在线最优化求解
(Online Optimization)

冯扬 (@fengyoung)
新浪微博-商业平台及产品部-推荐引擎
2014-12-09

摘要：最优化求解问题可能是我们在工作中遇到的最多的一类问题了：从已有的数据中提炼出最适合的模型参数，从而对未知的数据进行预测。当我们面对高维高数据量的场景时，常见的批量处理的方式已经显得力不从心，需要有在线处理的方法来解决此类问题。本文以模型的稀疏性作为主线，逐一介绍几个在线最优化求解算法，并进行推导，力求讲清楚算法的来龙去脉，以及不同算法之间的区别和联系，达到融会贯通。在各个算法原理介绍之后，都会给出该算法的工程实现伪代码，可以用于实际工作的参考。

1 动机与目的

在实际工作中，无论是工程师、项目经理、产品同学都会经常讨论一类话题：“从线上对比的效果来看，某某特征或因素对 xx 产品的最终效果有很大的影响”。这类话题本质上说的是通过已有的数据反映出某些特定的因素对结果有很强的正（或负）相关性。而如何定量计算这种相关性？如何得到一套模型参数能够使得效果达到最优？这就是最优化计算要做的事情。

举一类典型点的例子：在推荐和广告计算中，我们经常会需要对某些值进行预测，例如在一条推荐或广告在曝光之前先预测用户是否会产生点击（CTR 预估），或者是否会由此产生某些转换（RPM 预估）。这类问题可以表示为：针对一个输入$X = [x_1, x_2 \dots x_N] \in \mathbb{R}^N$，通过某个函数$h(X)$计算（预测）输出$y \in \mathbb{R}$。根据$y$值为连续的还是离散的，预测问题被划分成回归问题（Regression）和分类问题（Classification）。而利用已有的样本数据$\{(X_j, y_j)|j = 1,2, \dots, M\}$训练$h(X)$的过程往往转换成一个最优化求解的过程。

无论是线性回归（Linear Regression）、逻辑回归（Logistic Regression）、支持向量机（SVM）、深度学习（Deep Learning）中，最优化求解都是基本的步骤。常见的梯度下降、牛顿法、拟牛顿法等属于批量处理的方法（Batch），每次更新都需要对已经训练过的样本重新训练一遍。而当我们面对高维高数据量的时候，批量处理的方式就显得笨重和不够高效，因此需要有在线处理的方法（Online）来解决相同的问题。关于在线最优化问题（Online Optimization）的论文比较多，逐一查找阅读费时费力，那么本文就以高维高数据量的应用场景中比较看重的稀疏性作为主线，来介绍一些在线最优化的方法。

本文的预期读者大概有如下几类：

1. **具有很深机器学习经验和背景的高阶人员：**就拿这篇文章当作一个关于在线最优化算法的回顾材料好了，如有错误和不足欢迎指正。
2. **具有一定机器学习经验的中级读者：**可以将本文作为一个理论资料进行阅读，略过“预备知识”部分，直接进入主题，将之前对于在线最优化算法的理解串联起来，希望能对将来的工作提供帮助。
3. **对机器学习有认识但是实践经验较少的初级读者：**从预备知识看起，逐一理解相关概念和方法，从而达到融会贯通的目的。
4. **仅仅对算法的工程实现感兴趣的读者：**大致浏览一下预备知识中的 2.3 节，了解我

们要讨论什么，然后直奔各算法的算法逻辑（伪代码），照着实现就 ok 鸟。

5. **高富帅和白富美：**只需要知道本文讨论的是一堆好用的求最优解的方法，可以用于分类预测、回归预测等一系列问题。然后吩咐攻城狮去实践就好了。还可以拿这篇文章嘲笑机器学习的屌丝：看你们都弄些啥，累死累活的，挣那么几个钢镚☺

2 预备知识

2.1 凸函数

如果$f(X)$是定义在 N 维向量空间上的实值函数，对于在$f(X)$的定义域C上的任意两个点X_1和X_2，以及任意[0,1]之间的值t都有：

$$f(tX_1+(1-t)X_2) \le tf(X_1)+(1-t)f(X_2)$$
$$\forall X_1, X_2 \in C, \quad 0 \le t \le 1$$

那么称$f(X)$是凸函数（Convex）[1]。一个函数是凸函数是它存在最优解的充分必要条件。

此外，如果$f(X)$满足：

$$f(tX_1+(1-t)X_2) < tf(X_1)+(1-t)f(X_2)$$
$$\forall X_1, X_2 \in C, \quad 0 < t < 1$$

则$f(X)$是严格凸函数（Strict Convex）。如图 1 所示，(a)为严格凸函数，(b)为凸函数。

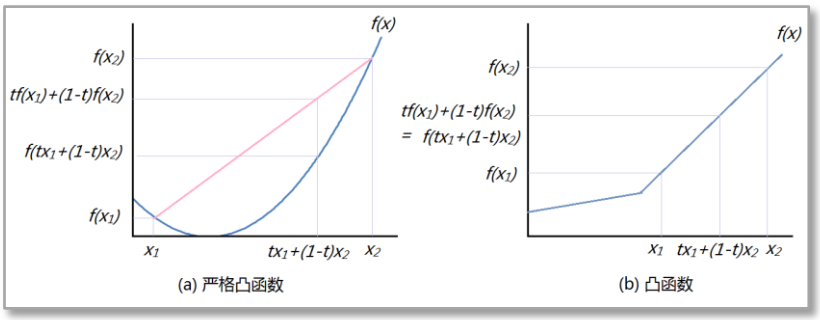

图 1 凸函数

2.2 拉格朗日乘数法及 KKT 条件

通常我们需要求解的最优化问题有如下三类：

(1) 无约束优化问题：。

$$X = \arg\min_X f(X)$$

含义是求解X，令目标函数$f(X)$最小；

(2) 有等式约束的优化问题：

$$X = \arg\min_X f(X)$$

$$s.t. \quad h_k(X) = 0;\ k = 1,2 \ldots, n$$

含义是在 n 个等式约束$\{h_k(X)\}$的条件下，求解X，令目标函数$f(X)$最小。

(3) 有不等式约束的优化问题：

$$X = \underset{X}{argmin}\, f(X)$$

$$\boldsymbol{s.t.} \quad \begin{matrix} h_k(X) = 0;\ k = 1,2 \dots, n \\ g_l(X) \le 0;\ l = 1,2 \dots, m \end{matrix}$$

含义是在 n 个等式约束$\{h_k(X)\}$以及 m 个不等式约束$\{g_l(X)\}$的条件下，求解X，令目标函数$f(X)$最小。

针对无约束最优化问题，通常做法就是对$f(X)$求导，并令$\frac{\partial}{\partial X} f(X) = 0$，求解可以得到最优值。如果$f(X)$为凸函数，则可以保证结果为全局最优解。

针对有等式约束的最优化问题，采用**拉格朗日乘数法（Lagrange Multiplier）**[2]进行求解：通过拉格朗日系数$A = [a_1, a_2 \dots a_n]^T \in \mathbb{R}^n$把等式约束和目标函数组合成为一个式子，对该式子进行最优化求解：

$$X = \underset{X}{argmin}[f(X) + A^T H(X)]$$

其中，$H(X) = [\, h_1(X),\, h_2(X) \dots h_n(X)]^T \in \mathbb{R}^n$，相当于将有等式约束的最优化问题转换成了无约束最优化求解问题，解决方法依旧是对$f(X) + A^T H(X)$的各个参数（X, A）求偏导，并令其等于 0，联立等式求解。

