Report of Deep Learning for Natural Langauge Processing

基于 LDA 模型的文本主题提取与分类

韩子轩
18210237202@163.com

Abstract

本实验基于 LDA 主题模型和 SVM 分类器实现了文本主题提取与分类。实验以金庸小说集作为语料库，将文本划分成段落，并进行采样。使用 KFold 方法划分训练集和测试集，训练 LDA 模型并使用 SVM 进行分类器训练和测试。实验探究了不同主题个数 T、基本单元选定“字”或“词”以及不同取值长度 K 的短文本和长文本对上述方法分类性能的影响。结果表明，分类准确率与 T、K 正相关，且以“字”为基本单元时分类性能普遍高于“词”为基本单元。

Introduction

从金庸小说集中均匀抽取 1000 个段落作为数据集（每个段落可以有 K 个 token，K 可以取 20，100，500，1000，3000），每个段落的标签就是对应段落所属的小说。利用 LDA 模型在给定的语料库上进行文本建模，主题数量为 T，并把每个段落表示为主题分布后进行分类（分类器自由选择），分类结果使用 10 次交叉验证（i.e. 900 做训练，剩余 100 做测试循环十次）。实现和讨论如下的方面：（1）在设定不同的主题个数 T 的情况下，分类性能是否有变化？；（2）以"词"和以"字"为基本单元下分类结果有什么差异？（3）不同的取值的 K 的短文本和长文本，主题模型性能上是否有差异？

Methodology

M1: LDA 模型

LDA（Latent Dirichlet Allocation），是一种文档主题生成模型，它可以将文档中每篇文档的主题按照概率分布的形式给出。也称为一个三层贝叶斯概率模型，包含词、主题和文档三层结构。所谓生成模型，就是说，我们认为一篇文章的每个词都是通过“以一定概率选

择了某个主题，并从这个主题中以一定概率选择某个词语”这样一个过程得到。文档到主题服从多项式分布，主题到词服从多项式分布。

LDA 是一种非监督机器学习技术，可以用来识别大规模文档集（document collection）或语料库（corpus）中潜藏的主题信息。它采用了词袋（bag of words）的方法，这种方法将每一篇文档视为一个词频向量，从而将文本信息转化为了易于建模的数字信息。但是词袋方法没有考虑词与词之间的顺序，这简化了问题的复杂性，同时也为模型的改进提供了契机。每一篇文档代表了一些主题所构成的一个概率分布，而每一个主题又代表了很多单词所构成的一个概率分布。

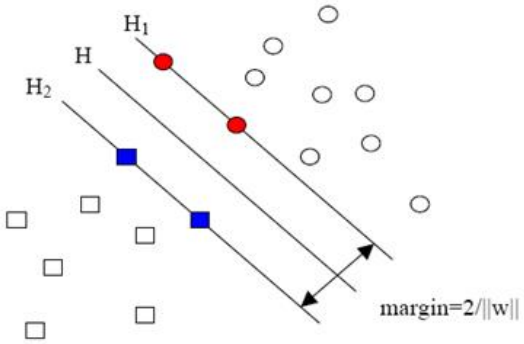
LDA 的核心思想是寻找到最佳的投影方法，将高维的样本投影到特征空间(feature space)，使得不同类别间的数据“距离”最大，而同一类别内的数据“距离”最小。现在有一组文档，希望通过 LDA 算法，用 K 个主题来表示每一个文档，以下为其中一种做法，被称为折叠吉布斯采样（collapsed Gibbs sampling）：

- 第一步，遍历每个文档，并随机的给每个文档中的每个词分配 K 个主题中的一个。
- 第二步，遍历每个文档 d：
  - 遍历当前文档 d 的所有单词 w：
    - 对每个主题 t，计算：
      - $p(topic\ t \mid document\ d)$，即当前文档 d 中，被赋给主题 t 的单词占总单词的比例
      - $p(word\ w \mid topic\ t)$，即单词 w 被赋给主题 t 占所有文档中主题 t 出现个数的比例
      - $p(topic\ t \mid document\ d) \cdot p(word\ w \mid topic\ t)$，用这个概率给单词 w 重新分配一个主题，最直观的想法就是取最大上述概率的主题

