計算機科学実験及演習 4 エージェント 課題 4

1029-28-2473 二見 颯

2018 年 11 月 15 日

1　課題 4-1

1.1　プログラム概要

リスティングデータによって与えられる類似の宿泊施設の提示価格を予測して、それによって自己の宿泊施設の価格推薦を行うことを考える。民泊市場にはリスティングデータに従う貸し手 1 人と価格推薦エージェント 1 体という条件で、価格推薦エージェントはリスティングデータの価格を予測して、それより少し価格を提示するとする。

1.2　外部仕様

リスティングデータとして、San Francisco-listings.csv を用いることとする。
課題 3 と同様に、96 の特徴量のうち'latitude', 'longitude', 'accommodates', 'property_type', 'room_type', 'number_of_reviews', 'review_scores_rating' を説明変数として、'price' を目的変数とする。ただし、'price' が 500 以下のデータのみを利用する。
bayes_opt.py
pip install scikit-optimize が必要。
Grid Search の代わりにベイズ最適化の手法を用いて、SVR の最適なハイパーパラメータを探索する。
agent.py
予測価格の何 % とするかをコマンドライン引数として取り、SVR による価格予測を用いて価格推薦を行う 1 体のエージェントをシミュレーションする。プログラムの実行は
./agent.py [**percentage**]
とする。以下は agent.py の実行例である。

```
$ ./agent.py 90
load sanfrancisco: read 1000 / 6196 data
start SVR fit...
cvxopt.solvers.qp: optimization succeeded
start SVR predict...
revenue by SVR agent: 24257.254376011988
the agent wins 298 times
mean of y_train: 172.424
revenue of mean price offering: 17343.63600000005
```

```
10 the agent wins 189 times
```

1.3 内部仕様

bayes_opt.py

load_sanfrancisco 関数によって San Francisco-listing.csv を読み込み、データの前処理ののちに X_pro, y_pro を得る。

spaces には SVR の各パラメータ (p, C, eps) の定義域を与えて、skopt.gp_minimize により bayes_opt 関数を最小化するようなパラメータを探索する。

bayes_opt では与えられたパラメータの SVR を構成して、交差検証によって得た決定係数 (5 回の平均) に-1 を掛けたものを返す。(決定係数は 1 に近づくように最大化したいため)

agent.py

load_sanfrancisco 関数によって得た X, y について、前半を訓練データ、後半を評価データとする。

bayes_opt.py によって発見した最適なパラメータで SVR を構成して、訓練データによって SVR を学習させ、それによって評価データを予測する。

コマンドライン引数を m として、SVR による予測価格の $m\%$ が提示価格となる。

リストの価格の 30% は原価として考えて、各提示価格について提示価格がリストの価格を下回ったときに (提示価格 - リストの価格 * 0.3) が利益となる。これらの総利益をエージェントの評価基準とする。利益はマイナス (すなわち赤字) となることも考えられる。

1.4 評価結果

選択した説明変数のうち数値データ ('latitude', 'longitude', 'accommodates', 'number_of_reviews', 'review_scores_rating') と目的変数'price' との相関を図 1 に示す。

