Is Mirror Descent a special case of Exponential Weights?

Dirk van der Hoeven and Tim van Erven

08-11-2017

Universiteit Leiden
The Netherlands

Setting: Online Linear Optimization

We consider the Online Linear Optimization setting, which proceeds in rounds $t = 1, \ldots, T$. In each round t we

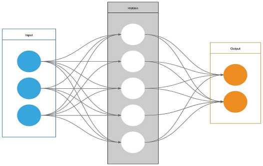
1. Choose a point $\boldsymbol{w}_t \in \mathcal{K} \in \mathit{reals}^d$, where $\mathcal{K}$ is a convex set.
2. Receive gradient of convex loss function $\boldsymbol{g}_t = \nabla f_t(\boldsymbol{w}_t)$
3. suffer loss $\langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle$

Goal: keep regret $\mathcal{R}_T(\boldsymbol{u})$ small

$$\mathcal{R}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle - \min_{\boldsymbol{u} \in \mathcal{K}} \sum_{t=1}^{T} \langle \boldsymbol{u}, \boldsymbol{g}_t \rangle$$

Motivation

The Online Linear Optimization setting has many uses, among them are training neural networks [1], gambling [2], spam filtering [3], and portfolio selection[4].



1 Neural network image by LearnDataSci from www.learndatasci.com

2 Gambling image by History Channel from http://www.history.com/news/ask-history/where-did-poker-originate

3 Spam image by Qwertyxp2000 from https://commons.wikimedia.org/wiki/File:Spam_can.png

4 stock market image by James Smith from https://pixabay.com/en/business-stock-finance-market-1730089/

Algorithms

Under appropriate conditions $\mathcal{R}_T(\boldsymbol{u}) = O(\sqrt{T})$

Multiple algorithms:

1. Online Gradient Descent.
2. Mirror Descent.
3. Exponential Weights.

Problem

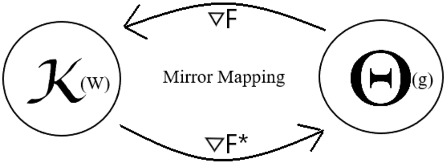
- Usually Gradient Descent and Exponential Weights are seen as special cases of Mirror Descent.
- Koolen (2016) found that Gradient Descent can be seen as a special case of Exponential Weights.
- Some interesting implications, but since Gradient Descent is a special case of Mirror Descent this also raises the following question:

Is Mirror Descent a special case of Exponential Weights?

Mirror Descent

Choose suitable Legendre function F^*. Initialize $\boldsymbol{w}_1 = \arg\min_w F^*(w)$, then update with:

$$\boldsymbol{w}_{t+1} = (\nabla F^*)^{-1}\Big(\nabla F^*(\boldsymbol{w}_t) - \eta \boldsymbol{g}_t\Big)$$



When $F^*(\boldsymbol{w}_t) = \frac{1}{2}||\boldsymbol{w}_t||_2^2$ we obtain Gradient Descent:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t,$$

Prediction with Expert Advice

- A special case of the Online Linear Optimization setting: The weight vector $\boldsymbol{w}_t$ as a probability distribution p_t on experts $k = 1, \ldots, K$.
- We choose a specific loss function: $\mathbb{E}_{k \sim p_t}[\langle \boldsymbol{e}_k, \boldsymbol{g}_t \rangle]$, where $\boldsymbol{e}_k$ is the basis vector in direction k and g_t^k is the loss of expert k at time t.
- Same goal as in Online Linear Optimization, find an algorithm that has regret that grows sub-linear with T.

Exponential Weights

Initialize p_t with *prior distribution* π, then update with:

$$p_{t+1}(k) = \frac{\pi(k)\ \exp(-\eta \sum_{i=1}^{t} g_i^k)}{\sum_{k=1}^{K} \pi(k)\ \exp(-\eta \sum_{i=1}^{t} g_i^k)},$$

where η is the learning rate. Usually, π is chosen as the uniform distribution over the experts.

Different interpretation

- A non-standard interpretation of Exponential Weights arises if we use a non-uniform prior over a continuous set of experts parametrized by $\boldsymbol{z} \in \mathcal{K}$.
- We use the **mean** of p_{t+1} as weights $\boldsymbol{w}_{t+1}$.
- With a multivariate normal distribution as a prior the mean of Exponential Weights is the Gradient Descent algorithm.

