CSC2419:Algorithms and Complexity in Private Data Analysis Winter 2018

Lecture 9: Differentially Private Empirical Risk Minimization

Aleksandar Nikolov *Scribe: Patricia Thaine*

1 Set up

Today we will talk about differentially private machine learning, and specifically, about supervised learning. Our dataset X will be labeled, and will define it as $X = \{(x_1, y_1), ..., (x_n, y_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$, where $x_i \in \mathcal{X}, y_i \in \mathcal{Y}$. Here $\mathcal{X}$ is the set of possible data points, and $\mathcal{Y}$ is the set of possible labels. We will focus on d-dimensional data, i.e. $\mathcal{X} = [0,1]^d$ or $\{0,1\}^d$, and binary labels, i.e. $\mathcal{Y} = \{\pm 1\}$.

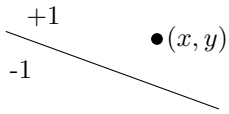
Given a family of functions $\{f_\theta : \theta \in \Theta\}$, our goal will be to find the function f_θ (or, equivalently, the parameter $\theta \in \Theta$) that best fits our labels. This will be modeled as minimizing a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Let's consider the following example:

$$l(y', y) = \begin{cases} 1, & \text{if } y' \neq y \\ 0, & \text{if } y' = y \end{cases}$$

Then if $f_\theta(x_i) = y_i$, i.e. f_θ correctly predicts the label of x_i, the loss $l(f_\theta(x_i), y_i)$ is 0, and otherwise it is 1. A classical example of a family of functions are linear predictors

$$f_\theta(x) = \text{sign}\big(\sum x_j \theta_j + \theta_0\big),\ \theta \in \mathbb{R}^{d+1}$$

Then $\sum x_j \theta_j + \theta_0 = 0$ is the equation of a hyperplane, and $f_\theta(x) = +1$ "above" the hyperplane, and $f_\theta(x) = -1$ below it. We have the following picture when $d = 2$:



$$\bullet (x, y) = \begin{cases} y = +1, & \text{loss is 0} \\ y = -1, & \text{loss is 1} \end{cases}$$

2 Empirical Risk minimization

In supervised machine learning, we have some fixed but unknown distribution D, and our goal is to find the function f_θ in our family that best predicts the labels of points sampled from this distribution. Formally, we want to find

$$\theta^* = \arg\min_{\theta \in \Theta} \mathop{\mathbb{E}}_{(x,y)\sim D} l(f_\theta(x), y) \tag{1}$$

where D is a distribution on $\mathcal{X} \times \mathcal{Y}$. This is the *risk minimization problem.* $\mathbb{E}_{(x,y)\sim D}\, l(f_\theta(x), y)$ is denoted $L(\theta, D)$. With the example loss function from the previous section, $L(\theta, D) = \mathbb{P}_{(x,y)\sim D}(f_\theta(x) \neq y))$, i.e. just the probability that f_θ misclassifies a labeled point drawn fro D.

Of course, this problem is unsolvable, unless we are given some form of access to D. The usual assumption then is that we are given a labeled dataset $X = \{(x_1, y_1)...(x_n, y_n)\}$ of IID samples from D. Then, instead of directly solving (1), we solve the *empirical risk minimization problem* (ERM) of finding

$$\theta_n = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} l(f_\theta(x_i), y_i). \tag{2}$$

This is the same as (1), except that we minimize the loss with respect to the empirical distribution induced by the dataset. The objective $\frac{1}{n}\sum_{i=1}^{n} l(f_\theta(x_i), y_i)$ is the empirical loss, and is denoted $L(\theta, X)$. Going back to our example, solving (2) corresponds to finding the function f_θ that best classifies the given dataset.

We hope that the θ_n found in (2) generalizes to the whole distribution. I.e. we would like that, as $n \to \infty$, $L(\theta_n, D) \to L(\theta^*, D)$. This means that our function f_{θ_n} classifies unseen examples drawn from the same distribution as well as it does the data set points. Statistical learning theory studies conditions on the family of functions $\{f_\theta : \theta \in \Theta\}$, the loss $l(y', y)$, and the distribution D, under which we can guarantee this convergence. While there are many interesting questions there, they are beyond the scope of this lecture. Instead, we will focus on how to solve the optimization problem (2) under differential privacy.

