

レポート提出票

科目名: 情報工学実験3

実験課題名: 課題2 パターン認識

実施日: 2021年 6月 3日

学籍番号: 4619055

氏名: 辰川力駆

共同実験者:

_____	_____
_____	_____
_____	_____
_____	_____

1 実験の要旨

データ分析にとって必要不可欠である前処理を演習や課題を通して理解し、Python を用いて実際に行うことでやり方を身に付ける。

2 実験の目的

演習や課題を通して、抽出や集約などの基本的なデータ前処理の仕方を理解する。

3 実験の原理

データ分析のためには主に、3つの前処理がある。

1. 「指標、表、グラフ作成」のための前処理

指標の計算、表やグラフに簡単に変換できるようなデータを準備することが目的である。必要な列 (Column) を全て持ち、扱いやすい単位で集約された行 (Row) が必要な範囲分存在するデータを作成する。

2. 「教師なし学習」の入力のための前処理

教師なし機械学習モデルに利用する説明変数を持つデータを準備することが目的である。1と同様なデータであるが、機械学習モデルの種類に応じて扱いやすいデータの型に変換したり、データのスケールを揃えたりする前処理が必要になる。

3. 「教師あり学習」の入力のための前処理

教師あり機械学習モデルに利用する学習データ、テストデータ、そしてモデルに適用するデータの準備をすることが目的である。1や2と同様の前処理だけでなく、例えば、複数の列を組み合わせて新たな特徴量に変換する前処理などが必要になることがある。

こういった前処理の目的や役割があるが、実際に前処理を進めていくにあたっての流れは、データ構造を対象とした前処理とデータ内容を対象とした前処理に分けている。

1. データ構造を対象とした前処理

複数の行にまたがったデータ全体に及ぶ処理であり、前処理の中でも早い段階で実施されることが多い。例えば、特定のデータを抽出したり、データを結合したり、あるルールに従って複数の行を1行にしたりする。

2. データ内容を対象とした前処理

行ごとのデータの値に応じた処理で、行ごとに独立で処理できる。比較的規模の小さい処理で、条件を変えて繰り返されることが多い。例えば、複数の列を組み合わせて新たな列を生成したり、日時のデータを月毎のデータに集約したりする。

4 実験方法

Python を Jupyter Notebook 上で実行することで、今回の実験を行う。

4.1 課題 1

reserve_tb から 50 % の確率で顧客をサンプリングする。サンプリングした各顧客ごとの予約件数を調査して平均を求める。顧客のサンプリングに関して以下の 2 つのやり方で行う。

reserve_tb の行を 50 % サンプリングした後に顧客を重複なく列にした場合と、reserve_tb から重複のない顧客の列を抽出し 50 % サンプリングをした場合で予約件数の平均がどの様になるかを確認し考察する。

4.2 課題 2

4.3 課題 3

4.4 課題 4

4.5 結果・検討・考察

実験結果の図 2,3 より、`myaverage_integral()` は、`myaverage_naive()` より処理速度が速い。これを計算量の観点から考える。

`myaverage_naive()` は、画像の高さ (px) × 幅 (px) × フィルタサイズ × フィルタサイズであるから、 $O(n^4)$ である。それに対し、`myaverage_integral()` は、画像の高さ (px) × 幅 (px) のみなので、 $O(n^2)$ である。これは挿入ソートやバブルソートなどの遅めのソートに匹敵する。

よって、積分画像の平均化フィルタの方が高速化できていることが計算量の観点からも分かる。図 3 において、`myaverage_integral()` がフィルタサイズの影響をあまり受けてない理由も同様である。

図 1: himeji_noise.png

5 まとめ

パターン認識とは、入力データを複数のクラスに対応させる処理のことである。

A 付録

ソースコード 1: kadai1

```
1  #4619055
2  import time
3  import numpy as np
4  from skimage.io import imread, imsave
5  from skimage.color import rgb2gray, gray2rgb
6  from skimage.transform import resize
7  from scipy.ndimage.filters import convolve, correlate
8  import skimage.transform
9  import matplotlib.pyplot as plt
10 %matplotlib inline
```