

8章 価格モデルの構築(1)

06t4056n 三沢博章

はじめに

- ・価格予測モデルを構築する方法の紹介
- ・ワイン価格を使ったデータセットを生成



- ・K近傍法を利用
- ・重み付けK近傍法を利用
- ・クロス評価の利用



各関数ごとにプログラムを動かしその動作を確認する

ワインの価格の設定とデータセットの構築

<価格の計算>

- ・ワインのレーティングと年齢(古さ)に基づき価格を付ける
- ・ワインにはそれぞれピーク年齢がある
- ・ピーク年齢に達するまでワインの価格は上がり

<データセットの構築>

- ・ワインを200本生成
- ・年齢とレーティングをランダムに生成

実行結果

```
>>> import numpredict
>>> numpredict.wineprice(95.0,3.0)
21.111111111111114
>>> numpredict.wineprice(95.0,8.0)
47.5
>>> numpredict.wineprice(99.0,1.0)
10.102040816326529
>>> data=numpredict.wineset1()
>>> data[0]
{'input': (99.153198550348577, 3.7908327838638511), 'result':
  22.562215053701468
}
>>> data[1]
{'input': (54.513746845439812, 33.607342800133431), 'result': 0.0}
```

K近傍法の利用

ワインの価格を手動で求める

・・・K近傍法の利用



非常のよく似たアイテム群を探し、これらの平均をとる。

そのために、、、

- ・二つのアイテム(与えられたアイテムとデータセットの各アイテム)がどれほど似ているか測定する
- ・リストを作成、ソートしてもっとも近傍にあるアイテムからK個(K=5)の結果の平均をとる

実行結果

```
>>> reload(numpredict)
>>> numpredict.knnestimate(data,(95.0,3.0))
27.628614216597885
>>> numpredict.knnestimate(data,(99.0,3.0))
24.392931063880798
>>> numpredict.knnestimate(data,(99.0,5.0))
29.756276817254466
>>> numpredict.wineprice(99.0,5.0) # 実際の価格
30.306122448979593
>>> numpredict.knnestimate(data,(99.0,5.0),k=1)
36.46628956716679
```

重み付け

アイテム同士が似ているほど両者の距離が小さくなる



距離を重みに変換

反比例関数・・・4章に記載

減法関数・・・定数から距離を引く(0以下の場合には0)

ガウス関数・・・距離が0のとき1で距離が増すにしたがって減る。しかし完全な0にはならない。

重み付けK近傍法

重み付けK近傍法の関数と通常のK近傍法の違い

- ・単純平均をとらないで加重平均をとる

加重平均とは、、、

加重平均の各アイテムの値には、合計する前に、それぞれの重みを掛ける。そして合計が出たら、これを重みの合計で割る。

アイテム x_i に、重み w_i がついているとき

$$\frac{w_1 x_1 + \dots + w_i x_i}{w_1 + \dots + w_i}$$

実行結果

```
>>> reload(numpredict)
<module 'numpredict' from 'numpredict.py'>
>>> numpredict2.weightedknn(data,(99.0,5.0))
30.11386221469019
```

<通常のK近傍法の結果>

```
>>> numpredict.knnestimate(data,(99.0,5.0))
29.756276817254466
```

<実際の価格>

```
>>> numpredict.wineprice(99.0,5.0)
30.306122448979593
```

今回の重み付きK近傍法のほうがより正確な結果が得られた

クロス評価

クロス評価とは、、、
データを評価用のデータと予測用のデータに分け、
予測用のデータを与えた結果と評価用データから
得た正解の値の差を求め、その総和を計算する。

これよりアルゴリズムの良し悪しを判断する。

総和が多い・・・悪い

総和が少ない・・・良い

ここでは、近傍群の数(Kの値)を変えて結果を見る

実行結果

```
>>> reload(numpredict)
<module 'numpredict' from 'numpredict.py'>
>>>
    numpredict2.crossvalidate(numpredict.knnestimate,data)
466.14522559743483
>>> def knn3(d,v): return numpredict.knnestimate(d,v,k=3)
...
>>> numpredict.crossvalidate(knn3,data)
426.06464714055966
>>> def knn1(d,v): return numpredict.knnestimate(d,v,k=1)
...
>>> numpredict.crossvalidate(knn1,data)
535.06592018874176
```