GENERAL ASSEMBLY

INTRO TO DATA SCIENCE LESSON 5: REGRESSION & REGULARIZATION

DATA VISUALIZATION

QUESTIONS?

I. REVIEW SUPERVISED LEARNING
II. LINEAR REGRESSION
III. REGULARIZATION

I. SUPERVISED LEARNING

Q: How does a classification problem work?

A: Data in, predicted labels out.

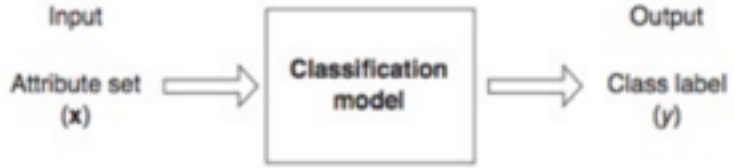
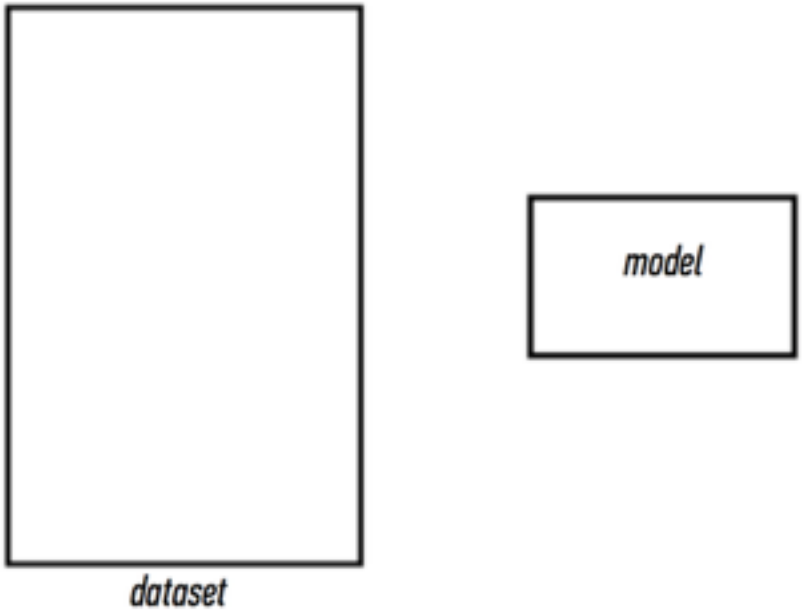


Figure 4.2. Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

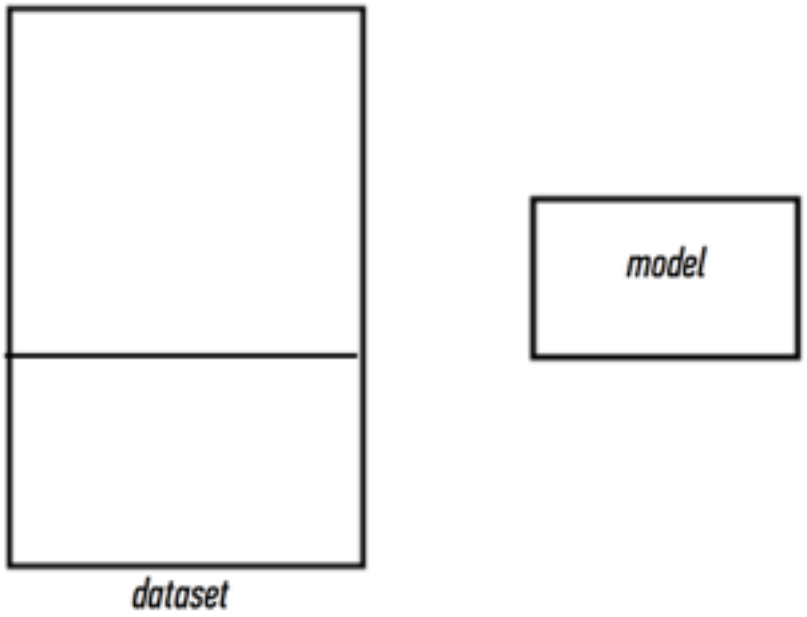
source: http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf

Q: What steps does a classification problem require?



