AdaGrad: Automatic Learning Rates

CS 294 Presentation

Arturo Fernandez

February 10, 2016

Table of contents

Needles in a Haystack

1. Why do all features have the same learning rate if they vary in importance and frequency seen?

"*Informally, our procedure gives frequently occuring features very low learning rates and infrequent features high learning rates, where the intuition is that each time an infrequent feature is seen, the learner should take notice.*"

2. Sparsity a key feature and assumption of framework (in terms of better regret analysis).

Needles in a Haystack

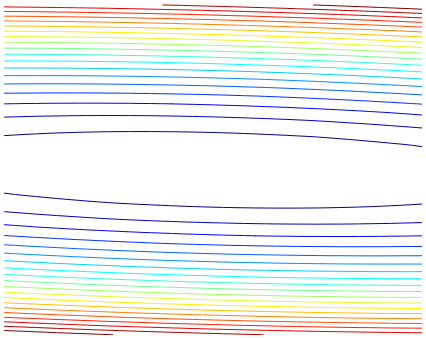
1. Why do all features have the same learning rate if they vary in importance and frequency seen?

"*Informally, our procedure gives frequently occuring features very low learning rates and infrequent features high learning rates, where the intuition is that each time an infrequent feature is seen, the learner should take notice.*"

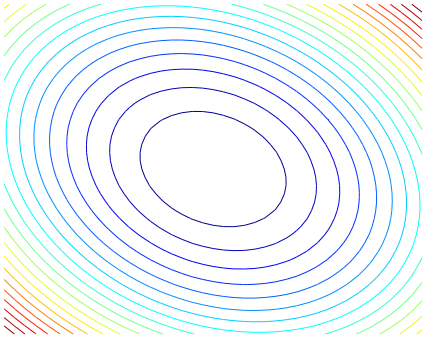
2. Sparsity a key feature and assumption of framework (in terms of better regret analysis).

Motivation

Why adapt to geometry?



Hard



Nice

y_t	$\phi_{t,1}$	$\phi_{t,2}$	$\phi_{t,3}$
1	1	0	0
-1	.5	0	1
1	-.5	1	0
-1	0	0	0
1	.5	0	0
-1	1	0	0
1	-1	1	0
-1	-.5	0	1

1. Frequent, irrelevant
2. Infrequent, predictive
3. Infrequent, predictive

Figure: show an example picture

Notation

- $\mathcal{X}$ = Closed, convex set which we restrict our parameter space to
- Bregman Divergence associated with a *strictly convex* and *differentiable* function ψ is

$$B_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y \rangle$$

Example: $\psi = \|\cdot\|$

$$d^2(x, y) = \|x - y\|^2 = \langle x - y, x - y \rangle = \|x\|^2 - \|y\|^2 - \langle 2y, x - y \rangle$$

Stochastic Optimization and Online Learning

Measure stochastic optimization efficacy by regret analysis

- Loss Function: $\phi_t(x) = f_t(x) + \varphi_t(x)$

Want to minize regret w.r.t the best static predictor

$$R_\phi(T) = \sum_{t=1}^{T} \phi_t(x_t) - \inf_{x \in \mathcal{X}} \sum_{t=1}^{T} \phi_t(x)$$

We'll denote the minimizer as x^*.

Algorithms under study

Primal-Dual Subgradient Method / Regularized Dual Averaging

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \eta \langle \bar{g}_t, x \rangle + \eta \phi(x) + \frac{1}{t} \psi_t(x) \right\}$$

where $\bar{g}_t = \frac{1}{t} \sum_{\tau=1}^{t} g_\tau$.

Composite Mirror Descent / Forward-backward splitting

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \eta \langle g_t, x \rangle + \eta \phi(x) + \frac{1}{t} B_{\psi_t}(x, x_t) \right\}$$

A temporary simplification

Let $\phi = 0$ *form now on*. Note that if $\psi = \|\cdot\|$ we have standard results.