3 (Constrained) Logistic regression

While the loss from our example so far is very natural, it makes solving the optimization problem (2) computationally hard. A way around this is to define a surrogate loss for which ERM is more tractable. We will focus on the *logistic loss* function, defined as

$$l(y', y) = \log(1 + e^{-y \cdot y}),$$

and the class of linear functions $f_\theta(x) = \sum_{i=1}^{d} \theta_i x_i + \theta_0$ over $x \in \mathcal{X} = [0,1]^d$. By adding a coordinate $x_0 = 1$ to every point x (so expanding $\mathcal{X}$ to $[0,1]^{d+1}$), we can simplify f_θ a bit to just $f_\theta(x) = \theta^\top x$. This set up is motivated by a statistical model in which the label y of a point x is a random variable with a distribution parametrized by θ, and the logarithm of the odds ratio $\frac{\mathbb{P}(y|\theta)}{\mathbb{P}(-y|\theta)}$ is $f_\theta(x)$. Then $\frac{1}{1+e^{-f_\theta(x)y}}$ is interpreted as $\mathbb{P}(y|\theta)$, and minimizing the logistic loss corresponds to finding the maximum likelihood estimate of θ given the data.

We will solve a constrained version of this problem in which f_θ varies over $\theta \in \Theta = B_2^{d+1}(R)$, where $B_2^{d+1}(R)$ is the Euclidean ball of radius R centered at 0. This is a form of regularization: it constrains the hypothesis f_θ we find to be "simple", thus helping with generalization. For us, this constraint will be crucial so that we can solve the ERM problem with differential privacy.

Instead of imposing a constraint, we can instead work with the ℓ_2-regularized logistic loss $l(y', y) = \log(1 + e^{-z}) + \frac{\lambda}{2} \cdot \|\theta\|_2^2$. This is similar to the constrained problem, as it forces the optimal solution to be in a ball of bounded radius. The techniques we will use adapt easily to the regularized setting as well.

While we will use logistic regression as a running example in this lecture, the methods apply much more generally to constrained empirical risk minimization with convex loss.

4 Nosiy Gradient Descent

Unlike least squares regression, there is no closed form expression for the optimal θ in the logistic regression problem. Instead we usually solve it (approximately) using a general purpose convex optimization algorithm. A popular choice is gradient descent:

$\theta^0 \leftarrow 0$
for $t = 1...T-1$ **do**
  $\tilde{\theta}^t \leftarrow \theta^{t-1} - \eta \nabla L(\theta^{t-1}, X)$
  $\theta^t \leftarrow \tilde{\theta}^t / \max\{1, \|\theta^t\|_2 / R\}$
end for
output $\frac{1}{T}\sum_{t=0}^{T-i} \theta^t$

Above, $\nabla L(\theta, X)$ is the gradient with respect to θ, i.e. it's a vector whose i-th coordinate is the partial derivative $(\nabla L(\theta, X))_i = \frac{\partial L(\theta, X)}{\partial \theta_i}$. At every step, the algorithm moves θ a little bit in the direction opposite to the gradient: this is the direction in which the loss locally decreases the fastest. If θ ever leaves the ball, it is scaled back inside. See Figure 1.

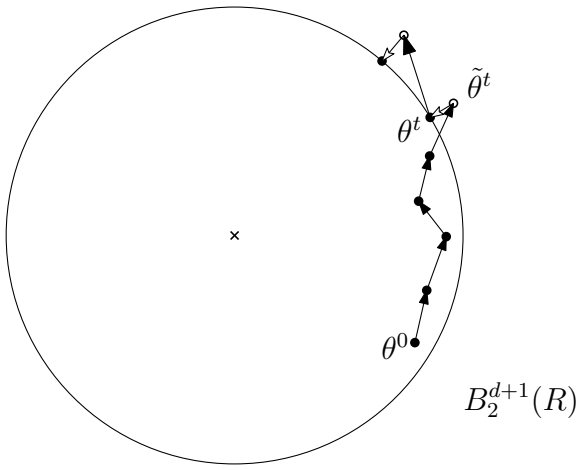


Figure 1: Gradient Descent

Our approach will be to try to make this algorithm differentially private. Notice that, in any single step, the only thing that depends on the database X is the gradient of the loss L. So, a natural strategy is to add noise to the gradient. We get the following noisy variant of gradient descent:

$\theta^0 \leftarrow 0$
for $t = 1...T-1$ **do**
  $\tilde{\theta}^t \leftarrow \theta^{t-1} - \eta(\nabla L(\theta^{t-1}, X) + w^t)$
  $\theta^t \leftarrow \tilde{\theta}^t / \max\{1, \|\theta^t\|_2 / R\}$
end for
output $\frac{1}{T}\sum_{t=0}^{T-1} \theta^t$

w^t is random noise that is added to make gradient descent differentially private. The noise will be Gaussian, and in the next section we will explore how large it needs to be.