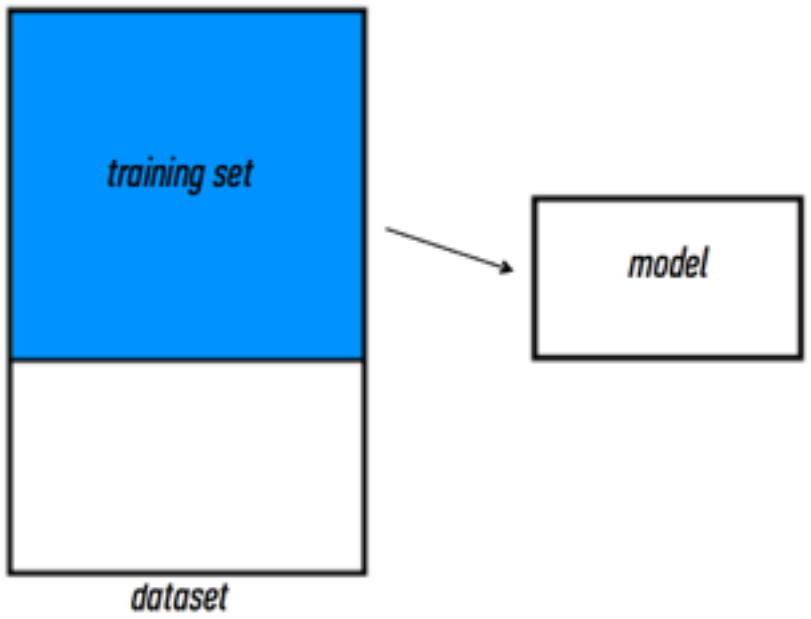
Q: What steps does a classification problem require?

1) split dataset



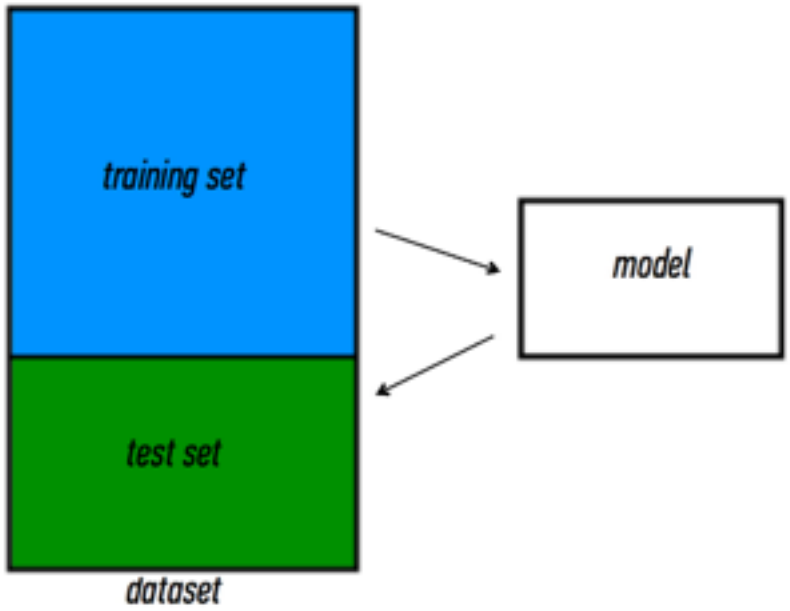
Q: What steps does a classification problem require?

1) *split dataset*
2) *train model*



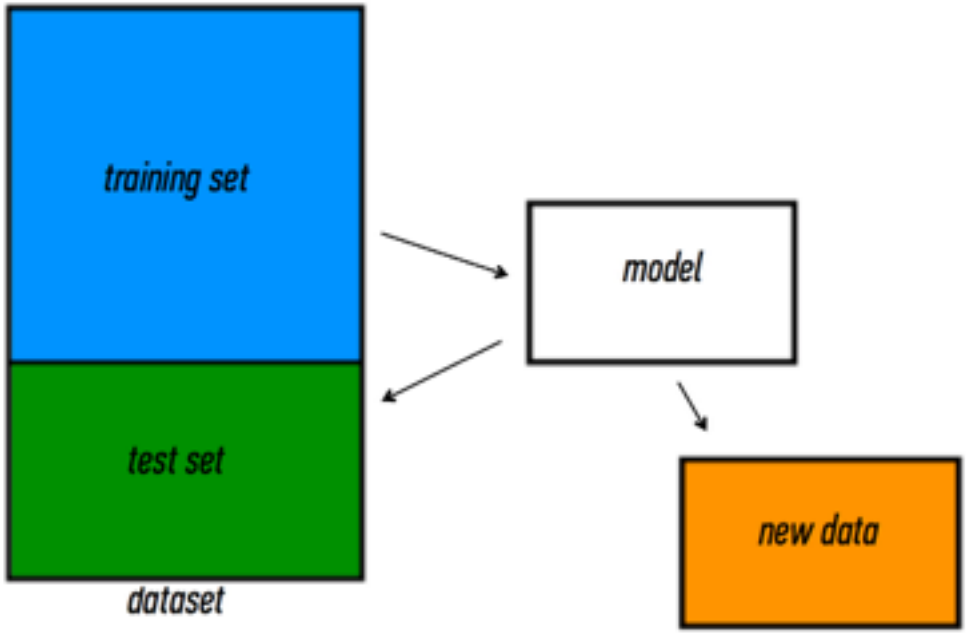
Q: What steps does a classification problem require?

