Report II of Deep Learning and Natural Langauge Processing

牛华坤
ZY2303315

Abstract

这份报告主要包括两部分内容。第一部分：从金庸小说语料库中抽取段落，并对其进行 LDA 文本建模；第二部分，根据段落的主题分布，使用随机森林方法进行文本分类。在此过程中讨论了不同主题个数下分类性能的变化、“词”与“字”分类结果的差异和不同取值的 K 导致的主题模型的性能差异。

Introduction

I1: 文本建模

报告从给定的金庸小说语料库中均匀抽取 1000 个段落作为数据集（每个段落有 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，并能根据模型的困惑度和一致性区分 LDA 模型的性能差异。

I2: 文本分类

报告将每个段落表示为主题分布后进行分类，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），分类方法采用随机森林法，根据最后的平均准确率作为分类性能的衡量标准。

Methodology

M1: LDA 模型概述

LDA（Latent Dirichlet Allocation）模型是一种用于文本分析的概率模型，它最早由 Blei 等人在 2003 年提出，旨在通过对文本数据进行分析，自动发现其隐藏的主题结构。被广泛应用于文本挖掘、信息检索、自然语言处理等领域。

LDA 模型的核心思想是将文本表示为一组概率分布，其中每个文档由多个主题混合而成，每个主题又由多个单词组成。LDA 模型的基本原理是先假设一个文本集合的生成过程

为：首先从主题分布中随机选择一个主题，然后从该主题的单词分布中随机选择一个单词，重复上述过程，直到生成整个文本。具体来说，LDA 模型的生成过程包括以下三个步骤：

1）对一篇文档的每个位置，从主题分布中抽取一个主题；

2）从上述被抽到的主题所对应的单词分布中抽取一个单词；

3）重复上述过程直至遍历文档中的每一个单词。

M2: LDA 模型生成

首先定义文档（doc）集合为 Doc，文章主题（topic）集合为 Topic，Doc 中的每个文档 doc 可以看做一个单词序列$< w_1, w_2, \ldots, w_n >$，其中$w_i$表示第 i 个单词，假设一篇文档 doc 共有 n 个单词。

Doc 中的所有不同单词组成一个集合 Voc，LDA 模型以文档集合 Doc 作为输入，最终训练出两个结果向量，k 表示 Topic 中所有主题的数量，m 表示 Voc 中所有词的数量。

对每个文档 doc，对应到不同 topic 的概率$\theta_d =< p_{t1}, p_{t2}, ..., p_{tk} >$，其中$p_{ti}$表示 doc 对应 Topic 中第 i 个 topic 中词的概率，计算方法如下：

$$p_{ti} = \frac{n_{ti}}{n}$$

n_{ti}表示 doc 中对应第 *i* 个 topic 的词的数目，*n* 表示 doc 中所有词的总数。

对 Topic 中的第 *t* 个 topic，生成不同单词的概率$\varphi_t =< p_{w1}, p_{w2}, \cdots, p_{wm} >$，其中，$p_{wi}$表示第 *t* 个 topic 生成 Voc 中第 *i* 个单词的概率。计算方法如下：

$$p_{wi} = \frac{N_{wi}}{N}$$

其中N_{wi}表示对应到第 *t* 个 topic 的 Voc 中第 *i* 个单词的数目，*N* 表示所有对应到第 *t* 个 topic 的单词总数。

LDA 的核心公式如下：

$$p(w|d) = p(w|t)p(t|d)$$

直观的看这个公式，就是以 Topic 作为中间层，可以通过当前的θ_d和φ_t给出了文档 doc 中出现单词 *w* 的概率。其中$p(t|d)$利用θ_d计算得到，$p(w|t)$利用φ_t计算得到。实际上，利用当前的θ_d和φ_t，我们可以为一个文档中的一个单词计算它对应任意一个 topic 时的$p(w|d)$，然后根据这些结果来更新这个词应该对应的 topic。然后，如果这个更新改变了这个单词所对应的 topic，就会反过来影响θ_d和φ_t。

