推荐算法与量化交易-A-5-6：FM算法-“自动特征组合”算法-基于模型算法

原创 崔啊 推荐算法与量化交易 2018-12-12

推荐系统回顾：

推荐算法与量化交易-A-0：推荐算法概述

推荐系统与量化交易-A-2：基于矩阵分解的推荐算法（SVD+FunkSVD+BaisSVD+SVD++ +timeSVD++）

推荐算法与量化交易-A-5-1：基于模型算法的“广义线性模型”推荐系统算法（线性模型+LR)

推荐算法与量化交易-A-5-5：XGBoost-lightGBM“集成提升树模型”算法-基于模型算法

本节内容：

一、FM算法

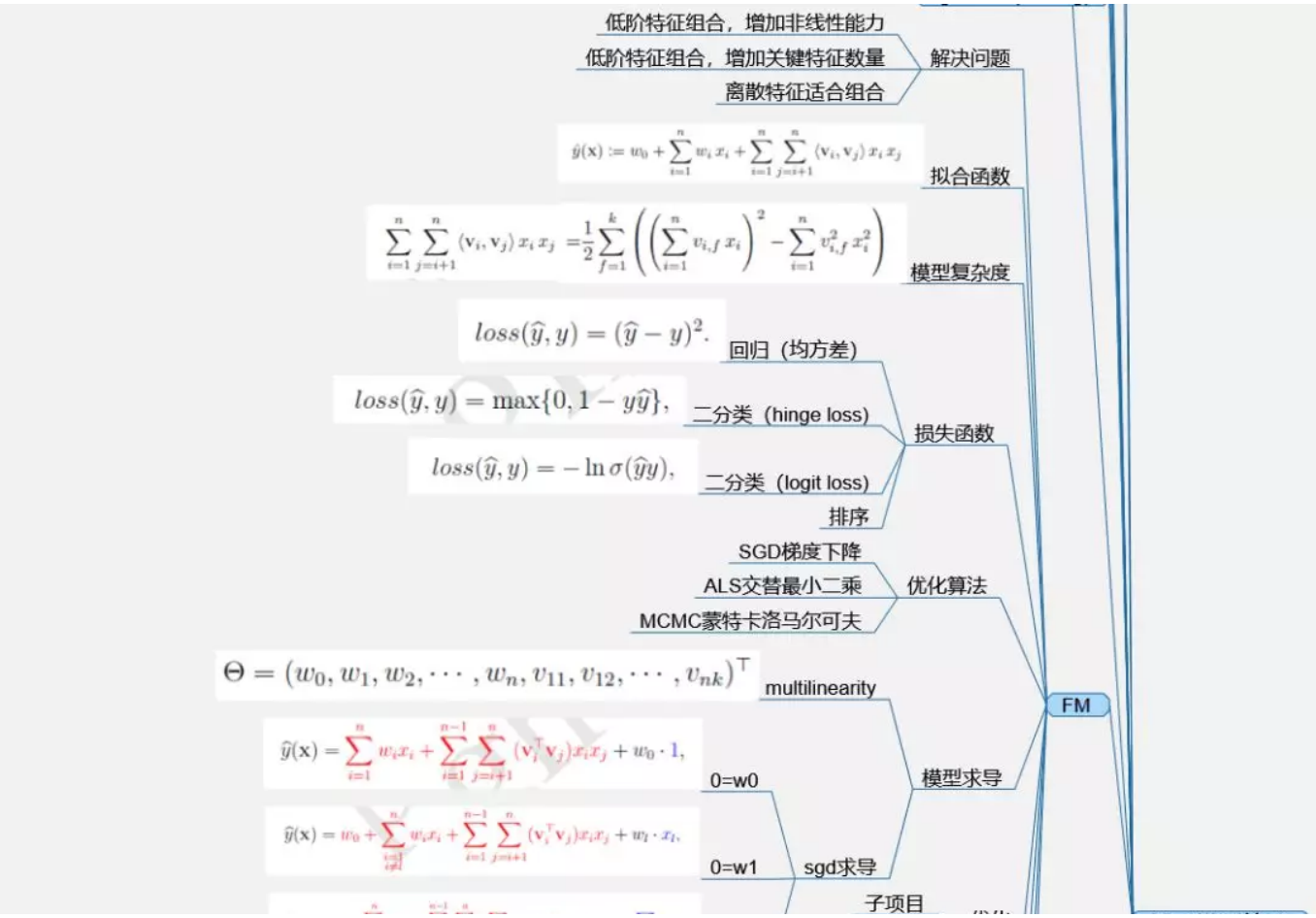
1，FM法的拟合函数理论推导

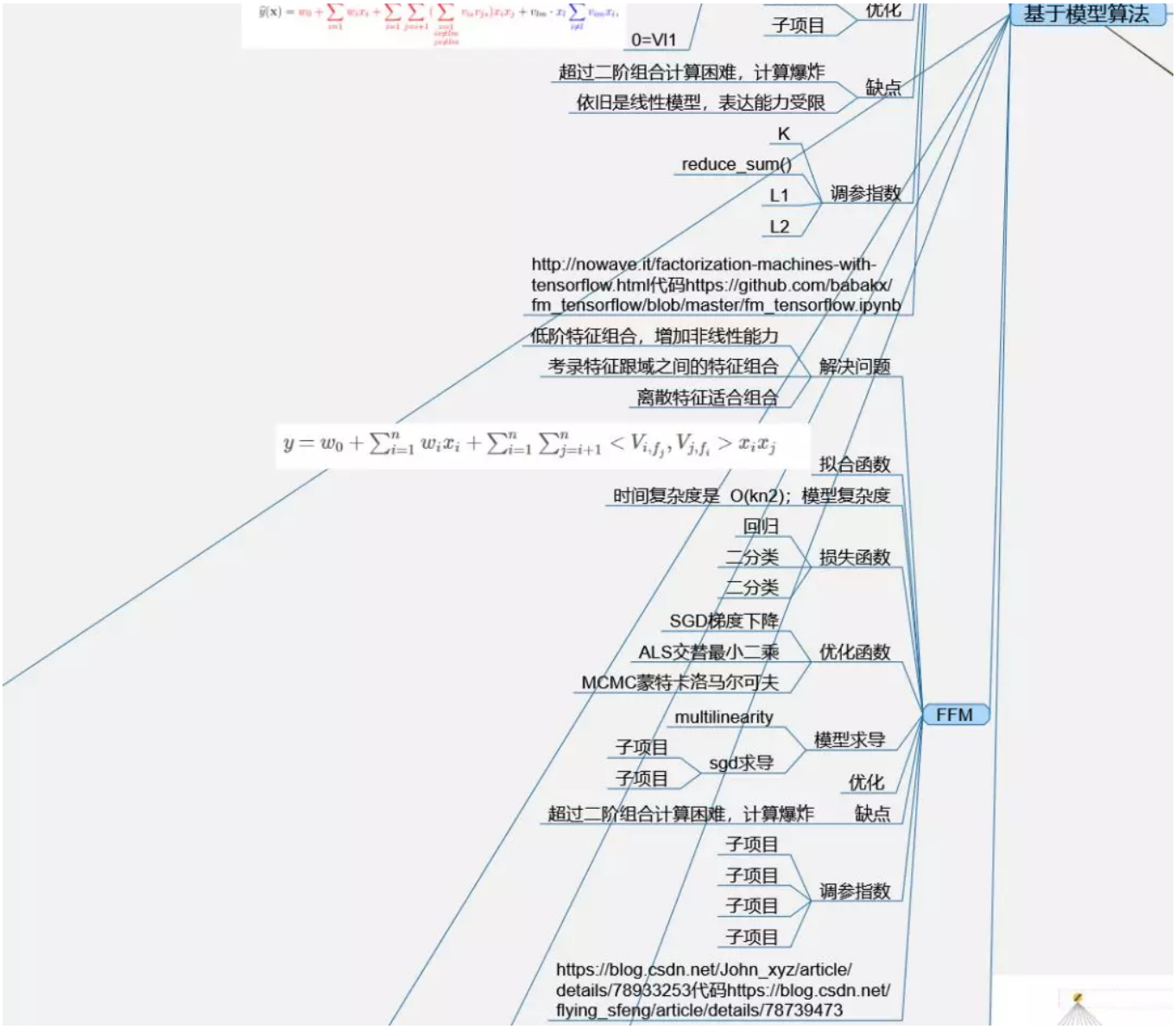
2，FM的分类算法损失函数

3，FM的回归算法损失函数

4，FM的梯度求解方式

5，FM算法的优点与缺点





其实谈到本章节的FM算法和FFM算法，都是在kaggle比赛而来的，似乎和另一个矩阵分解算法一样，也是因为kaggle比赛而来的，那么我们就讲一下为什么提出了FM算法。

