CS229M/STATS214: Machine Learning Theory

Lecturer: Tengyu Ma
Scribe: Jeff Z. HaoChen and Carrie Wu

Lecture # 17
Mar 15th, 2021

17.1 Review and overview

In the last lecture, we described the "Be the leader" (BTL) algorithm for online convex optimization (OCO), a strategy that has non-positive regret. However, BTL is not a valid strategy because it uses f_t, the convex loss function for time t which the learner does not know when making the choice w_t at time t. Based on the intuition gleaned from BTL, we introduced the "Follow the regularized leader" (FTRL) algorithm which is a valid strategy. At time t, FTRL chooses w_t based on the following minimization problem:

$$w_t = \operatorname{argmin} \sum_{i=1}^{t-1} f_i(w) + \frac{1}{\eta}\phi(w). \tag{17.1}$$

We proved an upper bound on the regret achieved by FTRL: if ϕ is a 1-strongly-convex regularizer with respect to the norm $\|\cdot\|$ on the action space Ω, then the FTRL algorithm satisfies the regret guarantee

$$\text{FTRL regret} = \sum_{t=1}^{T} f_t(w_t) - \operatorname*{argmin}_{w\in\Omega} \sum_{t=1}^{T} f_t(w) \le \frac{D}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(w_t)\|_*^2, \tag{17.2}$$

where $D = \max_{w\in\Omega} \phi(w) - \min_{w\in\Omega} \phi(w)$. Furthermore, if for all t and w we have the uniform bound $\|\nabla f_t(w)\|_* \le G$, then the above implies that the regret is upper bounded by $2\sqrt{DG} \times \sqrt{T}$.

$$\text{FTRL regret} \le 2\sqrt{DG} \times \sqrt{T}. \tag{17.3}$$

We then demonstrated how FTRL applies to online linear regression and the expert problem.

In this lecture we improve on the regret bound for the expert problem by using a different regularizer in the FTRL algorithm. We also show how we can reduce any online convex optimization problem to an online *linear* optimization problem without losing anything in the regret bound. Finally, we return to our discussion on algorithmic regularization, demonstrating how this phenomenon manifests in gradient descent for classification.

17.2 More on FTRL and the expert problem

Last lecture, we analyzed the use of FTRL with ℓ_2-regularization (i.e. $\phi(w) = \frac{1}{2}\|w\|^2$) on the expert problem and proved that FTRL gave an $O(\sqrt{NT})$ regret guarantee. Here, we show that if we change our regularization, we can get a better regret guarantee which is logarithmic in N, i.e., the regret is $O(\sqrt{(logN)\cdot T})$.

The new regularizer we choose is the *(negative) entropy regularizer*:

$$\phi(p) = -H(p) = \sum_{j=1}^{N} p(j) \log p(j), \tag{17.4}$$

where $p \in \Delta(N)$ is in the set of distributions over $[N]$. We first introduce the following nice property of this regularizer:

Lemma 17.1. *$\phi(p)$ defined above is 1-strongly convex with respective to the ℓ_1 norm $\|\cdot\|_1$.*

Proof. By definition of strong convexity, we need to show that for all $p, q \in \Delta(N)$,

$$\phi(p) - \phi(q) - \langle \nabla\phi(q), p - q\rangle \geq \frac{1}{2}\|p - q\|_1^2. \tag{17.5}$$

From direct computation, we know the gradient of $\phi(q)$ is

$$\nabla\phi(q) = \begin{bmatrix} 1 + \log q(1) \\ \cdots \\ 1 + \log q(N) \end{bmatrix}. \tag{17.6}$$

Plugging this into the LHS of (17.5), we get

$$\begin{aligned}
&\phi(p) - \phi(q) - \langle \nabla\phi(q), p - q\rangle && (17.7)\\
=&\sum_{j=1}^{N} p(j)\log p(j) - \sum_{j=1}^{N} q(j)\log q(j) - \sum_{j=1}^{N} (1 + \log q(j))\,(p(j) - q(j)) && (17.8)\\
=&\sum_{j=1}^{N} p(j)\log p(j) - \sum_{j=1}^{N} p(j)\log q(j) - \sum_{j=1}^{N} (p(j) - q(j)) && (17.9)\\
=&\sum_{j=1}^{N} p(j)\log\frac{p(j)}{q(j)} && (17.10)\\
=&KL(p||q), && (17.11)
\end{aligned}$$

where $KL(p||q)$ is the KL-divergence between p and q. (We used the fact that $\sum_{j=1}^{N} p(j) = \sum_{j=1}^{N} q(j) = 1$ to get (17.10).) Finally, we finish the proof by applying Pinsker's inequality: $KL(p||q) \geq \frac{1}{2}\|p - q\|_1^2$.