1) *split dataset*
2) *train model*
3) *test model*



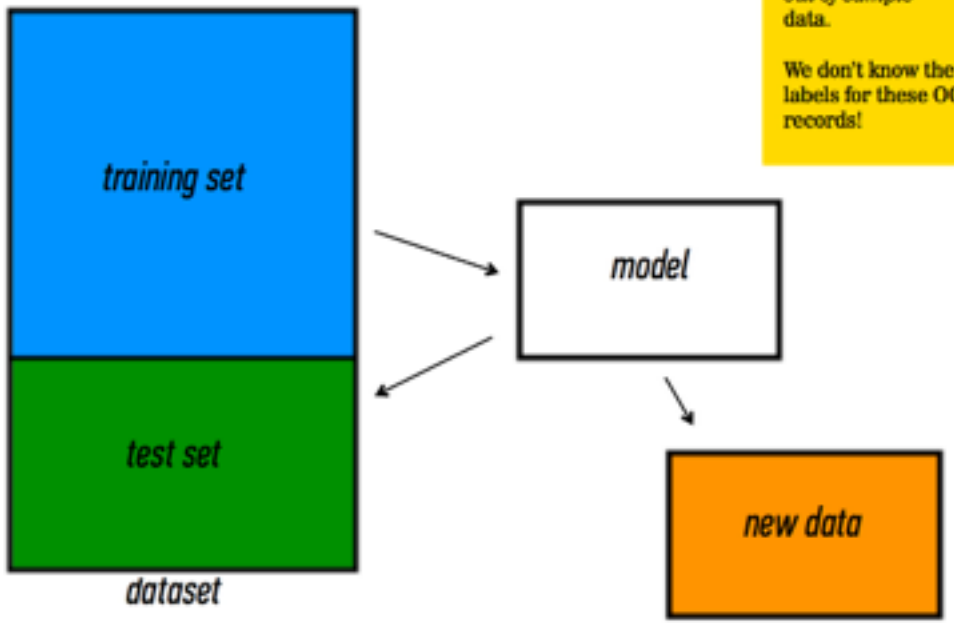
Q: What steps does a classification problem require?

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



Q: What steps does a classification 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: Why should we use training & test sets?

Thought experiment:
Suppose instead, we train our model using the entire dataset.

Q: Why should we use training & test sets?

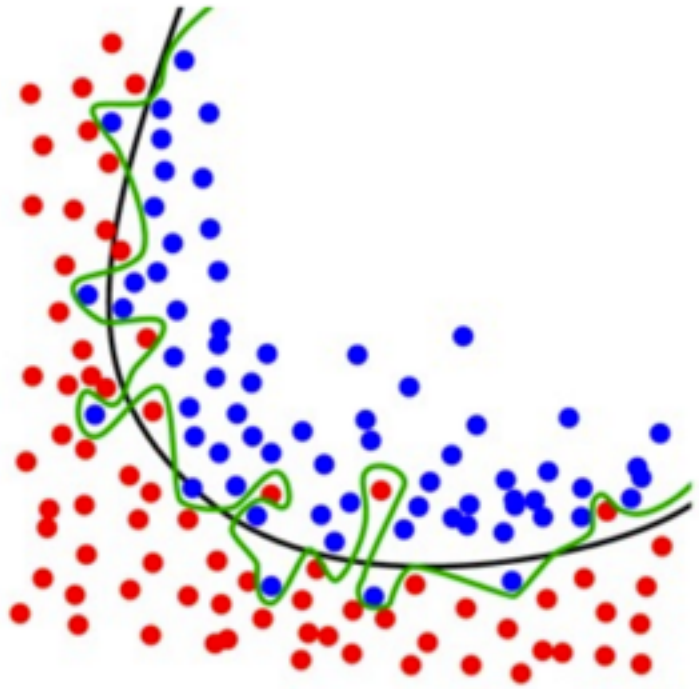
Thought experiment:
Suppose instead, we train our model using the entire dataset.
Q: How low can we push the training error?

Q: Why should we use training & test sets?

