パターン認識中間レポート 2

澤 祐里

（21–1–037–0801）

2021 年 7 月 6 日

目次

1　本論

1.1　課題 1

1.1.1　結果

以下の計測方法で得られた識別精度を表 1 から表 4 にそれぞれ示す。全ての計測方法において、それぞれのデータで最も精度が高かったものを赤く表示している。

- k 最近傍法（k=5）：kNN5
- 線形サポートベクトルマシン：linearSVM
- 非線形サポートベクトルマシン（RBF カーネル）：nonlinearSVM
- ランダムフォレスト：randomForest

表 1　knn5 の結果

data	平均精度
1	1.000
2	0.987
3	0.997
4	0.753
5	0.963

表 2　linearSVM の結果

data	平均精度
1	1.000
2	0.990
3	0.460
4	0.443
5	0.909

表 3　nonlinearSVM の結果

data	平均精度
1	1.000
2	0.990
3	0.997
4	0.800
5	0.450

表 4　randomForest の結果

data	平均精度
1	1.000
2	0.983
3	0.997
4	0.770
5	0.933

以下の、図 1 は data1 で線形サポートベクトルマシン、図 2 は data2 で線形サポートベクトルマシン、図 3 は data3 で k 再近傍法、図 4 は data4 で非線形サポートベクトルマシンを使った際のグラフである。

図 1　Data1_linearSVM

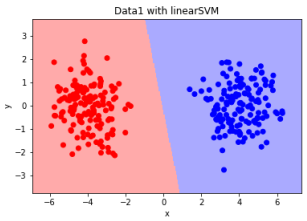


図 2　Data2_linearSVM

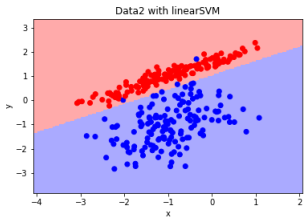


図 3　Data3_kNN5

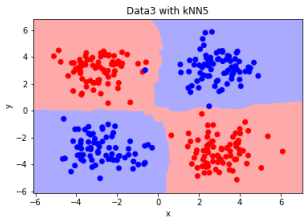
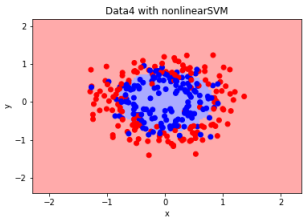


図 4　Data4_nonlinearSVM



1.1.2　考察

- そのデータに対して適していると思われる手法とその理由

data1　適していると思われる手法は、線形サポートベクトルマシンである。理由は、過学習を避けるために単純なモデルである線形サポートベクトルマシンから試していくと、線形サポートベクトルマシンの時点で精度が「1.000」となっており、十分な精度があるためである。

data2　適していると思われる手法は、線形サポートベクトルマシンである。理由は、data1 の時と同じように単純なモデルから試していくと、線形サポートベクトルマシンの時点で、精度が「0.990」となっており、十分な精度があるためである。

data3　適していると思われる手法は、k 再近傍法 (k=5) である。理由は、data1 の時と同じように単純なモデルから試していくと、線形サポートベクトルマシンでは精度が「0.460」となり、精度が悪い。そのため、次に単純なモデルである k 再近傍法 (k=5) を試すと、精度が「0.997」となり、十分な精度があるため、適していると思われる。

data4 適していると思われる手法は、非線形サポートベクトルマシンである。理由は、data1 の時と同じように単純なモデルから試していくと、非線形サポートベクトルマシンは、精度が「0.443」となり、精度が悪い。次に、単純なモデルである k 再近傍法 (k=5) の精度は、「0.753」となり、精度がやや悪い。次に、非線形サポートベクトルマシンとランダムフォレストの精度を見ると、それぞれ精度が「0.800」,「0.770」となり、非線形サポートベクトルマシンの精度が一番高くなるため、適していると思われる。

data5 適していると思われる手法は、k 再近傍法 (k=5) である。理由は、data1 の時と同じように単純なモデルから試していくと、線形サポートベクトルマシンの精度は「0.909」となり、精度は十分といえない。次に、単純なモデルである k 再近傍法 (k=5) の精度は、「0.963」となり、十分な精度があるため、適していると思われる。