□

Hence, ϕ is a satisfies the condition on the regularizer for our FTRL regret guarantee. To obtain the regret bound (17.3), we also need to bound $D = \sup\phi(p) - \inf\phi(p)$ and $G = \sup\|\nabla f_t(w)\|_\infty$ (since $\|\cdot\|_\infty$ is the dual norm of $\|\cdot\|_1$). Since negative entropy is always non-positive and (positive) entropy is always bounded above by $\log N$, we bound D with

$$D = \sup\phi(p) - \inf\phi(p) \leq -\inf\phi(p) = -\inf(-H(p)) = \sup(H(p)) \leq \log N, \tag{17.12}$$

and we bound G with

$$G = \|\nabla f_t(w)\|_\infty = \|l_t\|_\infty \leq 1. \tag{17.13}$$

Plugging these two into the regret bound (17.3) we get bound $O(\sqrt{(\log N) \cdot T})$.

Thus far, we have looked at FTRL and the expert problem abstractly: at each time t we choose action p_t based on the update

$$p_t = \operatorname*{argmin}_{p \in \Delta(N)} \sum_{i=1}^{t+1} f_t(p) - \frac{1}{\eta} H(p). \tag{17.14}$$

Can we get an exact analytical solution for p_t? Since we are minimizing a convex function, we can call some off-the-shelf convex optimization algorithm to solve this at each step. Another way is to write down the KKT conditions and solve that set of equations. Instead, we will show that there exists much simpler ways to solve this update. In particular, we will be using the *Gibbs variational principle*, which is essentially the KKT conditions under the hood.

Lemma 17.2 (Gibbs variational principle)**.** *Let ν, μ be probability distributions on $[N]$. Then*

$$\sup_{\nu} \left(\mathbb{E}_\nu[f] - KL(\nu||\mu) \right) = \log \mathbb{E}_\mu \left[e^f \right], \tag{17.15}$$

where $\mathbb{E}_\nu[f] = \mathbb{E}_{x \sim \nu}[f(x)] = \langle v, f \rangle$ and $\mathbb{E}_\mu \left[e^f \right] = \mathbb{E}_{x \sim \mu} \left[e^{f(x)} \right]$. Moreover, the optimal solution is attained at

$$\nu(x) \propto \mu(x) \cdot e^{f(x)}. \tag{17.16}$$

Intuitively, Lemma 17.2 says that taking the supremum over distributions μ of a linear function plus the KL divergence as the regularizer will give us the same distribution as exponentiating f.

If we take μ to be the uniform distribution on $[N]$ and replace f with $-f$ in Lemma 17.2, we get the following corollary:

Corollary 17.3. Let ν be a probability distribution. Then, $\mathbb{E}_\nu[f] - H(\nu)$ is minimized at $\nu(x) \propto e^{-f(x)}$.

Proof. When μ is uniform distribution, we have

$$\begin{aligned} KL(\nu||\mu) &= \sum_x \nu(x) \log \frac{\nu(x)}{\mu(x)} && (17.17) \\ &= \log N - \sum_x \nu(x) \log \frac{1}{\nu(x)} && (17.18) \\ &= \log N - H(\nu). && (17.19) \end{aligned}$$

So $\sup_\nu \left(\mathbb{E}_\nu[-f] - KL(\nu||\mu) \right) = -\inf_\nu \left(\mathbb{E}_\nu[f] - H(\nu) + \log N \right)$. This means that the value of ν that attains the infimum of $\mathbb{E}_\nu[f] - H(\nu)$ is the same ν attaining the supremum of $\mathbb{E}_\nu[-f] - KL(\nu||\mu)$, which by Lemma 17.2 is proportional to $e^{-f(x)}$. □

