Statistical Techniques in Robotics (16-831, S22) Lecture #07 (Wednesday, February 9)

Follow the Regularized Leader

Lecturer: Kris Kitani *Scribes: Jinkun Liu, Thomas Xu*

1 Review

In the last lecture, we covered three main topics. First, the Winnow algorithm was discussed, which is an online linear classification algorithm. Second, online convex optimization was introduced, including the related concepts used to define an convex optimization problem. Finally, the Follow the Leader algorithm was introduced as a generic method for solving optimization problems.

1.1 Winnow Algorithm

The Winnow algorithm is an online linear classification algorithm that operates on binary features as inputs. The key assumption of the Winnow algorithm is that only a certain subset of the features are relevant for classification. Another key feature of Winnow algorithm is that it uses multiplicative weight updates, unlike the Perceptron algorithm which uses additive updates. Below is the pseudo code for this algorithm:

Algorithm 1 Winnow algorithm

```
$\mathbf{w}^{(1)} \leftarrow \{1, ..., 1\}$                                ▷ Initialize weights
for $t = 1, \cdots, T$ do
    Receive ($\mathbf{x}^{(t)} \in \{0,1\}^N$)                       ▷ Receive feature vector
    $\hat{y}^{(t)} = \mathbf{1}[\langle \mathbf{w}^{(t)}, \mathbf{x}^{(t)} \rangle > \Theta]$        ▷ Predict label
    Receive ($y^{(t)} \in \{0,1\}$)                               ▷ Receive true label
    $w_n^{(t+1)} \leftarrow w_n^{(t)}(1+\beta)^{(y^{(t)}-\hat{y}^{(t)})\cdot x_n^{(t)}}$          ▷ Update weights
end for
```

Here, Θ (often assumed equal to N) is the threshold that is used to define the decision boundary (hyperplane), and β is a constant parameter that affects the magnitude of weight updates. A typical value is $\beta = 1$.

Assuming $\beta = 1$, the total number of mistakes M made by the learner using the Winnow algorithm is upper-bounded as:

$$M < 2 + 3k(\log_2 N + 1) \tag{1}$$

where N is the total number of input features and k is the number of relevant features.

1.2 Online Convex Optimization

Online convex optimization serves as a generalized framework of many online tasks. In this framework, an online convex optimizer interacts with nature by producing predictions using its parameters $\boldsymbol{w}^{(t)} \in \mathcal{S}$, which are updated using the loss functions $l^{(t)}$ received from nature. Here, $\mathcal{S}$ must be a convex set and $l^{(t)}$ must be a convex function, both of which are defined below.

Convex Set A set $\mathcal{S}$ is considered a convex set if for all $\boldsymbol{w}, \boldsymbol{v} \in \mathcal{S}$:

$$\alpha \boldsymbol{w} + (1-\alpha)\boldsymbol{v} \in \mathcal{S} \qquad \forall \alpha \in [0,1] \tag{2}$$

Convex Function A function $f : \mathcal{S} \to \mathbb{R}$ is considered a convex function if for all $\boldsymbol{w}, \boldsymbol{v} \in \mathcal{S}$:

$$f(\alpha \boldsymbol{w} + (1-\alpha)\boldsymbol{v}) \leq \alpha f(\boldsymbol{w}) + (1-\alpha) f(\boldsymbol{v}) \qquad \forall \alpha \in [0,1] \tag{3}$$

Lipschitz Continuity A function $f(\cdot)$ is "L-Lipschitz" over set $\mathcal{S}$ with respect to a norm $\|\cdot\|$ if:

$$|f(\boldsymbol{u}) - f(\boldsymbol{w})| \leq L||\boldsymbol{u} - \boldsymbol{w}|| \qquad \forall \boldsymbol{u}, \boldsymbol{w} \in \mathcal{S} \tag{4}$$

For a problem to be solved with convex optimization, the function and solution space must both be convex; in addition, the function must be either convex-Lipschitz-bounded or convex-smooth-bounded.