针对有不等式约束的最优化问题，常用的方法是 **KKT 条件（Karush-Kuhn-Tucker Conditions）**[3]：同样地，把所有的不等式约束、等式约束和目标函数全部写为一个式子：

$$L(X, A, B) = f(X) + A^T H(X) + B^T G(X)$$

KKT 条件是说最优值必须满足以下条件：

$$\frac{\partial}{\partial X} L(X, A, B) = 0$$

$$H(X) = 0$$

$$B^T G(X) = 0$$

其中，$B = [b_1, b_2 \dots b_m]^T \in \mathbb{R}^m$，$G(X) = [\, g_1(X),\, g_2(X) \dots g_m(X)]^T \in \mathbb{R}^m$。KKT 条件是求解最优值$X^*$的必要条件，要使其成为充分必要条件，还需要$f(X)$为凸函数才行。

在 KKT 条件中，$B^T G(X) = 0$这个条件最有趣，因为 $g_l(X) \le 0$，如果要满足这个等式，需要 $b_l = 0$或者 $g_l(X) = 0$。在我们后面的推导中会用到这个性质。

2.3 从 Batch 到 Online

我们面对的最优化问题都是无约束的最优化问题（有约束最优化问题可以利用拉格朗日乘数法或 KKT 条件转换成无约束最优化问题），因此我们通常可以将它们描述成：

$$W = \underset{W}{argmin}\, \ell(W, Z)$$

$$Z = \{(X_j, y_j) | j = 1,2, \dots M\} \tag{2-3-1}$$

$$y_j = h(W, X_j)$$

这里Z为观测样本集合（训练集）；X_j为第 j 条样本的特征向量；$y_j = h(W, X_j)$为预测值；$h(W, X_j)$为特征向量到预测值的映射函数；$\ell(W, Z)$为最优化求解的目标函数，也称作损失函数，损失函数通常可以分解为各样本损失函数的累加，即$\ell(W, Z) = \sum_{j=1}^{M} \ell(W, Z_j)$；$W$为特征权重，也就是我们需要求解的参数。以线性回归和逻辑回归为例，它们的映射函数和损失函数分别为：

Linear Regression $h(W, X_j) = W^T X_j$ $\ell(W, Z) = \sum_{j=1}^{M} (y_j - W^T X_j)^2$

Logistic Regression $h(W, X_j) = \frac{1}{1 + e^{-W^T X_j}}$ $\ell(W, Z) = \sum_{j=1}^{M} log(1 + e^{-y_j W^T X_j})$

在 2.1 中我们给出了无约束最优化问题解析解的求法。而在我们实际的数值计算中，通常做法是随机给定一个初始的$W^{(0)}$，通过迭代，在每次迭代中计算损失函数在当前$W^{(t)}$的下降方向，并更新W，直到损失函数稳定在最小的点[4]。例如著名的梯度下降法（GD, Gradient Descent）就是通过计算损失函数的在当前$W^{(t)}$处的梯度[5]（Gradient），以梯度$\nabla_W \ell(W^{(t)}, Z)$的反方向作为下降方向更新$W$，如果损失函数是一个非平滑的凸函数（Non-smooth convex），在不可导处用次梯度[6]（Subgradient）方向的反方向作为下降方向。算法如下[7]：

Algorithm 1. Batch Gradient Descent

Repeat until convergence {

$$W^{(t+1)} = W^{(t)} - \eta^{(t)} \nabla_W \ell(W^{(t)}, Z)$$

}

GD 是一种批量处理的方式（Batch），每次更新W的时候都要扫描所有的样本以计算一个全局的梯度$\nabla_W \ell(W, Z)$。

考虑另一种权重更新策略[7]：

Algorithm 2. Stochastic Gradient Descent

Loop {

for j=1 to M {

$$W^{(t+1)} = W^{(t)} - \eta^{(t)} \nabla_W \ell(W^{(t)}, Z_j)$$

}

}

在算法 2 中，每次迭代仅仅根据单个样本更新权重W，这种算法称作随机梯度下降[8]（SGD, Stochastic Gradient Descent）。

与 GD 相比较，GD 每次扫描所有的样本以计算一个全局的梯度，SGD 则每次只针对一个观测到的样本进行更新。通常情况下，SGD 能够比 GD“更快”地令W逼近最优值。当样本数M特别大的时候，SGD 的优势更加明显，并且由于 SGD 针对观测到的“一条”样本更新W，很适合进行增量计算，实现梯度下降的 Online 模式（OGD, Online Gradient Descent）。

2.4 正则化

正则化（Regularization）的意义本质上是为了避免训练得到的模型过度拟合（overfitting）训练数据。我们用图 2 来说明什么是过拟合，该图来自于王科的微博（@王小科科科）。图 2 是一个局部加权线性回归（Locally weighted linear regression）的训练结果，当学习度为 1 时，相当于进行线性回归，这时候模型与训练样本以及实际曲线拟合得都不够好，模型处于欠拟合（underfitting）状态；当学习度逐渐增加到 4 的过程中，模型逐渐与实际曲线吻合；随着学习度继续增加，越来越多的样本直接落到模型曲线上（模型拟合训练数据），但是模型却与实际曲线相差越来越大，出现了过拟合。

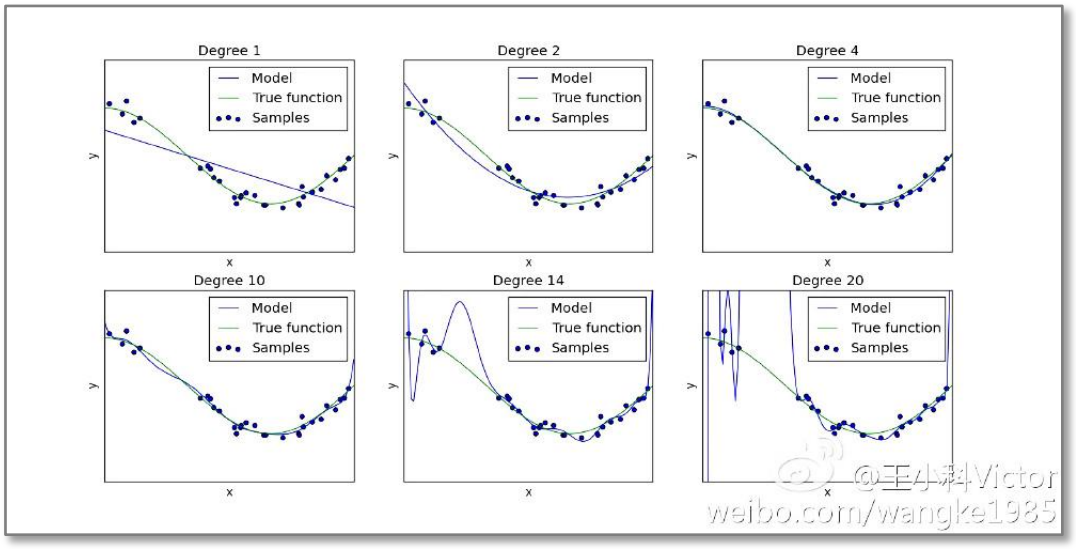

图 2 欠拟合 & 过拟合

过拟合体现出来的现象就是特征权重W的各个维度的绝对值非常大：一些大正数，一些大负数。这种模型虽然能够很好匹配样本（如图 2 中 Degree = 20 的情况），但是对新样本做预测的时候会使得预测值与真实值相差很远。

为了避免过拟合的情况，我们通常在损失函数的基础上加一个关于特征权重W的限制，限制它的模不要太大，如果用$\psi(W)$表示特征权重W的一种求模计算，那么(2-3-1)转换成：

$$W = argmin_W \ell(W, Z)$$

$$\textbf{\textit{s.t.}} \quad \psi(W) < \delta$$

这个时候我们面对的是一个有约束的最优化问题。根据 KKT 条件，我们知道当选取合适的系数λ，那么这个有约束最优化问题等价于如下无约束最优化问题：

$$W = argmin_W [\ell(W, Z) + \lambda\psi(W)]$$

其中$\psi(W)$称作正则化因子（Regularizer），是一个关于W求模的函数，我们常用正则化因子有 L1 和 L2 正则化：

L1 Regularization $$\psi(W) = \|W\|_1 = \sum_{i=1}^{N} |w_i|$$

L2 Regularization $$\psi(W) = \|W\|_2^2 = \sum_{i=1}^{N} w_i^2 = W^T W$$

不管是使用 L1 还是 L2 正则化，其基本原理都是一样的，即在最小化损失函数$\ell(W, Z)$的同时，还要考虑W的模带来的贡献，从而避免W的维度上取一些绝对值很大的值。

