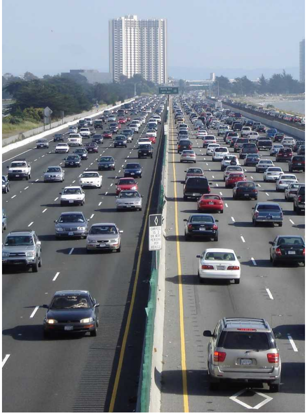
Scale-invariant online learning

Michał Kempka Wojciech Kotłowski

IDSS Seminar, 27.11.2018

Online learning example: travel time estimation



- At every timestamp t, navigation software needs to predict travel time y_t at a given road segment
- Given feature vector $\boldsymbol{x}_t \in \mathbb{R}^d$ representing current traffic conditions, predict $\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t$ with a linear model
- Observe real y_t and measure prediction loss, e.g. $(y_t - \widehat{y}_t)^2$
- Improve model parameters $\boldsymbol{w}_t \to \boldsymbol{w}_{t+1}$

Online learning example: spam filtering

- At every timestamp t, spam filter needs to classify an incoming email as spam/no-spam ($y_t \in \{+1, -1\}$)
- Given feature vector $\boldsymbol{x}_t \in \mathbb{R}^d$ representing email's body, predict $\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t$ with a linear model
- Receive feedback y_t from a user and measure prediction loss, e.g. *logistic loss* $\log(1 + e^{-y_t \widehat{y}_t})$
- Improve model parameters $\boldsymbol{w}_t \rightarrow \boldsymbol{w}_{t+1}$

Online learning with linear models

At each trial $t = 1, \ldots, T$:

  Nature reveals input instance $\boldsymbol{x}_t \in \mathbb{R}^d$

  Learner predicts with a linear model $\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t$, where $\boldsymbol{w}_t \in \mathbb{R}^d$

  Nature reveals label y_t

  Learner suffers loss $\ell(y_t, \widehat{y}_t)$

Online learning with linear models

revealed before prediction!

At each trial $t = 1, \ldots, T$:

Nature reveals input instance $\boldsymbol{x}_t \in \mathbb{R}^d$

Learner predicts with a linear model $\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t$, where $\boldsymbol{w}_t \in \mathbb{R}^d$

Nature reveals label y_t

Learner suffers loss $\ell(y_t, \widehat{y}_t)$

Online learning with linear models

At each trial $t = 1, \ldots, T$:

- Nature reveals input instance $\boldsymbol{x}_t \in \mathbb{R}^d$
- Learner predicts with a linear model $\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t$, where $\boldsymbol{w}_t \in \mathbb{R}^d$
- Nature reveals label y_t
- Learner suffers loss $\ell(y_t, \widehat{y}_t)$, convex and L-Lipschitz in $\widehat{y}_t$

L-Lipschitz = (sub)derivative bounded by L

Loss function	$\ell(y, \widehat{y})$	$\partial_{\widehat{y}} \ell(y, \widehat{y})$	L
logistic	$\log\left(1 + e^{-y\widehat{y}}\right)$	$\frac{-y}{1+e^{y\widehat{y}}}$	1
hinge	$\max\{0, 1 - y\widehat{y}\}$	$-y\mathbf{1}[y\widehat{y} \leq 1]$	1
absolute	$\lvert\widehat{y} - y\rvert$	$\mathrm{sgn}(\widehat{y} - y)$	1

Without loss of generality assume $L = 1$

Online learning with linear models

At each trial $t = 1, \ldots, T$:

Nature reveals input instance $\boldsymbol{x}_t \in \mathbb{R}^d$

Learner predicts with a linear model $\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t$, where $\boldsymbol{w}_t \in \mathbb{R}^d$

Nature reveals label y_t

Learner suffers loss $\ell(y_t, \widehat{y}_t)$, convex and L-Lipschitz in $\widehat{y}_t$

No stochastic assumptions on the data sequence $(\boldsymbol{x}_t, y_t)$ are made

Minimize regret relative to oracle weight vector $\boldsymbol{w}^\star \in \mathbb{R}^d$:

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;=\; \sum_{t=1}^{T} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t) - \sum_{t=1}^{T} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}^\star),$$

Goal: sublinear regret for any $\boldsymbol{w}^\star$ and any data sequence $(\boldsymbol{x}_t, y_t)$

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

Make a small step along negative gradient of the loss

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\le\; \frac{\|\boldsymbol{w}^\star\|^2}{2\eta} \;+\; \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla_t\|^2 \qquad \text{(starting at } \boldsymbol{w}_1 = \boldsymbol{0}\text{)}$$

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \ \leq \ \frac{\|\boldsymbol{w}^\star\|^2}{2\eta} \ + \ \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla_t\|^2 \qquad \text{(starting at } \boldsymbol{w}_1 = \boldsymbol{0}\text{)}$$

Optimal in-hindsight tuning $\eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$
to minimize the regret (impossible in practice)

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\le\; \|\boldsymbol{w}^\star\| \sqrt{\sum_t \|\nabla_t\|^2} \qquad \text{for } \eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$$

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\le\; \|\boldsymbol{w}^\star\| \sqrt{\sum_t \|\nabla_t\|^2} \qquad \text{for } \eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$$

- Separate fixed learning rate per feature

$$w_{t+1,i} = w_{t,i} - \eta_i \nabla_{t,i},$$

Each feature has its own learning rate

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\le\; \|\boldsymbol{w}^\star\| \sqrt{\sum_t \|\nabla_t\|^2} \qquad \text{for } \eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$$

- Separate fixed learning rate per feature

$$w_{t+1,i} = w_{t,i} - \eta_i \nabla_{t,i},$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\le\; \sum_{i=1}^{d} \left(\frac{w_i^{\star 2}}{2\eta_i} + \frac{\eta_i}{2} \sum_{t=1}^{T} \nabla_{t,i}^2 \right) \qquad (\boldsymbol{w}_1 = 0)$$

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \; \le \; \|\boldsymbol{w}^\star\| \sqrt{\sum_t \|\nabla_t\|^2} \qquad \text{for } \eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$$

- Separate fixed learning rate per feature

$$w_{t+1,i} = w_{t,i} - \eta_i \nabla_{t,i},$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \; \le \; \sum_{i=1}^{d} \left(\frac{w_i^{\star 2}}{2\eta_i} + \frac{\eta_i}{2} \sum_{t=1}^{T} \nabla_{t,i}^2 \right) \qquad (\boldsymbol{w}_1 = 0)$$

Optimal in-hindsight tuning $\eta_i^\star = \frac{|w_i^\star|}{\sqrt{\sum_t \nabla_{t,i}^2}}$

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\text{regret}_T(\boldsymbol{w}^\star) \;\leq\; \|\boldsymbol{w}^\star\| \sqrt{\sum_t \|\nabla_t\|^2} \qquad \text{for } \eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$$