Convexification Non-convex problems can be converted into convex ones. One method is convexification by **randomization**, such as in the case of converting WMA to RWMA. Another method is convexification by **surrogate loss**, which replaces the original non-convex loss function with a surrogate loss function that upper-bounds the original loss function, and is itself a convex function.

1.3 Follow the Leader

Follow the Leader (FTL) is a generic algorithm for online (not necessarily convex) optimization problems. Its main idea is that the learner should go by the best choice seen so far. The general pseudocode for FTL is as seen below:

Algorithm 2 Follow the Leader

```
for $t = 1, 2, \cdots, T$ do
    $\boldsymbol{w}^{(t)} = \arg\min_{\boldsymbol{w} \in W} \sum_{i=1}^{t-1} f^{(i)}(\boldsymbol{w})$        ▷ Update parameter
    RECEIVE($f^{(t)} : W \to \mathbb{R}$)        ▷ Receive loss
end for
```

The performance of FTL is affected by the loss function used. Note that W is a convex set. With a linear loss of the form $f^{(t)} = \boldsymbol{w} \cdot \boldsymbol{z}^{(t)}$, $O(T)$ loss and regret is possible in an adversarial scenario; the quadratic loss $f^{(t)} = \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{z}^{(t)}\|_2^2$ can achieve $O(\sqrt{T})$ regret, as we will derive in the main summary.

2 Summary

2.1 Follow the Leader Regret Bound

To derive the generic FTL's regret bound (without any assumptions on the form of the loss function), we upper-bound the regret using the technique **one-step look ahead cheater**. The regret of FTL is of the form:

$$R(\boldsymbol{u}) = \sum_t [f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{u})] \tag{5}$$

where $\boldsymbol{u}$ is any set of parameters, $\boldsymbol{w}^{(t)}$ is learner's set of parameters at time t, and $f^{(t)}$ is the loss function returned by nature. We can upper-bound the performance of $\boldsymbol{u}$ by setting it to a one-step look ahead cheater that has access to the loss function at time t, $\boldsymbol{w}^{(t+1)}$:

$$\sum_t [f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{u})] \leq \sum_t [f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)})] \tag{6}$$

Subtracting $f^{(t)}(\boldsymbol{w}^{(t)})$ from both sides, we get:

$$\sum_{t=1}^{T} f^{(t)}(\boldsymbol{w}^{(t+1)}) \leq \sum_{t=1}^{T} f^{(t)}(\boldsymbol{u}) \quad \forall \boldsymbol{u} \tag{7}$$

This relationship indicates that the loss of a series of one-step look ahead cheaters is equal to or smaller than the loss of any other set of parameters $\boldsymbol{u}$. To prove (7), we use proof by induction and assume it's true for $T-1$, and will prove that it holds true for T:

$$\sum_{t=1}^{T-1} f^{(t)}(\boldsymbol{w}^{(t+1)}) \leq \sum_{t=1}^{T-1} f^{(t)}(\boldsymbol{u}) \tag{8}$$

Adding $f^{(t)}(\boldsymbol{w}^{(T+1)})$, the loss of the one-step look ahead cheater for the next time step, to both sides:

$$\sum_{t=1}^{T} f^{(t)}(\boldsymbol{w}^{(t+1)}) \leq f^{(T)}(\boldsymbol{w}^{(T+1)}) + \sum_{t=1}^{T-1} f^{(t)}(\boldsymbol{u}) \tag{9}$$

Note that the added term has been collapsed into the summation on the left side of 9, but not on the right side due the different parameters. However, since we assume that (8) and by extension (9) is true for all $\boldsymbol{u}$, we can simply substitute $\boldsymbol{u} = \boldsymbol{w}^{(T+1)}$:

$$\sum_{t=1}^{T} f^{(t)}(\boldsymbol{w}^{(t+1)}) \leq \sum_{t=1}^{T} f^{(t)}(\boldsymbol{w}^{(T+1)}) \tag{10}$$

