

个人资料



tbkken

+ 加关注

发私信



访问:2822次

积分:146分

排名:千里之外

原创:12篇

转载:2篇

译文:0篇

评论:2条

文章搜索

Q

文章分类

- 项目管理(3)
- SVN(2)
- Apache(2)
- 文档版本管理(2)
- 单元测试(0)
- 单元测试(1)
- 可行性分析报告(1)
- 数据分析(7)
- sh(0)
- 主成分分析(1)
- R(3)
- 推荐引擎(1)
- 协同推荐(1)
- 手机型号(1)
- EXCEL(1)
- 运营推广(3)
- 留存率(3)
- 移动互联网运营分析(1)
- 贝叶斯(1)
- 分类(1)
- 决策树(1)
- 算法(1)
- weka(1)
- Linux(1)

文章存档

“移动开发那点事”——主题征文活动 浓缩六届精华, 国内大数据领域最纯粹技术盛会 CSDN高校俱乐部专家巡讲师招募
移动开发者大会最新议题发布, 八折抢票！ 新版论坛全民公测！ 2012年10月当选微软MPP的CSDN会员名单揭晓！

原 基于贝叶斯算法的文本分类算法

分类: 贝叶斯 分类

2012-10-11 21:08

301人阅读

评论(0)

收藏

举报

因为要做一个关于数据挖掘的算法应用PPT，虽然知道很多数据挖掘的算法怎么使用，但是需要讲解它们的原理，还真的需要耗费很多精力，之前做一个曲线拟合，已经发在博客里，现在做贝叶斯算法的基础原理。

1、基本定义：

分类是把一个事物分到某个类别中。一个事物具有很多属性，把它的众多属性看作一个向量，即 $x=(x_1,x_2,x_3,...,x_n)$ ，用 x 这个向量来代表这个事物， x 的集合记为 X ，称为属性集。类别也有很多种，用集合 $C=\{c_1,c_2,...,c_m\}$ 表示。一般 X 和 C 的关系是不确定的，可以将 X 和 C 看作是随机变量， $P(C|X)$ 称为 C 的后验概率，与之相对的， $P(C)$ 称为 C 的先验概率。

根据贝叶斯公式， $P(C|X)=P(X|C)P(C)/P(X)$ ，但在比较不同 C 值的后验概率时，分母 $P(X)$ 总是常数，忽略掉， $P(C|X)=P(X|C)P(C)$ ，先验概率 $P(C)$ 可以通过计算训练集中属于每一个类的训练样本所占的比例，容易估计，对类条件概率 $P(X|C)$ 的估计，这里我只说朴素贝叶斯分类器方法，因为朴素贝叶斯假设事物属性之间相互条件独立， $P(X|C)=\prod P(x_i|c_i)$ 。

2、文本分类过程

例如文档：Good good study Day day up可以用一个文本特征向量来表示， $x=(\text{Good}, \text{good}, \text{study}, \text{Day}, \text{day}, \text{up})$ 。

在文本分类中，假设我们有一个文档 $d \in X$ ，类别 c 又称为标签。我们把一堆打了标签的文档集合作为训练样本， $\in X \times C$ 。例如： $=\{\text{Beijing joins the World Trade Organization, China}\}$ 对于这个只有一句话的文档，我们把它归类到 China，即打上china标签。

朴素贝叶斯分类器是一种有监督学习，常见有两种模型，多项式模型(Multinomial Model)即为词频型和伯努利模型(Bernoulli Model)即文档型。二者的计算粒度不一样，多项式模型以单词为粒度，伯努利模型以文件为粒度，因此二者的先验概率和类条件概率的计算方法都不同。

计算后验概率时，对于一个文档 d ，多项式模型中，只有在 d 中出现过的单词，才会参与后验概率计算，伯努利模型中，没有在 d 中出现，但是在全局单词表中出现的单词，也会参与计算，不过是作为“反方”参与的。这里暂不考虑特征抽取、为避免消除测试文档时类条件概率中有为0现象而做的取对数等问题。