L1 和 L2 正则化的区别主要有两个：(1) L1 正则化在 0 处不可导，而 L2 正则化可导。好在无论是 L1 还是 L2 正则化本身都是凸函数，因此在计算 L1 正则化的梯度方向的可以采用次梯度代替；(2) 在 Batch 模式下，L1 正则化通常产生更加稀疏（Sparse）的模型，W的更多维度为 0，这些为 0 的维度就代表了不是很相关的维度，从而起到了特征选择（Feature Selection）的目的。

我们对稀疏性（Sparsity）比较感兴趣。除了特征选择的作用以外，稀疏性可以使得预测计算的复杂度降低。那么为什么 L1 正则化会产生这种稀疏性？通常可以根据文献[9]中的

理解，如图 3 所示：假如特征维度$N=2$，那么 L1 正则化引入的约束条件是W只能取转置方形中的值（图 3-(a)中黑色方框内），L2 正则化对应的是一个圆形区域（图 3-(b)中黑色圆形区域），图 3 中绿色部分代表损失函数的等高线，等高线与约束区域的首次相交的地方就是最优解。可以看到，由于 L1 正则化的约束区域与坐标轴相交的地方都有“角”出现，等高线更容易在这个“角”上与约束区域相交，导致另一个维度上的权重值为 0；而 L2 正则化则没有这种性质，因为没有“角”，等高线在坐标轴上与约束区域相交的概率大为减小。这样从直观上就解释了 L1 正则化更容易产生稀疏性的原因。

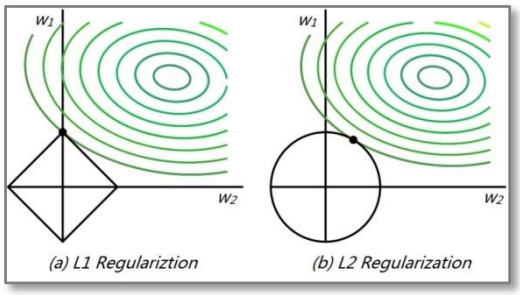

图 3 L1 正则化与 L2 正则化产生稀疏解示意图

那么在 Online 模式下呢，不同于 Batch，Online 中每次W的更新并不是沿着全局梯度进行下降，而是沿着某个样本的产生的梯度方向进行下降，整个寻优过程变得像是一个“随机”查找的过程（SGD 中 Stochastic 的来历），这样 Online 最优化求解即使采用 L1 正则化的方式，也很难产生稀疏解。后面介绍的各个在线最优化求解算法中，稀疏性是一个主要的追求目标。

3 在线最优化求解算法

在前面我们做了一些热身，下面将针对在线最优化求解介绍一些算法。在 2.4 中介绍了 L1 正则化在 Online 模式下也不能产生较好的稀疏性，而稀疏性对于高维特征向量以及大数据集又特别的重要。因此，我们沿着提升模型稀疏性的主线进行算法介绍。

3.1 TG

为了得到稀疏的特征权重W，最简单粗暴的方式就是设定一个阈值，当W的某维度上系数小于这个阈值时将其设置为 0（称作简单截断）。这种方法实现起来很简单，也容易理解。但实际中（尤其在 OGD 里面）W的某个系数比较小可能是因为该维度训练不足引起的，简单进行截断会造成这部分特征的丢失。

截断梯度法（TG, Truncated Gradient）是由 John Langford，Lihong Li 和 Tong Zhang 在 2009 年提出[10]，实际上是对简单截断的一种改进。下面首先描述一下 L1 正则化和简单截断的方法，然后我们再来看 TG 对简单截断的改进以及这三种方法在特定条件下的转化。

3.1.1 L1 正则化法

由于 L1 正则项在 0 处不可导，往往会造成平滑的凸优化问题变成非平滑凸优化问题，因此在每次迭代中采用次梯度计算 L1 正则项的梯度。权重更新方式为：

online时不会产生稀疏解 $$W^{(t+1)}=W^{(t)}-\eta^{(t)}G^{(t)}-\eta^{(t)}\lambda\mathrm{sgn}(W^{(t)})\tag{3-1-1}$$

注意，这里$\lambda\in\mathbb{R}$是一个标量，且$\lambda\geq 0$，为 L1 正则化参数；$\mathrm{sgn}(v)$为符号函数，如果$V=[v_1,v_2,\dots,v_N]\in\mathbb{R}^N$是一个向量，$v_i$是向量的一个维度，那么有$\mathrm{sgn}(V)=[\mathrm{sgn}(v_1),\mathrm{sgn}(v_2),\dots,\mathrm{sgn}(v_N)]\in\mathbb{R}^N$；$\eta^{(t)}$为学习率，通常将其设置成$1/\sqrt{t}$的函数；$G^{(t)}=\nabla_W\ell\left(W^{(t)},Z^{(t)}\right)$代表了第$t$次迭代中损失函数的梯度，由于 OGD 每次仅根据观测到的

一个样本进行权重更新，因此也不再使用区分样本的下标j。

3.1.2 简单截断法

以k为窗口，当t/k不为整数时采用标准的 SGD 进行迭代，当t/k为整数时，采用如下权重更新方式：k步做一次截断

$$W^{(t+1)} = T_0\left(W^{(t)} - \eta^{(t)}G^{(t)}, \theta\right)$$
$$T_0(v_i, \theta) = \begin{cases} 0 & if\ |v_i| \le \theta \\ v_i & otherwise \end{cases} \quad (3\text{-}1\text{-}2)$$

注意，这里面$\theta \in \mathbb{R}$是一个标量，且$\theta \ge 0$；如果$V = [v_1, v_2, \dots, v_N] \in \mathbb{R}^N$是一个向量，$v_i$是向量的一个维度，那么有$T_0(V, \theta) = [T_0(v_1, \theta), T_0(v_2, \theta), \dots, T_0(v_N, \theta)] \in \mathbb{R}^N$。

3.1.3 截断梯度法（TG）

上述的简单截断法被 TG 的作者形容为 ***too aggressive***，因此 TG 在此基础上进行了改进，同样是采用截断的方式，但是比较不那么粗暴。采用相同的方式表示为：

$$W^{(t+1)} = T_1\left(W^{(t)} - \eta^{(t)}G^{(t)}, \eta^{(t)}\lambda^{(t)}, \theta\right)$$
$$T_1(v_i, \alpha, \theta) = \begin{cases} max(0, v_i - \alpha) & if\ v_i \in [0, \theta] \\ min(0, v_i + \alpha) & if\ v_i \in [-\theta, 0] \\ v_i & otherwise \end{cases} \quad (3\text{-}1\text{-}3)$$

其中$\lambda^{(t)} \in \mathbb{R}$且$\lambda^{(t)} \ge 0$。TG 同样是以$k$为窗口，每$k$步进行一次截断。当$t/k$不为整数时$\lambda^{(t)} = 0$，当$t/k$为整数时$\lambda^{(t)} = k\lambda$。从公式(3-1-3)可以看出，$\lambda$和$\theta$决定了$W$的稀疏程度，这两个值越大，则稀疏性越强。尤其令$\lambda = \theta$时，只需要通过调节一个参数就能控制稀疏性。

