GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
LECTURE 4: REGRESSION & REGULARIZATION

LAST TIME:

- REVIEW KNN
- LINEAR REGRESSION

QUESTIONS?

I. REVIEW REGRESSION
II. REGULARIZATION
III. LOGISTIC REGRESSION

I. LINEAR REGRESSION

	continuous	*categorical*
supervised	regression	classification
unsupervised	dimension reduction	clustering

Q: What is a **regression** model?

A: A functional relationship between input & response variables

$$y = \alpha + \beta_1 x_1 + \dots + \beta_n x_n + \varepsilon$$

Q: What do the terms in this model mean?

$$y = \alpha + \beta_1 x_1 + \ldots + \beta_n x_n + \varepsilon$$

A: y = **response variable** (the one we want to predict)

x = **input variable** (the one we use to train the model)

α = **intercept** (where the line crosses the y-axis)

β = **regression coefficients** (the model "parameters")

ε = **residual** (the prediction error)

Q: What problems have we seen?

A:

1) Correlated predictor variables
2) Large number of parameters allow us to overfit

Q: What can we do about this?

A: If prediction is our only goal – nothing.

Q: What can we do about this?

A: If prediction is our only goal – nothing.

Otherwise,

1) Drop correlated predictors
2) Get more data

II: POLYNOMIAL REGRESSION

Consider the following **polynomial regression** model:

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \varepsilon$$

Consider the following **polynomial regression** model:

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \varepsilon$$

Q: This represents a nonlinear relationship. Is it still a linear model?

Consider the following **polynomial regression** model:

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \varepsilon$$

Q: This represents a nonlinear relationship. Is it still a linear model?

A: Yes, because it's linear in the β's!

Polynomial regression allows us to fit very complex curves to data.

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n + \varepsilon$$

Polynomial regression allows us to fit very complex curves to data.

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n + \varepsilon$$

But there is one problem with the model we've written down so far.

This model displays **multicollinearity**, which means the predictor variables are highly correlated with each other.

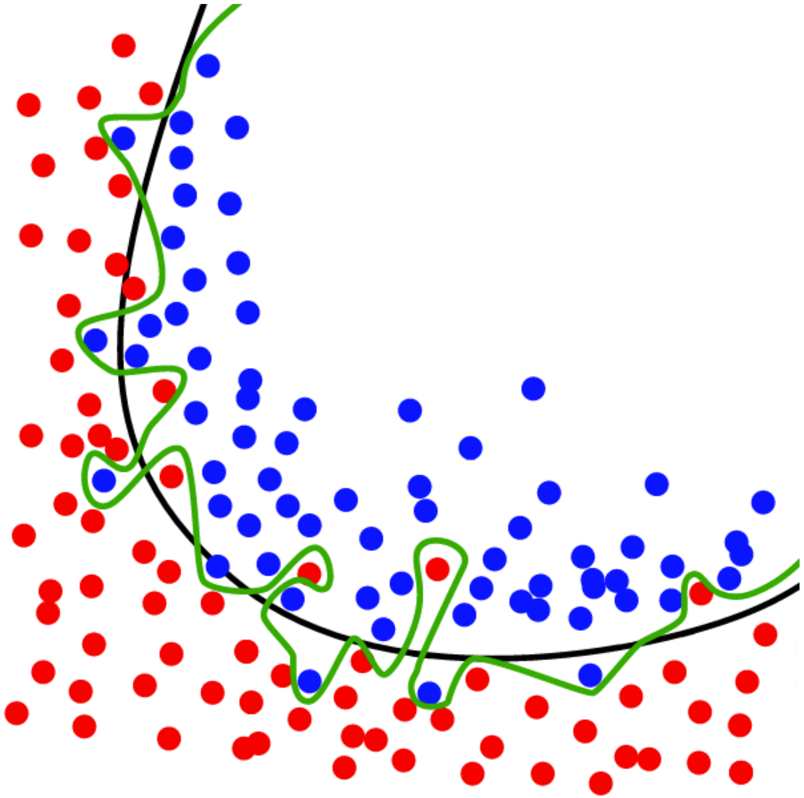
$$y = \alpha + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n + \varepsilon$$

```
> x <- seq(1, 10, 0.1)
> cor(x^9, x^10)
[1] 0.9987608
```

Q: Can a regression model be too complex?

