Title of Thesis is here

by

Thiruvenkadam Sivaprakasam Radhakrishnan

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master's in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2023

Chicago, Illinois

Defense Committee:
Prof. Firstname1 Lastname1, Chair and Advisor
Prof. Firstname2 Lastname2
Prof. Firstname3 Lastname3
Prof. Firstname4 Lastname4
Prof. Firstname5 Lastname5

ACKNOWLEDGMENTS

The thesis has been completed... (INSERT YOUR TEXTS)

YOUR INITIAL

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AMS	American Mathematical Society
CTAN	Comprehensive TEX Archive Network
TUG	TEX Users Group
UIC	University of Illinois at Chicago
UICTHESI	Thesis formatting system for use at UIC.

SUMMARY

Put your summary of thesis here.

Things to add:

1. Online Learning and Online Convex optimization
    (a) General intro
    (b) FoReL
    (c) Mirror Descent
    (d) Hedge
    (e) Mirror Descent with KL Divergence
2. Policy gradients
    (a) General intro
    (b) Actor critic methods
    (c) PPO
3. NeuRD
4. Making the connection between Mirror Descent and Policy gradients
    (a) MDPO
    (b) MMD
5. NeuRD fix

(a) MDPO

(b) MMD

6. Experiments

(a) Rock paper scissors; Khun Poker

(b) MDPO vs MDPO-NR

(c) MMD vs MMD-NR

CHAPTER 1

DERIVATIONS

1.1 Online Learning

1.1.1 FoReL

1.2 Online Mirror Descent

The FoReL update rule is,

$$\begin{aligned} w_{t+1} &= argmin_w R(w) + \sum_{i=1}^{t} \langle w, z_t \rangle \\ &= argmin_w R(w) + \langle w, z_{1:t} \rangle \\ &= argmax_w \langle w, -z_{1:t} \rangle - R(w) \end{aligned}$$

Let $g(\theta) = argmax_w \langle w, \theta \rangle - R(w)$. Then the FoReL update rule can be written as,

$$\theta_{t+1} = \theta_t - z_t w_{t+1} = g(\theta_{t+1})$$

where $g(\theta)$ is a link function that projects the predictions back to the convex set S.

Using different regularization functions yield different algorithms that have different regret bounds.

Theorem 1 *If R is a $(\frac{1}{\eta})$-strongly-convex function over S with respect to some norm $\|.\|$, and OMD is run on a sequence with the following link function*

$$g(\theta) = argmax_w(\langle w, \theta \rangle - R(w))$$

then,

$$\forall u \in S, Regret_T(u) \leq R(u) - min_{v \in S} R(v) + \eta \sum_{t=1}^{T} \|z\|_*^2$$

where $\|.\|_*$ is the dual norm.

1.2.1 Hedge

Hedge or normalized Exponentiated Gradient is OMD with entropic regularization. The link function here is

$$g_i(\theta) = \frac{e^{\eta\theta[i]}}{\sum_j e^{\eta\theta[j]}}. \tag{1.1}$$

Fitting this into the OMD framework yields the following update rule,

$$w_{t+1}[i] = \frac{w_t[i]e^{-\eta z_t[i]}}{\sum_j w_t[j]e^{-\eta z_t[j]}}$$

We can analyze the regret bounds of Hedge with $R(w) = \frac{1}{\eta}\sum_i w[i]log(w[i])$.

It is also useful to analyze OMD with the language of duality. The framework utilizing duality makes it easier in deriving new algorithms and also in proving tighter regret bounds.

1.2.2 Fenchel Conjugacy

The Fenchel conjugate of a function f is defined as,

$$f^*(\theta) = max_u \langle u, \theta \rangle - f(u)$$

Fenchel conjugate by definition implies the Fenchel-Young inequality:

$$\forall u, f^*(\theta) \geq \langle u, \theta \rangle - f(u)$$

.

If u is a sub-gradient of f^* at θ and if f^* is differentiable, then the equality condition holds when $u = \nabla f^*(\theta)$.