完成上述循环就能得到收敛的结果。

M2: SVM 分类器

支持向量机（Support Vector Machine，SVM）是一种经典的机器学习方法，广泛应用于分类和回归分析中。其基本思想是找到一个最优的超平面$wx + b = 0$，将不同类别的样本点分开，并且使得最靠近这个超平面的样本点到该超平面的距离最大化。这些最靠近超平面的样本点被称为支持向量，因为它们对于定义超平面起着决定性作用。



对于线性可分的情况，SVM 的目标是找到一个超平面，使得所有正样本点在超平面一侧，负样本点在另一侧，且使得支持向量到超平面的距离之和最大。对于线性不可分的情况，可以使用核技巧（kernel trick）将输入空间映射到高维特征空间，使得在高维特征空间中样本线性可分。常用的核函数有线性核、多项式核、高斯核等。分割超平面表示为

$$f(x) = w\phi(x) + b$$

SVM 的训练过程可以看作是一个凸优化问题，其目标是最大化间隔同时使得分类误差最小化。这个优化问题可以通过拉格朗日对偶性转化为一个对偶优化问题，从而可以使用凸优化方法进行求解。

$$\min_{\lambda}[\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j(\phi(x_i)\cdot\phi(x_j)) - \sum_{j=1}^{n}\lambda_i]$$

$$s.t. \quad \sum_{i=1}^{n}\lambda_i y_i = 0, \quad \lambda_i \geq 0, \quad C - \lambda_i - \mu_i = 0$$

Experimental Studies

(1) 初始化实验参数：数据集段落数量为 1000，划分模式为“字”和“词”，段落长度$P \in \{20,100,500,1000\}$，主题数量$T \in \{2,5,10,20,50,100\}$，交叉验证组数 CV = 10。

(2) 数据集加载：以金庸 16 部小说集为语料库，遍历语料库文件夹中的文件，逐个读取文件内容，去除文本中的特定冗余字符串。若采用“词”划分，则使用 jieba 分词库对文本进行分词。读取各部小说，将小说按每 P 个 Token 划分为多个段落。按照每个文档的长度，确定每个文件中抽取的段落数量，利用 np.random.choice 方法从每个文件的段落中随机选取一定数量的段落索引，并将选取的段落以及对应的小说名构建成 LabeledDoc 对象，存入 self.sampledData 列表中。本作业采用 10Fold 交叉验证方法，因此 Set 将被平均分为 10 折，其中 9 折作为训练集 TrainSet， 1 折作为测试集 TestSet。

(3) 训练 LDA 模型：基于 gensim 库所提供的 LDA 模型 API，首先将训练数据中的文档

数据提取出来，然后使用 corpora.Dictionary 创建词典。接着，将文档数据转换为词袋表示形式，最后调用 models.LdaModel 训练得到 LDA 模型，并返回模型、词典和词袋列表。

(4) 训练 LDA 模型：首先，从训练数据中提取标签。然后，利用 LDA 模型提取文档的主题分布，并将主题分布作为特征输入 SVM 模型进行训练。最后，计算训练精度并返回训练好的模型和精度。

(5) 分类验证：首先，从测试集中提取文档数据和标签。然后，使用训练好的 LDA 模型将文档数据转换为主题分布，并将主题分布作为特征输入训练好的 SVM 模型进行分类。最后，计算测试精度并返回。

本实验采用 10Fold 交叉验证方法，因此(2)中训练集与测试集的划分将轮换 10 组，重复(2)-(5)步骤，计算平均准确率作为最终结果。

Table 1: “字”基本单元测试准确度

	20	100	500	1000
2	0.14	0.159	0.215	0.209
5	0.156	0.184	0.332	0.425
10	0.152	0.212	0.427	0.49
20	0.156	0.225	0.47	0.586
50	0.16	0.239	0.525	0.613
100	0.191	0.258	0.449	0.605

