計算機科学実験及演習 4 (エージェント) レポート 3

佐竹誠

2018/11/2

1　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、出力します。回帰式によって民泊の価格を予測します。ハイパーパラメータなどを調整し予測精度を評価するために交差検証を行うモードと、データすべてを用いて回帰式を作成するモードがあります。

2　外部仕様

2.1　プログラム名

SVR.cpp がプログラムファイルです。

2.2　ファイルの説明

2.2.1　SVR.cpp

サポートベクター回帰によって作成された回帰式を表示するプログラムファイルです。

2.2.2　quadprog++.cc

二次計画問題を解くライブラリです。SVR.cpp で使用しています。

2.2.3　quadprog++.hh

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

2.2.4　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。

2.2.5　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。

2.3　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。

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「C を指定してください：」と表示されるので、定数 C を入力してください。ただし定数 C とはサポートベクター回帰で使用するパラメータで、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す。

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「交差検証を行いますか？ (y/n)：」と表示されるので、交差検証を行う場合は y、データ全てで学習する場合は n を入力してください。

2.4　出力

交差検証を行うかどうかで出力形式が変わります。

2.4.1　交差検証を行った場合

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。

- weight
  学習によって得た各ベクトルの重みです。次元数分だけ表示されます。閾値と合わせて回帰式を表します。
- theta
  学習によって得た閾値です。重みと合わせて回帰式を表します。
- 平均二乗誤差
  回帰式を評価用データに適用した値と真値との平均二乗誤差です。回帰式の精度の良さを表す評価値で、小さいほど精度の良い回帰式です。

2.4.2　交差検証を行わなかった場合

データ全てを使って学習して得た回帰式の重みと閾値を表示します。それぞれの意味は上記の「交差検証を行なった場合」にあるものと同じです。また学習データの次元数が 2 次元出会った場合のみグラフを出力します。

2.5　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 SVR.cpp のコンパイルが行われ、svr 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。

2.6　実行例

2.6.1　実行例 1:交差検証を行なった場合

学習データを SFlisting.csv、次元数を 3、属性を review_scores_rating”, ”host_total_listings_count”, ”number_of_reviews”、カーネルトリックなしで交差検証させた結果は次のようになります。

```
$ ./svr
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：3
使用する属性を指定してください：review_scores_rating host_total_listings_count number_of_reviews
C を指定してください：1000
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
交差検証を行いますか？ (y/n)：y
0
weight[0]：0.0375695
weight[1]：-0.0501673
weight[2]：-0.0435527
theta：-0.0638329
平均二乗誤差：0
.....
1
weight[0]：0.0297821
weight[1]：-0.0446534
weight[2]：-0.0110916
theta：-0.0695339
平均二乗誤差：0.0368149
.....
2
weight[0]：-0.0195779
weight[1]：-0.0573687
weight[2]：0.0696994
theta：-0.438613
平均二乗誤差：0.0150747
.....
3
weight[0]：-0.019931
weight[1]：-0.031791
weight[2]：0.041218
theta：0.0515753
平均二乗誤差：0.0122181
.....
4
weight[0]：-0.0683545
weight[1]：-0.0974951
weight[2]：0.28663
theta：-0.538247
平均二乗誤差：0.106586
.....
平均：0.0341388
```