Dual Averaging

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \frac{\eta_t}{t} \sum_{\tau=1}^{t} \langle g_\tau, x \rangle + \frac{1}{2t} \|x\|^2 \right\}$$

Gradient Descent

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \eta_t \langle g_t, x \rangle + \frac{1}{2} \|x - x_t\|^2 \right\}$$

Adapting to the Geometry of the Space

- Receive $g_t \in \partial f_t(x_t)$
- Gradient Descent:

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \frac{1}{2}\|x - x_t\|^2 + \eta_t \langle g_t, x \rangle \right\}$$

- Let $\|x\|_A^2 = \langle x, Ax \rangle$ with $A \succeq 0$.

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \left\{ \frac{1}{2}\|x - x_t\|_A^2 + \eta_t \langle g_t, x \rangle \right\}$$

Adapting to the Geometry of the Space

- Standard Regret Bound (Zankevich)

$$\sum_{t=1}^{T} [\, f_t(x_t) - f_t(x^*) \,] \leq \frac{1}{2\eta} \|x_1 - x^*\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2$$

- Regret Bound with Matrix

$$\sum_{t=1}^{T} [\, f_t(x_t) - f_t(x^*) \,] \leq \frac{1}{2\eta} \|x_1 - x^*\|_A^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|_{A^{-1}}^2$$

Adapting to the Geometry of the Space

- Standard Regret Bound (Zankevich)

$$\sum_{t=1}^{T} [\, f_t(x_t) - f_t(x^*) \,] \leq \frac{1}{2\eta} \|x_1 - x^*\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2$$

- Regret Bound with Matrix

$$\sum_{t=1}^{T} [\, f_t(x_t) - f_t(x^*) \,] \leq \frac{1}{2\eta} \|x_1 - x^*\|_A^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|_{A^{-1}}^2$$

- What happens if we minimze A in hindsight?

$$\min_{A} \sum_{t=1}^{T} \langle g_t, A^{-1} g_t \rangle \quad \text{subject to} \quad A \succeq 0,\ \operatorname{tr}(A) \leq C$$

Adapting to the Geometry of the Space

- Standard Regret Bound (Zankevich)

$$\sum_{t=1}^{T} [\, f_t(x_t) - f_t(x^*) \,] \leq \frac{1}{2\eta} \|x_1 - x^*\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|^2$$

- Regret Bound with Matrix

$$\sum_{t=1}^{T} [\, f_t(x_t) - f_t(x^*) \,] \leq \frac{1}{2\eta} \|x_1 - x^*\|_A^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|g_t\|_{A^{-1}}^2$$

- What happens if we minimze A in hindsight?

$$\min_A \sum_{t=1}^{T} \langle g_t, A^{-1} g_t \rangle \quad \text{subject to} \quad A \succeq 0,\ \text{tr}(A) \leq C$$

Solutions

Let $G_T = \sum_{t=1}^{T} g_t g_t^T$. Below, c is chosen to satisty trace constraint.

Unrestricted Form:

$$A = c \left(\sum_{t=1}^{T} g_t g_t^T \right)^{1/2} = c \cdot G_T^{1/2}$$

Too expensive to compute $O(d^3)$ for Root and use $O(d^2)$ for MatVec

Diagonal:

$$A = c \cdot \mathbf{Diag} \left(\sum_{t=1}^{T} g_t g_t^T \right)^{1/2} = c \cdot \mathbf{Diag}(G_T)^{1/2}$$

Inverse and root of diagonal matrix in $O(n)$

Final Regret bound

Let $g_{1:t} = [g_1|\dots|g_t] \in \mathbb{R}^{d\times t}$ and $g_{1:t,j}$ be the j-th row of $g_{1:t}$.

