WEEK 3 RECAP

MODELING REDUX

Review Topics

I. MODELING CONCEPTUALLY

II. MODELING PRACTICALLY

MODELING CONCEPTUALLY

Why Do We Model?

The goal of modeling is to uncover patterns hidden in data. These patterns take the form of relationships between the characteristics — or features — of our data. When we understand these relationships, we can make predictions on new data.

Example

In NYC, the monthly rent of an apartment falls on average $200 for every 13.2 minutes beyond the mean one-way commute time.

Example

In NYC, the monthly rent of an apartment falls on average $200 for every 13.2 minutes beyond the mean one-way commute time.

Given an apartment and its median commute time, this information can help us predict its monthly rent.

Example

In NYC, the monthly rent of an apartment falls on average $200 for every 13.2 minutes beyond the median one-way commute time.

Given an apartment and its median commute time, this information can help us predict its monthly rent.

That in a nutshell is modeling.

Types of Models

SUPERVISED

Using labeled data, we learn the
relationship between data features and our target value.

This target can be a continuous real-value (1.2, 2.76, etc) or
it can be a discrete value ('spam', 'not spam'). In the former
it is called regression, and in the latter it is called classification.

Types of Models

UNSUPERVISED

Using unlabeled data, we uncover the structure in the data.

Examples of this are clustering and dimensionality reduction.

We're going to focus on supervised learning for now…

We're going to focus on supervised learning for now…

Regression specifically.

We have been given a data set that is based upon sending a survey to employees to assess their job satisfaction. From this survey, we have their salary and their level of satisfaction.

Modeling Conceptually

We have been given a data set that is based upon sending a survey to employees to assess their job satisfaction. From this survey, we have their salary and their level of satisfaction.

We are going to use this data to fit a model.

We have been given a data set that is based upon sending a survey to employees to assess their job satisfaction. From this survey, we have their salary and their level of satisfaction.

We are going to use this data to fit a model.

To evaluate the goodness of fit of our model, we are going to minimize the sum of squared errors. This is known as our cost function.

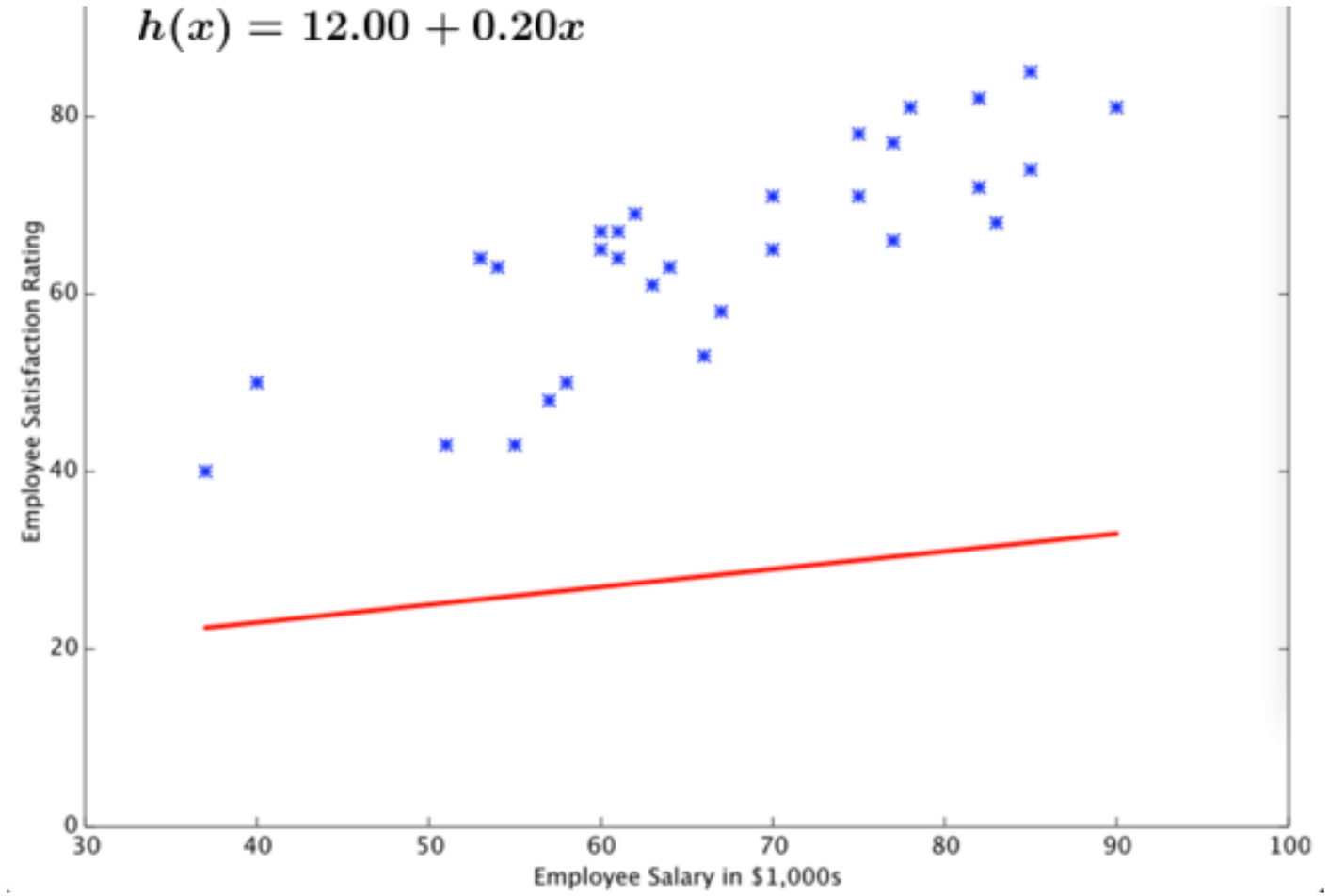
We have been given a data set that is based upon sending a survey to employees to assess their job satisfaction. From this survey, we have their salary and their level of satisfaction.

