Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

Hyperparameter Tuning

Now, we know how to **evaluate** a learning algorithm, just like how we **evaluate** a hypothesis.

Once we knew how to evaluate a hypothesis, we started **optimizing** our parameters for the **best** hypothesis. So, we could do the same for our **learning algorithm**.

Each λ value creates a slightly **different** learning algorithm: we can **optimize** this **hyperparameter** to create the **best** learning algorithm.

How to tune our algorithm

When we were **optimizing** our hypothesis, we started by **randomly** trying hypotheses. Then, we used an **analytical** approach.

By "analytical", we mean directly creating an equation, and solving it.

We don't always have **simple** equations to work with: with all of our data, it's hard to come up with **manageable** equations. So, we **won't** try doing it **analytically**.

So, we could **randomly** try λ values and pick the **best** one. This is pretty **close** to what we usually end up doing. For each value we pick, we'll use **cross-validation** to evaluate.

For now, we'll systematically go through λ values: $\lambda = .1, .2, .3 \ldots$

Concept 1

Hyperparameter tuning is how we **optimize** our **learning algorithm** to create the **best** hypotheses.

The simplest way to do this is to try **multiple** different values of λ. For each value, we use **cross-validation** to evaluate that learning algorithm.

Finally, we pick whichever λ gives you the **best** algorithm, and thus the **best** hypotheses.

Hyperparameter Tuning: Two kinds of optimization

There's something often **confusing** about hyperparameter tuning to students:

When we're **optimizing** λ, we have to determine the quality of **each** learning algorithm.

But, to get the **quality** of that algorithm, we have to optimize Θ based on that **single** learning algorithm.

If we do cross-validation, then we have to optimize k times!

That means, **every time** we try a different λ value, we have to do one optimization problem. But trying different λ values is a **different** kind of optimization.

That means we have **two layers** of optimization!

Clarification 2

We **optimize** λ by trying many values.

But, for each λ value, we have to **optimize** Θ.

So, we have to optimize Θ **repeatedly** in order to optimize λ **once**! This gives us λ^*.

But, our goal is to get a **hypothesis**. So we use that λ^* to, finally, get our θ^*

Pseudocode Example

This technique is **not** limited to regression. Thus, we'll be a bit more **general**: we won't assume an **analytical** solution. Instead, we **optimize** by just trying different Θ values.

We can represent this in pseudocode:

If this pseudocode isn't helpful to you, don't worry! Some students like it, some don't.

LAMBDA-OPTIMIZATION($\mathcal{D}$, lambda_values, theta_values)

```
1  for λ in lambda_values          #Try lambda values
2      for Θ in theta_values       #Try theta values
3          Calculate J(Θ)          #Compare values
4      Choose best theta value Θ*  #Best for each lambda
5  Choose best lambda value λ*
6
7  return λ*
```

To reiterate: this λ^* will then we used to get our final result, θ^*.