首先，说行一下，Kaggle比赛多以模式识别为主，而且是多以计算广告的CTR为主，这种CTR的计算广告比赛会放出几个文件，一般的是APP文件（用户下载了那些APP及具体数据），user文件（用户文件，年龄，住址，性别……），AD文件（广告文件，广告信息，广告位，广告主.....）,这样的几个文件，一般根据用户自己的选择模型和方式自己去把几个表格融合在一起，那么，这些特征，一般会经过大家的特征工程处理后，把连续特征分桶，然后再哑变量处理，这样就是离散特征，包括绝大部分的这种DataFrame格式的文件都会这种处理成离散特征的形式，那么针对于离散特征，一般的厂家都会放出一个Baseline版本（基于LR算法或者Linear算法）的结果，一般都是以logloss作为输出，而采用的特征都只有user这一项表格里的数据。

那么这种厂家以LR为计算模型，只采用一个USER表格数据做logloss输出的Baseline计算结果，参赛的人会花费大量的时间把各种不同表格数据融合在一起，然后再连续特征离散化，然后再哑变量，这才是初步特征工程。更复杂的是需要人对业务有深刻的理解，然后自己去组合特征，找到特征与特征之间的关系，再组合成复杂的特征，放进数据表里，继续再用Linear做提升，毕竟“数据决定模型上限”，但是，有一个关键的问题，在实际业务中还好处理，但是在kaggle比赛中不好处理是因为，所有数据都脱敏啦，你根本不知道这个数据表示含义。

这个时候参赛的台湾大学的阮毓钦就提出了再Linear算法的基础上可以增加自动特征组合的特性，去寻找特征间的隐性关系，来提升计算效果，在实际应用中效果非常好。所以，自动组合特征的以FM算法为最简单基础的一些列算法应运而生，我们接下来介绍的这都是FM及变体算法，这类算法有很多好处，如：

