seis-ml-api

情報科学類二年 江畑 拓哉 (201611350)

1 全体案

以降この図に記載された名称を用いて説明を行う。

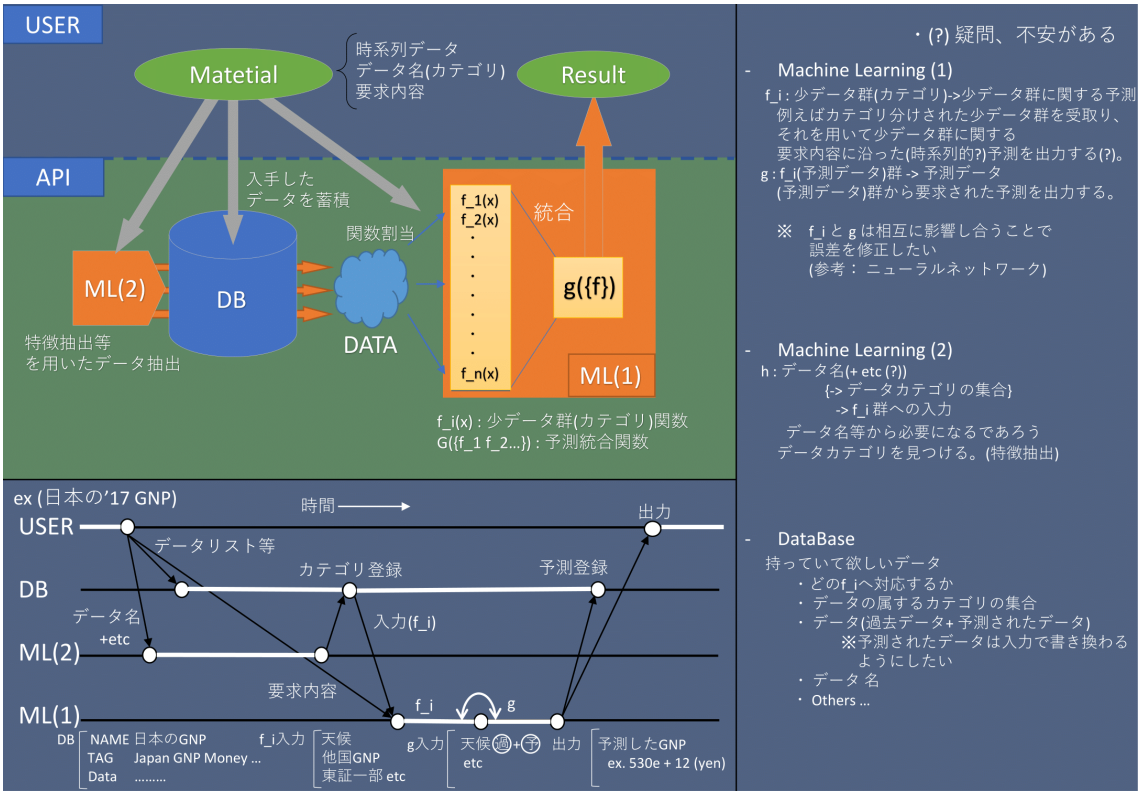

Figure 1: 全体案

2 機械学習部分（１）

メインとなる機械学習部分 Machine Learning(1) については回帰アルゴリズム又はランダムフォレストと重回帰分析を用いたいと考えている。そのうち、非線形回帰分析又はランダムフォレストに関しては、f_i 関数部分で用いたいと考えている。同様に、重回帰分析については、g 関数部分で用いたいと考えている。
¡2017-07-17 月¿ f_i 関数についてはランダムフォレスト、g 関数については重回帰分析を行いたいと考えている。

2.1 回帰分析

一般的なアルゴリズムには以下の種類がある。 [9]

2.1.1 線形回帰

- 一般化線形モデル
  分布関数に残差が従っている場合に用いることができる。よって複数の分布族でこれを試し、最適な分布を選び出していくのだろうと考えられる。これが元になって、ロジスティック回帰、ポアソン回帰などが含まれている

- ロジスティック回帰
  ベルヌーイ分布に従う変数の統計的回帰モデルである。ベルヌーイ分布とはそもそも確率 p で 1 を確率 $q = 1 - p$ で 0 を取る離散分布 $(f(k;p) = p^k * q^{1-k})$ である。これはクラスタリング（教師なし）や分類（教師あり）に用いる方が有用であるようで、そちらの例のほうが多く見受けられた。
  モデルは以下の通りである。
  $y = \alpha/(1 + \beta * exp(-x * y))$

2.1.2 非線形回帰

- 多項式回帰
  線形回帰を曲線にしたもので、ｎ次方程式のモデルを作成する。係数判定には最急降下法やその変形の確率的勾配法が用いられるようである。最急降下法とは、目的関数を偏微分することで求まる関数を用いた最小値探索法で、この変が確率的勾配降下法と呼ばれている。問題は過剰適合（オーバーフッティング）が起きてしまうことであるが、その対策については今後調べていく予定である。

