Regression

Albert Bifet

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

May 2012

COMP423A/COMP523A Data Stream Mining

Outline

1. Introduction
2. Stream Algorithmics
3. Concept drift
4. Evaluation
5. Classification
6. Ensemble Methods
7. **Regression**
8. Clustering
9. Frequent Pattern Mining
10. Distributed Streaming

Data Streams

Big Data & Real Time

Regression

Definition

Given a **numeric** class attribute, a regression algorithm builds a model that predicts for every unlabelled instance *I* a **numeric** value with accuracy.

$$y = f(x)$$

Example

Stock-Market price prediction

Example

Airplane delays

Evaluation

1. Error estimation: *Hold-out or Prequential*
2. Evaluation performance measures: *MSE or MAE*
3. Statistical significance validation: *Nemenyi test*

Evaluation Framework

2. Performance Measures

Regression **mean** measures

- Mean square error:

$$MSE = \sum (f(x_i) - y_i)^2 / N$$

- Root mean square error:

$$RMSE = \sqrt{MSE} = \sqrt{\sum (f(x_i) - y_i)^2 / N}$$

Forgetting mechanism for estimating measures

Sliding window of size w with the most recent observations

2. Performance Measures

Regression **relative** measures

- Relative Square error:

$$RSE = \sum (f(x_i) - y_i)^2 / \sum (\bar{y}_i - y_i)^2$$

- Root relative square error:

$$RRSE = \sqrt{RSE} = \sqrt{\sum (f(x_i) - y_i)^2 / \sum (\bar{y}_i) - y_i)^2}$$

Forgetting mechanism for estimating measures

Sliding window of size w with the most recent observations

2. Performance Measures

Regression **absolute** measures

- Mean absolute error:

$$MAE = \sum(|f(x_i) - y_i|)/N$$

- Relative absolute error:

