10-425/625: Introduction to Convex Optimization (Fall 2023)

Lecture 21: Adaptive Step Sizes

Instructor:[1] *Matt Gormley* *November 20, 2023*

21.1 Adaptive step sizes

Motivation Another big topic in stochastic optimization these days is *adaptive step sizes*.

To motivate, let's consider a logistic regression problem, where x_{ij} are binary, and many of them are zero. For example, classifying if a given movie review is positive or negative:

Piece of subtle art. Maybe a masterpiece. Doubtlessly a special story about the ambiguity of existence.

Some words are common (blue) and uninformative and some rare (green) and informative. Here:

- x_{ij} represents whether the jth word is present in ith review
- y_i represents the ith review is positive or negative (sentiment)

If we aim to minimize the negative log-likelihood of this binary logistic regression model, then our overall objective is $f(\beta) = \sum_{i=1}^n f_i(\beta)$ for n training examples, where f_i is the the cross entropy loss,

$$\ell(\hat{y}, y_i) = -y_i \log(p(y_i = 1 \mid x_i, \beta)) - (1 - y_i) \log(p(y_i = 0 \mid x_i, \beta)),$$

applied to our model $p(y_i = 1 \mid x_i, \beta) = \frac{1}{1+\exp(-x_i^T \beta)}$ which we can simplify to:

$$f_i(\beta) = -y_i x_i^T \beta + \log(1 + \exp(x_i^T \beta))$$

The gradient of this per-example objective is:

$$\nabla f_i(\beta) = \left(-y_i + \frac{1}{1 + \exp(-x_i^T \beta)} \right) x_i.$$

[1]These notes were originally written by Ryan Tibshirani for 10-725 Fall 2019 (original version: here) and were edited and adapted for 10-425/625.

Observation: For some feature j, $x_{ij} = 0$ implies that $\nabla_{\beta_j} f_i(\beta) = 0$. Also $\|\nabla f_i(\beta)\|_2$ is large when ith review is misclassified.

So what does SGD do?

- Gives equal weight to common and to rare informative words.
- Diminishing step sizes t_k means the rare informative features are learned very slowly ...

To escape this long wait, we'll have to adapt the step sizes to pick up the informative features.

21.1.1 Example Algorithm: AdaGrad

AdaGrad (Duchi, Hazan, and Singer 2010): very popular adaptive method. Let $g^{(k)} = \nabla f_{i_k}(x^{(k-1)})$, and update for $j = 1, \ldots, p$:

$$x_j^{(k)} = x_j^{(k-1)} - \alpha \frac{g_j^{(k)}}{\sqrt{\sum_{\ell=1}^k (g_j^{(\ell)})^2 + \epsilon}}$$

Notes:

- AdaGrad does not require tuning learning rate: $\alpha > 0$ is fixed constant, learning rate decreases naturally over iterations
- Learning rate of rare informative features diminishes slowly
- Can drastically improve over SGD in sparse problems
- Main weakness is monotonic accumulation of gradients in the denominator ... AdaDelta, Adam, AMSGrad, etc. improve on this, popular in training deep nets

21.2 Preliminaries

21.2.1 Notation

For vectors x and y, we use $\langle x, y \rangle$ to denote their dot product. We also use $[x]_+ = max(0, x)$.

21.2.2 Setting

We consider the settings of stochastic learning and online optimization. Below we formalize the problem as online learning.

Regularized loss minimization Regularized loss minimization yields the optimization problem:

$$w^* = \arg\min_{w \in \Omega} \frac{1}{n} \sum_{t=1}^{n} f_t(w) + r(w) \tag{21.1}$$

where $w \in \mathbb{R}^d$ are the model parameters, $f_t : \Omega \to \mathbb{R}$ is a loss function, and $r : \Omega \to \mathbb{R}$ is a reguralization function. Ω is a convex set of parameters. f_t is differentiable and convex. r is convex.

