STATS 701 – Theory of Reinforcement Learning
Online Learning with Full Information

Ambuj Tewari

Associate Professor, Department of Statistics, University of Michigan
tewaria@umich.edu
https://ambujtewari.github.io/stats701-winter2021/

Slide Credits: Wouter Koolen @ CWI Amsterdam, The Netherlands

Winter 2021

Plan

1. The Experts Problem; Exponential Weights

2. Two Peeks Beyond the Basics
    - Follow the Regularized Leader and Mirror Descent
    - Online Quadratic Optimization; Online Newton Step

3. Conclusion and Extensions

From Learning Parameters to Picking Actions

We now turn to the second elementary online learning task.

- Decision Theoretic Online Learning
- Experts setting (also: Hedge setting)
- Prediction with Expert Advice

From Learning Parameters to Picking Actions

We now turn to the second elementary online learning task.

- Decision Theoretic Online Learning
- Experts setting (also: Hedge setting)
- Prediction with Expert Advice

Protocol: Prediction With Expert Advice

Given: game length T, number K of experts

For $t = 1, 2, \ldots, T$,

- Learner chooses a distribution $\boldsymbol{w}_t \in \triangle_K$ on K "experts".
- Adversary reveals loss vector $\boldsymbol{\ell}_t \in [0, 1]^K$.
- Learner's loss is the **dot loss** $\boldsymbol{w}_t^{\intercal} \boldsymbol{\ell}_t = \sum_{k=1}^K w_t^k \ell_t^k$

From Learning Parameters to Picking Actions

We now turn to the second elementary online learning task.

- Decision Theoretic Online Learning
- Experts setting (also: Hedge setting)
- Prediction with Expert Advice

Protocol: Prediction With Expert Advice

Given: game length T, number K of experts

For $t = 1, 2, \ldots, T$,

- Learner chooses a distribution $\boldsymbol{w}_t \in \triangle_K$ on K “experts”.
- Adversary reveals loss vector $\boldsymbol{\ell}_t \in [0,1]^K$.
- Learner’s loss is the **dot loss** $\boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t = \sum_{k=1}^K w_t^k \ell_t^k$

The goal: control the regret (w.r.t. the best expert after T rounds)

$$\mathcal{R}_T = \sum_{t=1}^{T} \boldsymbol{w}_t^\intercal \boldsymbol{\ell}_t - \min_{k \in [K]} \sum_{t=1}^{T} \ell_t^k$$

using a computationally efficient algorithm for learner.

Let's apply what we know

Observations:

- Dot loss $\boldsymbol{u} \mapsto \boldsymbol{u}^\intercal \boldsymbol{\ell}_t$ is *linear* (hence convex).
- Gradient $\boldsymbol{\ell}_t \in [0,1]^K$ bounded by $\|\boldsymbol{\ell}_t\| \leq \sqrt{K}$.
- Probability simplex $\triangle_K$ is contained in unit ball.

So: Instance of Online Convex Optimization.
OGD with $D = 1$ and $G = \sqrt{K}$ gives $\mathcal{R}_T \leq \sqrt{KT}$.

Let's apply what we know

Observations:

- Dot loss $\boldsymbol{u} \mapsto \boldsymbol{u}^\intercal \boldsymbol{\ell}_t$ is *linear* (hence convex).
- Gradient $\boldsymbol{\ell}_t \in [0, 1]^K$ bounded by $\|\boldsymbol{\ell}_t\| \leq \sqrt{K}$.
- Probability simplex $\triangle_K$ is contained in unit ball.

So: Instance of Online Convex Optimization.
OGD with $D = 1$ and $G = \sqrt{K}$ gives $\mathcal{R}_T \leq \sqrt{KT}$.
Q: Optimal?

Let's apply what we know

Observations:

- Dot loss $\boldsymbol{u} \mapsto \boldsymbol{u}^\intercal \boldsymbol{\ell}_t$ is *linear* (hence convex).
- Gradient $\boldsymbol{\ell}_t \in [0,1]^K$ bounded by $\|\boldsymbol{\ell}_t\| \leq \sqrt{K}$.
- Probability simplex $\triangle_K$ is contained in unit ball.