①，减少了人工特征工程的时间和工作难度，由算法内部自己完成

②，降低了人们对业务的强硬性的理解和深入，可以降低对人的业务线能力

③，增加了数据内部的非线性发掘能力，同时又有数学基础

④，本身的低阶展示数据增加模型拟合能力，增加了自动组合的数据又增加了模型的发觉及泛华能力

⑤，输出结果提升幅度大，模型难度低，计算框架实施简单。

一、FM算法：

1-1，FM算法的拟合函数：

经过上面的介绍，我们知道FM的算法，其实就是在Linear的算法拟合函数上增加了特征的自动组合计算实现，那需要回顾Linear算法的请学习：《推荐算法与量化交易-A-5-1：基于模型算法的“广义线性模型”推荐系统算法（线性模型+LR)》，这里只对Linear的公式应用做表述：

①Linear线性模型的拟合函数：

$$y = w_0 + \sum_{i=1}^{n} w_i x_i$$

注释：上面公式可以看出只有一个x_i，那么这个特征还是离散的，特征与与特征间是没有关联的，但是实际中，特征之间是有关联的，是相互作用的

②因为FM算法就是为了增加特征之间关联的，用以学习出特征之间的非线性表达能力，而又是在Linear算法的拟合函数基础上更改得到的，那么，FM拟合函数表达式定义为：