III: REGULARIZATION

Recall our earlier discussion of **overfitting**.



source: http://upload.wikimedia.org/wikipedia/commons/1/19/Overfitting.svg

Recall our earlier discussion of **overfitting**.

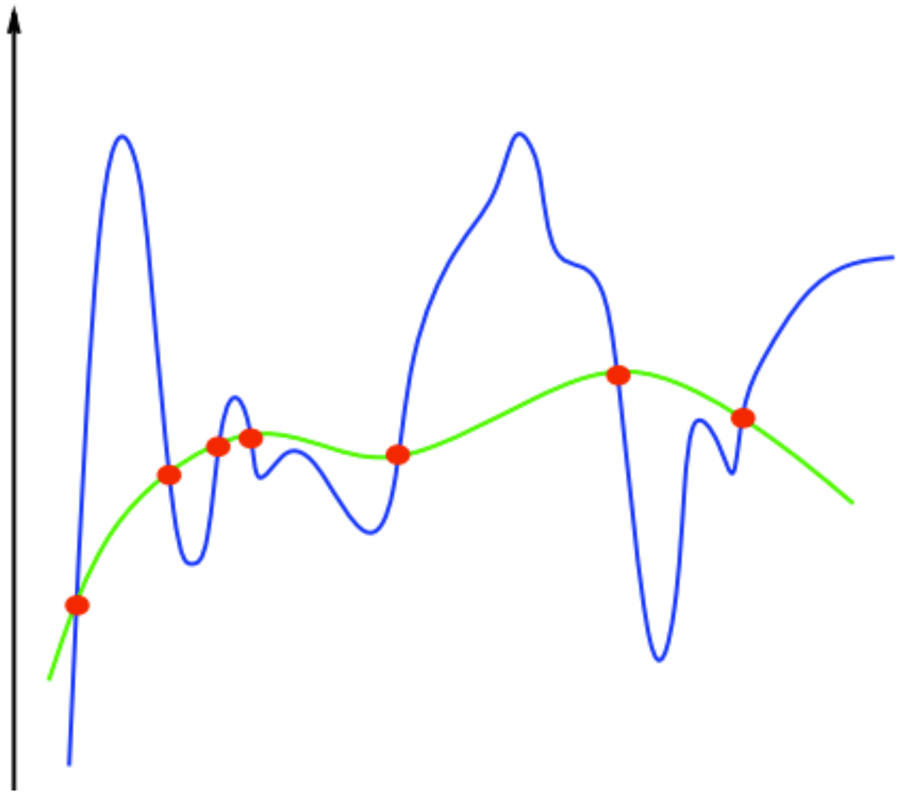
In other words, an overfit model matches the **noise** in the dataset instead of the **signal**.

The same thing can happen in regression.

It's possible to design a regression model that matches the noise in the data instead of the signal.

This happens when our model becomes *too complex* for the data to support.

OVERFITTING EXAMPLE (REGRESSION)



source: http://www.mit.edu/~9.520/spring12/slides/class02/class02.pdf

Q: How do we define the **complexity** of a regression model?

Q: How do we define the **complexity** of a regression model?

A: One method is to define complexity as a function of the size of the coefficients.

Q: How do we define the **complexity** of a regression model?

A: One method is to define complexity as a function of the size of the coefficients.

Ex 1: $\Sigma\,|\beta_i|$

Ex 2: $\Sigma\,{\beta_i}^2$

Q: How do we define the **complexity** of a regression model?

A: One method is to define complexity as a function of the size of the coefficients.