5 Advanced composition (for Gaussian noise)

Recall the Gaussian noise mechanism: for a function f that maps databases to m-dimensional vectors, we release $\mathcal{M}_{\text{Gauss}}(x) = f(x) + w$ where every w_i is an independent Gaussian with mean 0 and variance $\sigma^2_{\varepsilon,\delta} \cdot \Delta_2 f^2$), $\sigma^2_{\varepsilon,\delta} \approx \theta\Big(\frac{k\sqrt{\log(1/\delta)}}{\varepsilon}\Big)$. Here $\Delta_2 f$ is the ℓ_2-sensitivity of f, defined as: $\Delta_2 f = \max_{x \sim x'} \|f(x) - f(x')\|_2$

Say we want to publish k functions $f_1, ..., f_k$ such that $\Delta_2 f_i \leq C$ for each one of them. One thing we can do is publish each one using the Gaussian noise mechanism with privacy parameters $(\varepsilon/k, \delta/k)$, and then use the composition theorem to argue about privacy. Then we would publish $f_1(x) + w^1, \ldots, f_k(x) + w^k$ where $w^i_j \sim \mathcal{N}(0, \sigma^2_{\varepsilon/k,\delta/k} C^2)$ for every i and j. This increases the variance of the noise by roughly k^2 with respect to only releasing a single function.

We can, however, do better: it's enough to set $w^i_j \sim \mathcal{N}(0, k\sigma^2_{\varepsilon,\delta} C^2)$, and, as above, output $f_1(x) +$

$w^1, ..., f_k(x) + w^k$. This is equivalent to running the Gaussian noise mechanism on the function

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_k(x) \end{pmatrix}.$$

The privacy guarantee follows from the privacy of the Gaussian noise mechanism and the observation

$$\begin{aligned} \Delta_2 f^2 &= \max_{x \sim x'} \|f(x) - f(x')\|_2^2 \\ &= \max_{x \sim x'} \sum_i \|f_i(x) - f_i(x')\|_2^2 \\ &\leq \sum_i \max_{x \sim x'} \|f_i(x) - f_i(x')\|_2^2 \\ &\leq kC^2. \end{aligned}$$

However, the analysis via the composition theorem still had something going for it, because it still works even when the choise of the function f_i is determined by $f_1(x) + w^1, \ldots, f_k(x) + w^k$. Can we get privacy with the improved variance above in this adaptive setting too? It turns that the answer is "yes": in fact the privacy analysis of the Gaussian noise mechanism we did in the beginning of the course can be easily adapted to this setting. That is, outputting $f_i(x) + w^i$, $w^i \sim \mathcal{N}(0, k\sigma^2_{\varepsilon,\delta} C^2)$ where f_i can depend on $f_1(x) + w^1, ..., f_{i-1}(x) + w^{i-1}$, is (ε, δ)-DP.

This is an instance of the advanced composition theorem: check the Dwork and Roth monograph for a more thorough discussion of it. Here we are stating it only for Gaussian noise, but, just like the simple composition theorem, a similar statement holds for adaptive composition of arbitrary (ε, δ)-differentially private mechanisms. The proof is again similar to the analysis of the Gaussian noise mechanism, but the parameters become worse, which is why we will stick to the Gaussian version.

6 Back to Noisy Gradient Descent

Let us now apply the Gaussian advanced composition theorem to the noisy gradient descent algorithm and determine how large the noise needs to be at each step. In this case the function $f_1, f_2, \ldots$ are simply the gradients $\nabla L(\theta^0, X)$, $\nabla L(\theta^1, X)$, etc. To apply the composition theorem, we need to bound the sensitivity of these gradients, now seen as functions of X.