根据公式(3-1-3)，我们很容易写出 TG 的算法逻辑：

Algorithm 3. Truncated Gradient

input θ
initial $W \in \mathbb{R}^N$
for t = 1,2,3... *do*
  $G = \nabla_W \ell\left(W, X^{(t)}, y^{(t)}\right)$
  refresh W *according to*

max -> min
$$w_i = \begin{cases} max(0, w_i - \eta^{(t)}g_i - \eta^{(t)}\lambda^{(t)}) & if\ \left(w_i - \eta^{(t)}g_i\right) \in [0, \theta] \\ max(0, w_i - \eta^{(t)}g_i + \eta^{(t)}\lambda^{(t)}) & if\ \left(w_i - \eta^{(t)}g_i\right) \in [-\theta, 0] \\ w_i - \eta^{(t)}g_i & otherwise \end{cases}$$

end
return W

3.1.4 TG 与简单截断以及 L1 正则化的关系

简单截断和截断梯度的区别在于采用了不同的截断公式T_0和T_1，如图 4 所示。

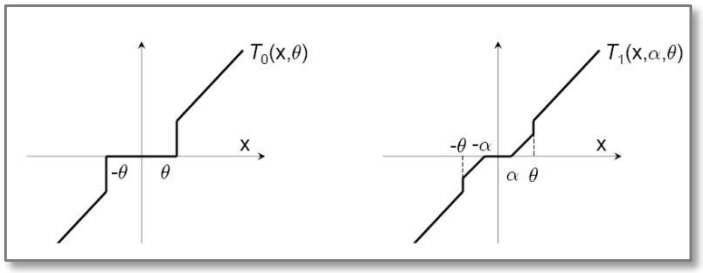

图 4 截断公式 T_0 & T_1 的曲线

为了清晰地进行比较，我们将公式(3-1-3)进行改写，描述特征权重每个维度的更新方式：

$$w_i^{(t+1)}=\begin{cases}Trnc\left(\left(w_i^{(t)}-\eta^{(t)}g_i^{(t)}\right),\lambda_{TG}^{(t)},\theta\right) & if\ mod(t,k)=0\\ w_i^{(t)}-\eta^{(t)}g_i^{(t)} & otherwise\end{cases}$$

$$\lambda_{TG}^{(t)}=\eta^{(t)}\lambda k \tag{3-1-4}$$

$$Trnc\left(w,\lambda_{TG}^{(t)},\theta\right)=\begin{cases}0 & if\ |w|\le\lambda_{TG}^{(t)}\\ w-\lambda_{TG}^{(t)}sgn(w) & if\ \lambda_{TG}^{(t)}\le|w|\le\theta\\ w & otherwise\end{cases}$$

如果令$\lambda_{TG}^{(t)}=\theta$，截断公式$Trnc\left(w,\lambda_{TG}^{(t)},\theta\right)$变成：

$$Trnc(w,\theta,\theta)=\begin{cases}0 & if\ |w|\le\theta\\ w & otherwise\end{cases}$$

此时 TG 退化成简单截断法。

如果令$\theta=\infty$截断公式$Trnc\left(w,\lambda_{TG}^{(t)},\theta\right)$变成：

$$Trnc\left(w,\lambda_{TG}^{(t)},\infty\right)=\begin{cases}0 & if\ |w|\le\lambda_{TG}^{(t)}\\ w & otherwise\end{cases}$$

如果再令$k=1$，那么特征权重维度更新公式变成

$$w_i^{(t+1)}=Trnc\left(\left(w_i^{(t)}-\eta^{(t)}g_i^{(t)}\right),\eta^{(t)}\lambda,\infty\right)=w_i^{(t)}-\eta^{(t)}g_i^{(t)}-\eta^{(t)}\lambda sgn(w_i^{(t)})$$

此时 TG 退化成 L1 正则化法。

3.2 FOBOS

3.2.1 FOBOS 算法原理

前向后向切分（FOBOS, Forward-Backward Splitting）是由 John Duchi 和 Yoram Singer 提出的[11]。从全称上来看，该方法应该叫 FOBAS，但是由于一开始作者管这种方法叫 FOLOS（Forward Looking Subgradients），为了减少读者的困扰，作者干脆只修改一个字母，叫 FOBOS。

在 FOBOS 中，将权重的更新分为两个步骤：

$$W^{(t+\frac{1}{2})}=W^{(t)}-\eta^{(t)}G^{(t)}$$

$$W^{(t+1)}=\underset{W}{argmin}\left\{\frac{1}{2}\left\|W-W^{(t+\frac{1}{2})}\right\|^2+\eta^{(t+\frac{1}{2})}\Psi(W)\right\} \tag{3-2-1}$$

前一个步骤实际上是一个标准的梯度下降步骤，后一个步骤可以理解为对梯度下降的结果进行微调。

观察第二个步骤，发现对W的微调也分为两部分：(1) 前一部分保证微调发生在梯度下降结果的附近；(2)后一部分则用于处理正则化，产生稀疏性。

如果将(3-2-1)中的两个步骤合二为一，即将$W^{(t+\frac{1}{2})}$的计算代入$W^{(t+1)}$中，有：

$$W^{(t+1)}=\underset{W}{argmin}\left\{\frac{1}{2}\left\|W-W^{(t)}+\eta^{(t)}G^{(t)}\right\|^2+\eta^{(t+\frac{1}{2})}\Psi(W)\right\}$$

令$F(W)=\frac{1}{2}\left\|W-W^{(t)}+\eta^{(t)}G^{(t)}\right\|^2+\eta^{(t+\frac{1}{2})}\Psi(W)$，如果$W^{(t+1)}$存在一个最优解，那么可以推断 0 向量一定属于$F(W)$的次梯度集合：

$$0 \in \partial F(W) = W - W^{(t)} + \eta^{(t)} G^{(t)} + \eta^{\left(t+\frac{1}{2}\right)} \partial \Psi(W)$$

由于$W^{(t+1)} = \underset{W}{argmin}\, F(W)$，那么有：

$$0 = \left\{ W - W^{(t)} - \eta^{(t)} G^{(t)} + \eta^{\left(t+\frac{1}{2}\right)} \partial \Psi(W) \right\} \Big|_{W=W^{(t+1)}}$$

上式实际上给出了 FOBOS 中权重更新的另一种形式：

$$W^{(t+1)} = W^{(t)} - \eta^{(t)} G^{(t)} - \eta^{\left(t+\frac{1}{2}\right)} \partial \Psi(W^{(t+1)})$$

我们这里可以看到，$W^{(t+1)}$不仅仅与迭代前的状态$W^{(t)}$有关，而且与迭代后的$\Psi(W^{(t+1)})$有关。可能这就是 FOBOS 名称的由来。

3.2.2 L1-FOBOS

关于 FOBOS 的收敛性和 Regret 就不在此讨论了，详情可参见论文[11]。这里我们来看看 FOBOS 如何在 L1 正则化下取得比较好的稀疏性。

在 L1 正则化下，有$\Psi(W) = \lambda \|W\|_1$。为了简化描述，用向量$V = [v_1, v_2 \dots v_N] \in \mathbb{R}^N$来表示$W^{\left(t+\frac{1}{2}\right)}$，用标量$\tilde{\lambda} \in \mathbb{R}$来表示$\eta^{\left(t+\frac{1}{2}\right)}\lambda$，并将公式(3-2-1)等号右边按维度展开：

$$W^{(t+1)} = \underset{W}{argmin} \sum_{i=1}^{N} \left(\frac{1}{2}(w_i - v_i)^2 + \tilde{\lambda}|w_i| \right) \tag{3-2-2}$$