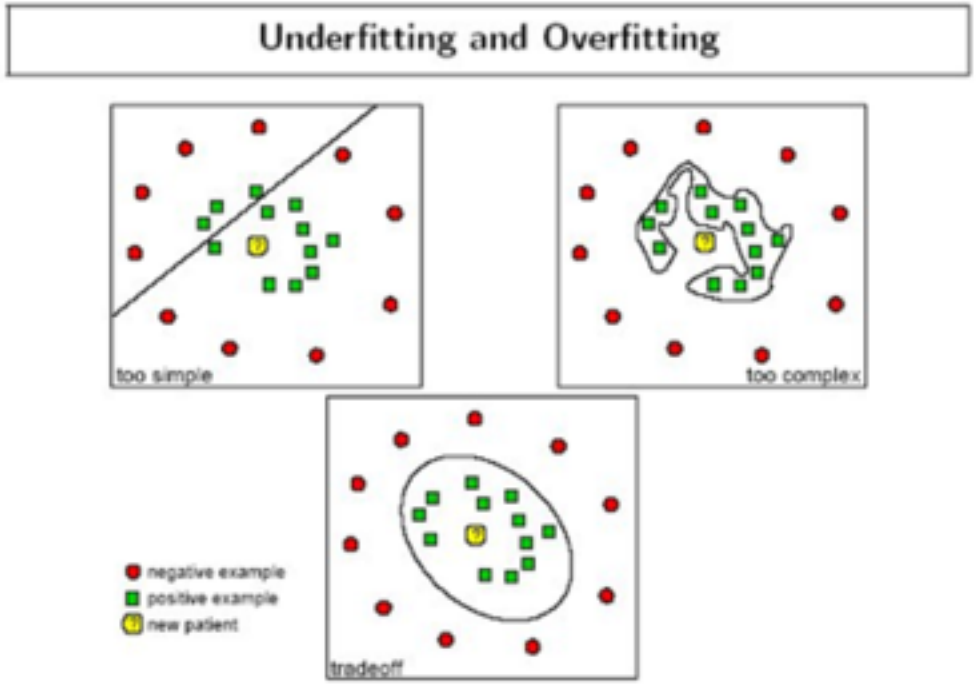
Thought experiment:
Suppose instead, we train our model using the entire dataset.
Q: How low can we push the training error?

- *We can make the model arbitrarily complex (effectively "memorizing" the entire training set).*

A: Down to zero!



source: http://www.dtreg.com



source: http://www.dtreg.com

Q: Why should we use training & test sets?

Thought experiment:
Suppose instead, we train our model using the entire dataset.
Q: How low can we push the training error?

- *We can make the model arbitrarily complex (effectively "memorizing" the entire training set).*

A: Down to zero!

NOTE

This phenomenon is called *overfitting*.

A: Training error is not a good estimate of OOS accuracy.

Suppose we do the train/test split.

Q: How well does generalization error predict OOS accuracy?

Thought experiment:

Suppose we had done a different train/test split.

Q: Would the generalization error remain the same?

Suppose we do the train/test split.

Q: How well does generalization error predict OOS accuracy?

Thought experiment:

Suppose we had done a different train/test split.

Q: Would the generalization error remain the same?

A: Of course not!

A: On its own, not very well.

Something is still missing!

Q: How can we do better?

Something is still missing!

Q: How can we do better?

Thought experiment:
Different train/test splits will give us different generalization errors.

Something is still missing!

Q: How can we do better?

Thought experiment:

Different train/test splits will give us different generalization errors.

Q: What if we did a bunch of these and took the average?

Something is still missing!

Q: How can we do better?

Thought experiment:
Different train/test splits will give us different generalization errors.
Q: What if we did a bunch of these and took the average?
A: Now you're talking!

A: Cross-validation.

Steps for n-fold cross-validation:

1) *Randomly split the dataset into n equal partitions.*

Steps for n-fold cross-validation:

1) *Randomly split the dataset into n equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*

Steps for n-fold cross-validation:

1) *Randomly split the dataset into n equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*
3) *Find generalization error.*

Steps for n-fold cross-validation:

1) *Randomly split the dataset into n equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*
3) *Find generalization error.*
4) *Repeat steps 2-3 using a different partition as the test set at each iteration.*

Steps for n-fold cross-validation:

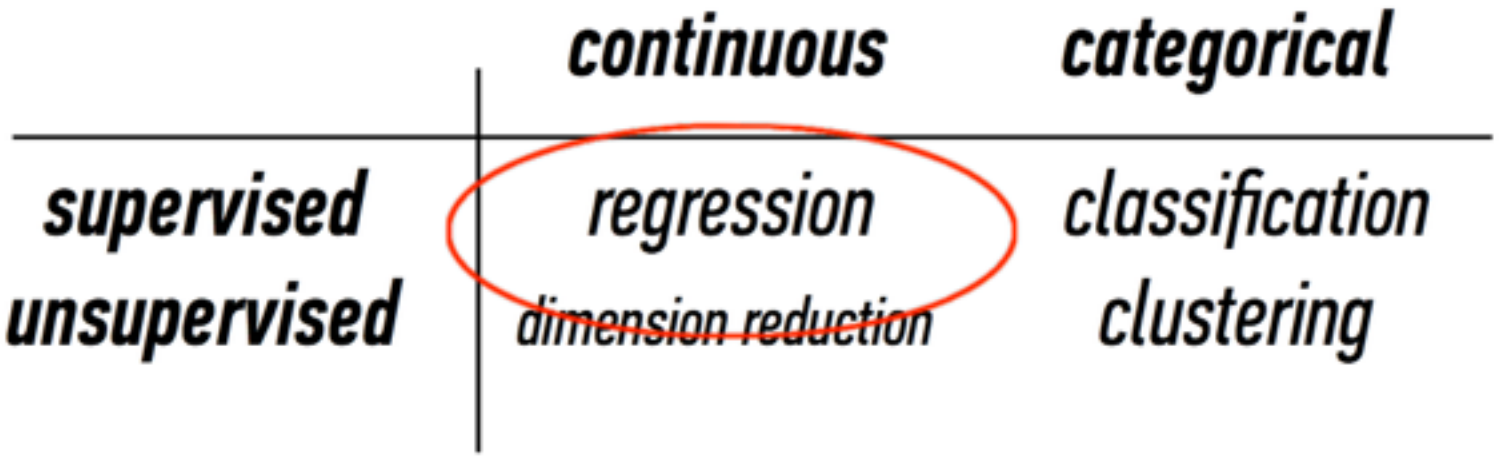
1) *Randomly split the dataset into n equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*
3) *Find generalization error.*
4) *Repeat steps 2-3 using a different partition as the test set at each iteration.*
5) *Take the average generalization error as the estimate of OOS accuracy.*

Features of n-fold cross-validation:

1) *More accurate estimate of OOS prediction error.*
2) *More efficient use of data than single train/test split.*
   - *Each record in our dataset is used for both training and testing.*
3) *Presents tradeoff between efficiency and computational expense.*
   - *10-fold CV is 10x more expensive than a single train/test split*
4) *Can be used for model selection.*

II. LINEAR REGRESSION

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


continuous
categorical
supervised
regression
classification
unsupervised
dimension reduction
clustering

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

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

A: A functional relationship between input & response variables.

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*:*

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?

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

Q: What do the terms in this model mean?

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

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

Q: What do the terms in this model mean?

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

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

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

Q: What do the terms in this model mean?

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

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)*

Q: What do the terms in this model mean?

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

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** *(the model "parameter")*

Q: What do the terms in this model mean?

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

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** *(the model "parameter")*

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

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

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

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

Linear regression involves several technical assumptions and is often presented with lots of mathematical formality.

The math is not very important for our purposes, but you should check it out if you get serious about solving regression problems.

Q: How do we fit a regression model to a dataset?

Q: How do we fit a regression model to a dataset?

A: In theory, minimize the sum of the squared residuals (OLS).

Q: How do we fit a regression model to a dataset?

A: In theory, minimize the sum of the squared residuals (OLS).

In practice, any respectable piece of software will do this for you.

Q: How do we fit a regression model to a dataset?
A: In theory, minimize the sum of the squared residuals (OLS).

In practice, any respectable piece of software will do this for you.

But again, if you get serious about regression, you should learn how this works!

III. 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.

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.

Q: Does anyone know what it is?

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.

Q: Does anyone know what it is?

A: This model violates one of the assumptions of linear regression!