Example regularizers include

- ℓ_1-regularization, $r(w) = \lambda ||w||_1$
- ℓ_2^2-regularization, $r(w) = \frac{\lambda}{2} ||w||_2^2$. This is equivalent to a Guassian prior on the parameters where λ is the inverse variance.

Online Learning In the online learning setting, we choose a sequence of parameters w_t for $t = 1, 2, 3, \ldots$. At each time step t, some adversary gives us another loss function f_t and we receive the loss $f_t(w_t)$.

Regret The goal is then to ensure that the total loss up to each time step T, $\sum_{t=1}^{T} f_t(w_t)$ is not much worse (larger) than $\min_w \sum_{t=1}^{T} f_t(w)$, which is the smallest total loss of any fixed set of parameters w chosen retrospectively.

$$R_T(w) := \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w) \tag{21.2}$$

Regularized regret The regularized regret simply incorporates the regularizer.

$$R_T(w) := \sum_{t=1}^{T} (f_t(w_t) + r(w_t)) - \sum_{t=1}^{T} (f_t(w) - r(w)) \tag{21.3}$$

Our goal is to choose an algorithm which bounds this (regularized) regret.

21.3 Bregman Divergences

Associated with this convex function, is a Bregman divergence, i.e. given $x, y \in D$:

$$D_\Phi(x, y) = \Phi(x) - \Phi(y) - \nabla\Phi(y)^T(x - y).$$

Given this Bregman divergence and any point y (potentially outside C but inside D) we can define the Bregman projection,

$$\Pi_C(y) = \arg\min_{x \in C} D_\Phi(x, y).$$

There are some main examples to keep in mind for this lecture:

1. **Usual Gradient Descent:** Suppose we take $\Phi(x) = \frac{1}{2}\|x\|_2^2$ (this is a 1-strongly convex function with respect to the Euclidean norm). Then we get,

   $$D_\Phi(x, y) = \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2 - y^T(x - y) = \frac{1}{2}\|x - y\|_2^2.$$

   As we will see in a little while our mirror descent updates in this case are identical to our (projected) GD updates from before.

2. **Exp Gradient Descent:** Suppose we take $\Phi(x) = \sum_{i=1}^d x_i \log x_i$, which is defined over the (strictly) positive reals. We get,

   $$\begin{aligned} D_\Phi(x, y) &= \sum_{i=1}^d x_i \log x_i - \sum_{i=1}^d y_i \log y_i - \sum_{i=1}^d (1 + \log y_i)(x_i - y_i) \\ &= \sum_{i=1}^d x_i \log(x_i/y_i) - \sum_{i=1}^d (x_i - y_i). \end{aligned}$$

   It turns out that $\Phi(x)$ is strictly convex over the simplex *with respect to the ℓ_1-norm*. To see this, we recall Pinsker's inequality (you might have seen this in a Stats class like 36-705/36-709) which tells us that for two distributions p, q (vectors on the d-dimensional simplex):

   $$\ell_1(p, q) \leq \sqrt{2\mathrm{KL}(p, q)}.$$

Thus, over the simplex we see that,

$$D_\Phi(x, y) = \mathrm{KL}(x, y) \geq \frac{1}{2}\|x - y\|_1^2,$$

i.e. equivalently Φ is 1-strongly convex with respect to the ℓ_1-norm on the simplex.

21.3.1 Properties of Bregman Divergences

There are a few properties of Bregman divergences that will be useful in our proof of the rate of convergence of mirror descent.