1.2.3 Bergman Divergences

For a differentiable function R, the Bergman divergence between two vectors is defined as,

$$D_R(w\|u) = R(w) - R(u) + (\langle R(u), w - u \rangle) \tag{1.2}$$

Bergman divergence is asymmetric and is always non-negative if R is convex.

1.2.4 Online Mirror Descent in terms of Duality

The link function in the OMD framework is defined as,

$$g(\theta) = argmax_w(\langle w, \theta \rangle - R(w)).$$

This can be also rewritten in terms of the conjugate of R as,

$$g(\theta) = \nabla R^*(\theta)$$

With this, we can obtain different algorithms by using different regularization functions and deriving the update rules by using their conjugate.

1.2.5 KL-Divergence and its Fenchel Conjugate

KL-Divergence is a distance metric between two probability distributions and is defined as,

$$D_{KL}(p\|q) = \sum_i p[i] log \frac{p[i]}{q[i]}$$

The Fenchel Conjugate of KL-Divergence is given by,

$$f_q^*(x) = \log(\sum_i q_i e^{x_i}).$$

1.3 MDPO

The on-policy MDPO update rule is written as,

$$\theta_{k+1} \leftarrow argmax_{\theta \in \Theta} \Psi(\theta, \theta_k)$$

where,

$$\Psi(\theta, \theta_k) = \mathbb{E}_{s\sim\rho_{\theta_k}}[\mathbb{E}_{a\sim\pi_\theta}[A^{\theta_k}(s,a)] - \frac{1}{t_k} KL(s; \pi_\theta, \pi_{\theta_k})]$$

The gradient of the above update rule is as follows:

$$\begin{aligned}
\nabla_\theta \Psi(\theta, \theta_k)|_{\theta=\theta_k} &= \mathbb{E}_{s\sim\rho_{\theta_k}}[\sum_a \nabla_\theta \pi_\theta(a|s) A^{\theta_k}(s,a)] \\
&= \mathbb{E}_{s\sim\rho_{\theta_k}}[\sum_a \pi_{\theta_k}(a|s) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\theta_k}(s,a)] \\
&= \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\pi_{\theta_k}}[\nabla \log \pi_{\theta_k}(a|s) A^{\theta_k}(s,a)]
\end{aligned}$$

For one-step MDPO, the gradient of the KL-Divergence term becomes 0. Hence it is proposed that the policy update at each iteration k is done through m steps of SGD.

$$\theta_k^{(0)} = \theta_k,$$

$$\theta_k^{(i+1)} \leftarrow \theta_k^{(i)} + \eta \nabla_\theta \Psi(\theta, \theta_k)|_{\theta=\theta_k^{(i)}}$$

and, $\theta_{k+1} = \theta_k^{(m)}$.

Then the gradient of the objective function evaluated at each step of the SGD update is,

$$\nabla_\theta \Psi(\theta, \theta_k)|_{\theta=\theta_k^{(i)}} = \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\pi_{\theta_k}}[\frac{\pi_{\theta_k}^{(i)}}{\pi_{\theta_k}} \nabla \log \pi_{\theta_k^{(i)}}(a|s) A^{\theta_k}(s,a)]$$
$$- \frac{1}{t_k} \mathbb{E}_{s\sim\rho_{\theta_k}}[\nabla_\theta KL(s; \pi_\theta, \pi_{\theta_k})|_{\theta=\theta_k^{(i)}}].$$

$$KL(s; \pi_\theta, \pi_{\theta_k}) = \sum_{a\in\mathcal{A}} \pi_{\theta_k^{(i)}}(a|s) \log \frac{\pi_{\theta_k^{(i)}}(a|s)}{\pi_{\theta_k}(a|s)}$$