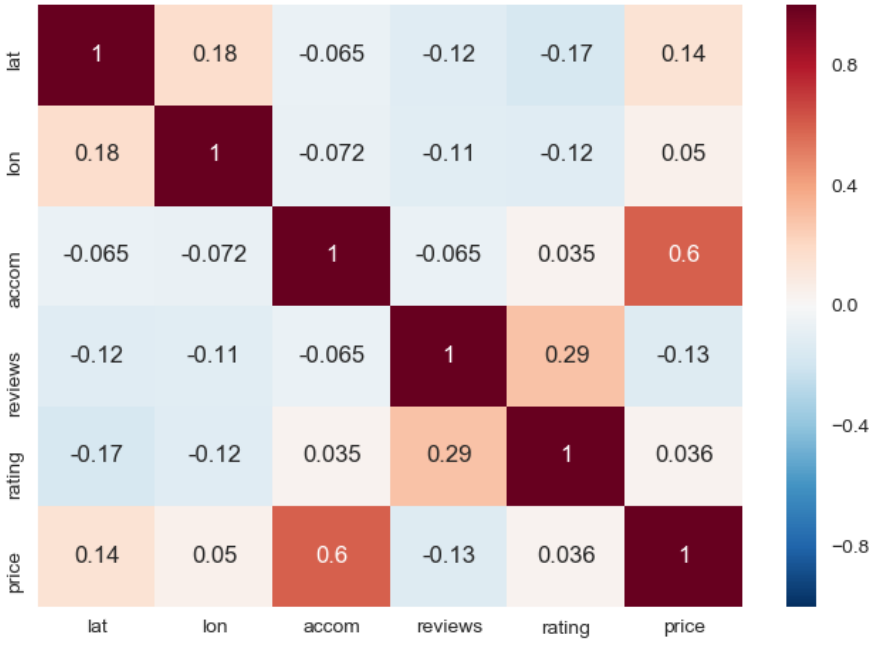

図 1　Correlation coefficient

SVR の予測精度 (決定係数) についてベイズ最適化により最適なハイパーパラメータ C, σ, ϵ を探索した。探索範囲は $2^{-1} \leq \sigma \leq 2^{5}$, $2^{-5} \leq C \leq 2^{15}$, $2^{-10} \leq \epsilon \leq 2^{-3}$ である。図 2 にベイズ最適化で探索したパラメータ C, σ とそのときの決定係数の値を示す。(データ数は 500 で 100 回探索した)

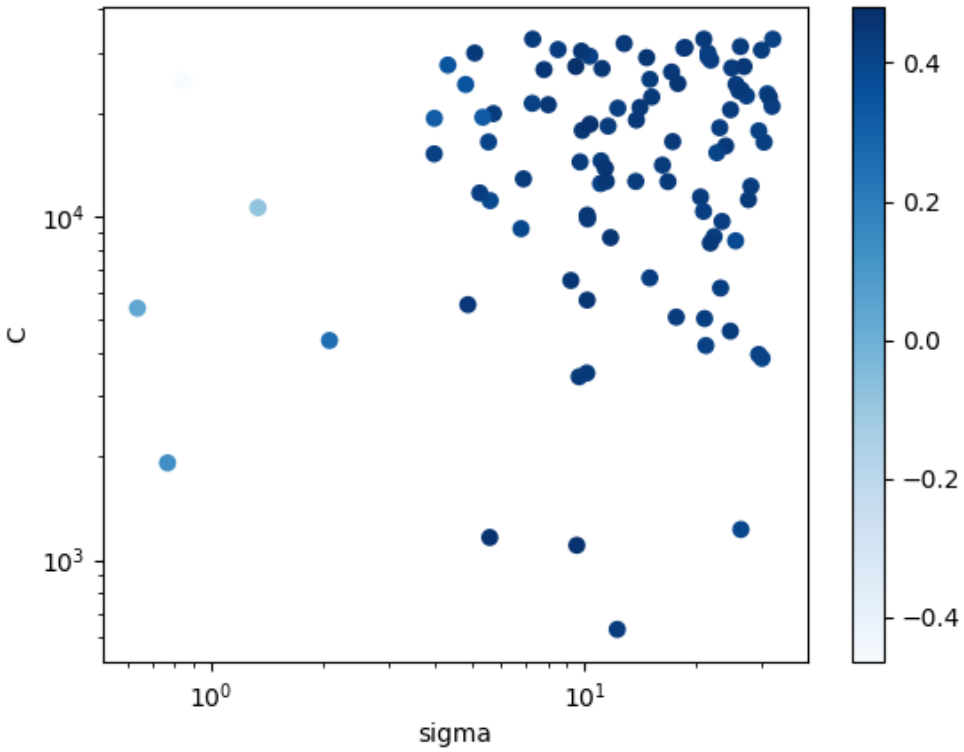

図 2　Bayes Optimization

決定係数は $\sigma = 9.50, C = 2.73 * 10^4, \epsilon = 9.27 * 10^{-2}$ のときに 0.481 と最大になった。

