

乖兔子的专栏

不积跬步，无以至千里！

目次视图

摘要视图

RSS 订阅

个人资料



蜡笔大龙猫

访问: 398603次

积分: 4069

等级:

排名: 第6728名

原创: 111篇 转载: 13篇

译文: 1篇 评论: 109条

文章搜索

文章分类

[机器学习](#) (11)

[主题模型](#) (9)

[推荐系统](#) (2)

[深度学习](#) (11)

[spark](#) (2)

[python](#) (9)

[C/C++](#) (14)

[Java](#) (0)

[设计模式](#) (2)

[读书笔记](#) (2)

[工具](#) (1)

[数据结构与算法](#) (6)

[图像处理](#) (3)

[笔试面试](#) (9)

[月赛BOJ](#) (2)

[程序设计导引及在线实践](#) (49)

【评论送书】机器学习、Spring MVC、Android CSDN日报20170509 ——《互联网时代架构师的职责与思考》 CSDN技术直播: php实战微信公众号开发!

CSDN技

LDA理解以及源码分析 (二)

标签: [c++](#) [LDA](#)

2015-12-09 17:31 2809人阅读 [评论\(0\)](#) [收藏](#) [举报](#)

分类:

[主题模型](#) (8) ▾ [C/C++](#) (13) ▾ [机器学习](#) (10) ▾

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

LDA系列的讲解分多个博文给出, 主要大纲如下:

- LDA相关的基础知识
 - 什么是共轭
 - multinomial分布
 - Dirichlet分布
- LDA in text
 - LAD的概率图模型
 - LDA的参数推导
 - 伪代码
- GibbsLDA++-0.2源码分析
- Python实现GibbsLDA
- 参考资料

GibbsLDA++-0.2源码分析

GibbsLDA++-0.2工具包下载地址为: [下载](#)

工具包里docs文件夹里有说明文件GibbsLDA++Manual.pdf, 按照要求编译就可以使用, 很方便。 (具体使用方法后面给出)

代码在文件夹src中, 主要有这么几个类: dataset, model, strtokenizer, utils以及lda.cpp, constants.h文件。

dataset

```

1 //两个全局变量, 分别存储word和id的对应。
2 typedef map<string, int> mapword2id;
3 typedef map<int, string> mapid2word;
4 //类document
5 class document {
6   public:

```

文章存档

- 2017年02月 (1)
- 2016年12月 (1)
- 2016年11月 (5)
- 2016年10月 (1)
- 2016年06月 (2)

展开

阅读排行

- [【Caffe】训练ImageNet](#) (56090)
- [Precision和Recall](#) (37369)
- [【Caffe】训练MNIST数据集](#) (19582)
- [LDA主题模型评估方法--F](#) (19215)
- [【Caffe】下载与安装](#) (13933)
- [Online LDA的python实现](#) (11257)
- [IntelliJ搭建spark开发环境](#) (9025)
- [【Caffe】简单介绍](#) (8991)
- [Python实现Bloom filter](#) (8962)
- [python实现list数组转置](#) (8291)

评论排行

- [LDA主题模型评估方法--F](#) (29)
- [【Caffe】训练ImageNet](#) (22)
- [Online LDA的python实现](#) (14)
- [微软暑期实习面试总结](#) (9)
- [【Caffe】下载与安装](#) (9)
- [【Caffe】训练MNIST数据集](#) (5)
- [Precision和Recall](#) (3)
- [开源项目kcws代码分析--](#) (3)
- [第6章练习题--3--武林--2;](#) (2)
- [python安装模块的方法](#) (2)

推荐文章

- * [CSDN日报20170509——《互联网时代架构师的职责与思考》](#)
- * [程序员要拥抱变化，聊聊Android即将支持的Java 8](#)
- * [彻底弄懂prepack与webpack的关系](#)
- * [用TensorFlow做个聊天机器人](#)
- * [分布式机器学习的集群方案介绍之HPC实现](#)
- * [Android音频系统：从AudioTrack到AudioFlinger](#)

最新评论

mengwang6334: 谢谢楼主的回答，训练步骤详情有点太长，csdn上放不下，我放在阿里云上了，麻烦楼主有时间帮忙看一下。...

