GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

LECTURE 5: REGRESSION AND REGULARIZATION

Paul Burkard

07/07/2015

LAST TIME:

- WHAT IS WEB SCRAPING?
    - HOW DO WE DO IT IN PYTHON?
    - HTML, XML, JSON, WEB APIS
- WHAT IS THE UNIX COMMAND LINE?
    - WHAT ARE SOME COMMON COMMANDS?

TODAY:

I. LINEAR REGRESSION

II. MODEL EVALUATION: CROSS-VALIDATION

III. REGULARIZATION

HANDS-ON: LINEAR REGRESSION AND REGULARIZATION

- What is **Linear Regression**?
  - What are the inputs and outputs?
  - What are some potential use cases?
- What is **Overfitting**?
  - How to we control for it?
  - What is **Cross-Validation**?
  - What is **Regularization**?
- Intro to **sklearn**, **patsy**, and **statsmodels**

I. LINEAR REGRESSION

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

A: A functional relationship between input & response variables

The **simple linear regression** *model captures a linear relationship between a single input variable x and a response variable y:*

$$y = \alpha + \beta x + \varepsilon$$

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 coefficient***s (the model "parameters")*

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

	continuous	*categorical*
supervised	???	???
unsupervised	???	???

	continuous	categorical
supervised	regression	classification
unsupervised	dim reduction	clustering

ASIDE: LINEAR ALGEBRA INTRO

We can extend this model to several input variables, giving us the **multiple linear regression** *model:*

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \\ \vdots \\ \mathbf{x}_n^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

II. CROSS-VALIDATION

	continuous	categorical
supervised	regression	classification
unsupervised	dim reduction	clustering

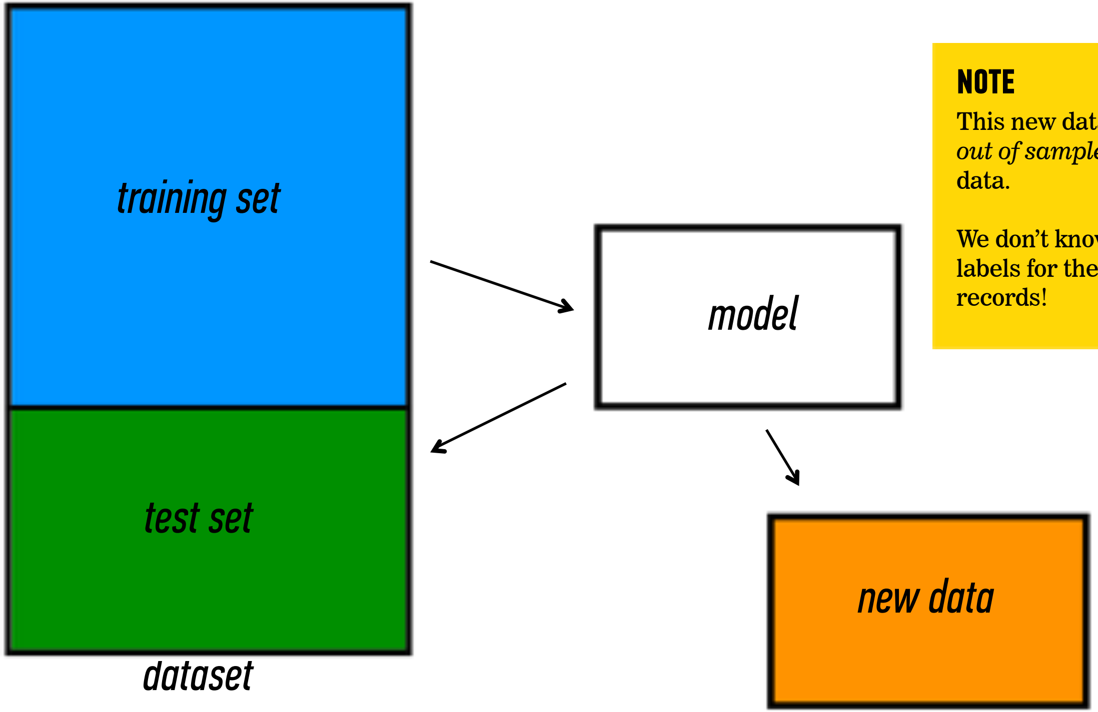
NOTE

Remember!

Regression is a supervised learning problem!

Q: What steps does a supervised learning problem require?

1) *split dataset*
2) *train model*
3) *test model*
4) *make predictions*



NOTE

This new data is called *out of sample* data.

We don't know the labels for these OOS records!

Q: What can go wrong if we don't follow these steps?

A: ***Overfitting!***

- *If we test our model against the training set it might perform quite well on the training set, but fail to* ***generalize*** *to new data*
- *The model might be overly* ***complex*** *and tailored to the training data*