下面给出一种 LDA 的学习过程：

LDA 算法开始时，先随机地给θ_d和φ_t赋值（对所有的 doc 和 t）。然后上述过程不断

重复，最终收敛到的结果就是 LDA 的输出。再详细说一下这个迭代的学习过程：

1）针对一个特定的文档 d_s 中的第 i 个单词 w_i，如果令该单词对应的 topic 为 t_j，可以把上述公式改写为：

$$p_j(w_i \mid d_s) = p(w_i \mid t_j)p(t_j \mid d_s)$$

2）现在我们可以枚举 Topic 中的 topic，得到所有的 $p_j(w_i \mid d_s)$，其中 $j = 1,...,k$。然后可以根据这些概率值结果为 d_s 中的第 i 个单词 w_i 选择一个 topic。最简单的想法是取令 $p_j(w_i \mid d_s)$ 最大的 t_j（注意，这个式子里只有 j 是变量），即 $\arg\max_j p_j(w_i \mid d_s)$。

3）然后，如果 d_s 中的第 i 个单词 w_i 在这里选择了一个与原先不同的 topic，就会对 θ_d 和 φ_t 有影响了（根据前面提到过的这两个向量的计算公式可以很容易知道）。它们的影响又会反过来影响对上面提到的 $p(w \mid d)$ 的计算。对 Doc 中所有的 doc 中的所有 w 进行一次 $p(w \mid d)$ 的计算并重新选择 topic 看作一次迭代。这样进行 n 次循环迭代之后，就会收敛到 LDA 所需要的结果了。

M3：随机森林算法

随机森林（Random Forest）算法是对 Bagging 算法进行了改进。

首先，RF 使用了 CART 决策树作为弱学习器，这让我们想到梯度提升树 GBDT。

第二，在使用决策树的基础上，RF 对决策树的建立做了改进，对于普通的决策树，我们会在节点上所有的 n 个样本特征中选择一个最优的特征来做决策树的左右子树划分，但是 RF 通过的随机选择节点上的一部分样本特征，这个数字小于 n，假设为 n_{sub}，然后在这些随机选择的 n_{sub}（小于 n）个样本特征中，选择一个最优的特征来做决策树的左右子树划分。这样进一步增强了模型的泛化能力。

RF 的主要优点是训练可以高度并行化，对于大数据时代的大样本训练速度有优势，适应特征维度很高的样本，可以给出各个特征对于输出的重要性等；RF 的主要缺点是在某些噪音比较大的样本集上，RF 模型容易陷入过拟合，取值划分比较多的特征容易对 RF 的决策产生更大的影响。

Experimental Studies

E1：文本预处理

对于金庸小说构成的语料库，需要对读取的文本进行预处理：

1）删除开头无用信息。

2）删除中文停词及标点符号（利用 cn_stopwords.txt 文件）。

3）删除空白符号，比如空格、换行符等。

4）根据每篇小说在语料库中的字词比例决定抽取的段落数，并根据抽取的段落数决定抽取的开始位置以对语料库进行均匀抽取。

5）将抽取结果保存为 csv 文件。

白马啸西风 ： 8
碧血剑 ： 58
飞狐外传 ： 52
连城诀 ： 26
鹿鼎记 ： 138
三十三剑客图 ： 8
射雕英雄传 ： 108
神雕侠侣 ： 114
书剑恩仇录 ： 62
天龙八部 ： 139
侠客行 ： 41
笑傲江湖 ： 111
雪山飞狐 ： 15
倚天屠龙记 ： 114
鸳鸯刀 ： 4
越女剑 ： 2

Figure 1：以字划分文档时的段落抽取数量

白马啸西风 ： 8
碧血剑 ： 57
飞狐外传 ： 52
连城诀 ： 26
鹿鼎记 ： 139
三十三剑客图 ： 8
射雕英雄传 ： 107
神雕侠侣 ： 117
书剑恩仇录 ： 61
天龙八部 ： 139
侠客行 ： 42
笑傲江湖 ： 111
雪山飞狐 ： 16
倚天屠龙记 ： 113
鸳鸯刀 ： 4
越女剑 ： 2