データ数 2000 を SanFraincisco リスティングデータからサンプリングして、その前半 1000 個を訓練データ、後半 1000 個を評価データとした。提示価格を予測価格の何 % とするかを変化させた時の総利益の変化を表 1 に示す。

(%)	総利益	成功回数	効率 (総利益/成功回数)
70	44470	837	53.1
75	47421	793	59.7
80	48558	739	65.7
85	49322	689	71.5
90	48622	629	77.3
95	47511	568	83.6
100	46029	510	90.2
105	42665	449	95.0
110	39205	394	99.5
115	37179	351	105.9
120	35160	313	112.3
mean	33871	359	94.3

表 1　agent by percentage

2　課題 4-2

2.1　プログラム概要

リスティングデータに従う貸し手 1 人と価格推薦エージェント 1, エージェント 2 が存在するとする。エージェントが最安の値段をつけたときに借り手がついて、収入が得られるとする。ベイズ最適化で最も精度が良かったパラメータの SVR を利用したものをエージェント 1、あまり精度が良くない SVR を利用したものをエージェント 2 とする。

2.2　外部仕様

./multiagent.py で実行する。以下は multiagent.py の実行例である。エージェント 1, 2 の予測精度 (決定係数), 総利益, 成功回数, 効率 (総利益/成功回数) を出力する。

```
$ ./multi_agent.py
load sanfrancisco: read 2000 / 6196 data
=== SVRAgent start ===
1 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: 0.4797468020396928
2 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: 0.33178028654882397
3 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: 0.4714137117838899
4 th cross validation...
```

```
cvxopt.solvers.qp: optimization succeeded
R2 score: 0.49906353860044683
5 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: 0.31606765685527194
5-fold average score: 0.41961439916562504
cross val score: 0.41961439916562504
start SVR fit...
cvxopt.solvers.qp: optimization succeeded
start SVR predict...
=== SVRAgent start ===
1 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: -0.05695700854191843
2 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: 0.03379843202281596
3 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: -0.00822695472188295
4 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: 0.06128181455545334
5 th cross validation...
cvxopt.solvers.qp: optimization succeeded
R2 score: 0.017475693763498
5-fold average score: 0.009474395415593185
cross val score: 0.009474395415593185
start SVR fit...
cvxopt.solvers.qp: optimization succeeded
start SVR predict...
revenue by SVR agent1: 16242.43831357982
the agent1 wins 340/1000 times
revenue1 per 1: 47.77187739288182
revenue by SVR agent2: 23875.955321474925
the agent2 wins 436/1000 times
revenue2 per 1: 54.761365416226894
```

2.3 内部仕様

multi_agent.py

load_sanfrancisco 関数によって San Francisco-listing.csv を読み込み、データの前処理ののちに X, y を得る。

load_sanfrancisco 関数によって得た X, y について、前半を訓練データ、後半を評価データとする。

SVRAgent 関数はパラメータを引数にとって、交差検証で精度を示したのち訓練データで学習、評価データの予測を行う。提示価格は予測価格の 85% とする。

エージェント 1, エージェント 2, リスティングデータの価格のうち、最小のものに借り手がつくとする。エージェントに借り手がついた場合、(提示価格 - リストの価格 * 0.3) がエージェントの利益となる。

2.4 評価結果

精度の良い Agent1($\sigma = 9.50, C = 2.70 * 10^4, \epsilon = 9.00 * 10^{-2}$) と精度のあまり良くない Agent2($\sigma = 1.0, C = 1.0, \epsilon = 0.1$) の総利益, 成功回数, 効率は表 2 のようになった。サンプリングされたデータを変更して 5 回実行した。

