Introduction to Machine Learning Algorithms

Lecture 1

Syllabus

1. Introduction;
   Linear Regression;
   Logistic Regression.

2. SVM classifiers : Linear separable case;
   Lagrangian Multiplier;
   Primal-Dual Problem;

3. SVM classifiers (cont) : Non-separable case;
   Perceptron Classifier;
   Artificial Neural Networks.

4. Finishing up ANN;
   Tree Methods;

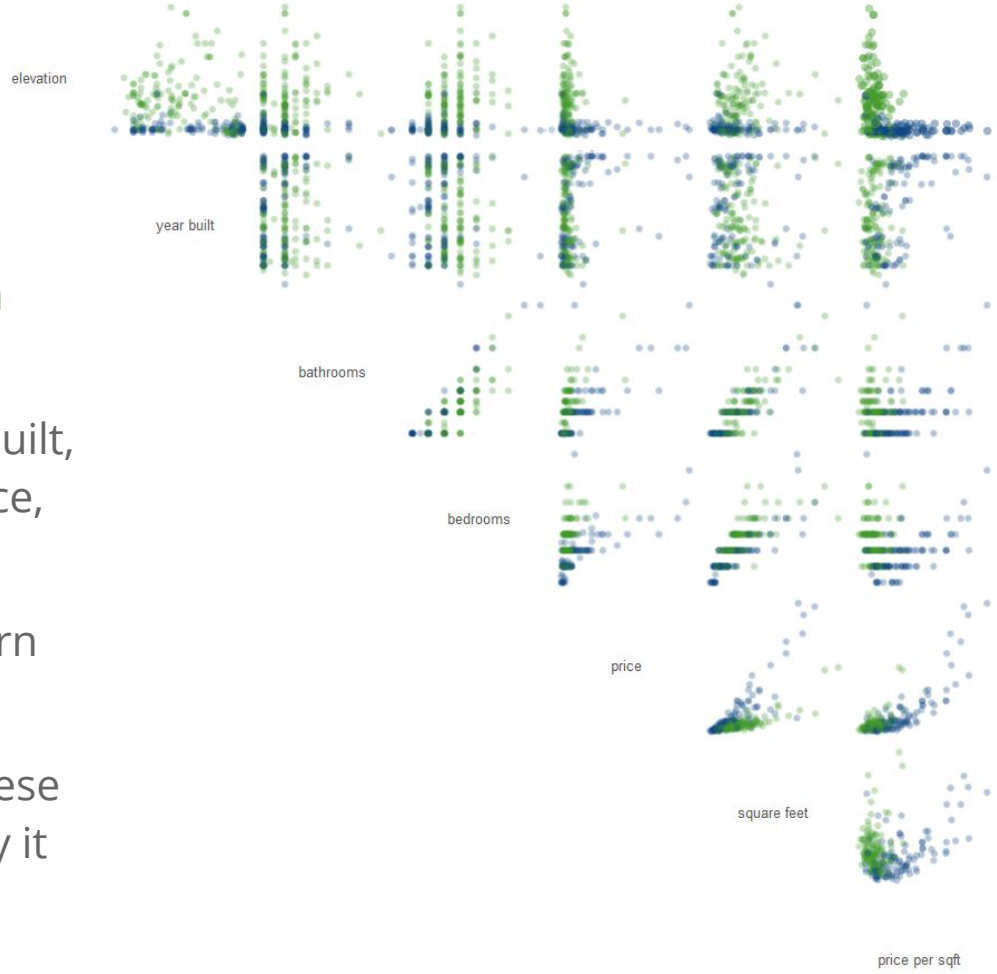
Outline

- Introduction
- Machine Learning Problem Types
  - Supervised vs. Unsupervised
  - Regression vs. Classification
  - Parametric Model vs. Non Parametric Model
- Common Evaluation Methods
- Linear Regression
  - Gradient Descent Methods
  - Underfitting vs. Overfitting: First Look
- Logistic Regression

Introduction

Example:

- Data on attributes of houses in **San Francisco** and **New York**;
- Attributes include: Elevation, year built, # of bathrooms and bedrooms, price, square feet, price per sqft.
- Given this dataset, what can we learn from it?
- If a new data case comes in with these attributes, can we predict which city it is in?



Source: R2D3, A Visual Introduction to Machine Learning http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Introduction

Machine Learning - use the help with computers to perform tasks such as prediction recognition, diagnosis, planning, robot control, etc.