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} * \sum_{j=i+1}^{n} w_{ij} x_i x_j$$

注释：多个特征关联肯定是多项式关联，特征x_i与特征x_j的组合用$x_i x_j$进行表示，，因为多项式组合计算这种离散特征会造成“维度灾难”，所以FM主要以二项式组合为主，我们介绍也以二项式组合讲解

③模型复杂度：可以从上面的FM拟合函数可以看出，仅仅是二项式特征的自动组合，那么这个组合相应的其实在增加我们的新的特征表达数量，而且增加的数据量的数量是：$\frac{n(n-1)}{2}$个，如果是多项式，那么增加的新的特征数量会更多，组合这些特征就会花费巨大的时间，特征是离散的，而且是哑变量的形式，那么能够相互共存的情况非常少，计算出相互共存的的情况是需要花费巨大时间，如何降低时间是优化计算的关键。

经过的上面FM拟合公式，公式新增的是一个w_{ij}参数，那就是证明只有在特征x_i与特征x_j都存在的情况下这个参数才有效，其余的都是无效的，而计算这个参数寻找这个参数才是花费巨大事件的关键，同时，求解这个个w_{ij}参数，也是非常巨大事件的，只有把w_{ij}这个参数解决掉才能够降低时间消耗，所以，这里我们需要引入一个辅助

参数$V_i = [v_{i1}, v_{i2}, \ldots, v_{ik}]^T$，把一个权重拆解为两个部分，每个部分只与自己的哑变量特征相关，那么利用：$W_{n\times n} = V_{n\times k}V_{n\times k}^T$可以进行转换：

因为：

$$\mathbf{V} = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1k} \\ v_{21} & v_{22} & \cdots & v_{2k} \\ \vdots & \vdots & & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nk} \end{pmatrix}_{n\times k} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_n \end{pmatrix}$$

那么：

$$\hat{\mathbf{W}} = \mathbf{V}\mathbf{V}^T = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_n \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^T & \mathbf{v}_2^T & \cdots & \mathbf{v}_n^T \end{pmatrix}$$

所以，FM的拟合公式进过辅助变量可以变化为：

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} * \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$$

其中：

$$< v_i, v_j >= \sum_{f=1}^{k} v_{if} v_{jf}$$

注释：这里涉及到一个参数K，在计算框架里会需要设置，代表了的就是辅助变量的长度，也可以理解为模型的表达能力

这时候的计算出来的FM模型复杂度是$O(kn^2)$,经过上面的计算w_{ij}展开，我们知道那么其实$\sum_{i=1}^{n} * \sum_{j=1}^{n} < V_i, V_j > x_i x_j$构成的是一个完整的对称矩阵，那么$\sum_{i=1}^{n} * \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$是这个对称矩阵的上三角部分（不包含对角线）,所以$\sum_{i=1}^{n} * \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$等于$\sum_{i=1}^{n} * \sum_{j=1}^{n} < V_i, V_j > x_i x_j$减去对角线再除以2，则有以下有推算过程：

$$\sum_{i=1}^{n} * \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$$

$$= \frac{1}{2} \sum_{i=1}^{n} * \sum_{j=1}^{n} < V_i, V_j > x_i x_j - \frac{1}{2} \sum_{i=1}^{n} < V_i, V_i > x_i x_i$$

$$= \frac{1}{2} (\sum_{i=1}^{n} * \sum_{j=1}^{n} * \sum_{f=1}^{k} v_{if} v_{jf} x_i x_j - \sum_{i=1}^{n} * \sum_{f=1}^{k} v_{if} v_{if} x_i x_i)$$

$$= \frac{1}{2} \sum_{f=1}^{k} * ((\sum_{i=1}^{n} v_{if} x_i)(\sum_{j=1}^{n} v_{jf} x_j) - \sum_{i=1}^{n} v_{if}^2 x_i^2)$$

$$= \frac{1}{2} \sum_{f=1}^{k} * ((\sum_{i=1}^{n} v_{if} x_i)^2 - \sum_{i=1}^{n} v_{if}^2 x_i^2))$$

