Online Linear Optimization

Projected Gradient Descent View


Follow-the-Regularized-Leader View

- Shai Shalev-Shwartz, Online Learning and Online Convex Optimization, Foundations and Trends in Machine Learning, 2012.
- Sasha Rakhlin, Lecture Notes on Online Learning.
Kalai-Vempala

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems, COLT 2003 (link is to journal version, 2005).

J. Hannan, Approximation to Bayes risk in repeated plays, Contributions to the Theory of Games, 1957.

Model:
- $K$ experts "embedded" in $R^d$ (possibly infinitely many)
- Given an Oracle $M: g \rightarrow K$ that finds the best expert given a linear cost function
Kalai-Vempala Algorithm: Follow-the-Perturbed-Leader

FPL given oracle $M$, parameter $\epsilon$  
For rounds $t = 1, \ldots, T$

- Choose $p_t$ uniformly at random from the cube $[0, 1/\epsilon]^n$
- Let $z = g_{1:t-1} + p_t$
- Play $w_t = M(w) = \arg\min_{w} z \cdot w$

- Pay $g_t \cdot w_t$ and observe $g_t$

Achieves:

$$\text{Regret} \leq \mathcal{O}(\sqrt{T})$$

(Hiding usual dependence on $\|g_t\|$, $\max_{w,w'} \|w - w'\|$, etc.)
Learning with Structure


Log(T) Regret for Strongly Convex f


Key point: exp-concavity is really the key property, not strong convexity.

You can get log(T) regret for:
- online linear regression
- online portfolio management
Second-Order Algorithms

We've mostly considered algorithms that approximate $f(x)$ by its gradient. Instead:

**Follow The Approximate Leader (version 1)**

Inputs: convex set $\mathcal{P} \subset \mathbb{R}^n$, and the parameter $\beta$.

- On period 1, play an arbitrary $x_1 \in \mathcal{P}$.
- On period $t$, play the leader $x_t$ defined as

  $$x_t \triangleq \arg \min_{x \in \mathcal{P}} \sum_{\tau=1}^{t-1} \tilde{f}_{\tau}(x)$$

  Where for $\tau = 1, \ldots, t - 1$, define $\nabla_{\tau} = \nabla f_{\tau}(x_{\tau})$ and

  $$\tilde{f}_{\tau}(x) \triangleq f_{\tau}(x_{\tau}) + \nabla_{\tau}^T (x - x_{\tau}) + \frac{\beta}{2} (x - x_{\tau})^T \nabla_{\tau} \nabla_{\tau}^T (x - x_{\tau})$$
Also for Classification in the Mistake Bound Model

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. 
A Second-Order Perceptron Algorithm, 

Francesco Orabona and Koby Crammer. 
New Adaptive Algorithms for Online Classification, 
NIPS 2010.
"Second-Order" Algorithms for Linear Functions

The per-coordinate gradient descent algorithm is from Matthew Streeter, Brendan McMahan

For general feasible sets
H. Brendan McMahan, Matthew Streeter.

John Duchi, Elad Hazan, and Yoram Singer.
Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, JMLR 2011.
The Idea

When we've analyzed adaptive algorithms, the simplest thing to do is to use add regularization of the form

\[ r_t(x) = \sigma_t \|x\|^2 = \sigma_t x^T I x \]

Instead, only add regularization in the direction of the t'th gradient:

\[ r_t(x) = \sigma_t x^T (g_t g_t^T) x = x^T A_t x \]
The Experts Setting / Entropic Regularization

Experts Setting

EG vs GD for Squared Error

Game Theory View

The unification of these ideas as online linear optimization using entropic regularization is a more recent view.
K-Armed Bandits (EXP3) and Contextual Bandits (EXP4)

Original EXP3 and EXP4 Analysis
Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, Robert E. Schapire
The Nonstochastic Multiarmed Bandit Problem,

Analysis for Losses (No Mixing Needed)

Improved EXP4 Analysis
H. Brendan McMahan, Matthew Streeter
Tighter Bounds for Multi-Armed Bandits with Expert Advice,
COLT 2009.