开源项目kcws代码分析--基于深度学习

```
7 //保存每个word对应的id
8 int * words;
9 string rawstr;
10 //文章的words总数
11 int length;
12 document() {}
13 document(int length) {}
14 document(int length, int * words) {}
15 document(int length, int * words, string rawstr) {}
16 document(vector<int> & doc) {}
17 document(vector<int> & doc, string rawstr) {}
18 ~document() {}
19 };
20 class dataset {
21 public:
22     document ** docs;
23     document ** _docs; // used only for inference
24     map<int, int> _id2id; // also used only for inference
25     int M; // documents总数
26     int V; // words总数
27     dataset() {}
28     dataset(int M) {}
29     ~dataset() {}
30     void deallocate() {}
31     void add_doc(document * doc, int idx) {}
32     void _add_doc(document * doc, int idx) {}
33     //根据pword2id写wordmap, 文件每行都是“word id”的格式
34     static int write_wordmap(string wordmapfile, mapword2id * pword2id)
35     //读wordmap中的内容, 存储到pword2id中
36     static int read_wordmap(string wordmapfile, mapword2id * pword2id);
37     static int read_wordmap(string wordmapfile, mapid2word * pid2word);
38     int read_trndata(string dfile, string wordmapfile); //读训练文件
39     int read_newdata(string dfile, string wordmapfile); //读推断的新文件
40     int read_newdata_withrawstrs(string dfile, string wordmapfile);
41 };
```

utils

```
1 class utils {
2 public:
3     // 解析命令行参数
4     static int parse_args(int argc, char ** argv, model * pmodel);
5     // 读<model_name>.others文件并解析模型参数
6     static int read_and_parse(string filename, model * model);
7     // 为当前的迭代生成模型名字, 命令行有个参数会指定什么时候需要保存模型, iter=-1时最后
8     static string generate_model_name(int iter);
9     // 排序
10    static void sort(vector<double> & probs, vector<int> & words);
11    static void quicksort(vector<pair<int, double> > & vect, int left, int right);
12 };
```

strtokenizer

```
1 class strtokenizer {
2 protected:
3     vector<string> tokens; //存储分后的词
4     int idx;//tokens的索引
5 public:
6     strtokenizer(string str, string separators = " ");//对str按照separators分割
7     void parse(string str, string separators); //同上
8     int count_tokens(); //返回tokens的大小
9     string next_token(); //返回idx当前索引的token值
10    void start_scan(); //idx = 0
```

蜡笔大龙猫: @mengwang6334: 我不清楚你的实验步骤，无法有针对性的解答。我想到的可以参考的三点，一是训...

开源项目kcws代码分析--基于深度
mengwang6334: 您好，感谢楼主的分享。我想问一下，我这边跑出来的模型，在分词时达不到陈老师demo上的效果。训练时我...

Kullback-Leibler Divergence, KL
momomomo22: 该图仅限百度内部用户交流使用。。。

【Caffe】训练ImageNet模型
元气少女缘结神: @faigel:我也一样

IntelliJ搭建spark开发环境
shui0855: 新建scala项目，pom文件是怎么出来的？？？

LDA主题模型评估方法--Perplexit
LFGxiaogang: 为什么我算的主题越少困惑度越低啊？

【Caffe】下载与安装
qq_33377927: 您好！请问如果不进行make pycaffe会有什么影响吗？

LDA主题模型评估方法--Perplexit
Cathy1272014: 您好，我想问一个问题关于LDA的问题，我在计算perplexity时候对我所有的训练集进行计算，然后在主...