注释：也可以理解为$(a+b+c)^2 - a^2 - b^2 - c^2$求交叉项

通过对每个特征引入辅助变量 Vi, 并对表达式进行化简，可以把时间复杂度降低到O(kn)

④这个时候拟合函数变化为：

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + + \frac{1}{2} \sum_{f=1}^{k} * ((\sum_{i=1}^{n} v_{if} x_i)^2 - \sum_{i=1}^{n} v_{if}^2 x_i^2))$$

以上介绍的是拟合函数，但是为了降低时间复杂化度，我们引入了辅助变量，为了降低时间复杂度，有了一个最低时间复杂度的拟合函数表示，但是FM算法是一个监督学习，同时又是一个既可以做分类算法，也可以做回归算法的方式，因为FM是基于线性拟合函数做的扩展，那么它自然是存在损失函数的，针对于不同问题，选择不同损失函数，优化损失函数进行求解，求解公式的参数就是最优解，那么下面分别介绍分类和回归算法。

2-2，FM的分类算法损失函数：

经过上面拟合函数介绍，现在应该进入的就是损失函数这个环节，现在进行分类算法的损失函数介绍。

在广义线性模型算法介绍了，线性模型想要做分类，那么就是把现行的拟合函数经过sigmoid激活函数非线性变换，然后再用对数损失就可以做分类，那么这里也是一样的，毕竟拟合函数只是增加了个二项式组合特征，其余原理未变 。

所以，FM的分类算法的损失函数的一种表达方式就是：

①对数损失函数logloss:拟合函数经过sigmoid激活函数激活后，用做对数似然，就是损失函数,表达式为：

▶首先对我们的上面拟合函数做非线性sigmoid转换:

$$\hat{y}, = \frac{1}{1+e^{-\{w_0 + \sum_{i=1}^{n} w_i x_i + + \frac{1}{2} \sum_{f=1}^{k} * \left(\left(\sum_{i=1}^{n} v_{if} x_i \right)^2 - \sum_{i=1}^{n} v_{if}^2 x_i^2 \right) \right\}}}$$

▶其次做对数似然展开：

$$\text{Logloss}（\hat{y},\text{y}）=-\ln\sigma(\hat{y}\text{y})=-\sum_{i=1}^{m}(y^{(i)}log(\hat{y},)+(1-y^{(i)})log(1-\hat{y},))$$

注释：这里依然是以2分类为例子计算，那么y{0.1},只有在激活函数是sigmoid函数时取值才是y{0,1},其余情况均为y{-1，+1}

②基于这种广义线性的分类损失函数,除去可以用LR的对数似然损失，其实还可以用SVM支持支持向量机的hinge loss合页损失函数，那么表述形式就是：

$$\text{Logloss}（\hat{y},\text{y}）=\max\{0,1-\hat{y}\text{y}\}$$

当然，根据支持向量机的情况，也做二分类展示，那么y{-1,1}

当y=+1时的情况是：

$$\text{Logloss}（\hat{y},\text{y}）=\max\{0,1-\hat{y}\}=\begin{cases}0, & \hat{y}\geq 1\\ 1-\hat{y}, & \hat{y}<1\end{cases}$$

当y=-1时的情况是：

$$\text{Logloss}（\hat{y},\text{y}）=\max\{0,1+\hat{y}\}=\begin{cases}0, & \hat{y}\leq -1\\ 1+\hat{y}, & \hat{y}>-1\end{cases}$$