High-probability bounds for EXP4
Stochastic Approaches to the Contextual Bandits Problem

Stochastic Setting


Chu, Li, Reyzin, Schapire, R. Contextual bandits with linear payoff functions, AISTATS 2011.

Model
• On each round each action has a feature vector \( x(a) \) associated with it. These can be chosen arbitrarily as long as:
• There exists a weight vector \( z^* \) such that \( <z^*, x> = E[\text{Reward}(a)| x] \) (realizability assumption).
• Goal: Do almost as well as selecting actions with the best weight vector.
Bandit Convex Optimization

General $T^{(3/4)}$ Regret
Abraham Flaxman, Adam Tauman Kalai, H. Brendan McMahan
*Online convex optimization in the bandit setting: gradient descent without a gradient*, SODA 2005.


For strongly convex functions $T^{(2/3)}$ Regret

For smooth convex functions, $T^{(2/3)}$ Regret
Bandit Linear Optimization


Varsha Dani, Thomas Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary, SODA 2006. Improves to regret $O(\text{poly}(d) T^{2/3})$.

Varsha Dani, Thomas Hayes, & Sham M. Kakade. The Price of Bandit Information for Online Optimization, NIPS 2007. The first $O(\sqrt{T})$ bound for online linear optimization, but with an inefficient algorithm. Also does lower bounds.


Peter L. Bartlett, et. al. High-Probability Regret Bounds for Bandit Online Linear Optimization, COLT 2008. High-probability $O(\sqrt{T})$ bounds, but the algorithm is not efficient.

Jacob Abernethy and Alexander Rakhlin. Beating the Adaptive Bandit with High Probability, COLT 2009. Extends “competing in the dark” with an efficient algorithm with high-probability bounds against an adaptive adversary, but only for some specific feasible sets. Has a good summary of existing results.
Online Submodular Minimization

Elad Hazan and Satyen Kale, Online Submodular Minimization, NIPS 2009.

Decision space is the set of all subsets of a ground set. Cost functions on each round are sub-modular:

A function $f : 2^{[n]} \to \mathbb{R}$ is called submodular if for all sets $S, T \subseteq [n]$ such that $T \subseteq S$, and for all elements $i \in E$, we have

$$f(T + i) - f(T) \geq f(S + i) - f(S).$$

Diminishing costs: adding $i$ to a larger set increases the cost less than adding $i$ to a smaller set. (For this to be interesting, we need the left-hand-side to be negative for some $i$). Submodularity is a kind of discrete analogue to convexity.

Simple case: linear set functions: (For minimization, again only interesting if some $a_i < 0$)

$$f(S) = \sum_{i \in S} a_i$$
Online Kernel Methods with a Budget of Support Vectors

We've mostly used simple hypothesis classes, e.g., generalized linear models. But what if we want to use kernels?

We don't know how to do this in the offline case, but online we have results:


Selective Sampling / Online Active Learning / Label Efficient Learning

For rounds $t = 1, 2, \ldots$
- adversary reveals feature vector $x$
- we predict a label (and incur loss)
- we only observe the true label $y$ if we select to query it

Goal: Achieve a good tradeoff between classification accuracy and the number of label queries we make.

This is a partial information setting, but we can control whether or not we observe the label.
Selective Sampling / Online Active Learning / Label Efficient Learning

N. Cesa-Bianchi, G. Lugosi, and G. Stoltz.  
Minimizing regret with label efficient prediction,  

F. Orabona and N. Cesa-Bianchi.  
Better algorithms for selective sampling, ICML 2011.

N. Cesa-Bianchi, C. Gentile, and F. Orabona.  
Robust bounds for classification via selective sampling, ICML 2009.

Worst-case analysis of selective sampling for linear classification  

Ofer Dekel, Claudio Gentile, and Karthik Sridharan.  
Robust selective sampling from single and multiple teachers, COLT 2010.
Other Problems

Online PCA
Manfred K. Warmuth, Dima Kuzmin
Randomized Online PCA Algorithms with Regret Bounds that are Logarithmic in the Dimension,

Online One-Class Prediction (e.g., outlier detection)

Online Ranking
Koby Crammer and Yoram Singer