Suppose $\|\nabla l(f_\theta(x_i), \theta)\|_2 \leq C$ for every $\theta \in \Theta$ and every $x_i \in \mathcal{X}$, where again we take gradients with respect to θ.
Then, because $\nabla L(\theta, X) = \frac{1}{n} \sum_{i=1}^n \nabla l(f_\theta(x_i), \theta)$,

$$\begin{aligned} \Delta_2 \nabla L(x, \theta) &= \max_{x \sim x'} \|\nabla L(x, \theta) - \nabla L(x', \theta)\| \\ &= \max_{x \in \mathcal{X}} \left\| \frac{1}{n} \nabla l(f_\theta(x_i), \theta) \right\|_2 \leq \frac{C}{n}. \end{aligned}$$

So, in private gradient descent, we can set $w_i^t \sim \mathcal{N}(0, T\sigma^2_{\varepsilon,\delta} \cdot \frac{C^2}{n^2})$, for each t and i in order to achieve (ε, δ)-differential privacy.

For logistic regression: with the loss $l(f_\theta(x_i), y_i) = \log(1 + e^{-y_i \theta^\top x_i})$ we have

$$\nabla \log(1 + e^{-y_i \theta^\top x_i}) = -\frac{1}{1 + e^{y_i \theta^\top x_i}} y_i x_i \Rightarrow \|\nabla \log(1 + e^{-y_i \theta^\top x_i})\|_2 \leq \|x_i\|_2 \leq \sqrt{d+1}.$$

So $w_t \sim \mathcal{N}(0, T\sigma_{\varepsilon,\delta}^2 \cdot \frac{d+1}{n^2})$ is enough noise to achieve (ε, δ)-differential privacy.

7 Error analysis

Finally, we want to say that the noisy gradient descent algorithm, given sufficient data, does actually come close to the optimal parameter vector of the logistic regression problem. Luckily, gradient descent is an incredibly robust algorithm, and it's possible to still give convergence guarantees under a very general noise model.

Algorithms like the noisy gradient descent algorithm above are special cases of Stochastic Gradient Descent. In general, in stochastic gradient descent we take a step in a random direction, which ideally has expectation equal to minus te gradient, and not too large variance. The general algorithm is

$\theta^0 \leftarrow 0$
for $t = 1...T-1$ **do**
  $\tilde{\theta}^t \leftarrow \theta^{t-1} - \eta z^t$
  $\theta^t \leftarrow \tilde{\theta}^t / \max\{1, \|\theta^t\|_2 / R\}$
end for
output $\frac{1}{T}\sum_{t=0}^{T-1} \theta^t$

Above z^t is a random variable, whose distribution may depend on $z^1, \ldots, z^{t-1}$. In our case $z^t = \nabla L(\theta^{t-1}, X) + w^t$. Another common variant of SGD is to set it equal to the gradient of a random point in the dataset, or the average of several random points. This is often done to speed up the algorithm, since the most expensive step in gradient descent is to compute the gradient of the entire loss function.

In the next section we will prove the following general guarantee for SGD.

Theorem 1 *Let $L(\theta, X)$ be convex in θ for all X. Suppose $\mathbb{E}[z_t \mid \theta^{t-1}] = \nabla L(\theta^{t-1}, X)$ and $\mathbb{E}\|z_t\|_2^2 \le B^2$. For $\eta = \frac{R}{BT^{1/2}}$ we have $\mathbb{E}L\Big(\frac{1}{T}\sum_{t=0}^{T-1}\theta^t, X\Big) \le \min_{\theta \in B_2^{d+1}(R)} L(\theta, X) + \frac{RB}{T^{1/2}}$.*

Using this theorem for private gradient descent, we have

$$z_t = \nabla L(\theta^{t-1}, X) + w_t$$

$$\mathbb{E}\|z_t\|_2^2 = \|\nabla L(\theta^{t+1}, X)\|_2^2 + \mathbb{E}\|w_t\|^2 \le C^2 + \frac{T\sigma_{\varepsilon,\delta}^2 C^2 (d+1)}{n^2} = C^2\Big(1 + \frac{T\sigma_{\varepsilon,\delta}^2 (d+1)}{n^2}\Big)$$

