

平成 28 年度 3 回生後期実験 (エージェント)

課題 1 SVM の作成

村田 叡

2016/10/14

1 プログラム概要

この課題ではサポートベクターマシン (SVM) の python3 による実装を行った。1 と -1 の 2 クラスのサンプル点集合を分類する評価機を生成する。線形評価関数又はカーネルトリックを用いた評価関数を生成する。外部ライブラリとして凡庸行列計算に numpy, プロットに matplotlib, 二次計画問題の計算に cvxopt を用いた。以下にその詳細を述べる。

2 外部仕様

2.1 svm.py

今回のコードは svm.py にて実装した。このコードは、引数としてサンプル点集合のファイル名、カーネル名、及びオプションをとる。サンプル点集合の形式については、実験のページに書かれてあるものに従った。カーネル名は、gauss, polynomial, sigmoid, linear のいずれかをとる。デフォルトのカーネル名は gauss である。-plot オプションを使うと結果を matplotlib でプロットしたものを表示する。

2.2 実行例と実行結果

```
# 必要なライブラリの導入
$ pip3 install -r requirements.txt
# サンプル点集合 sample_circle.dat のガウスカーネルの SVM を作成し、プロットする。
$ python3 svm.py sample_data/sample_circle.dat gauss --plot
>>  $\alpha$  : [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00 ...
>>  $\theta$  : 3.56258579268
>> passed :100 / 100
>> f(x) = +14.8961965139*K([31.0, 12.0],x) -6.46323001951*K([40.0, 24.0],x) ...
```

上記のように、各 α の値、 θ の値、サンプル点による識別器の識別率、識別器の関数が結果として得られる。なお、 $f(x)$ の K は 各カーネルを指す。

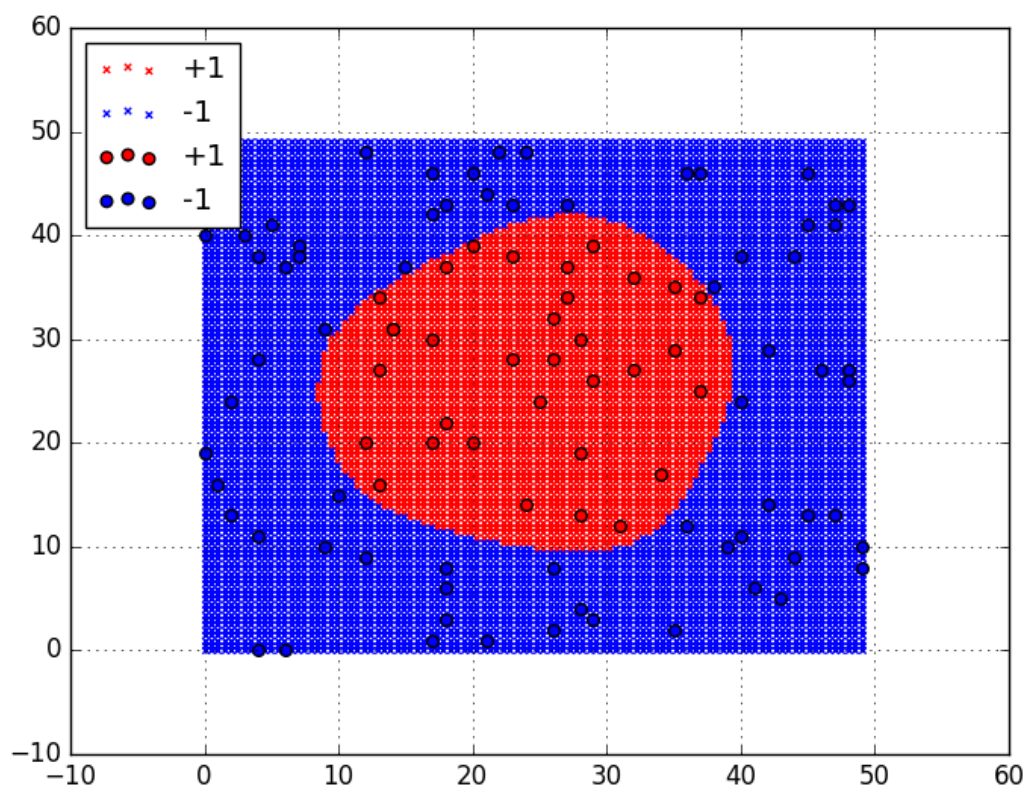


図 1 ガウスカーネルの SVM

3 内部仕様

以下では `svm.py` での大域変数と各関数について説明する。

3.1 kernels

この大域変数は、各カーネル関数の辞書である。辞書式のラムダ式で定義しておくことで容易に関数内でコード内で使用できる。python の計算は遅い可能性が高いので、numpy を多用している。

3.2 solve(x, y, kernel)

この関数では必要な行列を定義し、ソルバーに渡して二次計画問題を解く。その結果として得られる、各 α の値、 θ の値、サンプル点による識別器の識別率、識別器の関数を表示する。この関数の戻り値は識別器 f である。

3.3 plot_f(f, x, y,num=100)

この関数では識別器 f , サンプル点ベクトル集合 x , クラス集合 y , を実際にプロットする作業を行う。表示領域を $\text{num} * \text{num}$ 等分したグリッドスペースと考え、サンプル点とグリッドでのクラスを図示する。

3.4 load_x_y(fileName)

ファイルからサンプルデータを読み込む関数である。書式は実験ページのものに従う。

3.5 parse_argv()

コマンドライン引数をパースする関数である。

4 考察

実験ページに有る実際のサンプルデータ二種 (線形分離可能な集合とそうでない集合) に対して、100 パーセントの分離に成功した。サンプル数を 1 から 100 までに減らした集合の場合でもうまく動作することを確認した。ガウスカネル、多項式カーネルともに完璧に分離できたが、シグモイドカーネルはうまく動作しなかった。理由としては、係数が微妙なのだと考えられるので、うまく動作する係数を探したい。