Ex 1: $\Sigma\,|\beta_i|$ this is called the **L1-norm**

Ex 2: $\Sigma\,{\beta_i}^2$ this is called the **L2-norm**

These measures of complexity lead to the following **regularization** techniques:

These measures of complexity lead to the following **regularization** techniques:

L1 regularization: $y = \Sigma\, \beta_i x_i + \varepsilon \quad st. \quad \Sigma\, |\beta_i| < s$

These measures of complexity lead to the following **regularization** techniques:

L1 regularization: $y = \Sigma\, \beta_i x_i + \varepsilon \quad st. \quad \Sigma\, |\beta_i| < s$

L2 regularization: $y = \Sigma\, \beta_i x_i + \varepsilon \quad st. \quad \Sigma\, \beta_i^2 < s$

These measures of complexity lead to the following **regularization** techniques:

L1 regularization: $y = \Sigma\, \beta_i x_i + \varepsilon \quad st. \quad \Sigma\, |\beta_i| < s$

L2 regularization: $y = \Sigma\, \beta_i x_i + \varepsilon \quad st. \quad \Sigma\, \beta_i^2 < s$

Regularization refers to the method of preventing **overfitting** by explicitly controlling model **complexity**.

These measures of complexity lead to the following **regularization** techniques:

Lasso **regularization**: $y = \Sigma\, \beta_i x_i + \varepsilon \quad st. \quad \Sigma\, |\beta_i| < s$

Ridge **regularization**: $y = \Sigma\, \beta_i x_i + \varepsilon \quad st. \quad \Sigma\, \beta_i^2 < s$

Regularization refers to the method of preventing **overfitting** by explicitly controlling model **complexity**.

These regularization problems can also be expressed as:

L1 regularization: $min(\|y - x\beta\|^2 + \lambda\|x\|)$

L2 regularization: $min(\|y - x\beta\|^2 + \lambda\|x\|^2)$

These regularization problems can also be expressed as:

OLS: $min(\|y - x\beta\|^2)$

L1 regularization: $min(\|y - x\beta\|^2 + \lambda\|x\|)$

L2 regularization: $min(\|y - x\beta\|^2 + \lambda\|x\|^2)$

We are no longer just minimizing error but also an additional term.

IV: LOGISTIC REGRESSION

	continuous	*categorical*
supervised	regression	classification
unsupervised	dimension reduction	clustering

Q: What is **logistic regression**?

Q: What is **logistic regression**?

A: A generalization of the linear regression model to *classification* problems.

In linear regression, we used a set of covariates to predict the value of a (continuous) outcome variable.

In linear regression, we used a set of covariates to predict the value of a (continuous) outcome variable.

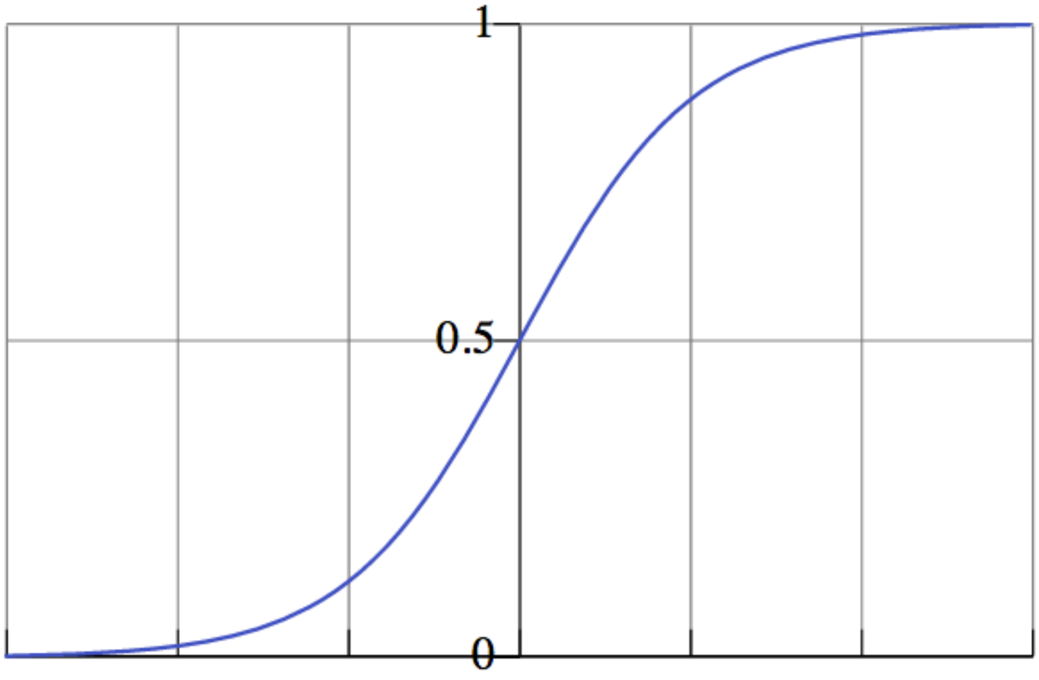
In logistic regression, we use a set of covariates to predict *probabilities* of (binary) class membership.