This bounds B, and plugging the bound into Theorem 1, we get

$$\begin{aligned}\mathbb{E}L\Big(\frac{1}{T}\sum_{t=0}^{T-1}\theta^t, X\Big) - \min_{\theta \in B_2^{d+1}(R)} L(\theta, X) \le \frac{RB}{T^{1/2}} &\le \frac{RC}{T^{1/2}} \cdot \Big(1 + \frac{T\sigma_{\varepsilon,\delta}^2 (d+1)}{n^2}\Big)^{1/2} \\ &\le \frac{RC}{T^{1/2}} + \frac{RC\sigma_{\varepsilon,\delta}\sqrt{d+1}}{n}\end{aligned}$$

Note that this error bound decomposes into two parts: one goes to 0 with T, and would be there even if we added no noise, i.e. just ran standard gradient descent. The second term in the error bound is due to the noise, and goes down with n, since the variance of our noise decreases with n as well. To bound the error by α, we set both terms to be less than $\frac{\alpha}{2}$ and we get the following setting of parameters

$$T \ge \frac{4R^2C^2}{\alpha^2}; \qquad\qquad n \ge \frac{2RC\sigma_{\varepsilon,\delta}\sqrt{d+1}}{\alpha} \gtrsim \frac{RC\sqrt{\log(1/\delta)}\cdot\sqrt{d+1}}{\alpha\varepsilon}.$$

In the case of logistic regression, we can just plug in $C \le \sqrt{d+1}$ in the bounds above.

8 Proof of Theorem 1

To be more concise, let's define $g(\theta) = L(\theta, X)$. We start with a basic fact about convex functions: the graph of a convex function always lies above any of its tangent hyperplanes (see Figure 2) for what this looks like in a single variable). This means that for all θ and θ', we have

$$g(\theta') \geq g(\theta) + (\theta' - \theta)^\top \nabla g(\theta) \tag{3}$$

Stated equivalently, the local linear approximation to g at θ (on the right hand side) is always an underestimate with respect to the actual function.

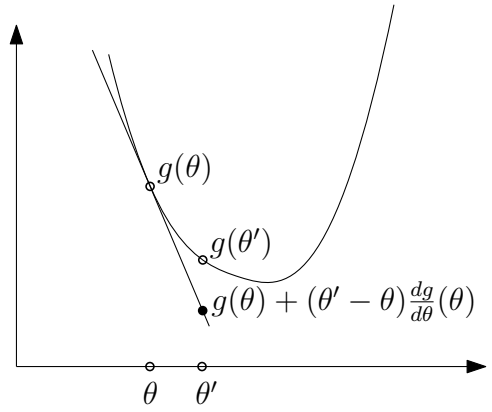


Figure 2: Illustration of (3)

Let $\theta^* = \arg\min_{\theta \in \Theta} g(\theta)$ be the optimal solution. Applying (3) to θ^* and θ^{t-1}, for any $1 \leq t \leq T$, we get

$$g(\theta^*) \geq g(\theta^{t-1}) + (\theta^* - \theta^{t-1})^\top \nabla g(\theta^{t-1})$$

or, equivalently,

$$g(\theta^{t-1}) - g(\theta^*) \leq (\theta^{t-1} - \theta^*)^\top \nabla g(\theta^{t-1})^\top = \mathbb{E}[(\theta^{t-1} - \theta^*)^\top z_t \mid \theta^{t-1}].$$

Taking expectations over θ^{t-1}, by the total law of expectation, we have

$$\begin{aligned}\mathbb{E}[g(\theta^{t-1}) - g(\theta^*)] \leq \mathbb{E}[\mathbb{E}[(\theta^{t-1} - \theta^*)^\top z_t \mid \theta^t]] &= \mathbb{E}[(\theta^{t-1} - \theta^*)^\top z_t] \\ &= \frac{1}{\eta}\mathbb{E}[(\theta^{t-1} - \theta^*)^\top (\theta^{t-1} - \tilde{\theta}^t)] = (\star)\end{aligned}$$

