获得点赞的个数作为权重，将每条评论对这一指标的打分加权平均，得到这门课程在该指标上的最终得分。

raw_data 目录下的 80 个文件，在依次运行 *data_process.m* 后，得到 *data.xlsx*，图 2 中展示了前五行的每一列的含义依次是：课程名、课程类别、推荐程度评分、课程内容评分、工作量评分、考核评分、是否有小组作业、是否有考试、是否有论文、课堂有趣程度评分、社交程度评分。

此后的所有程序均以 *data.xlsx* 作为原始数据文件输入。

3.3 推荐系统

PKourseUtopia 开发并推出了两种创新的课程推荐策略，旨在满足用户个性化的学习偏好。

首先介绍的策略，详细说明于文件 *PKourseUtopia_Recommend_v1.m* 中，引导用户依据“小组作业”“考试”“论文”以及“积极社交”四个关键维度进行偏好比较，基于此建立多准则决策模型的初始决策矩阵。通过这一模型，我们能够精准地计算出针对每位用户的四个指标的标准化权重。利用这些权重，课程将按照维度得分经过标准化处理后进行排序，确保了各维度在综合评价中的均衡性。进一步地，选取权重较高的两个指标保留正值，而较低的两个指标取负值，这样更新后的权重与课程特征值相乘，计算得到每门课程的总评分。在此基础上，从 80 门课程中精选出 20 门得分最高的课程，这些课程

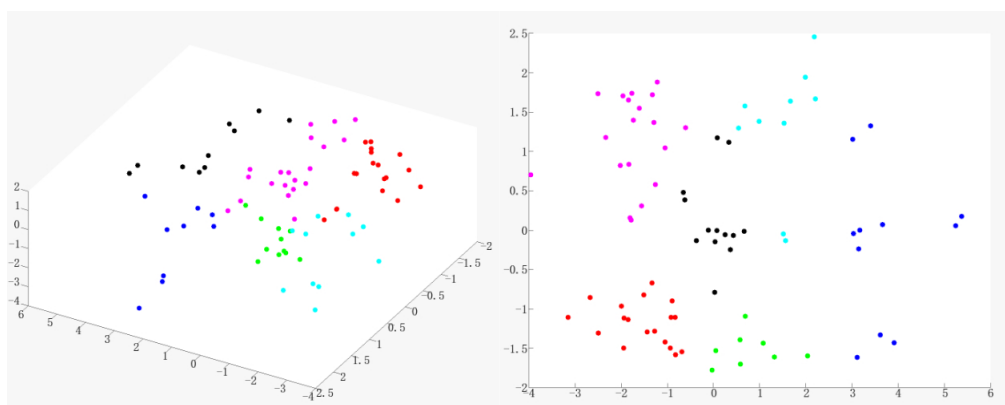


图 3: 左图为降至二维，聚 6 类的可视化结果；右图为降至三维，聚 6 类的可视化结果

最符合用户的独特偏好。最后，我们根据“推荐程度”“课程内容”“工作量”“考核”和“课堂有趣程度”五个维度的得分平均值，对这 20 门课程进行最终排序，向用户推荐排名前五的课程。

在第一种推荐方案中，我们还注重于课程评论的视觉呈现。这些评论摘自“非官方课程测评 @ 北京大学”，并以词云的形式展现，其中评论的字体大小与其获得的点赞数成正比，从而直观地显示了学生群体的共识和兴趣点。附录 5.3 中展示了此种推荐方案生成的结果与可视化词云。

接下来，第二种策略在 *PKourseUtopia_Recommend_v2.m* 文件中有所展示，这一策略更加简洁直接。在此策略中，用户可对全部九个维度进行详细打分，其中在“小组作业”“考试”和“论文”三个二元选择维度中加入了“无所谓”的选项。用户完成打分后，系统首先依据二元选择维度筛选出完全符合用户要求的课程集合 C_1 ；随后，在剩余六个维度上计算集合 C_1 中的课程与用户偏好之间的欧氏距离，并依此距离进行升序排列，从而推荐出最符合用户期望的前 15 门课程。