Set $s_t = [\|g_{1:t,j}\|_2]_{j=1}^{d}$, $A_t = \mathbf{Diag}(s_t)$,

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \left\{ \frac{1}{2}\|x - x_t\|_{A_t}^2 + \eta\langle g_t, x\rangle \right\}$$

and let

$$D_\infty := \max_t \|x_t - x^*\|_\infty \leq \sup_{x\in\mathcal{X}} \|x - x^*\|_\infty$$

Theorem

$$R_f(T) \leq 2D_\infty \sum_{j=1}^{d} \|g_{1:T,j}\|_2 = \sqrt{2d}D_\infty \sqrt{\inf_{s\succeq 0, \langle 1,s\rangle \leq d} \sum_{t=1}^{T} \|g_t\|^2_{\mathbf{Diag}(s)^{-1}}}$$

Some notes

- Tighter results from full matrix but computationally expensive so we skip.
- Can compare previous boun to Zinkevich online gradient algorithm which gives

$$R_f(T) \leq \sqrt{2} D_2 \sqrt{\sum_{t=1}^{T} \|g_t\|^2}$$

  when the optimal η is chosen in *hindisight* and $\sup_{x,y \in \mathcal{X}} \|x - y\|_2 \leq D_2$.
- When $\mathcal{X} - = \{x : \|x\|_\infty \leq 1\}$, $D_2 = 2\sqrt{d}$ and $D_\infty = 2$
- Tight in the minimax sense so need assumptions: Gradient Sparsity.

SVM / Hinge Loss Example

$f_t(x) = [1 - y_t \langle z_t, x \rangle]_+$ where $z_t \in \{-1, 0, 1\}^d$

- If $z_{t(j)} \neq 0$ with probability $\propto j^{-\alpha}$ for $\alpha > 1$

$$\mathbb{E}\left[\sum_{i=1}^{d} \|g_{1:T,i}\|_2\right] \leq \sum_{i=1}^{d} \sqrt{p_i T} = O\left(\max\{\log d, d^{1-\alpha/2}\}\sqrt{T}\right)$$

- Online Gradient Descent (OGD) yields best case regret $O(\sqrt{dT})$

AdaGrad *can* be exponentially snaller in the dimension d.

The Algorithm

INPUT: $\eta > 0$, $\delta \geq 0$
VARIABLES: $s \in \mathbb{R}^d$, $H \in \mathbb{R}^{d \times d}$, $g_{1:t,i} \in \mathbb{R}^t$ for $i \in \{1, \ldots, d\}$
INITIALIZE $x_1 = 0$, $g_{1:0} = []$
FOR $t = 1$ to T
Suffer loss $f_t(x_t)$
Receive subgradient $g_t \in \partial f_t(x_t)$ of f_t at x_t
UPDATE $g_{1:t} = [g_{1:t-1}\ g_t]$, $s_{t,i} = \|g_{1:t,i}\|_2$
SET $H_t = \delta I + \mathrm{diag}(s_t)$, $\psi_t(x) = \frac{1}{2}\langle x, H_t\, x\rangle$

Primal-Dual Subgradient Update (3):

$$x_{t+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \eta \left\langle \frac{1}{t} \sum_{\tau=1}^{t} g_\tau, x \right\rangle + \eta\varphi(x) + \frac{1}{t}\psi_t(x) \right\}.$$

Composite Mirror Descent Update (4):

$$x_{t+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \{\eta\langle g_t, x\rangle + \eta\varphi(x) + B_{\psi_t}(x, x_t)\}.$$

Figure 1: ADAGRAD with diagonal matrices

ℓ_1-regularization

Let $\varphi(x) = \lambda\|x\|_1$ and $H_t = \delta I + \mathbf{Diag}(s_t)$, which has ith diagonal element $H_{t,ii} = \delta + \|g_{1:t,i}\|_2$.