In linear regression, we used a set of covariates to predict the value of a (continuous) outcome variable.

In logistic regression, we use a set of covariates to predict *probabilities* of (binary) class membership.

These probabilities are then mapped to *class labels*, thus solving the classification problem.

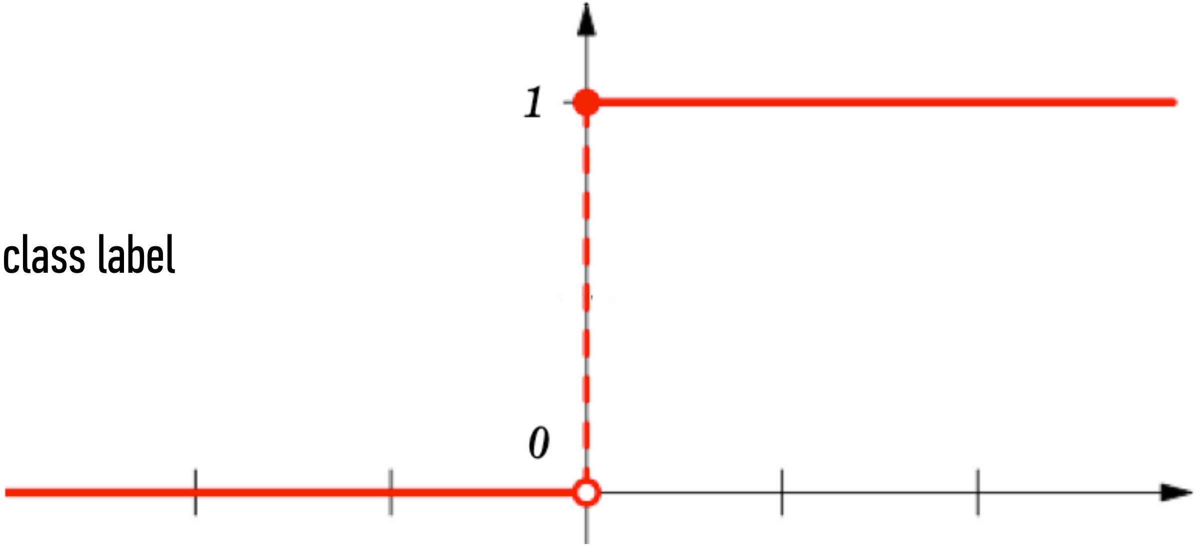
probability of belonging to class



value of independent variable

NOTE

Probability predictions look like this.



class label

value of independent variable

NOTE

Probabilities are "snapped" to class labels (eg by threshholding at 50%).

The logistic regression model is an *extension* of the linear regression model, with a couple of important differences.

The logistic regression model is an *extension* of the linear regression model, with a couple of important differences.

The first difference is in the outcome variable.