In (10), the left side is the loss for always using the one-step look ahead cheater, and the right side is the loss for using the single cheater at the end of the sequence for all time steps. Recall the definition of the minimizer: $\boldsymbol{w}^{(T+1)} = \mathrm{argmin}_{\boldsymbol{u} \in S} \sum_{t=1}^{T} f^{(t)}(\boldsymbol{u})$. This allows us to upper-bound the right hand side of (10) by all $\boldsymbol{u}$:

$$\sum_{t=1}^{T} f^{(t)}(\boldsymbol{w}^{(t+1)}) \leq \sum_{t=1}^{T} f^{(t)}(\boldsymbol{u}) \tag{11}$$

Multiply both sides by -1 and add $f^{(t)}(\boldsymbol{w}^{(t)})$ to the summation on both sides, we finally obtain the regret bound of the generic FTL algorithm proposed in (6):

$$\begin{aligned} R(\boldsymbol{u}) &= \sum_t [f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{u})] && (12) \\ &\leq \sum_t [f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)})] && (13) \end{aligned}$$

2.1.1 Follow the Leader (Quadratic Loss) Regret Bound

To derive the regret bound for FTL with quadratic loss, we simply substitute the specific quadratic loss function into the generic regret bound derived for generic FTL above (13). Note that we assume points $\boldsymbol{z}$ are bounded such that $||\boldsymbol{z}||_2^2 \leq L$. Recall that the quadratic loss is given by:

$$f^{(t)}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{z}^{(t)}\|_2^2 \tag{14}$$

This leads to the following expression for a single time step:

$$f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)}) = \frac{1}{2}||\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}||_2^2 - \frac{1}{2}||\boldsymbol{w}^{(t+1)} - \boldsymbol{z}^{(t)}||_2^2 \tag{15}$$

Additionally, recall that the minimizer of this quadratic loss is simply the mean:

$$\boldsymbol{w}^{(t)} = \frac{1}{t-1}\sum_{i=1}^{t-1} \boldsymbol{z}^{(i)} \tag{16}$$

Knowing this, we can rewrite the weight update at each time step in an incremental form:

$$\boldsymbol{w}^{(t+1)} = (1 - \frac{1}{t})\boldsymbol{w}^{(t)} + (\frac{1}{t})\boldsymbol{z}^{(t)} \tag{17}$$

Substituting this incremental form into (15) and rearranging yields:

$$\begin{aligned} f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)}) &= \frac{1}{2}||\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}||_2^2 - \frac{1}{2}||(1 - \frac{1}{t})\boldsymbol{w}^{(t)} + (\frac{1}{t})\boldsymbol{z}^{(t)} - \boldsymbol{z}^{(t)}||_2^2 && (18) \\ &= \frac{1}{2}\Big(1 - (1 - \frac{1}{t})^2\Big)||\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}||_2^2) && (19) \end{aligned}$$

Note that the $(1 - \frac{1}{t})$ term becomes squared after it is factored out of the squared L2 norm. Next, we make use of the following known inequality (illustrated in Figure 1):

$$\frac{1}{2}\Big(1 - (1 - \frac{1}{t})^2\Big) \leq \frac{1}{t} \tag{20}$$

Applying this known inequality to (19) yields:

$$f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)}) \leq \Big(\frac{1}{t}\Big)||\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}||_2^2 \tag{21}$$

Next, we make use of the following assumptions:

