計算機科学実験及演習 4 (エージェント) レポート 4

佐竹誠

2018/11/16

1　課題概要

レポート 3 で作成した SVR を用いて類似の宿泊施設の提示価格を予測し、それに応じて自己の宿泊施設の価格推薦を行うエージェントの作成を考えます。まず価格推薦戦略について異なるふたつの状況を仮定した上で考察し、それぞれ戦略を考案します。その後それらの戦略の性能をシミュレートするプログラムを作成します。

2　価格推薦戦略

2.1　課題 4-1

SVR での予測が 100% 正しいとすると予測よりわずかに小さい価格を設定すれば収入を最大化することができます。しかし実際に SVR で 100% 正しい予測をすることはできません。そのため収入を最大化しようとするあまり予測との差を小さくしすぎると SVR の誤差で競合施設の価格を大きく予測してしまい、結果自己の施設の価格の方が高くなってしまう可能性があります。したがってまずは作成した SVR の誤差を確認し、予測よりどれほど価格を設定すれば安全であるかを考えます。相関係数を調べた結果から学習に使用する属性を accomodates, bedrooms, beds, minimum_nights の 4 属性で予測を行なったところ、平均二乗誤差は 18486.5 となりました (カーネルトリックなし、C=1000)。すなわち実際の誤差は約 135 と考えることができます。以上より SVR の誤差を帳消しするために予測値から 135 を引いた後、それより少し小さな値を取るために 0.9 倍した価格に設定する戦略を取ります。

2.2　課題 4-2

他の価格推薦エージェントも同様に予測値の少し下という価格推薦を行うと、価格競争が発生して双方の利潤が失われてしまいます。今回は利潤の合計を大きくすることを目標とし、価格推薦エージェントが自己の他にもう 1 つ存在すると考えるとそれぞれの利潤がちょうど半分になることが理想です。そこで毎回市場に参加するのではなく、確率的に市場に参加することを考えます。こうすることで費用を肥大化させることなく利潤合計を維持することが可能になります。条件より 1/2 の確率で市場に参加するエージェントを作成します。またとにかく入札することを防ぐために市場に参加するには市場価格の半分の費用がかかると仮定し、それぞれのエージェントにかかる費用も計算します。

3　シミュレーションプログラム概要

民泊のリステイングデータが書かれたデータファイルを読み込み、サポートベクター回帰によって回帰式を作成、民泊の価格を予測し、その予測値を用いて価格設定を行い、利益を計算します。

4　外部仕様

4.1　プログラム名

simulation.cpp がプログラムファイルです。

表 1　相関係数表

属性	相関係数
host_listings_count	-0.156674
host_total_listings_count	-0.156674
zipcode	-0.1798
latitude	0.100701
longitude	0.151766
accommodates	0.471581
bathrooms	0.0106641
bedrooms	0.391548
beds	0.335893
guests_included	0.133243
minimum_nights	0.375137
maximum_nights	0.282948
availability_30	0.254088
availability_60	0.239205
availability_90	0.206584
availability_365	0.0620253
number_of_reviews	-0.0478367
review_scores_rating	0.163354
review_scores_accuracy	0.132849
review_scores_cleanliness	0.177434
review_scores_checkin	0.0729342
review_scores_communication	0.0803387
review_scores_location	0.0973538
review_scores_value	0.0813066
calculated_host_listings_count	-0.13964

4.2　ファイルの説明

4.2.1　simulation.cpp

サポートベクター回帰によって作成された回帰式に基づいて民泊の価格を予測、利益を計算し表示するプログラムファイルです。

4.2.2　quadprog++.cc

二次計画問題を解くライブラリです。simulatin.cpp で使用しています。

4.2.3 quadprog++.hh

quadprog++.cc のヘッダーファイルです。simulation.cpp でインクルードしています。

4.2.4 SFlisting.csv

Airbnb から提供されているリスティングデータ San Francisco-listings.csv を本プログラムで利用できるよう編集したものです。

4.2.5 Makefile

コンパイルするための make コマンドを使えるようにするファイルです。

4.3 入力方法

プログラムを実行すると、まず「データファイルを指定してください：」と表示されるので、使用するデータファイルを指定してください。ただしデータファイルは最初の行に各列の属性名が、それ以降にデータが記されているものに限ります。さらにデータには属性 price が存在し、price のデータは$から始まる価格が格納されているとし、他の属性のデータは数値データとします。またデータ数の上限は 100 個になっており、データ数が 100 個より多いデータファイルを入力した場合は上から 100 個のデータを使用します。上から 80 個のデータを学習に、残り 20 個のデータを評価に使用します。

