Efficient Online Learning via Randomized Rounding

Nicolò Cesa-Bianchi
Università degli Studi di Milano

Ohad Shamir*
Microsoft Research New England

NIPS
December 2011
Online Learning

• For $t = 1, 2, \ldots$
  – **Adversary**: Pick loss function $\ell_t$
  – **Learner**: Predict $w_t$; Receive $\ell_t$ and suffer loss $\ell_t(w_t)$

• Goal - minimize regret:

$$\sum_{t=1}^{T} \ell_t(w_t) - \min_{w\in W} \sum_{t=1}^{T} \ell_t(w)$$

• Lots of attention in recent years
  – No stochastic assumptions
  – Simple and efficient algorithms for convex problems
  – Strong theoretical guarantees
An Inconvenient Truth

- Most online algorithms can be seen as variants of the same algorithmic framework (mirror-descent / FTRL)

\[ w_{t+1} = \operatorname*{arg\,min}_{w \in W} \eta_t \langle \nabla \ell_t(w_t), w \rangle + B_\psi(w || w_t) \]

- Srebro, Sridharan and Tewari (2011) – ignoring tractability, mirror descent is in some sense universal

- Is this the only way to do online learning?
This Paper

• A completely different approach to **efficient** online learning
  – R² Forecaster: based on “randomized playout” and randomized rounding of sub-gradients
  – **Efficient** (polynomial runtime) whenever one can efficiently compute an empirical risk minimizer

• Applications:
  – **We solve an open question** linking efficient batch learning and online transductive learning
  – **First efficient online algorithm** for collaborative filtering using trace-norm constrained matrices
Starting Point: Prediction of Binary Sequences

• At each round, predict $y_t \in \{-1, +1\}$, using 0-1 loss
  – Randomized predictions allowed:
    • predict $p_t \in [-1, +1]$
    • suffer expected loss $|p_t - y_t|$

• Goal: minimize regret w.r.t. comparison class of prediction sequences $\mathcal{F} \subseteq [-1, +1]^T$

• Minimax regret analysis provided by Chung (1994), Cesa-Bianchi et al. (1997)
Prediction of Binary Sequences

- Implicit in those papers: simple (but inefficient) \textbf{minimax-optimal} algorithm

- At round $t$:
  1. Define random sequences
     - $s^- = (y_1, y_2, \ldots, y_{t-1}, -1, Y_{t+1}, \ldots, Y_T)$
     - $s^+ = (y_1, y_2, \ldots, y_{t-1}, +1, Y_{t+1}, \ldots, Y_T)$
  2. Define the ERM difference value
     \[ r_t = \left( \inf_{f \in \mathcal{F}} \sum_{i=1}^{T} |f_i - s^-_i| \right) - \left( \inf_{f \in \mathcal{F}} \sum_{i=1}^{T} |f_i - s^+_i| \right) \]

     ERM value on $s^-$
     ERM value on $s^+$
Minimax regret of algorithm equals the Rademacher complexity

\[ R_T(\mathcal{F}) = \mathbb{E}_\sigma \left[ \sup_{f \in \mathcal{F}} \sum_{t=1}^{T} \sigma_t f_t \right] \]

– Used to characterize sample complexity of learning \( \mathcal{F} \) in a batch statistical setting
Prediction of Binary Sequences

- Our first observation: **algorithm can be made computationally efficient**
  - Instead of computing $\mathbb{E}[r_t]$, enough to use a single draw of $r_t$

- However, still **extremely limited** setting:
  - Binary sequences
  - 0-1/absolute loss

- **Next**: how to deal with
  - Real valued outcomes
  - General convex Lipschitz loss functions
The $R^2$ Forecaster

• First attempt: **Extend** minimax analysis to more general setups
  – Won’t work: Minimax analysis **extremely brittle**
• Different approach: **Reduce** the problem to predicting binary sequences

$$ (y_1, y_2, \ldots, y_{t-1}) \rightarrow \left( \partial \ell(p_1, y_1), \partial \ell(p_2, y_2), \ldots, \partial \ell(p_{t-1}, y_{t-1}) \right) \rightarrow (+1, -1, -1, \ldots, +1) $$
The $R^2$ Forecaster

