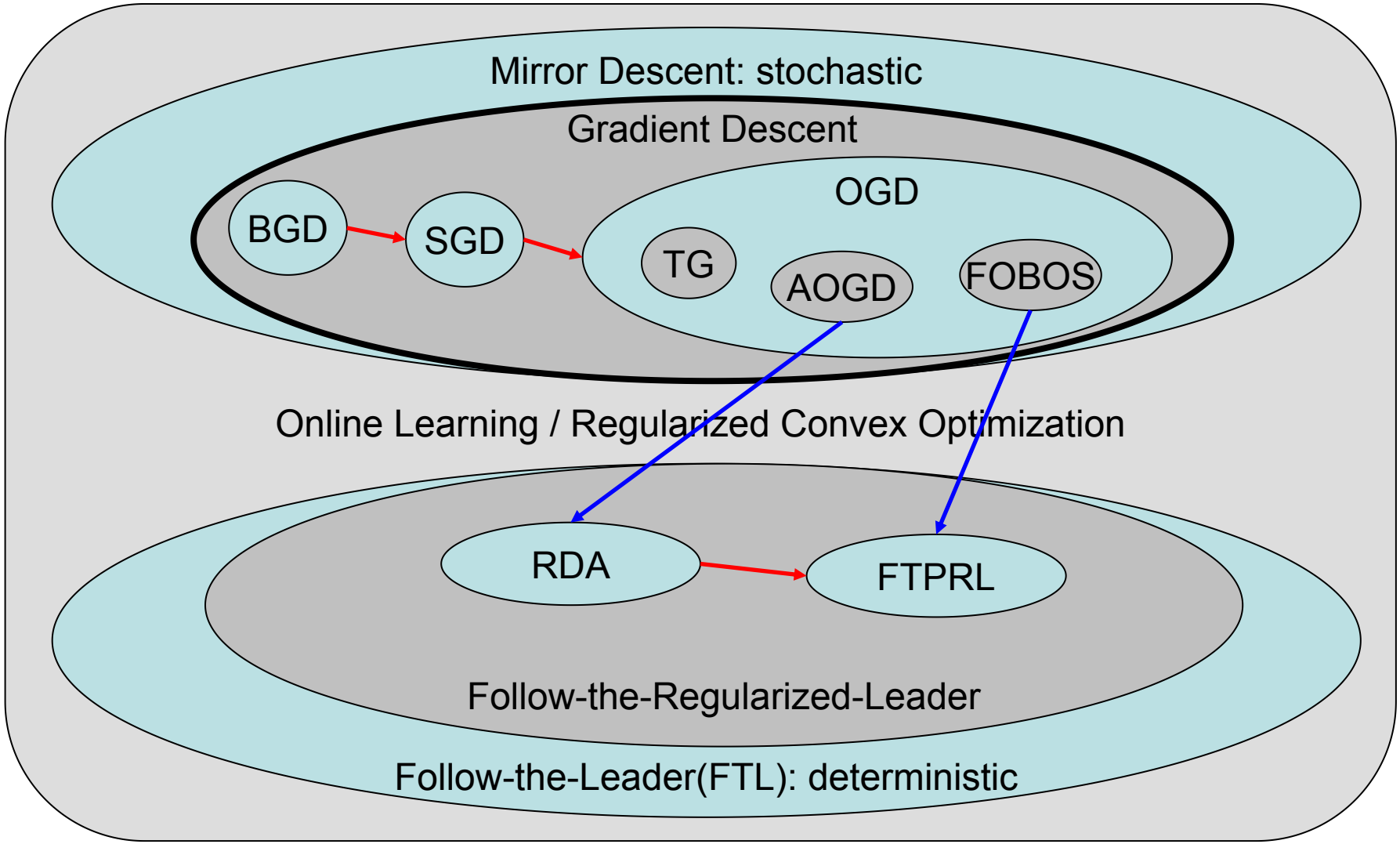
Online Learning/Optimization
算法实现与理论分析

poetniu

2015.9.21

goal


Mirror Descent: stochastic
Gradient Descent
OGD
BGD
SGD
TG
AOGD
FOBOS
Online Learning / Regularized Convex Optimization
RDA
FTPRL
Follow-the-Regularized-Leader
Follow-the-Leader(FTL): deterministic

outline

- 背景
- Online Optimization算法实现
  - Online Optimization
  - Truncated Gradient
  - FOBOS
  - RDA
  - FTPRL
- Online Optimization理论分析
  - Preliminary
  - Algorithms
  - FTRL-style & Implicit and Composite Updates
  - Experiments
- 总结
- References