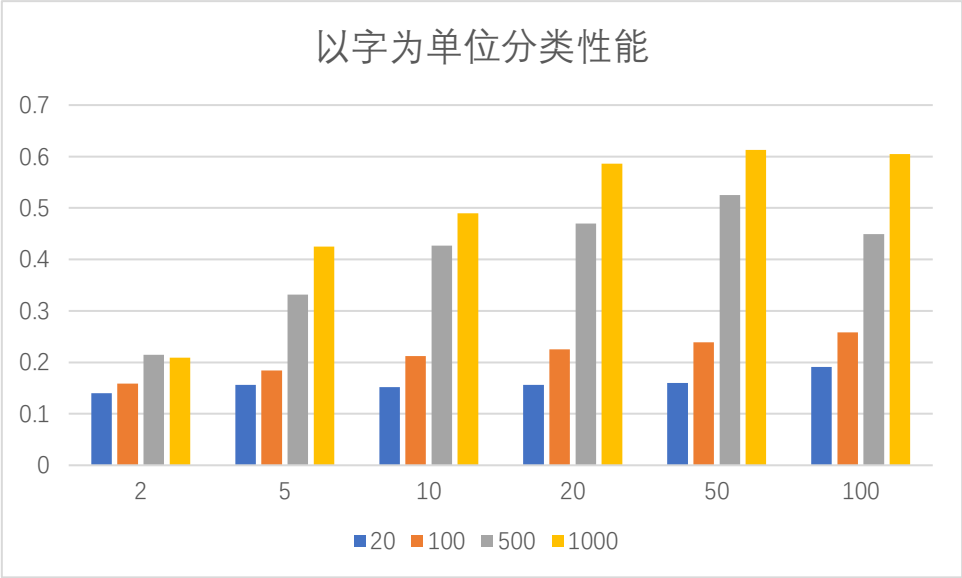
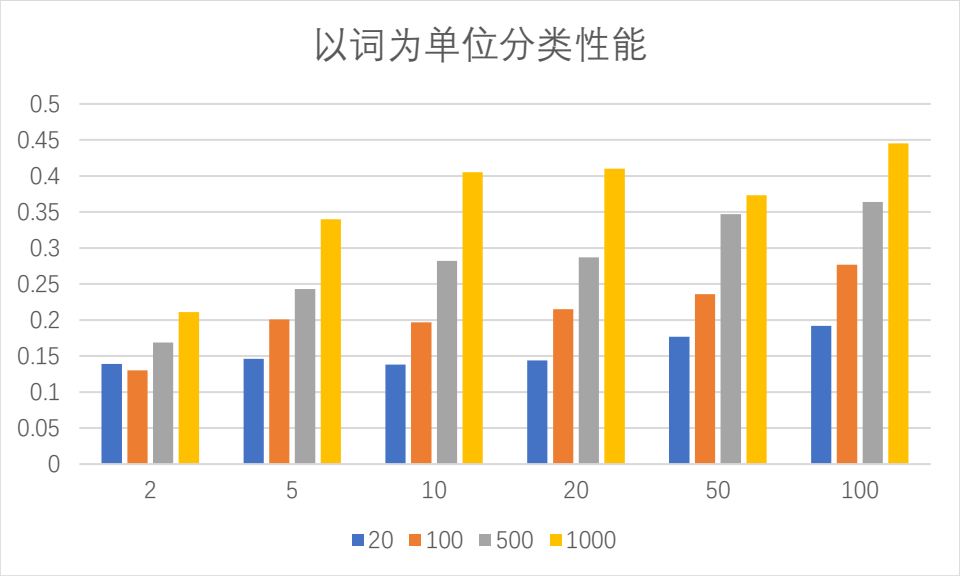


Table 2: “词”基本单元测试准确度

	20	100	500	1000
2	0.139	0.13	0.169	0.211

5	0.146	0.201	0.243	0.34
10	0.138	0.197	0.282	0.405
20	0.144	0.215	0.287	0.41
50	0.177	0.236	0.347	0.373
100	0.192	0.277	0.364	0.445



Conclusions

1. 随着主题数 T 增加, 分类准确率随之上升。更多主题数提供了更多的特征信息用于分类, 并且增加了模型的泛化能力。但主题数量的增加也会增加模型的复杂度和计算成本。
2. 对比“字”和“词”作为基本单元的分类结果，可以发现“字”单元的分类性能整体显著高于“词”的分类性能性能。这表明在所选语料库中, 不同小说间用“字”的主题差异大于用“词”的主题差异。这可能时因为字的语义更加明确，能够更准确地捕捉语义信息、语境相关性，并降低数据稀疏性和歧义性带来的影响。
3. 随着 token 增加，分类准确率显著上升。