2-3，FM的回归算法损失函数：

经过上面拟合函数介绍，现在应该进入的就是损失函数这个环节，现在进行分类算法的损失函数介绍。

我们在广义线性模型算法介绍了，线性模型想要做回归，其实有两种损失函数可以介绍，一种是均方差损失，另一种是绝对值损失，这里以线性回归的均方差（MSE）损失为主进行介绍：

$$\text{Logloss}(\hat{y},\text{y})=（\hat{y}-\text{y}）^2=\{w_0+\sum_{i=1}^{n}w_ix_i++\frac{1}{2}\sum_{f=1}^{k}*((\sum_{i=1}^{n}v_{if}x_i)^2-\sum_{i=1}^{n}v_{if}^2x_i^2))-\text{y}\}^2$$

以上就是分对于FM算法，分别针对于回归和分类问题的损失函数，如果你前面认真仔细看了系列文章，那么，就对损失函数在应用上会变通了。

2-4，FM的梯度求解方式：

无论是FM的分类还是回归算法，都有对应的损失函数，损失函数主要以：均方差损失mse,对数似然损失logloss，合页损失logloss，在介绍“广义线性模型”时，这几种损失函数方式都可以用：随机梯度下降(SGD)的方法求解。当然这只是一个方法，因为求解方式很多类似于线性回归的求解方式可以采用{最小二乘，随机梯度下降}并存状态一样。

①随机梯度下降求解（SGD）：

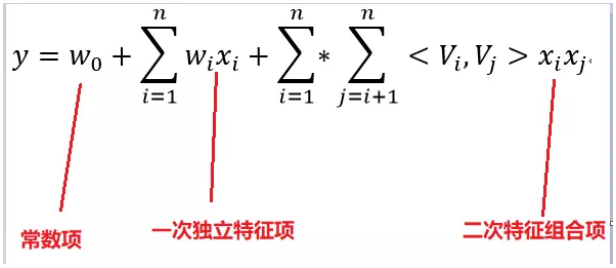
进行随机梯度下降算法之前，再次看一下拟合函数：

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} * \sum_{j=i+1}^{n} < V_i, V_j > x_i x_j$$

进过最简单的拟合函数，显现里面关于“权重”的系数比较多，以简单的二项式组合为例，我们发现有w_0，w_i，v_i，v_j四个参数需要求解，如果更多项式组合，那么导致的更多的参数。所以，把这个参数归成一个集合，然后求解比较容易。所以这里引入一个概念Multilinearity,就是代表FM模型未知参数的合集，表示为：

$$\Theta = (w_0, w_1, w_2, \ldots\ldots w_n, v_{11}, v_{12}, \ldots . v_{nk})^T$$

再一次，观察拟合函数，其实拟合函数由三部分组成：常数项，一次独立特征项，二次组合特征项。



对于这种三个项式组成，求解参数，那么我们其实需要划分为两类，马上求解的参数的项算为一类，其余的所有算为另一类，那么其实就是可以表达成，对于任意的$\theta \in \Theta$，就会存在两个函数项一个与θ有关，另一个无关，则表达式为：

$$\hat{y}(\mathbf{x}) = g_\theta(\mathbf{x}) + \theta h_\theta(\mathbf{x})$$

那么，我们用Multilinearity的思维和项式分解再次理解拟合函数就是：

▶当$\theta = w_0$的时候，

根据上面拆分为两个部分的原则求解，则拟合函数变化为：

$$\hat{y}(x) = w_0 * 1 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} * \sum_{j=i+1}^{n} < V_i^T, V_j > x_i x_j$$

注释：蓝色部分表示的就是参数相关部分，红色色部分表示的就是参数不相关部分

▶当$\theta = w_l (l = 1,2 \ldots . . n)$的时候，

根据上面拆分为两个部分的原则求解，则拟合函数变化为：

$$\hat{y}(x) = w_0 + \sum_{\substack{i=1 \\ i \neq l}}^{n} w_i x_i + \sum_{i=1}^{n} * \sum_{j=i+1}^{n} < V_i^T, V_j > x_i x_j + w_l * x_l$$