We are going to use this data to fit a model.

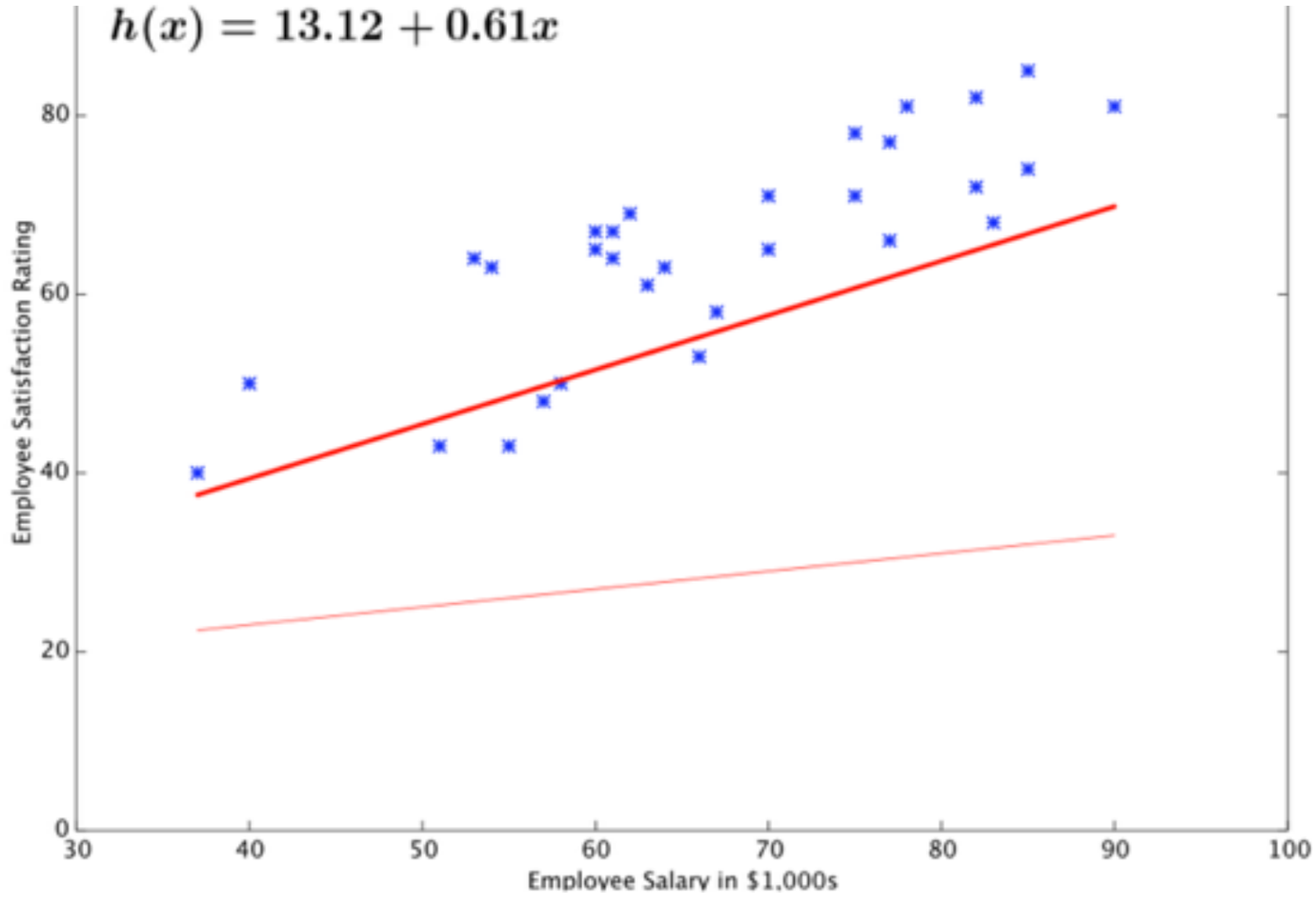
To evaluate the goodness of fit of our model, we are going to minimize the sum of squared errors. This is known as our cost function.