Machine Learning Essentials:

- Input Vectors (feature vectors, sample, example, instance, datacase etc.)
- Outputs
- Training regime
    - Batch Methods vs Online Methods
- Performance Evaluation

Source: R2D3, A Visual Introduction to Machine Learning http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Introduction

Example machine learning problem: Decide whether to play tennis at a given day.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Introduction

Example machine learning problem: Decide whether to play tennis at a given day.

Input Attributes
- or -
Input Variables
- or -
Features
- or -
Attributes

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Introduction

Example machine learning problem: Decide whether to play tennis at a given day.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Target Variable
- or -
Class Label
- or -
Goal
- or -
Output Variable

Introduction

Training Data

Features

Class labels

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6.3	2.7	4.9	1.8	virginica
5.4	3.9	1.3	0.4	setosa
5.9	3.0	5.1	1.8	virginica
5.1	3.3	1.7	0.5	setosa
5.0	2.0	3.5	1.0	versicolor
5.0	3.3	1.4	0.2	setosa
6.4	3.1	5.5	1.8	virginica
6.6	3.0	4.4	1.4	versicolor
4.8	3.0	1.4	0.1	setosa
5.1	2.5	3.0	1.1	versicolor
5.3	3.7	1.5	0.2	setosa
6.0	2.9	4.5	1.5	versicolor

Test Data

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
4.6	3.4	1.4	0.3
6.6	3.0	4.4	1.4
6.5	3.0	5.5	1.8
6.4	2.9	4.3	1.3
5.0	2.3	3.3	1.0
5.1	3.8	1.6	0.2

Evaluation Methods

Model

Prediction Results

"virginica"
"Versicolor"
"Setosa"
"Virginica"
"setosa"
"virginica"

Source: Iris dataset, UCI Machine Learning Repository

Supervised Learning vs. Unsupervised learning

Supervised Learning:

- Output variables (class labels) are given.
- The relationship between input and output is known.

Reinforced Learning:

- Output variables are not known, but actions are rewarded or punished.

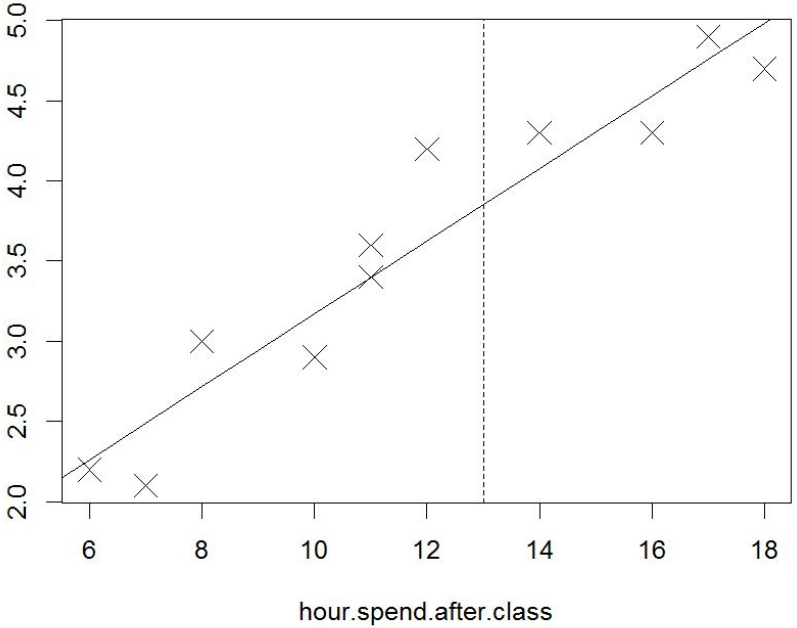
Unsupervised Learning:

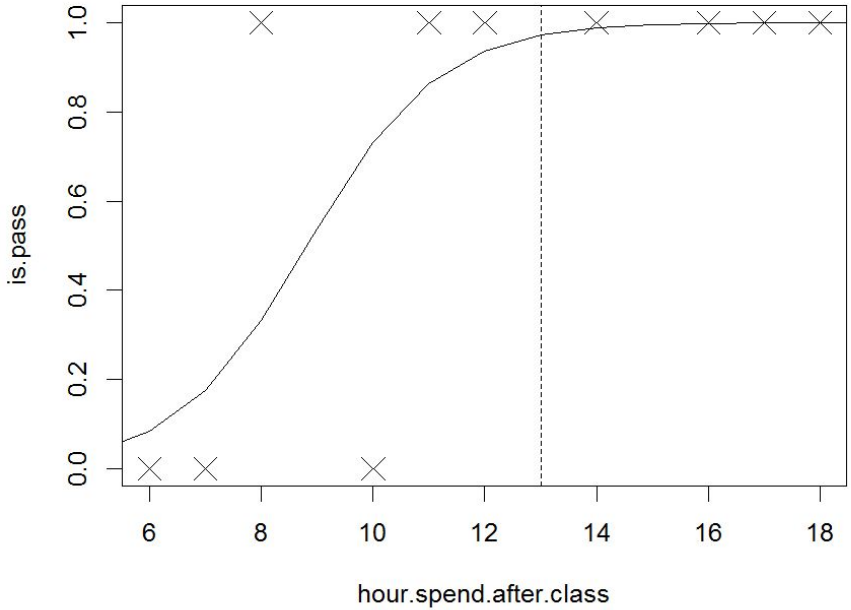
- Learn patterns from data without output variable or feedback.

(Semi-supervised Learning:)

- Only a small amount of data is labeled.

Regression vs Classification





Parametric Model vs. Non Parametric Model

Parametric Model

- Models can be represented solely by parameters.
- Fixed & finite number of parameters that doesn't depend on the data size
- Meaningful & interpretable parameters
- Ex: Linear Regression, SVM (primal problem)

Non Parametric Model

- Models are represented parameters and data
- Number of the parameters depends on the data size
- Infinite amount of parameters
- Parameters are not meaningful
- Ex: KNN, SVM (dual problem)

Evaluation Methods

- Regression: Mean squared error (MSE), sum squared error (SSE), etc.
- Classification: Accuracy, False Positive Rate, etc.
- Variations:
  - With cost matrix
  - With penalty terms to restrict model complexity
    - AIC/BIC
    - L1 regularization (LASSO)
    - L2 regularization (Ridge)

Evaluation Methods - Regression

- Mean squared error (MSE)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

- Sum squared error (SSE)
- Root Mean Squared Error
- Mean Absolute Error
- Coefficient of Determination

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

Evaluation Methods - Classification

- Confusion Matrix

		Predicted condition			
	Total population	Predicted Condition positive	Predicted Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
True condition	condition positive	**True positive**	**False Negative** (Type II error)	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$
	condition negative	**False Positive** (Type I error)	**True negative**	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$
	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	

Source: Wikipedia

Evaluation Methods - Classification

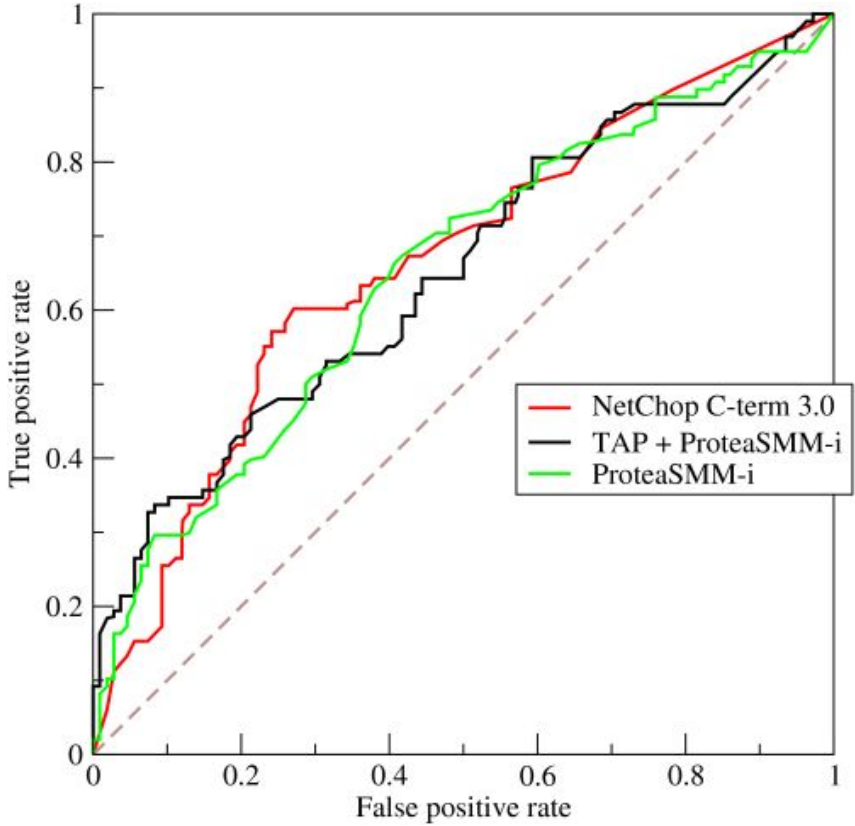
AUC: Area Under Curve

Curve: Receiver operating characteristic (ROC)