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
```

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

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

Multicollinearity causes the linear regression model to break down, because it can't tell the predictor variables apart.

Q: What can we do about this?

Q: What can we do about this?

A: Replace the correlated predictors with uncorrelated predictors.

Q: What can we do about this?

A: Replace the correlated predictors with uncorrelated predictors.

$$y = \alpha + \beta_1 f_1(x) + \beta_2 f_2(x^2) + \ldots + \beta_n f_n(x^n) + \varepsilon$$

So far, we've seen how polynomial regression allows us to fit complex nonlinear relationships, and even to avoid multicollinearity (by using basis functions).

So far, we've seen how polynomial regression allows us to fit complex nonlinear relationships, and even to avoid multicollinearity (by using basis functions).

Q: Can a regression model be too complex?

IV. REGULARIZATION

Recall our earlier discussion of **overfitting**.

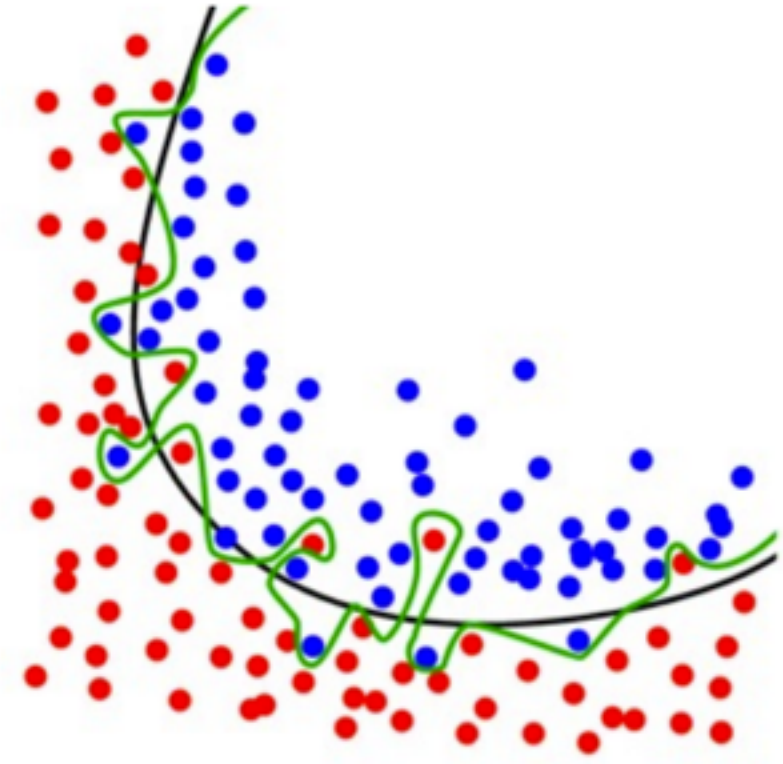
Recall our earlier discussion of **overfitting**.

When we talked about this in the context of classification, we said that it was a result of matching the training set too closely.

Recall our earlier discussion of **overfitting**.

When we talked about this in the context of classification, we said that it was a result of matching the training set too closely.

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

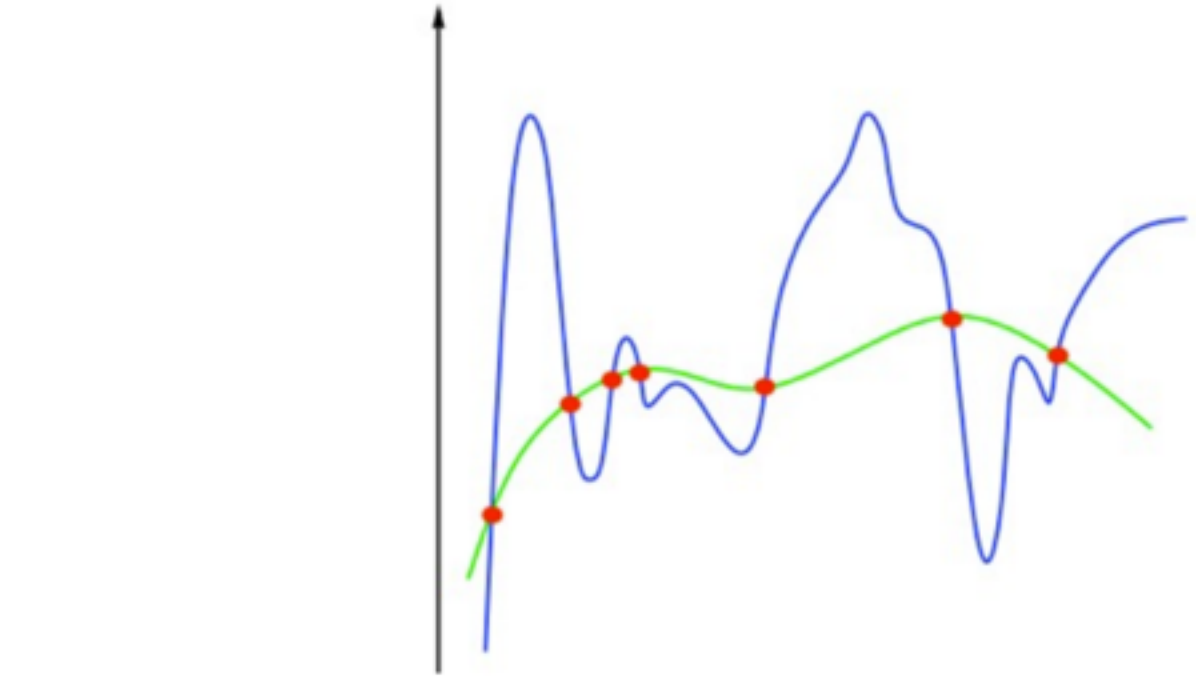


source: http://upload.wikimedia.org/wikipedia/commons/1/19/Overfitting.svg

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.



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**.

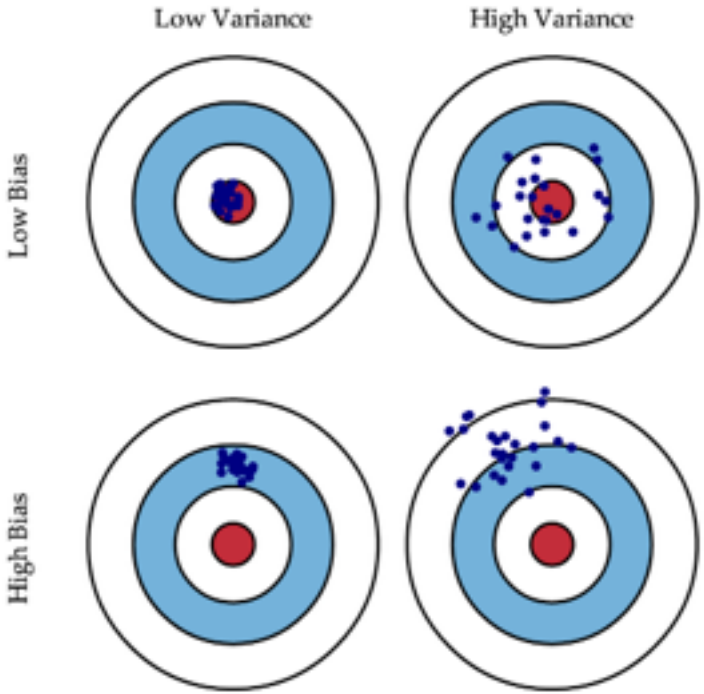
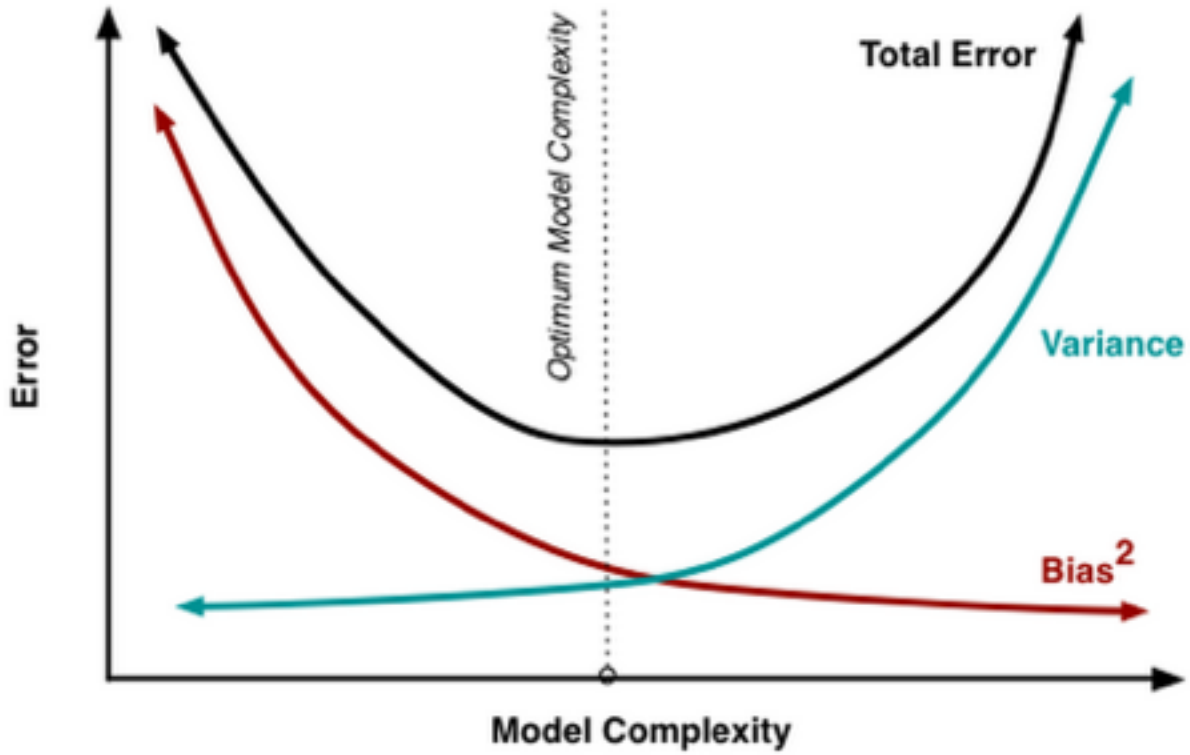


Fig. 1 Graphical illustration of bias and variance.