注释：蓝色部分表示的就是参数相关部分，红色色部分表示的就是参数不相关部分

▶当$\theta = V_{lm}$的时候，

根据上面拆分为两个部分的原则求解，则拟合函数变化为：

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} * \sum_{j=i+1}^{n} < \sum_{\substack{s=1 \\ is \neq lm \\ js \neq lm}} v_{is} v_{js} > x_i x_j + v_{lm} * x_l \sum_{i \neq l} v_{im} x_i$$

注释：蓝色部分表示的就是参数相关部分，红色色部分表示的就是参数不相关部分

我们上面对拟合函数用了$\hat{y}(\mathrm{x}) = g_\theta(\mathrm{x}) + \theta h_\theta(\mathrm{x})$进行不同内部参数的二分的选择情况，那么其实$h_\theta(\mathrm{x})$部分最好求解，求导也简单，但是到了$g_\theta(\mathrm{x})$这一部分，因为涉及到的项式比较多，求解特别麻烦，这里我们有一个独特的方法可以应用，就是上一节讲解了lightGBM时候用的“直方图差速法”就可以就行迅速求解$g_\theta(\mathrm{x}) = \hat{y}(\mathrm{x}) - h_\theta(\mathrm{x})$即可。

那么，我们就可以知道了，针对于拟合函数$\hat{y}(\mathrm{x})$针对于未知函数θ在不同状态下的导数是一个合集，而且不同状态可以表述为：

$$\frac{\partial \hat{y}(\mathrm{x})}{\partial \theta} = \begin{cases} 1, & \theta = w_0; \\ x_l, & \theta = w_l,\ l = 1, 2, \cdots, n; \\ x_l \sum\limits_{\substack{s=1 \\ s \neq l}}^{n} v_{sm} x_s, & \theta = v_{lm},\ l = 1, 2, \cdots, n;\ m = 1, 2, \cdots, k, \end{cases}$$

注解：因为$\theta \in \Theta$，并且Θ是一个集合，所以求导出来针对不同选择，其实也是一个集合状态

经过以上介绍，其实$\theta \in \Theta$选项只与我们的$\hat{y}(\mathrm{x}) = g_\theta(\mathrm{x}) + \theta h_\theta(\mathrm{x})$二分区别项里面的$h_\theta(\mathrm{x})$正相关，所以可以知道：

$$h_\theta(\mathrm{x}) = \frac{\partial \hat{y}(\mathrm{x})}{\partial \theta}$$

★★梯度下降的最优解求法：

上面的是讲解了如何对拟合函数的设置不同未知参数集合的考录情况，但是，FM是存在损失函数的，根本问题还是需要求解损失函数的，用拟合函数的求导来进行求解未知“权重”，现在进行的就是这一步

无论是回归问题还是分类问题，其实拟合函数都可以简化表示成：

$$L = \sum_{i=1}^{N} loss(\hat{y}(\mathrm{x}^{(i)}), y^{(i)})$$

注释：具体细分到分类问题，损失函数为对数损失、合页损失，针对于回归问题损失函数为均方差损失，公式不再具体进行表述

无论基于那种损失函数，但是这几种损失函数都可以用随机梯度下降进行求解，那么，随机梯度下降求解是负梯度求解损失函数的最小值，才可以求解出未知参数，最小化损失求导变为：

$$\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{N} loss(\hat{y}(\mathrm{x}^{(i)}), y^{(i)})$$

★★★正则化过拟合问题：

因为我们知道FM的拟合函数是基于Linear算法的，二损失函数其实也是基于广义线性模型的损失函数，无论是Linear和LR，防止过拟合的的办法都是L1/L2正则化，最常用的是L2正则化，因为L1正则化会去除近零项，是一种奇特的“降维”方式，这里我们以L2做演示，那么，损失函数的表达就是：

$$\Theta^* = \arg\min_{\Theta} \sum_{i=1}^{N} \left(loss(\hat{y}(\mathbf{x}^{(i)}), y^{(i)}) + \sum_{\theta \in \Theta} \lambda_\theta \theta^2 \right)$$