So: Instance of Online Convex Optimization.
OGD with $D = 1$ and $G = \sqrt{K}$ gives $\mathcal{R}_T \leq \sqrt{KT}$.
Q: Optimal?
Maybe not. There are no points with loss difference $\sqrt{K}$ in the simplex ...

Exponential Weigths / Hedge Algorithm

Algorithm: Exponential Weights (EW)

EW with *learning rate* $\eta > 0$ plays weights in round t:

$$w_t^k = \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s^k}}{\sum_{j=1}^K e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}}. \tag{EW}$$

Exponential Weigths / Hedge Algorithm

Algorithm: Exponential Weights (EW)

EW with *learning rate* $\eta > 0$ plays weights in round t:

$$w_t^k = \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s^k}}{\sum_{j=1}^K e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}}. \tag{EW}$$

or, equivalently, $w_1^k = \frac{1}{K}$ and

$$w_{t+1}^k = \frac{w_t^k e^{-\eta \ell_t^k}}{\sum_{j=1}^K w_t^j e^{-\eta \ell_t^j}} \tag{EW, incremental}$$

Exponential Weigths / Hedge Algorithm

Algorithm: Exponential Weights (EW)

EW with *learning rate* $\eta > 0$ plays weights in round t:

$$w_t^k = \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s^k}}{\sum_{j=1}^K e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}}. \tag{EW}$$

or, equivalently, $w_1^k = \frac{1}{K}$ and

$$w_{t+1}^k = \frac{w_t^k e^{-\eta \ell_t^k}}{\sum_{j=1}^K w_t^j e^{-\eta \ell_t^j}} \tag{EW, incremental}$$

Theorem (EW regret bd, Freund and Schapire 1997)

The regret of EW is bounded by $\mathcal{R}_T \leq \frac{\ln K}{\eta} + T\frac{\eta}{8}$.

Corollary

Tuning $\eta = \sqrt{\frac{8 \ln K}{T}}$ *yields* $\mathcal{R}_T \leq \sqrt{T/2 \ln K}$.

EW Analysis

Applying *Hoeffding's Lemma* to the loss of each round gives

$$\sum_{t=1}^{T} \boldsymbol{w}_t^{\intercal} \boldsymbol{\ell}_t \;\leq\; \sum_{t=1}^{T} \left(\underbrace{\frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} w_t^k e^{-\eta \ell_t^k} \right)}_{\text{"mix loss"}} + \underbrace{\eta/8}_{\text{overhead}} \right)$$

Crucial observation is that cumulative mix loss *telescopes*

$$\begin{aligned}
\sum_{t=1}^{T} \frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} w_t^k e^{-\eta \ell_t^k} \right) &= \sum_{t=1}^{T} \frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} \frac{e^{-\eta \sum_{s=1}^{t-1} \ell_s^k}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}} e^{-\eta \ell_t^k} \right) \\
&= \sum_{t=1}^{T} \frac{-1}{\eta} \ln \left(\frac{\sum_{k=1}^{K} e^{-\eta \sum_{s=1}^{t} \ell_s^k}}{\sum_{j=1}^{K} e^{-\eta \sum_{s=1}^{t-1} \ell_s^j}} \right) \\
&\overset{\text{telescopes}}{=} \frac{-1}{\eta} \ln \left(\sum_{k=1}^{K} e^{-\eta \sum_{t=1}^{T} \ell_t^k} \right) + \frac{\ln K}{\eta} \\
&\leq \min_{k \in [K]} \sum_{t=1}^{T} \ell_t^k + \frac{\ln K}{\eta}.
\end{aligned}$$

Summary so far

Balancing act: “model complexity” vs “overfitting”

Theorem (OGD)

$$\mathcal{R}_T \leq \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T$$

Theorem (EW)

$$\mathcal{R}_T \leq \frac{\ln K}{\eta} + \frac{\eta}{8} T$$

Summary so far

Balancing act: "model complexity" vs "overfitting"

Theorem (OGD)

$$\mathcal{R}_T \;\leq\; \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T$$

Theorem (EW)

$$\mathcal{R}_T \;\leq\; \frac{\ln K}{\eta} + \frac{\eta}{8} T$$

Generates many follow-up questions:

- What if horizon T is not fixed? Anytime guarantees?
- What if gradient bound G is not known a priori?
- Can we have the actual gradient norms?
- What if model complexity (D) is not known? Not uniformly bounded? See Orabona and Cutkosky ICML'20 tutorial.