- The primal-dual subgradient update is

$$x_{t+1,i} = \text{sgn}(-\bar{g}_{t,i})\frac{\eta t}{H_{t,ii}}\left[|\bar{g}_{t,i}| - \lambda\right]_+$$

- The Composite-mirror descent update is

$$x_{t+1,i} = \text{sgn}\left(x_{t,i} - \frac{\eta}{H_{t,ii}}g_{t,i}\right)\left[\left|x_{t,i} - \frac{\eta}{H_{t,ii}}g_{t,i}\right| - \frac{\lambda t}{H_{T,ii}}\right]_+$$

Lazy Computation

For both updates it is clear that we can perform "lazy" computation when the gradient vectors are sparse, a frequently occurring setting when learning for instance from text corpora. Suppose that from time step t_0 through t, the ith component of the gradient is 0. Then we can evaluate the above updates on demand since $H_{t,ii}$ remains intact. For composite mirror-descent, at time t when $x_{t,i}$ is needed, we update

$$x_{t,i} = \operatorname{sign}(x_{t_0,i}) \left[|x_{t_0,i}| - \frac{\lambda\eta}{H_{t_0,ii}}(t - t_0) \right]_+ .$$

Even simpler just in time evaluation can be performed for the the primal-dual subgradient update. Here we need to keep an unnormalized version of the average $\bar{g}_t$. Concretely, we keep track of $u_t = t\bar{g}_t = \sum_{\tau=1}^{t} g_\tau = u_{t-1} + g_t$, then use the update (24):

$$x_{t,i} = \operatorname{sign}(-u_{t,i}) \frac{\eta t}{H_{t,ii}} \left[\frac{|u_{t,i}|}{t} - \lambda \right]_+ ,$$

where H_t can clearly be updated lazily in a similar fashion.

Text Classification

Reuters RCV1 document classification task—$d = 2 \cdot 10^6$ features, approximately 4000 non-zero features per document

$$f_t(x) := [1 - \langle x, \xi_t \rangle]_+$$

where $\xi_t \in \{-1, 0, 1\}^d$ is data sample

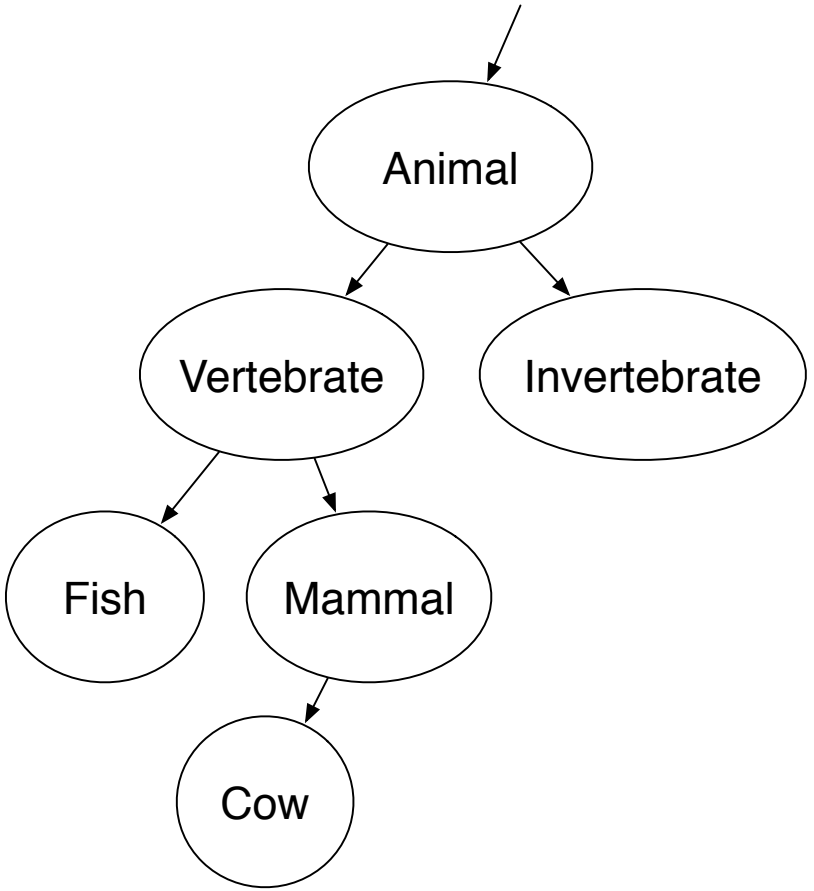
	FOBOS	AdaGrad	PA[1]	AROW[2]
Ecomonics	.058 (.194)	**.044** (**.086**)	.059	.049
Corporate	.111 (.226)	**.053** (**.105**)	.107	.061
Government	.056 (.183)	**.040** (**.080**)	.066	.044
Medicine	.056 (.146)	**.035** (**.063**)	.053	.039