Figure 2：以词划分文档时的段落抽取数量

E2: LDA 文本建模

将得到的语料库数据在 LDA 模型中进行建模，并进行模型评估：

1）读取抽取数据并构建词典。

2）将词典中的数据取出并表示为向量化形式。

3）进行 LDA 模型训练。

4）评估 LDA 模型性能。

5）保存 LDA 模型。

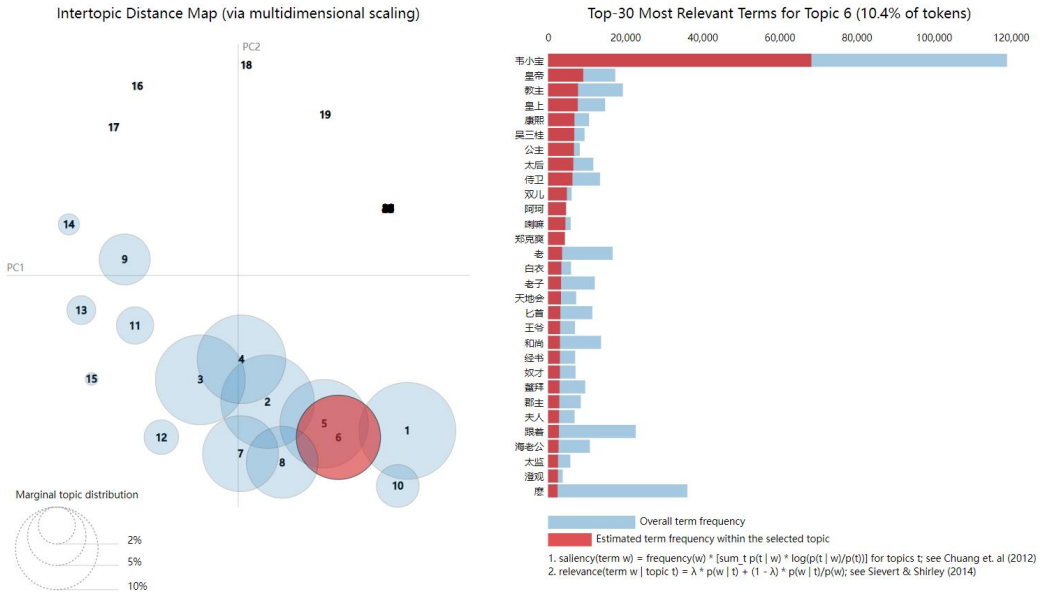

Figure 3：K=500,T=50 时以词为基本单元的模型评估

E3：随机森林文本分类

利用随机森林方法对标记好的段落进行分类：

1）取出每个段落的主题分布。

2）取出段落一一对应的标签。

3）按照十折交叉验证法划分数据集。

4）训练随机森林模型

5）交叉验证后得到平均准确率。

最终得到的模型评价结果如下表所示：

Table 1 模型评价指标

基本单元	K	T	困惑度	一致性	准确率
字	20	50	-6.013	0.201	0.286
	100	50	-6.410	0.253	0.448
	500	50	-6.792	0.325	0.759
	1000	50	-6.846	0.386	0.821
	3000	50	-7.116	0.441	0.862
词	20	50	-6.990	0.352	0.343
	100	50	-7.602	0.396	0.492
	500	50	-8.799	0.431	0.785
	1000	50	-9.146	0.474	0.869
	3000	50	-9.535	0.553	0.920
	500	16	-8.563	0.391	0.748
	500	100	-8.904	0.464	0.774

Conclusions

C1: LDA 主题模型性能

从 Table1 中可以看出，随着 K 值的增加，字和词为基本单元的主题模型的性能呈现出相似的趋势，困惑度变小，一致性变大，模型的性能更好，因而其分类结果的准确率也随之增加，这说明长文本的相较短文本蕴含的信息量更大，符合直观认知，使用长文本能更有效提取出文本主题。

C2: 随机森林模型分类性能

从 Table1 中可以看出，对于 K 固定到 500 时，T 分别取 16、50、100，准确率先上升后下降，分类模型的性能先提高后降低，显然主题数过小时，难以衡量不同主题间的差异，主题数过大又过于分散，因而需要取合适的主题数。

从 Table1 中可以看出，在相同的 K 和 T 下，以词为基本单元的分类结果准确率明显优于以字为基本单元的分类结果准确率，这是由于词的信息熵更大，能更好体现不同篇小说的特点，符合直观认知。

References

[1] https://zhuanlan.zhihu.com/p/658568949