Next, we use a nice little trick, known as polarization: for any two vectors x and y in $\mathbb{R}^{d+1}$, we have $\|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2x^\top y$, so we can express their dot product as $x^\top y = \frac{1}{2}(\|x\|_2^2 + \|y\|_2^2 - \|x - y\|_2^2)$. Applying this to $x = \theta^{t-1} - \theta^*$ and $y = \theta^{t-1} - \tilde{\theta}^t$, on the right hand side above, we get

$$\begin{aligned}(\star) &= \frac{1}{2\eta}\mathbb{E}[\|\theta^{t-1} - \theta^*\|_2^2 + \|\theta^{t-1} - \tilde{\theta}^t\|_2^2 - \|\tilde{\theta}^t - \theta^*\|_2^2] \\ &= \frac{1}{2\eta}\mathbb{E}[\|\theta^{t-1} - \theta^*\|_2^2 - \|\tilde{\theta}^t - \theta^*\|_2^2] + \frac{\eta}{2}\mathbb{E}[\|z_t\|_2^2] \\ &\leq \frac{1}{2\eta}\mathbb{E}[\|\theta^{t-1} - \theta^*\|_2^2 - \|\tilde{\theta}^t - \theta^*\|_2^2] + \frac{\eta B^2}{2} = (\star\star).\end{aligned}$$

Now we need one final inequality:

$$\|\theta^t - \theta^*\|_2 \leq \|\tilde{\theta}^t - \theta^*\|_2. \tag{4}$$

The argument for this is similar to the one we used in the analysis of the projection mechanism. When $\|\tilde{\theta}^t\|_2 \leq R$ the two sides above are equal. Otherwise, $\theta^t = R\tilde{\theta}^t / \|\tilde{\theta}^t\|_2$ is the closest point to $\tilde{\theta}^t$ inside the ball $B_2^{d+1}(R)$. This means that θ^*, $\tilde{\theta}^t$ and θ^t form a triangle with a non-acute (i.e. right or obtuse) angle at θ^t, as in Figure 3. Therefore, the right hand side in (4) equals the length of the side of the triangle opposite the non-acute angle, and the left hand side is the length of one of the other sides, which can be only smaller, by the cosine law.

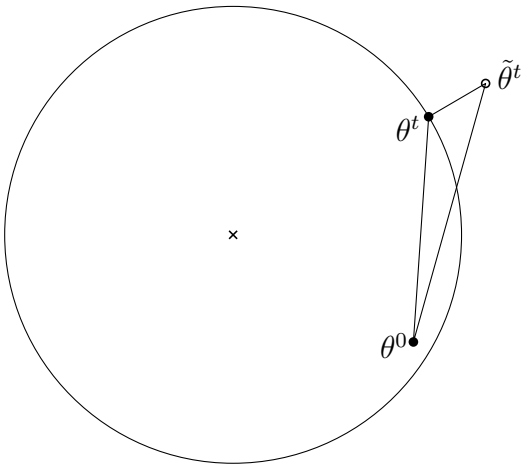


Figure 3: Projection back to the ball

Plugging (4) back into our calculations, we see that

$$(\star\star) \leq \frac{1}{2\eta}\mathbb{E}[\|\theta^{t-1} - \theta^*\|_2^2 - \|\theta^t - \theta^*\|_2^2] + \frac{\eta B^2}{2}.$$

Putting everything together, we have shown that

$$\mathbb{E}[g(\theta^{t-1}) - g(\theta^*)] \leq \frac{1}{2\eta}\mathbb{E}[\|\theta^{t-1} - \theta^*\|_2^2 - \|\theta^t - \theta^*\|_2^2] + \frac{\eta B^2}{2}.$$

We have one of these inequalities for each $t \in \{1, \ldots, T\}$, and if average them, the right hand sides telescope, and we get

$$\begin{aligned}
\mathbb{E}\left[g\left(\frac{1}{T}\sum_{t=0}^{T-1}\theta^t\right) - g(\theta^*)\right] &\leq \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} g(\theta^t) - g(\theta^*)\right] \\
&= \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} g(\theta^{t-1}) - g(\theta^*)\right] \\
&\leq \frac{1}{2T\eta}\mathbb{E}[\|\theta^0 - \theta^*\|_2^2 - \|\theta^T - \theta^*\|_2^2] + \frac{\eta B^2}{2} \\
&\leq \frac{R^2}{2T\eta} + \frac{\eta B^2}{2}.
\end{aligned}$$

The first inequality above follows from the convexity of g. The final inequality follows because $\theta^0 = 0$ and $\theta^* \in B_2^{d+1}(R)$, so $\|\theta^0 - \theta^*\|_2 \leq R$. Optimizing over the choice of η finishes the proof.