
机器学习实验报告

实验题目： 兴趣点类别预测

班 级： 2018 级人工智能（济南）

小组成员

周宇鹏（18AI）

田 婕（18AI）

桂孝强（18AI）

2020 年 02 月 20 日

目 录

引言.....	1
1. 问题分析.....	1
2. 实验方法.....	1
2.1 基于 WORD2VEC 和 SVM 分类器的求解方法.....	2
2.1.1 基于 word2vec 模型的数据表示.....	2
2.1.2 基于 SVM 模型的多分类学习.....	3
2.2 基于 word2vec 和 RF、神经网络和 KNN 的求解方法.....	3
2.3 基于 word2vec 和最近邻的求解方法.....	4
2.4 基于 BERT 结构修改得到的求解方法.....	4
2.5 基于马尔科夫假设统计类别共现的推测方法.....	5
3. 实验评估.....	6
3.1 数据集分析.....	6
3.2 数据集划分.....	8
3.3 模型选择.....	8
3.4 超参数选择.....	9
3.5 测试集预测.....	10
4. 团队分工.....	10
5. 参考文献.....	11

引言

机器学习实验二可以看作是多分类学习任务，本实验的数据集由 9548 条轨迹记录组成，每条轨迹由多个签到记录组成，每个签到记录由<位置 id#类别@时间戳>组。考虑到有 1270977 个的已有标记的位置以及 248377 个左右的未知标记的位置，已标记位置与未标记位置比例接近 4: 1，因此我们将数据集按 4: 1 的比例划分为训练集和验证集。多分类学习的任务是在训练集上训练得到性能比较好的分类模型，在验证集上对分类模型的参数进行测试及调整以获得最优性能，然后在测试集上预测结果。对于本实验，我们尝试使用多种方法解决，最终通过调试比较选择出了效果最优的方法。我们利用 word2vec 模型将每个位置 id 映射到欧氏空间中，得到处理后的数据集，然后采用 SVM 模型对数据集进行训练，并得到了性能最优的模型，最后我们在测试集上报告了我们的预测结果。

关键词：多分类学习，word2vec，SVM

1. 问题分析

本次实验为兴趣点类别预测。在 data1 中给定了用户的签到数据集，数据集由 9548 条轨迹记录组成，每条轨迹记录由多个签到记录组成，每个签到记录由<位置 id#类别@时间戳>组成，部分签到记录的类别未给出，标记为 Null。本次实验的任务为预测签到记录中类别未知的位置的类别，即预测 data3 中的所有位置的类别，要求使用的评测指标为 micro-F1 和 macro-F1。

经过小组内讨论分析，我们认为可以将该问题看作一个多分类问题，即在由已知类别的位置 id 组成的数据集上训练多分类模型，并预测未知类别的位置 id 的类别。

2. 实验方法

基于上述对问题的分析，我们在搜集相关资料后尝试了以下几种方法。我们设计的方法受到了自然语言处理算法的启发。这是因为轨迹数据和自然语言处理中的语句一样，都是一种序列数据，我们很自然的可以将其相关联，一条序列整

体表达了很多信息。我们推测将自然语言处理中的算法应用于轨迹信息的处理也能得到很好的效果。

2.1 基于 word2vec 和 SVM 分类器的求解方法

2.1.1 基于 word2vec 模型的数据表示

特征提取是机器学习任务的一个重要步骤，word2vec 广泛应用于自然语言处理的词嵌入任务中，将所有的词表示成低维稠密向量，从而可以在词向量空间上定性衡量词与词之间的相似性。因此，我们首先考虑了能否使用 word2vec 进行特征提取，然后使用多分类器进行分类。在 word2vec 生成的向量空间内保留了一定的相似性关系。我们推测能够通过这些向量来预测类别为 Null 的位置的类别。因此，我们采用 word2vec 模型[1]将原始数据集处理为可应用于多分类学习的数据集。