次に「データの次元数を指定してください：」と表示されるので、使用するデータの次元数を指定してください。

次に「使用する属性を指定してください：」と表示されるので、使用する属性の名前を次元数分だけスペースで区切って入力してください。

次に「使用するカーネルを指定してください (0:カーネルトリックなし　1:多項式カーネル　2:ガウスカーネル)：」と表示されるので、使用したいカーネルを数字で指定してください。

最後に「シミュレーション方法を指定してください（1:課題 4-1; 2:課題 4-2)：：」と表示されるので、使用するシミュレーション方法を指定してください。それぞれのシミュレーション方法については前章の価格推薦戦略を参照ください。

4.4 出力

シミュレーション方法で出力形式が変わります。

4.4.1 課題 4-1

5 分割したブロックごとの結果を表示したのち、平均二乗誤差の平均値を出力します。各結果の意味は以下の通りです。

- f[]
  学習によって得た推定値です。評価用データ分だけ表示されます。
- label[]
  評価用データの price です。評価用データ分だけ表示されます。

- 理論上の最大収益 (実際の市場価格の和)
  評価用データの price の和です。収益がこの値に近いほど良い価格を設定できています。
- SVR を用いて価格設定を行なったときの収益
  SVR を用いて価格設定を行い、label の値よりも小さかった価格の和です。収益を表します。
- 単純な価格設定を行なった時の収益
  SVR の価格設定の性能を評価するために計算した、単純な価格設定を行なった時の label の値よりも小さかった価格の和です。具体的には学習用データの price の平均をすべての施設に設定するという戦略を取ります。

4.4.2 課題 4-2

- 理論上の最大収益 (実際の市場価格の和)
  評価用データの price の和です。エージェント A、エージェント B の収益が合計がこの値に近いほど良い価格を設定できています。
- エージェント A の収益、エージェント B の収益
  SVR を用いて価格設定を行い、それぞれシミュレーションに基づいて入札したときの収益です
- エージェント A の費用、エージェント B の費用
  シミュレーションを行った結果それぞれのエージェントが必要になった費用です。

4.5 コンパイル方法

プログラムファイルがあるディレクトリで make コマンドを実行すると quadprog++.cc と simulation.cpp のコンパイルが行われ、simulation という実行可能ファイルができます。実行後は画面の指示に従って条件を入力してください。

4.6 実行例

4.6.1 実行例 1:課題 4-1

学習データを SFlisting.csv、次元数を 4、属性を accomodates, bedrooms, beds, minimum_nights、カーネルトリックなしでシミュレートさせた結果は次のようになります。

```
$ ./simulation
データファイルを指定してください：SFlisting.csv
データの次元数を指定してください：4
使用する属性を指定してください：accomodates bedrooms beds minimum_nights
使用するカーネルを指定してください (0:カーネルトリックなし　1:多項式カーネル　2:ガウスカーネル)：0
シミュレーション方法を指定してください（1:課題 4-1; 2:課題 4-2）：1
f[0]:54.8481 label[0]:92
f[1]:84.8792 label[1]:166
f[2]:84.8792 label[2]:125
f[3]:70.0154 label[3]:90
f[4]:55.1914 label[4]:100
f[5]:195.52 label[5]:216
f[6]:77.458 label[6]:80
f[7]:70.0154 label[7]:46
f[8]:70.0154 label[8]:44
f[9]:143.901 label[9]:170
f[10]:70.0154 label[10]:100
f[11]:98.9848 label[11]:108
f[12]:33.1337 label[12]:30
f[13]:55.9098 label[13]:45
f[14]:84.3483 label[14]:109
f[15]:62.6339 label[15]:80
f[16]:224.872 label[16]:100
f[17]:85.41 label[17]:38
f[18]:173.253 label[18]:110
f[19]:70.0154 label[19]:270
理論上の最大収益 (実際の市場価格の和)：$2119
SVR を用いて価格設定を行なったときの収益：$1152.69
単純な価格設定を行なった時の収益：$581.85
```

4.6.2　実行例 2:課題 4-2

学習データを SFlisting.csv、次元数を 4、属性を accomodates, bedrooms, beds, minimum_nights、カーネルトリックなしでシミュレートさせた結果は次のようになります。

```
$ ./simulation
データファイルを指定してください：SFlisting.csv
データの次元数を指定してください：4
使用する属性を指定してください：accomodates, bedrooms, beds, minimum_nights
使用するカーネルを指定してください (0:カーネルトリックなし　1:多項式カーネル　2:ガウスカーネル)：0
シミュレーション方法を指定してください（1:課題 4-1; 2:課題 4-2）：2
理論上の最大収益 (実際の市場価格の和)：$1706
エージェント A の収益：$529.899
エージェント B の収益：$217.185
エージェント A の費用：$596.5
エージェント B の費用：$389
```