OVERFITTING

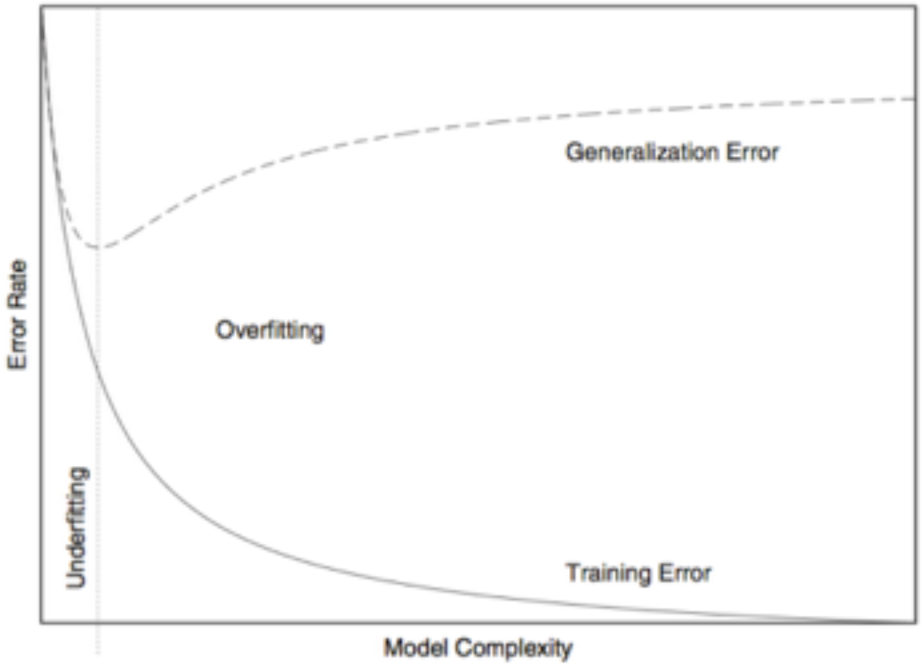
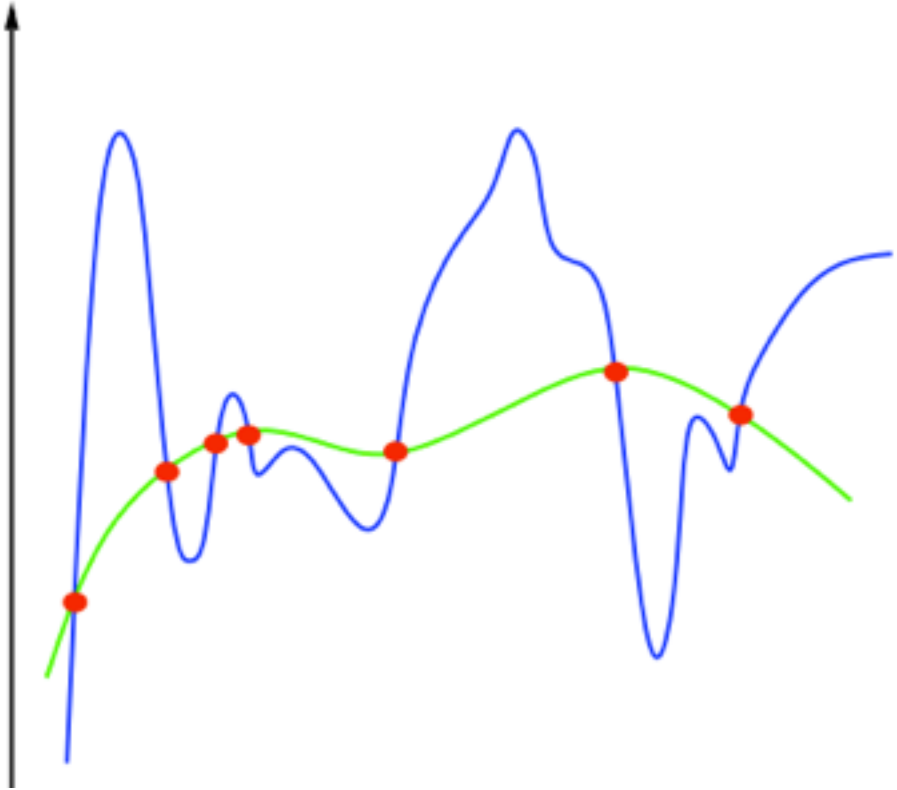


FIGURE 18-1. Overfitting: as a model becomes more complex, it becomes increasingly able to represent the training data. However, such a model is overfitted and will not generalize well to data that was not used during training.

source: Data Analysis with Open Source Tools, by Philipp K. Janert. O'Reilly Media, 2011.

OVERFITTING EXAMPLE (REGRESSION)



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

Q: How can we avoid overfitting?

*A: One way is **Cross-Validation***

- *Pre-splitting the dataset into train/test sets is one form of cross-validation*
- *There are plenty of others (**k-fold**, **leave-one-out**, etc)*

*Steps of **k-fold Cross-Validation**:*

- *Partition dataset into k random, equal-sized subsets*
- *For each subset, hold it out as the test set and train on the rest*
- *Report the average of the testing performances as the model's estimated generalization performance*

III. REGULARIZATION

Q: What is regularization?

*A: Any built-in method to **reduce complexity** of a model in an effort to **lower the risk of overfitting***

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**

*The basic **Ordinary Least Squares** solution to regression problems can also be expressed as:*

OLS: Choose β s.t. $min(\|y - x\beta\|^2)$

*Here, the function in parenthesis is called the **Cost Function** and in general it is what you want to minimize when searching for solutions to machine learning problems.*

Thus, the regularization problems can be expressed as:

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

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

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

- *We are no longer just minimizing error but also an additional term.*
- *Thus, large values of* β *will be discouraged*

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***.*

Q: What problems might we see?

A:

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

Q: What can we do about this?

A:

1) Drop correlated predictors

2) Get more data

Q: Do regression models have to depend linearly on input variables?

A: NO

We can use almost any transformation of a single input variable (aka $f(x_i)$) as a separate input variable, as long as we don't mix them (aka $f(x_i, x_j)$)

Some nonlinear laws in nature:

$$F = G\frac{m_1 m_2}{r^2}$$

$$F = \frac{1}{4\pi\varepsilon_0}\frac{qQ}{r^2} = k_e\frac{qQ}{r^2},$$

$$x(t) = A\cos(\omega t + \phi),$$