图 1 轨迹中按时间顺序排列的位置 id

我们可以将一条轨迹数据看成一条语句，组成轨迹的每个签到记录看作一个词语，因为签到记录是按照时间顺序排列的，因此语句中词语的上下文特征隐含表示了签到记录的时间特征。通过这样的处理，我们利用 word2vec 模型将签到记录中的位置 id 映射到欧氏空间中，得到了每个位置 id 的特征向量，然后，我们对特征向量做了归一化处理得到了特征数据集 X，将位置 id 的类别数值化为对应特征向量的类别得到了标记数据集 Y。




名称 ^	值
 test_data	2529x300 double
 train_data	10076x300 double
 train_target	10076x1 double

图 2 划分后的数据集

最后，我们将这些数据集保存为可以随时存取的 matlab 格式。根据我们的

任务，我们需要对列表标记为 Null 的类别进行预测，因此，这些类标记为 Null 的即为我们的测试集。我们将其他的数据按照 4: 1 的比例划分为训练集和验证集。

2.1.2 基于 SVM 模型的多分类学习

在将原始数据集处理为可用于多分类学习的数据集后，我们利用 SVM 模型对数据集进行了训练并得到了较好的效果。

首先，我们采用 RBF 核（径向基函数）的 SVM 分类器作为基分类器，RBF 核本质是在衡量样本和样本之间的“相似度”，在一个刻画“相似度”的空间中，让同类样本更好的聚在一起，进而线性可分。RBF 核形式如下式：

$$\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (1)$$

我们采取的多分类学习方法为一对一的学习方法，即针对任意两个类别都训练一个 SVM 分类器。对于本实验共有 103 个分类，因此需要设计 5253 个 SVM 分类器，最终采取投票的方式得到预测结果。

通过设计对比实验，我们发现基于 word2vec 和 SVM 的模型在我们提出的方法中具有最好的性能。我们基于调试验证的方法确定了 word2vec 的窗口大小和生成向量维数，采用网格寻优的方法寻找 RBF 核的最优参数 C 和 gamma，并报告了使用最优参数的五折交叉验证的模型性能。我们利用找到的最优参数，不再划分训练集和验证集，使用全部数据来训练 SVM 模型，并对测试集进行预测。

2.2 基于 word2vec 和 RF、神经网络、KNN 求解方法

在方法一的启发下，我们也对其他分类器进行了尝试，分别尝试了随机森林[5]、神经网络[6]、K 近邻作为分类器。

和方法一类似，我们首先从轨迹数据集中提取位置 id 序列，然后经过 Word2vec 处理位置 id 序列，得到的每个位置 id 对应的向量。之后我们分别使用随机森林分类器、神经网络分类器、K 近邻分类器进行训练和预测。同时，我们设计了实验，比较得出了不同方法之间的性能差异。

2.3 基于 word2vec 和最近邻的求解方法

本次实验类比产生词向量的思路，使用 word2vec 提取轨迹点特征向量。该方法与上述方法相比不同在于 word2vec 的输入，本方法将数据集中的一条轨迹看作一个句子，轨迹上的兴趣点看作分词，同时对数据集进行相应处理，将已知分类兴趣点用类别表示，未知分类兴趣点用位置 id 表示。

使用 gensim 中的 Word2Vec 模块进行实现。输入经过上述处理的数据，获得训练好的模型。通过使用 model.most_similar 方法获得与未知类别位置最相似的种类，但是发现大部分未知类别最相似的仍为未知类别，难以获得最终类别。

我们又考虑通过训练的得到的向量进行分类，通过计算未知类别位置对应的向量与各个类别的距离，该位置类别对应与距离最近的类别。最终得到结果中大部分被分类为‘Train Station’，准确率仅在 9%左右。推测原因为 data1 中有很多位置 id 对应于‘Train Station’，经过统计，累计共有 435777 个位置 id 对应于‘Train Station’，占总数的 34.3%；因此此种方法没有很好地解决问题。