Summary so far

Balancing act: “model complexity” vs “overfitting”

Theorem (OGD)

$$\mathcal{R}_T \;\le\; \tfrac{D^2}{2\eta} + \tfrac{\eta}{2} G^2 T$$

Theorem (EW)

$$\mathcal{R}_T \;\le\; \tfrac{\ln K}{\eta} + \tfrac{\eta}{8} T$$

Generates many follow-up questions:

- What if horizon T is not fixed? Anytime guarantees?
- What if gradient bound G is not known a priori?
- Can we have the actual gradient norms?
- What if model complexity (D) is not known? Not uniformly bounded? See Orabona and Cutkosky ICML’20 tutorial.

Need refined analyses $\Rightarrow$ Restarts (doubling trick), decreasing η_t (AdaGrad/AdaHedge), learning the learning rate η (MetaGrad), ...

Summary so far

Balancing act: "model complexity" vs "overfitting"

Theorem (OGD)

$$\mathcal{R}_T \leq \frac{D^2}{2\eta} + \frac{\eta}{2} G^2 T$$

Theorem (EW)

$$\mathcal{R}_T \leq \frac{\ln K}{\eta} + \frac{\eta}{8} T$$

Generates many follow-up questions:

- What if horizon T is not fixed? Anytime guarantees?
- What if gradient bound G is not known a priori?
- Can we have the actual gradient norms?
- What if model complexity (D) is not known? Not uniformly bounded? See Orabona and Cutkosky ICML'20 tutorial.

Need refined analyses $\Rightarrow$ Restarts (doubling trick), decreasing η_t (AdaGrad/AdaHedge), learning the learning rate η (MetaGrad), ...
Active research area!

FTRL/MD "sneak peek"

Q: What if my domain does not look like either ball or simplex?

FTRL/MD "sneak peek"

Q: What if my domain does not look like either ball or simplex?

Algorithm: Follow the Regularized Leader (FTRL) (with linearized losses)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \sum_{s=1}^{t} f_s(\boldsymbol{u}) + \frac{1}{\eta} R(\boldsymbol{u})$$

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(\boldsymbol{w}_s) \rangle + \frac{1}{\eta} R(\boldsymbol{u})$$

Algorithm: Mirror Descent (MD)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \langle \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t) \rangle + \frac{1}{\eta} B(\boldsymbol{u} \| \boldsymbol{w}_t)$$

FTRL/MD "sneak peek"

Q: What if my domain does not look like either ball or simplex?

Algorithm: Follow the Regularized Leader (FTRL) (with linearized losses)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \sum_{s=1}^{t} f_s(\boldsymbol{u}) + \frac{1}{\eta}R(\boldsymbol{u})$$

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(\boldsymbol{w}_s)\rangle + \frac{1}{\eta}R(\boldsymbol{u})$$

Algorithm: Mirror Descent (MD)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \langle \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t)\rangle + \frac{1}{\eta}B(\boldsymbol{u}\|\boldsymbol{w}_t)$$

		Regularizer R	**Bregman Divergence** B
Examples:	OGD	sq. Euclidean norm	sq. Euclidean distance
	EW	Shannon entropy	Kullback-Leibler divergence

FTRL/MD "sneak peek"

Q: What if my domain does not look like either ball or simplex?