Setup

In the Online Linear Optimization setting, in each round t expert $\boldsymbol{z}$ receives loss $\langle \boldsymbol{z}, \boldsymbol{g}_t \rangle$.

Our loss becomes:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{z} \sim p_t}[\langle \boldsymbol{z}, \boldsymbol{g}_t \rangle] &= \langle \mathbb{E}_{\boldsymbol{z} \sim p_t}[\boldsymbol{z}], \boldsymbol{g}_t \rangle \\ &= \langle \boldsymbol{w}_t, \boldsymbol{g}_t \rangle. \end{aligned}$$

We update π with:

$$p_{t+1}(\boldsymbol{z}) = \frac{\pi(\boldsymbol{z}) \exp(-\eta \sum_{i=1}^{t} \langle \boldsymbol{z}, \boldsymbol{g}_i \rangle)}{\int_{\mathcal{K}} \pi(\boldsymbol{z}) \exp(-\eta \sum_{i=1}^{t} \langle \boldsymbol{z}, \boldsymbol{g}_i \rangle) d\boldsymbol{z}}.$$

Prior from an exponential family

Many distributions such as the normal, poisson, exponential, gamma, multinomial and many more can be written in the exponential family form:

$$p(\boldsymbol{z}) = e^{\langle \boldsymbol{\theta}, T(\boldsymbol{z}) \rangle - F(\boldsymbol{\theta})} K(\boldsymbol{z}),$$

where $\boldsymbol{\theta}$ is the natural parameter, $T(\boldsymbol{z})$ is the sufficient statistic, $F(\boldsymbol{\theta})$ is the cumulant generating function, and $K(\boldsymbol{z})$ is the carrier measure.

Mean: $\mathbb{E}_p[\boldsymbol{z}] = \nabla F(\boldsymbol{\theta})$

Main Result

Theorem

Let p_{t+1} be the Exponential Weights distribution at time $t+1$ with a prior from an exponential family. Let Mirror Descent be used with F^*, the convex conjugate of cumulant generating function F. Then the Mirror Descent algorithm is the mean of the Exponential Weights algorithm:

$$\mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{t+1}}[\boldsymbol{z}] = \boldsymbol{w}_{t+1} = \nabla F\Big(\nabla F^*(\boldsymbol{w}_t) - \eta \boldsymbol{g}_t\Big).$$

Example

With a standard multivariate normal prior the Exponential Weights distribution at time $t+1$ is: $p_{t+1}(z) = N(z|\boldsymbol{w}_{t+1}, I)$. The cumulant generating function is:

$$F(\sum_{i=1}^{t} \boldsymbol{g}_t) = \frac{1}{2}||\sum_{i=1}^{t} \boldsymbol{g}_t||_2^2$$

This gives the following mean:

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{t+1}}[\boldsymbol{z}] &= (\nabla F^*)^{-1}\Big(\nabla F^*(\boldsymbol{w}_t) - \eta \boldsymbol{g}_t)\Big) \\ &= \boldsymbol{w}_t - \eta \boldsymbol{g}_t.\end{aligned}$$

Applications of main result

1. Efficient sampling in the Linear Bandit setting
2. Nice theoretical properties of cumulant generating functions (self-concordant barriers)
3. Prior on the learning rate (exploit easy cases!)
4. Scale free algorithms (scaling of the loss becomes irrelevant)

Conclusion

A large class of Online optimization algorithms is like learning distributions.

References I

Koolen, W. (2016). Gradient descent as exponential weights. *Blog February 21: http://blog.wouterkoolen.info/GDasEW/post.html/*.

van der Hoeven, D. (2016). Is mirror descent a special case of exponential weights? *MSC Thesis. Available from: http://pub.math.leidenuniv.nl/~hoevendvander/*.

Interpretation Gradient Descent

Gradient Descent does not learn the variance!

Can we learn the variance? Yes, with the online Newton algorithm:

$$p_{t+1}(z) = N(z|\boldsymbol{w}_{t+1}, \big(\sum_{i=1}^{t} c\boldsymbol{g}_t\boldsymbol{g}_t^T\big)^{-1})$$