2.4 基于 BERT 结构修改得到的求解方法

该方法由自然语言处理中的完形填空任务得到启发，因为轨迹中的 Null 类似于完形填空中的空格，联想到在 BERT 模型[7]预训练时通过预测 Mask 位置来学习参数，考虑使用对 BERT 模型进行修改，使得我们的模型可以对位置为 Null 的位置进行类别预测。

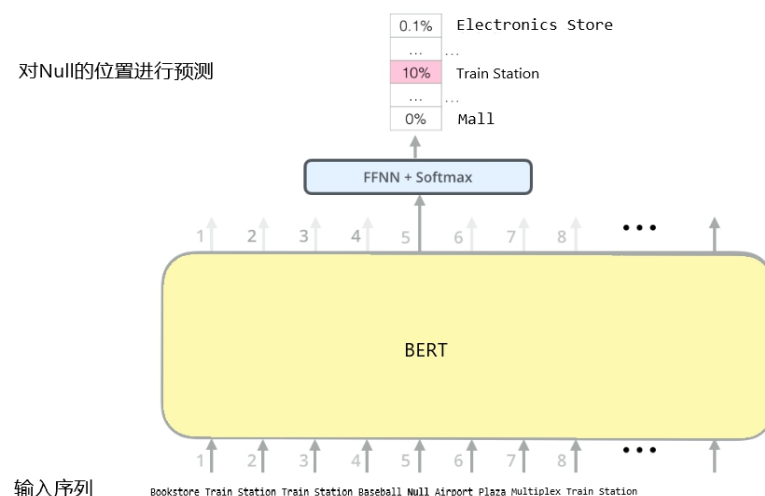


图 3 类似 BERT 结构的多分类模型示意图

在 BERT 和 Transformer[8]的基础上，我们实现了自己的模型，模型主要和 BERT 的区别如下：

我们的模型取消了 BERT 的 Segment Embedding，增加了类别嵌入和时间嵌入。类别嵌入考虑每个位置的类别，时间嵌入则考虑每个位置的时间并。通过对类别嵌入向量和时间嵌入向量进行向量加法，我们得到了我们模型的输入向量，模型主体由 8 个 Encoder 堆叠而成，每个 Encoder 由注意力模块和前馈神经网络组成，我们设计了残差连接和多头注意力机制，实现方法大致和 Transformer 相同。

同时，我们取消了 BERT 语言模型预训练中的 Next Sentence Prediction 任务，保留了 Mask 预测任务，对于训练集，我们首先将所有 Null 的数据直接从轨迹中移除，我们的训练方法是，对于一条输入的轨迹，我们随机的对某些位置类别为 Mask，其余位置类别不变，然后对 Mask 位置进行预测。使用负对数似然损失函数来计算损失并优化参数。使用模型进行预测时，我们先将类别为 Null 的位置标记为 Mask，再使用训练好的模型对 Mask 的位置进行预测，就可以得到预测的类别。

在对比实验中，该方法效果并不是非常理想，我们认为原因有如下三点：

第一点是因为类别过少，整个数据集一共只有 103 个类别，和自然语言处理中的词汇表相比差距悬殊，导致在自然语言处理中效果很好的方法在兴趣点类别预测中失效。

第二点是因为轨迹中的类别数量差距悬殊（如在所有轨迹节点中，类别为“Train Station”的节点占比高达 34.4%），相当于自然语言处理中的停用词，但却缺少有效的方法考虑。

第三点是由于基于序列的方法无法做到同时考虑多条序列。

2.5 基于马尔科夫假设统计类别共现的推测方法

通过统计类别之间的共现频率来推断 Null 所代表的类别也是一种思路，我们根据这个思路实现了类似于马尔科夫过程的方法，但我们的方法也和传统的马尔科夫过程有所区别，可以考虑位置的两侧。