- Separate fixed learning rate per feature

$$w_{t+1,i} = w_{t,i} - \eta_i \nabla_{t,i},$$

$$\text{regret}_T(\boldsymbol{w}^\star) \;\leq\; \sum_{i=1}^{d} \left(|w_i^\star| \sqrt{\sum_t \nabla_{t,i}^2} \right) \qquad \text{for } \eta_i^\star = \frac{|w_i^\star|}{\sqrt{\sum_t \nabla_{t,i}^2}}$$

Better than the previous bound
(single tuning per feature)

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+} \quad \boldsymbol{w}_t)$$

$$\mathrm{regret}_T(\quad \frac{\|\boldsymbol{w}^\star\|}{\sum_t \|\nabla_t\|^2}$$

Can we get the optimal SGD regret bound:

$$\sum_{i=1}^{d} \left(|w_i^\star| \sqrt{\sum_t \nabla_{t,i}^2} \right)$$

with some adaptive tuning strategy?

- Separate fixe

$$w_{t+1,i} = w_{t,i} - \eta_i \nabla_{t,i},$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\leq\; \sum_{i=1}^{d} \left(|w_i^\star| \sqrt{\sum_t \nabla_{t,i}^2} \right) \qquad \text{for } \eta_i^\star = \frac{|w_i^\star|}{\sqrt{\sum_t \nabla_{t,i}^2}}$$

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\text{regret}_T(\boldsymbol{w}^\star) \;\le\; \|\boldsymbol{w}^\star\| \sqrt{\sum_t \|\nabla_t\|^2} \qquad \text{for } \eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$$

- Separate fixed learning rate per feature

$$w_{t+1,i} = w_{t,i} - \eta_i \nabla_{t,i},$$

$$\text{regret}_T(\boldsymbol{w}^\star) \;\le\; \sum_{i=1}^{d} \left(|w_i^\star| \sqrt{\sum_t \nabla_{t,i}^2} \right) \qquad \text{for } \eta_i^\star = \frac{|w_i^\star|}{\sqrt{\sum_t \nabla_{t,i}^2}}$$

- Adaptive learning rate per feature (AdaGrad [Duchi et al., 2011])

$$w_{t+1,i} = w_{t,i} - \eta_{i,t} \nabla_{t,i}, \qquad \text{where } \eta_{i,t} = \frac{\eta_i}{\sqrt{\epsilon + \sum_{j \le t} \nabla_{j,i}^2}}$$

Tuning the learning rate mimics the optimal tuning

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\text{regret}_T(\boldsymbol{w}^\star) \ \leq \ \|\boldsymbol{w}^\star\| \sqrt{\sum_t \|\nabla_t\|^2} \qquad \text{for } \eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$$

- Separate fixed learning rate per feature

$$w_{t+1,i} = w_{t,i} - \eta_i \nabla_{t,i},$$

$$\text{regret}_T(\boldsymbol{w}^\star) \ \leq \ \sum_{i=1}^{d} \left(|w_i^\star| \sqrt{\sum_t \nabla_{t,i}^2} \right) \qquad \text{for } \eta_i^\star = \frac{|w_i^\star|}{\sqrt{\sum_t \nabla_{t,i}^2}}$$

- Adaptive learning rate per feature (AdaGrad [Duchi et al., 2011])

$$w_{t+1,i} = w_{t,i} - \eta_{i,t} \nabla_{t,i}, \qquad \text{where } \eta_{i,t} = \frac{\eta_i}{\sqrt{\epsilon + \sum_{j \leq t} \nabla_{j,i}^2}}$$

$$\text{regret}_T(\boldsymbol{w}^\star) \ \leq \ \sum_{i=1}^{d} \left(\frac{\max_t |w_i^\star - w_{i,t}|^2}{2\eta_i} + \eta_i \right) \sqrt{\epsilon + \sum_t \nabla_{t,i}^2}$$

Stochastic Gradient Descent (SGD)

- Fixed learning rate

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_t, \qquad \text{where } \nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t)$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\le\; \|\boldsymbol{w}^\star\| \sqrt{\sum_t \|\nabla_t\|^2} \qquad \text{for } \eta^\star = \frac{\|\boldsymbol{w}^\star\|}{\sqrt{\sum_t \|\nabla_t\|^2}}$$

- Separate fixed learning rate per feature

$$w_{t+1,i} = w_{t,i} - \eta_i \nabla_{t,i},$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\le\; \sum_{i=1}^{d} \Big(|w_i^\star|$$

Not there yet: still requires to tune η_i depending on unknown $\boldsymbol{w}^\star$!

- Adaptive learning rate per feature (AdaGrad [Duchi et al., 2011])

$$w_{t+1,i} = w_{t,i} - \eta_{i,t} \nabla_{t,i}, \qquad \text{where } \eta_{i,t} = \frac{\eta_i}{\sqrt{\epsilon + \sum_{j \le t} \nabla_{j,i}^2}}$$

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;\le\; \sum_{i=1}^{d} \Big(\frac{\max_t |w_i^\star - w_{i,t}|^2}{2\eta_i} + \eta_i \Big) \sqrt{\epsilon + \sum_t \nabla_{t,i}^2}$$

Feature scales

$$w_{t+1,i} \;=\; w_{t,i} - \eta_i \nabla_{t,i}$$

Feature scales

$$w_{t+1,i} \; = \; w_{t,i} - \eta_i \nabla_{t,i}$$

By the chain rule $\nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t) = \underbrace{\frac{\partial \ell(y_t, \widehat{y})}{\partial \widehat{y}}}_{g_t} \Big|_{\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t} \boldsymbol{x}_t$:

$$w_{t+1,i} \; = \; w_{t,i} \; - \; \eta_i g_t x_{t,i}$$

For example, for squared-error loss:
$\nabla_{\boldsymbol{w}_t} (y_t - \boldsymbol{w}_t^\top \boldsymbol{x}_t)^2 = \underbrace{2(y_t - \boldsymbol{w}_t^\top \boldsymbol{x}_t)}_{g_t} \boldsymbol{x}_t$

Feature scales

$$w_{t+1,i} \;=\; w_{t,i} - \eta_i \nabla_{t,i}$$

By the chain rule $\nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t) = \underbrace{\frac{\partial \ell(y_t, \widehat{y})}{\partial \widehat{y}}}_{g_t} \Big|_{\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t} \boldsymbol{x}_t$:

$$w_{t+1,i} \;=\; w_{t,i} \;-\; \eta_i g_t x_{t,i}$$

Suppose feature i has a physical unit $[X_i]$, while the label and prediction are dimensionless (like in, e.g., classification)
$\Longrightarrow$ i-th weight coordinate w_i must have unit $1/[X_i]$