我们可以看到，在求和公式$\sum_{i=1}^{N} \left(\frac{1}{2}(w_i - v_i)^2 + \tilde{\lambda}|w_i| \right)$中的每一项都是大于等于 0 的，所以公式(3-2-2)可以拆解成对特征权重W每一维度单独求解：

$$w_i^{(t+1)} = \underset{w_i}{\min} \left(\frac{1}{2}(w_i - v_i)^2 + \tilde{\lambda}|w_i| \right)$$

首先，假设w_i^*是$\text{minimize}_{w_i} \left(\frac{1}{2}(w_i - v_i)^2 + \tilde{\lambda}|w_i| \right)$的最优解，则有$w_i^* v_i \geq 0$，这是因为：

> 反证法：
> 假设：$w_i^* v_i < 0$，那么有：
> $$\frac{1}{2}v_i^2 < \frac{1}{2}v_i^2 - w_i^* v_i + \frac{1}{2}(w_i^*)^2 < \frac{1}{2}(w_i^* - v_i)^2 + \tilde{\lambda}|w_i^*|$$
> 这与w_i^*是$\text{minimize}_{w_i} \left(\frac{1}{2}(w_i - v_i)^2 + \tilde{\lambda}|w_i| \right)$的最优解矛盾，故假设不成立，$w_i^* v_i \geq 0$

既然有$w_i^* v_i \geq 0$，那么我们分两种情况$v_i \geq 0$和$v_i < 0$来讨论：

(1) 当$v_i \geq 0$ 时：

由于$w_i^* v_i \geq 0$，所以$w_i^* \geq 0$，相当于对$\text{minimize}_{w_i}\left(\frac{1}{2}(w_i - v_i)^2 + \tilde{\lambda}|w_i|\right)$ 引入了不等式约束条件 $-w_i \leq 0$；

为了求解这个含不等式约束的最优化问题，引入拉格朗日乘子 $\beta \geq 0$，由 KKT 条件，有：$\frac{\partial}{\partial w_i}\left(\frac{1}{2}(w_i - v_i)^2 + \tilde{\lambda}w_i - \beta w_i\right)\Big|_{w_i = w_i^*} = 0$ 以及 $\beta w_i^* = 0$；

根据上面的求导等式可得：$w_i^* = v_i - \tilde{\lambda} + \beta$；

分为两种情况：

① $w_i^* > 0$ ：

由于 $\beta w_i^* = 0$ 所以 $\beta = 0$

这时候有：$w_i^* = v_i - \tilde{\lambda}$

又由于$w_i^* > 0$，所以$v_i - \tilde{\lambda} > 0$

② $w_i^* = 0$ ：

这时候有：$v_i - \tilde{\lambda} + \beta = 0$

又由于$\beta \geq 0$，所以$v_i - \tilde{\lambda} \leq 0$

所以，在$v_i \geq 0$时，$w_i^* = \max(0, v_i - \tilde{\lambda})$

(2) 当$v_i < 0$ 时：

采用相同的分析方法，在$v_i < 0$时，有：$w_i^* = -\max(0, -v_i - \tilde{\lambda})$

综合上面的分析，可以得到在 FOBOS 在 L1 正则化条件下，特征权重的各个维度更新的方式为：

$$w_i^{(t+1)} = sgn(v_i)max(0, |v_i| - \tilde{\lambda}) = sgn\left(w_i^{(t)} - \eta^{(t)} g_i^{(t)}\right) max\left\{0, \left|w_i^{(t)} - \eta^{(t)} g_i^{(t)}\right| - \eta^{(t+\frac{1}{2})}\lambda\right\} \quad (3\text{-}2\text{-}3)$$

其中，$g_i^{(t)}$为梯度$G^{(t)}$在维度i上的取值。

根据公式(3-2-3)，我们很容易就可以设计出 L1-FOBOS 的算法逻辑：

Algorithm 4. Forward-Backward Splitting with L1 Regularization

input λ
initial $W \in \mathbb{R}^N$
for $t = 1,2,3\ldots$ *do*
  $G = \nabla_W \ell\left(W, X^{(t)}, y^{(t)}\right)$
  refresh W *according to*

$$w_i = sgn\left(w_i - \eta^{(t)} g_i\right) max\left\{0, \left|w_i - \eta^{(t)} g_i\right| - \eta^{(t+\frac{1}{2})}\lambda\right\}$$

end
return W

3.2.3 L1-FOBOS 与 TG 的关系

从公式(3-2-3)可以看出，L1-FOBOS 在每次更新W的时候，对W的每个维度都会进行判定，当满足$\left|w_i^{(t)} - \eta^{(t)} g_i^{(t)}\right| - \eta^{(t+\frac{1}{2})}\lambda \leq 0$ 时对该维度进行“截断”，令$w_i^{(t+1)} = 0$。那么我们怎

么去理解这个判定条件呢？如果我们把判定条件写成$\left|w_i^{(t)}-\eta^{(t)}g_i^{(t)}\right|\le\eta^{(t+\frac{1}{2})}\lambda$，那么这个含义就很清晰了：当一条样本产生的梯度不足以令对应维度上的权重值发生足够大的变化（$\eta^{(t+\frac{1}{2})}\lambda$），认为在本次更新过程中该维度不够重要，应当令其权重为 0。

某个维度上的梯度不足够大时截断为零

对于 L1-FOBOS 特征权重的各个维度更新公式(3-2-3)，也可以写作如下形式：

$$w_i^{(t+1)}=\begin{cases}0 & if\ \left|w_i^{(t)}-\eta^{(t)}g_i^{(t)}\right|\le\eta^{(t+\frac{1}{2})}\lambda\\ \left(w_i^{(t)}-\eta^{(t)}g_i^{(t)}\right)-\eta^{(t+\frac{1}{2})}\lambda sgn\left(w_i^{(t)}-\eta^{(t)}g_i^{(t)}\right) & otherwise\end{cases}$$

比较上式与 TG 的特征权重维度更新公式(3-1-4)，我们发现如果令$\theta=\infty,k=1,\lambda_{TG}^{(t)}=\eta^{(t+\frac{1}{2})}\lambda$，L1-FOBOS 与 TG 完全一致。我们可以认为 L1-FOBOS 是 TG 在特定条件下的特殊形式。

3.3 RDA

3.3.1 RDA 算法原理

不论怎样，简单截断、TG、FOBOS 都还是建立在 SGD 的基础之上的，属于梯度下降类型的方法，这类型方法的优点就是精度比较高，并且 TG、FOBOS 也都能在稀疏性上得到提升。但是有些其它类型的算法，例如 RDA，是从另一个方面来求解 Online Optimization 并且更有效地提升了特征权重的稀疏性。

正则对偶平均（RDA, Regularized Dual Averaging）是微软十年的研究成果，RDA 是 Simple Dual Averaging Scheme 一个扩展，由 Lin Xiao 发表于 2010 年[12]。

在 RDA 中，特征权重的更新策略为：

$$W^{(t+1)}=\underset{W}{argmin}\left\{\frac{1}{t}\sum_{r=1}^{t}\langle G^{(r)},W\rangle+\Psi(W)+\frac{\beta^{(t)}}{t}h(W)\right\}\qquad(3\text{-}3\text{-}1)$$

其中，$\langle G^{(r)},W\rangle$表示梯度$G^{(r)}$对W的积分平均值（积分中值）；$\Psi(W)$为正则项；$h(W)$为一个辅助的严格凸函数；$\{\beta^{(t)}|t\ge1\}$是一个非负且非自减序列。

本质上，公式(3-3-1)中包含了 3 个部分：(1) 线性函数$\frac{1}{t}\sum_{r=1}^{t}\langle G^{(t)},W\rangle$，包含了之前所有梯度（或次梯度）的平均值（dual average）；(2) 正则项$\Psi(W)$；(3) 额外正则项$\frac{\beta^{(t)}}{t}h(W)$，它是一个严格凸函数。