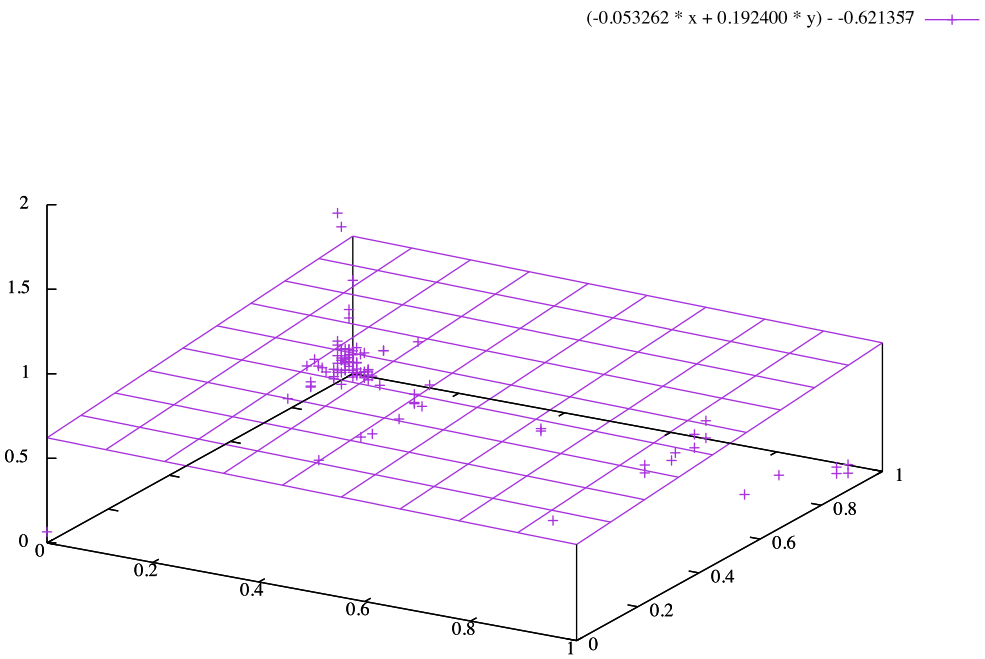

図 1　実行例 2 の出力画像

2.6.2　実行例 2:交差検証を行なわない場合

学習データを SFlisting.csv、次元数を 2、属性を review_scores_rating", "host_total_listings_count"、カーネルトリックなしで交差検証させずに学習させたは次のようになります。

```
$ ./svr
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：0
使用する属性を指定してください：review_scores_rating host_total_listings_count
C を指定してください：1000
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
交差検証を行いますか？ (y/n)：n
weight[0]：-0.0532617
weight[1]：0.1924
theta：-0.621357
```

2.7　エラー処理

2.7.1　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。

2.7.2　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。

2.7.3　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。

2.7.4　"交差検証を行いますか？"に対して y と n 以外が入力されたとき

全ての入力が終わった時点で”y または n を入力してください”と表示し、プログラムを終了します。

3　内部仕様

3.1　主要な大域変数の説明

3.1.1　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。

3.1.2　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。

3.1.3　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。

3.1.4　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。

3.1.5　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。

3.1.6　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。

3.1.7 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。

3.1.8 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。

3.1.9 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。

3.1.10 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。

3.1.11 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。

3.1.12 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。

3.1.13 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。

3.1.14 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。

3.1.15 int data_n

交差検証の分割数を表す int 型の変数です。本プラグラムでは 5 に指定しています。

3.1.16 double data[][]

利用する生のデータを格納する double 配列です。

3.1.17 double label[]

データのラベルを格納する double 配列です。

3.1.18 double theta

得られた識別器の閾値を格納する double 型の変数です。

3.1.19 int total_data_size

使用するデータの数を表す int 型の変数です。

3.1.20 int training_data_size

交差検証で用いる学習用データの数を表す int 型の変数です。

3.1.21 int estimate_data_size

交差検証で用いる評価用データの数を表す int 型の変数です。

3.1.22 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。

3.1.23 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、入力から受け取った値をそのまま格納します。

3.1.24 bool estimate_bool

交差検証を行うかどうかを表す bool 型の変数です。入力によって変化し、交差検証を行う場合は true、行わない場合は false が格納されます。

3.1.25 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。

3.2 各関数の説明

3.2.1 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。

3.2.2 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。

4 結果

各カーネルを SFlisting.csv のすべてのデータに対して、次元数を 3、属性を review_scores_rating, host_total_listings_count, number_of_reviews に指定して学習しました。また、事前に交差検証を行い、各カーネルについて C = 1,10,100,1000 の中から最適な C を選択しました。

4.1 評価値 (平均二乗誤差)

4.1.1 カーネルトリックなし (C=1)

0.0122746

4.1.2 多項式カーネル (C=1)

0.166481

4.1.3 ガウスカーネル (C=1)

0.133065

4.2 回帰式

4.2.1 カーネルトリックなし (C=1)

- weight[0] : -0.0746138
- weight[1] : -0.0702444
- weight[2] : 0.0956708
- theta : -0.415366

4.2.2 多項式カーネル (C=1)

- weight[0] : -0.041004
- weight[1] : -0.019693
- weight[2] : 0.0106131
- theta : -0.767934

4.2.3 ガウスカーネル (C=1)

- weight[0]：-0.399473
- weight[1]：-0.160136
- weight[2]：0.187056
- theta：-0.808821

5 考察

5.1 カーネル間の予測精度の比較

結果からわかるように、カーネルトリックなしの誤差が大幅に小さく、次にガウスカーネル、多項式カーネルが良い精度を出しています。これは使用した次元数が少なく、複雑な分布を描くことがなかったためだと考えられます。

5.2 使用した属性

本レポートで提示した結果では主に review_scores_rating, host_total_listings_count, number_of_reviews を使用しています。各属性を選択した理由は次の通りです。

5.2.1 review_scores_rating

レビュー評価の高い民泊は設備やサービスが充実しており、価格が高くなると考えました。しかし、安いことがレビュー評価につながるケースもあるため一概に相関があるとは言えません。

5.2.2 host_total_listings_count

民泊を多く経営しているホストはノウハウや大量仕入れによってコストが下げられ、価格が低くなると考えました。

5.2.3 number_of_reviews

レビューの多さは利用者の多さに比例するため、価格が低い民泊で多くなると考えました。