Test set classification error rate
(sparsity of final predictor in parenthesis)

[1] Crammer et al., 2006
[2] Crammer et al., 2009

Image Ranking

ImageNet (Deng et al., 2009), large-scale hierarchical image database



Train 15,000 rankers/classifiers to rank images for *each* noun (as in Grangier and Bengio, 2008)

Data
$\xi = (z^1, z^2) \in \{0,1\}^d \times \{0,1\}^d$ is pair of images

$$f(x; z^1, z^2) = \left[1 - \left\langle x, z^1 - z^2 \right\rangle\right]_+$$

Image Ranking Results

Precision at k: proportion of examples in top k that belong to category. Average precision is average placement of all positive examples.

Algorithm	Avg. Prec.	P@1	P@5	P@10	Nonzero
AdaGrad	**0.6022**	0.8502	0.8130	0.7811	**0.7267**
AROW	0.5813	**0.8597**	**0.8165**	**0.7816**	1.0000
PA	0.5581	0.8455	0.7957	0.7576	1.0000
Fobos	0.5042	0.7496	0.6950	0.6545	0.8996

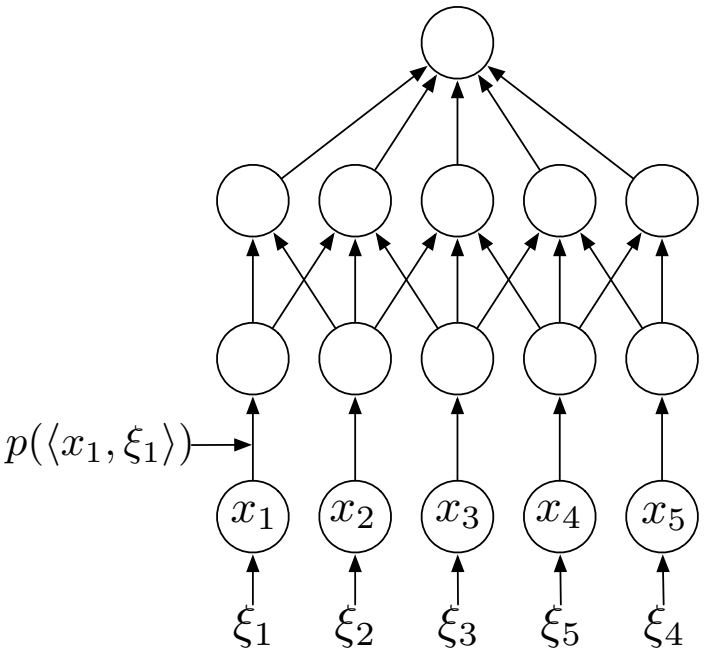
Neural Network Learning

Wildly non-convex problem:

$$f(x;\xi) = \log\left(1 + \exp\left(\left\langle\left[p(\langle x_1, \xi_1\rangle) \ \cdots \ p(\langle x_k, \xi_k\rangle)\right], \xi_0\right\rangle\right)\right)$$

