第一回レポート課題（回帰）

- Online News Popularity Prediction

  K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.

- 58種類の素性（特徴）からWebニュースの人気（シェア数）を予測
- 説明変数

  0. url: URL of the article (non-predictive)
  1. timedelta: Days between the article publication and the dataset acquisition (non-predictive)
  2. n_tokens_title: Number of words in the title
  3. n_tokens_content: Number of words in the content
  4. n_unique_tokens: Rate of unique words in the content
  5. n_non_stop_words: Rate of non-stop words in the content
  6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content
  7. num_hrefs: Number of links
  8. num_self_hrefs: Number of links to other articles published by Mashable
  9. num_imgs: Number of images
  10. num_videos: Number of videos
  ……

- 目的変数

  60. shares

コンペティション形式

- 学習データ：30000
- テストデータ：残り
  - 講義ページからダウンロード（元データを探すことはしないで。。。）

- 評価指標
  - Mean absolute error

$$\mathrm{MAE} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left| y_i - \hat{y}_i \right|$$

- ランキング
  - スコアリングサーバ: http://www.nlab.ci.i.u-tokyo.ac.jp/~nakayama/ds15/report1/index.php
    - テストサンプルの推定結果を一行ずつ記載したテキストファイルを提出
    - ユーザ名を忘れずに（区別できればなんでもよい。ただし同じ名前をずっと使用すること）
    - 提出した結果は上書きされる。最後のものだけ保存されるので注意。
  - 現在は、提出されたテストデータの推定結果のうち、所定の2000サンプルでスコアリングしている
  - 最終的なスコアは、締め切り後に残りのサンプルで算出

サンプルプログラム (線形回帰、sample.py)

```
import numpy as np
import pandas as pd
import regression as reg

N = 20000

train_data = pd.read_csv(“train.csv”)  #学習データ
test_data = pd.read_csv(“test.csv”)  #テストデータ

train_data = train_data.drop(['url'], axis=1) #remove 'url' information.
train_data = train_data.drop(['timedelta'], axis=1) #remove 'url' information.
X = np.matrix(train_data.drop(['shares'], axis=1))
y = np.matrix(train_data['shares']) #This is the target

XTrain = X[:N,:] #use the first N samples for training
yTrain = y[:,:N]
XVal = X[N:,:] #use the rests for validation
yVal = y[:,N:]

w = reg.standRegres(XTrain,yTrain) #linear regression

yHatTrain = np.dot(XTrain,w)
yHatVal = np.dot(XVal,w)

print "Training error ", np.mean(np.abs(yTrain - yHatTrain.T))
print "Validation error ", np.mean(np.abs(yVal - yHatVal.T))

yHatTest = np.dot(np.matrix(test_data),w)
np.savetxt('result.txt', yHatTest)
```

```
-3.487996103076620784e+04
4.794085739100871433e+04
-1.264702646353116688e+04
1.903336971212502685e+05
6.820109483577609353e+04
-4.492000889108623960e+04
2.027670462503317103e+05
2.284462739066630820e+04
1.185053637056280713e+05
1.736125060031416797e+04
3.946789935058748961e+04
-6.082048353184589359e+03
8.163972390325141532e+03
…
```

(result.txt)

test.csv と同じ順番で、各テストサンプルの予測値が一行ずつ入っている。

課題詳細

- レポート内容
  - Online News Popularity Predictionの問題に取り組み、以下の点を中心にA4用紙2~3枚程度にまとめよ。
    講義で扱っていない技術を用いても構わない。

  1．予測性能を向上させるための自分なりの工夫点と結果、考察（必須）
  - コードを載せる必要はない（実装がポイントの場合は載せてかまわない）

  2．その他、自由にデータを分析した結果（+α）
  - 学習データ数と性能、正則化パラメータの関係
  - 各特徴の寄与の分析

  3．ここまでの講義の感想、要望など（必須）

課題詳細

- 評価の方針
  - アイデアや試行錯誤の過程を重視
  - スコアが悪くても、しっかり考察してくれればＯＫ
    （もちろん良くなればプラスに評価しますが）
  - 面白い分析を期待します

- 提出先
  - 以下のアドレスへメールで提出すること（質問もこちらへ）
  - ds2015@nlab.ci.i.u-tokyo.ac.jp
  - 件名は「データサイエンスレポート課題１」
  - 氏名、学籍番号、所属を忘れずに

- 締め切り
  - 12月11日（金）
  - スコアリングサーバも同時に締め切り後、テストサンプルでランキング。

工夫できそうなところ

- 回帰のオフセット
- カテゴリ変数の扱い
  - ダミー変数？
  - 分けてモデルを作る？
- 手法・パラメータチューニング
  - 正則化項の入れ方（リッジ回帰）
  - クロスバリデーション（URLからデータの時期が分かる）
- 誤差の評価
  - L2? L1?

- 原論文を読んでみる