- Illustrates the performance of a binary classifier as its discrimination threshold is varied.
- Plot the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

Ex:

	prob	label
1	0.29	FALSE
2	0.45	TRUE
3	0.63	FALSE
4	0.93	TRUE



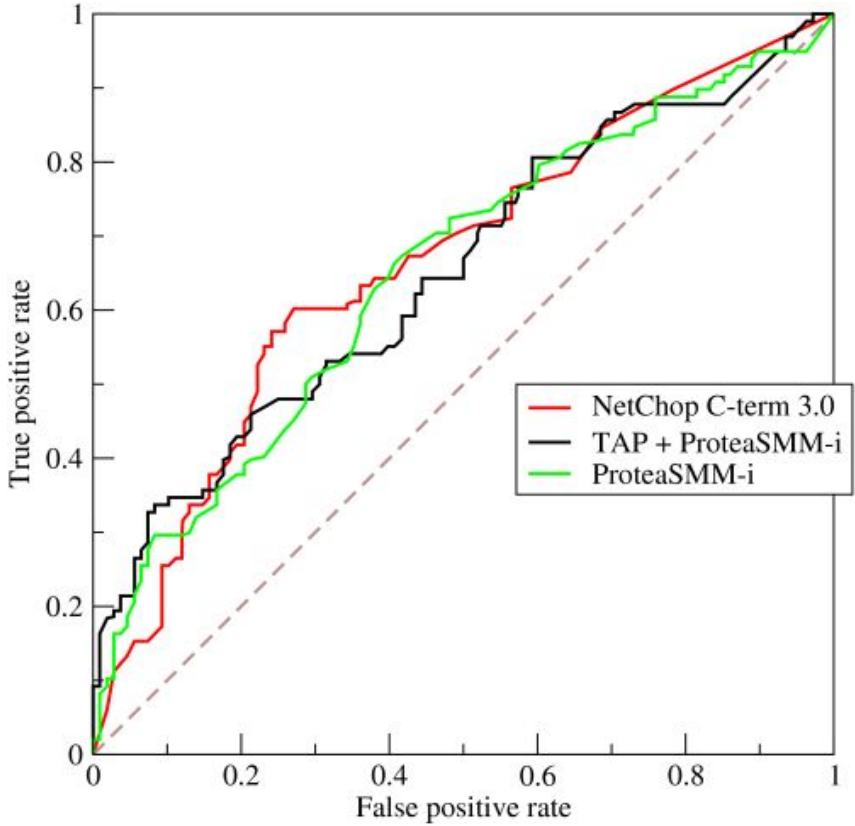
Evaluation Methods - Classification

Properties:

- Examine how our prediction rank against the true class label.
- Freedom to choose or change thresholds for different applications.
- Independent of the fraction of the test population which is class 0 or class 1.

Example Applications:

- Credit Evaluation
- Advertising Strategies



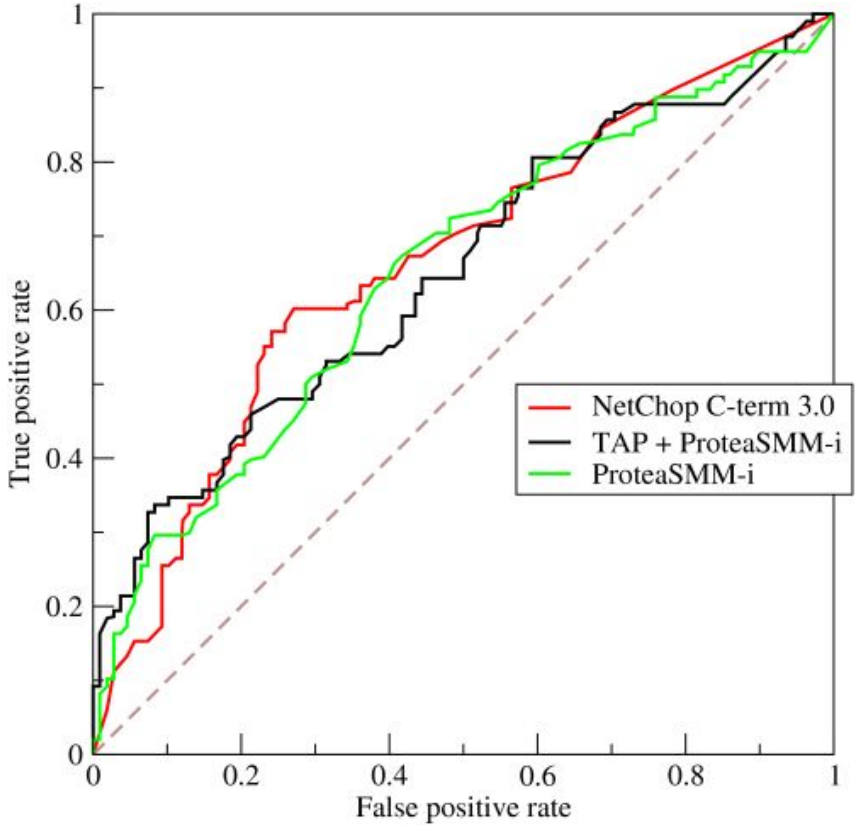
Evaluation Methods - Classification

Properties:

- Examine how our prediction rank against the true class label.
- Freedom to choose or change thresholds for different applications.
- Independent of the fraction of the test population which is class 0 or class 1.

Example Applications:

- Credit Evaluation
- Advertising Strategies



Evaluation Methods - Variation

Cost Matrix

	actual negative	actual positive
predict negative	$C(0,0) = c_{00}$	$C(0,1) = c_{01}$
predict positive	$C(1,0) = c_{10}$	$C(1,1) = c_{11}$

Ex: Evaluate the performance of a product sale prediction model.

id	sale.price	buy.price
834275002656	29.9	19
853277004048	34.9	28
630996194515	39.8	21
658183225057	39.8	29
845218010867	39.9	28
797776047024	39.9	31
681326100690	31.9	24
803516676949	39.9	31

Evaluation Methods - Variation

Regularization:

- Regularization terms can be added to (or subtracted from) the model's loss function (or objective function).
- Restrict the parameter space in order to prevent a model gets too complicated.
- Prevent overfitting

Evaluation Methods - Variation

Regularization:

- L0 regularization
  - sum($|w|^0$)
  - Controls the number of parameters

- L1 regularization: LASSO (least absolute shrinkage and selection operator)
  - sum($|w|^1$)
  - Ability to restrain parameters to be 0.

- L2 regularization: Ridge
  - sum($|w|^2$)

Linear Regression

Notations

x : training data
y: target variable *or* class label
$x^{(i)}$: i-th data case
x_j : j-th feature of x
m: Total number of data cases
n: Total number of features

Linear Regression

Model Function $h_\theta(x) = \theta_0 + \theta_1 x_1$

Let x_0 be a vector of 1s, $x_0 = [1, 1, 1, \ldots, 1]^T$ $h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$

The difference between target and predictions $\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$

Define the cost function $J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Our goal is to find the parameter that minimizing the cost func $J(\theta)$

- $\hat{\theta} = \arg\min_\theta J(\theta)$
- One method: Gradient descent

Gradient Descent

- An iterative method to find the max/min* values.
- *: Local max/min; sensitive to starting point
- Intuition: At each step, walk towards the steepest direction.
- Used widely in practice

Steps:

1. Pick a starting point
2. Repeat
   a. Calculate the gradient of the function at current point
   b. Move a step towards the direction of the gradient to redu $J(\theta)$
3. Stop when $J(\theta)$ is small enough

Example: Gradient Descent in Linear Regression

At each iteration, update the parameter $\theta_i \leftarrow \theta_i - \alpha \frac{\partial J(\theta)}{\partial \theta_i}$, where α is the step size.

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 \\
&= \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \frac{\partial}{\partial \theta_j} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\
&= \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \frac{\partial}{\partial \theta_j} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)} \right)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial J(\theta)}{\partial \theta_0} &= \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\
\frac{\partial J(\theta)}{\partial \theta_1} &= \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}
\end{aligned}
$$

Stochastic Gradient Descent

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

Problem: every update, we have to sum over all data points. For large dataset, it is very slow.