- この結論が得られたのはデータにどのような性質があるからか

data1 図 1 を見てわかる通り、データが線形分離可能であるため、線形サポートベクトルマシンが適しているとなった。

data2 図 2 を見ると、data1 と同じくデータは線形分離可能であるため、線形サポートベクトルマシンが適しているとなった。

data3 図 3 を見ると、データは線形分離不可能であるため、線形サポートベクトルマシンの精度は悪い。しかし、同じクラスのデータ同士で固まっているため、次に単純なモデルである k 再近傍法 (k=5) が適しているとなった。

data4 図 4 を見ると、データは線形分離不可能であるため、線形サポートベクトルマシンの精度は悪い。そして、赤のクラスのデータに囲われるように青のクラスのデータが存在しており、それらの距離が近いため、k 再近傍法も精度は悪い。非線形サポートベクトルマシンの精度は比較的に高かったことから、データを多次元化すると、ソフトマージンでの線形分離が可能になったと考えられる。

data5 k 再近傍法 (k=5) の精度が高かったことから、一つのデータの近傍 5 個のデータで同じクラスのデータが多かったと考えられる。

1.2　課題 2

1.2.1　結果

以下の表 5 に、課題 1 でとりあげた 4 つの手法を Data5 に適用した時の、学習にかかった時間、テストにかかった時間、平均精度をそれぞれ示す。

表 5　それぞれの手法でのデータ 5 の結果

手法	学習にかかった時間	テストにかかった時間	平均精度
k 最近傍法 (k=5)	0.00796 [秒]	0.08403 [秒]	0.963
線形サポートベクトルマシン	0.12577 [秒]	0.00114 [秒]	0.909
非線形サポートベクトルマシン	0.46725 [秒]	0.05381 [秒]	0.450
ランダムフォレスト	0.17586 [秒]	0.00995 [秒]	0.933

1.2.2　考察

- 非線形サポートベクトルマシンは、他の手法に比べて、学習にかかった時間とテストにかかった時間の両方がともに長い傾向に見られる。これは、非線形サポートベクトルマシンが多次元のデータを扱っているために時間がかかっていると考えられる。
- 線形サポートベクトルマシンとランダムフォレストの学習にかかった時間は k 再近傍法 (k=5) に比べて、長くなる傾向が見られる。しかし、テストにかかった時間では、反対の結果となり k 再近傍法 (k=5) の方が長くなっている。これは、線形サポートベクトルマシンやランダムフォレストでは、モデルを適切に学習することが精度に繋がっているため、時間がかかるものの、k 再近傍法では、テスト時に近傍点を見つけるため、新しいデータと既存の全データとの距離を知る必要がある。そのため、k 再近傍法ではテストにかかる時間の方が長くなる。また、k 再近傍法では、上記の処理がメインであるため、準備や学習にはほとんど時間がかからない。

1.3　課題 3

1.3.1　課題 3-1

非線形サポートベクトルマシンのパラメータを以下の図 5 から図 7 のようにして実行したところ、以下の図 8 から図 9 のような結果が得られた。特にパラメータセット 2 とパラメータセット 3 は、課題 1 で得られた最良の精度を超える精度となった。

図 5　パラメータセット 1

```
def nonlinearSVM():
    # 非線形SVMによる学習（今回はデフォルトパラメータにする）
    # もしパラメータを変える場合はここの直下を変える
    # パラメータここから
    # パラメータC
    svm_C=5.0
    # 非線形カーネルの種類
    svm_kernel='sigmoid'
    # 非線形カーネルのパラメータ
    svm_gamma='scale'
    # svm_gamma を変えるときは ' auto' でなく何か数字にします.
    # パラメータここまで
```

図 6　パラメータセット 2

```
def nonlinearSVM():
    # 非線形SVMによる学習（今回はデフォルトパラメータにする）
    # もしパラメータを変える場合はここの直下を変える
    # パラメータここから
    # パラメータC
    svm_C=30.0
    # 非線形カーネルの種類
    svm_kernel='poly'
    # 非線形カーネルのパラメータ
    svm_gamma='scale'
    # svm_gamma を変えるときは ' auto' でなく何か数字にします.
    # パラメータここまで
```

図 7　パラメータセット 3

```
def nonlinearSVM():
    # 非線形SVMによる学習（今回はデフォルトパラメータにする）
    # もしパラメータを変える場合はここの直下を変える
    # パラメータここから
    # パラメータC
    svm_C=100.0
    # 非線形カーネルの種類
    svm_kernel='rbf'
    # 非線形カーネルのパラメータ
    svm_gamma='scale'
    # svm_gamma を変えるときは ' auto' でなく何か数字にします.
    # パラメータここまで
```