The gradient of the KL-Divergence term is given by,

$$\begin{aligned}
\nabla_\theta KL(s; \pi_\theta, \pi_{\theta_k})|_{\theta=\theta_k^{(i)}} &= \sum_{a\in\mathcal{A}} [\nabla_{\theta_k^{(i)}}(\pi_{\theta_k^{(i)}}(a|s) \log \pi_{\theta_k}^{(i)}(a|s)) - \nabla_{\theta_k^{(i)}}(\pi_{\theta_k^{(i)}}(a|s) \log \pi_{\theta_k}(a|s))] \\
&= \log \pi_{\theta_k^{(i)}}(a|s) \nabla_{\theta_k^{(i)}} \pi_{\theta_k^{(i)}}(a|s) + \nabla_{\theta_k^{(i)}} \pi_{\theta_k^{(i)}}(a|s) - \log \pi_{\theta_k}(a|s) \nabla_{\theta_k^{(i)}} \pi_{\theta_k^{(i)}}(a|s) \\
&= \sum_{a\in\mathcal{A}} [(\log \pi_{\theta_k}^{(i)}(a|s) + 1 - \log \pi_{\theta_k}(a|s)) \nabla_{\theta_k^{(i)}} \pi_{\theta_k^{(i)}}(a|s)].
\end{aligned}$$

As for the first term of the gradient, it can be seen that the gradient includes a term to account for the fact that the action a was sampled from the policy π_{θ_k}

$$
\begin{aligned}
\nabla_\theta \Psi(\theta, \theta_k)|_{\theta=\theta_k^{(i)}} &= \mathbb{E}_{s\sim\rho_{\theta_k}}[\sum_a \nabla_{\theta_k^{(i)}} \pi_{\theta_k^{(i)}}(a|s) A^{\theta_k}(s,a)] \\
&= \mathbb{E}_{s\sim\rho_{\theta_k}}[\sum_a \pi_{\theta_k}(a|s) \frac{\pi_{\theta_k^{(i)}}(a|s)}{\pi_{\theta_k}(a|s)} \frac{\nabla_{\theta_k^{(i)}} \pi_{\theta_k^{(i)}}(a|s)}{\pi_{\theta_k^{(i)}}(a|s)} A^{\theta_k}(s,a)] \\
&= \mathbb{E}_{s\sim\rho_{\theta_k}, a\sim\pi_{\theta_k}}[\frac{\pi_{\theta_k^{(i)}}(a|s)}{\pi_{\theta_k}(a|s)} \nabla_{\theta_k^{(i)}} \log \pi_{\theta_k^{(i)}}(a|s) A^{\theta_k}(s,a)]
\end{aligned}
$$

CHAPTER 2

ONLINE LEARNING AND ONLINE CONVEX OPTIMIZATION

2.1 Online Learning

Online Learning is a sub-domain of machine learning that has important theoretical and practical applications. In Online Learning, a learner is tasked with predicting the answer to a set of questions over a sequence of consecutive rounds. At each round t, a question x_t is taken from an instance domain $\mathcal{X}$, and the learner is required to predict an answer, p_t to this question. After the prediction is made, the correct answer y_t, from a target domain $\mathcal{Y}$ is revealed and the learner suffers a loss $l(p_t, y_t)$. The prediction p_t could belong to $\mathcal{Y}$ or a larger set, $\mathcal{D}$.

There are many special cases of Online learning that translate to popular Online learning problems. Some common ones are,

Online Classification: $\mathcal{Y} = \mathcal{D} = \{0, 1\}$, and typically the loss function is the 0-1 loss: $l(p_t, y_t) = |p_t - y_t|$.

Online Regression:

Expert's case:

The goal of an Online learning algorithm is to minimize the cumulative loss across all the rounds it has been through so far. The learner uses the information from the previous rounds to improve its prediction on present and future rounds.

The sequence of questions can be deterministic, stochastic or even adversarial. This means, for any online learning algorithm an adversary can make the cumulative loss unbounded, by simply providing an opposing answer to the algorithm's answer as the correct answer. To make learning possible, certain restrictions are imposed on the structure of the problem.

Realizability: It is assumed that the answers are generated by a target mapping $h^* : \mathcal{X} \to \mathcal{Y}$, and that h^* is taken from a fixed set, $\mathcal{H}$ called the hypothesis class. Now, for any Online learning algorithm, A, $M_A(\mathcal{H})$ is the number of mistakes A makes on a sequence of questions, labelled by some $h^* \in \mathcal{H}$. $M_A(\mathcal{H})$ is called the *mistake-bound* of A.