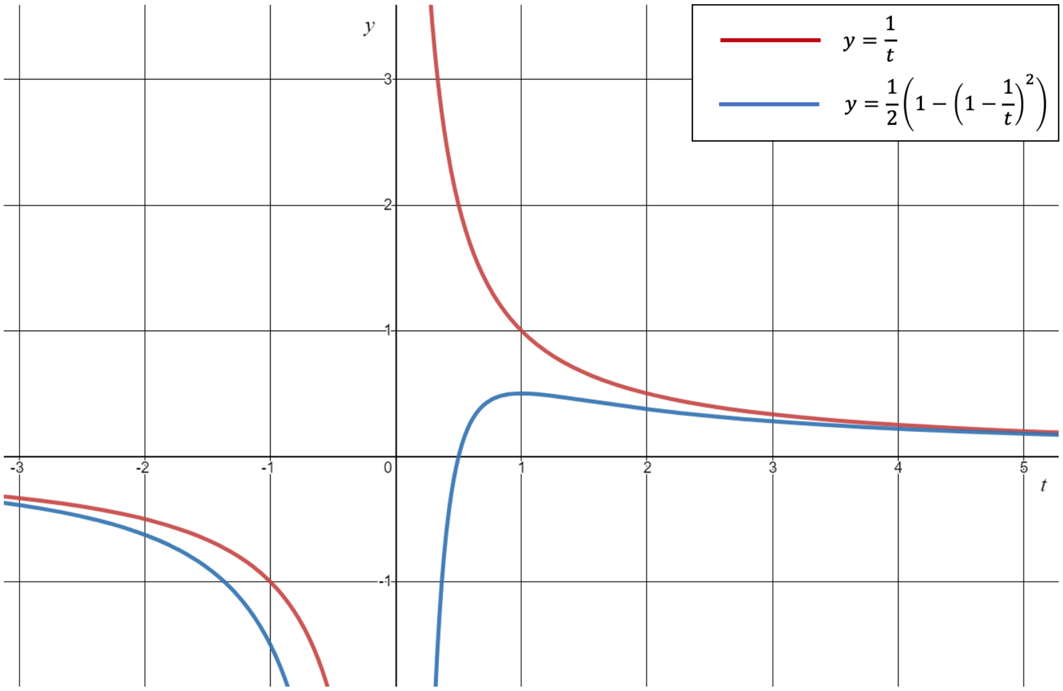

Figure 1: The known inequality used in (20).

1. $L = \max_t ||\boldsymbol{z}^{(t)}||$
2. $||\boldsymbol{w}^{(t)}|| \leq L$
3. $||\boldsymbol{w}^{(t)} - \boldsymbol{z}^{(t)}|| \leq 2L$

The first assumption states that L is a constant value and comes from our earlier assumption that each point z is bounded. The second assumption states that the average value of the points (see (16)) is less than or equal to the max value of the individual points. The third assumption results from the first two; $w^{(t)}$ and $z^{(t)}$ can be thought of as two vectors with max length L, and from the triangle inequality, the line that connects their ends can have max length $2L$.

Applying these assumptions yields the following regret bound for a single time step:

$$f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)}) \leq \left(\frac{1}{t}\right)4L^2 \tag{22}$$

To obtain the regret bound over all time steps, we substitute (22) into (13):

$$R(\boldsymbol{u}) \leq \sum_t [f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)})] \leq 4L^2 \sum_t (\frac{1}{t}) \tag{23}$$

Additionally, we make use of the following known inequality:

$$\sum_{t=1}^{T} (\frac{1}{t}) \leq 1 + \int_1^T (\frac{1}{t})dt = \log(T) + 1 \tag{24}$$

Combining (22), (23), and (24) yields the final expression for the regret bound for FTL with quadratic loss:

$$\text{Regret} \leq 4L^2\Big(\log(T) + 1\Big) \tag{25}$$

Note that this grows sub-linearly in time ($O(\log(T))$), making FTL with quadratic loss a no-regret algorithm. Another observation for quadratic loss is that since the weight updates are done via averaging, the weights do not change drastically in one time step; this ensures smoother weight udpates and the stability of the algorithm.

2.2 Follow the Regularized Leader

Compared to FTL, Follow the Regularized Leader (FTRL) adds an explicit regularization term ψ to the parameter update step. One benefit of the regularization is that it improves the stability of the chosen parameters $\boldsymbol{w}$ ($\boldsymbol{w} \in S$ where S is a convex set).

The generic algorithm for FTRL is shown below.

Algorithm 3 Follow the Regularized Leader

```
for $t = 1, 2, \cdots, T$ do
    $\boldsymbol{w}^{(t)} = \arg\min_{\boldsymbol{w} \in S} \sum_{i=1}^{t-1} f^{(i)}(\boldsymbol{w}) + \psi(\boldsymbol{w})$    ▷ Update parameter with regularization
    Receive($f^{(t)} : S \to \mathbb{R}$)    ▷ Receive loss
end for
```

Here, $\psi(\boldsymbol{w})$ represents the regularization term, and is a scalar-valued function: $\psi : S \to \mathbb{R}$. FTL can be seen as a special case of FTRL where $\psi = 0$.