Feature scales

$$w_{t+1,i} \; = \; w_{t,i} - \eta_i \nabla_{t,i}$$

By the chain rule $\nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t) = \underbrace{\frac{\partial \ell(y_t, \widehat{y})}{\partial \widehat{y}}}_{g_t} \Big|_{\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t} \boldsymbol{x}_t$:

$$\overset{1/[X_i]}{w_{t+1,i}} \; = \; \overset{1/[X_i]}{w_{t,i}} \; \underset{\text{units do not match!}}{-} \; \eta_i \underset{\text{dimensionless}}{g_t} \overset{[X_i]}{x_{t,i}}$$

Suppose feature i has a physical unit $[X_i]$, while the label and prediction are dimensionless (like in, e.g., classification)

$\Longrightarrow$ i-th weight coordinate w_i must have unit $1/[X_i]$

Feature scales

$$w_{t+1,i} \;=\; w_{t,i} - \eta_i \nabla_{t,i}$$

By the chain rule $\nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t) = \underbrace{\frac{\partial \ell(y_t, \widehat{y})}{\partial \widehat{y}}}_{g_t} \Big|_{\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t} \boldsymbol{x}_t$:

$1/[X_i]$ $\quad$ $1/[X_i]$ $\quad$ $[X_i]$

$$w_{t+1,i} \;=\; w_{t,i} \;-\; \eta_i g_t x_{t,i}$$

. . . unless $[\eta_i] = 1/[X_i]^2$ $\quad$ dimensionless

Suppose feature i has a physical unit $[X_i]$, while the label and prediction are dimensionless (like in, e.g., classification)
$\implies$ i-th weight coordinate w_i must have unit $1/[X_i]$

Feature scales

$$w_{t+1,i} \; = \; w_{t,i} - \eta_i \nabla_{t,i}$$

By the chain rule $\nabla_t = \nabla_{\boldsymbol{w}_t} \ell(y_t, \boldsymbol{x}_t^\top \boldsymbol{w}_t) = \underbrace{\frac{\partial \ell(y_t, \widehat{y})}{\partial \widehat{y}}}_{g_t} \Big|_{\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_t} \boldsymbol{x}_t$:

$1/[X_i]$ $\quad$ $1/[X_i]$ $\quad$ $[X_i]$

$$w_{t+1,i} \; = \; w_{t,i} \; - \; \eta_i g_t x_{t,i}$$

…unless $[\eta_i] = 1/[X_i]^2$ $\quad$ dimensionless

Suppose feature i has a physical unit $[X_i]$, while the label and prediction are dimensionless (like in, e.g., classification)
$\implies$ i-th weight coordinate w_i must have unit $1/[X_i]$

Learning rate should compensate units on each coordinate! (in fact, the optimal in-hindsight tuning $\eta_i = \frac{|w_i^\star|}{\sqrt{\sum_t \nabla_{t,i}^2}}$ achieves exactly that)

Single learning rate is unable to compensate units.

Feature scales

AdaGrad [Duchi et al., 2011]:

$$w_{t+1,i} \;=\; w_{t,i} \;-\; \frac{\eta}{\sqrt{\epsilon + \sum_{j \le t} \nabla_{j,i}^2}} \nabla_{t,i}$$

Feature scales

AdaGrad [Duchi et al., 2011]

$$\overset{1/[X_i]}{w_{t+1,i}} = \overset{1/[X_i]}{w_{t,i}} - \frac{\overset{1/[X_i]\ ?}{\eta}}{\sqrt{\epsilon + \sum_{j \le t} \underset{[X_i]}{\nabla_{j,i}^2}}} \overset{[X_i]}{\nabla_{t,i}}$$

Learning rate still needs to compensate units, but cannot do so for all coordinates at the same time

Feature scales

AdaGrad [Duch , 201]

$$\overset{1/[X_i]}{w_{t+1,i}} = \overset{1/[X_i]}{w_{t,i}} - \frac{\overset{1/[X_i]\ ?}{\eta}}{\underset{[X_i]}{\sqrt{\epsilon + \sum_{j\leq t} \nabla_{j,i}^2}}} \overset{[X_i]}{\nabla_{t,i}}$$

Learning rate still needs to compensate units, but cannot do so for all coordinates at the same time

- Also applies to RMSprop [Tieleman and Hinton, 2012] and Adam [Kingma and Ba, 2014]
- Heuristically solved by Adadelta [Zeiler, 2012]

Feature scales

AdaGrad [Duch , 2011]

$$\overset{1/[X_i]}{w_{t+1,i}} = \overset{1/[X_i]}{w_{t,i}} - \frac{\overset{1/[X_i]\ ?}{\eta}}{\sqrt{\epsilon + \sum_{j\le t} \underset{[X_i]}{\nabla^2_{j,i}}}} \overset{[X_i]}{\nabla_{t,i}}$$

Learning rate still needs to compensate units, but cannot do so for all coordinates at the same time

- Also applies to RMSprop [Tieleman and Hinton, 2012] and Adam [Kingma and Ba, 2014]
- Heuristically solved by Adadelta [Zeiler, 2012]

Motivation: fully adaptive algorithms need to resolve this scaling issue

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall i, t \quad x_{t,i} \mapsto a_i x_{t,i} \quad w_i \mapsto a_i^{-1} w_i \qquad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

In particular: if $\boldsymbol{w}^\star$ is optimal (loss-minimizer) for sequence $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, then $\boldsymbol{A}^{-1}\boldsymbol{w}^\star$ is optimal for sequence $\{(\boldsymbol{A}\boldsymbol{x}_t, y_t)\}_{t=1}^T$

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

In particular: if $\boldsymbol{w}^\star$ is optimal (loss-minimizer) for sequence $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, then $\boldsymbol{A}^{-1}\boldsymbol{w}^\star$ is optimal for sequence $\{(\boldsymbol{A}\boldsymbol{x}_t, y_t)\}_{t=1}^T$

Example: minimizing squared error loss:

$$\boldsymbol{w}^\star = \Big(\sum_t \boldsymbol{x}_t \boldsymbol{x}_t^\top\Big)^{-1} \sum_t \boldsymbol{x}_t y_t$$

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

In particular: if $\boldsymbol{w}^\star$ is optimal (loss-minimizer) for sequence $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, then $\boldsymbol{A}^{-1}\boldsymbol{w}^\star$ is optimal for sequence $\{(\boldsymbol{A}\boldsymbol{x}_t, y_t)\}_{t=1}^T$

Example: minimizing squared error loss:

$$\boldsymbol{w}^\star \quad \mapsto \quad \Big(\sum_t \boldsymbol{A}\boldsymbol{x}_t(\boldsymbol{A}\boldsymbol{x}_t)^\top\Big)^{-1} \sum_t \boldsymbol{A}\boldsymbol{x}_t y_t$$

