# Follow the Regularized Leader – Proximal

## 简介

在[1]中，McMahan等介绍了Google内部用于点击率预估的在线机器学习系统，此系统的核心算法就是Follow The Regularized Leader – Proximal，以下简称FTRL-Proximal。

## FTRL-Proximal算法

|  |
| --- |
| w1,I = 0  For t = 1 to T do  Receive feature vector xt  Predict pt = σ( xt ∙ wt)  Observe label yt ∈ {0, 1}  gi = (pt - yt)xi  Update w using |

以下将探索为什么FTRL-Proximal使用以上的更新方式，以及证明以上更新方式是有效的。

## 在线学习

文献[2]中定义在线学习为以下过程：

学习分为一连串的回合，在回合t开始的时候，学习程序获取问题xt，并给出答案pt, 之后得到正确答案yt，得到损失为L(pt,yt)，并根据此损失进行自我修正。目标是在学习过程中尽量减少损失。

|  |
| --- |
| For t = 1, 2, …  Receive question xt ∈χ  Predict h(xt, wt) = pt ∈ D  Receive true answer yt ∈ Y  Suffer loss L(pt,yt)  Update hypothesis wt |

t 时间参数，表示第t回合

xt 第t条问题，或者特征向量

χ 问题集、特征集等

h 模型【如逻辑回归、线性回归、SVM等】

wt 模型参数

pt 模型预估结果

D 模型预估结果集合【{0,1}为二元分类、[0,1]为回归】

yt 实际答案

Y 答案集合

L 损失函数 【如，】

不同在线学习算法主要是采用了不同的更新方案

## Regret Analysis

让为Loss函数，[3]中定义Regret：

即：算法的损失与通过后见之明而得到的最少损失的对比。

显然，如果随着时间T的迁移，Regret接近于一个常量，则算法的wT与最优的w\*是一致的，则此在线学习方法是有效的。

## Follow the Leader

FTL的思想是采用以下更新方案：

其Regret上限为：

以下我们将尝试使用FTL来解决两个最优问题。

Online Quadratic Optimization

Loss函数为，

由FTL定义得出：

即：所有当前见过的样本的Centroid。

令，则Online Quadratic Optimization Regret上限是

而

因此，最终wt会收敛到w\*，FTL能有效地解决Online Quadratic Optimization问题。

Online Linear Optimization

Loss函数为

zt的取值为

使用FTL则有以下

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t | zt | w | ft(w) |  |  |
| 1 | -0.5 | 0 | 0 | 0 | argmin{-0.5w} = 1 |
| 2 | 1 | 1 | 1 | 1 | argmin{-0.5w+1w}=-1 |
| 3 | -1 | -1 | 1 | 2 | argmin{-0.5w+1w-1w}=1 |
| 4 | 1 | 1 | 1 | 3 | argmin{-0.5w+1w-1w+1w}=-1 |
| … | … | … | … | … | … |
| T | … | … | … | T-1 |  |

从以上分析得出，FTL的Loss是O(T)，而最优的参数应该是w=1,Loss为0，因此FTL的Regret为O(T)，w不收敛，FTL无效。

## Follow the Regularized Leader

在Online Linear Optimization中，我们发现FTL算法针对此问题是无效的，其原因组合要是因为每次更新的wt抖动太大，导致最终无法收敛。而FTRL刚好使用Regularization解决这个问题。

FTRL的更新方案是：

显然不同的Regularization函数将给出不同的Regret上限。

可以证明FTRL的Regret上限为

针对上述的Online Linear Optimization，使用的FTRL，得出

其Regret上限为

如果f是L-Lipschitz，且，则有[TODO:square root of t instead T?]

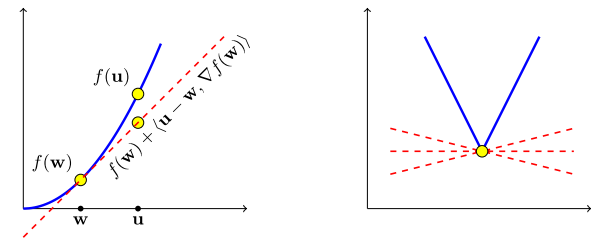
而，wt收敛，FTRL为有效算法。

## 凸函数的线性近似法（Linearization of Convex Function）

由凸函数的性质

可以得到凸函数的线性近似法

因此，我们可以使用线性近似的方法来求解。



## Follow the Regularized Leader – Proximal

对于的FTRL：

即Learning Rate为**常量** 的Online Gradient Descent。

如果使用以下更新

则得到Adaptive Learning Rate:

由于FTRL采用了线性近似法来获取Loss函数：

σ-strongly convex函数

FTRL-Proximal求解

[\sigma t increases as t increases, thus restricting w to be close to w\_t, otherwise, the cost of moving away will increase very fast.]越大

## Parallelized Algorithm

## Equivalence of RDA, COMID, FTRL-Proximal

## 基本概念

L-Lipschitz函数

Dual Norm

## 公式推导

1. Follow The Leader

上限的证明

Online Quadratic Optimization

## 文献

[1] McMahan et al, Ad Click Prediction: A View from the Trenches, KDD 13, 2013

[2] Shai Shalev-Shartz, Online Learning and Online Convex Optimization, Foundations and Trends in Machine Learning Vol. 4, No. 2, 2011

[3]McMahan, Dekel, Online Learning Lecture Notes, CSE599s, University of Washington, Spring 2012