We will solve this problem iteratively using gradient descent.

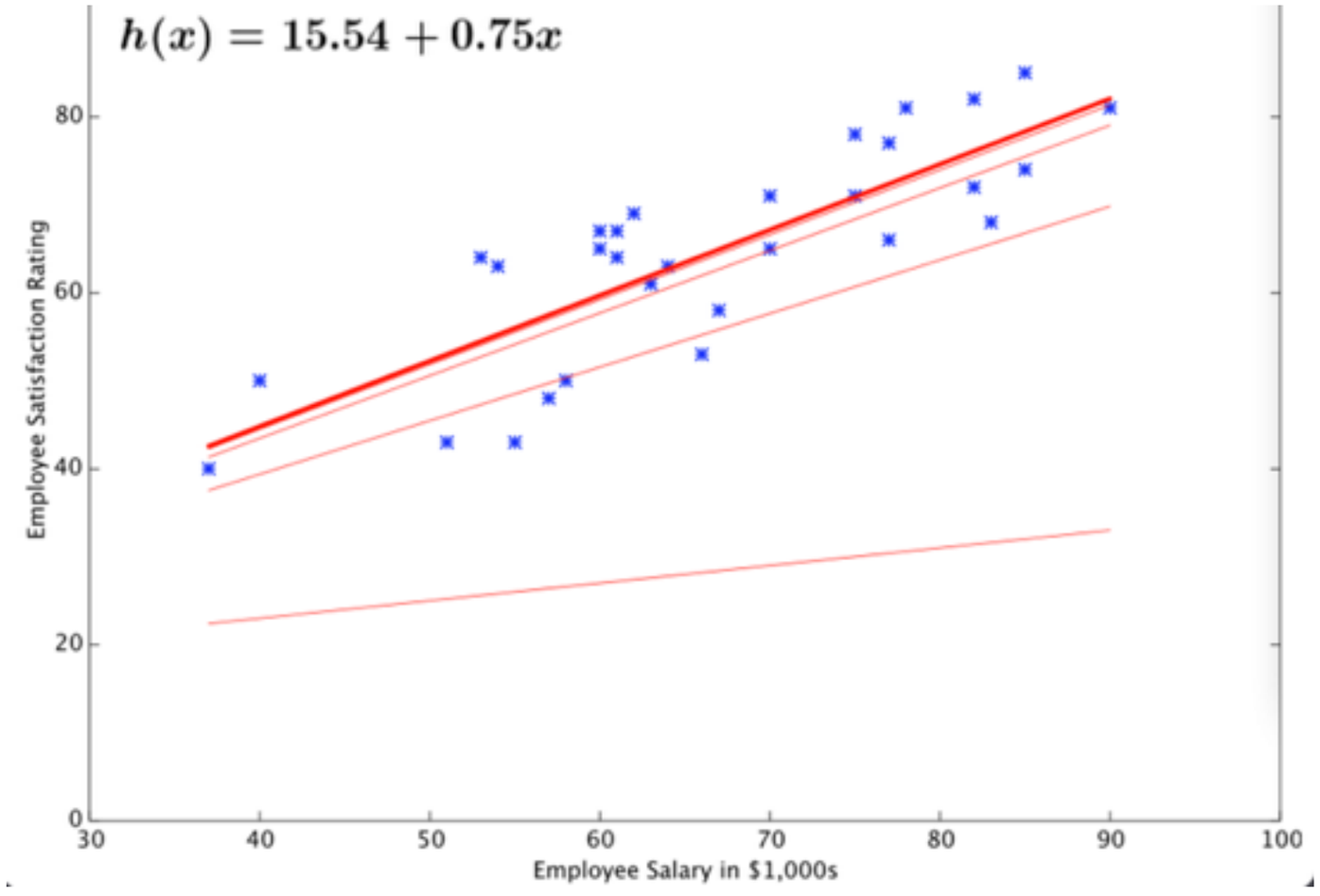
We begin by initializing our model using random values.



We evaluate the goodness of fit by measuring our error and then update our coefficients.



We repeat again and again and again until the model is no longer able to improve. At this point is has reached its minimum and is said to have 'converged'. Our coefficients are now set for this model.



Super.

Now can you please me how you knew where to move the line to each time?

Sure.

You remember the cost function. And you remember it was — for this problem at least — the sum of squared errors. The objective was to minimize it.