在这里，基于统计语言模型和马尔科夫假设，我们给出了概率预测公式的定

义：

假定有一条轨迹 S ，中间的位置的类别为 C ，轨迹由一连串特定顺序排列的词 $\omega_1, \omega_2, \omega_3, \dots, \omega_n, C, \omega_{n+2}, \omega_{n+3}, \dots, \omega_{2n+1}$ 组成，轨迹的长度为 $2n + 1$ 。 $count(c_i)$ 表示所有 $\omega_1, \omega_2, \omega_3, \dots, \omega_n, c_i, \omega_{n+2}, \omega_{n+3}, \dots, \omega_{2n+1}$ 序列的个数。定义 C 的概率公式如下

$$P(\omega_1, \omega_2, \omega_3, \dots, \omega_n, c_i, \omega_{n+2}, \omega_{n+3}, \dots, \omega_{2n+1}) = \frac{count(c_i)}{\sum_n count(c_i)} \quad (2)$$

基于我们的公式，如要求得某个标记为 `Null` 的位置的概率，我们需要统计任意两个位置类别的共现概率。类似自然语言处理，我们可以推广语料库的概念，即经过处理的所有轨迹数据。我们基于训练集中的所有轨迹来制作我们的语料库。对于每一条轨迹，我们只提取位置类别，可以生成一条位置类别组成的序列，对于数据集中的所有轨迹，我们对对应生成序列，就可以得到语料库。得到了语料库以后，我们就可以基于语料库得出每个位置类别的共现频率。考虑到可能涉及多条语句的推断，且数据集无法涵盖所有情况，我们需要在计算共现概率的时候进行概率平滑，确保不会因为某条语句的某个类别概率等于 0 而造成的连乘概率失效。

预测时，对于每个位置类别为 `Null` 的数据，找到语料库中包含它的位置 `id` 的所有语句，对于每条轨迹，我们都可以使用上述方法，得到实际类别的概率向量 V 。我们对每条语句生成的概率向量进行标量乘法。可以得到最终的概率向量如图，将概率最大的类别作为预测位置类别。