We will use this observation to derive FTRL's **regret bound** using the already derived FTL regret bound. In particular, we can treat the regularization term $\psi(\boldsymbol{w})$ as the loss function $f^{(0)}(\boldsymbol{w})$ at $t = 0$. This is because at $t = 0$, no loss function has been received from nature, so the minimizer will set $\boldsymbol{w}$ to the value that minimizes $\psi(\boldsymbol{w})$ only:

$$\boldsymbol{w}^{(t)} = \arg\min_{\boldsymbol{w} \in S} \sum_{i=1}^{t-1} f^{(i)}(\boldsymbol{w}) + \psi(\boldsymbol{w}) \tag{26}$$

$$= \arg\min_{\boldsymbol{w} \in S} \sum_{i=1}^{t-1} f^{(i)}(\boldsymbol{w}) + f^{(0)}(\boldsymbol{w}) \tag{27}$$

Recall the regret bound for generic FTL from (13):

$$R(\boldsymbol{u}) = \sum_t [f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{u})] \leq \sum_t [f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)})] \tag{28}$$

Add the loss function $f^{(0)}$ at $t = 0$ to both sides:

$$[f^{(0)}(\boldsymbol{w}^{(0)}) - f^{(0)}(\boldsymbol{u})] + \sum_{i=1}^{T}[f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{u})] \tag{29}$$

$$\leq [f^{(0)}(\boldsymbol{w}^{(0)}) - f^{(0)}(\boldsymbol{w}^{(1)})] + \sum_{i=1}^{T}[f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)})] \tag{30}$$

Substitute $f^{(0)}$ with ψ, which is essentially a change in notation:

$$[\psi(\boldsymbol{w}^{(0)}) - \psi(\boldsymbol{u})] + \sum_{i=1}^{T}[f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{u})] \tag{31}$$

$$\leq [\psi(\boldsymbol{w}^{(0)}) - \psi(\boldsymbol{w}^{(1)})] + \sum_{i=1}^{T}[f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)})] \tag{32}$$

Subtract $[\psi(\boldsymbol{w}^{(0)}) - \psi(\boldsymbol{u})]$ from both sides yields the regret bound for the generic FTRL algorithm:

$$R(\boldsymbol{u}) \leq \left[\psi(\boldsymbol{u}) - \psi(\boldsymbol{w}^{(1)}\right] + \sum_{t=1}^{T}\left[f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)})\right] \tag{33}$$

This regret bound will be used as a intermediate step for the derivation of regret bound of FTRL with linear loss and quadratic regularization.

2.2.1 FTRL with Linear Loss Function and Quadratic Regularization

For this section, let us consider a linear loss function of the form $f^{(t)} = \boldsymbol{w} \cdot \boldsymbol{z}^{(t)}$. Recall from last lecture that the unregularized FTL with linear loss has unstable weight updates and $O(T)$ regret. Next, we add in quadratic regularization:

$$\psi(\boldsymbol{w}) = \frac{1}{2\eta}||\boldsymbol{w}||_2^2 \tag{34}$$

Note that this regularization function penalizes large values and tries to "pull" parameters towards zero. This new term changes the FTRL learner prediction rule to:

$$\boldsymbol{w}^{(t)} = \operatorname*{arg\,min}_{\boldsymbol{w} \in S}\left(\frac{1}{2\eta}||\boldsymbol{w}||_2^2 + \boldsymbol{w} \cdot \sum_{i=1}^{t-1}\boldsymbol{z}^{(i)}\right) \tag{35}$$

To solve the minimizer, take the derivative of the cumulative loss (the expression within the parentheses in (35)) with respect to $\boldsymbol{w}$ and set equal to zero to obtain:

$$\frac{2\boldsymbol{w}}{2\eta} + \sum_{i=1}^{t-1}\boldsymbol{z}^{(i)} = 0 \tag{36}$$

$$\boldsymbol{w}^{(t)} = -\eta\sum_{i=1}^{t-1}\boldsymbol{z}^{(i)} \tag{37}$$

Note that the shorthand $\theta^{(t)} = \sum_{i=1}^{t-1} \boldsymbol{z}^{(i)}$ is often used.