3.3.2 L1-RDA

我们下面来看看在 L1 正则化下，RDA 中的特征权重更新具有什么样的形式以及如何产生稀疏性。

令$\Psi(W)=\lambda\|W\|_1$，并且由于$h(W)$是一个关于W的严格凸函数，不妨令$h(W)=\frac{1}{2}\|W\|_2^2$，此外，将非负非自减序列$\{\beta^{(t)}|t\ge1\}$定义为$\beta^{(t)}=\gamma\sqrt{t}$，将 L1 正则化代入公式(3-3-1)有：

$$W^{(t+1)}=\underset{W}{argmin}\left\{\frac{1}{t}\sum_{r=1}^{t}\langle G^{(r)},W\rangle+\lambda\|W\|_1+\frac{\gamma}{2\sqrt{t}}\|W\|_2^2\right\}\qquad(3\text{-}3\text{-}2)$$

针对特征权重的各个维度将其拆解成 N 个独立的标量最小化问题：

$$\underset{w_i\in\mathbb{R}}{minimize}\left\{\bar{g}_i^{(t)}w_i+\lambda|w_i|+\frac{\gamma}{2\sqrt{t}}w_i^2\right\} \tag{3-3-3}$$

这里$\lambda>0$，$\frac{\gamma}{\sqrt{t}}>0$；$\bar{g}_i^{(t)}=\frac{1}{t}\sum_{r=1}^{t}g_i^{(r)}$；公式(3-3-3)就是一个无约束的非平滑最优化问题。

其中第 2 项$\lambda|w_i|$在$w_i=0$处不可导。假设w_i^*是其最优解，并且定义$\xi\in\partial|w_i^*|$为$|w_i|$在w_i^*的次导数，那么有：

$$\partial|w_i^*|=\begin{cases}\{-1<\xi<1\} & if\ w_i^*=0\\ \{1\} & if\ w_i^*>0\\ \{-1\} & if\ w_i^*<0\end{cases} \tag{3-3-4}$$

如果对公式(3-3-3)求导（求次导数）并等于 0，则有：

次梯度最优化条件 $$\bar{g}_i^{(t)}+\lambda\xi+\frac{\gamma}{\sqrt{t}}w_i=0 \tag{3-3-5}$$

由于$\lambda>0$，我们针对(3-3-5)分三种情况$\left|\bar{g}_i^{(t)}\right|<\lambda$、$\bar{g}_i^{(t)}>\lambda$和$\bar{g}_i^{(t)}<-\lambda$来进行讨论：

(1) **当$\left|\bar{g}_i^{(t)}\right|<\lambda$ 时：**

还可以分为三种情况：

① 如果$w_i^*=0$，由(3-3-5)得$\xi=-\bar{g}_i^{(t)}/\lambda\in\partial|0|$，满足(3-3-4)

② 如果$w_i^*>0$，由(3-3-4)得$\xi=1$，则$\bar{g}_i^{(t)}+\lambda+\frac{\gamma}{\sqrt{t}}w_i>\bar{g}_i^{(t)}+\lambda\geq 0$，不满足 (3-3-5)

③ 如果$w_i^*<0$，由(3-3-4)得$\xi=-1$，则$\bar{g}_i^{(t)}-\lambda+\frac{\gamma}{\sqrt{t}}w_i<\bar{g}_i^{(t)}-\lambda\leq 0$，不满足 (3-3-5)

所以，当$\left|\bar{g}_i^{(t)}\right|<\lambda$时，$w_i^*=0$

(2) 当$\bar{g}_i^{(t)}>\lambda$ 时：

采用相同分析方法得到$w_i^*<0$，有$\xi=-1$满足条件，即：$w_i^*=-\frac{\sqrt{t}}{\gamma}\left(\bar{g}_i^{(t)}-\lambda\right)$

(3) 当$\bar{g}_i^{(t)}<-\lambda$ 时：

采用相同分析方法得到$w_i^*>0$，有$\xi=1$满足条件，即：$w_i^*=-\frac{\sqrt{t}}{\gamma}\left(\bar{g}_i^{(t)}+\lambda\right)$

综合上面的分析，可以得到 L1-RDA 特征权重的各个维度更新的方式为：

$$w_i^{(t+1)}=\begin{cases}0 & \text{if } \left|\bar{g}_i^{(t)}\right|<\lambda\\ -\frac{\sqrt{t}}{\gamma}\left(\bar{g}_i^{(t)}-\lambda sgn(\bar{g}_i^{(t)})\right) & otherwise\end{cases} \tag{3-3-6}$$

这里我们发现，当某个维度上累积梯度平均值的绝对值$\left|\bar{g}_i^{(t)}\right|$小于阈值$\lambda$的时候，该维度权重将被置 0，特征权重的稀疏性由此产生。

根据公式(3-3-6)，可以设计出 L1-RDA 的算法逻辑：

Algorithm 5. Regularized Dual Averaging with L1 Regularization

input $\gamma,\ \lambda$

initialize $W \in \mathbb{R}^N, G = 0 \in \mathbb{R}^N$

for $t = 1,2,3\ldots$ *do*

$$G = \frac{t-1}{t}G + \frac{1}{t}\nabla_W \ell\left(W, X^{(t)}, y^{(t)}\right)$$

refresh W *according to*

$$w_i^{(t+1)} = \begin{cases} 0 & \text{if } |g_i| < \lambda \\ -\frac{\sqrt{t}}{\gamma}(g_i - \lambda sgn(g_i)) & otherwise \end{cases}$$

end

return W

3.3.3 L1-RDA 与 L1-FOBOS 的比较

在 3.2.3 中我们看到了 L1-FOBOS 实际上是 TG 的一种特殊形式，在 L1-FOBOS 中，进行“截断”的判定条件是$\left|w_i^{(t)} - \eta^{(t)} g_i^{(t)}\right| \le \lambda_{TG}^{(t)} = \eta^{(t+\frac{1}{2})}\lambda$。通常会定义$\eta$为与$\frac{1}{\sqrt{t}}$正相关的函数（$\eta = \Theta\left(\frac{1}{\sqrt{t}}\right)$），因此 L1-FOBOS 的“截断阈值”为$\Theta\left(\frac{1}{\sqrt{t}}\right)\lambda$，随着$t$的增加，这个阈值会逐渐降低。

相比较而言，从(3-3-6)可以看出，L1-RDA 的“截断阈值”为λ，是一个常数，并不随着t而变化，因此可以认为 L1-RDA 比 L1-FOBOS 在截断判定上更加 ***aggressive***，这种性质使得 L1-RDA 更容易产生稀疏性；此外，RDA 中判定对象是梯度的累加平均值$\bar{g}_i^{(t)}$，不同于 TG 或 L1-FOBOS 中针对单次梯度计算的结果进行判定，避免了由于某些维度由于训练不足导致截断的问题。并且通过调节λ一个参数，很容易在精度和稀疏性上进行权衡。

3.4 FTRL

在 3.3.3 中我们从原理上定性比较了 L1-FOBOS 和 L1-RDA 在稀疏性上的表现。有实验证明，L1-FOBOS 这一类基于梯度下降的方法有比较高的精度，但是 L1-RDA 却能在损失一定精度的情况下产生更好的稀疏性。那么这两者的优点能不能在一个算法上体现出来？这就是 FTRL 要解决的问题。

FTRL（Follow the Regularized Leader）是由 Google 的 H. Brendan McMahan 在 2010 年提出的[14]，后来在 2011 年发表了一篇关于 FTRL 和 AOGD、FOBOS、RDA 比较的论文[15]，2013 年又和 Gary Holt, D. Sculley, Michael Young 等人发表了一篇关于 FTRL 工程化实现的论文[16]。

