个人资料

xceman1997

【朴素贝叶斯 贝叶斯_代码实现_训练算法

标签： 算法 数据结构 cla delete training

2012-09-07 16:01 14120人阅读 评论(1) 收藏 举报

分类： NLP（101）

说一下Train函数的实现。

在上文中，我提到过，朴素贝叶斯也有两种模型：贝努力模型和多项式模型。小弟第一次实现朴素贝叶斯，就老老实实按照基本原理做了一个贝努力模型；多项式模型也不难，变通一下就行。不废话了，直接上代码了，有点长，不过很容易看懂：

```cpp
bool NaiveBayes::Train (const char * sFileSample, int iClassNum, int iFeaTypeNum,
                        string & sSegmenter, int iFeaExtractNum, const char * sFileMode
{
    // 防御性代码
    if (iClassNum <= 0 || iFeaTypeNum <= 0 || iFeaExtractNum <= 0)
        return false;

    ifstream in (sFileSample, ios_base::binary);
    ofstream out (sFileModel);
    if (!in || !out)
    {
        cerr << "Can not open the file" << endl;
        return false;
    }

    // 这些都是临时数据结构，用来临时存储模型参数，特征选择需要的参数等等
    // 1. the temp data structure for model parameters
    // 1.1 the total number of document in training samples
    int iTotalDocNum = 0;
    // 1.2 the prior probability of class, temparaly it store the doc number in this class
    double * pClassPriorProb = new double [iClassNum];
    memset (pClassPriorProb, 0, iClassNum*sizeof(double));
    // 1.3 the prior probability of feature type, temparaly it stores the doc number in this
个主要用于特征选择，bayes模型本身并不需要这个参数）
    double * pFeaItemPriorProb = new double [iFeaTypeNum];
    memset (pFeaItemPriorProb, 0, iFeaTypeNum*sizeof(double));
    // 1.4 the chi-
square value that feature falls into class, temparaly it stores the doc number for this clas
以看到，特征选择算法主要用卡方选择）
    double ** ppChiMatrix = new double * [iClassNum];
    for (int i=0; i<iClassNum; i++)
    {
        ppChiMatrix[i] = new double [iFeaTypeNum];
        memset (ppChiMatrix[i], 0, iFeaTypeNum*sizeof(double));
    }
    // 1.5 the post-probability for class and feature
    double ** ppPProbMatrix = new double * [iClassNum];
    for (int i=0; i<iClassNum; i++)
    {
```

访问： 318675次
积分： 5088
等级： BLOG 6
排名： 第2883名

原创： 169篇 转载： 154篇
译文： 0篇 评论： 69条

文章搜索

文章分类

基础知识 (46)
c/c++ (26)
NLP (102)
hadoop (15)
Java (18)
Python (17)
机器学习 (140)
Linux (7)
研发管理 (10)
杂谈 (3)
DL (49)

文章存档

2015年12月 (3)
2015年11月 (1)
2015年08月 (1)
2015年07月 (1)
2015年06月 (7)

展开

阅读排行

【朴素贝叶斯】实战朴素

(14115)
【学习笔记】Jaccard相 (6320)
【Deep Learning学习笔 (6191)
【朴素贝叶斯】实战朴素 (4530)
【读书笔记】《大数据— (4496)
【朴素贝叶斯】实战朴素 (4363)
【朴素贝叶斯】实战朴素 (3807)
【hadoop学习】在Mac ((3559)
【Deep Learning学习笔 (3374)
【deep learning学习笔记 (3148)

评论排行

【java学习】Jni在hadoc (7)
【deep learning学习笔记 (5)
【deep learning学习笔记 (4)
【Deep Learning学习笔 (4)
【朴素贝叶斯】实战朴素 (3)
【读书笔记】《大数据— (3)
【hadoop学习】在MacC (2)
【学习笔记】《卓有成效 (2)
【deep learning学习笔记 (2)
【资源】机器学习资源积 (2)

推荐文章

*App竞品技术分析 （6）热修复
*架构设计：系统间通信（17）——服务治理与Dubbo 中篇（分析）
*你的计划为什么执行不下去？怎么破？
*图解堆算法、链表、栈与队列（多图预警）
*【android】仿360手机卫士的简易设计思路及源码
*Android平台Camera实时滤镜实现方法探讨(九)--磨皮算法探讨(一)

