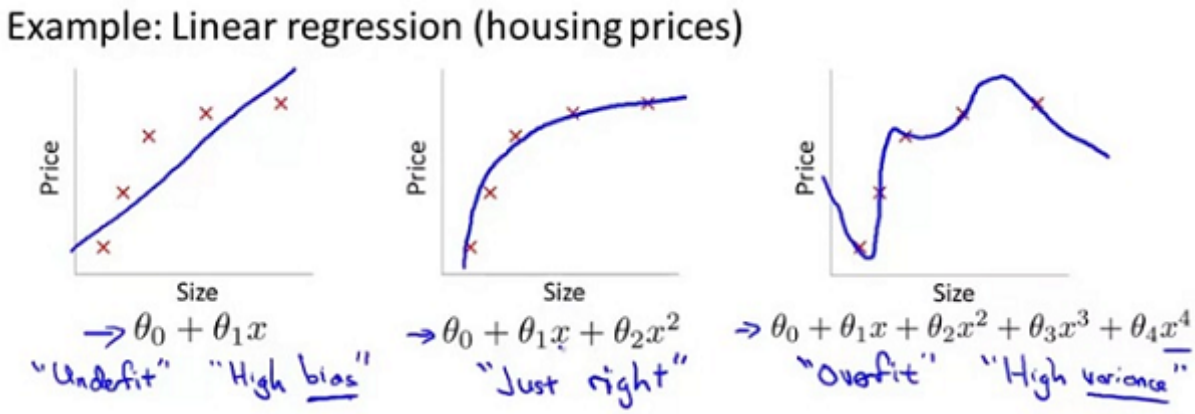
The Problem of Overfitting

What is overfitting?, Let's keep using our running example of predicting housing prices with linear regression where we want to predict the price as a function of the size of the house.



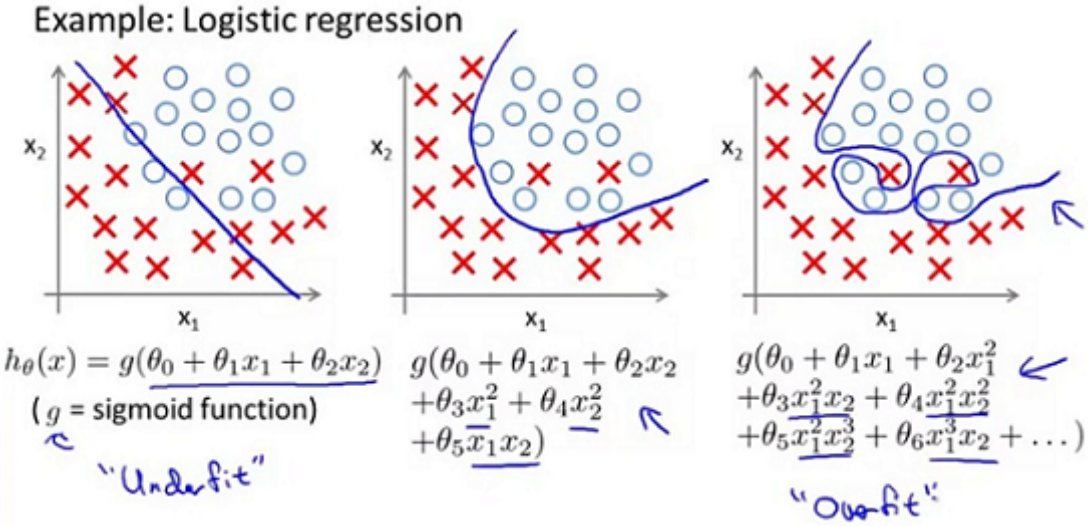
On the first plot of the data, one thing we could do is fit a linear function to this data, and if we do that, maybe we get that sort of straight line fit to the data. But this isn't a very good model. Looking at the data, it seems pretty clear that as the size of the housing increases, the housing prices plateau, or kind of flattens out as we move to the right and so this algorithm does not fit the training and we call this problem **underfitting**, and another term for this is that this algorithm has **high bias**.

The idea is that if a fitting a straight line to the data, then, it's as if the algorithm has a very strong preconception, or a **very strong bias** that housing prices are going to vary linearly with their size and despite the data to the contrary. Despite the evidence of the contrary is preconceptions still are bias, still closes it to fit a straight line and this ends up being a poor fit to the data.

Now, in the middle, we could fit a quadratic functions enter and, with this data set, we fit the quadratic function, and we got a curve, that works pretty well for the data. On the one hand, the third example seems to do a very good job fitting the training set and, that is processed through all of our data, at least. But, this is still a very wiggly curve, So, it's going up and down all over the place, and, we don't actually think that's such a good model for predicting housing prices. So, this problem we call **overfitting**, and, another term for this is that this algorithm has **high variance**.

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$), but fail to generalize to new examples (predict prices on new examples (the term generalized refers to how well a hypothesis applies even to new examples. That is to data to houses that it has not seen in the training set).