背景

- Batch Learning (offline approach)
  - Get all the data and train the algorithm in one go
- Disadvantages when data is big
  - Requires all data to be loaded into memory
  - Periodic retraining is necessary
  - Very time consuming with big data!
- Example: Real Time Bidding
  - RTB algorithms are usually based on logistic regression
  - Whether or not to bid on a user is determined by the probability that the user will click on an add
  - Each day billions of bids are processed
  - Each bid has to be processed within 80 milliseconds

背景

- Example: Fraud Detection

Detecting Fraudulent Credit Card Transactions

- The probability that a transaction is using a stolen credit card is typically estimated with logistic regression
- Billions of transactions are analyzed each day

- Online Learning
  - Pass each data point sequentially through the algorithm
  - Only requires one data point at a time in memory
  - Allows for on-the-fly training of the algorithm

Online Learning

- When we have a continuous stream of data
- When It is important to update the algorithm in real time – can hit a moving target
- When training speed is important
- Parameters are "jumpy" around the optimal values

Batch

- When it is very important to get the exact optimal values
- When data can fit in memory
- When training time is not of the essence

outline

- 背景
- Online Optimization算法实现
  - Online Optimization
  - Truncated Gradient
  - FOBOS
  - RDA
  - FTPRL
- Online Optimization理论分析
  - Preliminary
  - Algorithms
  - FTRL-style & Implicit and Composite Updates
  - Experiments
- 总结
- References

Online Optimization

- Batch Gradient Descent

Algorithm 1. Batch Gradient Descent

Repeat until convergence {

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

}

- Stochastic Gradient Descent (SGD)

Algorithm 2. Stochastic Gradient Descent

Loop {

for j=1 to M {

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

}

}

- Online Gradient Descent (OGD)

Online Optimization

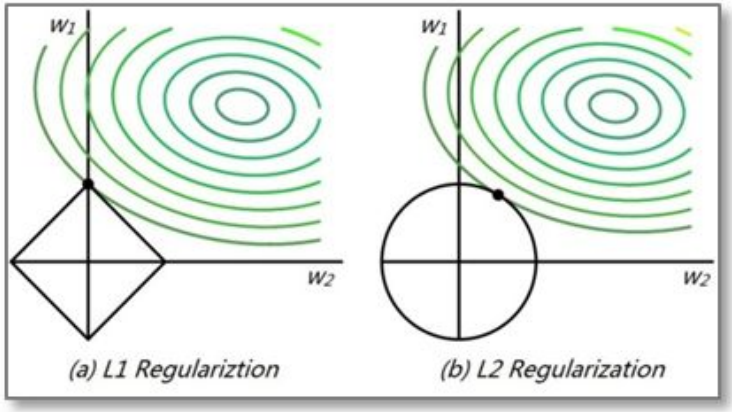
- Regularization
  - trade-off: overfitting & underfitting

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

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

- L1 & L2
  - 凸函数
  - L1在0处不可导，采用subgradients

- Sparsity
  - Batch下L1更容易产生稀疏解, global
  - Online下很难产生稀疏解, local
  - 稀疏性



Truncated Gradient

- L1正则化法
  - sgn(v)为符号函数

$$W^{(t+1)} = W^{(t)} - \eta^{(t)} G^{(t)} - \eta^{(t)} \lambda sgn(W^{(t)})$$

- 简单截断法
  - k为窗口，t/k为整数时截断
  - θ 是一个正数

$$W^{(t+1)} = T_0(W(t) - \eta^{(t)} G^{(t)}, \theta)$$

$$T_0(v_i, \theta) = \begin{cases} 0 & if\ |v_i| \leqslant \theta \\ v_i & otherwise \end{cases}$$

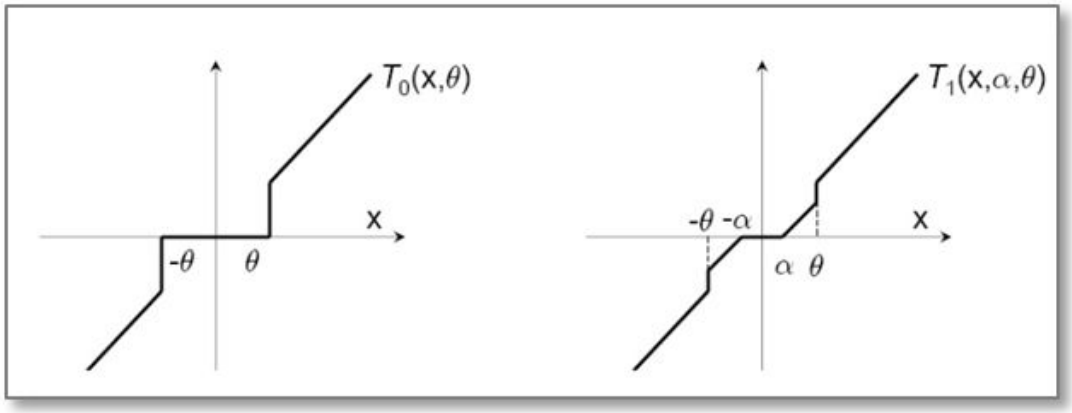
- 截断梯度法（TG）
  - t/k不为整 λ =0
  - t/k为整 λ (t)=k λ
  - λ , θ 越大，稀疏性越强