A relaxation from realizable assumption is that the answers are not generated by some fixed mapping h^*, but the learner is still only required to be competitive with the best fixed predictor from $\mathcal{H}$. This is the regret of an Online learning algorithm for not having followed a fixed hypothesis $h^* \in \mathcal{H}$.

$$Regret_T(h^*) = \sum_{t=1}^{T} l(p_t, y_t) - \sum_{t=1}^{T} l(h^*(x_t), y_t), \tag{2.1}$$

The regret of A with $\mathcal{H}$ is,

$$Regret_T(\mathcal{H}) = max_{h^* \in \mathcal{H}} Regret_T(h^*) \tag{2.2}$$

2.2 Online Convex Optimization

An established approach to design efficient online learning algorithm has been using convex optimization. This typically frames online learning as an online convex optimization problem as follows:

input: a convex set S for t = 1, 2. ... predict a vector $w_t \in S$ receive a convex loss function $f_t : S \mapsto \mathbb{R}$

Reframing Equation 2.2 in terms of convex optimization, we refer to a competing hypothesis here as some vector u from the convex set S.

$$Regret_T(u) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u) \tag{2.3}$$

and similarly, the regret with respect to a set of competing vectors U is,

$$Regret_T(U) = max_{u \in U} Regret_T(u) \tag{2.4}$$

As stated in the case of online learning, the set U can be same as S or different in other cases. In this work, the default setting is $U = S$ and $S = \mathbb{R}$ unless specified otherwise.

2.2.1 FoReL

Follow-the-Regularized-leader (FoReL) is a classic learning algorithm for online convex optimization, where the algorithm tries to minimize the loss on all past rounds along with a regularization term. The regularization term is used to stabilize the solution and prevent it from oscillating too much every round preventing converging to a solution.

The learning rule can be written as,

$$\forall t, w_t = argmin_{w \in S} \sum_{i=1}^{t-1} f_i(w) + R(w).$$

where $R(w)$ is the regularization term. Different regularization functions lead to different algorithms with varying regret bounds.

In the case of linear loss functions with respect to some z_t, i.e., $f_t(w) = \langle w, z_t \rangle$, and $S = \mathbb{R}^d$, if FoReL is run with l_2-norm regularization $R(w) = \frac{1}{2\eta}\|w\|_2^2$, then the learning rule can be written as,

$$w_{t+1} = -\eta \sum_{i=1}^{t} z_i = w_t - \eta z_t \tag{2.5}$$

Since, $\nabla f_t(w_t) = z_t$, this can also be written as, $w_{t+1} = w_t - \eta \nabla f_t(w_t)$. This update rule is also commonly known as Online Gradient Descent. The regret of FoReL run on Online linear optimization with a euclidean-norm regularizer is:

$$Regret_T(U) \leq BL\sqrt{2T}.$$

where $U = u : \|u\| \leq B$ and $\frac{1}{T}\sum_{t=1}^{T} \|z_t\|_2^2 \leq L^2$ with $\eta = \frac{B}{L\sqrt{2T}}$.

This can also be generalized to Convex Functions in general through linearization using the property of convex functions. For a convex set S, a convex function $f : S \mapsto \mathbb{R}$ is convex iff $\forall w \in S, \exists z$ such that,

$$\forall u \in S, f(u) \leq f(w) + \langle u - w, z \rangle \tag{2.6}$$

Following this, in Online Convex Optimization for each round t, there exists a z_t such that for all competing hypothesis u,

$$f_t(w_t) - f_t(u) \leq \langle w_t - u, z_t \rangle.$$

where $z_t \in \partial f_t(w_t)$ is a sub-gradient of f_t at w_t.

Then, for a sequence of convex loss functions $f_1, \ldots, f_T$ and vectors $w_1, \ldots, w_T$ and if for all t, $z_t \in \partial f_t(w_t)$,

$$\sum_{t=1}^{T} (f_t(w_t) - f_t(u)) \leq \sum_{t=1}^{T} (\langle w_t, z_t \rangle - \langle u, z_t \rangle) \tag{2.7}$$

This implies, the regret of an algorithm for Online Convex Optimization is upper bounded by the regret with respect to the linearization of the sequence of convex functions.