2.1多项式模型

1) 基本原理

在多项式模型中，设某文档 $d=(t_1,t_2,...,t_k)$ ， t_k 是该文档中出现过的单词，允许重复，则

先验概率 $P(c)$ = 类 c 下单词总数/整个训练样本的单词总数

类条件概率 $P(t_k|c)=(\text{类}c\text{下单词}t_k\text{在各个文档中出现过的次数之和}+1)/(\text{类}c\text{下单词总数}+|V|)$

V 是训练样本的单词表（即抽取单词，单词出现多次，只算一个）， $|V|$ 则表示训练样本包含多少种单词。 $P(t_k|c)$ 可以看作是单词 t_k 在证明 d 属于类 c 上提供了多大的证据，而 $P(c)$ 则可以认为是类别 c 在整体上占多大比例(有多大可能性)。

2) 举例

给定一组分好类的文本训练数据，如下：

docId	doc	类别 In c=China?
1	Chinese Beijing Chinese	yes
2	Chinese Chinese Shanghai	yes

2012年10月(3)
2012年09月(6)
2012年06月(5)

阅读排行

- 五、泛单点登录(SSO)系 (526)
- 读懂你的用户留存 (518)
- 基于贝叶斯算法的文本分 (301)
- AARRR模型——揭开应 (281)
- 主成分分析 (257)
- 决策树算法 (238)
- Linux环境下安装R (170)
- 3、关于手机型号的前缀 (104)
- 二、SVN服务器的搭建和 (90)
- 四、可行性分析报告 (89)

评论排行

- 五、泛单点登录(SSO)系 (2)
- 一、SVN服务器的搭建和 (0)
- 决策树算法 (0)
- 基于贝叶斯算法的文本分 (0)
- 4、数据分析师对AARRR (0)
- 读懂你的用户留存 (0)
- AARRR模型——揭开应 (0)
- 3、关于手机型号的前缀 (0)
- 2、推荐引擎以及协同过 (0)
- 主成分分析 (0)

推荐文章

- * 企业级JavaEE开发框架bossgr
- * OpenStack成都站10月27日火速
- * Android Content Provider详解六
- * Silverlight中设计焦点和文本框回
- * 如何在iOS中使用libxml
- * RazorSourceGenerator 代码生成

最新评论

- 五、泛单点登录(SSO)系统需求 (tbkken: @cmengwei:是啊, 你也做程序员了?)
- 五、泛单点登录(SSO)系统需求 (cmengwei: 学习 第一个

3	Chinese Macao	yes
4	Tokyo Japan Chinese	no

给定一个新样本Chinese Chinese Chinese Tokyo Japan, 对其进行分类。该文本用属性向量表示为d=(Chinese, Chinese, Chinese, Tokyo, Japan), 类别集合为Y={yes, no}。

类yes下总共有8个单词, 类no下总共有3个单词, 训练样本单词总数为11, 因此P(yes)=8/11, P(no)=3/11。类条件概率计算如下:

$P(\text{Chinese} | \text{yes}) = (5+1)/(8+6) = 6/14 = 3/7$

$P(\text{Japan} | \text{yes}) = P(\text{Tokyo} | \text{yes}) = (0+1)/(8+6) = 1/14$

$P(\text{Chinese} | \text{no}) = (1+1)/(3+6) = 2/9$

$P(\text{Japan} | \text{no}) = P(\text{Tokyo} | \text{no}) = (1+1)/(3+6) = 2/9$

分母中的8, 是指yes类别下textc的长度, 也即训练样本的单词总数, 6是指训练样本有Chinese, Beijing, Shanghai, Macao, Tokyo, Japan 共6个单词, 3是指no类下共有3个单词。

有了以上类条件概率, 开始计算后验概率:

$P(\text{yes} | d) = (3/7)^3 \times 1/14 \times 1/14 \times 8/11 = 108/184877 \approx 0.00058417$

$P(\text{no} | d) = (2/9)^3 \times 2/9 \times 2/9 \times 3/11 = 32/216513 \approx 0.00014780$