Error
Optimum Model Complexity
Total Error
Variance
Bias^2
Model Complexity

These regularization problems can also be expressed as:

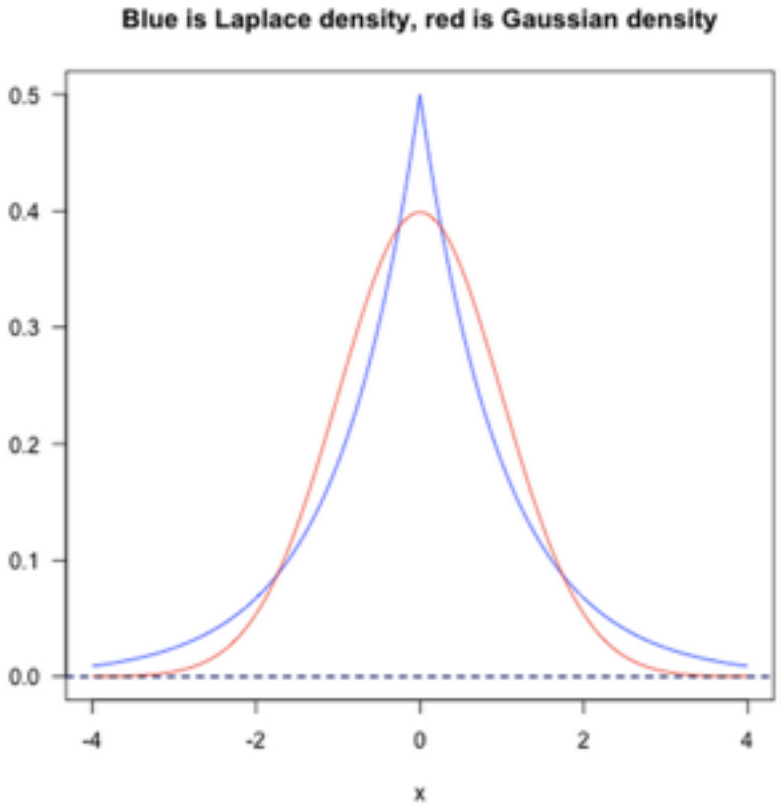
OLS: $min(\|y - x\beta\|^2)$

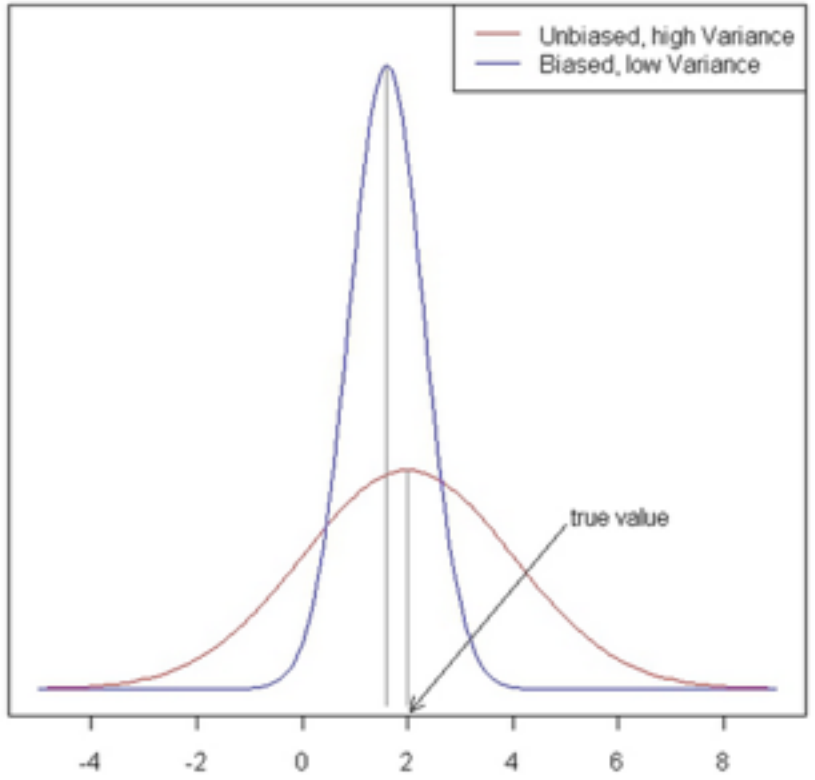
L1 regularization: $min(\|y - x\beta\|^2 + \lambda\|x\|)$

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

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

L1 = LAPLACE PRIOR
L2 = GAUSSIAN PRIOR




Unbiased, high Variance
Biased, low Variance
true value
-4
-2
0
2
4
6
8

Q: What problems have we seen?

Q: What problems have we seen?

A:

1) Correlated predictor variables

2) Large number of parameters allow us to overfit

V. EVALUATION METRICS

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i| \,.$$

$$R^2 \equiv 1 - \frac{SS_{\mathrm{res}}}{SS_{\mathrm{tot}}}.$$

$$SS_{\mathrm{tot}} = \sum (y_i - \bar{y})^2$$

$$SS_{\mathrm{res}} = \sum_i (y_i - f_i)^2$$

LAB