Beyond Euclidean regularization, FoReL can also be run with other regularization functions and yield similar regret bounds given that the regularization functions are strongly convex.

Definition 1 *For any σ-strongly-convex function $f : S \mapsto \mathbb{R}$ with respect to a norm $\|.\|$, for any $w \in S$,*

$$\forall z \in \partial f(w), \forall u \in S, f(u) \geq f(w) + \langle z, u - w \rangle + \frac{\sigma}{2} \|u - w\|^2. \tag{2.8}$$

Lemma 1 *For a FoReL algorithm producing a sequence of vectors $w_1, \ldots, w_T$ with a sequence of loss functions $f_1, \ldots, f_T$, for all $u \in S$,*

$$\sum_{t=1}^{T}(f_t(w_t) - f_t(u)) \leq R(u) - R(w_1) + \sum_{t=1}^{T}(f_t(w_t) - f_t(w_{t+1}))$$

2.2.2 FoReL with Strongly Convex Regularizers

From Lemma 1, the regret bound is given by,

$$\sum_{t=1}^{T}(f_t(w_t) - f_t(u)) \leq R(u) - R(w_1) + \sum_{t=1}^{T}(f_t(w_t) - f_t(w_{t+1}))$$

If f_t is L-Lipschitz with respect to some norm $\|.\|$ then,

$$f_t(w_t) - f_t(u) \leq L\|w_t - w_{t+1}\|$$

If $\|w_t - w_{t+1}\|$ is small that leads to a better regret bound. It can be shown that if the regularization function $R(w)$ is strongly convex with respect to the same norm $\|.\|$ then $\|w_t - w_{t+1}\|$ is also bounded.

For a sequence of predictions $w_1, w_2, \ldots$ of the FoReL algorithm, with a regularizer $R : S \mapsto \mathbb{R}$,

$$f_t(w_t) - f_t(w_{t+1}) \leq L_t\|w - t - w_{t+1}\| \leq \frac{L_t^2}{\sigma}.$$

if f_t is L-Lipschitz with respect to $\|.\|$ and R is σ-strongly-convex.

Theorem 2 *FoReL run on a sequence of convex functions $f_1, \ldots, f_T$ such that f_t is L_t-Lipschitz, with a σ-strongly-convex regularization function has a regret bound given by,*

$$Regret_T(u) \leq R(u) - min_{v \in S} R(v) + \frac{TL^2}{\sigma}$$

where $\frac{1}{T}\sum_{t=1}^{T} L_t^2 \leq L^2$.

To add: derived regret bounds for euclidean and entropic regularizers

2.3 Online Mirror Descent

CHAPTER 3

INTRODUCTION

This is how we cite a paper [1].

Below is the example of algorithm block.

The example of table is below.