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

In particular: if $\boldsymbol{w}^\star$ is optimal (loss-minimizer) for sequence $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, then $\boldsymbol{A}^{-1}\boldsymbol{w}^\star$ is optimal for sequence $\{(\boldsymbol{A}\boldsymbol{x}_t, y_t)\}_{t=1}^T$

Example: minimizing squared error loss:

$$\boldsymbol{w}^\star \quad \mapsto \quad \boldsymbol{A}^{-1}\Big(\sum_t \boldsymbol{x}_t \boldsymbol{x}_t^\top\Big)^{-1} \boldsymbol{A}^{-1}\boldsymbol{A}\sum_t \boldsymbol{x}_t y_t$$

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

In particular: if $\boldsymbol{w}^\star$ is optimal (loss-minimizer) for sequence $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, then $\boldsymbol{A}^{-1}\boldsymbol{w}^\star$ is optimal for sequence $\{(\boldsymbol{A}\boldsymbol{x}_t, y_t)\}_{t=1}^T$

Example: minimizing squared error loss:

$$\boldsymbol{w}^\star \quad \mapsto \quad \boldsymbol{A}^{-1}\Big(\sum_t \boldsymbol{x}_t \boldsymbol{x}_t^\top\Big)^{-1} \sum_t \boldsymbol{x}_t y_t = \boldsymbol{A}^{-1}\boldsymbol{w}^\star$$

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

In particular: if $\boldsymbol{w}^\star$ is optimal (loss-minimizer) for sequence $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, then $\boldsymbol{A}^{-1}\boldsymbol{w}^\star$ is optimal for sequence $\{(\boldsymbol{A}\boldsymbol{x}_t, y_t)\}_{t=1}^T$

A learning algorithm is scale-invariant if it returns the same predictions under arbitrary rescaling of the data:

$$\forall t \;\; \boldsymbol{x}_t \mapsto \boldsymbol{A}\boldsymbol{x}_t \quad \Longrightarrow \quad \boldsymbol{w}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{w}_t \quad \Longrightarrow \quad \boldsymbol{w}_t^\top \boldsymbol{x}_t \mapsto \boldsymbol{w}_t^\top \boldsymbol{x}_t$$

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

In particular: if $\boldsymbol{w}^\star$ is optimal (loss-minimizer) for sequence $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, then $\boldsymbol{A}^{-1}\boldsymbol{w}^\star$ is optimal for sequence $\{(\boldsymbol{A}\boldsymbol{x}_t, y_t)\}_{t=1}^T$

A learning algorithm is scale-invariant if it returns under arbitrary rescaling of the data:

$$\forall t \;\; \boldsymbol{x}_t \mapsto \boldsymbol{A}\boldsymbol{x}_t \quad \Longrightarrow \quad \boldsymbol{w}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{w}_t \quad \Longrightarrow \quad \boldsymbol{w}_t^\top \boldsymbol{x}_t \mapsto \boldsymbol{w}_t^\top \boldsymbol{x}_t$$

no initial data normalization required!

Scale invariance

A natural symmetry in the linear problems

Rescaling the features followed by the inverse scaling of the weights keep the predictions (and hence losses) invariant:

$$\forall t \quad \boldsymbol{x}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}_t, \quad \boldsymbol{w} \mapsto \boldsymbol{A}\boldsymbol{w} \quad \Longrightarrow \quad \boldsymbol{x}_t^\top \boldsymbol{w} \mapsto \boldsymbol{x}_t^\top \boldsymbol{w}$$

for any diagonal matrix $\boldsymbol{A}$

In particular: if $\boldsymbol{w}^\star$ is optimal (loss-minimizer) for sequence $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, then $\boldsymbol{A}^{-1}\boldsymbol{w}^\star$ is optimal for sequence $\{(\boldsymbol{A}\boldsymbol{x}_t, y_t)\}_{t=1}^T$

A learning algorithm is scale-invariant if it returns under arbitrary rescaling of the data:

$$\forall t \;\; \boldsymbol{x}_t \mapsto \boldsymbol{A}\boldsymbol{x}_t \quad \Longrightarrow \quad \boldsymbol{w}_t \mapsto \boldsymbol{A}^{-1}\boldsymbol{w}_t \quad \Longrightarrow \quad \boldsymbol{w}_t^\top \boldsymbol{x}_t \mapsto \boldsymbol{w}_t^\top \boldsymbol{x}_t$$

no initial data normalization required!

Motivation: A fully adaptive algorithm needs to be scale-invariant

Past work

Scale-invariant algorithms with bounded predictions
[Ross et al., 2013, Orabona et al., 2015]

Assumption: $|x_{t,i}w_i^\star| \leq C$ for all i, t for some constant C

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \ = \ O\left(d\sqrt{C^2 T}\right)$$

Past work

Scale-invariant algorithms with bounded predictions
[Ross et al., 2013, Orabona et al., 2015]

Assumption: $|x_{t,i} w_i^\star| \leq C$ for all i, t for some constant C

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;=\; O\left(d\sqrt{C^2 T}\right)$$

Compare with optimal SGD regret:

$$\sum_{i=1}^{d} \left(|w_i^\star| \sqrt{\sum_t \nabla_{t,i}^2} \right)$$

Past worl

$$C^2T \implies \sum_t (\nabla_{t,i} w^\star_{t,i})^2$$

$$d \implies \sum_i$$

Scale-invariant algorithms with bounded pr
[Ross et al., 2013, Orabona et al., 2015]

Assumption: $|x_{t,i} w_i^\star| \le C$ for all i, t for some constant C

$$\mathrm{regret}_T(\boldsymbol{w}^\star) = O\left(d\sqrt{C^2 T}\right)$$

Compare with optimal SGD regret:

$$\sum_{i=1}^{d}\left(|w_i^\star|\sqrt{\sum_t \nabla_{t,i}^2}\right)$$

Past work

Scale-invariant algorithms with bounded predictions [Ross et al., 2013, Orabona et al., 2015]

Assumption: $|x_{t,i} w_i^\star| \leq C$ for all i, t for some constant C

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \; = \; O\left(d\sqrt{C^2 T}\right)$$

[Luo et al., 2016] considers a more general version of scale invariance, but also with bounded predictions

Past work

Scale-invariant algorithms with bounded predictions
[Ross et al., 2013, Orabona et al., 2015]

Assumption: $|x_{t,i} w_i^\star| \leq C$ for all i, t for some constant C

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;=\; O\left(d\sqrt{C^2 T}\right)$$

[Luo et al., 2016] considers a more general version of scale invariance, but also with bounded predictions

Some more recent work on unconstrained online learning:
[McMahan and Streeter, 2010, McMahan and Abernethy, 2013, Orabona, 2013, Cutkosky and Boahen, 2017, Cutkosky and Orabona, 2018]

