機械学習に入門した頃の回想録

2015年9月10日

データサイエンティスト養成読本
機械学習入門編　刊行イベント

@sfchaos

自己紹介

- 福島真太朗
- 所属：トヨタIT開発センター
- twitterID： @sfchaos
- 仕事：クルマ関係のデータマイニング

第II部特集1「機械学習ソフトウェアの概観」を担当させていただきました

機械学習と私

2000　2006　2009　2011　2015

大学・大学院

社会人

物理・応用数学
(非線形力学系・カオス)

金融工学
(確率/偏微分方程式)

データマイニング・機械学習

・サービスの入会，解約予測
・機械の故障予測
・
・クルマのデータマイニング

知識表現
(オントロジー)

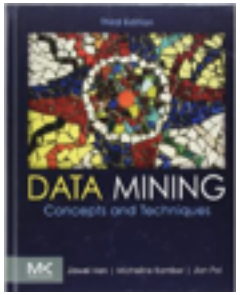
…

勉強手始めに心がけたこと

- Data-Driven学習：理論と実践を両輪で回す

理論

「朱鷺の杜」で調べて,
リンク先の論文で深堀り



実践

実データで遭遇した問題例

- 不均衡データ
  正例が少なく，負例が圧倒的に多いデータ
- サービスの解約ユーザが0人と予測

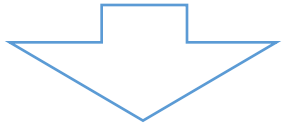
33	1	4	2	8	0	6	0	3	5	0 ・・・
6	1	3	2	2	0	5	0	4	5	0 ・・・
39	1	3	3	9	1	4	2	3	5	0 ・・・
9	1	2	3	3	2	3	2	4	5	0 ・・・
31	1	2	4	7	0	2	0	7	9	0 ・・・
30	1	2	4	7	1	4	2	3	5	0 ・・・

訓練データの読み込み・加工

```
import pandas as pd  # 0.16.2
# The Insurance Company Benchmark データの読み込み
## 訓練データ
dtype_train = {c: object if c in range(64) else float for c in range(86)}
tic_train = pd.read_csv('ticdata2000.txt', header=None, delimiter='\t',dtype=dtype_train)
### 数値の正規化
cols_num = range(65, 85)
tic_train_num = tic_train[cols_num]
tic_train[cols_num] = (tic_train_num - tic_train_num.mean())/tic_train_num.std()
### 訓練データの説明変数，目的変数
X_train, y_train = tic_train.drop(85, 1).pipe(pd.get_dummies), tic_train[85]
```

学習と予測

```
from sklearn import svm
from sklearn import metrics
# サポートベクタマシンによる学習
svc = svm.SVC()
svc.fit(X_train, y_train)
# テストデータに対する予測
pred = svc.predict(X_test)
# 予測精度の評価
print(metrics.classification_report(tic_test_label, pred, target_names=['0', '1']))
```



	precision	recall	f1-score	support
0	0.94	1.00	0.97	3762
1	0.00	0.00	0.00	238
avg / total	0.88	0.94	0.91	4000

すべて負例と予測!!

不均衡データの調査

- 朱鷺の杜「不均衡データ」

不均衡データ

Top / 不均衡データ

[トップ] [編集 | 凍結 | 差分 | バックアップ | 添付 | リロード] [新規 | 一覧 | 単語検索 | 最終更新 | ヘルプ]

朱鷺の杜Wiki

- 機械学習
- 学会・研究会
- 研究者
- 書籍
- 論文
- 特集・解説記事
- 論文英語
- ブログ
- フリーウェア
  - python
  - nuffgy
- テストデータ
- 日本語版PRML本
- カーネル多変量解析
- 人気ページ
- 今日の人気ページ
- ページ一覧

参加しよう

- Wiki超入門
- 練習用ページ
- 数式の表示
- こびとさん

編集用ID: ibis
パスワード: 「VC次元」のVのフルスペルで最初だけ大文字

最新の30件

2015-09-08
- People

2015-09-03
- English

2015-08-25
- Blog

これらのキーワードがハイライトされています：不均衡データ

不均衡データ (imbalanced data)