- Parameters: $\eta$, horizon $T$, Loss Lipschitz parameter $\rho$, loss bound $b$
- For $t = 1, \ldots, T$
  1. Define the random sequences
     - $s^- = (z_1, z_2, \ldots, z_{t-1}, -1, Y_{t+1}, \ldots, Y_T)$
     - $s^+ = (z_1, z_2, \ldots, z_{t-1}, +1, Y_{t+1}, \ldots, Y_T)$
     where $z_1, z_2, \ldots, z_{t-1}$ determined on previous rounds
  2. Compute $\eta T$ independent draws of random variable
     $$b \left( \inf_{f \in F} \sum_{i=1}^{T} |f_i - s_i^-| - \inf_{f \in F} \sum_{i=1}^{T} |f_i - s_i^+| \right)$$
  3. Predict their empirical average $p_t$, receive outcome $y_t$ and suffer loss
     $$\ell(p_t, y_t)$$
Theorem

- Suppose loss is bounded and 1-Lipschitz
- For any comparison class $\mathcal{F}$, with probability $1 - \delta$, regret of $R^2$ Forecaster at most

$$R_T(\mathcal{F}) + O\left(\sqrt{\frac{1}{\eta} \log \left(\frac{T}{\delta}\right)} T\right)$$

- If can compute ERM in time $c$, then $R^2$ forecaster runs in time $O(c \eta T^2)$
Application 1: Transductive Online Learning

- Kakade and Kalai (NIPS 2005)
  - Online learning implies batch learning, but is the reverse true?
  - Show that for binary classification, efficient batch learning $\rightarrow$ efficient transductive online learning...
  - ...however, at inferior rate ($T^{3/4}$)
  - Main open question: can be improved?

- $R^2$ forecaster achieves
  - Optimal $\sqrt{T}$ rate (up to log factors)
  - Strictly more general setting
Application 2: Collaborative Filtering

• Goal: predict entries of a mostly unknown matrix

• Motivation:
  – recommender systems (Netflix),
  – dealing with incomplete data
  – ...

• Lots of recent work in a batch statistical setting
  (observed entries sampled i.i.d.)
  – But i.i.d. assumption questionable in practice...
Application 2: Collaborative Filtering

- **Online setting:**
  - Start with unknown \( m \times n \) matrix
  - For \( t=1,2,... \)
    - **Adversary** chooses matrix entry location \((i_t, j_t)\) and value \( y_{i_t,j_t} \); reveals location
    - **Learner** predicts value \( p_{i_t,j_t} \)
    - **Adversary** reveals value; learner suffers loss \( \ell(p_{i_t,j_t}, y_{i_t,j_t}) \)

- Mild assumption: adversary doesn’t pick same entry location twice
- Regret measured against all fixed \( m \times n \) matrices with trace-norm \( O(\sqrt{mn}) \)
Application 2: Collaborative Filtering

- Standard online techniques don’t work as-is
  - For $m \times n$ matrices, per-round regret is a trivial $\sqrt{mn/T}$
- $R^2$ forecaster also doesn’t seem to work as-is ...
  - Sequence of entry locations needs to be specified in advance
  - Horizon needs to be known
- ... but we show how to make it work
  - Using unique properties of Rademacher complexity in this setting
  - Regret guarantees parallel recent sample complexity guarantees in a stochastic setting (Shalev-Shwartz and S., 2010)
  - See poster for details!
Conclusions

• A novel and very different approach to efficient online learning
  – Efficient whenever ERM efficiently computable

• Two applications:
  – Transductive online to batch learning
  – Online collaborative filtering

• Open questions:
  – Other applications? Extension to other settings?
  – Make fully practical
  – Relationship to standard methods?