4.7　エラー処理

4.7.1　指定されたデータファイルがないとき

全ての入力が終わった時点で”ファイルが見つかりません”と表示し、プログラムを終了します。

4.7.2　指定されたカーネルが正しくないとき

全ての入力が終わった時点で”正しくカーネルを指定してください (0:カーネルトリックなし　1:多項式カーネル　2:ガウスカーネル)”と表示し、プログラムを終了します。

4.7.3　指定された属性がないとき

全ての入力が終わった時点で”指定された属性が存在しません”と表示し、プログラムを終了します。

4.7.4　指定されたシミュレーション方法が正しくないとき

全ての入力が終わった時点で”正しくシミュレーション方法を指定してください（1:課題 4-1; 2:課題 4-2）”と表示し、プログラムを終了します。

4.7.5　データファイルに属性 price がないとき

データを読み取った時点で”価格が存在しません”と表示し、プログラムを終了します。

5　内部仕様

5.1　主要な大域変数の説明

5.1.1　double G[][]

二次計画問題を解く際に使用するデータを格納する double 二次元配列です。

5.1.2 double g0[]

二次計画問題を解く際に使用する double 配列です。関数の一階微分を表しており、本プログラムでは全ての要素に-1 が格納されています。

5.1.3 double CE[][]

二次計画問題を解く際に使用する double 二次元配列です。本プログラムではラベルの配列を設定しています。

5.1.4 double ce0[]

二次計画問題を解く際に使用する double 配列です。本プログラムでは全て 0 に設定しています。

5.1.5 double CI[][]

二次計画問題を解く際に使用する double 二次元配列です。本プログラムでは単位行列に設定しています。

5.1.6 double ci0[]

二次計画問題を解く際に使用する double 配列です。本プログラムでは全て 0 に設定しています。

5.1.7 double pre_alpha[]

二次計画問題の計算結果として出力される α ベクトルを格納する double 配列です。

5.1.8 int n

二次計画問題を解く際に使用する int 変数です。本プログラムではデータの数が設定されています。

5.1.9 int m

二次計画問題を解く際に使用する int 変数です。CE と ce0 の配列の要素数を指定しており、本プログラムでは n の 2 倍に設定しています。

5.1.10 int p

二次計画問題を解く際に使用する int 変数です。CI と ci0 の要素数を指定しており、本プログラムでは 1 に設定しています。

5.1.11 int N

データの次元数を表す int 型の変数です。入力から受け取った値をそのまま格納します。

5.1.12 int kernel

使用するカーネルを表す int 型の変数です。0,1,2 の三値しか取らず、それぞれカーネルトリックなし、多項式カーネル、ガウスカーネルを表します。入力から受け取った値をそのまま格納します。

5.1.13 std::string file_name

使用するデータファイルの名前を表す string 型の変数です。入力から受け取った値をそのまま格納します。

5.1.14 int simulation

シミュレーション方法を表す int 型の変数です。入力から受け取った値をそのまま格納します。

5.1.15 double sigma

ガウスカーネルを使用する際に使用する double 型の変数です。本プログラムでは $\sqrt{5}$ を設定しています。

5.1.16 int data_n

学習用データと評価用データの分割を表す int 型の変数です。データ数をこの変数で割った値が評価用データの数になります。本プラグラムでは 5 に指定しています。

5.1.17 double data[][]

利用する生のデータを格納する double 配列です。

5.1.18 double label[]

データのラベルを格納する double 配列です。

5.1.19 double theta

得られた識別器の閾値を格納する double 型の変数です。

5.1.20 int total_data_size

使用するデータの数を表す int 型の変数です。

5.1.21 int training_data_size

学習用データの数を表す int 型の変数です。

5.1.22 int estimate_data_size

評価用データの数を表す int 型の変数です。

5.1.23 double epsilon

サポートベクトル回帰で用いる double 型の定数 ϵ です。誤差の許容範囲を表し、本プログラムでは 0.1 に設定しています。

5.1.24 double C

サポートベクトル回帰で用いる double 型の定数 C です。モデルの単純さと推定の悪さのトレードオフを決定するパラメータで、本プログラムでは 1000 に設定しています。

5.1.25 std::string using_labels[]

サポートベクター回帰の学習に使用する属性の名前を保持しておく string 配列です。入力をスペース区切りで格納します。

5.2 各関数の説明

5.2.1 double kernel_result(double* x, double* y, int kernel, int degree, double sigma, int d)

カーネルの計算結果を得るための関数です。計算結果を double 型で返します。各引数の意味は以下の通りです。

- double* x,y
  計算する二つの配列の引数です。
- int kernel
  使用するカーネルを表す引数です。(0:内積 1:多項式カーネル 2:ガウスカーネル)
- int degree
  データの次元数を表す引数です。
- double sigma
  ガウスカーネル計算時に使用するシグマ定数を表す引数です。

