goo.gl/eKxKZH

Linear Models

Machine Learning Workshop Series

Angelos Filos

October 19, 2017

Imperial College Data Science Society

Table of Contents

Theory

Problem Definition

Provided a set $\mathcal{S}$ of (x_i, y_i) pairs:

$$\mathcal{S} = \{(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)\}$$

Find a function f that maps $x_i \to y_i$:

$$f : \mathcal{X} \to \mathcal{Y}$$

f is a:

- **(Discriminate) Model**
- **Map Function**
- **Function Approximation**

for $\mathcal{S}$, such that:

$$f(x_i) = \hat{y}_i \approx y_i, \quad \forall i$$

Linear Model

f is a **Linear Model**, iff:

$$f(x_i) = w * x_i, \quad x_i \in \mathbb{R}^n; y_i \in \mathbb{R}^m; w \in \mathbb{R}^{m \times n} \tag{1}$$

For a finite $\mathcal{S}$ where $|\mathcal{S}| = k$, we express (1) in matrix format as:

$$f(\mathbf{X}) = \hat{\mathbf{y}} = \mathbf{X} * \mathbf{w}, \quad \mathbf{X} \in \mathbb{R}^{k \times n}; \mathbf{y} \in \mathbb{R}^{k \times m}; \mathbf{w} \in \mathbb{R}^{n \times m}$$

Matrices **X** and **y** are constructed by stacking the k pairs as row vectors.

Mini Demo

Metric

Use **Mean Squared Error (MSE)** as the metric function in order to evaluate how good/bad the model f is, such that:

$$MSE_f = \frac{1}{2k}\sum_{i=1}^{k}(y_i - \hat{y}_i)^2 = \frac{1}{2k}\sum_{i=1}^{k}(y_i - w_i * x_i)^2$$

In matrix form:

$$MSE_f = \frac{1}{2k}(\mathbf{y} - \mathbf{X} * \mathbf{w})^T(\mathbf{y} - \mathbf{X} * \mathbf{w}) \tag{2}$$

Target: minimize MSE_f in order to increase accuracy! Apply optimization algorithms to achieve it.

Solutions

Minimize the metric (a.k.a error) function.

Analytic Solution

Maximum Likelihood Estimator: differentiate 2 and find its minimum. There is a global minimiser, since MSE_f is convex (quadratic).

$$\mathbf{w_{MLE}} = (\mathbf{X}^T\mathbf{X})^{-1} * \mathbf{X}^T * \mathbf{y}$$

Computational Solution

Gradient Descent: make incremental updates, towards the correct solution, using the derivative of the error at each step.

$$w_{t+1} = w_t - \eta(y_i - \hat{y})x_i$$

Application

Context

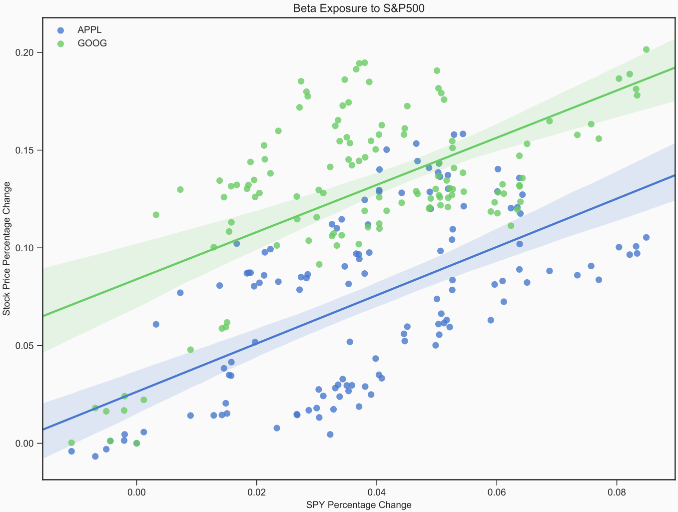
Factor models are a way of explaining the returns of one asset via a linear combination of the returns of other assets. The general form of a factor model is:

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n \tag{3}$$

An asset's β to another is just the β factor from (3).
Look for:

$$y_{stock} = \beta x_{benchmark} + \alpha = \begin{bmatrix} x_{benchmark} & 1 \end{bmatrix} \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

Results


Beta Exposure to S&P500
APPL
GOOG
Stock Price Percentage Change
SPY Percentage Change
0.20
0.15
0.10
0.05
0.00
0.00
0.02
0.04
0.06
0.08

Questions?

Codelab

Setup

1. Create a `Github` account.
2. Sign-in `cocalc` using your `Github` credentials.
3. Create a new project in `cocalc`.
4. Clone (green button at top RHS) in `zip` format the **Linear Models** repository.
5. `Upload` the `zip` file to newly created `cocalc` project.
6. Click on the `zip` file and extract the compressed files.
7. Navigate to the extracted folder `~/Linear-Models-master/notebooks/Demo.ipynb`.
8. Select from the menu bar: `Kernel` → `Change Kernel` → `Python 3 (Anaconda)`.

Challenges

Beta Hedging

Come up with a Beta Hedging strategy using Linear Models

Generic Machine Learning API for Linear Models

Create a generic API, encapsulating all the different implementations and exposing simple methods:

- `fit`
- `predict`

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the participants individually and, unless expressly stated to the contrary, are not the opinion or position of the ICDSS, its cosponsors, or its committees. The ICDSS does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented. Attendees should note that sessions are video-recorded and may be published in various media, including print, audio and video formats without further notice.