3.4 聚类系统

程序 *PKourseUtopia_kmeansCluster.m* 采用 kmeans 算法对 80 门课程做聚类。步骤如下：

1. 首先从原始数据中分别提取出课程名和其他九维数据备用。
2. 在拿到九维的数据后首先采用 PCA 对数据进行降维，提取出数据的主要成分以便后续的聚类操作与可视化，从最后的结果角度选择采用将原数据降为二维或三维进行后续操作，能够保留更多的信息的同时也有更好的可视化效果。其中需要注意在 PCA 降维前先将数据进行标准化，从而保证数据不同维度之间的影响大致相同。随后计算数据的协方差矩阵并降序排列特征值，并选择前二/三大的特征值对应的特征向量从而将原始的九维数据映射为二/三维。
3. 随后对降维后的三维数据进行 kmeans 聚类操作，根据对实验结果的不断调试，最终选择将 80 门课程总体聚为六类。kmeans 函数的原理是首先将 80 门课程看作 80 个向量，从 80 个向量中随机挑选 6 个作为初始聚类中心，然后开始迭代过程：对每个向量计算它离这六个中心的欧式距离，并找到离它最近的聚类中心，其归属于该中心。在计算过 80 个向量的类别之后，分别将六个聚类中心更新为其类别中的所有向量的平均值，从而得到六个新的聚类中心，然后重复上述迭代过程。对于迭代的停止条件分别使用了两种不同的算法判断。第一种是忍量判别法，其实现过程较为复杂，并且运行发现迭代次数基本达到所设上限仍不停止，但分类效果视觉上能够接受，从而保留代码部分但最终并不采用其进行判断。因此考虑另一种判断迭代停止的算法：当每个向量所属的聚类中心相比上次迭代的聚类中心不再变化时则停止迭代。实验发现这种迭代算法的迭代次数只需要 20 次不到即可得到很多的分类结果，运行速度比上种方法快很多且效果很好，因此最终采用该方法进行判断。
4. 在 kmeans 聚类结束后，输出 6 类分别包含的课程名，进而可以通过结果反映出哪些课程在考量的九个维度中比较相近。最后进行结果的可视化，将六类所包含的课程分别用不同的颜色展示在二/三维坐标系中并输出。同时程序也实现了将每一个课程的课程名展示在相应的坐标点旁的功能，但为了更清

晰的视觉效果，只展示单纯的聚类结果（图 3）。

4. 结果

利用上下文学习、多准则决策以及主成分分析等先进技术，我们开发了 PKourseUtopia 课程推荐系统。该系统在针对北京大学 80 门通选课的数据集上表现出色，准确地从自然语言的课程评价中提取出了量化的指标得分。利用这些得分，推荐系统可以迅速地对课程进行个性化筛选，为用户推荐符合其偏好的高质量课程；同时，聚类算法进一步对课程进行分组，以供建议与首选课程特性相似的其他选项。

PKourseUtopia 系统的优势在于其简洁有效的方法，它不仅实现了极具应用价值的推荐功能，而且其推荐结果具有可操作性。甚至在即将到来的下一学期选课阶段，都足以支持学生在决策过程中做出更加明智的选择。该系统的实用性和准确性预示着它在未来学术环境中的广泛应用潜力。

5. 附录

5.1 附录 1

Instruction:

给出一些关于课程的评价，请分别从这五个维度对课程进行打分，并且符合 demonstrations 中的格式：

是否有小组作业：1 代表有，0 代表没有；

考试：2 代表开卷考试，1 代表闭卷考试，0 代表不考试；

论文：1 代表需要写论文，0 代表不需要；

课堂有趣程度：打分范围 1, 2, 3, 4, 5，其中 5 代表非常有趣，1 代表完全无趣；

社交程度：打分范围 1, 2, 3, 4, 5，其中 5 代表非常需要社交，1 代表完全不需要社交；

Demonstration:

Demonstration 之间以连续的三个“#”作为分隔，一共两个；评分部分以 [] 作为标识

###

北大唯一真神！

课程听感：上课时建议刷树洞参与地概实时讨论，因为听了也白听。但是段子还可以听一下，如果有些段子你感觉比较冒犯可以选择性忽略。作业/任务量：两次线上的选择题，可以多次提交，刷到满分，一次小组报告，随便水水就行。关于考试：建议参考往年题，选择填空全是往年题。复习的时候可以树洞上求参考笔记，赵克常上课强调要考的地方一般会考（两个班不一样），最后两道计算题除了数据万年不变。谁选择了地概，谁就选择了成功！

[

评分：

是否有小组作业：1

考试：1

论文：0

课堂有趣程度：5

社交程度：2

]

解释：

评价中提到“一次小组报告”，可以判断该课程应当有小组作业；评论中“关于考试”部分未提及开卷，并且说明需要较多复习，可以判断应当为闭卷考试；评论中未提及论文相关内容，可以判断没有论文；评论中提到“段子还可以听一下”，说明课程中穿插一些有趣的幽默内容，因为课堂可能会非常有趣；评论中提到“小组报告”，可能会有一定的小组讨论，这应当需要一定的社交，但是“随便水水就行”，说明社交压力并不大。

###

.....

###

Prompt:

请分析以下课程评价，并对每段生成符合格式的评分，不需要进行解释：

.....

表 1: 通过 Instruction 与 Demonstration，完成上下文学习，使用大预言模型将自然语言的课程评价转换为五个维度的评分



图 4: 词云展示: 以“中国美术史”“现当代建筑赏析”为例

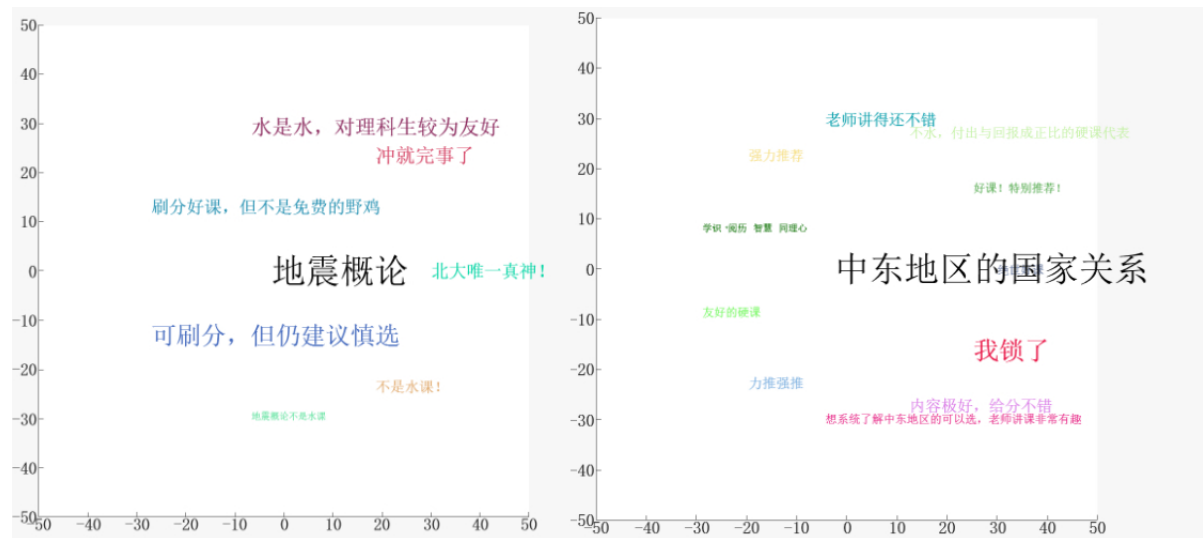


图 5: 词云展示: 以“地震概论”“中东地区的国家关系”为例

5.2 附录 2

全部的代码文件包括 *PKourse_Utopia_DataGenerate.py*, *PKourse_Utopia_DataProcess.m*, *PKourse_Utopia_KmeansCluster.m*, *PKourse_Utopia_Recommend_v1.m*, *PKourse_Utopia_Recommend_v2.m*, 还包括原始数据集合 *raw_data* 与经过处理后的数据 *data.xlsx*, 我们将在此部分提供尽可能详细的代码运行指导:

PKourse_Utopia_DataGenerate.py 是利用大语言模型生成原始数据的代码, 示例代码中以 *raw_data* 中的一个文件作为处理对象, 在实际运行时可以自由调整。也可以对代码进行简单的修改, 使之顺序处理 *raw_data* 中的每一个文件, 我们没有这样做的主要原因是希望对每一组运行结果有更好的控制。如果希望在其他课程目录上使用 *PKourseUtopia* 相同的处理方法, 可以直接调用 *PKourse_Utopia_DataGenerate.py* 以生成原始数据, 需要注意的是, 必须先将自然语言的课程评价以 *raw_data* 中的文件形式存储, 否则可能会遇到某些格式不匹配的问题。当然, 在最原始的情形下, *raw_data* 中每个文件的 *I* 列至 *M* 列均没有数值, 这些数值是 *PKourse_Utopia_DataGenerate.py* 生成并写入的。

PKourse_Utopia_DataGenerate.py 从 *raw_data* 生成 *data.xlsx*, 由于不同的自然语言评论获得了不同的点赞数, 我们将点赞数多少作为参考该评论的权重大小, 生成 *data.xlsx* 中二项性指标之外的内容。

三个二项性指标的生成不参考点赞数，而是对模型针对每条评论生成的结果进行直接平均，如果超过某个阈值，则认为该项指标为 1（即对应“有小组作业”“有考试”或“有论文”），这里的阈值可以灵活调整，这与模型的生成能力相关。

PKourse_Utopia_KmeansCluster.m 实现对于课程的聚类，这里我们在主成分分析后，保留了前两维进行聚类，并将全部课程分成 6 类。主成分的保留维度与聚类的类别数均是可以调整的参数，这里我们基本复用了课程代码。

PKourse_Utopia_Recommend_v1.m 首先初始化环境并从 *data.xlsx* 中读入课程数据，包括课程名称、得分和特征。这些信息分为两类：用于排序的得分 (*score*) 和用于多准则决策的特征 (*feature*)。脚本通过一系列用户输入收集对教学特征的偏好，以此构建一个反映用户偏好的矩阵。基于这个矩阵，脚本计算出每个特征的权重。然后，它使用加权和标准化的特征来计算每门课程的综合得分，并根据这些得分推荐符合用户偏好的顶级课程。此外，脚本还包括一个生成课程评论词云的功能，提供用户直观的课程评价概览，从而增强用户体验。

PKourse_Utopia_Recommend_v2.m 是另一个课程推荐机制，在这里用户可以在全部维度上设置自己喜欢的分数，然后直接产生对应的推荐列表。具体的实现过程在前文 (3.3) 中已经述及，此处不再赘述。

5.3 附录 3

以下展示 2 组不同的输入下，推荐算法生成的课程推荐结果。

在输入的 6 个判断分别是 $\frac{1}{5}, 3, 1, 5, 1, \frac{1}{5}$ 时，输出结果：“为您推荐的 4 门课程如下：现当代建筑赏析、化学与社会、中国美术史、环境科学导论”。并提供这 4 门课程的评价展示，图 4 展示了以“现当代建筑赏析”“中国美术史”为例的输出。

在输入的 6 个判断分别是 $7, 1, 1, \frac{1}{7}, \frac{1}{7}, 1$ 时，输出结果：“为您推荐的 4 门课程如下：中国经济思想史、浪漫主义时期的欧洲音乐、中东地区的国家关系、地震概论”。并提供这 4 门课程的评价展示，图 5 展示了以“地震概论”“中东地区的国家关系”为例的输出。

人为地分析两组输入的含义，可以看出用户在课程是否有小组作业、考试、论文及课程社交程度这四个特征上大致的偏好程度。与 *raw_data* 中课程的实际信息对比，发现推荐系统确实为用户提供了比较准确地符合用户偏好且评价较好的课程。