Machine Learning Theory (CS 6783)

Lecture 16: Online Convex Optimization/Learning

1 Online Convex Optimization Setting

For the purpose of this lecture let us modify the online learning protocol a bit (this can be done w.l.o.g.). First, Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, that is the instance space pair. Let $\mathcal{F}$ be a convex subset of a vector space. $\ell : \mathcal{F} \times \mathcal{Z} \mapsto \mathbb{R}$ is the loss function. For each $z \in \mathcal{Z}$ let $\ell(\cdot, z)$ be a convex function.

For $t = 1$ to n

Learner picks $\hat{\mathbf{y}}_t \in \mathcal{F}$

Receives instance $z_t \in \mathcal{Z}$

Suffers loss $\ell(\hat{\mathbf{y}}_t, z_t)$

End

The goal again is to minimize regret :

$$\text{Reg}_n := \frac{1}{n}\sum_{t=1}^{n} \ell(\hat{\mathbf{y}}_t, z_t) - \inf_{\mathbf{f}\in\mathcal{F}} \frac{1}{n}\sum_{t=1}^{n} \ell(\mathbf{f}, z_t)$$

2 Examples

Online Linear SVM In the case of SVM we are interested in linear predictors with constraint on the ℓ_2 norm of the predictor. In this case, $\mathcal{X} \subset \mathbb{R}^d$, $\mathcal{Y} = \{\pm 1\}$. $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $\ell(\mathbf{f}, (\mathbf{x}, y)) = \max\{0, 1 - y \cdot \mathbf{f}^\top \mathbf{x}\}$, $\mathcal{F} = \{\mathbf{f} : \|\mathbf{f}\|_2 \leq R\}$. Feel free to change hinge loss to any convex loss line square loss, logistic loss etc. Also feel free to replace the constraint $\|\mathbf{f}\|_2 \leq R$ by some other convex constraint. Regret is given by

$$\text{Reg}_n = \frac{1}{n}\sum_{t=1}^{n} \max\{0, 1 - y_t \cdot \hat{\mathbf{y}}_t^\top \mathbf{x}_t\} - \inf_{\mathbf{f}\in\mathcal{F}} \frac{1}{n}\sum_{t=1}^{n} \max\{0, 1 - y_t \cdot \mathbf{f}_t^\top \mathbf{x}_t\}$$

Regularized Linear Prediction Another set of problems that automatically fits the online convex optimization framework are regularized loss minimization problem. Here again $\mathcal{X} \subset \mathbb{R}^d$, $\mathcal{Y}$ could be say $[-1, 1]$. Now consider the case when $\ell(\mathbf{f}, (x, y)) = \phi(\mathbf{f}^\top \mathbf{x}, y) + \mathbf{R}(\mathbf{f})$. Where $\phi : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ is some loss convex in first argument. $\mathbf{R} : \mathcal{F} \mapsto \mathbb{R}$ is a convex function. As an example think of the regularized version of SVM or online ridge regression, or online Lasso.

Matrix Prediction/Collaborative Filtering Imagine we have a bunch of M users and a bunch of N products. We want to predicts ratings of users for various products in an online fashion. Eg. on round t we are given $x_t \in [M] \times [N]$ the position of the matrix we are required to predict. Learner then picks the predicted rating. Finally the true rating is revealed and learner suffers loss for predicting wrong.

$$\text{Reg}_n = \frac{1}{n}\sum_{t=1}^{n} |\hat{\mathbf{y}}_t[x_t] - y_t| - \inf_{\mathbf{f}\in\mathcal{F}} \frac{1}{n}\sum_{t=1}^{n} |\mathbf{f}[x_t] - y_t|$$

Think of $\mathcal{F}$ as a convex set where each $\mathbf{f} \in \mathcal{F}$ is an $M \times N$ matrix. Each $\hat{y}_t$ is also an $M \times N$ matrix.

2.1 Online Linear Optimization

Though we are concerned with general convex losses, it suffices (in many cases with no additional cost) to only consider online linear optimization where the loss is linear rather than general convex. The reason for this is the following. First, given any $z_1, \ldots, z_n \in \mathcal{Z}$ let $\mathbf{f}^* = \operatorname{argmin}_{\mathbf{f}\in\mathcal{F}} \sum_{t=1}^{n} \ell(\mathbf{f}, z_t)$. Now note that by convexity,

$$\begin{aligned}\sum_{t=1}^{n} \ell(\hat{\mathbf{y}}_t, z_t) - \sum_{t=1}^{n} \ell(\mathbf{f}^*, z_t) &\le \sum_{t=1}^{n} \langle \nabla \ell(\hat{\mathbf{y}}_t, z_t), \hat{\mathbf{y}}_t - \mathbf{f}^* \rangle \\ &\le \sum_{t=1}^{n} \langle \nabla \ell(\hat{\mathbf{y}}_t, z_t), \hat{\mathbf{y}}_t \rangle - \inf_{\mathbf{f}\in\mathcal{F}} \sum_{t=1}^{n} \langle \nabla \ell(\hat{\mathbf{y}}_t, z_t), \mathbf{f} \rangle\end{aligned}$$