$$W^{(t+1)} = T_1(W^{(t)} - \eta^{(t)} G^{(t)}, \eta^{(t)} \lambda^{(t)}, \theta)$$

$$T_1(V, \alpha, \theta) = \begin{cases} max(0, v_i - \alpha) & if\ v_i \in [0, \theta] \\ min(0, v_i - \alpha) & if\ v_i \in [-\theta, 0] \\ v_i & otherwise \end{cases}$$

Truncated Gradient

- TG与简单截断关系



- TG

Algorithm 3. Truncated Gradient

1 *input* θ

2 *initial* $W \in \mathbb{R}^N$

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

4 $G = \nabla_W \ell\left(W, X^{(t)}, y^{(t)}\right)$

5 *refresh* W *according to*

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

6 *end*

7 *return W*

Forward-Backward Splitting

- FOBOS, 2009
  - 梯度下降
  - 梯度下降结果微调
  - 微调在梯度下降附近，处理正则稀疏性
- L1-FOBOS

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

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

$$w_i^{(t+1)} = sgn(v_i) max(0, |v_i| - \tilde{\lambda})$$

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

Algorithm 4. Forward-Backward Splitting with L1 Regularization

input λ

initial $W \in \mathbb{R}^N$

for t = 1,2,3... *do*

  $G = \nabla_W \ell(W, X^{(t)}, y^{(t)})$

  refresh W *according to*

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

end

return W

- L1-FOBOS是TG在特定条件下的特殊形式

Regularized Dual Averaging

- RDA, 2009
- L1-RDA

$$W^{(t+1)} = arg\min_{W}\left\{\frac{1}{2}\sum_{r=1}^{t}\left\langle G^{(t)}, W\right\rangle + \Psi(W) + \frac{\beta^{(t)}}{t}h(W)\right\}$$

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

Algorithm 5. Regularized Dual Averaging with L1 Regularization

1 *input* $\gamma,\ \lambda$

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

3 *for* t = 1,2,3... *do*

4 $G = \frac{t-1}{t}G + \frac{1}{t}\nabla_W \ell\left(W, X^{(t)}, y^{(t)}\right)$

5 *refresh* W *according to*

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

6 *end*

7 *return W*

- L1-RDA依据常数λ截断
- L1-FOBOS截断λ η随t增加减小
- L1-RDA更容易产生稀疏解

Follow the Regularized Leader

- FTRL, 2010
  - L1-FOBOS基于梯度下降准确性较好
  - L1-RDA稀疏性更好

- L1-FOBOS与L1-RDA形式统一
  - L1-FOBOS

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

  - L1-RDA

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

- FTRL-Proximal

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

Follow the Regularized Leader

- FTRL-Proximal

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

- Per-Coordinate Learning Rates

$$\eta_i^{(t)} = \frac{\alpha}{\beta + \sqrt{\sum_{s=1}^{t} \left(g_i^{(s)}\right)^2}}$$

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*

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

    $z_i = z_i + g_i - \sigma_i w_i$

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

  end

end

return W

Online Optimization算法实现

- Gradient Descent：准确性
  - 简单截断法
  - TG
  - FOBOS
- Dual Averaging：稀疏性
  - RDA
- FTRL-Proximal：稀疏性+准确性
  - FOBOS+RDA
- 目前最好
  - RDA, FTRL-Proximal

outline

- 背景
- Online Optimization算法实现
  - Online Optimization
  - Truncated Gradient
  - FOBOS
  - RDA
  - FTPRL
- Online Optimization理论分析
  - Preliminary
  - Algorithms
  - FTRL-style & Implicit and Composite Updates
  - Experiments
- 总结
- References

Preliminary(1)

- Follow-the-Leader

Follow-The-Leader (FTL)

$$\forall t, \quad \mathbf{w}_t = \operatorname*{argmin}_{\mathbf{w} \in S} \sum_{i=1}^{t-1} f_i(\mathbf{w})$$

  - Failure of FTL
    - w shifts drastically from round to round
    - predictions are not stable!