We can also decompose the equation for the minimizer into a recursive form:

$$\boldsymbol{w}^{(t)} = -\eta \sum_{i=1}^{t-1} \boldsymbol{z}^{(i)} \tag{38}$$

$$\implies \boldsymbol{w}^{(t)} = -\eta\Big(\sum_{i=1}^{t-2} \boldsymbol{z}^{(i)} + \boldsymbol{z}^{(t-1)}\Big) \tag{39}$$

$$\implies \boldsymbol{w}^{(t)} = -\eta \sum_{i=1}^{t-2} \boldsymbol{z}^{(i)} - \eta \boldsymbol{z}^{(t-1)} \tag{40}$$

$$\implies \boldsymbol{w}^{(t)} = \boldsymbol{w}^{(t-1)} - \eta \boldsymbol{z}^{(t-1)} \tag{41}$$

(41) is an update equation that allows us to update $\boldsymbol{w}$ incrementally at each time step. Furthermore, we can see that the update term $-\eta\boldsymbol{z}^{(t-1)}$ is proportional to $\boldsymbol{z}^{(t-1)}$, the gradient of the loss function $f^{(t)} = \boldsymbol{w} \cdot \boldsymbol{z}^{(t)}$ at $t-1$.

To derive the **regret bound** for this FTRL variant, substitute the linear loss function and quadratic regularization function into the generic FTRL regret bound (33):

$$R^{(T)}(\boldsymbol{u}) \leq \Big[\psi(\boldsymbol{u}) - \psi(\boldsymbol{w}^{(1)}\Big] + \sum_{t=1}^{T} \Big[f^{(t)}(\boldsymbol{w}^{(t)}) - f^{(t)}(\boldsymbol{w}^{(t+1)})\Big] \tag{42}$$

$$\leq \Big[\frac{1}{2\eta}||\boldsymbol{u}||_2^2 - 0\Big] + \sum_{t=1}^{T} \Big[\langle \boldsymbol{w}^{(t)}, \boldsymbol{z}^{(t)}\rangle - \langle \boldsymbol{w}^{(t+1)}, \boldsymbol{z}^{(t)}\rangle\Big] \tag{43}$$

$$= \frac{1}{2\eta}||\boldsymbol{u}||_2^2 + \sum_{t=1}^{T} \langle \boldsymbol{w}^{(t)} - \boldsymbol{w}^{(t+1)}, \boldsymbol{z}^{(t)}\rangle \tag{44}$$

Since from (41), we know the difference between $\boldsymbol{w}^{(t)}$ and $\boldsymbol{w}^{(t+1)}$ is simply the weight update $\eta\boldsymbol{z}^{(t)}$, the above expression becomes:

$$R^{(T)}(\boldsymbol{u}) \leq \frac{1}{2\eta}||\boldsymbol{u}||_2^2 + \sum_{t=1}^{T} \eta||\boldsymbol{z}^{(t)}||^2 \tag{45}$$

To simplify this, we will make use of the following shorthand notation:

$$L = \max_{\boldsymbol{z}} ||\boldsymbol{z}||_2 \tag{46}$$

$$B = \max_{\boldsymbol{u} \in S} ||\boldsymbol{u}||_2 \tag{47}$$

This yields:

$$R^{(T)}(\boldsymbol{u}) \leq \frac{1}{2\eta}B^2 + \eta T L^2 \tag{48}$$

Taking the derivative of the right-hand side and solving for the optimal value of η yields:

$$\eta = \frac{B}{L\sqrt{2T}} \tag{49}$$

Substituting this expression for η yields the final regret bound for FTRL, with a Euclidean regularizer term and a linear loss function:

$$R^{(T)}(\boldsymbol{u}) \leq BL\sqrt{2T} \tag{50}$$

Note that FTRL with Euclidean regularization and a linear loss function is just a specific example of convex optimization. More widely, online convex optimization generalizes these results to other forms of convex regularization functions, as well as any sequence of Lipschitz loss functions.