図 8　パラメータセット 1 の結果

```
method: nonlinearSVM

data: Data5

<< 2021年7月6日19時22分50秒に実行 >>
==================================
Data5を使ってnonlinearSVMの学習とテストを行っていきます.
==================================

データ数が多いので識別境界の図示は行いません
学習にかかった時間（交差確認の各試行の平均）：0.09489 [秒]
テストにかかった時間（交差確認の各試行の平均）：0.03609 [秒]

交差確認を行って求めた平均精度は 0.805 です.
```

図 9　パラメータセット 2 の結果

```
method: nonlinearSVM

data: Data5

<< 2021年7月6日19時24分35秒に実行 >>
==================================
Data5を使ってnonlinearSVMの学習とテストを行っていきます.
==================================

データ数が多いので識別境界の図示は行いません
学習にかかった時間（交差確認の各試行の平均）：0.04971 [秒]
テストにかかった時間（交差確認の各試行の平均）：0.01425 [秒]

交差確認を行って求めた平均精度は 0.969 です.
```

図 10　パラメータセット 3 の結果

```
method: nonlinearSVM

data: Data5

<< 2021年7月6日19時25分44秒に実行 >>
==================================
Data5を使ってnonlinearSVMの学習とテストを行っていきます.
==================================

データ数が多いので識別境界の図示は行いません
学習にかかった時間（交差確認の各試行の平均）：0.07933 [秒]
テストにかかった時間（交差確認の各試行の平均）：0.02728 [秒]

交差確認を行って求めた平均精度は 0.974 です.
```

パラメータと精度を表にすると、以下の表 6 のようになった。

表 6　それぞれのパラメータセットでの精度

パラメータセット	パラメータ C	非線形カーネルの種類	非線形カーネルのパラメータ	平均精度
1	5.0	sigmoid	scale	0.805
2	30.0	poly	scale	0.969
3	100.0	rbf	scale	0.974

1.3.2　課題 3-2

1.3.3　結果

決定木の最大深さ 1,5,10 にそれぞれ変えて実行したところ、表 7 のような結果となった。また、以下の図 11 から図 13 のような識別境界が得られた。

表 7　木の最大深さごとの精度

木の最大深さ	平均精度
1	0.540
5	0.733
10	0.670

図 11　木の最大深さ 1 の識別境界

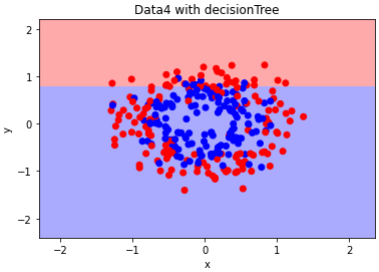


図 12　木の最大深さ 5 の識別境界

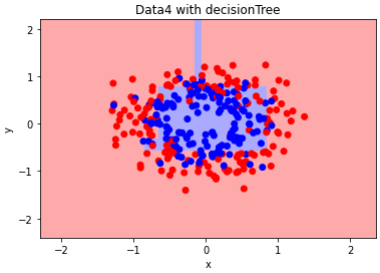
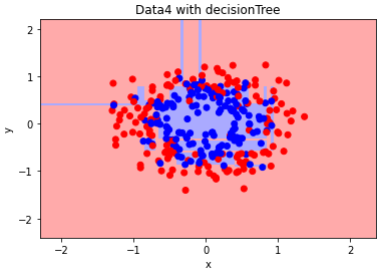


図 13　木の最大深さ 10 の識別境界



1.3.4　考察

- 木の最大深さが 1 の時に、図 11 のようになったのは、分岐が二つしかないため、y がある値 n 以上なら赤、それ以外なら青のように分けられたと考えられる。
- 木の最大深さが 5 から 10 になった時に、精度が下がってしまったのは、分岐が多くなったために、外れ値を条件にしてしまう「過学習」が起こったためだと考えられる。

1.3.5　課題 3-3

「ConvergenceWarning」は線形サポートベクトルマシンを使っている際に出力される。線形サポートベクトルマシンは 2 次元線形分離することで識別するが、その線形分離の線が 1 意に収束しない時に、「ConvergenceWarning」となる。