$$RAE = \sum(|f(x_i) - y_i|)/\sum(|\hat{y}_i - y_i|)$$

Forgetting mechanism for estimating measures

Sliding window of size w with the most recent observations

Linear Methods for Regression

Linear Least Squares fitting

- Linear Regression Model

$$f(x) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j = \mathbf{X}\beta$$

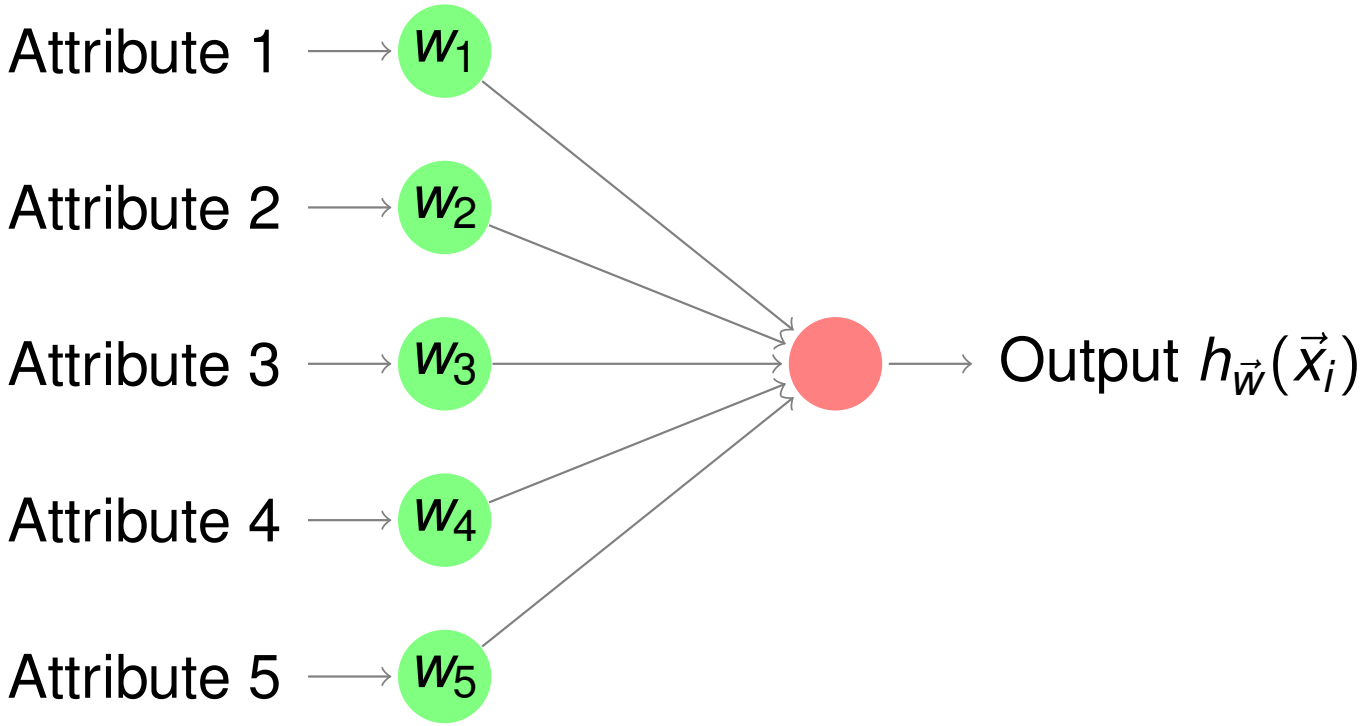
- Minimize residual sum of squares

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2 / N = (\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$$

- Solution:

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

Perceptron



- Data stream: $\langle \vec{x}_i, y_i \rangle$
- Classical perceptron: $h_{\vec{w}}(\vec{x}_i) = \vec{w}^T \vec{x}_i$,
- Minimize Mean-square error: $J(\vec{w}) = \frac{1}{2} \sum (y_i - h_{\vec{w}}(\vec{x}_i))^2$

Perceptron

- Minimize Mean-square error: $J(\vec{w}) = \frac{1}{2}\sum(y_i - h_{\vec{w}}(\vec{x}_i))^2$
- Stochastic Gradient Descent: $\vec{w} = \vec{w} - \eta\nabla J\vec{x}_i$
- Gradient of the error function:

$$\nabla J = -\sum_i (y_i - h_{\vec{w}}(\vec{x}_i))$$

- Weight update rule

$$\vec{w} = \vec{w} + \eta\sum_i (y_i - h_{\vec{w}}(\vec{x}_i))\vec{x}_i$$

Fast Incremental Model Tree with Drift Detection FIMT-DD

FIMT-DD differences with HT:

1. Splitting Criterion
2. Numeric attribute handling using BINTREE
3. Linear model at the leaves
4. Concept Drift Handling: Page-Hinckley
5. Alternate Tree adaption strategy

Splitting Criterion

Standard Deviation Reduction Measure

- Classification

$$\text{Information Gain} = \text{Entropy(before Split)} - \text{Entropy(after split)}$$

$$\text{Entropy} = -\sum^{c} p_i \cdot \log p_i$$

$$\text{Gini Index} = \sum^{c} p_i(1 - p_i) = 1 - \sum^{c} p_i^2$$

- Regression

$$\text{Gain} = \text{SD(before Split)} - \text{SD(after split)}$$

$$\text{StandardDeviation (SD)} = \sqrt{\sum(\bar{y} - y_i)^2/N}$$

Numeric Handling Methods

Exhaustive Binary Tree (BINTREE – Gama et al, 2003)

- Closest implementation of a batch method
- Incrementally update a binary tree as data is observed
- Issues: high memory cost, high cost of split search, data order

Page Hinckley Test

- The CUSUM test

$$g_0 = 0, \qquad g_t = \max\left(0, g_{t-1} + \epsilon_t - \upsilon\right)$$

$$\text{if } g_t > h \text{ then alarm and } g_t = 0$$

- The Page Hinckley Test

$$g_0 = 0, \qquad g_t = g_{t-1} + (\epsilon_t - \upsilon)$$

$$G_t = \min(g_t)$$

$$\text{if } g_t - G_t > h \text{ then alarm and } g_t = 0$$

Lazy Methods

kNN Nearest Neighbours:

1. Mean value of the k nearest neighbours

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

2. Depends on distance function