Lemma 21.1 (Three-point Property)**.** *For $x, y, z \in D$,*

$$D_\Phi(x, y) + D_\Phi(z, x) - D_\Phi(z, y) = (\nabla\Phi(x) - \nabla\Phi(y))^T(x - z).$$

Proof: We simply use the definition of the Bregman divergence. ∎

Lemma 21.2 (Pythagoras Theorem)**.** *Suppose that C is a convex set, $x \in C$ and $y \in \mathbb{R}^d$. Then,*

$$D_\Phi(x, \Pi_C(y)) + D_\Phi(\Pi_C(y), y) \leq D_\Phi(x, y).$$

Proof: We simply use the first-order optimality conditions for the Bregman projection, i.e. we know that,

$$\Pi_C(y) = \arg\min_{x \in C} D_\Phi(x, y),$$

so this means that,

$$(\nabla\Phi(\Pi_C(y)) - \nabla\Phi(y))^T(\Pi_C(y) - x) \leq 0,$$

for any $x \in C$. This is the claimed result. ∎

21.4 Algorithms

21.4.1 Stochastic Gradient Descent

SGD defines a simple update for each iteration.

$$w_{t+1} = w_t - \eta_t(f_t'(w_t) + r'(w_t)) \tag{21.4}$$

where $f_t'(w)$ is the gradient of f_t or one of its subgradients at point w, and $r'(w)$ is equivalently a gradient or subgradient of $r(w)$.

21.4.2 Mirror Descent

Let $\phi_t = f_t + r$ denote the sum of the loss function and regularizer at time t. The update for Mirror Descent [7, 1] is then,

$$w_{t+1} = \arg\min_{w \in \Omega} \eta \langle \phi_t'(w_t), w - w_t \rangle + B_\psi(w, w_t) \tag{21.5}$$

$$= \arg\min_{w \in \Omega} \eta \langle f_t'(w_t) + r'(w_t), w - w_t \rangle + B_\psi(w, w_t) \tag{21.6}$$

where B_ψ is a Bregman divergence and ϕ_t' is a subgradient of ϕ_t.

The Bregman divergence for ψ is defined as:

$$B_\psi(w, v) = \psi(w) - \psi(v) - \langle \nabla\psi(v), w - v \rangle \tag{21.7}$$

where $\nabla\psi$ is the gradient of ψ.

Intuition: This MD update minimizes a linear approximation of the function ϕ_t at the current parameters w_t while ensuring that the next w_{t+1} is close to w_t. Notice that we could equivalently replace the second term in the update with the first-order taylor expansion of ϕ_t at w_t since $\phi_t(w_t)$ is constant w.r.t. w. This would give the equivalent update:

$$w_{t+1} = \arg\min_{w \in \Omega} \eta(\phi_t(w_t) + \langle \phi_t'(w_t), w - w_t \rangle) + B_\psi(w, w_t) \tag{21.8}$$

Mirror Map The mirror map ψ must have two properties:

1. continuously differentiable, and
2. α-strongly convex with respect to a norm $||\cdot||$ on the set of possible values w.

An example of such a function would be $\psi(w) = \frac{1}{2}||w||_2^2$.

Local Approximation Description of Mirror Descent A natural way to generalize the gradient descent algorithm is simply to use a general Bregman divergence to measure proximity in the local linear approximation of gradient descent. Despite being a seemingly minor modification to the update step, it's worth noting that changing this term essentially re-shapes the space we're optimizing over in a non-trivial way – changing how we measure distances is similar to stretching and shrinking the space around the current iterate, and is also at the heart of things like Newton's method.

Concretely, given our current iterate x^t we compute the next iterate by solving the program:

$$x^{t+1} = \arg\min_{x \in C} f(x^t) + \nabla f(x^t)^T (x - x^t) + \frac{1}{\eta} D_\Phi(x, x^t).$$

We cannot always solve for this iteration in closed-form (similar to how we can't always solve a prox. computation in closed form). However, it will turn out that for nice mirror maps ϕ this iteration has a simple description.

21.4.3 Composite Objective Mirror Descent

Composite objective mirror descent (COMID) [4] uses the following update.

$$w_{t+1} = \arg\min_{w \in \Omega} \eta \langle f_t'(w_t), w - w_t \rangle + \eta r(w) + B_\psi(w, w_t) \tag{21.9}$$

This update is identical to that of Mirror Descent in Eq. (21.6), except that *we do not linearize* $r(w)$, but instead include it directly in the minimization.