We now apply the Gibbs variational principle to the expert problem. Notice that our FTRL update for the expert problem at time t can be written as

$$\operatorname*{argmin}_{p_t \in \Delta(N)} \left\langle \sum_{i=1}^{t-1} l_i, p_t \right\rangle - \frac{1}{\eta} H(p_t) = \operatorname*{argmin}_{p_t \in \Delta(N)} \left\langle \eta \sum_{i=1}^{t-1} l_i, p_t \right\rangle - H(p_t), \tag{17.20}$$

where l_i is the vector of expert losses at time i. Letting $f = \eta \sum_{i=1}^{t-1} l_i$, we know from Corollary 17.3 that the minimizer is attained at $p_t \propto \exp\left(-\eta \sum_{i=1}^{t-1} l_i\right)$, or equivalently,

$$p_t(j) = \frac{\exp(-\eta L_t(j))}{\sum_{k=1}^{N} \exp(-\eta L_t(k))}, \tag{17.21}$$

where $L_t = \sum_{i=1}^{t-1} l_i$ is the cumulative loss vector. Basically, solving the expert problem is to look a the historical loss of each expert and take softmax to find the probability distribution of how much to trust each expert.

This algorithm is also called the "Multiplicative Weights Update", which has been studied before online learning framework became popular [AHK05, FS97, LW94]. One way of doing multiplicative weights update is the following: Let $\tilde{p}_t$ be the unnormalized distribution that we keep track of. At each time step t, for each expert j, we look at $l_{t-1}(j)$. if $l_{t-1}(j) = 1$, i.e. the expert made a mistake at the previous time step, we update $\tilde{p}_t(j) = \tilde{p}_{t-1}(j) \cdot \exp(-\eta)$; otherwise we make no change. We then get a distribution by normalizing $\tilde{p}_t$:

$$p_t = \frac{\tilde{p}_t}{\|\tilde{p}_t\|_1}. \tag{17.22}$$

17.3 Convex to linear reduction

In the previous section we considered the expert problem, where the loss function is a *linear* function of the parameters. At first glance we may think this is a very restrictive constraint for online convex optimization. However, as we will see in this section, we can always assume f_t to be linear in online convex optimization without loss of generality. That means that for online learning, the linear case is the hardest one.

More concretely, assume we have an algorithm $\mathcal{A}$ that solves the linear case. Given any online convex optimization, we will build an algorithm $\mathcal{A}'$ which invokes algorithm $\mathcal{A}$ in the following fashion: for $t = 1, \ldots, T$,

1. The learner invoke $\mathcal{A}$ to get output action $w_t \in \Omega$.
2. The environment gives the learner the loss function $f_t(\cdot)$.
3. The learner construct a linear function $g_t(w) = \langle \nabla f_t(w_t), w \rangle$, which is the local linear approximation of f at w. (Technically the local linear approximation of f and w is $\langle \nabla f_t(w_t), w - w_t \rangle$, but we drop the w_t shift for convenience.)
4. The learner feeds $g_t(\cdot)$ to algorithm $\mathcal{A}$ as the loss function.

We have the following informal claim[1]:

Proposition 17.3.1 (Informal)**.** If a deterministic algorithm $\mathcal{A}$ has regret no more than $\gamma(T)$ for linear cases for some function $\gamma(\cdot)$, then $\mathcal{A}'$ stated above has regret no more than $\gamma(T)$ for convex functions.

[1]For rigorous proof, we need additional assumptions and restrictions on f_t, g_t.

Proof. For all $w \in \Omega$, the regret guarantee on $\mathcal{A}$ tells us that

$$\sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} g_t(w) \leq \gamma(T). \tag{17.23}$$

Since f_t is convex, we also know that

$$g_t(w_t) - g_t(w) = \langle \nabla f_t(w_t), w_t - w \rangle \geq f_t(w_t) - f_t(w). \tag{17.24}$$

Therefore, for all $w \in \Omega$,

$$\sum_{t=1}^{T} f_t(w_t) - \sum_{t=1}^{T} f_t(w) \leq \sum_{t=1}^{T} g_t(w_t) - \sum_{t=1}^{T} g_t(w) \tag{17.25}$$

$$\leq \gamma(T). \tag{17.26}$$

Hence, the regret for $\mathcal{A}'$ is upper bounded by $\gamma(T)$ as well. □

17.3.1 Online gradient descent

In this section we combine the FTRL framework with ℓ_2-regularization and the online-to-linear reduction. The resulting algorithm is *online gradient descent*.