5.2.2 double get_norm(double* x, double* y, int degree)

二乗ノルムを得るための関数です。計算結果を double 型で返します。各引数の意味は以下の通りです。

- double* x,y
  計算する二つの配列の引数です。
- int degree
  データの次元数を表す引数です。

6 結果

SFlisting.csv のすべてのデータに対して、属性を accomodates, bedrooms, beds, minimum_nights に指定して学習し、性能を評価しました。

6.1 課題 4-1

f[0]:54.8481 label[0]:92
f[1]:84.8792 label[1]:166
f[2]:84.8792 label[2]:125
f[3]:70.0154 label[3]:90
f[4]:55.1914 label[4]:100
f[5]:195.52 label[5]:216
f[6]:77.458 label[6]:80
f[7]:70.0154 label[7]:46
f[8]:70.0154 label[8]:44
f[9]:143.901 label[9]:170
f[10]:70.0154 label[10]:100
f[11]:98.9848 label[11]:108
f[12]:33.1337 label[12]:30
f[13]:55.9098 label[13]:45
f[14]:84.3483 label[14]:109
f[15]:62.6339 label[15]:80
f[16]:224.872 label[16]:100
f[17]:85.41 label[17]:38
f[18]:173.253 label[18]:110
f[19]:70.0154 label[19]:270
理論上の最大収益 (実際の市場価格の和)：$2119
SVR を用いて価格設定を行なったときの収益：$1152.69
単純な価格設定を行なった時の収益：$581.85

6.2 課題 4-2

理論上の最大収益 (実際の市場価格の和)：$1706
エージェント A の収入：$626.964
エージェント B の収入：$400.862
エージェント A の費用：$577.5
エージェント B の費用：$599

7　考察

7.1　課題 4-1

7.1.1　価格設定

今回は SVR の予測値の誤差が非常に大きいため予測値から$135 引いたあと 0.9 倍するという加工を施しました。135 という値は課題 3 で作成したプログラムを用いて交差検証した結果の二乗平均誤差から得られた値ですが、データをみるとわかるように価格が$135 以下の施設も多く、良い加工とは言えません。また予測値が 135 を下回った際には 0.9 倍するだけになってしまい誤差を解消できない点も問題です。この点を根本的に改善するには SVR の精度を向上させることが必要になります。

7.1.2　SVR の精度

価格推薦戦略の章でも述べたように、平均二乗誤差が 18486.5 と非常に大きく精度が悪いです。使用した属性は相関係数上位 4 つです。より多くの属性を使用した方が精度が上がるのかもしれませんが、表 1 を見ればわかるように相関係数が 0 付近の属性が多く、追加しても精度にあまり差がなかったため本プログラムではこの 4 つの属性を使用しています。

定数 C に関しては平均二乗誤差がもっとも小さくなるのは 100 のときの 9328.43 ですが、予測値からその平方根を引き 0.9 倍した値を価格にしてシミュレートしたところ 1000 のときが最も収入が大きくなったため 1000 を用いています。

より精度が高くなれば誤差解消の加工が必要なくなるため、より適切な価格設定が行えるようになると考えています。

7.1.3　データセットによる変化

データセットに関しては良い相関係数を持つ属性が存在するかどうかが重要です。今回使用したリスティングデータ San Francisco-listings.csv では相関係数が 0 付近の属性が多く、結果 4 属性しか学習に使用しませんでした。より多くの属性を使用することができれば SVR の精度も上がりますし、逆にすべての属性の相関係数が 0 付近であれば SVR の精度はさらに悪化します。

良い属性を持つデータセットという観点では、特定の地域によらず世界各地の民泊施設のデータセットであれば緯度経度など位置による価格の変化が大きくなることが推測できます。また今回は試せていませんが説明文の文字数なども属性として使用できるかもしれません。

7.2　課題 4-2

7.2.1　エージェントの戦略と評価基準

本プログラムではエージェントの戦略として 1/2 の確率で市場に参加するという戦略を採用しました。

これは収入の合計を最大化するという評価基準においてあまり良い戦略ではありませんでした。なぜならどちらも出店しないことでその分の収入が得られなくなることはあっても、収入の合計が増えることはないからです。この評価基準であれば毎回出店することでどちらかが得られるはずだった収入を取りこぼすことはなくなります。しかし無駄な費用が大きくなってしまいます。

一方費用を最小化するという評価基準においても、お互いが出店してどちらかの費用が無駄になってしまう

ことがあるため良い戦略ではありません。この評価基準を採用するとなるべく出店しないことが費用を抑えることになりますが、収入も小さくなってしまいます。

以上より、評価基準によって市場に参加する確率を変動させるべきだと考えます。元手が少なく費用を抑えたいときは確率を下げ、無駄になる費用があっても収入を求めるなら確率を上げるべきです。