Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

Regularization

So far, we've shown how to make the **best** model for our **training data**. But now, we want to move to our **real** goal: performing well on **test data**.

This means we want to make a model that is **general**: it can apply well to **new data**.

Regularizers

Only focusing on training data is a **weakness** for our model - if by chance, we have a training data that doesn't **match** our overall distribution, we are likely to make a **bad model**.

Example: You flip **4 coins**, and get **3 heads**. You determine that this coin has a **75% chance** of landing heads. It turns out this **isn't true**: it's a fair coin, and you got **unlucky**.

You can also increase sample size (flip more times), but for complex problems this isn't always an option!

We may need a **second** way to measure our performance: one that focuses **less** on **current** performance, and **more** on predicting how **generalizable** it is.

We call this type of function a **regularizer**.

Definition 1

A **regularizer** is an added term to our **loss function** that helps measure how **general** our hypothesis is.

By **optimizing** with this term, we hope to create a model that works better with **new data**.

This function takes in our **vector of parameters** Θ as an input: $R(\Theta)$

Example: You figure that the coin is **equally likely** to bias towards heads or tails: even if it's **weighted**, you don't know **which way**. So, you start with **50-50** odds, and **adjust** that based on evidence.

Instead of just focusing on the **specific** data for our coin, we consider how coins act in **general**.

Regularizer for Regression: Prior Knowledge

Now, the question is, **how** do we choose our regularizer? What will make our model more **general**?

We want to **resist** the effects of random **chance**, like in the **coin** example above. In that example, we improved our guess by starting with a **prior assumption**.

If you have some **previous** guess, or past experience, you might have some **model** you **expect** to work well: the data has to **convince** you otherwise.

So, we might consider a model **more different** from that past one, Θ_{prior}, to be **suspicious**, and less likely to be good.

Concept 2

If we have a **prior** hypothesis Θ_{prior} to work with, we might improve our **new** model by encouraging it to be **closer** to the old one.

$$R(\Theta) = \left\|\Theta - \Theta_{\text{prior}}\right\|^2$$

We measure how **similar** they are using **square distance**.

Example: You have a **pretty good** model for **predicting** company profits, but it isn't perfect. You decide to train a **better** one, but you expect it to be **similar** to your old one.

Regularizer for Regression: No Prior Hypothesis

But, what if we **don't have** a prior hypothesis? What if we have **no clue** what a **good** solution looks like?

Well, just like in the **coin** example, we don't expect it to be **more likely** to be **weighted** towards heads or tails.

So, even if we **didn't know** most coins are fair coins, we still would've chosen **50-50** as our guess.

In this case, as far as we know, every θ_k term is **equally likely** to be **positive or negative** - we have no clue.

So, **on average**, we could push for it to be **closer to zero**, so it doesn't drift in any direction too strongly.

Key Equation 3

In general, our **regularizer for regression** will be given by **square magnitude** of θ:

$$R(\Theta) = \|\theta\|^2 = \theta \cdot \theta$$

This approach is called **Ridge Regression**.

We'll discuss why it's called "ridge" regression once we find our solution.

Why not include θ_0?

One thing you might immediately notice is that we used the magnitude of θ instead of Θ: this omits θ_0. Why would we do that?

We'll show that we need to **allow** the **offset** to have whatever value works best, and we shouldn't **punish** it.

For simplicity, we won't do any regularization here: we can make our point without it.

This is best shown with a **visual** example. Let's take an example with one input x_1. So, we have a **linear** function: $h(x) = \theta_1 x_1 + \theta_0$.

images/regression_images/Regression_Keep_Offset.png

Our regression example.

Let's suppose we **push** for a **much lower** (offset) θ_0 term, while keeping everything else the **same**:

images/regression_images/Regression_Remove_Offset.png

Reducing our offset pulls our line further away from all of our data! That's not helpful.

This shows that we **need** our offset! We use it to **slide** our hyperplane around the space: if all of our data is **far** from $(0, 0)$, we need to be able to **move** our **entire line**.

And regularizing θ_1 wouldn't make this any better: it would just be flatter.

So, we'll keep θ_0 **separate** and **allow** it to take whatever value is **best**.

Concept 4

We **do not regularize** our **offset** term, θ_0.

Instead, we allow θ_0 to **shift** our hyperplane wherever it **needs** to be.

The other terms θ control the **orientation** of the hyperplane: the **direction** it is **facing**. We **regularize** this to push it towards less "complicated" orientations.

This will be discussed more in-depth in the Classification chapter!

A second benefit of regularization

Another benefit of regularization is that it solves a second problem: having **multiple optimal solutions**.

If we have **multiple** best outcomes, we have to pick which one to **choose**. We can make this choice by **picking** the one with the **smallest** magnitude.

We can **visualize** the problem of "multiple best solutions" a couple different ways:

images/regression_images/Regularizer_Multiple_Soluti

There are many **planes** that can go through this line: multiple equally good solutions!

images/regression_images/Regression_Multiple_Solutions_Example.png

This compares different hypotheses (θ_1) and sees how well they perform (J): two are equally good!