- Follow-the-Regularized-Leader
  - adding regularization

Follow-the-Regularized-Leader (FoReL)

$$\forall t, \quad \mathbf{w}_t = \operatorname*{argmin}_{\mathbf{w} \in S} \sum_{i=1}^{t-1} f_i(\mathbf{w}) + R(\mathbf{w})$$

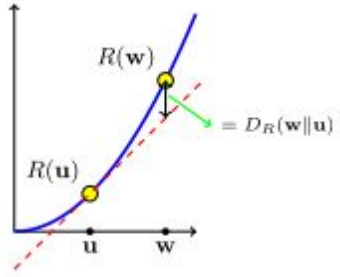
Preliminary(2)

- Mirror Descent
  - projected gradient descent

$$x_{k+1} = \pi_C(x_k - \alpha_k \nabla f(x_k)). \quad x_{k+1} = \arg\min_{x \in C} \left\{ \langle x, \nabla f(x_k) \rangle + \frac{1}{\alpha_k} \frac{\|x - x_k\|_2^2}{2} \right\}.$$

  - genderalized PGD

$$x_{k+1} = \arg\min_{x \in C} \left\{ \langle x, \nabla f(x_k) \rangle + \frac{1}{\alpha_k} d(x, x_k) \right\} \quad \text{with} \quad \alpha_k > 0.$$



  - Bregman divergences and MDA

$$B_\psi(x, y) := \psi(x) - \psi(y) - \langle x - y, \psi'(y) \rangle, \quad \text{with} \quad \psi'(y) \in \partial\psi(y),$$

$$x_{k+1} = \nabla\phi^\star(\nabla\varphi(x_k) - \alpha_k \nabla f(x_k)), \quad \text{where } \phi^\star(y) := \max_{z \in C} [\langle z, y \rangle - \varphi(z)].$$

  - The simplest is gradient descent $\varphi = \frac{1}{2}\mathrm{x}^2$ $x_{k+1} = x_k - a_k \nabla f(x_k)$

Preliminary(3)

- Gradient descent / forward Euler
  - gradient descent iteration (with step size c):

$$\mathbf{x}^{k+1} = \mathbf{x}^k - c\nabla f(\mathbf{x}^k)$$

  - $\mathbf{x}^{k+1}$ *minimizes* the following local quadratic approximation of f:

$$f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle + \frac{1}{2c}\|\mathbf{x} - \mathbf{x}^k\|_2^2$$

  - compare with forward Euler iteration, a.k.a. the explicit update:

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \Delta t \cdot \nabla f(\mathbf{x}(t))$$

- Backward Euler / implicit gradient descent
  - backward Euler iteration, also known as the implicit update:

$$\mathbf{x}(t+1) \overset{\text{solve}}{\longleftarrow} \mathbf{x}(t+1) = \mathbf{x}(t) - \Delta t \cdot \nabla f\left(\mathbf{x}(t+1)\right).$$

  - equivalent to:
    1. $\mathbf{u}(t+1) \overset{\text{solve}}{\longleftarrow} \mathbf{u} = \nabla f\left(\mathbf{x}(t) - \Delta t \cdot \mathbf{u}\right)$,
    2. $\mathbf{x}(t+1) = \mathbf{x}(t) - \Delta t \cdot \mathbf{u}(t+1)$.

Preliminary(4)

- sub-gradients

Lemma 2.5. Let S be a convex set. A function $f : S \to \mathbb{R}$ is convex iff for all $\mathbf{w} \in S$ there exists $\mathbf{z}$ such that

$$\forall \mathbf{u} \in S, \quad f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \mathbf{z} \rangle. \tag{2.3}$$

Definition 2.3 (sub-gradients). A vector $\mathbf{z}$ that satisfies Equation (2.3) is called a *sub-gradient* of f at $\mathbf{w}$. The set of sub-gradients of f at $\mathbf{w}$ is denoted $\partial f(\mathbf{w})$. Furthermore, if f is differentiable at $\mathbf{w}$ then $\partial f(\mathbf{w})$ contains a single element — the gradient of f at $\mathbf{w}$, $\nabla f(\mathbf{w})$.

An illustration of sub-gradients is given in Figure 2.1.

