1 Derivations

Online Learning is an interesting sub-domain of machine learning that has important theoretical and practical applications.

In Online Learning, a learner is tasked with learning to answer a set of questions over a sequence of consecutive rounds. At each round t, a question x_t is taken from an instance domain $\mathbb{X}$, 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 $\mathbb{Y}$ is revealed and the learner suffers a loss $l(p_t, y_t)$. The prediction p_t could belong to $\mathbb{Y}$ or a larger set, $\mathbb{D}$.

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

Online Classification: $\mathbb{Y} = \mathbb{D} = \{0, 1\}$, 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 $h^* : \mathbb{X} \to \mathbb{Y}$, and that h^* is taken from a fixed set, $\mathbb{H}$ called the hypothesis class. Now, for any Online learning algorithm, A, $M_A(\mathbb{H})$ is the number of mistakes A makes on a sequence of questions, labelled by some $h^* \in \mathbb{H}$. $M_A(\mathbb{H})$ 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 $\mathbb{H}$. This is the regret of an Online learning algorithm for not having followed a fixed hypothesis $h^* \in \mathbb{H}$.

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

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

$$Regret_T(\mathbb{H}) = max_{h^* \in \mathbb{H}} Regret_T(h^*) \tag{2}$$

1.1 FTRL

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

$$Regret_T(u) = \sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(u) \tag{3}$$

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

$$Regret_T(U) = max_{u \in U} Regret_T(u) \tag{4}$$

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.

Follow the leader: $\forall t$, $w_t = argmin_{w \in S} \sum_{i=1}^{t-1} f_i(w)$.

Follow the regularized leader: $\forall t$, $w_t = argmin_{w \in S} \sum_{i=1}^{t-1} f_i(w) + R(w)$,

where the regularization term stabilizes the solution. Different regularization functions lead to different algorithms with varying regret bounds.

A common example is FoReL with a squared l-2 norm regularization.

$w_{t+1} = -\eta \sum_{i=1}^{t} z_i = w_t - \eta z_t$ (add derivation).

This is commonly known as Online gradient descent.

Mirror Descent:

Mirror descent with entropy regularization

Mirror descent with KL Divergence regularizations

Mirror Descent Policy optimization (MDPO)

The update rule for on-policy MDPO is given by,

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

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