

[注册](#) (http://member.broadview.com.cn/register?returnUrl=http%3a%2f%2fwww.broadview.com.cn%2farticle%2f318)

[登录](#) (http://member.broadview.com.cn/log-in?returnUrl=http%3a%2f%2fwww.broadview.com.cn%2farticle%2f318)

[图书](#)

[人工智能与机器学习 \(/book?category=88\)](#)

[数据处理与大数据 \(/book?category=1\)](#)

[Web技术 \(/book?category=9\)](#)

[移动开发 \(/book?category=18\)](#)

[游戏与VR/AR \(/book?category=23\)](#)

[程序设计与软件工程 \(/book?category=27\)](#)

[前端技术 \(/book?category=55\)](#)

[产品与设计 \(/book?category=38\)](#)

[云计算 \(/book?category=96\)](#)

[办公软件 \(/book?category=47\)](#)

[IT与互联网 \(/book?category=70\)](#)

[电子书 \(/book?tab=ebook\)](#)

[专题](#)

[大数据 \(/tech/3\)](#)

[数据库 \(/tech/14\)](#)

[互联网+ \(/tech/16\)](#)

[安全技术 \(/tech/17\)](#)

[架构设计 \(/tech/18\)](#)

[游戏设计与开发 \(/tech/19\)](#)

[产品与设计 \(/tech/20\)](#)

[编程语言 \(/tech/21\)](#)

[移动开发 \(/tech/22\)](#)

[前端技术 \(/tech/23\)](#)

[人工智能 \(/tech/24\)](#)

[阿里巴巴技术专区 \(/tech/25\)](#)

[《Keras快速上手》专区 \(/tech/26\)](#)

[嘉宾访谈室 \(/tech/27\)](#)

[帮助](#)

[关于我们 \(/support/1\)](#)

[用户协议 \(/support/2\)](#)

[联系我们 \(/support/3\)](/support/3)

[作译者帮助 \(/support/4\)](/support/4)

[关于积分 \(/support/5\)](/support/5)

[注册 \(http://member.broadview.com.cn/register?returnUrl=http%3a%2f%2fwww.broadview.com.cn%2farticle%2f318\)](http://member.broadview.com.cn/register?returnUrl=http%3a%2f%2fwww.broadview.com.cn%2farticle%2f318)

[登录 \(http://member.broadview.com.cn/log-in?returnUrl=http%3a%2f%2fwww.broadview.com.cn%2farticle%2f318\)](http://member.broadview.com.cn/log-in?returnUrl=http%3a%2f%2fwww.broadview.com.cn%2farticle%2f318)



[\(/user/cart\)](/user/cart)



博文视点

电子工业出版社旗下IT出版专业品牌

[图书 \(/book\)](/book)

[电子书 \(/book?tab=ebook\)](/book?tab=ebook)

[专题 \(/tech\)](/tech)

[帮助 \(/support\)](/support)



[\(/user/shelf\)](/user/shelf)



[\(/user/cart\)](/user/cart)

[首页 \(http://www.broadview.com.cn\)](http://www.broadview.com.cn) > [专题 \(/tech\)](/tech) > [机器学习算法实现解析——libFM之libFM的模型处理部分 \(/article/318\)](/article/318)



机器学习算法实现解析—— libFM之libFM的模型处理部分

赵志勇 (/space/index/25912)

2017-06-19

推荐0 收藏0 浏览445



本节主要介绍的是libFM源码分析的第三部分——libFM的模型处理。

3.1、libFM中FM模型的定义

libFM模型的定义过程中主要包括模型中参数的设置及其初始化，利用模型对样本进行预测。在libFM中，首先定义FM模型，在fm_model类中实现对FM模型的定义，fm_model类在“\libfm-1.42.src\src\fm_core\fm_model.h”中。在定义fm_model类之前，使用到了一些数据类：

```
#include "../util/matrix.h"
#include "../util/fmatrix.h"

#include "fm_data.h"
```