统计共现概率时， n 是一个重要的超参数，我们经过测试， n 的大小可以取 2 或 3。在验证集下的准确率大概在 13% 左右。

3. 实验评估

3.1 数据集分析

为了更好更准确的预测数据并且能够全面的了解数据分布情况，我们对训练集的情况进行了一些可视化的分析。

在不考虑位置 id 的情况下，我们对数据中出现的类别进行统计，得到相关数据如下图：

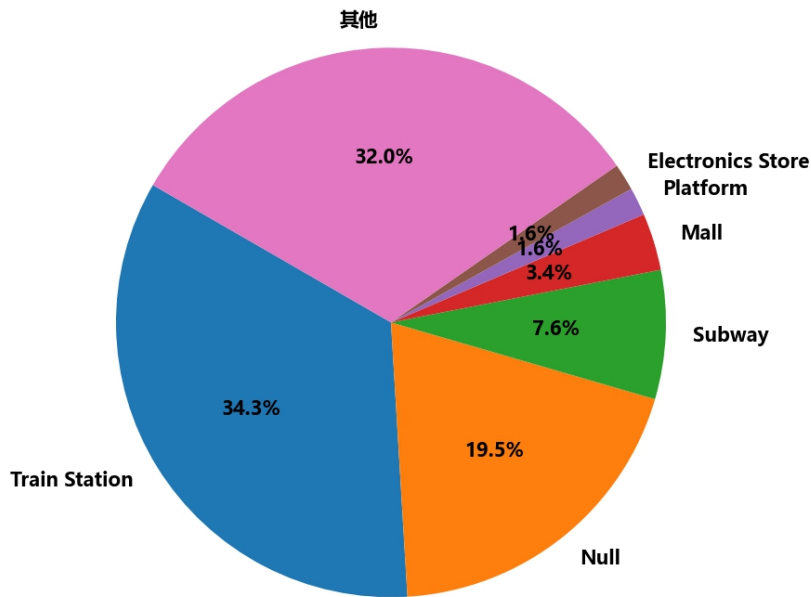


图 4 不考虑位置 id 时的数据分布

我们发现数据中 34.3% 的位置的类别都是 Train Station，Train Station 在训练集中所占的比重很大，会对实验结果产生一定程度的影响。

在考虑位置 id 的情况下，统计训练集中出现的位置的类别比例，统计图见下图：

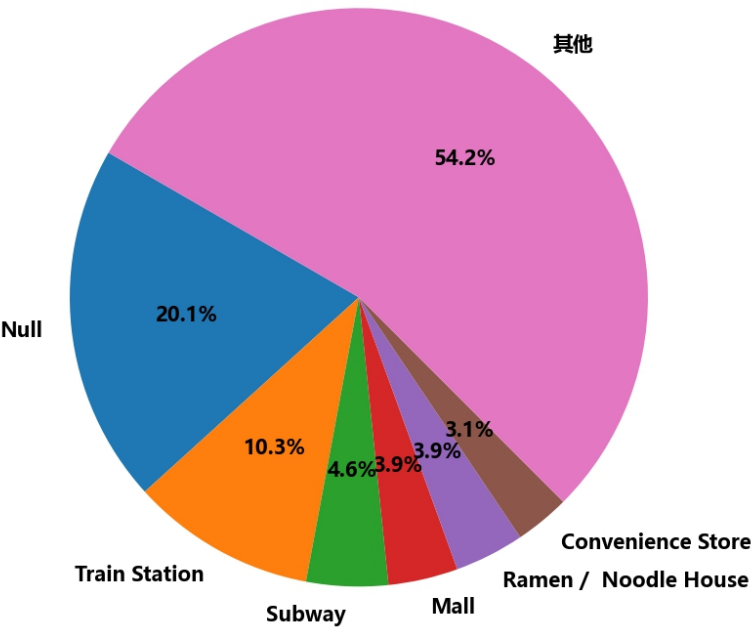


图 5 考虑位置 id 时的数据分布

经过对位置 id 进行分析, 整个数据集共有 12605 个位置 id, 其中 Null 为 2529 个, 占 20.1%。Train Station 和 Subway 等和交通相关的位置占比较高。

3.2 数据集划分

在设计模型时, 我们需要将数据集划分为训练集和验证集来对模型进行评估和优化。通过对数据集的分析可得, 在给定数据中共有 1270977 个的已有标记的位置和 248377 个左右的未知标记的位置, 已标记位置与未标记位置比例接近 4: 1, 因此我们认为将数据集按 4: 1 的比例划分为训练集和验证集的方法是比较合理的。

3.3 模型选择

我们的评估方法也沿用经典的分类器评估方式, 并且使用 k 折交叉评估, 因为 Null 的比例为 20.1%, 因此, 延续数据集划分时的思路, 我们认为使用五折交叉验证能比较好的仿真测试集。

由于精确的进行超参数选择需要消耗大量算力, 我们对与一些学习器的超参数设置是通过多次人工调试得来, 没有精确的进行超参数选择。

对于随机森林分类器, 我们设置的基学习器个数为 100, 决策树最大深度为 50, 使用基尼不纯度来计算信息增益。对于神经网络分类器, 我们设计的神经网络层数为 5 层, 使用 ReLU 作为激活函数。对于 k 近邻分类器, 我们设置 k 的大小为 5。

我们的实验评估结果见下表, 我们发现 word2vec+SVM 的组合无论是 micro-F1 和 macro-F1 都明显优于其他算法, 具有显著优势。因此, 在模型选择实验中, 我们最终选择 word2vec+SVM 模型, 并继续设计实验调整超参数, 充分发挥其性能。

表 1 不同算法的性能比较

算法	micro-F1	macro-F1
word2vec+SVM	33.0%	19.2%
word2vec+RF	23.9%	7.4%

word2vec+KNN	22.4%	10.0%
word2vec+MLP	26.3%	15.1%
word2vec+最近邻	9.2%	2.7%
修改的 BERT 结构	11.1%	0.15%
马尔科夫假设统计类别共现	10.2%	0.13%