(Agent1) 決定係数	総利益	成功回数	(Agent2) 決定係数	総利益	成功回数
0.419	16242	340	0.00947	23875	436
0.473	19088	386	0.0000350	23035	396
0.528	17584	384	0.0115	24607	422
0.486	17119	338	0.0175	24666	439
0.442	16432	330	-0.0013	24807	450

表 2　Agent1 vs Agent2

3 考察

課題 4-1 では、表 1 より提示価格を予測価格の 85% とした場合が総利益が最大になった。提示価格の予測価格に対する割合を下げることで、予測価格がリスティングデータの価格より高い場合にも提示価格が予測価格を下回る場合が多くなり、成功回数は上がるが、一回の提供あたりの利益が小さくなる。そのトレードオフを考慮して 85% あたりが望ましい。

課題 4-2 で 2 つのエージェントが存在する場合、どちらのエージェントが他より利益を上げることができるかは単純に決定することはできない。(予測精度が高い方が総利益が常に高くなるとはいえない) 価格の予測精度が低くても他のエージェントとリスティングデータの価格より低い価格を提示することができれば、利益を得ることができて、他のエージェントはそのとき利益を逃すからである。原価 (30%) を下回らない範囲で最低の価格を提示するようなエージェントがあれば、他のエージェントはどんなに正確に価格予測してリスティングデータに対し大きな利益が得られるような提示を行っても、それを得る機会を失い、最終的に原価に近い価格を提示し続けたエージェントが最大の総利益を得る。

しかし、このように価格競争が発生して提示価格は限りなく原価に近づいた場合、一方のエージェントが価格競争に勝利して全ての借り手に民泊を提供できたとしてもほとんど利益を上げられなくなる。それぞれのエージェントが自身の利益を追求すると、エージェント全体の利益は最小化されてしまう。(囚人のジレンマといえる)

一方、お互いのエージェントが交代で市場に参加すれば、それぞれのエージェントは借り手を独占することはできないが、結果的に総利益を最大にできることがある。

3.1 ベイズ最適化

ハイパーパラメータの探索において、一般的な手法としてグリッドサーチがある。グリッドサーチは離散的なパラメータの組の候補について全探索を行う。一方、ベイズ最適化では、ある連続的な定義域をもつ入力(パラメータ)x に対して、ブラックボックス関数 $f(x)$ を最小化するような x を逐次的に求める手法である。次の観測点 x の候補の求め方は全探索ではなく、これまでの関数の評価値に基づいて判断する。

EI 戦略のベイズ最適化の場合、

3.2 回帰モデルの評価

決定係数は

$$R2 = 1 - \frac{\sum_{i=0}(y_i - \hat{y_i})^2}{\sum_{i=0}(y_i - \bar{y_i})^2} \tag{1}$$

で表すことができる。R2 の分子は回帰モデルでは説明できない目的変数のばらつき (変動)、分母は目的変数の平均値のまわりにおけるばらつき (変動) を表していて、R2 は目的変数自体の変動のうち、回帰モデルで説明できるものの割合を示しているといえる (決定係数が 1 に近づくほど良いモデルであるといえる) 決定係数 (R2) は分子に 2 乗誤差 (MSE)$\sum_{i=0}(y_i - \hat{y_i})^2$ が存在していて、MSE に対して、R2 は単調減少する。

平均二乗誤差 (MSE) は

$$\sum_{i=0}(y_i - \hat{y_i})^2/n$$

平均絶対誤差 (MAE) は

$$\sum_{i=0}\|y_i - \hat{y_i}\|/n$$

と表される。

4 参考文献

- scikit-learn と Tensorflow による実践機械学習 Aurelien Geron 著
- Python 機械学習プログラミング Sebastian Raschka 著
- ベイズ最適化 (https://www.kumilog.net/entry/bayesian-optimization)
- 精度評価指標と回帰モデルの評価 (https://funatsu-lab.github.io/open-course-ware/basic-theory/accuracy-index/)
- 機械学習のためのベイズ最適化入門 (https://www.slideshare.net/hoxo_m/ss-77421091)