最新评论

【deep learning学习笔记】最近
love_apple_yan: 欢迎加入机器学习研究QQ群445858879，可以跟悉尼科技大学博导徐亦达教授亲切交流，不过最好使用...
Denoising Autoencoders (dA)
TJU_LUNA: 感谢分享
【deep learning学习笔记】Gree
幺鸡报晓: 最后说原作者有附伪代码，但我搜到的论文都没有伪代码。能给个地址吗
【读书笔记】《大数据——互联
火星十一郎: 这本书我有了，求其他参考资料，推荐系统 最优化等
【读书笔记】《大数据——互联
火星十一郎: 大神，求这本书的pdf
【读书笔记】基于Autoencoder
IncStrive: lz您好，第二个问题“autoencoder最后训练出的结果怎么用”解决了吗？
Logistic Regression（c++）源
xceman1997: @chinachenyadong:的确是的。非常感谢您的批评指正！
Logistic Regression（c++）源
相门码农: 您好，貌似在LogisticRegression.cpp中的211行处：if (ClassProbV...

```
37.         ppPProbMatrix[i] = new double [iFeaTypeNum];
38.         memset (ppChiMatrix[i], 0, iFeaTypeNum*sizeof(double));
39.     }
40.     // 1.6 for the feature selection （表示哪些特征被选中了）
41.     int * pFeaSelected = new int [iFeaTypeNum];
42.     memset (pFeaSelected, 0, iFeaTypeNum*sizeof(int));
43. 
44.     // 2. iterate the training samples and fill count into the temp data structure
45.     string sLine;
46.     int i = 0;
47.     while (getline (in, sLine))
48.     {
49.         // show some information on screen
50.         if (0 == i%10000)
51.             cout << i << "\n";
52.         i++;
53. 
54.         // 2.1 the total number of doc
55.         iTotalDocNum++;
56. 
57.         // 2.2 split the sample into class and feature items
58.         string::size_type iSeg = sLine.find_first_of (sSegmenter);
59.         string sTmp = sLine.substr (0, iSeg);
60.         int iClassId = atoi (sTmp.c_str());
61.         if (iClassId >= iClassNum)
62.             continue;
63.         pClassPriorProb [iClassId]++;
64. 
65.         // 2.3 count the rest feature items
66.         iSeg += sTmp.length();
67.         sTmp = sLine.substr (iSeg);
68.         istringstream isLine (sTmp);
69.         string sTmpItem;
70.         while (isLine >> sTmpItem)
71.         {
72.             int iFeaItemId = atoi (sTmpItem.c_str());
73.             if (iFeaItemId >= iFeaTypeNum)
74.                 continue;
75.             // add the count
76.             pFeaItemPriorProb [iFeaItemId]++;
77.             ppChiMatrix [iClassId][iFeaItemId]++;
78. 
79.         }
80.     }
81. 
82.     // 3. calculate the model parameters
83.     // 3.1 the chi-square value as well as the post-probabilty
84.     for (int i=0; i<iClassNum; i++)
85.     {
86.         for (int j=0; j<iFeaTypeNum; j++)
87.         {
88.             double dA = ppChiMatrix[i][j];
89.             double dB = pFeaItemPriorProb[j] - dA; // currently pFeaItemPriorProb[i] == sum_
[j])
90.             double dC = pClassPriorProb [i] - dA;  // currently pClassPriorProb[i] == sum_j
[j])
91.             double dD = (double)iTotalDocNum - dA - dB - dC;
92. 
93.             // the chi value
94.             double dNumerator = dA * dD;
95.             dNumerator -= dB * dC;
96.             dNumerator = pow (dNumerator, 2.0);
97.             double dDenominator = dA + dB;
98.             dDenominator *= (dC + dD);
99.             dDenominator += DBL_MIN; // for smoothing
100.             ppChiMatrix[i][j] = dNumerator / dDenominator;
101. 
102.             // the post-probability: p(feature|class)
103.             ppPProbMatrix[i][j] = dA / pClassPriorProb [i];
104.         }
105.     }
106. 
107.     // 3.2 the prior probability of class
108.     for (int i=0; i<iClassNum; i++)
109.         pClassPriorProb [i] /= iTotalDocNum;
110. 
111.     // 3.3 the prior probability of feature
112.     for (int i=0; i<iFeaTypeNum; i++)
113.         pFeaItemPriorProb [i] /= iTotalDocNum;
```