Past work

Scale-invariant algorithms with bounded predictions [Ross et al., 2013, Orabona et al., 2015]

Assumption: $|x_{t,i} w_i^\star| \leq C$ for all i, t for some constant C

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \ = \ O\left(d\sqrt{C^2 T}\right)$$

[Luo et al., 2016] considers a more general version of scale invariance, but also with bounded predictions

Some more recent work on unconstrained online learning: [McMahan and Streeter, 2010, McMahan and Abernethy, 2013, Orabona, 2013, Cutkosky and Boahen, 2017, Cutkosky and Orabona, 2018]

Prior to this work: [Kotłowski, 2017]

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL_1

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_1$

Parameter: $\epsilon = 1$

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_1$

Parameter: $\epsilon = 1$

Keep track of data statistics:

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S^2_{t,i} = \sum_{j \le t} \nabla^2_{j,i}, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

Maximum value at a given feature

Sum of squared gradients

Sum of gradients

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_1$

Parameter: $\epsilon = 1$

Keep track of data statistics:

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S_{t,i}^2 = \sum_{j \le t} \nabla_{j,i}^2, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

and an auxilary variable $\beta_{t,i} = \min\{\beta_{t-1,i}, \frac{\epsilon(S_{t-1,i}^2 + M_{t,i}^2)}{x_{t,i}^2 t}\}$ with $\beta_{0,i} = \epsilon$

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_1$

Parameter: $\epsilon = 1$

Keep track of data statistics:

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S_{t,i}^2 = \sum_{j \le t} \nabla_{j,i}^2, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

and an auxilary variable $\beta_{t,i} = \min\{\beta_{t-1,i}, \frac{\epsilon(S_{t-1,i}^2 + M_{t,i}^2)}{x_{t,i}^2 t}\}$ with $\beta_{0,i} = \epsilon$

$$w_{t,i} = \beta_{t,i} \frac{\operatorname{sgn}(\theta_i)}{2\sqrt{S_{t-1,i}^2 + M_{t,i}^2}} \left(e^{|\theta_i|/2} - 1\right), \quad \text{where} \quad \theta_i = \frac{G_{t,i}}{\sqrt{S_{t-1,i}^2 + M_{t,i}^2}}$$

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_1$

Parameter: $\epsilon = 1$

Keep track of data statistics:

$1/[X_i]$

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S_{t,i}^2 = \sum_{j \le t} \nabla_{j,i}^2, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

and an auxilary variable $\beta_{t,i} = \min\{\beta_{t-1,i}, \frac{\epsilon(S_{t-1,i}^2 + M_{t,i}^2)}{x_{t,i}^2 t}\}$ with $\beta_{0,i} = \epsilon$

unitless

$[X_i]$

$$w_{t,i} = \beta_{t,i} \frac{\operatorname{sgn}(\theta_i)}{2\sqrt{S_{t-1,i}^2 + M_{t,i}^2}} \left(e^{|\theta_i|/2} - 1\right), \quad \text{where} \quad \theta_i = \frac{G_{t,i}}{\sqrt{S_{t-1,i}^2 + M_{t,i}^2}}$$

unitless

$\sqrt{[X_i]^2}$

$\sqrt{[X_i]^2}$

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_1$

Parameter: $\epsilon = 1$

Keep track of data statistics:

$$M_{t,i} = \max_{j \leq t} |x_{j,i}|, \quad S^2_{t,i} = \sum_{j \leq t} \nabla^2_{j,i}, \quad G_{t,i} = \sum_{j \leq t} \nabla_{j,i}$$

and an auxilary variable $\beta_{t,i} = \min\{\beta_{t-1,i}, \frac{\epsilon(S^2_{t-1,i}+M^2_{t,i})}{x^2_{t,i} t}\}$ with $\beta_{0,i} = \epsilon$

$$w_{t,i} = \beta_{t,i} \frac{\operatorname{sgn}(\theta_i)}{2\sqrt{S^2_{t-1,i} + M^2_{t,i}}} \left(e^{|\theta_i|/2} - 1\right), \quad \text{where} \quad \theta_i = \frac{G_{t,i}}{\sqrt{S^2_{t-1,i} + M^2_{t,i}}}$$

$$\operatorname{regret}_T(\boldsymbol{w}^\star) \;=\; \sum_{i=1}^{d} \tilde{O}\left(|w^\star_i| \sqrt{\max_t x^2_{t,i} + \sum_t \nabla^2_{t,i}}\right),$$

where $\tilde{O}(\cdot)$ hides logarithmic factors

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_1$

Parameter: $\epsilon = 1$

Keep track of data statistics:

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S^2_{t,i} = \sum_{j \le t} \nabla^2_{j,i}, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

and an auxilary variable $\beta_{t,i} = \min\{\beta_{t-1,i}, \frac{\epsilon(S^2_{t-1,i}+M^2_{t,i})}{x^2_{t,i}t}\}$ with $\beta_{0,i} = \epsilon$

$$w_{t,i} = \beta_{t,i}\frac{\mathrm{sgn}(}{2\sqrt{S^2_{t-1,i}+M^2_{t,i}}}\Big(e \quad 1\Big), \quad \text{where} \quad \theta_i = \frac{G_{t,i}}{\sqrt{S^2_{t-1,i}+M^2_{t,i}}}$$

Optimal up to logarithmic terms

$$\mathrm{regret}_T(\boldsymbol{w}^\star) \;=\; \sum_{i=1}^{d} \tilde{O}\Big(|w^\star_i|\sqrt{\max_t x^2_{t,i} + \sum_t \nabla^2_{t,i}}\Big),$$

where $\tilde{O}(\cdot)$ hides logarithmic factors

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_1$

Algorithm 1: ScInOL$_1(\epsilon)$

```
Initialization: $S_i^2, G_i, M_i \leftarrow 0, \beta_i \leftarrow \epsilon \ (i = 1, \ldots, d)$
for $t = 1, \ldots, T$ do
    Receive $\boldsymbol{x}_t \in \mathbb{R}^d$
    for $i = 1, \ldots, d$ do
        $M_i \leftarrow \max\{M_i, |x_{t,i}|\}$
        if $x_{t,i} \neq 0$ then $\beta_i \leftarrow \min\{\beta_i, \epsilon(S_i^2 + M_i^2)/(x_{t,i}^2 t)\}$
        $w_{t,i} = \frac{\beta_i \operatorname{sgn}(\theta_i)}{2\sqrt{S_i^2+M_i^2}}\left(e^{|\theta_i|/2} - 1\right)$,    where $\theta_i = \frac{G_i}{\sqrt{S_i^2+M_i^2}}$
    Predict with $\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_{t,i}$, receive loss $\ell_t(\widehat{y}_t)$ and compute
     $g_t = \partial_{\widehat{y}_t} \ell_t(\widehat{y}_t)$
    for $i = 1, \ldots, d$ do
        $G_i \leftarrow G_i - g_t x_{t,i}$
        $S_i^2 \leftarrow S_i^2 + (g_t x_{t,i})^2$
```