3.1 Support of leading faction

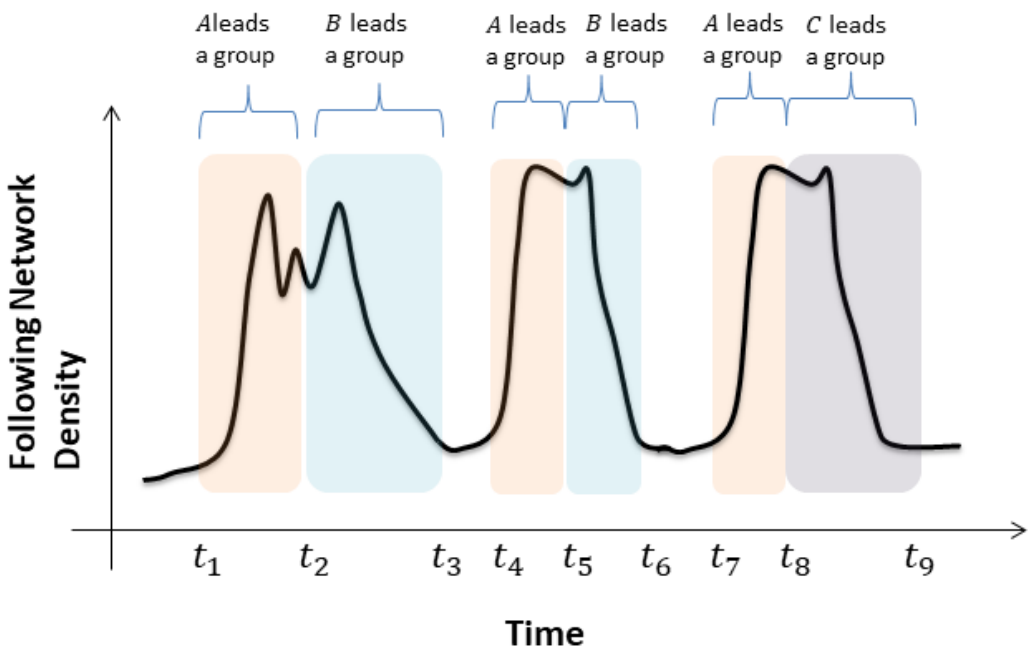

Figure 1. An example of image in thesis

Procedure 1: FindFactionsAndInitiators

input : An adjacency matrix E^* of dynamic network

output: A time series of faction sets $\mathcal{F}^*$, and a time series of initiator sets $\mathcal{L}^*$

for $i \leftarrow 1$ **to** t^* **do**

/* Get a matrix at time $t = i$ */

$E \leftarrow E^*_{t=i}$;

/* FindInitiators(E) returns all nodes which have zero outgoing degree */

$\mathcal{L} \leftarrow$FindInitiators(E) ;

$\mathcal{F} = \emptyset$;

for $l \in \mathcal{L}$ **do**

/* FindReachNodeFrom(E, l) returns all nodes which have any directed path to l */

$F_l \leftarrow$FindReachNodeFrom(E, l) ;

$\mathcal{F} = \mathcal{F} \cup \{F_l\}$

end

$\mathcal{F}^*_{t=i} = \mathcal{F}$ and $\mathcal{L}^*_{t=i} = \mathcal{L}$

end

TABLE I

Table Caption1

	Method	**Null hypothesis** H_0
Zero mean/median test	t-test	A sample has a normal distribution with zero mean and unknown variance.
	Sign test	A sample has a distribution with zero median.
	Wilcoxon signed rank test	A sample has a symmetric distribution around zero median.
Normality test	Kolmogorov-Smirnov test (KS test)	A sample comes from a normal distribution.
	Chi-square goodness-of-fit test	A sample comes from a normal distribution with a mean and variance estimated from a sample itself.
	Jarque-Bera test	A sample comes from a normal distribution with an unknown mean and variance.
	Anderson-Darling test	A sample comes from a normal distribution.

APPENDICES

Appendix A

SOME ANCILLARY STUFF

Ancillary material should be put in appendices.

Appendix B

SOME MORE ANCILLARY STUFF

CITED LITERATURE

1. Farine, D. R., Strandburg-Peshkin, A., Couzin, I. D., Berger-Wolf, T. Y., and Crofoot, M. C.: Individual variation in local interaction rules can explain emergent patterns of spatial organization in wild baboons. *Proceedings of the Royal Society of London B: Biological Sciences* , 284(1853), 2017.

VITA

NAME: NAME LASTNAME

EDUCATION: Ph.D., Computer Science, University of Illinois at Chicago, Chicago, Illinois, 2018.

M.Eng., Computer Engineering, University of Illinois at Chicago, Chicago, Illinois, 20xx.

B.Eng., Computer Engineering, University of Illinois at Chicago, Chicago, Illinois, 20xx.

ACADEMIC EXPERIENCE: Research Assistant, Computational Population Biology Lab, Department of Computer Science, University of Illinois at Chicago, xxxx - 2018.

Teaching Assistant, Department of Computer Science, University of Illinois at Chicago:

- Computer Algorithm I, Spring xxxx and Fall xxxx.
- Secure Computer Systems, Fall xxxx