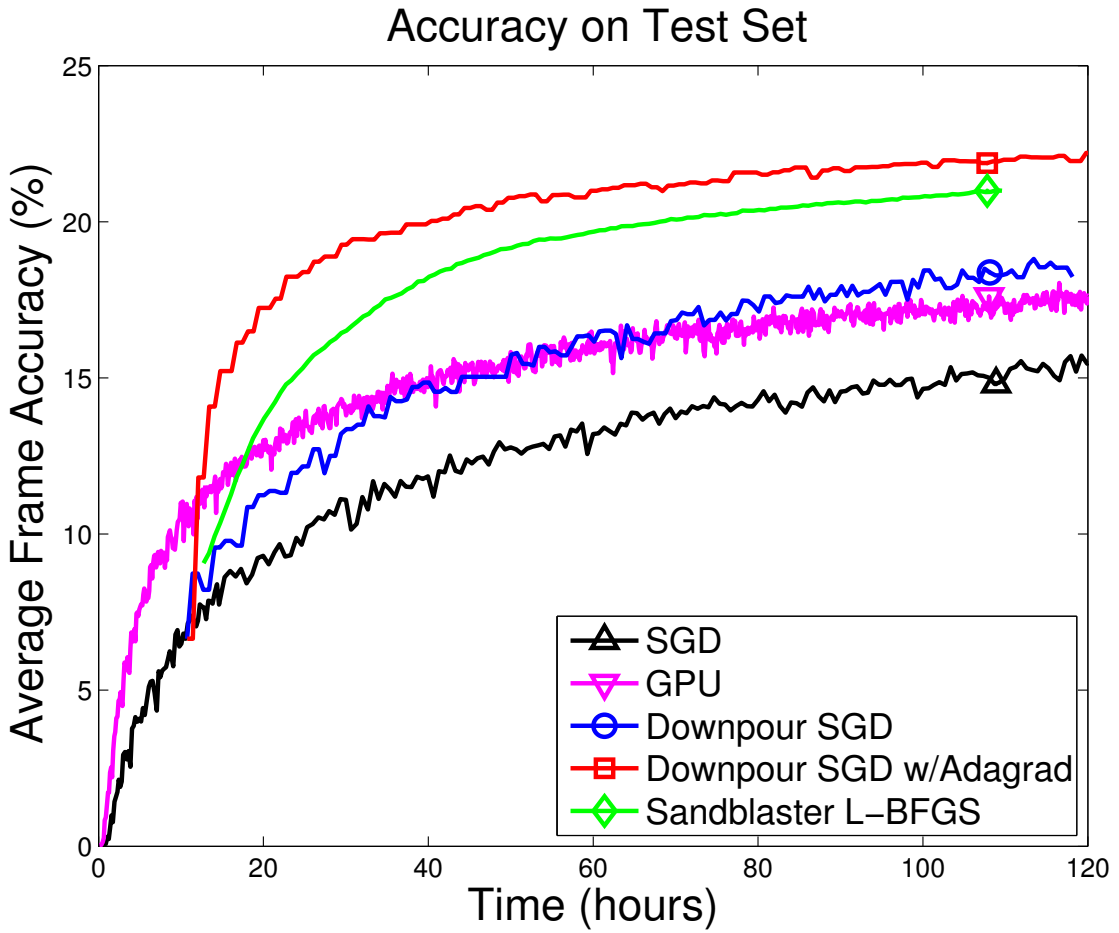
where

$$p(\alpha) = \frac{1}{1+\exp(\alpha)}$$



Idea: Use stochastic gradient methods to solve it anyway

Neural Network Learning



(Dean et al. 2012)

Distributed, $d = 1.7 \cdot 10^9$ parameters. SGD and AdaGrad use 80 machines (1000 cores), L-BFGS uses 800 (10000 cores)

Questions

- AdaDelta: Diagonal Hessian approximation to use some second order information (slightly better, marginally)
- Any ideas to translate to dense case?

References

- Duchi et al. Adaptive subgradient methods for online learning and stochastic optimization
- Duchi et al. Adagrad slides
- Karpathy Convolutional Neural Net Trainer Comparison
- Perla Notes on Adagrad
- Reid Meet the Bregman Divergences