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 2: ScInOL$_2$

A more aggressive update, but with weaker guarantees

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 2: ScInOL$_2$

A more aggressive update, but with weaker guarantees

Parameter: $\epsilon = 1$

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S_{t,i}^2 = \sum_{j \le t} \nabla_{j,i}^2, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

and a reward variable $\eta_{t,i} = \epsilon - \sum_{j \le t} \nabla_{j,i} w_{j,i}$

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 2: ScInOL$_2$

A more aggressive update, but with weaker guarantees

Parameter: $\epsilon = 1$

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S^2_{t,i} = \sum_{j \le t} \nabla^2_{j,i}, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

and a reward variable $\eta_{t,i} = \epsilon - \sum_{j \le t} \nabla_{j,i} w_{j,i}$

$$w_{t,i} = \eta_{t-1,i} \frac{\operatorname{sgn}(\theta_i) \min\{|\theta_i|, 1\}}{2\sqrt{S^2_{t-1,i} + M^2_{t,i}}}, \quad \text{where} \quad \theta_i = \frac{G_{t,i}}{\sqrt{S^2_{t-1,i} + M^2_{t,i}}}$$

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 2: ScInOL$_2$

A more aggressive update, but with weaker guarantees

Parameter: $\epsilon = 1$

$1/[X_i]$

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S^2_{t,i} = \sum_{j \le t} \nabla^2_{j,i}, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

$[X_i]$

and a reward va unitless $_{t,i} = \epsilon - \sum_{j \le t} \nabla_{j,i} w_{j,i}$

$$w_{t,i} = \eta_{t-1,i} \frac{\operatorname{sgn}(\theta_i) \min\{|\theta_i|, 1\}}{2\sqrt{S^2_{t-1,i} + M^2_{t,i}}}, \quad \text{where} \quad \theta_i = \frac{G_{t,i}}{\sqrt{S^2_{t-1,i} + M^2_{t,i}}}$$

$\sqrt{[X_i]^2}$

unitless

$\sqrt{[X_i]^2}$

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 2: ScInOL$_2$

A more aggressive update, but with weaker guarantees

Parameter: $\epsilon = 1$

$$M_{t,i} = \max_{j \le t} |x_{j,i}|, \quad S^2_{t,i} = \sum_{j \le t} \nabla^2_{j,i}, \quad G_{t,i} = \sum_{j \le t} \nabla_{j,i}$$

and a reward variable $\eta_{t,i} = \epsilon - \sum_{j \le t} \nabla_{j,i} w_{j,i}$

$$w_{t,i} = \eta_{t-1,i} \frac{\operatorname{sgn}(\theta_i) \min\{|\theta_i|, 1\}}{2\sqrt{S^2_{t-1,i} + M^2_{t,i}}}, \quad \text{where} \;\; \theta_i = \frac{G_{t,i}}{\sqrt{S^2_{t-1,i} + M^2_{t,i}}}$$

$$\operatorname{regret}_T(\boldsymbol{w}^\star) \;=\; \sum_{i=1}^{d} \tilde{O}\left(|w^\star_i| \sqrt{\max_t x^2_{t,i} + \sum_t \nabla^2_{t,i}} \right),$$

but the coefficients in the logarithmic factors depend on the ratio between the largest and (non-zero) smallest feature values.

Scale-invariant algorithms

Scale Invariant Online Learning, Algorithm 1: ScInOL$_2$

Algorithm 2: ScInOL$_2(\epsilon)$

```
Initialization: $S_i^2, G_i, M_i \leftarrow 0, \eta_i \leftarrow \epsilon \ (i = 1, \ldots, d)$
for $t = 1, \ldots, T$ do
    Receive $\boldsymbol{x}_t \in \mathbb{R}^d$
    for $i = 1, \ldots, d$ do
        $M_i \leftarrow \max\{M_i, |x_{t,i}|\}$
        $w_{t,i} = \frac{\operatorname{sgn}(\theta_i)\min\{|\theta_i|,1\}}{2\sqrt{S_i^2+M_i^2}}\eta_i,$    where $\theta_i = \frac{G_i}{\sqrt{S_i^2+M_i^2}}$
    Predict with $\widehat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_{t,i}$, receive loss $\ell_t(\widehat{y}_t)$ and compute
     $g_t = \partial_{\widehat{y}_t}\ell_t(\widehat{y}_t)$
    for $i = 1, \ldots, d$ do
        $G_i \leftarrow G_i - g_t x_{t,i}$
        $S_i^2 \leftarrow S_i^2 + (g_t x_{t,i})^2$
        $\eta_i \leftarrow \eta_i - g_t x_{t,i} w_{t,i}$
```

Artificial data experiment

Experimental setup:

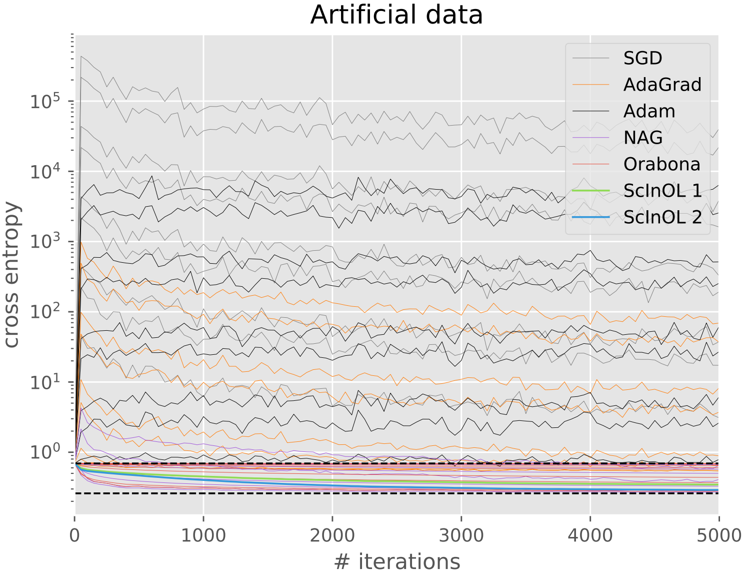
- $\boldsymbol{x} \in \mathbb{R}^{21}$ with $x_i \sim N(0, \sigma_i)$, $\sigma_i \in \{2^{-10}, \ldots, 2^{10}\}$
- $y \sim \mathrm{Bernoulli}(p(\boldsymbol{x}))$, where $p = \mathrm{sigmoid}(\boldsymbol{x}^\top \boldsymbol{w}^\star)$ with $w_i^\star = \pm\frac{1}{\sigma_i}$
- Linear models with cross entropy (logistic) loss
- Algorithms run on a sequence of 5 000 examples and tested on 100K examples (repeated 10 times for stability)