モデルは以下の通りである。
$y = a + b * x + c * x^2 \ldots$

- 一般化加法モデル
  平滑化スプラインを複数生成して重回帰させていると考える。

- 平滑化回帰モデル
  複雑に変化するデータに用いる。歪み度合いを指定して曲線回帰を行う。

2.1.3 TODO それぞれの実行例を Python などで実行する

2.1.4 TODO 多項式回帰の過剰適合に対する対策

2.1.5 TODO 確率的勾配法についての説明

2.2 ランダムフォレスト [2] [1]

理解に困難があったため、ランダムフォレストの回帰部分について自己解釈をここに示す。

まず、ランダムフォレストの概要についてである。ランダムフォレストとは、複数の決定木を用いていわゆる学習モデルの森を作り、それぞれについて回帰予測の結果を算出し、すべての予測位結果を統合する（回帰予測なので、つまりは平均を取る)。

ここで回帰の条件木である回帰木の分岐基準について説明を加えると、以下の式の ΔI (二乗平均誤差？）を最大化することである。

$\Delta I = (S(A) - S(A_L) - S(A_R))/N$

（すなわち、(全体の分散）-（(右の木の分散) + (左の木の分散)）が最大となるということである。）

また、この式の S(B) は尤離度と呼ばれ、 $\overline{t_B}$ を条件ノード B での目標変数 t の平均値であるとすれば、

$S(B) = \sum_{k \in A}(t_k - \overline{t_B})^2$

2.2.1 決定木学習 [7] [8]

ランダムフォレスト内で用いた決定木についての理解を深めたい。

決定機学習とは、目的変数と説明変数のデータから木構造の分類器を生成を作成し、再帰的にデータを分割していく。分割基準は先程示した式の通りである。そして停止条件（例えば、深さなど）を満たしたところで分割を停止する。

2.2.2 TODO 上の式 ΔI の理解

2.2.3 TODO ランダムフォレストと回帰を用いた例の実行

2.3 重回帰分析

多重回帰分析は、複数の条件データのある際に用いる回帰分析の手法である。

f_i 関数によってある意味無限個のデータを作成し（予測した曲線から抜き取れば良い)、それぞれのデータ群の値を用いて重回帰分析を行う。

モデルは以下のとおりである。

$y = a_0 + a_1 * x_1 + a_2 * x_2 + a_3 + x_3 \ldots$

2.3.1 TODO Python で scikit-learn を用いて具体的な動作確認 [5]

2.3.2 TODO 多重境界性の対策

2.4 TODO ガウス過程についての理解

3 機械学習部分（２）

副目標である機械学習部分 Machine Learning(2) についてはクラスタリングを用いたいくつかのカテゴリの分類や、深層学習のアルゴリズムで（事前に用意された）関連タグを当てはめる手法を考えている。

3.1 TODO クラスタリング

3.2 カテゴリ分類

ある程度カテゴリを作成して、タイトル（と説明文）からテキスト分類を行う。(CNN/RNN)

3.2.1 TODO CNN

Convolution Neural Network (CNN) は、主に画像認識で人気のある分類方法である。

3.2.2 TODO RNN

3.2.3 SDA [3] [4] [6]

Stacked Denoising Autoencoder の略、Autoencoder とは自己符号器の意味を示しており、stack とはそれを積み上げるということで、Denoising とはノイズ除去の意味であるから、和訳するならば、”たくさん積み上げたノイズ除去を行う自己符号器 ”といったところである。

3.2.4 DAE

Denoising Autoencoder のことである。AE の入力ベクトルの一部にノイズを加える破壊分布 $C(\overline{X}|X)$ を考え（例えばガウスノイズや塩胡椒ノイズ）、入力 X ＝ x に対してこの破壊分布からデータ ($\overline{X}$) をサンプリングして、入力データを復元する。つまり、ノイズを蹴り飛ばして必要な要素を抽出して復元していることになる。

4 データベース部分

References

[1] deaikei. *Python* **でランダムフォレストを実装する男**. URL: http://qiita.com/deaikei/items/52d84ccfedbfc3b222cb.

[2] Koichi Hamada. **「はじめてでもわかる** *RandomForest* **入門－集団学習による分類・予測 －」 －第７回データマイニング＋***WEB* **勉強会@東京**. URL: https://www.slideshare.net/hamadakoichi/randomforest-web.

[3] hogefugabar. **深層学習でニュース記事を分類する**. URL: http://qiita.com/hogefugabar/items/c27ed578717c5e7288c0.

[4] Kai Sasaki. *AutoEncoder* **で特徴抽出**. URL: https://www.slideshare.net/lewuathe/auto-encoder-v2.

[5] Python Data Science. *scikit-learn* **で線形回帰**. URL: http://pythondatascience.plavox.info/scikit-learn/%E7%B7%9A%E5%BD%A2%E5%9B%9E%E5%B8%B0.

[6] :tochikuji. *chainer* **で** *Stacked denoising Autoencoder*. URL: http://tochikuji.hatenablog.jp/entry/20150916/1442406243.

[7] 下畑光夫. **決定機学習**. https://www.slideshare.net/mitsuoshimohata/ss-35949886.

[8] **決定木** *Matlab*. URL: https://jp.mathworks.com/help/stats/classification-trees-and-regression-trees.html#bsw6a9f.

[9] **非線形回帰分析**. URL: http://i.cla.kobe-u.ac.jp/murao/class/2014-SeminarB2/9_NonlinearRegression.pdf.