Concretely, given any convex online optimization problem, we first do the online-to-linear reduction, then we use FTRL with ℓ_2 regularization ($\phi(w) = \frac{1}{2}\|w\|_2^2$) to solve the resulting linear case. This gives us the following update:

$$w_t = \operatorname{argmin} \sum_{i=1}^{t-1} g_i(w) + \frac{1}{\eta}\|w\|_2^2 \tag{17.27}$$

$$= \operatorname*{argmin}_{w \in \Omega} \sum_{i=1}^{t-1} \langle \nabla f_i(w_i), w \rangle + \frac{1}{\eta}\|w\|_2^2 \tag{17.28}$$

$$= \Pi_\Omega \left(-\eta \cdot \sum_{i=1}^{t-1} \nabla f_i(w_i) \right), \tag{17.29}$$

where $\Pi_\Omega(\cdot)$ is the projection operator onto the set Ω.The last equality is because for any vector a, we have

$$\operatorname*{argmin}_{w \in \Omega} \langle a, w \rangle + \frac{1}{\eta}\|w\|_2^2 = \operatorname*{argmin}_{w \in \Omega} \frac{1}{2\eta}\|w + \eta a\|_2^2 - \eta\|a\|_2^2 \tag{17.30}$$

$$= \operatorname*{argmin}_{w \in \Omega} \|w + \eta a\|_2^2 \tag{17.31}$$

$$= \operatorname*{argmin}_{w \in \Omega} \|w - (-\eta a)\|_2^2 \tag{17.32}$$

$$= \Pi_\Omega(-\eta a). \tag{17.33}$$

Intuitively, we can think of this algorithm as gradient descent with "lazy" projection:

$$u_t = u_{t-1} - \eta \nabla f_{t-1}(w_{t-1}), \tag{17.34}$$

$$w_t = \Pi_\Omega(u_t). \tag{17.35}$$

Similarly, we can define gradient descent with "eager" projection (which can get similar regret bounds):

$$u_t = w_{t-1} - \eta \nabla f_{t-1}(w_{t-1}), \tag{17.36}$$

$$w_t = \Pi_\Omega(u_t). \tag{17.37}$$

This concludes our discussion of online learning in this course.

17.4 Algorithmic regularization for classification

We return now to the subject of algorithmic regularization. In the previous lectures we have discussed the following two cases of algorithmic regularization for the regression problem (assuming overparameterization, i.e., $n \ll d$):

- For linear model $y = \langle x, \beta \rangle$ where $\beta \in \mathbb{R}^d$ is the parameter and $x \in \mathbb{R}^d$ is the data input, if we initialize at $\beta = 0$, then gradient descent converges to a minimum norm solution.
- For quadratically parameterized model $y = \langle x, \beta \odot \beta \rangle$ where β is the parameter and x is the data input, if we initialize at some small β, then gradient descent converges to a sparse ground truth $\beta^\star$.

In this section, we will discuss algorithmic regularization for classification problem. In particular, we consider binary classification with logistic loss. Let $\{(x_i, y_i)\}_{i=1}^n$ be a separable dataset with $y_i \in \{\pm 1\}$, $x_i \in \mathbb{R}^d$. We have a linear model $h_w(x) = w^\top x$, and we minimize the empirical logistic loss

$$\widehat{L}(w) = \frac{1}{n} \sum_{i=1}^n \ell(y_i h_w(x_i)) \tag{17.38}$$

$$= \frac{1}{n} \sum_{i=1}^n \ell(y_i w^\top x_i), \tag{17.39}$$

where $\ell(t) = \log(1 + \exp(-t))$ is the logistic loss.