本节我们首先从形式上将 L1-FOBOS 和 L1-RDA 进行统一，然后介绍从这种统一形式产生 FTRL 算法，以及介绍 FTRL 算法工程化实现的算法逻辑。

3.4.1 L1-FOBOS 和 L1-RDA 在形式上的统一

L1-FOBOS 在形式上，每次迭代都可以表示为（这里我们令$\eta^{\left(t+\frac{1}{2}\right)} = \eta^{(t)} = \Theta(\frac{1}{\sqrt{t}})$是一个随 t 变化的非增正序列）：

$$W^{(t+\frac{1}{2})} = W^{(t)} - \eta^{(t)} G^{(t)}$$

$$W^{(t+1)} = \underset{W}{argmin}\left\{\frac{1}{2}\left\|W - W^{(t+\frac{1}{2})}\right\|_2^2 + \eta^{(t)}\lambda\|W\|_1\right\}$$

把这两个公式合并到一起，有：

$$W^{(t+1)} = \underset{W}{argmin}\left\{\frac{1}{2}\|W - W^{(t)} + \eta^{(t)}G^{(t)}\|_2^2 + \eta^{(t)}\lambda\|W\|_1\right\}$$

通过这个公式很难直接求出$W^{(t+1)}$的解析解，但是我们可以按维度将其分解为 N 个独立的最优化步骤：

$$\underset{w_i\in\mathbb{R}}{minimize}\left\{\frac{1}{2}\left(w_i - w_i^{(t)} + \eta^{(t)}g_i^{(t)}\right)^2 + \eta^{(t)}\lambda|w_i|\right\}$$

$$= \underset{w_i\in\mathbb{R}}{minimize}\left\{\frac{1}{2}\left(w_i - w_i^{(t)}\right)^2 + \frac{1}{2}\left(\eta^{(t)}g_i^{(t)}\right)^2 + w_i\eta^{(t)}g_i^{(t)} + w_i^{(t)}\eta^{(t)}g_i^{(t)} + \eta^{(t)}\lambda|w_i|\right\}$$

$$= \underset{w_i\in\mathbb{R}}{minimize}\left\{w_i g_i^{(t)} + \lambda|w_i| + \frac{1}{2\eta^{(t)}}\left(w_i - w_i^{(t)}\right)^2 + \left[\frac{\eta^{(t)}}{2}\left(g_i^{(t)}\right)^2 + w_i^{(t)}g_i^{(t)}\right]\right\}$$

由于$\frac{\eta^{(t)}}{2}\left(g_i^{(t)}\right)^2 + w_i^{(t)}g_i^{(t)}$与变量$w_i$无关，因此上式可以等价于：

$$\underset{w_i\in\mathbb{R}}{minimize}\left\{w_i g_i^{(t)} + \lambda|w_i| + \frac{1}{2\eta^{(t)}}\left(w_i - w_i^{(t)}\right)^2\right\}$$

再将这 N 个独立最优化子步骤合并，那么 L1-FOBOS 可以写作：

$$W^{(t+1)} = \underset{W}{argmin}\left\{G^{(t)}\cdot W + \lambda\|W\|_1 + \frac{1}{2\eta^{(t)}}\|W - W^{(t)}\|_2^2\right\}$$

而对于 L1-RDA 的公式(3-3-2)，我们可以写作：

$$W^{(t+1)} = \underset{W}{argmin}\left\{G^{(1:t)}\cdot W + t\lambda\|W\|_1 + \frac{1}{2\eta^{(t)}}\|W - 0\|_2^2\right\}$$

这里$G^{(1:t)} = \sum_{s=1}^{t} G^{(s)}$；如果令$\sigma^{(s)} = \frac{1}{\eta^{(s)}} - \frac{1}{\eta^{(s-1)}}$，$\sigma^{(1:t)} = \frac{1}{\eta^{(t)}}$，上面两个式子可以写作：

$$W^{(t+1)} = \underset{W}{argmin}\left\{G^{(t)}\cdot W + \lambda\|W\|_1 + \frac{1}{2}\sigma^{(1:t)}\|W - W^{(t)}\|_2^2\right\} \quad (3\text{-}4\text{-}1)$$

$$W^{(t+1)} = \underset{W}{argmin}\left\{G^{(1:t)}\cdot W + t\lambda\|W\|_1 + \frac{1}{2}\sigma^{(1:t)}\|W - 0\|_2^2\right\} \quad (3\text{-}4\text{-}2)$$

需要注意，与论文[15]中的 Table 1 不同，我们并没有将 L1-FOBOS 也写成累加梯度的形式。

比较(3-4-1)和(3-4-2)这两个公式，可以看出 L1-FOBOS 和 L1-RDA 的区别在于：(1) 前者对计算的是累加梯度以及 L1 正则项只考虑当前模的贡献，而后者采用了累加的处理方式；(2) 前者的第三项限制W的变化不能离已迭代过的解太远，而后者则限制W不能离 0 点太远。

3.4.2 FTRL 算法原理

FTRL综合考虑了 FOBOS 和 RDA 对于正则项和W限制的区别，其特征权重的更新公式为：

$$W^{(t+1)} = \underset{W}{argmin}\left\{G^{(1:t)}\cdot W + \lambda_1\|W\|_1 + \lambda_2\frac{1}{2}\|W\|_2^2 + \frac{1}{2}\sum_{s=1}^{t}\sigma^{(s)}\|W - W^{(s)}\|_2^2\right\} \quad (3\text{-}4\text{-}3)$$

注意，公式(3-4-3)中出现了L2正则项$\frac{1}{2}\|W\|_2^2$，在论文[15]的公式中并没有这一项，但是在其2013年发表的FTRL工程化实现的论文[16]却使用到了L2正则项。事实上该项的引入并不影响FRTL的稀疏性，后面的推导过程会显示这一点。L2正则项的引入仅仅相当于对最优化过程多了一个约束，使得结果求解结果更加“平滑”。

公式(3-4-3)看上去很复杂，更新特征权重貌似非常困难的样子。不妨将其进行改写，将最后一项展开，等价于求下面这样一个最优化问题：

$$W^{(t+1)} = \underset{W}{argmin}\left\{\left(G^{(1:t)} - \sum_{s=1}^{t}\sigma^{(s)}W^{(s)}\right)\cdot W + \lambda_1\|W\|_1 + \frac{1}{2}\left(\lambda_2 + \sum_{s=1}^{t}\sigma^{(s)}\right)\|W\|_2^2 + \frac{1}{2}\sum_{s=1}^{t}\sigma^{(s)}\left\|W^{(s)}\right\|_2^2\right\}$$

由于$\frac{1}{2}\sum_{s=1}^{t}\sigma^{(s)}\left\|W^{(s)}\right\|_2^2$相对于$W$来说是一个常数，并且令$Z^{(t)} = G^{(1:t)} - \sum_{s=1}^{t}\sigma^{(s)}W^{(s)}$，上式等价于：

$$W^{(t+1)} = \underset{W}{argmin}\left\{Z^{(t)}\cdot W + \lambda_1\|W\|_1 + \frac{1}{2}\left(\lambda_2 + \sum_{s=1}^{t}\sigma^{(s)}\right)\|W\|_2^2\right\}$$

针对特征权重的各个维度将其拆解成N个独立的标量最小化问题：

$$\underset{w_i\in\mathbb{R}}{minimize}\left\{z_i^{(t)}w_i + \lambda_1|w_i| + \frac{1}{2}\left(\lambda_2 + \sum_{s=1}^{t}\sigma^{(s)}\right)w_i^2\right\}$$

到这里，我们遇到了与(3-3-3)类似的优化问题，用相同的分析方法可以得到：

$$w_i^{(t+1)} = \begin{cases} 0 & \text{if } \left|z_i^{(t)}\right| < \lambda_1 \\ -\left(\lambda_2 + \sum_{s=1}^{t}\sigma^{(s)}\right)^{-1}\left(z_i^{(t)} - \lambda_1 sgn(z_i^{(t)})\right) & otherwise \end{cases} \quad (3\text{-}4\text{-}4)$$