Getting back to online convex optimization, for each round t, there exists $\mathbf{z}_t$ such that for all $\mathbf{u}$,

$$f_t(\mathbf{w}_t) - f_t(\mathbf{u}) \leq \langle \mathbf{w}_t - \mathbf{u}, \mathbf{z}_t \rangle.$$

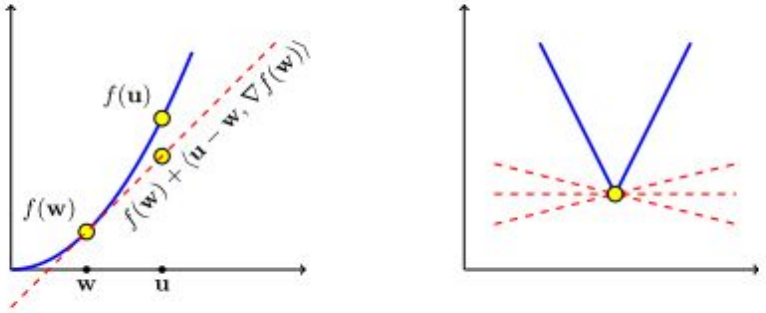


Fig. 2.1 Left: The right-hand side of Equation (2.3) is the tangent of f at $\mathbf{w}$. For a convex function, the tangent lower bounds f. Right: Illustration of several sub-gradients of a non-differentiable convex function.

Do What?

- Follow-the-Regularized-Leader: RDA
- Mirror Descent: FOBOS
- Follow-the-Proximally-Regularized-Leader(FTRL-Proximal)
- A unified analysis
  - Extend to implicit updates
  - Handle composite objectives
  - Formulation as instances of follow-the-regularized-leader
- Application
  - Sparse Model via L1 Regularization
    - RDA introduces more sparsity than FOBOS
  - Quadratic stabilizing regularization
  - Experimental comparison: FOBOS, RDA, FTRL-Proximal

Algorithms

- FOBOS

$$x_{t+1} = \arg\min_x g_t \cdot x + \lambda \|x\|_1 + \frac{1}{2} \|Q_{1:t}^{\frac{1}{2}}(x - x_t)\|_2^2.$$

- RDA

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + t\lambda \|x\|_1 + \frac{1}{2} \sum_{s=1}^{t} \|Q_s^{\frac{1}{2}}(x - 0)\|_2^2.$$

- FTRL-Proximal

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + t\lambda \|x\|_1 + \frac{1}{2} \sum_{s=1}^{t} \|Q_s^{\frac{1}{2}}(x - x_s)\|_2^2.$$

Algorithms

- Three components
  - (A) An approximation to the sum of previous loss functions
  - (B) Terms for the non-smooth composite terms
  - (C) Stabilizing regularization for low regret

		(A)	(B)	(C)
COMID	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+\ \phi_{1:t-1} \cdot x + \alpha_t \Psi(x)$	$+\frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}} (x - x_s) \rVert^2$
RDA	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+\ \alpha_{1:t} \Psi(x)$	$+\frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}} (x - 0) \rVert^2$
FTPRL	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+\ \alpha_{1:t} \Psi(x)$	$+\frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}} (x - x_s) \rVert^2$
AOGD	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+\ \phi_{1:t-1} \cdot x + \Psi(x)$	$+\frac{1}{2} \sum_{s=1}^{t} \lVert Q_s^{\frac{1}{2}} (x - 0) \rVert_2^2$

outline

- 背景
- Online Optimization算法实现
  - Online Optimization
  - Truncated Gradient
  - FOBOS
  - RDA
  - FTPRL
- Online Optimization理论分析
  - Preliminary
  - Algorithms
  - FTRL-style & Implicit and Composite Updates
  - Experiments
- 总结
- References

Implicit and Composite Updates for FTRL

- Standard subgradient FTRL

$$x_{t+1} = \arg\min_x \left(\sum_{s=1}^{t} \nabla f_s(x_s) \right) \cdot x + R_{1:t}(x).$$

- Implicit update for FTRL

$$x_{t+1} = \arg\min_x \left(\sum_{s=1}^{t-1} \nabla f_s(x_{s+1}) \right) \cdot x + f_t(x) + R_{1:t}(x).$$

- When f is not differentiable

$$x_{t+1} = \arg\min_x g'_{1:t-1} \cdot x + f_t(x) + R_{1:t}(x),$$

- Introducing a fixed convex function(L1 for sparsity) & giving the composite objective update

$$x_{t+1} = \arg\min_x \left(\sum_{s=1}^{t} \nabla f_s(x_s) \right) \cdot x + \alpha_{1:t}\Psi(x) + R_{1:t}(x),$$

- Finally, implicit update with a composite objective

$$x_{t+1} = \arg\min_x g'_{1:t-1} \cdot x + f_t(x) + \alpha_{1:t}\Psi(x) + R_{1:t}(x),$$