A similar problem (underfitting or overfitting) can apply to logistic regression as well



So, once again, the first plot is an example of **underfitting** or of the hypothesis having **high bias**. And, finally, at the third example, if we were to generate lots of high-order polynomial terms of features, then, logistical regression may contort itself, may try really hard to find a decision boundary that fits our training data or go to great lengths to contort itself, to fit every single training example well.

If the features x_1 and x_2 offer predicting, the cancer, concretely if cancer is a malignant, benign breast tumors. This really doesn't look like a very good hypothesis, for making predictions. And so, once again, this is an instance of overfitting and, of a hypothesis having high variance and, being unlikely to generalize well to new examples.

Video Question: Consider the medical diagnosis problem of classifying tumors as malignant or benign. If a hypothesis $h_\theta(x)$ has overfit the training set, it means that:

- It makes accurate predictions for examples in the training set and generalizes well to make accurate predictions on new, previously unseen examples.
- It does not make accurate predictions for examples in the training set, but it does generalize well to make accurate predictions on new, previously unseen examples.

> It makes accurate predictions for examples in the training set, but it does not generalize well to make accurate predictions on new, previously unseen examples.

- It does not make accurate predictions for examples in the training set and does not generalize well to make accurate predictions on new, previously unseen examples.

Addressing overfitting:

If we think overfitting is occurring, what can we do to address it? Plotting the hypothesis, could be one way to try to decide what degree polynomial to use. But that doesn't always work. And, in fact more often we may have learning problems that where we just have a lot of features. And there is not just a matter of selecting what degree polynomial. And, in fact, when we have so many features, it also becomes much harder to plot the data and it becomes much harder to visualize it, to decide what features to keep or not.

$$
\begin{aligned}
x_1 &= \text{size of house} \\
x_2 &= \text{no. of bedrooms} \\
x_3 &= \text{no. of floors} \\
x_4 &= \text{age of house} \\
x_5 &= \text{average income in neighborhood} \\
x_6 &= \text{kitchen size} \\
&\vdots \\
x_{100}
\end{aligned}
$$

So concretely, if we're trying predict housing prices sometimes we can just have a lot of different features. And all of these features seem kind of useful. But, if we have a lot of features, and, very little training data, then, over fitting can become a problem.

In order to address over fitting, there are two main options for things that we can do.

1) Reduce the number of features:

- Manually select which features to keep.
- Use a model selection algorithm (studied later in the course).

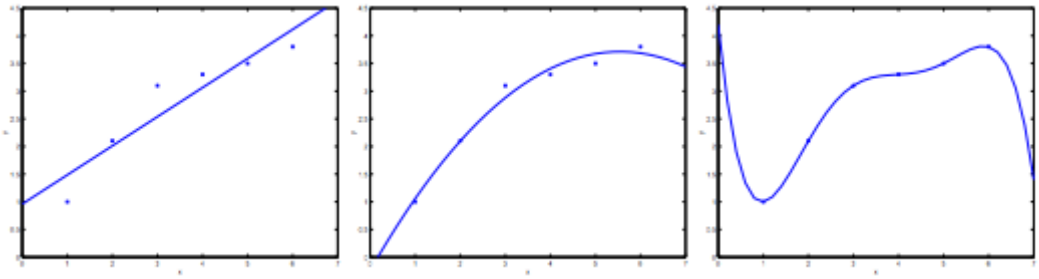
This idea of reducing the number of features can work well, and, can reduce overfitting. But, the disadvantage is that, by throwing away some of the features, is also throwing away some of the information we have about the problem.

2) Regularization

- Keep all the features, but reduce the magnitude/values of parameters θ_j.
- Works well when we have a lot of features, each of which contributes a bit to predicting y.

Summary

Consider the problem of predicting y from $x \in \mathbb{R}$. The leftmost figure below shows the result of fitting a $y = \theta_0 + \theta_1 x$ to a dataset. We see that the data doesn't really lie on straight line, and so the fit is not very good.



Instead, if we had added an extra feature x^2, and fit $y = \theta_0 + \theta_1 x + \theta_2 x^2$, then we obtain a slightly better fit to the data (See middle figure). Naively, it might seem that the more features we add, the better. However, there is also a danger in adding too many features: The rightmost figure is the result of fitting a 5^{th} order polynomial $y = \sum_{j=0}^{5} \theta_j x^j$. We see that even though the fitted curve passes through the data perfectly, we would not expect this to be a very good predictor of, say, housing prices (y) for different living areas (x). Without formally defining what these terms mean, we'll say the figure on the left shows an instance of **underfitting**—in which the data clearly shows structure not captured by the model—and the figure on the right is an example of **overfitting**.

This terminology is applied to both linear and logistic regression. There are two main options to address the issue of overfitting:

1) Reduce the number of features:

- Manually select which features to keep.
- Use a model selection algorithm (studied later in the course).

2) Regularization

- Keep all the features, but reduce the magnitude of parameters θ_j.
- Regularization works well when we have a lot of slightly useful features.