从(3-4-4)可以看出，引入 L2 正则化并没有对 FTRL 结果的稀疏性产生任何影响。

3.4.3 Per-Coordinate Learning Rates

前面介绍了 FTRL 的基本推导，但是这里还有一个问题是一直没有被讨论到的：关于学习率$\eta^{(t)}$的选择和计算。事实上在 FTRL 中，每个维度上的学习率都是单独考虑的（Per-Coordinate Learning Rates）。

在一个标准的 OGD 里面使用的是一个全局的学习率策略$\eta^{(t)} = \frac{1}{\sqrt{t}}$，这个策略保证了学习率是一个正的非增长序列，对于每一个特征维度都是一样的。

考虑特征维度的变化率：如果特征 1 比特征 2 的变化更快，那么在维度 1 上的学习率应该下降得更快。我们很容易就可以想到可以用某个维度上梯度分量来反映这种变化率。在 FTRL 中，维度i上的学习率是这样计算的：

$$\eta_i^{(t)} = \frac{\alpha}{\beta + \sqrt{\sum_{s=1}^{t}\left(g_i^{(s)}\right)^2}} \quad (3\text{-}4\text{-}5)$$

由于$\sigma^{(1:t)} = \frac{1}{\eta^{(t)}}$，所以公式(3-4-4)中$\sum_{s=1}^{t}\sigma^{(s)} = \frac{1}{\eta_i^{(t)}} = \left(\beta + \sqrt{\sum_{s=1}^{t}\left(g_i^{(s)}\right)^2}\right)/\alpha$。这里的$\alpha$和$\beta$是需要输入的参数，(3-4-4)中学习率写成累加的形式，是为了方便理解后面 FTRL 的迭代计算逻辑。

3.4.4 FTRL 算法逻辑

到现在为止，我们已经得到了 FTRL 的特征权重维度的更新方法（公式(3-4-4)），每个特

征维度的学习率计算方法（公式(3-4-5)），那么很容易写出 FTRL 的算法逻辑（这里是根据(3-4-4)和(3-4-5)写的 L1&L2-FTRL 求解最优化的算法逻辑，而论文[16]中 Algorithm 1 给出的是 L1&L2-FTRL 针对 Logistic Regression 的算法逻辑）：

Algorithm 6. FTRL-Proximal with L1 & L2 Regularization

input $\alpha,\ \beta,\ \lambda_1,\ \lambda_2$

initialize $W \in \mathbb{R}^N,\ Z = 0 \in \mathbb{R}^N,\ Q = 0 \in \mathbb{R}^N$

for t =1,2,3… do

  $G = \nabla_W \ell\left(W, X^{(t)}, y^{(t)}\right)$ *# gradient of loss function*

  for i in 1,2,…,N do *# for each coordinate*

    $\sigma_i = \frac{1}{\alpha}\left(\sqrt{q_i + g_i^2} - \sqrt{q_i}\right)\ \&\ q_i = q_i + g_i^2$ *# equals* $\frac{1}{\eta^{(t)}} - \frac{1}{\eta^{(t-1)}}$

    $z_i = z_i + g_i - \sigma_i w_i$

    $$w_i = \begin{cases} 0 & \text{if } \left|z_i^{(t)}\right| < \lambda_1 \\ -\left(\lambda_2 + \frac{\beta + \sqrt{q_i}}{\alpha}\right)^{-1}\left(z_i - \lambda_1 sgn(z_i)\right) & otherwise \end{cases}$$

  end

end

return W

FTRL 里面的 4 个参数需要针对具体的问题进行设置，指导性的意见参考论文[16]。

4 结束语

本文作为在线最优化算法的整理和总结，沿着稀疏性的主线，先后介绍了简单截断法、TG、FOBOS、RDA 以及 FTRL。从类型上来看，简单截断法、TG、FOBOS 属于同一类，都是梯度下降类的算法,并且 TG 在特定条件可以转换成简单截断法和 FOBOS；RDA 属于简单对偶平均的扩展应用；FTRL 可以视作 RDA 和 FOBOS 的结合，同时具备二者的优点。目前来看，RDA 和 FTRL 是最好的稀疏模型 Online Training 的算法。

谈到高维高数据量的最优化求解，不可避免的要涉及到并行计算的问题。作者之前有篇博客[4]讨论了 batch 模式下的并行逻辑回归，其实只要修改损失函数，就可以用于其它问题的最优化求解。另外，对于 Online 下，文献[17]给出了一种很直观的方法：

Algorithm 7. ParallelSGD$(\{\boldsymbol{Z_1}, \boldsymbol{Z_2} \ldots \boldsymbol{Z_m}\}, \boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{W_0}, \boldsymbol{k})$

for all $i \in \{1,2 \ldots k\}$ *parallel do*

  $v_i = SGD(\{Z_1, Z_2 \ldots Z_m\}, T, \eta, W_0)$ *on client*

end

Aggregate from all computers $v = \frac{1}{k}\sum_{i=1}^{k} v_i$ *and return* v

对于 Online 模式的并行化计算，一方面可以参考 ParallelSGD 的思路，另一方面也可以借鉴 batch 模式下对高维向量点乘以及梯度分量并行计算的思路。总之，在理解算法原理的基础上将计算步骤进行拆解，使得各节点能独自无关地完成计算最后汇总结果即可。

最后，需要指出的是相关论文里面使用的数学符号不尽相同，并且有的论文里面也存在一定的笔误，但是并不影响我们对其的理解。在本文中尽量采用了统一风格和含义的符号、变量等，因此在与参考文献中的公式对比的时候会稍有出入。另外，由于笔者的水平有限，行文中存在的错误难以避免，欢迎大家指正、拍砖。

参考文献

[1] *Convex function*. http://en.wikipedia.org/wiki/Convex_function

[2] *Lagrange multiplier*. http://en.wikipedia.org/wiki/Lagrange_multiplier

[3] Karush–Kuhn–Tucker conditions. http://en.wikipedia.org/wiki/Karush-Kuhn-Tucker_conditions

[4] *冯扬. 并行逻辑回归*. http://blog.sina.com.cn/s/blog_6cb8e53d0101oetv.html

[5] *Gradient*. http://sv.wikipedia.org/wiki/Gradient

[6] *Subgradient*. http://sv.wikipedia.org/wiki/Subgradient

[7] Andrew Ng. *CS229 Lecture notes*. http://cs229.stanford.edu/notes/cs229-notes1.pdf

[8] *Stochastic Gradient Descent*. http://en.wikipedia.org/wiki/Stochastic_gradient_descent

[9] T. Hastie, R. Tibshirani & J. Friedman. *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction.* Springer Series in Statistics. Springer, 2009

[10] John Langford, Lihong Li & Tong Zhang. *Sparse Online Learning via Truncated Gradient*. Journal of Machine Learning Research, 2009

[11] John Duchi & Yoram Singer. *Efficient Online and Batch Learning using Forward Backward Splitting*. Journal of Machine Learning Research, 2009

[12] Lin Xiao. *Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization*. Journal of Machine Learning Research, 2010

[13] *Convex Set*. http://en.wikipedia.org/wiki/Convex_set

[14] H. Brendan McMahan & M Streter. *Adaptive Bound Optimization for Online Convex Optimization*. In COLT, 2010

[15] H. Brendan McMahan. *Follow-the-Regularized-Leader and Mirror Descent: Equivalence Theorems and L1 Regularization*. In AISTATS, 2011

[16] H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, Jeremy Kubica, *Ad Click Prediction: a View from the Trenches*. In ACM SIGKDD, 2013

[17] Martin A. Zinkevich, Markus Weimer, Alex Smola & Lihong Li. *Parallelized Stochastic Gradient Descent*. In NIPS 2010