Motivation for Implicit Updates and Composite Objectives

- Empirically, implicit updates outperform linearized updates, more robustness
- Examples
  - Importance weights

$$f_t(x) = \frac{1}{2}(x-3)^2 \quad and \quad x_t = 2$$

  - L1-regularization

$$f_t(x) = g \cdot x + \| x \|$$

  - Composite updates

For Now

- FTRL-style: done!
- Next: mirror descent

		(A)		(B)	(C)
COMID	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+$	$\phi_{1:t-1} \cdot x + \alpha_t \Psi(x)$	$+\frac{1}{2}\sum_{s=1}^{t} \|Q_s^{\frac{1}{2}}(x - x_s)\|^2$
RDA	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+$	$\alpha_{1:t} \Psi(x)$	$+\frac{1}{2}\sum_{s=1}^{t} \|Q_s^{\frac{1}{2}}(x - 0)\|^2$
FTPRL	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+$	$\alpha_{1:t} \Psi(x)$	$+\frac{1}{2}\sum_{s=1}^{t} \|Q_s^{\frac{1}{2}}(x - x_s)\|^2$
AOGD	$\arg\min_x$	$g'_{1:t-1} \cdot x + f_t(x)$	$+$	$\phi_{1:t-1} \cdot x + \Psi(x)$	$+\frac{1}{2}\sum_{s=1}^{t} \|Q_s^{\frac{1}{2}}(x - 0)\|_2^2$

Mirror Descent Follows The Leader

- Let $f_t(x) = g_t \cdot x + \psi(x)$
- FTRL

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + R_{1:t}(x).$$

- Mirror Descent: on each round
  - The simplest is gradient descent
  - Re-written
  - Bregman divergences
  - Composite-objective mirror descent (COMID)

$$x_{t+1} = x_t - \eta g_t = -\eta g_{1:t}.$$

$$x_{t+1} = \arg\min_x g_t \cdot x + \frac{1}{2\eta_t}\|x - x_t\|_2^2.$$

$$x_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_{1:t}(x, x_t)$$

$$x_{t+1} = \arg\min_x g_t \cdot x + \Psi(x) + \mathcal{B}_{1:t}(x, x_t),$$

An Equivalence Theorem for Proximal Regularization

- For COMID (FOBOS)

Theorem 4. *Let R_t be a sequence of differentiable origin-centered convex functions ($\nabla R_t(0) = 0$), with R_1 strongly convex, and let Ψ be an arbitrary convex function. Let $x_1 = \hat{x}_1 = 0$. For a sequence of loss functions $f_t(x) + \Psi(x)$, let the sequence of points played by the implicit-update composite-objective mirror descent algorithm be*

$$\hat{x}_{t+1} = \arg\min_x \; f_t(x) + \alpha_t \Psi(x) + \tilde{\mathcal{B}}_{1:t}(x, \hat{x}_t), \tag{12}$$

where $\tilde{R}_t(x) = R_t(x - \hat{x}_t)$, and $\tilde{\mathcal{B}}_t = \mathcal{B}_{\tilde{R}_t}$, so $\tilde{\mathcal{B}}_{1:t}$ is the Bregman divergence with respect to $\tilde{R}_1 + \cdots + \tilde{R}_t$. Consider the alternative sequence of points x_t played by a proximal FTRL algorithm, applied to these same f_t, defined by

$$x_{t+1} = \arg\min_x \; (g'_{1:t-1} + \phi_{1:t-1}) \cdot x + f_t(x) + \alpha_t \Psi(x) + \tilde{R}_{1:t}(x) \tag{13}$$

for some $g'_t \in \partial f_t(x_{t+1})$ and $\phi_t \in \partial(\alpha_t \Psi)(x_{t+1})$. Then, these algorithms are equivalent, in that $x_t = \hat{x}_t$ for all $t > 0$.

An Equivalence Theorem for Proximal Regularization

- For FOBOS

Corollary 5. *Let $f_t(x) = g_t \cdot x$. Then, the following algorithms play identical points:*

- *Gradient descent with positive semi-definite learning rates Q_t, defined by:*

$$x_{t+1} = x_t - Q_{1:t}^{-1} g_t.$$

- *FTRL-Proximal with regularization functions $\tilde{R}_t(x) = \frac{1}{2}\left\|Q_t^{\frac{1}{2}}(x - x_t)\right\|_2^2$, which plays*

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + \tilde{R}_{1:t}(x).$$

Bregman divergence is

$$\mathcal{B}_R(x, y) = R(x) - (R(y) + \nabla R(y) \cdot (x - y))$$

for any $x, y \in \mathbb{R}^n$. We then have the update

$$x_{t+1} = \arg\min_x g_t \cdot x + \frac{1}{\eta_t}\mathcal{B}_R(x, x_t), \tag{8}$$

or explicitly (by setting the gradient of (8) to zero),

$$x_{t+1} = r^{-1}(r(x_t) - \eta_t g_t) \tag{9}$$

where $r = \nabla R$. Letting $R(x) = \frac{1}{2}\|x\|_2^2$ so that $\mathcal{B}_R(x, x_t) = \frac{1}{2}\|x - x_t\|_2^2$ recovers the algorithm of Equation (7). One way to see this is to note that $r(x) = r^{-1}(x) = x$ in this case.