【Caffe】训练ImageNet模型



甩脂机懒人塑身机健身机运

¥569.40/台

广告

```
12     string token(int i); // 返回tokens[i]的值
    };
```

model类是最重要的类，实现核心代码

```
1 class model {
2     public:
3         // fixed options
4         string wordmapfile; // file that contains word map [string -> int]
5         string trainlogfile; // training log file
6         string tassign_suffix; // suffix for topic assignment file
7         string theta_suffix; // suffix for theta file
8         string phi_suffix; // suffix for phi file
9         string others_suffix; // suffix for file containing other parameters
10        string twords_suffix; // suffix for file containing words-per-topic
11        string dir; // model directory
12        string dfile; // data file
13        string model_name; // model name
14        int model_status; // model status: 本代码提供est, estc和inf三种模式
15        dataset * ptrndata; // 指针，指向训练文档集
16        dataset * pnewdata; // 指针，指向推断的新文档集
17        mapid2word id2word; // word map [int -> string]
18     /*下面是模型参数和变量*/
19        int M; // documents数量
20        int V; // words数量 (不重复)
21        int K; // topics数量
22        double alpha, beta; // LDA超参数
23        int niters; // Gibbs采样迭代的次数
24        int liter; // 需要将模型保存为文件的迭代次数
25        int savestep; // saving period
26        int twords; // 输出每个topic的前twords个词
27        int withdrawstrs;
28     /*下面是LDA核心算法中的变量*/
29        double * p; // 采样的临时变量
30        int ** z; // M x doc.size(), 文档中words的topic分布
31        int ** nw; // V x K, nw[i][j]: 词i在主题j上出现的次数
32        int ** nd; // M x K, nd[i][j]: 文章i中属于主题j的词的个数
33        int * nwsum; // K, nwsum[j]: 属于主题j的词的数量
34        int * ndsum; // M, ndsum[i]: 文章i中词的数量
35        double ** theta; // M x K, document-topic分布
36        double ** phi; // K x V, topic-word分布
37     /*下面的变量只有在推断时用到*/
38        int inf_liter;
39        int newM;
40        int newV;
41        int ** newz;
42        int ** newnw;
43        int ** newnd;
44        int * newnwsum;
45        int * newndsum;
46        double ** newtheta;
47        double ** newphi;
48     // -----
49        model() {
50        ~model();
51        // 对变量设置初值，构造函数中调用
52        void set_default_values();
53        // 解析命令行得到LDA参数
54        int parse_args(int argc, char ** argv);
55        // 初始化模型，调用后面的init()设置初值
56        int init(int argc, char ** argv);
57        // 加载已有的LDA模型继续训练或者推断
58        int load_model(string model_name);
59        // 保存LDA模型文件
```



```
60     int save_model(string model_name);
61     int save_model_tassign(string filename);
62     int save_model_theta(string filename);
63     int save_model_phi(string filename);
64     int save_model_others(string filename);
65     int save_model_twords(string filename);
66     // 保存inf的结果
67     int save_inf_model(string model_name);
68     int save_inf_model_tassign(string filename);
69     int save_inf_model_newtheta(string filename);
70     int save_inf_model_newphi(string filename);
71     int save_inf_model_others(string filename);
72     int save_inf_model_twords(string filename);
73     // 训练模型前的初始化, init()函数中会调用
74     int init_est();
75     int init_estc();
76     // LDA核心算法!!! 下面详细介绍
77     void estimate();
78     int sampling(int m, int n);
79     //计算迭代之后的theta和phi
80     void compute_theta();
81     void compute_phi();
82     // 推断初始化, 在init()中调用
83     int init_inf();
84     // 在已知LDA模型中, 对未知的新文档进行推断
85     void inference();
86     int inf_sampling(int m, int n);
87     void compute_newtheta();
88     void compute_newphi();
89 }
```

