An Introduction to Online Convex Optimization

Wouter M. Koolen

CWI
Centrum Wiskunde & Informatica

ILPS Lunch, Friday 12th May, 2017
About me

Tenure tracker in CWI Machine Learning group on VENI grant

I work on Machine Learning Theory
- Online learning
- Easy data
- Game tree search

Adversarial Intelligence blog

http://blog.wouterkoolen.info

Local chair COLT’17 Amsterdam ← check it out!
Grand Goal of Machine Learning

Design systems that improve performance by learning from data.

\[ \text{system} + \text{data} = \text{better system} \]

- Batch learning: training $\rightarrow$ production
- Online learning: continuously improving.
Overview of today’s content

- Example
- OCO problem
- Classic algorithm for OCO
- Modern OCO developments
Example: spam classification (linear model)

- A new email arrives.
  Encoded as feature vector $x_t$ (bag of words, ...)
- System assigns a spam rating $w_t^T x_t$
  Puts it in inbox or spam folder
- User intervenes if misclassified 😊
  System gets actual label $y_t = \{-1, +1\}$

Question: How to pick $w_t$?

We need a loss function. Variety of choices:

- square loss $(w_t^T x_t - y_t)^2$
- logistic loss $-\ln \left(1 + e^{-y_t w_t^T x_t}\right)$
- hinge loss $\max\{0, 1 - y_t w_t^T x_t\}$
Example: spam classification (linear model)

- A new email arrives.
  Encoded as feature vector $x_t$ (bag of words, ...)
- System assigns a spam rating $w_t^T x_t$
  Puts it in **inbox** or **spam folder**
- User intervenes if misclassified 😊
  System gets actual label $y_t = \{-1, +1\}$

**Question**

How to pick $w_t$?
Example: spam classification (linear model)

- A new email arrives.
  - Encoded as feature vector $x_t$ (bag of words, ...)
- System assigns a spam rating $w_t^T x_t$
  - Puts it in **inbox** or **spam folder**
- User intervenes if misclassified 😞
  - System gets actual label $y_t = \{-1, +1\}$

**Question**

How to pick $w_t$?

We need a loss function. Variety of choices:

- **Square loss** $\left( w_t^T x_t - y_t \right)^2$
- **Logistic loss** $- \ln \left( 1 + e^{-y_t w_t^T x_t} \right)$
- **Hinge loss** $\max\{0, 1 - y_t w_t^T x_t\}$
What is OCO and why is it useful?

- Model for sequential decision making problems like spam filtering, portfolio investment, route planning, data compression, etc ...
- Close fit to a range of practical problems
- Very crisp (theoreticians)
- Many of the features of hard, complex problems
- Powerful and principled methods
- Basis of reductions
  - online to batch (for statistical learning)
  - bandits (for partial information problems)
  - saddle point problems (solving games)
  - non-convex problems
Online Convex Optimisation, Protocol
Online Convex Optimisation, Protocol

\[ f_1(w_1), \nabla f_1(w_1), w_2 f_2(w_2), \nabla f_2(w_2), \ldots \]
Online Convex Optimisation, Protocol

\[ w_1 \quad f_1(w_1), \quad \nabla f_1(w_1) \quad w_2 \quad f_2(w_2), \quad \nabla f_2(w_2) \ldots \]
Online Convex Optimisation, Protocol

\[ f_1(w_1), \nabla f_1(w_1) \]
Online Convex Optimisation, Protocol

$f_1(w_1), \nabla f_1(w_1)$
Online Convex Optimisation, Protocol

\[ f_1(w_1), \nabla f_1(w_1) \]

\[ f_2(w_2), \nabla f_2(w_2) \]
Online Convex Optimisation, Protocol

\[ f_1(w_1), \nabla f_1(w_1) \]

\[ f_2(w_2), \nabla f_2(w_2) \]
Online Convex Optimisation, Protocol
Online Convex Optimisation, Objective

\[ R_T = \sum_{t=1}^{T} f_t(w_t) - \min_{u} \sum_{t=1}^{T} f_t(u) \]

**Definition (Regret)**

- **Online loss**
- **Optimal loss**
Online Gradient Descent [Zinkevich, 2003]

\[ \mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t) \]
Online Gradient Descent [Zinkevich, 2003]

\[ w_{t+1} = w_t - \eta \nabla f_t(w_t) \]

\[ = \arg \min_w \sum_{s=1}^{t} w^\top \nabla f_s(w_s) + \frac{1}{\eta} \|w\|^2 \]
Online Gradient Descent [Zinkevich, 2003]

\[ \mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t) \]

\[ = \arg \min_{\mathbf{w}} \sum_{s=1}^{t} \mathbf{w}^\top \nabla f_s(\mathbf{w}_s) + \frac{1}{\eta} \| \mathbf{w} \|^2 \]

Worst-case regret guarantee:

\[ R_T = O \left( \sqrt{T} \right) \]
Online Gradient Descent [Zinkevich, 2003]

\[ w_{t+1} = w_t - \eta \nabla f_t(w_t) \]

\[ = \arg \min_w \sum_{s=1}^{t} w^T \nabla f_s(w_s) + \frac{1}{\eta} \| w \|^2 \]

Worst-case regret guarantee:

\[ R_T = O\left(\sqrt{T}\right) \]
Modern OCO

**Question**

Can we improve performance by using more domain knowledge?
Modern OCO

Question
Can we improve performance by using more domain knowledge?

Refined measures of complexity of OCO problems
- Gradient norms (maybe the gradients vanish)
- Curvature (strongly convex, exp concave, mixable)
- Stochastic scenarios (not adversarial but friendly data)
Question

Can we improve performance by using more domain knowledge?

Refined measures of complexity of OCO problems

- Gradient norms (maybe the gradients vanish)
- Curvature (strongly convex, exp concave, mixable)
- Stochastic scenarios (not adversarial but friendly data)

In both cases the key is

- Adaptive tuning of the learning rate
- Knowledge about the loss beyond convexity (add quadratic)

Go-to algorithms: AdaGrad, Online Newton Step, MetaGrad
Conclusion

I hope you got a flavour of OCO.

Happy to discuss in more detail.

Thanks!