- For COMID

Corollary 6. *Consider Implicit-Update Composite-Objective Mirror Descent, which plays*

$$\hat{x}_{t+1} = \arg\min f_t(x) + \alpha_t \Psi(x) + \frac{1}{2}\left\|Q_{1:t}^{\frac{1}{2}}(x - \hat{x}_t)\right\|^2. \tag{14}$$

Then an equivalent FTPRL update is

$$x_{t+1} = \arg\min_x \; (g'_{1:t-1} + \phi_{1:t-1}) \cdot x + f_t(x) + \alpha_t \Psi(x) + \frac{1}{2}\sum_{s=1}^{t}\left\|Q_s^{\frac{1}{2}}(x - x_s)\right\|^2 \tag{15}$$

for some $g'_t \in \partial f_t(x_{t+1})$ and $\phi_t \in \partial(\alpha_t \Psi)(x_{t+1})$.

For Now

- COMID (FOBOS) = FTRL-Proximal: done!

For the moment, suppose $\Psi(x) = 0$. So far, we have shown conditions under which gradient descent on $f_t(x) = g_t \cdot x$ with an adaptive step size is equivalent to follow-the-proximally-regularized-leader.

- Next: AOGD = RDA

. In this section, we show that mirror descent on the *regularized* functions $f_t^R(x) = g_t \cdot x + R_t(x)$, with a certain natural step-size, is equivalent to a follow-the-regularized-leader algorithm with origin-centered regularization.

An Equivalence Theorem for Origin-Centered Regularization

- For AOGD

Theorem 7. *Let $f_t(x) = g_t \cdot x$, and let $f_t^R(x) = g_t \cdot x + R_t(x)$, where R_t is a differentiable convex function. Let Ψ be an arbitrary convex function. Consider the composite-objective mirror-descent algorithm which plays*

$$\hat{x}_{t+1} = \arg\min_x \nabla f_t^R(\hat{x}_t) \cdot x + \Psi(x) + \mathcal{B}_{1:t}(x, \hat{x}_t), \tag{20}$$

and the FTRL algorithm which plays

$$x_{t+1} = \arg\min_x f_{1:t}^R(x) + \phi_{1:t-1} \cdot x + \Psi(x), \tag{21}$$

for $\phi_t \in \partial\Psi(x_{t+1})$ such that $g_{1:t} + \nabla R_{1:t}(x_{t+1}) + \phi_{1:t-1} + \phi_t = 0$. If both algorithms play $\hat{x}_1 = x_1 = 0$, then they are equivalent, in that $x_t = \hat{x}_t$ for all $t > 0$.

Corollary 8. *Let $f_t(x) = g_t \cdot x$ and $f_t^R(x) = g_t \cdot x + \frac{\sigma_t}{2}\|x\|_2^2$. Then the following algorithms play identical points:*

- *FTRL, which plays $x_{t+1} = \arg\min_x f_{1:t}^R(x)$.*
- *Gradient descent on the functions f^R using the step size $\eta_t = \frac{1}{\sigma_{1:t}}$, which plays*

$$x_{t+1} = x_t - \eta_t \nabla f_t^R(x_t)$$

- *Revisionist constant-step size gradient descent with $\eta_t = \frac{1}{\sigma_{1:t}}$, which plays*

$$x_{t+1} = -\eta_t g_{1:t}.$$

For Now

- Algorithms
  - FOBOS
  - RDA
  - FTRL-Proximal
- FTRL-style & Implicit and Composite Updates
  - FTRL: RDA & FTRL-Proximal
  - COMID & FOBOS = FTRL-Proximal
  - AOGD = FTRL (RDA)
- Regret Analysis is omitted