数据类的具体定义在“机器学习算法实现解析——libFM之libFM的数据处理部分”中定义。fm_model类的代码如下所示：

```

// fm_model模型类
class fm_model {
    private:
        DVector<double> m_sum, m_sum_sqr;// 分别对应着交叉项的中的两项
    public: //fm模型中的参数
        double w0;// 常数项
        DVectorDouble w;// 一次项的系数
        DMatrixDouble v;// 交叉项的系数矩阵

    public:
        // 属性
        // the following values should be set:
        uint num_attribute;// 特征的个数

        bool k0, k1;// 是否包含常数项和一次项
        int num_factor;// 交叉项因子的个数

        double reg0;// 常数项的正则参数
        double regw, regv;// 一次项和交叉项的正则系数

        double init_stdev;// 初始化参数时的方差
        double init_mean;// 初始化参数时的均值

        // 函数
        fm_model();// 构造函数，主要完成参数的初始化
        void debug();// debug函数
        void init();// 初始化函数，主要用于生成各维度系数的初始值
        // 对样本进行预测
        double predict(sparse_row<FM_FLOAT>& x);
        double predict(sparse_row<FM_FLOAT>& x, DVector<double> &sum, DVector
};

```

FM模型的一般形式如下所示：

$$\hat{y} := w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

其中， w_0 为常数项系数， w_i 为一次项系数， \mathbf{v}_i 和 \mathbf{v}_j 为交叉项系数。对于交叉项系数 \mathbf{v}_i ，其具体的形式为：

$$\mathbf{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,k})_{1 \times k}$$

在FM模型的定义中，首先需要分别定义三个参数： w_0 ， w 和 v 。其次，需要定义模型中需要使用到的函数，包括初始化init函数和预测predict函数。

3.2、FM的初始化

完成FM模型初始化过程主要包括两个部分：

构造函数fm_model()

init()函数

构造函数fm_model()的具体实现如下所示：

```
// fm_model类的构造函数
fm_model::fm_model() {
    num_factor = 0; // 交叉项中因子的个数
    init_mean = 0; // 初始化的均值
    init_stdev = 0.01; // 初始化的方差
    reg0 = 0.0; // 常数项的正则化参数
    regw = 0.0; // 一次项的正则化参数
    regv = 0.0; // 交叉项的正则化参数
    k0 = true; // 是否包含常数项
    k1 = true; // 是否包含一次项
}
```

init()函数的具体实现如下所示：

```
// 初始化fm模型的函数
void fm_model::init() {
    w0 = 0; // 常数项的系数
    w.setSize(num_attribute); // 设置一次项系数的个数
    v.setSize(num_factor, num_attribute); // 设置交叉项的矩阵大小
    w.init(0); // 初始化一次项系数为0
    v.init(init_mean, init_stdev); // 按照均值和方差初始化交叉项系数
    // 交叉项中的两个参数，设置其大小为num_factor
    m_sum.setSize(num_factor);
    m_sum_sqr.setSize(num_factor);
}
```

在初始化的过程中，除了基本的数据类型外，还涉及到自定义的三种数据类型，分别为：DVectorDouble，DMatrixDouble和DVector，这三种数据类型在“机器学习算法实现解析——libFM之libFM的数据处理部分”中有详细说明。

3.3、利用FM模型对样本进行预测

在libFM中，fm_model类中实现了两种预测函数，分别为：

```
double predict(sparse_row<FM_FLOAT>& x);
double predict(sparse_row<FM_FLOAT>& x, DVector<double> &sum, DVector<double>
```

两者的区别主要是下面的函数多了两个参数，一个是sum，另一个是sum_sqr，这两个参数分别对应着交叉项计算过程中的两项。

FM模型中的计算方法为：

$$\hat{y} := w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

其中，对于交叉项的计算，在FM算法中提出了快速的计算方法，即：

$$\begin{aligned}
& \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \\
&= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j - \frac{1}{2} \sum_{i=1}^n \langle \mathbf{v}_i, \mathbf{v}_i \rangle x_i x_i \\
&= \frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n \sum_{f=1}^k v_{i,f} v_{j,f} x_i x_j - \sum_{i=1}^n \sum_{f=1}^k v_{i,f} v_{j,f} x_i^2 \right) \\
&= \frac{1}{2} \sum_{f=1}^k \left(\left(\sum_{i=1}^n v_{i,f} x_i \right) \cdot \left(\sum_{j=1}^n v_{j,f} x_j \right) - \sum_{i=1}^n v_{i,f}^2 x_i^2 \right) \\
&= \frac{1}{2} \sum_{f=1}^k \left(\left(\sum_{i=1}^n v_{i,f} x_i \right)^2 - \sum_{i=1}^n v_{i,f}^2 x_i^2 \right)
\end{aligned}$$