Stochastic Gradient Descent can be used:

Repeat:

For i in 1 to m :

Perform gradient calculation using only data point i

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

Linear Regression

So far...

- We mentioned a iterative methods to calculate $\hat{\theta} = \arg\min_{\theta} J(\theta)$

Further reading:

- Use matrix representation to calculate $\hat{\theta}$.
- The normal equations $\theta = (X^T X)^{-1} X^T \vec{y}.$
- http://cs229.stanford.edu/notes/cs229-notes1.pdf
  Part I.2, Page 7 - 11.

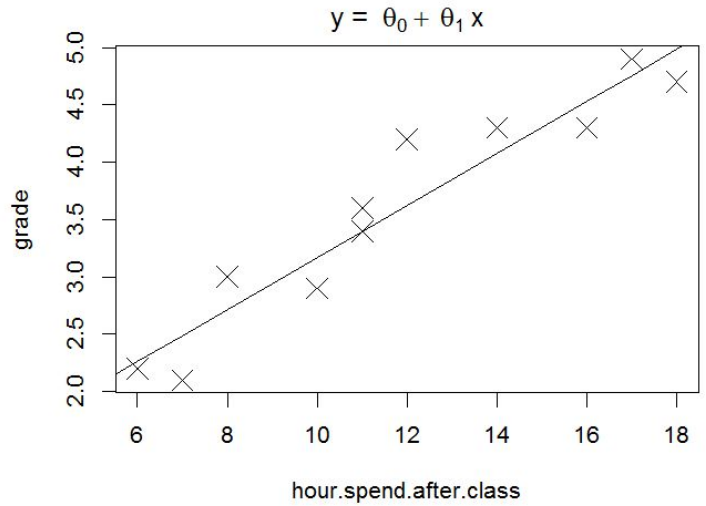
Linear Regression - Variation

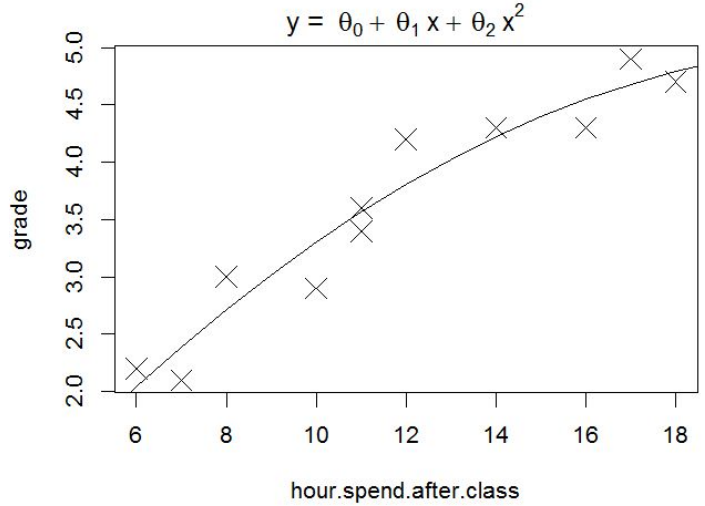
We can apply feature transformation to make the model appear “non-linear”.