比较大小, 即可知道这个文档属于类别china。

2.2伯努利模型

1) 基本原理

P(c)= 类c下文件总数/整个训练样本的文件总数

$P(\text{tk} | c) = (\text{类c下包含单词tk的文件数} + 1) / (\text{类c下单词总数} + 2)$

2) 举例

使用前面例子中的数据, 模型换成伯努利模型。

类yes下总共有3个文件, 类no下有1个文件, 训练样本文件总数为11, 因此P(yes)=3/4, P(Chinese | yes)=(3+1)/(3+2)=4/5, 条件概率如下:

$P(\text{Japan} | \text{yes}) = P(\text{Tokyo} | \text{yes}) = (0+1)/(3+2) = 1/5$

$P(\text{Beijing} | \text{yes}) = P(\text{Macao} | \text{yes}) = P(\text{Shanghai} | \text{yes}) = (1+1)/(3+2) = 2/5$

$P(\text{Chinese} | \text{no}) = (1+1)/(1+2) = 2/3$

$P(\text{Japan} | \text{no}) = P(\text{Tokyo} | \text{no}) = (1+1)/(1+2) = 2/3$

$P(\text{Beijing} | \text{no}) = P(\text{Macao} | \text{no}) = P(\text{Shanghai} | \text{no}) = (0+1)/(1+2) = 1/3$

有了以上类条件概率, 开始计算后验概率,

$P(\text{yes} | d) = P(\text{yes}) \times P(\text{Chinese} | \text{yes}) \times P(\text{Japan} | \text{yes}) \times P(\text{Tokyo} | \text{yes}) \times (1 - P(\text{Beijing} | \text{yes})) \times (1 - P(\text{Shanghai} | \text{yes})) \times (1 - P(\text{Macao} | \text{yes})) = 3/4 \times 4/5 \times 1/5 \times 1/5 \times (1-2/5) \times (1-2/5) \times (1-2/5) = 81/15625 \approx 0.005$

$P(\text{no} | d) = 1/4 \times 2/3 \times 2/3 \times 2/3 \times (1-1/3) \times (1-1/3) \times (1-1/3) = 16/729 \approx 0.022$

因此, 这个文档不属于类别china。

后记: 文本分类是作为离散型数据的, 以前糊涂是把连续型与离散型弄混一块了, 朴素贝叶斯用于很多方面, 数据就会有连续和离散的, 连续型时可用正态分布, 还可用区间, 将数据的各属性分成几个区间段进行概率计算, 测试时看其属性的值在哪个区间就用哪个条件概率。再有TF、TDIDF, 这些只是描述事物属性时的不同计算方法, 例如文本分类时, 可以用单词在本文档中出现的次数描述一个文档, 可以用出现还是没出现即0和1来描述, 还可以用单词在本类文档中出现的次数与这个单词在剩余类出现的次数(降低此属性对某类的重要性)相结合来表述。

摘自: <http://m.oschina.net/blog/56724>

上一篇: 4、数据分析师对AARRR模型的应用思考

分享到:  

下一篇: 决策树算法

顶0

踩0







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

专区推荐内容

-  高性能计算
-  从模拟器看泰泽系统
-  一个简单的游戏服务器框架
-  工欲善其事,必先利其器-Wind...
-  无需编程知识, 让你零基础打造HT...
-  并行计算性能测试



更多招聘职位

-  [【CIC 济南】诚聘售前工程师、商务经理、I...](#)
-  [【新华社浙江分社】【新华社浙江分社】诚聘](#)
-  [【2345网址导航】诚聘 C++高级开发工程师](#)
-  [【登邦信息】诚聘 用户体验设计师等](#)
-  [【全景赛斯】诚聘 高级软件工程师](#)
-  [【上海交大】e-learning lab诚聘研发工程](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#)

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有

✉ 联系邮箱: [webmaster\(at\)csdn.net](mailto:webmaster(at)csdn.net)

Copyright © 1999-2012, CSDN.NET, All Rights Reserved