Algorithm: Follow the Regularized Leader (FTRL) (with linearized losses)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \sum_{s=1}^{t} f_s(\boldsymbol{u}) + \frac{1}{\eta} R(\boldsymbol{u})$$

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(\boldsymbol{w}_s)\rangle + \frac{1}{\eta} R(\boldsymbol{u})$$

Algorithm: Mirror Descent (MD)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \langle \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t)\rangle + \frac{1}{\eta} B(\boldsymbol{u}\|\boldsymbol{w}_t)$$

		Regularizer R	**Bregman Divergence** B
Examples:	OGD	sq. Euclidean norm	sq. Euclidean distance
	EW	Shannon entropy	Kullback-Leibler divergence

Other entropies: Burg, Tsallis, Von Neumann, ... Connections to continuous exponential weights [van der Hoeven et al., 2018].

FTRL/MD "sneak peak" performance

Algorithm: Follow the Regularized Leader (FTRL) (with linearized losses)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \sum_{s=1}^{t} f_s(\boldsymbol{u}) + \frac{1}{\eta} R(\boldsymbol{u})$$

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(\boldsymbol{w}_s)\rangle + \frac{1}{\eta} R(\boldsymbol{u})$$

Algorithm: Mirror Descent

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \langle \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t)\rangle + \frac{1}{\eta} B(\boldsymbol{u}\|\boldsymbol{w}_t)$$

FTRL/MD "sneak peak" performance

Algorithm: Follow the Regularized Leader (FTRL) (with linearized losses)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \sum_{s=1}^{t} f_s(\boldsymbol{u}) + \frac{1}{\eta}R(\boldsymbol{u})$$

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \sum_{s=1}^{t} \langle \boldsymbol{u}, \nabla f_s(\boldsymbol{w}_s)\rangle + \frac{1}{\eta}R(\boldsymbol{u})$$

Algorithm: Mirror Descent

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{U}} \langle \boldsymbol{u}, \nabla f_t(\boldsymbol{w}_t)\rangle + \frac{1}{\eta}B(\boldsymbol{u}\|\boldsymbol{w}_t)$$

Theorem (AdaFTRL, Orabona and Pál 2015)

Fix a norm $\|\cdot\|$ with associated dual norm $\|\cdot\|_\star$. Let $R : \mathcal{U} \to [0, D^2]$ be strongly convex w.r.t. $\|\cdot\|$. AdaFTRL ensures

$$\mathcal{R}_T \leq 2D\sqrt{\sum_{t=1}^{T} \|\nabla f_t(\boldsymbol{w}_t)\|_\star^2} + 2 \cdot \textit{loss range}.$$

Quadratic Losses

So far we used convexity to "linearise"

$$f_t(\boldsymbol{u}) \geq f_t(\boldsymbol{w}_t) + \langle \boldsymbol{u} - \boldsymbol{w}_t, \nabla f_t(\boldsymbol{w}_t) \rangle,$$

and our methods essentially operated on linear losses. But what if we know there is curvature?

- How to represent/quantify curvature?
- How to efficiently manipulate curvature?
- How much can we reduce the regret?

Curvature assumptions

Assumption: Quadratic loss lower bound

There is a matrix $\boldsymbol{M}_t \succeq \mathbf{0}$ such that

$$f_t(\boldsymbol{u}) \;\geq\; \underbrace{f_t(\boldsymbol{w}_t) + \langle \boldsymbol{u} - \boldsymbol{w}_t, \nabla f_t(\boldsymbol{w}_t)\rangle + \frac{1}{2}(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{M}_t(\boldsymbol{u} - \boldsymbol{w}_t)}_{=:q_t(\boldsymbol{u})}$$

for each $\boldsymbol{u} \in \mathcal{U}$.

Curvature assumptions

Assumption: Quadratic loss lower bound

There is a matrix $M_t \succeq \mathbf{0}$ such that

$$f_t(\boldsymbol{u}) \;\geq\; \underbrace{f_t(\boldsymbol{w}_t) + \langle \boldsymbol{u} - \boldsymbol{w}_t, \nabla f_t(\boldsymbol{w}_t)\rangle + \frac{1}{2}(\boldsymbol{u} - \boldsymbol{w}_t)^\intercal \boldsymbol{M}_t(\boldsymbol{u} - \boldsymbol{w}_t)}_{=:q_t(\boldsymbol{u})}$$

for each $\boldsymbol{u} \in \mathcal{U}$.