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

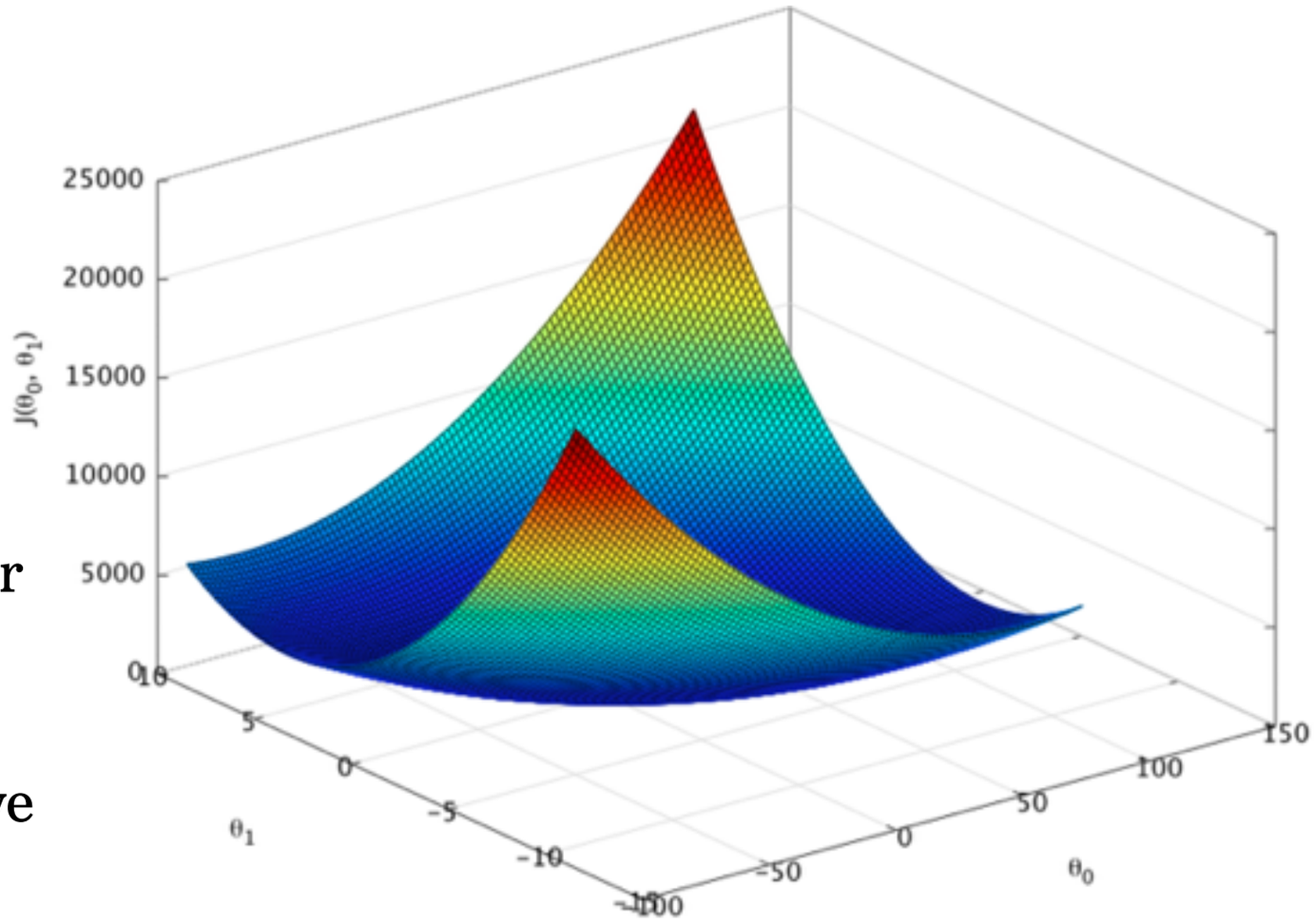
Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Each θ pair is associated with a sum of squared errors value (left axis). As we land on a point on the landscape, we can use the derivatives "see" which direction we can move to decrease our total cost.

The size of our step parameter tells us how far we will "jump" in the downward direction. We may end up either higher or lower when we actually land.