注释：当然，λ_θ代表的是针对于特定θ状态的学习率，所以，其实λ_θ也应该是一个集合状态的。，也是同θ一样，需要针对于情况来分组设定，分组学习。

那么，λ_θ状态细分其实是根据w_0，w_i，v_i，v_j变化的，所以，λ_θ可以表示成：

λ^0，$\lambda^w_{\pi(i)}$，$\lambda^v_{\pi(i),j}$，$i \in \{1, 2, \cdots, n\}$，$j \in \{1, 2, \cdots, k\}$，当然，依旧是分为三组情况考录状态。

★★★★求导参数更新：

经过以上可知，我们针对于回归问题利用“随机梯度下降法则”的求导可以表述成：

$$\frac{\partial loss^R(\hat{y}(\mathbf{x}), y)}{\partial \theta} = 2(\hat{y}(\mathbf{x}) - y)\frac{\partial \hat{y}(\mathbf{x})}{\partial \theta}$$

经过以上可知，我们针对于分类问题利用“随机梯度下降法则”的求导可以表述成：

$$\begin{aligned} \frac{\partial loss^C(\hat{y}(\mathbf{x}), y)}{\partial \theta} &= -\frac{1}{\sigma(\hat{y}(\mathbf{x})y)}\sigma(\hat{y}(\mathbf{x})y)[1 - \sigma(\hat{y}(\mathbf{x})y)]\frac{\partial \hat{y}(\mathbf{x})}{\partial \theta}y \\ &= [\sigma(\hat{y}(\mathbf{x})y) - 1]y\frac{\partial \hat{y}(\mathbf{x})}{\partial \theta} \end{aligned}$$

所以最终求解的未知参数可以表述成：

$$w_0 = w_0 - \eta\left(\frac{\partial loss(\hat{y}(\mathbf{x}), y)}{\partial w_0} + 2\lambda^0 w_0\right)$$

$$w_i = w_i - \eta\left(\frac{\partial loss(\hat{y}(\mathbf{x}), y)}{\partial w_i} + 2\lambda^w_{\pi(i)} w_i\right)$$

$$v_{ij} = v_{ij} - \eta\left(\frac{\partial loss(\hat{y}(\mathbf{x}), y)}{\partial v_{ij}} + 2\lambda^v_{\pi(i),j} v_{ij}\right)$$

②交替最小二乘法求解（ALS）：

在介绍“线性回归”得时候，介绍了两种求解方法：{随机梯度下降，最小二乘}，那么，针对于FM的回归问题，只是拟合函数稍微变化，损失函数没有变化，其实FM的回归问题问题依旧可以选择“交替最小二乘法”进行求解，这里不做更多介绍，具体，请查阅《推荐算法与量化交易-A-5-1：基于模型算法的“广义线性模型”推荐系统算法（线性模型+LR)》学习最小二乘法则。

1-5，FM算法的优点与缺点

优点	1，增加特征交叉，增加非线性表达能力 2，可以进行正则化防止过拟合 3，学习理论基础扎实 4，减少认为特征工程，减少时间
缺点	1，特征需要归一化 2，只能进行二元交叉，多元计算难度大

整体注释：

FM算法仅仅是“广义线性模型算法”的理论升级，但是实际却带来了质的提升，不同于“集成策略学习”法则，它让人拜托了繁杂的特征工程，并且在特征工程中增加特征之间的关联工程，并且，他又不同于纯DNN的黑盒理论，具有极强的数学理论基础，后人们为了更加完善的去思考特征之间的关联，特征之间的非线性表达能力，所以在此基础上发展了FFM,deepFM,XdeepFM算法，后面几章我们会分别对此介绍，因为,deepFM需要深度学习的DNN知识，我们会把深度学习的DNN知识增加上来，至于基于时间序列RNN网络和基于空间序列的CNN网络，我们会在介绍完推荐系统知识进行纯深度学习章节算法的介绍

您的每一次批评，是我们共同进步的阶梯。