Python ではじめる機械学習

4.5 自動特徴量選択

15T4032G 芝山直希

2018 年 6 月 8 日

自動特徴量選択？

わかりやすい (予測が早い) モデルを作りたいけど、どの特徴量を捨てればいいかわからない…
どんな特徴量を追加 (削減) すれば複雑な (汎化性能の高い) モデルができるの？よくわからない…
自動選択する 3 つの基本的な (教師あり学習向け) 手法：

- 単変量統計
- モデルベース選択
- 反復選択

実世界データでは、特徴量選択での性能の大幅な向上はあまりないが、無意味ではない

単変量統計

統計的にターゲットと高い関連性を持つ特徴量を選択する手法

- 個々の特徴量を個別に考慮するため、交互作用を認識できない
- モデルを使用せずに特徴量選択を行う手法であり、計算が高速
- scikit-learn では単変量特徴量選択用のクラスが複数用意され、p-値の閾値であるスレッショルドの計算方式等が異なる
  - それぞれのクラスには、クラス分類用の f_classif テスト (デフォルト)、回帰用の f_regression テストが用意されており、p-値 (ターゲットとの関連性のなさを表す) の計算方法が異なる
  - 関数１：SelectKBest…残す特徴量数を指定する
  - 関数２：SelectPercentile…全特徴量の何割を残すか指定する

特徴量が多すぎてモデルを作れない、多くの特徴量がターゲットと関連がないと思われる場合などで有効

モデルベース特徴量選択

教師あり学習モデルを用いて各特徴量の重要性を判断し、重要なものを残す手法

- 最終的に使用する教師あり学習モデルと異なるモデルでも使用できる
- 各特徴量の重要性を出力できるモデルを重要度判断に用いる
  - 決定木 (をベースとする) モデル、線形モデルなど
- すべての特徴量を同時に考慮する手法であり、交互作用を認識できる
- scikit-learn では SelectFromModel クラスを用いる
  - 主要引数：教師あり学習モデル、スレッショルドの方式
- 使用するモデルに依存するが、単変量選択手法よりも強力

反復特徴量選択

使用する特徴量が異なる複数のモデルを用いて選ぶ手法

- 使う特徴量を減らしていく方針と、増やしていく方針がある
- このタイプの手法の一つに、再帰的特徴量削減 (recursive feature elimination:RFE) がある
  - 特徴量をすべて使うモデルを学習し、「最も重要でない特徴量を削除して再学習する」工程を特徴量が事前に設定した数になるまで繰り返す
  - 各特徴量の重要性を出力できるモデルが必要
  - scikit-learn には RFE クラスが用意されている 主要引数：モデル、残す特徴量数
- 選択性能は高めだが、複数のモデルを学習するため計算量と時間がかかる