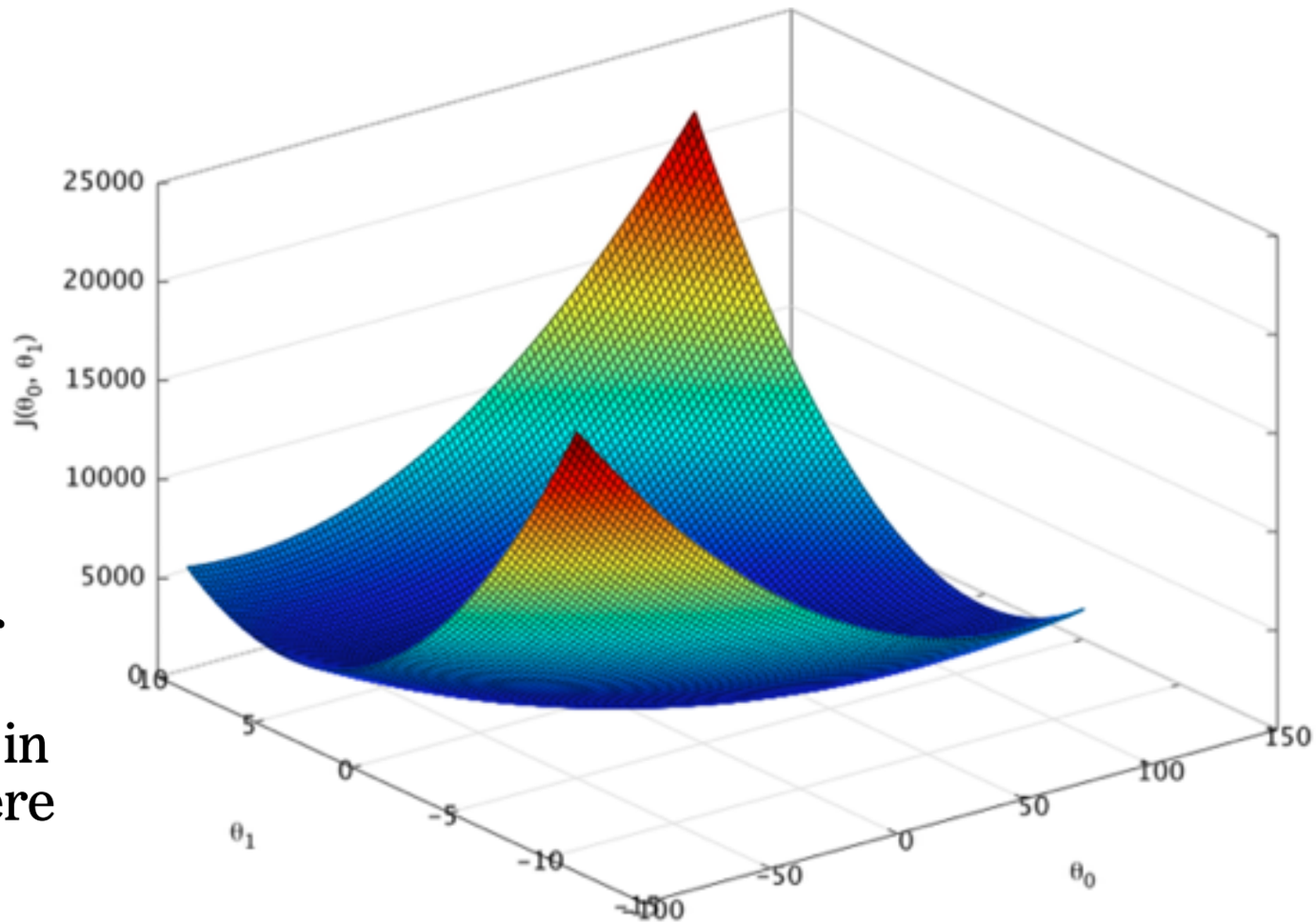


Modeling Conceptually

When we land, we will have a new pair of coefficients and a new sum of squared errors.

We then repeat the entire process until we get to the point there is no longer a "down" anymore from where we end up.

It may be that there is a lower level, but our derivatives — "our down indicator" say that in every direction around us, there is only up or flat.



Ok, great.

So, how does this whole regularization L1 and L2 stuff fit in to this?

Modeling Conceptually

So glad you asked.

The goal of regularization is to make your model worse.

The goal of regularization is to make your model worse.

But Alex, that sounds like a terrible idea. I want an amazing model! I want to feel about my model like Kayne would feel about Kayne's model if Kayne had a model.

Modeling Conceptually

Here's the deal.

A model has to fit data that is not in the training set.