Algorithms:

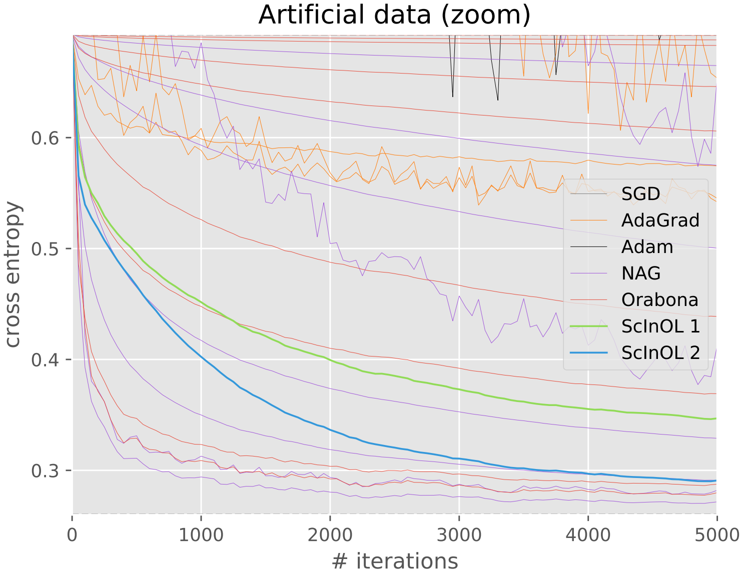
- SGD (with learning rate $\sim 1/\sqrt{t}$), AdaGrad, Adam
- NAG (Normalized Adaptive Gradient) [Ross et al., 2013]
- Scale-free Mirror Descent [Orabona et al., 2015]
- Algorithms from this work

All algorithms (except the last one) have their learning rates set to values from $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$

Artificial data experiment



Artificial data experiment



Experiment - datasets

Name[1]	features	records	classes	scale[2]
Bank	53	41188	2	6.05E+05
Census	381	299285	2	1.81E+06
Covertype	54	581012	7	1.31E+06
Madelon	500	2600	2	1.09E+00
MNIST	728	70000	10	5.83E+03
Shuttle	9	58000	7	7.46E+00

[1] datasets (excluding MNIST) available in the UCI repository

[2] computed as a ratio of highest to lowest positive L_2 norms of features

Experiment - algorithms

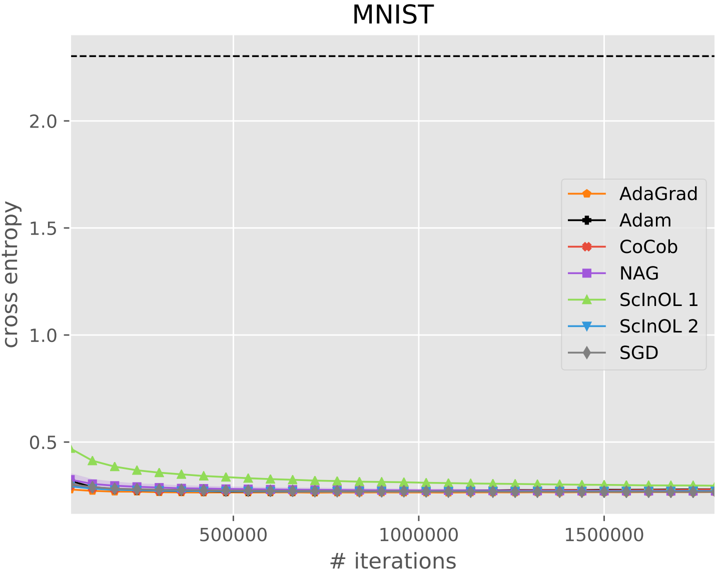
- SGD with decreasing η (as $\sim 1/\sqrt{t}$)
- AdaGrad
- Adam
- NAG
- COCOB [Orabona and Tommasi, 2017]
- ScInOL$_1$
- ScInOL$_2$

All but 3 last algorithms tested with different learning rates: 1.0, 0.1, 0.01, 0.001, 0.0001

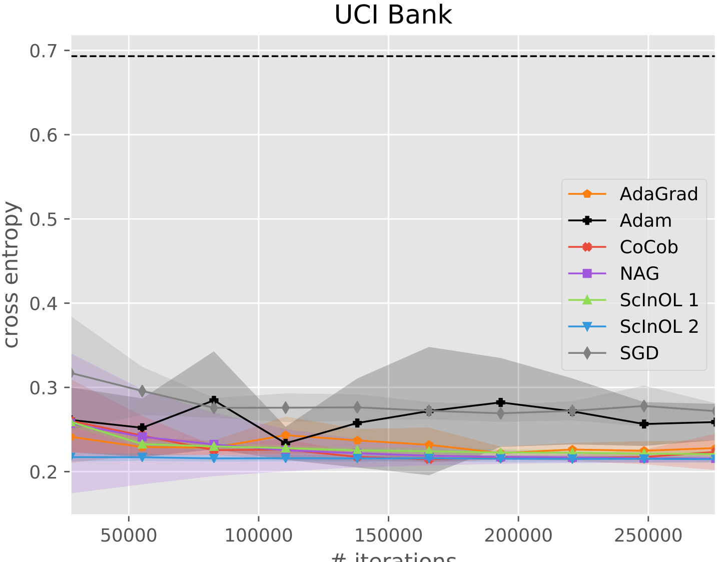
Experiment - setup

- logistic regression initialized with zeros, trained on cross entropy
- minibatch size $= 1$ (online GD)
- test error measured after each training epoch
- each configuration run 10 times (pale strokes of graph lines signify $\pm$ standard deviations)
- for algorithms with varying learning rate configurations, only the best ones are shown