For many choices of $r(w)$, this update has a closed form.

There are several first-order algorithms which are **special cases** of composite objective mirror descent:

- Forward-backward splitting (e.g. [8, 2])
- Projected gradient method
- Mirror descent
- Truncated gradient [6]

21.4.4 Regularized Dual Averaging

Regularized Dual Averaging (RDA) [9] has the following update. Here ψ_t is called the *proximal* term.

$$\bar{g}_t = \frac{t-1}{t} \bar{g}_{t-1} + \frac{1}{t} f_t'(w_t) \tag{21.10}$$

$$w_{t+1} = \arg\min_{w \in \Omega} \eta \langle \bar{g}_t, w \rangle + \eta r(w) + \frac{1}{t} \psi_t(w) \tag{21.11}$$

where $\eta > 0$ is a fixed step size, and $\bar{g}_t$ keeps a running average of the subgradients:

$$\bar{g}_t = \frac{1}{t} \sum_{s=1}^{t} f_s'(w_s) \tag{21.12}$$

Like Composite Objective Mirror Descent, the regularizer r is included in its entirety and not linearized.

Again, for many choices of $r(w)$, this update has a closed form.

Intuition: The RDA update minimizes a linear term involving the average gradient, the full regularizer, and an additional function ψ_t which is strongly convex.

21.4.5 AdaGrad

The AdaGrad family of algorithms [5, 3] are defined by the Composite Objective Mirror Descent and Regularized Dual Averaging updates for a particular choice of ψ_t which is "adapted over time in a data-driven way."

$$\text{COMID:} \quad w_{t+1} = \arg\min_{w \in \Omega} \eta \left\langle f_t'(w_t), w - w_t \right\rangle \quad + \eta r(w) + B_\psi(w, w_t) \tag{21.13}$$

$$\text{RDA:} \quad w_{t+1} = \arg\min_{w \in \Omega} \eta \left\langle \bar{g}_t, w \right\rangle \quad + \eta r(w) + \frac{1}{t}\psi_t(w) \tag{21.14}$$

The key contribution of AdaGrad is defining the proximal functions to be the squared Mahalanobis norm:

$$\psi_t(w) = \frac{1}{2} \left\langle w, H_t w \right\rangle \tag{21.15}$$

$$\text{where } H_t = \delta I + \text{diag}(G_t)^{1/2} \qquad \text{(Diagonal)} \tag{21.16}$$

$$\text{or } H_t = \delta I + G_t^{1/2} \qquad \text{(Full Matrix)} \tag{21.17}$$

$$\text{and } G_t = \sum_{s=1}^{t} f_s'(w)^T f_s'(w) \tag{21.18}$$

In the diagonal case, the values in $H_t, ii = \delta + \sqrt{\sum_{s=1}^{t} (f_s'(w)_i)^2}$ are the sum of the squares of the ith element of the gradient over all time steps up to t.

With this definition of ψ_t, the updates can then be simplified to:

$$\text{COMID:} \quad w_{t+1} = \arg\min_{w\in\Omega} \langle \eta f_t'(w_t) - H_t w_t, w\rangle \quad + \eta r(w) + \frac{1}{2}\langle w, H_t w\rangle \tag{21.19}$$

$$\text{RDA:} \quad w_{t+1} = \arg\min_{w\in\Omega} \langle \eta t \bar{g}_t, w\rangle \quad + \eta r(w) + \frac{1}{2}\langle w, H_t w\rangle \tag{21.20}$$