Experiments

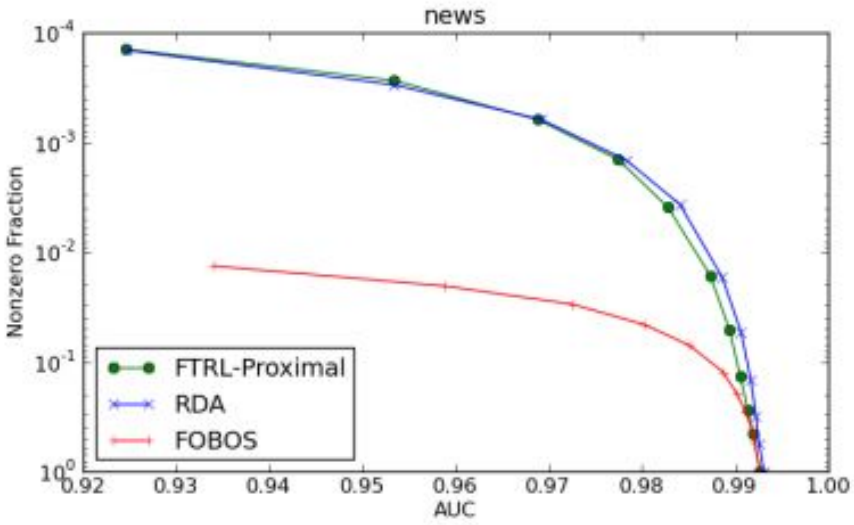
- AUC

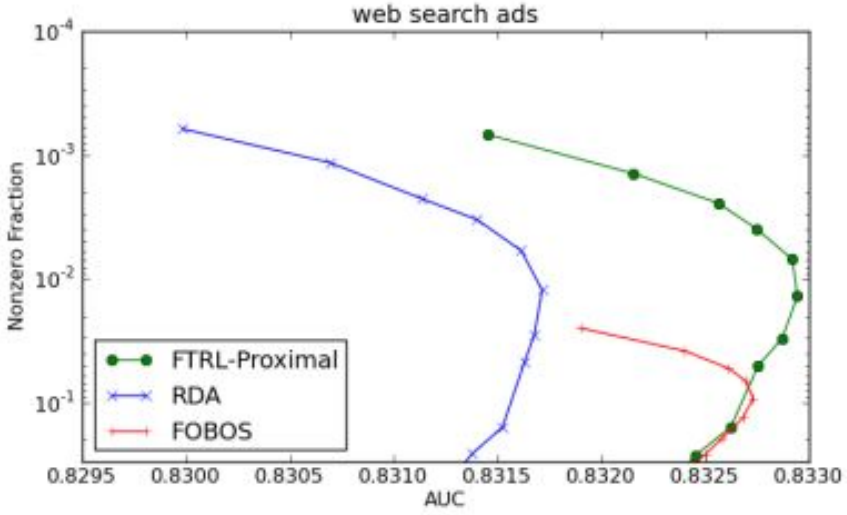
Table 2: AUC (area under the ROC curve) for online predictions and sparsity in parentheses. The best value for each dataset is shown in bold. For these experiments, λ was fixed at $0.05/T$.

DATA	FTRL-PROXIMAL	RDA	FOBOS
BOOKS	0.874 (0.081)	**0.878** (**0.079**)	0.877 (0.382)
DVD	0.884 (0.078)	0.886 (**0.075**)	**0.887** (0.354)
ELECTRONICS	0.916 (0.114)	**0.919** (**0.113**)	0.918 (0.399)
KITCHEN	0.931 (**0.129**)	**0.934** (0.130)	0.933 (0.414)
NEWS	0.989 (**0.052**)	**0.991** (0.054)	0.990 (0.194)
RCV1	0.991 (**0.319**)	**0.991** (0.360)	0.991 (0.488)
WEB SEARCH ADS	**0.832** (**0.615**)	0.831 (0.632)	0.832 (0.849)

Experiments

- Sparisty & AUC
  - FTRL-Proximal & RDA





outline

- 背景
- Online Optimization算法实现
  - Online Optimization
  - Truncated Gradient
  - FOBOS
  - RDA
  - FTPRL
- Online Optimization理论分析
  - Preliminary
  - Algorithms
  - FTRL-style & Implicit and Composite Updates
  - Experiments
- 总结
- References

总结

- Online Optimization算法实现
  - Batch -> Online
  - 简单截断、梯度截断、FOBOS
  - RDA、FTRL-Proximal
- Online Optimization理论分析
  - Algorithms
  - FTRL-style & Implicit and Composite Updates
  - Experiments

References

- FOBOS: Efficient learning using forward-backward splitting
- RDA: Dual averaging methods for regularized stochastic learning and online optimization
- FTRL: A unified view of regularized dual averaging and mirror descent with implicit updates
- http://www.wbrecom.com/?p=264
- http://www.seas.ucla.edu/~vandenbe/236C/lectures/qnewton.pdf
- http://www.slideshare.net/VolhaBanadyseva/thomas-jensen-machine-learning
- Mirror Descent: http://www.stats.ox.ac.uk/~lienart/blog_opti_mda.html
- explicit & implicit update: http://www.math.ucla.edu/~wotaoyin/summer2013/slides/Lec05_ProximalOperatorDual.pdf