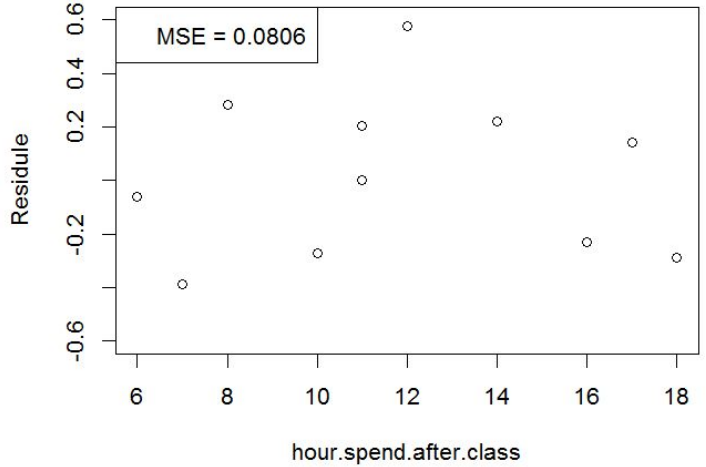
EX: Polynomial Regression

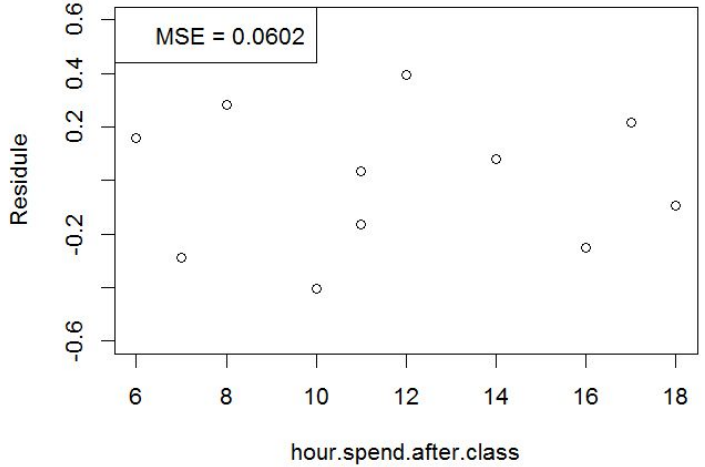
- Useful when there is reason to believe the relationship between two variables is curvilinear.
- $y = \theta_0 + \theta_1 x + \theta_2 x^2$

Might lead to overfitting!