Modeling Conceptually

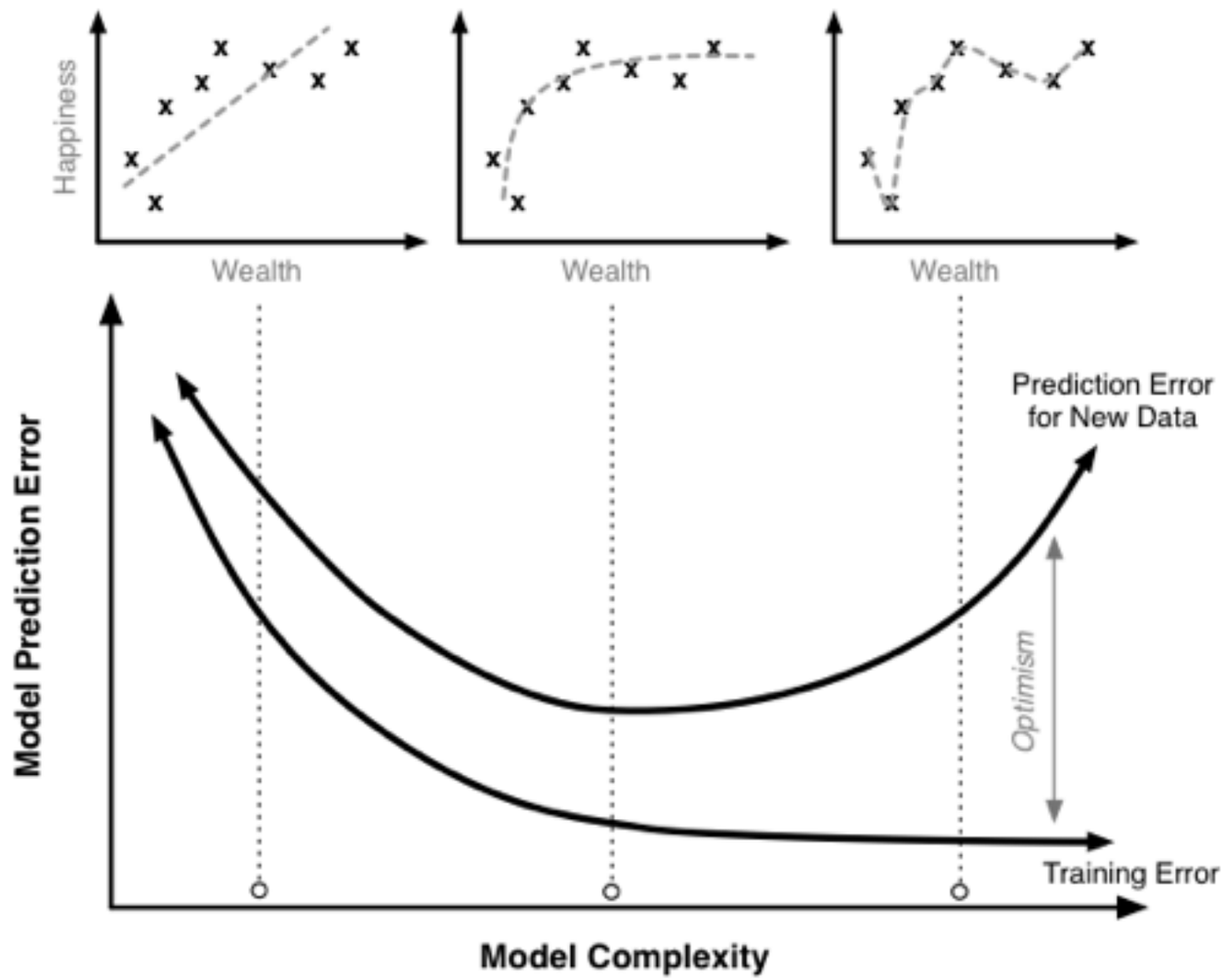
Here's the deal.

A model has to fit data that is not in the training set.

With enough features, you could have nearly 0 error on your training set, but when you take that model and apply it to your test set — and any new data, the model will be lousy.

It will be lousy because you fit the noise instead of the signal.