利用上面的计算公式，libFM中的两个函数的实现如下所示：

```

// 对样本进行预测，其中x表示的是一行样本
double fm_model::predict(sparse_row<FM_FLOAT>& x) {
    return predict(x, m_sum, m_sum_sqr);
}

double fm_model::predict(sparse_row<FM_FLOAT>& x, DVector<double> &sum, DVect
double result = 0; // 最终的结果
// 第一部分
if (k0) { // 常数项
    result += w0;
}

// 第二部分
if (k1) { // 一次项
    for (uint i = 0; i < x.size; i++) { // 对样本中的每一个特征
        assert(x.data[i].id < num_attribute); // 验证样本的正确性
        // w * x
        result += w(x.data[i].id) * x.data[i].value;
    }
}

// 第三部分
// 交叉项，对应着公式，有两重循环
for (int f = 0; f < num_factor; f++) { // 外层循环
    sum(f) = 0;
    sum_sqr(f) = 0;
    for (uint i = 0; i < x.size; i++) {
        double d = v(f, x.data[i].id) * x.data[i].value;
        sum(f) += d;
        sum_sqr(f) += d*d;
    }
    result += 0.5 * (sum(f)*sum(f) - sum_sqr(f)); // 得到交叉项的值
}
return result;
}

```

在交叉项的计算过程中，sum(f)和sum_sqr(f)与公式中的对应关系为：

$$\text{sum}(f) \rightarrow \sum_{i=1}^n v_{i,f} x_i$$

$$\text{sum_sqr}(f) \rightarrow \sum_{i=1}^n v_{i,f}^2 x_i^2$$

3.4、其他

剩下的代码便是debug函数，debug函数用于打印中间的结果，其具体的代码如下所示：

```
// debug函数，主要用于输出中间调试的结果
void fm_model::debug() {
    std::cout << "num_attributes=" << num_attribute << std::endl; // 特征的个数
    std::cout << "use w0=" << k0 << std::endl; // 是否包含常数项
    std::cout << "use w1=" << k1 << std::endl; // 是否包含一次项
    std::cout << "dim v =" << num_factor << std::endl; // 交叉项中因子的个数
    std::cout << "reg_w0=" << reg0 << std::endl; // 常数项的正则化参数
    std::cout << "reg_w=" << regw << std::endl; // 一次项的正则化参数
    std::cout << "reg_v=" << regv << std::endl; // 交叉项的正则化参数
    std::cout << "init ~ N(" << init_mean << "," << init_stdev << ")" << std::endl;
}
```

参考文献

Rendle S. Factorization Machines[C]// IEEE International Conference on Data Mining. IEEE Computer Society, 2010:995-1000.

Rendle S. Factorization Machines with libFM[M]. ACM, 2012.

读者评论

请登录后发表评论

邮箱

密码

登 录

注册 (/register)

相关博文

社区使用反馈专区 (/article/20)

陈晓猛 (/space/index/8008) 2016-10-04

尊敬的博文视点用户您好： 欢迎您访问本站，您在本站点访问过程中遇到任何问题，均可以在本页留言，我们会根据您的意见和建议，对网站进行不断的优化和改进，给您带来更好的访问体验！ 同时，您被采纳的意见和建议，管理员也会赠送您相应的积分...

2240 392 2 5

迎战“双12”！《Unity3D实战核心技术详解》独家预售开启！ (/article/70)

陈晓猛 (/space/index/8008) 2016-12-05

时隔一周，让大家时刻挂念的《Unity3D实战核心技术详解》终于开放预售啦！ 这本书不仅满足了很多年轻人的学习欲望，并且与实际开发相结合，能够解决工作中真实遇到的问题。预售期间优惠多多，实在不容错过！ Unity 3D实战核心技术详解 ...

1846 34 0 1

新书试读员征集 (/article/67)

陈晓猛 (/space/index/8008) 2016-12-01

活动获奖名单公布 恭喜以下10位读者朋友获得本次试读员征集的名额，请私信 陈晓猛 将您的QQ号发给小编，以便加入读者VIP群，给您送出图书！ 获奖名单 程续缘 悠悠的一杯茶 313150284@qq.com llaomao 落伍特青年 ...

665 12 0 0



微信公众号



微博