下面详细分析estimate()函数和sampling()函数

```
1 void model::estimate() {
2     if (twords > 0) { //如果要求输出主题下的词, 则读wordmap文件排序后输出
3         dataset::read_wordmap(dir + wordmapfile, &id2word);
4     }
5     printf("Sampling %d iterations!\n", niters);
6     int last_iter = liter; //记录迭代次数
7     for (liter = last_iter + 1; liter <= niters + last_iter; liter++) {
8         printf("Iteration %d ...\n", liter);
9         // 对所有的z[m][n]采样一个主题
10        for (int m = 0; m < M; m++) {
11            for (int n = 0; n < ptrndata->docs[m]->length; n++) {
12                // sampling根据Gibbs采样公式p(z_i|z_{-i}, w)采样
13                int topic = sampling(m, n);
14                z[m][n] = topic;
15            }
16        }
17        //判断是否需要保存这次迭代的模型
18        if (savestep > 0) {
19            if (liter % savestep == 0) {
20                printf("Saving the model at iteration %d ...\n", liter);
21                compute_theta();
22                compute_phi();
23                save_model(utils::generate_model_name(liter));
24            }
25        }
26    }
27    //迭代结束, 根据theta和phi的公式计算值, 并且保存模型文件
28    printf("Gibbs sampling completed!\n");
29    printf("Saving the final model!\n");
30    compute_theta();
31    compute_phi();
```

```

32     liter--;
33     save_model(utils::generate_model_name(-1));
34 }

1     int model::sampling(int m, int n) {
2         //采样算法，排除当前词，根据其他词的主题分布计算当前词的分布
3         int topic = z[m][n];//获得当前主题
4         int w = ptrndata->docs[m]->words[n]; //获得当前词的id
5         //首先，排除当前词，即涉及的统计变量个数减一
6         nw[w][topic] -= 1;
7         nd[m][topic] -= 1;
8         nwsum[topic] -= 1;
9         ndsum[m] -= 1;
10        //临时计算变量
11        double Vbeta = V * beta;
12        double Kalpha = K * alpha;
13        /*通过累加来采样*/
14        //计算当前词在每个主题下的概率
15        for (int k = 0; k < K; k++) {
16            p[k] = (nw[w][k] + beta) / (nwsum[k] + Vbeta) *
17                (nd[m][k] + alpha) / (ndsum[m] + Kalpha);
18        }
19        // 概率和累加
20        for (int k = 1; k < K; k++) {
21            p[k] += p[k - 1];
22        }
23        //随机生成小数
24        double u = ((double)random() / RAND_MAX) * p[K - 1];
25        //根据随机生成的值，采样主题编号
26        for (topic = 0; topic < K; topic++) {
27            if (p[topic] > u) {
28                break;
29            }
30        }
31        //新的topic对应的参数增加1
32        nw[w][topic] += 1;
33        nd[m][topic] += 1;
34        nwsum[topic] += 1;
35        ndsum[m] += 1;
36        return topic;
37    }

```



顶 踩
1 0

[上一篇 LDA理解以及源码分析（一）](#)

[下一篇 【正则表达式】pyahocorasick介绍](#)

我的同类文章

主题模型 (8)	C/C++ (13)	机器学习 (10)
--------------------------	----------------------------	---------------------------

- | | |
|--|---|
| • LDA理解以及源码分析（一） 2015-12-09 阅读 3786 | • Spark LDA 2015-12-08 阅读 5107 |
| • Topic Model的分类总结 (L...) 2013-07-29 阅读 5141 | • LDA相关论文汇总 2013-07-25 阅读 8114 |
| • LDA主题模型评估方法--Per... 2013-07-18 阅读 19138 | • 搜索背后的奥秘——浅谈语... 2013-06-06 阅读 905 |

1 腾讯云CPS推广,轻松月入上千

2 [微三云]—为企业量身定制

参考知识库



Python知识库

23292 关注 | 1612 收录



算法与数据结构知识库

16056 关注 | 2320 收录

猜你在找

《C语言/C++学习指南》加密解密篇（安全相关算法）深入理解Spark 2.1 Core 十二TimSort 的原理与源码分析

Android核心技术——Android数据存储

ListView源码分析二

360度解析亚马逊AWS数据存储服务

WINVNC源码分析二图像

C++标准模板库从入门到精通

Android单元测试框架源码分析二浅析Robolectric

使用 AdaBoost 算法进行二分类实战

Zookeeper源码分析之请求处理链二



甩脂机懒人塑身机健身机运

¥569.40/台

广告



十佳笔记本电



加盟婴儿游泳



三国志13中文



超薄手机



高配台式电脑



手机报价大全



沙盘模型

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved