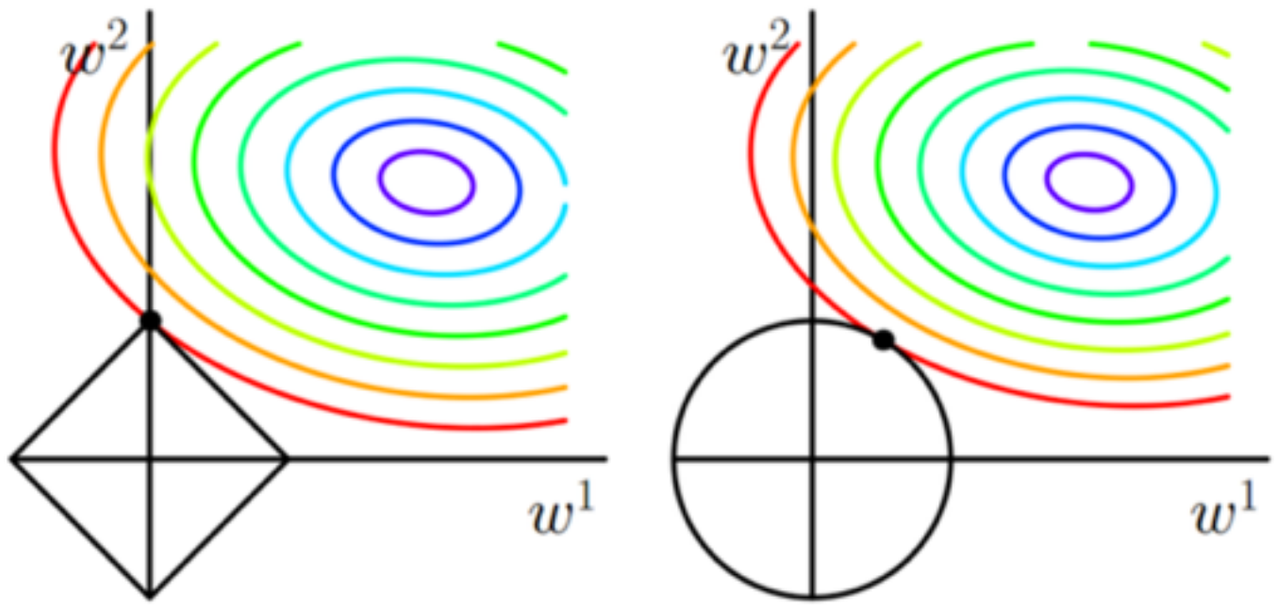
Modeling Conceptually



But, L1 and L2.

Like what is that, exaaactly?

On the axes are the coefficients. The circles represent the different levels of the cost function. Each color represents an equal level of SSE. The center is the lowest level. Our perfectly fit model would have the w1, w2 coefficients of that point. To avoid overfitting our model, we require a solution that falls within the diamond or the circle. We take the coefficients that result from the intersection of the two points.



(a) ℓ_1-ball meets quadratic function. ℓ_1-ball has corners. It's very likely that the meet-point is at one of the corners.

(b) ℓ_2-ball meets quadratic function. ℓ_2-ball has no corner. It is very unlikely that the meet-point is on any of axes.

But where did those two shapes come from?

Modeling Conceptually

These are the unit circles for the Lp norms. L1 is a diamond and L2 is a circle.

Every (x, y) point has a distance to the origin of exactly 1.

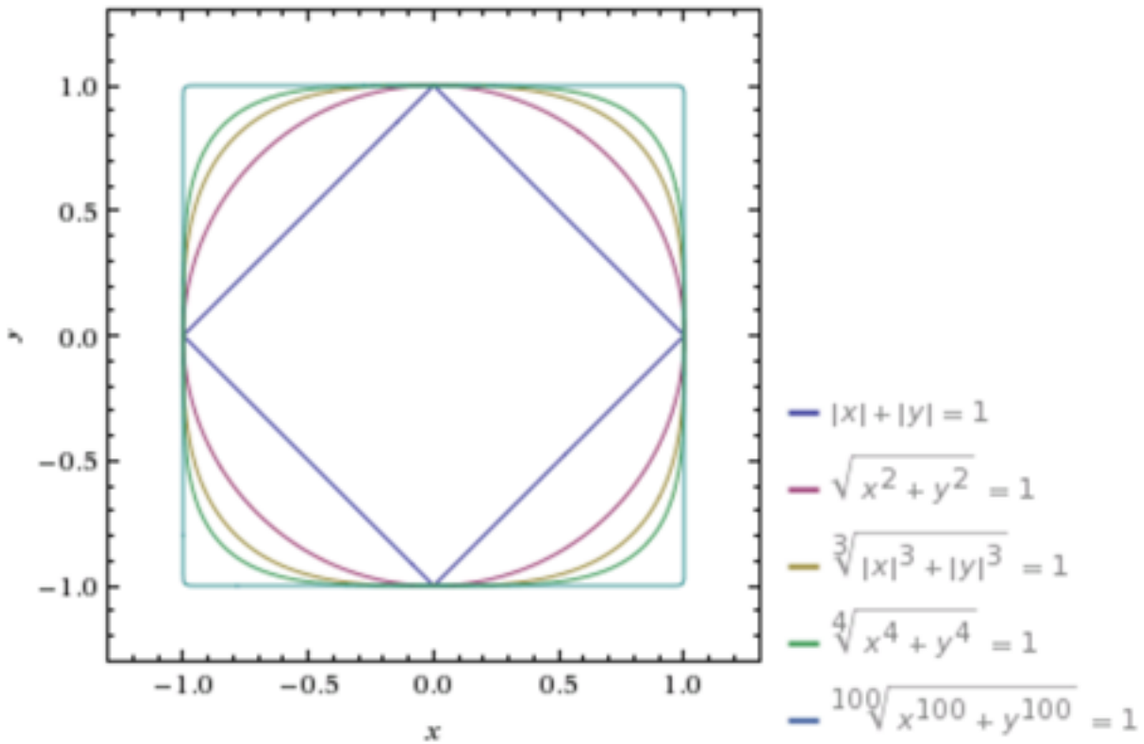
When we use regularization, we impose the constraint that the cost function/loss function will be minimized at a point in this circle.

The result is that the coefficients are close to 0.