In order to observe algorithmic regularization, we need to ensure that there exist multiple global minima for this setup. This is the case here: because the dataset is linearly separable, there exists some w such that $y_i w^\top x_i > 0$ for all i. Clearly any w' in a small neighborhood of w also classifies all the data correctly; hence, there exists an infinite number of separating classifiers $\overline{w}$ with unit norm. For any of these $\overline{w}$, note that $\widehat{L}(\alpha \overline{w}) \to 0$ as $\alpha \to \infty$, hence intuitively all of "$\infty \overline{w}$" classifiers are global minima.

Having shown the existence of multiple global minima, we now show that gradient descent will actually converge to the solution which maximizes the *margin*. We first define the *normalized margin* for a separating classifier w as

$$\gamma(w) = \frac{\min_{i \in [n]} y_i w^\top x_i}{\|w\|_2}. \tag{17.40}$$

We call $\overline{\gamma} = \max_w \gamma(w)$ the *max margin*. Now we are ready to state the theorem:

Theorem 17.4 ([SHN+18])**.** *Gradient descent with iterates w_t converges to the direction of a max-margin solution:*

$$\gamma(w_t) \to \overline{\gamma} \quad as \quad t \to \infty. \tag{17.41}$$

In other words, gradient descent on logistic loss is equivalent to the SVM.[2]

Here, we provide the intuition behind the proof. The proof of this theorem follows these steps:

1. By standard convex optimization arguments, $\widehat{L}(w_t) \to 0$ as $t \to \infty$.
2. For sufficiently large t, $\|w_t\|_2 \to \infty$.
3. For sufficiently large t, w_t will separate the data (since the loss goes to 0).
4. As $z \to \infty$, $l(z) = \log(1 + \exp(-z)) \approx \exp(-z)$ (i.e. logistic loss is similar to exponential loss).
5. When $\|w\|_2 = q$ is big, the loss $\widehat{L}(w)$ mainly depends on supporting data $\{(x_i, y_i) : y_i \overline{w}^\top x_i = \gamma(w)\}$.

To see the last bullet point: letting $\overline{w} = w/\|w\|_2$, we notice that

$$\widehat{L} = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i w^\top x_i) \tag{17.42}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \exp\left(-q y_i \overline{w}^\top x_i\right) \tag{17.43}$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \exp\left(-q y_i \overline{w}^\top x_i\right) 1\left[y_i \overline{w}^\top x_i = \gamma(w)\right] \tag{17.44}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \exp\left(-q\gamma(w)\right) 1\left[y_i \overline{w}^\top x_i = \gamma(w)\right]. \tag{17.45}$$

Here the first approximation (17.43) is because of the logistic loss vs. exponential loss approximation, while the second approximation (17.44) is because for any data x_i, y_i that is not a support vector, i.e.

$$\overline{w}^\top x_i y_i \geq \gamma(w) + \epsilon, \tag{17.46}$$

for $\epsilon > 0$, then

$$\exp(-q y_i \overline{w}^\top x_i) \leq \exp(-q\gamma(w)) \exp(-q\epsilon), \tag{17.47}$$

and as $q \to \infty$ the term $\exp(-q\epsilon) \to 0$, making such terms negligible.

In conclusion, minimizing the (approximate) loss (17.45) is (informally) equivalent to maximizing the margin. (Note that if you examine (17.45), there are actually two ways to make the loss small: maximizing the margin or making q large. The rigorous proof shows that when q is large, the margin is already very close to the max margin. These are technical details that we will not concern ourselves with.)

[2]This result is still very limited it only works without regularization, and one needs to run gradient descent for a long time before this convergence in direction happens. Also, SVM is not always the best possible solution.

Bibliography

[AHK05] Sanjeev Arora, Elad Hazan, and Satyen Kale, *Fast algorithms for approximate semidefinite programming using the multiplicative weights update method*, 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05), IEEE, 2005, pp. 339–348.

[FS97] Yoav Freund and Robert E Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, Journal of computer and system sciences **55** (1997), no. 1, 119–139.

[LW94] Nick Littlestone and Manfred K Warmuth, *The weighted majority algorithm*, Information and computation **108** (1994), no. 2, 212–261.

[SHN+18] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro, *The implicit bias of gradient descent on separable data*, The Journal of Machine Learning Research **19** (2018), no. 1, 2822–2878.