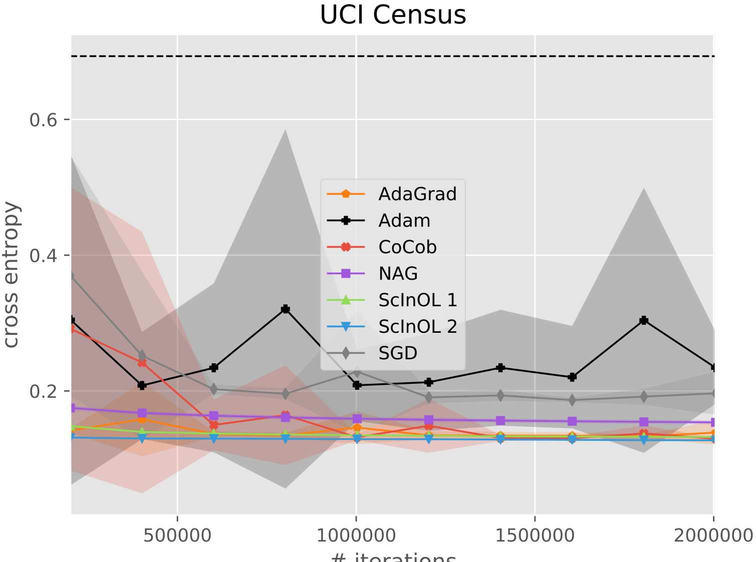
Experiment - results



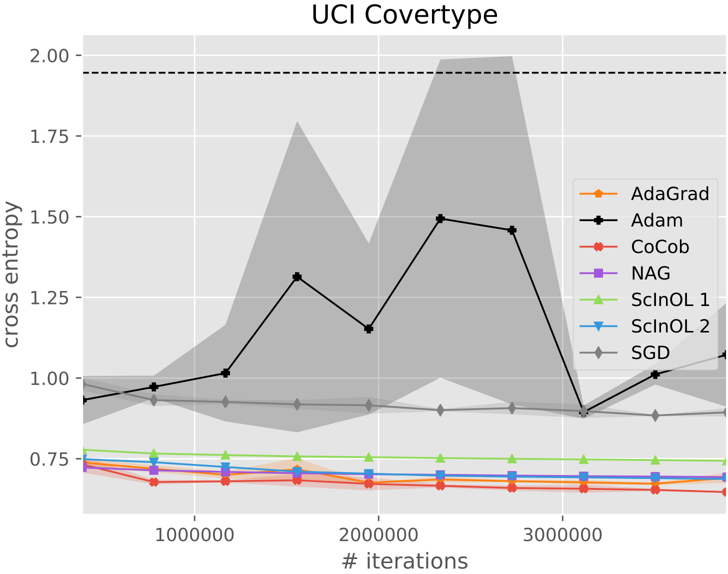
Experiment - results



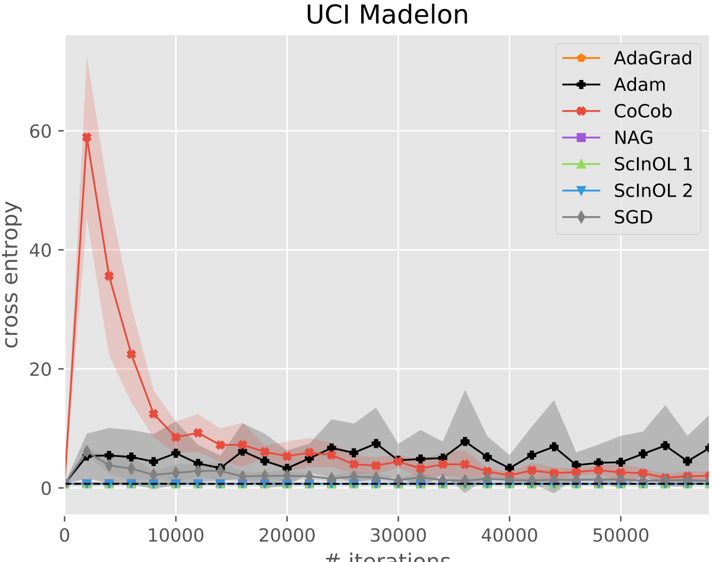
Experiment - results



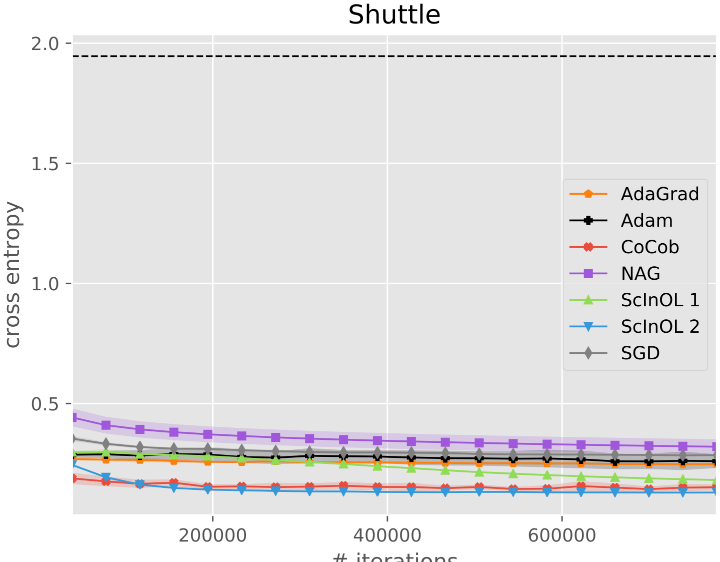
Experiment - results



Experiment - results



Experiment - results



Future work

- adjustments for batchsize > 1
- adjustments for deep models and comparison with batch-normalization
- analysis of 'dirty tricks' used in COCOB algorithm which seem to be responsible for its good performance

References I

- Cutkosky, A. and Boahen, K. A. (2017). Online learning without prior information. In *Conference on Learning Theory (COLT)*, pages 643–677. PMLR.
- Cutkosky, A. and Orabona, F. (2018). Black-box reductions for parameter-free online learning in banach spaces. In *Conference on Learning Theory (COLT)*, volume 75, pages 1493–1529. PMLR.
- Duchi, J. C., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Kotłowski, W. (2017). Scale-invariant unconstrained online learning. In *Algorithmic Learning Theory (ALT)*, volume 76, pages 412–433. PMLR.
- Luo, H., Agarwal, A., Cesa-Bianchi, N., and Langford, J. (2016). Efficient second order online learning by sketching. In *Neural Information Processing Systems (NIPS)*, pages 902–910. Curran Associates, Inc.
- McMahan, H. B. and Abernethy, J. (2013). Minimax optimal algorithms for unconstrained linear optimization. In *Advances in Neural Information Processing Systems (NIPS) 26*, pages 2724–2732.

References II

- McMahan, H. B. and Streeter, M. J. (2010). Adaptive bound optimization for online convex optimization. In *Conference on Learning Theory (COLT)*, pages 244–256.
- Orabona, F. (2013). Dimension-free exponentiated gradient. In *Advances in Neural Information Processing Systems (NIPS) 26*, pages 1806–1814.
- Orabona, F., Crammer, K., and Cesa-Bianchi, N. (2015). A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99(3):411–435.
- Orabona, F. and Tommasi, T. (2017). Training deep networks without learning rates through coin betting. In *Advances in Neural Information Processing Systems (NIPS) 30*, pages 2157–2167.
- Ross, S., Mineiro, P., and Langford, J. (2013). Normalized online learning. In *Proc. of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 537–545.
- Tieleman, T. and Hinton, G. (2012). Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.
- Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.