21.5 Summary of Algorithm Updates

Below, we reiterate all the updates in one place:

$$\begin{aligned}
\text{SGD:} \quad & w_{t+1} = w_t - \eta_t(f_t'(w_t) + r'(w_t)) \\
\text{MD:} \quad & w_{t+1} = \arg\min_{w\in\Omega} \eta \langle f_t'(w_t), w - w_t\rangle + \eta r(w) + B_\psi(w, w_t) \\
\text{COMID:} \quad & w_{t+1} = \arg\min_{w\in\Omega} \eta \langle f_t'(w_t), w - w_t\rangle + \eta r(w) + B_\psi(w, w_t) \\
\text{RDA:} \quad & w_{t+1} = \arg\min_{w\in\Omega} \eta \langle \bar{g}_t, w\rangle + \eta r(w) + \frac{1}{t}\psi_t(w) \\
\text{AdaGrad-COMID:} \quad & w_{t+1} = \arg\min_{w\in\Omega} \eta \langle f_t'(w_t) - H_t w_t, w\rangle + \eta r(w) + \frac{1}{2}\langle w, H_t w\rangle \\
\text{AdaGrad-RDA:} \quad & w_{t+1} = \arg\min_{w\in\Omega} \eta \langle t\bar{g}_t, w\rangle + \eta r(w) + \frac{1}{2}\langle w, H_t w\rangle
\end{aligned}$$

21.6 Derived Algorithms

ℓ_1-regularization For the regularizer $r(w) = \lambda||w||_1$, we have the following updates.

$$\text{RDA:} \quad w_{t+1,i} = \text{sign}(-\bar{g}_{t,i})\eta\sqrt{t}[|\bar{g}_{t,i}| - \lambda]_+ \tag{21.21}$$

$$\text{AdaGrad-RDA:} \quad w_{t+1,i} = \text{sign}(-\bar{g}_{t,i})\frac{\eta t}{H_{t,ii}}[|\bar{g}_{t,i}| - \lambda]_+ \tag{21.22}$$

$$\text{Fobos (COMID):} \quad w_{t+1,i} = \text{sign}(w_{t,i} - \eta_t g_{t,i}\left[|w_{t,i} - \eta_t g_{t,i}| - \eta_t\lambda\right]_+ \tag{21.23}$$

$$\text{AdaGrad-COMID:} \quad w_{t+1,i} = \text{sign}\left(w_{t,i} - \frac{\eta}{H_{t,ii}}g_{t,i}\right)\left[\left|w_{t,i} - \frac{\eta}{H_{t,ii}}g_{t,i}\right| - \frac{\lambda\eta}{H_{t,ii}}\right]_+ \tag{21.24}$$

(21.25)

where $[x]_+ = \max(0, x)$.

References

[1] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. URL `http://www.sciencedirect.com/science/article/pii/S0167637702002316`.

[2] J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research*, 10:2899–2934, 2009. URL `http://dl.acm.org/citation.cfm?id=1755882`.

[3] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *COLT*, 2010. URL `http://www.colt2010.org/papers/023Duchi.pdf`.

[4] J. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari. Composite objective mirror descent. 2010. URL `http://eprints.pascal-network.org/archive/00007140/`.

[5] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011. URL `https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf`.

[6] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. In *Advances in neural information processing systems*, pages 905–912, 2009. URL `http://papers.nips.cc/paper/3585-sparse-online-learning-via-truncated-gradient`.

[7] A. Nemirovsky and D. Yudin. *Problem complexity and method efficiency in optimization.* Wiley, New York, 1983.

[8] Y. Singer and J. C. Duchi. Efficient learning using forward-backward splitting. In *Advances in Neural Information Processing Systems*, pages 495–503, 2009. URL `http://papers.nips.cc/paper/3793-efficient-learning-using-forward-backward-splitting.pdf`.

[9] L. Xiao. Dual averaging method for regularized stochastic learning and online optimization. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2009. URL `http://papers.nips.cc/paper/`

3882-dual-averaging-method-for-regularized-stochastic-learning-and-online-opti
pdf.