The logistic regression model is an *extension* of the linear regression model, with a couple of important differences.

The main difference is in the outcome variable.

The key variable in any regression problem is the **response type** of the outcome variable y given the value of the covariate x:

$$E(y|x)$$

The key variable in any regression problem is the **conditional mean** of the outcome variable y given the value of the covariate x:

$$E(y|x)$$

In linear regression, we assume that this conditional mean is a linear function taking values in $(-\infty, +\infty)$:

$$E(y|x) = \alpha + \beta x$$

In logistic regression, we've seen that the conditional mean of the outcome variable takes values only in the unit interval $[0, 1]$.

In logistic regression, we've seen that the conditional mean of the outcome variable takes values only in the unit interval $[0, 1]$.

The first step in extending the linear regression model to logistic regression is to map the outcome variable $E(y|x)$ into the unit interval.

In logistic regression, we've seen that the conditional mean of the outcome variable takes values only in the unit interval $[0, 1]$.

The first step in extending the linear regression model to logistic regression is to map the outcome variable $E(y|x)$ into the unit interval.

Q: How do we do this?

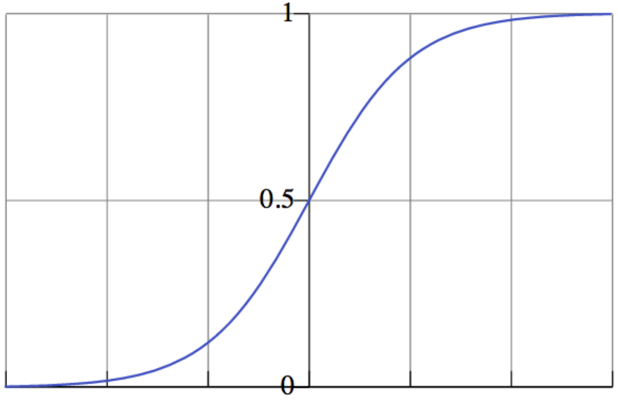
A: By using a transformation called the **logistic function**:

$$E(y|x) = \pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

A: By using a transformation called the **logistic function**:

$$E(y|x) = \pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

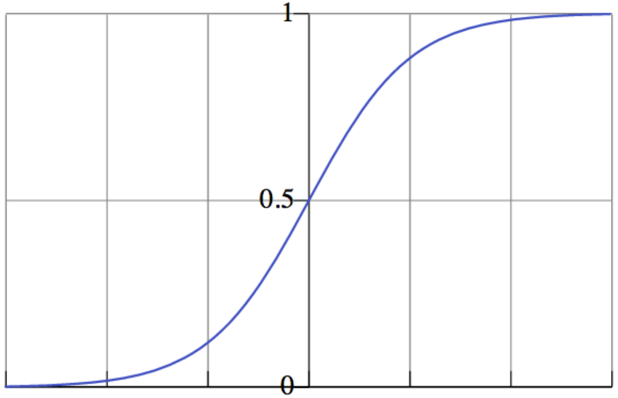
We've already seen what this looks like:



A: By using a transformation called the **logistic function**:

$$E(y|x) = \pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

We've already seen what this looks like:



NOTE

For any value of x, y is in the interval [0, 1]

This is a nonlinear transformation!

The **logit function** is an important transformation of the logistic function. Notice that it returns the linear model!

$$g(x) = ln(\frac{\pi(x)}{1-\pi(x)}) = \alpha + \beta x$$

The **logit function** is an important transformation of the logistic function. Notice that it returns the linear model!

$$g(x) = ln(\frac{\pi(x)}{1-\pi(x)}) = \alpha + \beta x$$

The logit function is also called the **log-odds function**.

The **logit function** is an important transformation of the logistic function. Notice that it returns the linear model!

$$g(x) = ln(\frac{\pi(x)}{1-\pi(x)}) = \alpha + \beta x$$

NOTE

This name hints at its usefulness in interpreting our results.

We will see why shortly.

The logit function is also called the **log-odds function**.