Either way, we can pick a solution based on lowest θ **magnitude**!

A Math Perspective: Unique Solutions

We can also view this problem more **mathematically**.

Let's look at our **analytical** solution:

$$\theta = \left(XX^T\right)^{-1} XY^T \tag{1}$$

This solution only works if $\left(XX^T\right)^{-1}$ is **valid**. But we have a problem: **not all matrices** have **inverses**.

If XX^T has a **determinant** of **zero**, then we cannot find an inverse.

This is an important idea in linear algebra! If you don't know what this means, here's a great video.

Without an inverse, we have **no unique solution**! This is a problem.

This is one thing our **regularizer** $R(\Theta)$ helps us solve: we'll see that our **new solution** will not have this problem!

The reason will be clear in the **algebra**, but it's **equivalent** to the reason we discussed the above: we take the best **models** that are all **equally good**, and pick the one with **lowest magnitude**.

Concept 5

Ridge Regression helps **improve** our model by

- Making our model more **general** and resistant to **overfitting**
- Making sure **solutions** are **unique**
- Keeping our matrix XX^T **invertible**, so we can find a **solution**.

Lambda, a.k.a. λ

We now have a term that can help us choose a more **general** hypothesis. One important question is, **how general** do we want it to be?

The more general we make our model, the **less specific** to our current data it is. This may seem like a good thing, but too much can make our model **worse**!

If λ is **too large**, then your model will stay **very close** to $||\theta|| = 0$. This probably isn't a good solution for most cases.

But if it's **too small**, then it **won't** have enough of an **effect**. So, we need to be able to adjust how **much** we're regularizing.

To do this, we will **scale** our regularizer by a **constant** factor, λ.

Definition 6

Lambda, or λ, is the constant we **scale** our **regularizer** by.

It represents **how strongly** we want to regularize: how much we prioritize **general** understanding over **specific** understanding.

Our new objective function

Now that we have our regularizer,

$$R(\Theta) = \lambda ||\theta||^2 \tag{2}$$

We can add it to our objective function:

Key Equation 7

The **objective function** for **ridge regression** is given as

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\Bigg(\underbrace{(\theta^T x^{(i)} + \theta_0)}_{\text{guess}} - \underbrace{y^{(i)}}_{\text{answer}}\Bigg)^2 + \underbrace{\lambda\|\theta\|^2}_{\text{Regularizer}}$$

This is the form we will **solve**.

Matrix Form Ridge Regression

Just like before, we'll switch from a **sum** to a **matrix** in order to solve this problem.

Creating an **equation** for both θ and θ_0 is, frankly, **annoying** to **derive**. **Instead**, we'll cheat a little, and keep θ_0 in and create our **matrix-form** objective function:

$$J = \frac{1}{n}\left(\tilde{X}\theta - \tilde{Y}\right)^T\left(\tilde{X}\theta - \tilde{Y}\right) + \lambda\left(\theta^T\theta\right) \tag{3}$$

Our work begins. Let's take the **gradient**: what we want to set to zero.

$$\nabla_\theta J = \frac{2}{n}\tilde{X}^T\left(\tilde{X}\theta - \tilde{Y}\right) + 2\lambda\theta = 0 \tag{4}$$

We do some algebra and **solve** as we do in the **official notes**:

Key Equation 8

The **solution** for **ridge regression optimization** is

$$\theta = \left(\tilde{X}^T\tilde{X} + n\lambda I\right)^{-1}\tilde{X}^T\tilde{Y}$$

Or, in our original notation,

$$\theta = \left(XX^T + n\lambda I\right)^{-1}XY^T$$

Our new term, $n\lambda I$

So, we already established that **regularization** helps us create more **general** hypotheses that are lower in magnitude.

But, how does this **mathematically** solve our invertibility problem?

$$\theta = \left(XX^T + n\lambda I\right)^{-1}XY^T \tag{5}$$

This term, $n\lambda I$, is added to the matrix we want to invert. Let's see what this matrix looks like. We'll use a (3×3) example:

I is the **identity matrix** in our notation.

$$n\lambda I = n\lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = n \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \tag{6}$$

This visual, having a "**ridge**" of λs along the diagonal, is why we call it **ridge regression**.

Invertibility

This term $n\lambda I$ **shifts** the values of XX^T so that we **avoid** having a **determinant of zero**.

Since the **determinant is nonzero**, we don't have to worry about an **uninvertible matrix**: we now have a **unique** inverse, and thus a **unique** solution.

Concept 9

Ridge Regression solves the problem of **matrix invertibility** (non-unique solutions) by adding a term $n\lambda I$, our **ridge** of diagonals.

This turns the inverse $(XX^T)^{-1}$ into

$$(XX^T + n\lambda I)^{-1}$$

Which can prevent a **determinant** of zero in our solution, given $\lambda > 0$.