动手实现Logistic Regression （
xceman1997: @a130737:哥们
儿您过奖了~

动手实现Logistic Regression （
JUAN425: 哥 你太强了 学习代码
中

```
114.
115.
116.    // 4. feature selection （这个函数下一篇文章再详细讲）
117.    FeaSelByChiSquare (ppChiMatrix, ppPProbMatrix, iClassNum,
118.        iFeaTypeNum, iFeaExtractNum, pFeaSelected);
119.
120.    // 5. dump the model into txt file
121.
122.    if (bCompactModel)      // output the parameters only for predicting
123.    {
124.        // 5.1 the prior probability of class
125.        out << iClassNum << endl;
126.        for (int i=0; i<iClassNum; i++)
127.        {
128.            out << pClassPriorProb [i] << "\n";
129.        }
130.        // 5.2 the actual selected feature type number
131.        int iActualFeaNum = 0;
132.        for (int j=0; j<iFeaTypeNum; j++)
133.        {
134.            if (1 == pFeaSelected[j])
135.                iActualFeaNum ++;
136.        }
137.        out << iActualFeaNum << endl;
138.        // 5.3 the post probability
139.        for (int i=0; i<iClassNum; i++)
140.        {
141.            for (int j=0; j<iFeaTypeNum; j++)
142.            {
143.                if (1 == pFeaSelected[j])
144.                {
145.                    out << j << ":" << ppPProbMatrix[i][j] << "\n";
146.                }
147.            }
148.        }
149.    }
150.    else                    // output the full information
151.    {
152.        // 5.1 the total number of document
153.        out << iTotalDocNum << endl;
154.
155.        // 5.2 the prior probability of class
156.        out << iClassNum << endl;
157.        for (int i=0; i<iClassNum; i++)      // classindex:priorprob
158.        {
159.            out << i << ":" << pClassPriorProb [i] << "\n";
160.        }
161.
162.        // 5.3 the prior probability of feature type: this is NO used in bayes model, recor
163.        //     and whether this feature is selected or not by any class
164.        out << iFeaTypeNum << "\n";
165.        for (int i=0; i<iFeaTypeNum; i++)    // featureId:priorprob:selected or not
166.        {
167.            out << i << ":" << pFeaItemPriorProb[i] << ":" << pFeaSelected << "\n";
168.        }
169.
170.        // 5.4 the chi-square value for class-feature pair
171.        for (int i=0; i<iClassNum; i++)
172.        {
173.            for (int j=0; j<iFeaTypeNum; j++)
174.            {
175.                out << ppChiMatrix[i][j] << "\n";
176.            }
177.        }
178.
179.        // 5.5 the post probability
180.        for (int i=0; i<iClassNum; i++)
181.        {
182.            for (int j=0; j<iFeaTypeNum; j++)
183.            {
184.                out << ppPProbMatrix[i][j] << "\n";
185.            }
186.        }
187.    }
188.
189.    // last, release the memory
190.    delete [] pClassPriorProb;
191.    delete [] pFeaItemPriorProb;
192.    for (int i=0; i<iClassNum; i++)
```

```
    {
        delete [] ppChiMatrix[i];
    }
    delete [] ppChiMatrix;
    for (int i=0; i<iClassNum; i++)
    {
        delete [] ppPProbMatrix[i];
    }
    delete [] ppPProbMatrix;
    delete [] pFeaSelected;

    return true;
}
```

在这个函数的主要功能就是统计，一方面统计后验概率，另一方面统计卡方特征选择所需要的参数，他们分别存储在ppChiMaxtrix和ppProbMatrix。这两个都是二维数组，数组的维度由类别数目和特征类型总数决定。数据结构的设计风格都是c的风格，自己搭建数据，并负责释放；并没有用c++ stl中的复杂数据结构，如：set、map等。中间经过特征选择，最后模型存储在文本文件当中。

接下来，讲特征选择算法。

上一篇 【朴素贝叶斯】实战朴素贝叶斯_代码实现_数据和接口

下一篇 【朴素贝叶斯】实战朴素贝叶斯_代码实现_特征选择1

查看评论

1楼 tony958 2014-11-12 20:59发表

正在学习bayes分类,拜读了,非常有帮助,大牛!.

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