識別問題において，各クラスのデータが生じる確率に大きな差がある場合．例えば，二値識別問題で正例が 1% で，負例が 99% といった状況．はずれ値検出を識別問題として解く場合などが該当する．こうしたデータについては，予測精度が著しく低下する場合があることが知られている．

文献1は，人工データに対してニューラルネット系の手法と適用して実験．不均衡データに対する対策は次の三種類

1. 少ない方のクラスをオーバーサンプリングしてもう一方のクラスの大きさに合わせる
2. 大きい方のクラスをサブサンプリングしてもう一方のクラスの大きさに合わせる
3. 一方のクラスを無視して，もう一方のクラスをカバーするような規則を獲得

* 各クラスごとに異なる損失を考えるコストを考慮した学習も 1 や 2 と同様の対策とみなせる

実験的に次のような結果を報告している

- 線形分離できる単純な問題では不均衡データの問題は生じないが，各クラスが複数の部分クラスで構成される場合には問題を生じる
- データ全体の量が増えても不均衡データの問題は解消できない
- サンプリングを使う二つの方法は，複雑なクラスに対して有効．また，データ数が多いときはサブサンプリングする方が良い．
- 一方のクラスを認識する場合には，多数派クラスの方を認識すべき

文献2は，各クラスが部分クラスで構成されているときに不均衡データ問題が生じる原因を実験的に調査．

小さなクラスを部分クラスに分けることで，各部分クラスの事例数が極端に減ることが問題としている．クラスごとではなく，各部分クラスをクラスタリングなどで見つけて識別するといった対策について論じている．

-- しましま

関連項目

- imbalanced data

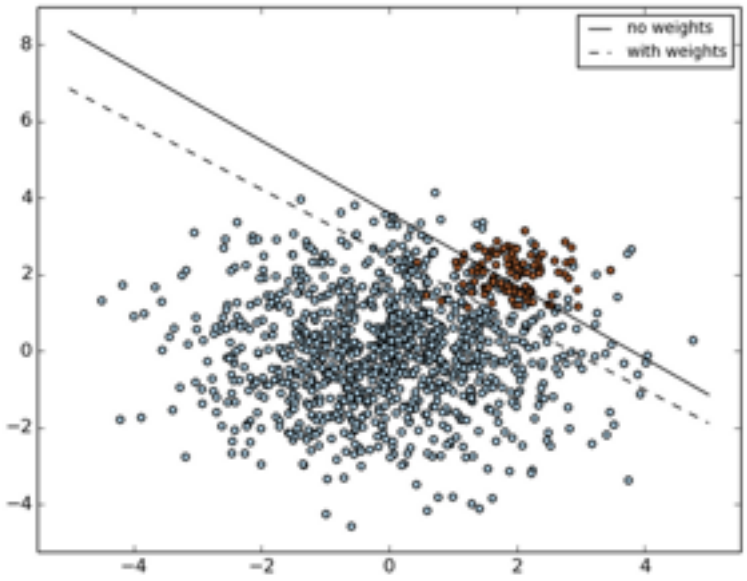
scikit-learnの機能の調査

- scikit-learn “SVM: Unbalanced problems”

1.4.1.3. Unbalanced problems

In problems where it is desired to give more importance to certain classes or certain individual samples keywords class_weight and sample_weight can be used.

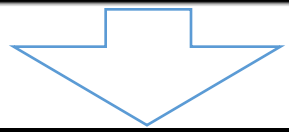
SVC (but not NuSVC) implement a keyword class_weight in the fit method. It's a dictionary of the form {class_label : value}, where value is a floating point number > 0 that sets the parameter C of class class_label to C * value.



不均衡データの調整

- クラスウェイトの調整

```
from sklearn import svm
from sklearn import metrics
# サポートベクタマシンによる学習
svc_imb = svm.SVC(class_weight={1:10, 0:1})
svc_imb.fit(X_train, y_train)
# テストデータに対する予測
pred = svc_imb.predict(X_test)
# 予測精度の評価
print(metrics.classification_report(tic_test_label, pred, target_names=['0', '1']))
```



	precision	recall	f1-score	support
0	0.94	0.99	0.96	3762
1	0.14	0.03	0.05	238
avg / total	0.89	0.93	0.91	4000

若干改善!!