各模型在经过进一步良好的超参数选择可以发挥出更好的性能。但在本实验中 SVM 表现性能相比其他模型有明显优势，因此这不会影响我们模型选择的正确性。

3.4 超参数选择

在模型选择实验中，我们发现了 word2vec+SVM 的算法组合在验证集上具有更高的性能效果，因此为了继续提高我们模型的性能，我们进一步对超参数进行了选择。其中，word2vec 的超参数主要包括 word2vec 的窗口大小、生成的向量长度；SVM 的超参数主要包括核函数的选择以及与核相关的参数。

对于 word2vec 的超参数的选择，我们通过多次实验调试发现当窗口大小设置为 10，生成的向量长度设置为 300 维时，生成的训练集的效果最理想。

表 2 word2vec 的参数选择

参数名称	参数值	解释
sg	1	sg=1 对应 skip-gram 算法
size	300	神经网络层数
window	10	窗口大小
min_count	1	频率过滤
negative	5	噪声个数
hs	1	softmax 层将会被使用

对于 SVM，我们选择了 RBF 核作为 SVM 分类器的核函数。下一步我们需要确定合适的 γ 和 C 的取值。我们设置了一组实验，通过设置一定的范围，按照一定的步长去选择超参数，对于选择的每组超参数 γ 和 C，我们使用五折交叉验证进行评估。我们计划通过这组实验，来选择出较好的超参数，从而训练出更好的模型。最终五折交叉验证中表现最好的超参数如下表。

表 3 SVM 的参数选择

参数名称	参数值
γ	27.8576
C	0.015625

3.5 测试集预测

通过模型选择，我们最终选择 word2vec+SVM 方法，并按照表 2 表 3 设置模型的超参数。此外，我们不再把数据集划分为训练集和验证集，而是把整个数据集作为训练集，对模型进行训练。然后，我们使用训练好的模型对类别标记为 Null 的数据进行预测，并将预测结果按照要求输出到文件。

4. 团队分工

团队成员在 1 月 13 日进行了第一次线上讨论，明确了大体的思路 and 方向，各自搜集相关资料对思路进行进一步的补充和完善。一周后进行了第二次线上讨论，基本明确了使用 word2vec 进行特征提取的思路。之后大家根据这一思路分头进行代码实现，并时常进行线上的讨论和交流。周宇鹏同学还尝试使用了 BERT 和马尔科夫假设统计类别共现的方法解决问题。最终我们在多种方案中选择了效果最优的方法，并完成实验报告的撰写。

表 4 小组成员工作表（排名不分前后）

姓名	主要工作
周宇鹏	word2vec+SVM、word2vec+随机森林、word2vec+神经网络、BERT、马尔科夫过程、超参数选择实验、撰写报告
田婕	word2vec+SVM、word2vec+最近邻、撰写报告、排版
桂孝强	数据集处理、word2vec+SVM、word2vec+k 近邻、撰写报告

5. 参考文献

- [1] Mikolov, T. , Chen, K. , Corrado, G. , & Dean, J. . (2013). Efficient Estimation of Word Representations in Vector Space. abs/1301.3781.
- [2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Computer Science, 1-12.
- [3] Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the context of locations for personalized location recommendations. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16). AAAI Press, 1188–1194.
- [4] Chen, Meng, Xiaohui Yu, and Yang Liu. "TraLFM: Latent factor modeling of traffic trajectory data." IEEE Transactions on Intelligent Transportation Systems 20.12 (2019): 4624-4634
- [5] Breiman, and Leo. "Bagging Predictors." Machine Learning 24.2(1996):123-140.
- [6] Rumelhart, D. E. , G. E. Hinton , and R. J. Williams . "Learning representations by back propagating errors. Cogn." Nature 5(1986).
- [7] Devlin, Jacob , et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." (2018).
- [8] Vaswani, Ashish , et al. "Attention Is All You Need." *arXiv* (2017).