Two main classes of instances

- squared Euclidean distance: $f_t(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u} - x_t\|^2$ satisfies the assumption with $\boldsymbol{M}_t = \boldsymbol{I}$. More generally, strongly convex functions have $\boldsymbol{M}_t \propto \boldsymbol{I}$.
- linear regression: $f_t(\boldsymbol{u}) = (y_t - \langle \boldsymbol{u}, \boldsymbol{x}_t\rangle)^2$ satisfies the assumption with $\boldsymbol{M}_t = \boldsymbol{x}_t\boldsymbol{x}_t^\intercal$. More generally, exp-concave functions have $\boldsymbol{M}_t \propto \nabla_t f_t(\boldsymbol{w}_t)\nabla_t f_t(\boldsymbol{w}_t)^\intercal$.

ONS Algorithm

Algorithm: Online Newton Step (FTRL variant)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \sum_{s=1}^{t} q_s(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}\|^2$$

Computing the iterate $\boldsymbol{w}_{t+1}$ amounts to minimising a convex quadratic. Often (depending on $\mathcal{U}$) closed-form solution or 1d line search.

- For $\boldsymbol{M}_t \propto \boldsymbol{I}$, takes $O(d)$ per round.
- For rank-one $\boldsymbol{M}_t$, can do update in $O(d^2)$ per round.
- In both cases, need to take care of projection onto $\mathcal{U}$.

ONS Performance

Algorithm: Online Newton Step (FTRL version)

$$\boldsymbol{w}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathcal{U}} \sum_{s=1}^{t} q_s(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}\|^2$$

Theorem (ONS strcvx bd, Hazan et al. 2006)

For the strongly convex case $\boldsymbol{M}_t \propto \boldsymbol{I}$, ONS guarantees

$$\mathcal{R}_T = O(\ln T)$$

Algorithm reduces to OGD with specific decreasing step-size η_t

Theorem (ONS expccv bd, Hazan et al. 2006)

For the exp-concave case $\boldsymbol{M}_t \propto \boldsymbol{g}_t \boldsymbol{g}_t^\intercal$, ONS guarantees

$$\mathcal{R}_T = O(d \ln T)$$

ONS Discussion

- Convex quadratics closed under taking sums. Run-time independent of T.
- Curvature gives huge reduction in regret: $\sqrt{T}$ to $\ln T$.
- Matrix sketching techniques allow trading off run-time $O(d^2)$ vs $O(d)$ with regret $O(\ln T)$ vs $O(\sqrt{T})$ [Luo et al., 2016].

Conclusion

- Online Learning a powerful and versatile tool
- Environment-as-black-box. Adversarial.
- Foundation for optimization, statistical learning, games, ...
- Techniques we saw here will reappear when we discuss adversarial bandits and adversarial MDPs

Conclusion

- Online Learning a powerful and versatile tool
- Environment-as-black-box. Adversarial.
- Foundation for optimization, statistical learning, games, ...
- Techniques we saw here will reappear when we discuss adversarial bandits and adversarial MDPs

Some (of many) cool things we left out:

- First-order (small loss) and second-order (small variance) bounds
- Adaptivity to friendly stochastic environments (best of both worlds, interpolation)
- Optimistic MD (predicting the upcoming gradient)
- Non-stationarity (tracking, adaptive/dynamic regret, path length)
- Beyond convexity (star-convex, geometrically convex, ...)
- Supervised Learning and (stochastic) complexities (VC, Littlestone, Rademacher, ...)

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1): 119–139, August 1997.

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *Learning Theory*, pages 499–513, 2006.

Haipeng Luo, Alekh Agarwal, Nicolò Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems 29*, pages 902–910. 2016.

Francesco Orabona and Dávid Pál. Scale-free algorithms for online linear optimization. In *Algorithmic Learning Theory*, pages 287–301, 2015.

Dirk van der Hoeven, Tim van Erven, and Wojciech Kotłowski. The many faces of exponential weights in online learning. volume 75 of *Proceedings of Machine Learning Research*, pages 2067–2092, 06–09 Jul 2018.