Now let $\mathcal{D}$ be the subset of vectors defined as, $\mathcal{D} = \{\nabla(\mathbf{f}, z) : \mathbf{f} \in \mathcal{F}, z \in \mathcal{Z}$. Now since in the online learning protocol, learner picks $\hat{\mathbf{y}}_t \in \mathcal{F}$ and then adversary picks $z \in \mathcal{Z}$, we can simply think of adversary as directly picking any $\nabla_t \in \mathcal{D}$ directly and this only increases the bound. Thus,

$$\frac{1}{n}\sum_{t=1}^{n} \ell(\hat{\mathbf{y}}_t, z_t) - \inf_{\mathbf{f}\in\mathcal{F}} \frac{1}{n}\sum_{t=1}^{n} \ell(\mathbf{f}, z_t) \le \frac{1}{n}\sum_{t=1}^{n} \langle \nabla_t, \hat{\mathbf{y}}_t \rangle - \inf_{\mathbf{f}\in\mathcal{F}} \frac{1}{n}\sum_{t=1}^{n} \langle \nabla_t, \mathbf{f} \rangle$$

What the above means is that if we have an algorithm for online linear optimization, we can use it as an algorithm for online convex optimization assuming the instance received on round t is the gradients of the convex function at the point $\hat{\mathbf{y}}_t$.

3 Online Gradient Descent

In this example we assume $\mathcal{F} = \{\mathbf{f} : \|\mathbf{f}\|_2 \le R\}$ and $\mathcal{D}$ is a set whose elements all have Euclidean norm bounded by B. We consider linear loss. That is at time t the loss is $\langle \nabla_t, \hat{\mathbf{y}}_t \rangle$.

Algorithm :

$$\hat{\mathbf{y}}_{t+1} = \Pi_{\mathcal{F}}\left(\hat{\mathbf{y}}_t - \eta \nabla_t\right)$$

where $\Pi_{\mathcal{F}}$ is the Euclidean projection on to set $\mathcal{F}$ and $\eta > 0$ is referred to as step-size.

$$\Pi_F(\mathbf{f}) = \begin{cases} \mathbf{f} & \text{if } \|\mathbf{f}\|_2 \le R \\ R\frac{\mathbf{f}}{\|\mathbf{f}\|_2} & \text{otherwise} \end{cases}$$

Claim 1. *If we use the online gradient descent algorithm with* $\eta = \frac{R}{B\sqrt{n}}$ *and* $\hat{\mathbf{y}}_1 = \mathbf{0}$*, then*

$$\frac{1}{n}\sum_{t=1}^{n}\langle \nabla_t, \hat{\mathbf{y}}_t\rangle - \inf_{\mathbf{f}\in\mathcal{F}}\frac{1}{n}\sum_{t=1}^{n}\langle \nabla_t, \mathbf{f}\rangle \leq \frac{RB}{\sqrt{n}}$$

Proof. Fix any $\mathbf{f}^* \in \mathcal{F}$. Note that,

$$\|\hat{\mathbf{y}}_{t+1} - \mathbf{f}^*\|_2^2 = \|\Pi_{\mathcal{F}}(\hat{\mathbf{y}}_t - \eta\nabla_t) - \mathbf{f}^*\|_2^2 \leq \|\hat{\mathbf{y}}_t - \eta\nabla_t - \mathbf{f}^*\|_2^2 = \|\hat{\mathbf{y}}_t - \mathbf{f}^*\|_2^2 + \eta^2\|\nabla_t\|_2^2 - 2\eta\langle\nabla_t, \hat{\mathbf{y}}_t - \mathbf{f}^*\rangle$$

Thus we can conclude that

$$\langle \nabla_t, \hat{\mathbf{y}}_t - \mathbf{f}^*\rangle \leq \frac{1}{2\eta}\left(\|\hat{\mathbf{y}}_t - \mathbf{f}^*\|_2^2 - \|\hat{\mathbf{y}}_{t+1} - \mathbf{f}^*\|_2^2\right) + \frac{\eta}{2}\|\nabla_t\|_2^2$$

Summing we get,

$$\begin{aligned}\sum_{t=1}^{n}\langle \nabla_t, \hat{\mathbf{y}}_t - \mathbf{f}^*\rangle &\leq \frac{1}{2\eta}\sum_{t=1}^{n}\left(\|\hat{\mathbf{y}}_t - \mathbf{f}^*\|_2^2 - \|\hat{\mathbf{y}}_{t+1} - \mathbf{f}^*\|_2^2\right) + \frac{\eta}{2}\sum_{t=1}^{n}\|\nabla_t\|_2^2 \\ &= \frac{1}{2\eta}\left(\|\hat{\mathbf{y}}_1 - \mathbf{f}^*\|_2^2 - \|\hat{\mathbf{y}}_{n+1} - \mathbf{f}^*\|_2^2\right) + \frac{\eta}{2}nB^2 \\ &\leq \frac{1}{2\eta}R^2 + \frac{\eta}{2}nB^2\end{aligned}$$

Using the η from the claim and dividing throughout by n gives the result. □

What is the lower bound for this problem? In fact it is not hard to see that the lower bound for this problem is also $\frac{RB}{\sqrt{n}}$ at least when dimensionality is huge. To see this assume the adversary simply plays on each round vector orthogonal to current $\hat{\mathbf{y}}_t$ and also orthogonal to previous $\nabla_1, \ldots, \nabla_{t-1}$.

This algorithm is worst case optimal (in terms of computational efficiency) for SVM (even for statistical learning). Why ? Think about sample complexity and amount of time needed to read the data.

4 Online Mirror Descent

Online gradient descent doesn't even type check in general vector space! If $\mathcal{F}$ and $\mathcal{D}$ have more interesting (convex) structures, can we get better bounds ? How do we design algorithms for these problems.