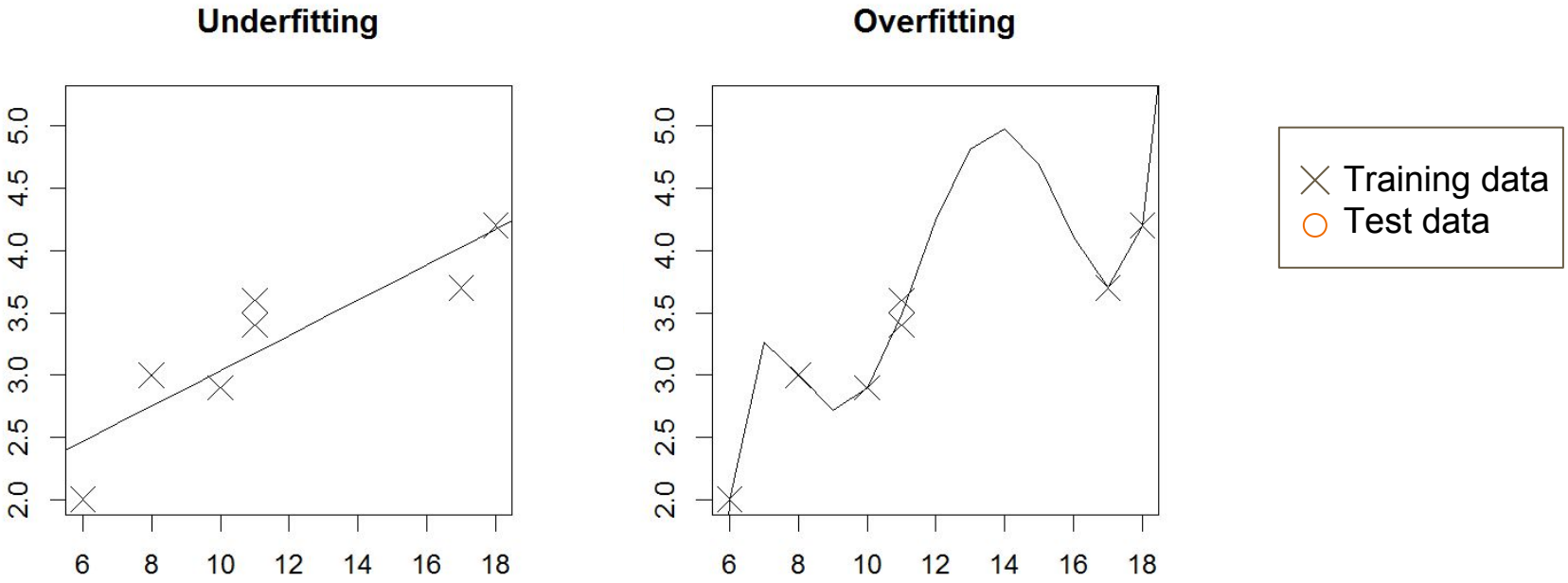

y = θ0 + θ1 x
grade
hour.spend.after.class


y = θ0 + θ1 x + θ2 x2
grade
hour.spend.after.class

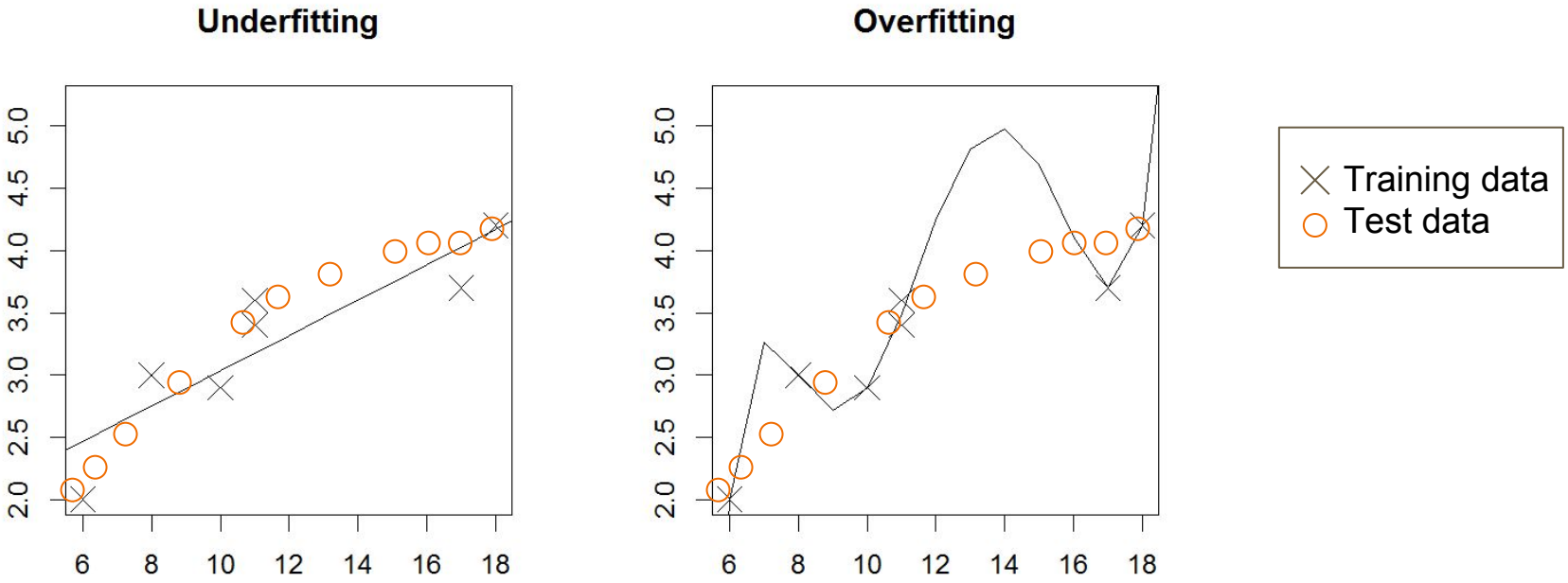

MSE = 0.0806
Residule
hour.spend.after.class


MSE = 0.0602
Residule
hour.spend.after.class

Bias vs. Variance (Underfitting vs. Overfitting)



Bias vs. Variance (Underfitting vs. Overfitting)



Logistic Regression

- Logistic function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$
- Logistic regression $F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$,
  where F(x) is the probability of the target variable if X is “positive”: $F(X) = P(Y = 1 \mid X)$
- It assumes the probability of the target variable being “positive” can be explained by a combination of the explanatory variable after logistic-transformation.

Logistic Regression

Why is logistic regression related to linear regression?

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \quad \Rightarrow \quad \frac{F(x)}{1 - F(x)} = e^{\beta_0 + \beta_1 x} \quad \Rightarrow \quad \ln\left(\frac{F(x)}{1 - F(x)}\right) = \beta_0 + \beta_1 x$$

$\ln\left(\frac{F(x)}{1 - F(x)}\right)$ is called “odds”. $\frac{F(x)}{1 - F(x)}$ is called “logit”.

In logistic regression, we assume a linear relationship between explanatory variable and the log odds of the target variable.

Other basic models

- We we not be talking about these models but they are worth looking at
  - Naive Bayes
  - K Nearest Neighbor
  - K-Means for clustering
  - Hierarchical clustering