In general, the p-norm is:

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

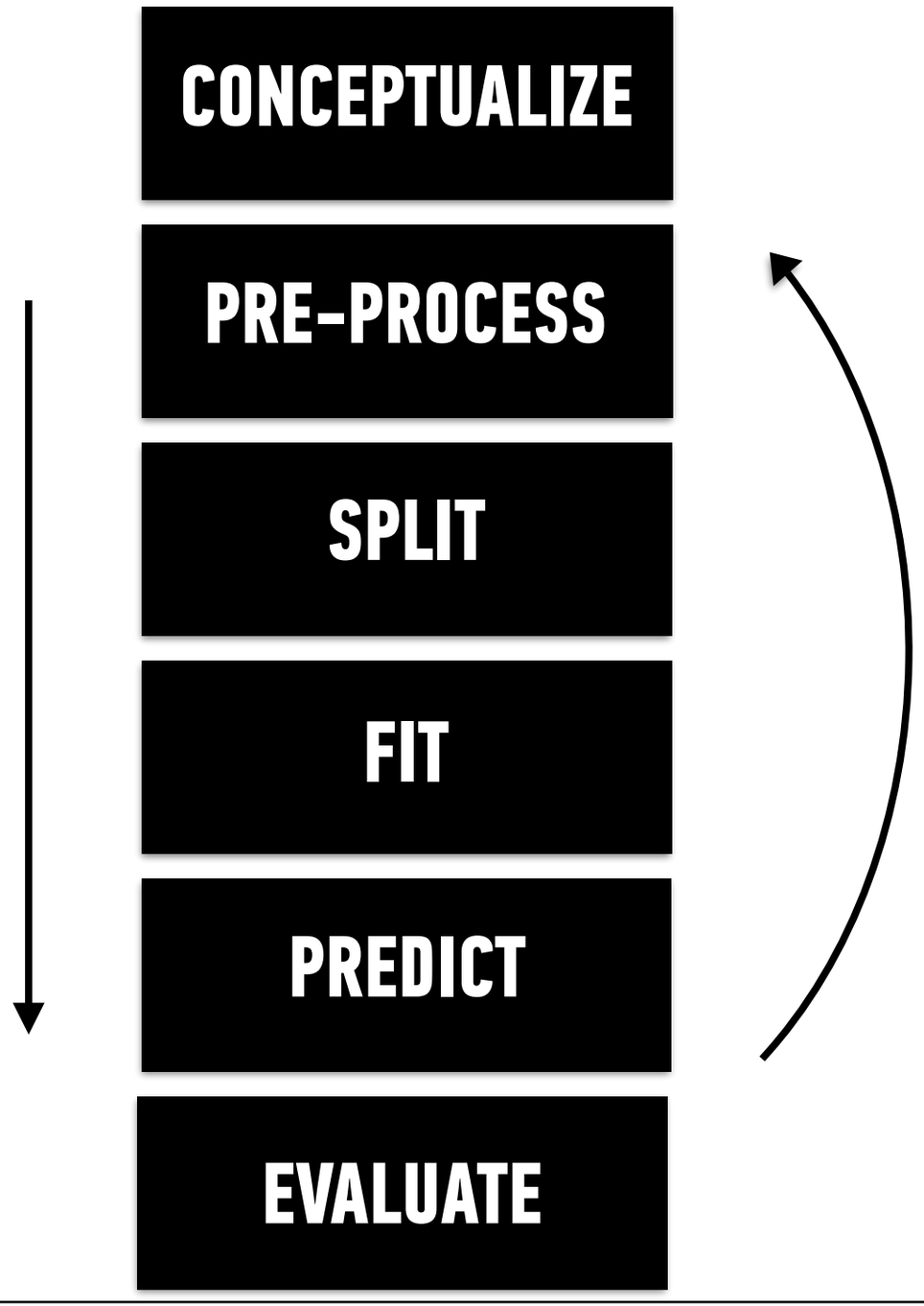
And looks like this:



Swell.

Now can you explain what that means in terms of terms of how I implement this in scikit-learn?

MODELING PRACTICALLY


CONCEPTUALIZE
PRE-PROCESS
SPLIT
FIT
PREDICT
EVALUATE

Modeling Practically

In scikit-learn, to fit a model, we do the following:

1. Select a model
2. Preprocess your features if appropriate for data and model: .StandardScaler()
3. Partition data into train and test sets: .train_test_split()
4. Fit the data on the training set: .fit(X_train, y_train)
5. Call predict on test set: .predict(X_test)
6. The output of that is a vector of your predictions
7. Use a metric to score your prediction y's vs. the actual y's:
   r2_score(y, y_pred)
8. Consider the delta between train and test performance, if
   it is large, perhaps you need to add regularization: LassoCV or RidgeCV

But, Gradient Descent!

When do I descend!?

Modeling Practically

Mostly, you don’t.

Mostly, you don't.

It happens behind the scenes to optimize your algorithms.

(It happens when you call fit.)

FIN.

QUESTIONS?