关键词分类——算法说明文档

参赛id：stdcoutzyx

针对该问题，本小组选择支持向量机（SVM）作为分类算法，最终形成的算法方案流程图如下：



图 1 最终算法方案流程图

在本方案中，仅仅使用了keyword\_class.txt一个源数据文件。

首先，对keyword\_class.txt进行中文分词，本方案使用了IKAnalyzer中文分词程序包[[1]](#footnote-1)。该分词程序包有支持两种分类模式，智能分词和细粒度分词，实验知细粒度分词效果较好，为本方案所采用。

分词完成后，对分词后的数据进行向量化，使之符合SVM程序包的输入标准，本方案使用台湾国立大学机器学习小组开发的liblinear[[2]](#footnote-2)程序包。经过向量化后将数据分为训练集和测试集。

向量化完成后，使用liblinear程序包对训练集进行训练。由于hadoop集群每个job的内存有限，因为不能一次性的将所有训练集读入内存进行训练得到模型。考虑到要将关键词分为33个类，因而如果使用liblinear训练一对一分类模型，将会有33(33-1)/2=528个分类器，运行时间过长且使得中间某个job失败的概率变大。又因为liblinear内部已经支持了多类分类，因而可以将多个类组成一组，训练一组对一组的分类器即可。

在本方案的实验过程中，使用按顺序三类一组的模式（即1,2,3一组，4,5,6一组，以此类推），训练11(11-1)/2=55个分类器时，达到排行榜上本组的最高准确率97.65%。但这样的话，程序运行时间仍然过高（130mins），使用按顺序四类一组的模式（即1,2,3,4一组，5,6,7,8一组，以此类推，多余的33类放在条目最少的13,14,15,16组内），这样训练时间降低为70mins，分类准确率降至97.60%。

训练完成后，使用得到的每个模型，对测试集中的数据进行预测，每个模型对应一个结果。实际上，针对每个模型的分类和预测其实在一个map-reduce过程中完成，在setup中读入训练集数据，训练模型；在map中使用模型对测试集数据进行分类。

得到多个结果文件后，就要对结果进行合并。由于每个模型都会对所有数据预测一遍，因而每个未标记的词语会得到多个预测值，在合并过程中，使用投票法获得最终预测值，即选择出现最多的预测值作为最终预测值。这样，就得到了最终的结果预测文件。

如此，就得到了最终结果文件。最终使用四类一组的模式得到的结果文件准确率为97.60%。

1. IKAnalyzer官网：<https://code.google.com/p/ik-analyzer/> [↑](#footnote-ref-1)
2. Liblinear官网：<http://www.csie.ntu.edu.tw/~cjlin/liblinear/> [↑](#footnote-ref-2)