Chap 8. 特徴量の重要度

Financial Machine Learning

Hirata Shinya

2023 年 7 月 23 日

Outline

8.1 はじめに

よくある誤り

データの一部で予測してバックテストをするのを都合のいい結果が出るまで繰り返すこと
科学的不正 (scientific fraud) と呼ばれる

同じデータに対して何度もテストを繰り返す行為こそが偽の発見へとつながる
20 回ぐらい繰り返せば 5 % 有意水準で有意になる結果が見つかる
American Statistical Association は倫理ガイドラインで警告している
そんなことをするよりも特徴量の重要度によって投資戦略を探すほうがいい！

Outline

8.2 特徴量重要度の重要性

特徴量重要度はバックテストを実行する前に行うべき分析のひとつ

分類器によって学習し、汎化誤差を評価した後、
どの特徴量がそのパフォーマンスに寄与したか？(= 特徴量重要度)
を分析する

重要度を知れば、

- 分類器の予測力を上げるシグナルを強める特徴量を追加することができるかも
- システムにノイズを追加するだけの特徴量を削除できるかも

8.2 特徴量重要度の重要性

どの情報源が分類器が見つけたパターンにとって不可欠かが変わればそのパターンに対する識見を得ることができる
どの特徴量が重要かわかったら数々の実験を行うことでさらに学習を進めることができる

- これらの特徴量は常に重要なのか、それとも特定の環境下においてのみ重要なのか
- 何が重要性の変化を引き起こすのか、
- これらのレジームスイッチは予測可能なのか、
- 他の関連する金融商品にも関係があるのか
- 他の資産クラスにも関係をもつのか。
- すべての金融商品で最も有効な特徴量はなんだろうか
- 投資対象全体で最高の相関関係をもつ特徴量の部分集合はなんだろうか。

マルコスのバックテスト第 1 法則

マルコスのバックテスト第 1 法則

バックテストはリサーチツールではない。特徴量の重要度こそがリサーチツールなのである。-Marcos Lopez de Prado(著者)

Outline

8.3 代替効果の影響を受ける特徴量重要度

代替効果 (substitution effect) による影響を受けるかに基づいて特徴量重要度を区分する

代替効果とは、ある特徴量の推定重要度が他の関連する特徴量の存在によって削減されるときに生じる効果
統計学や計量経済学では「多重共線性 (multi collinearity)」と呼ばれるものの機械学習分野における類似物

代替効果の影響を受ける重要度

- 平均不純度減少量 (MDI)
- 平均正解率減少量 (MDA)

代替効果の影響を受けない重要度 (8.4 節)

- 単一特徴量重要度 (SFI)

(8.6) 人工データによる実験

特徴量重要度手法が人工データにどのように機能するかをテストする

3 種類の特徴量から構成されるデータセット (X,y) を生成する

1. 有益な特徴量 (Informative): ラベルの決定に用いられる特徴量
2. 冗長な特徴量 (Redundant)：有益な特徴量をランダムに線形結合した特徴量。代替効果を引き起こす
3. ノイズ (Noise)：ラベルの決定に無関係な特徴量

スニペット 8.7 は 10000 個の観測値に対して、計 40 個の特徴量からなる合成データを作成している

1. 10 個の有益な特徴量
2. 10 個の冗長な特徴量
3. 20 個のノイズ

スニペット 8.8 では任意の手法による特徴量重要度計算の呼び出しを行っている
デフォルトの分類器としてバギング決定木 (6 章) を使用している

スニペット 8.9 はデータ生成、特徴量重要度分析、出力の収集と処理までの全要素を呼び出すためのメイン関数

スニペット 8.10 は特徴量重要度プロット関数

8.3.1. 平均不純度減少量 (Mean Decrease Impurity)

MDI は、RF のようなツリーベース分類器特有の、高速な、インサンプル（IS）で説明上での重要度（explanatory-importance）を測る方法である。

まず、決定木が分類問題を解くときに使われる手法である、
不純度について説明する

不純度

決定木の不純度とは、決定木がどれだけうまく分割できているかの指標

- **ジニ不純度**
- 情報エントロピー
- 誤り率

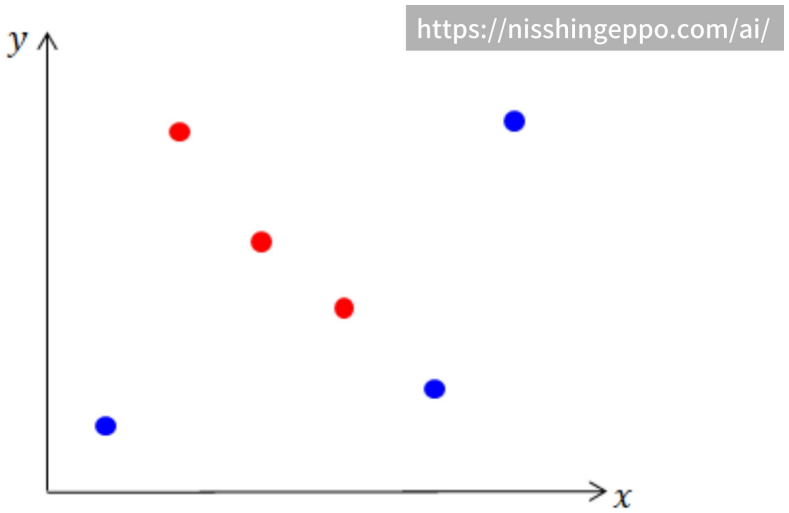
ジニ不純度の説明は以下のサイトを参考に作成
https://nisshingeppo.com/ai/decision-tree/

ジニ不純度

ジニ不純度 G

$$G = 1 - \sum_{t=1}^{c} p_i^2$$

c: ラベル数, p: あるラベルの数をデータ数で割ったもの（割合）



分割前：

$$G = 1 - \{(\frac{3}{6})^2 + (\frac{3}{6})^2\} = 0.5$$

分割した場合は各領域ごとに計算して加重平均を取る

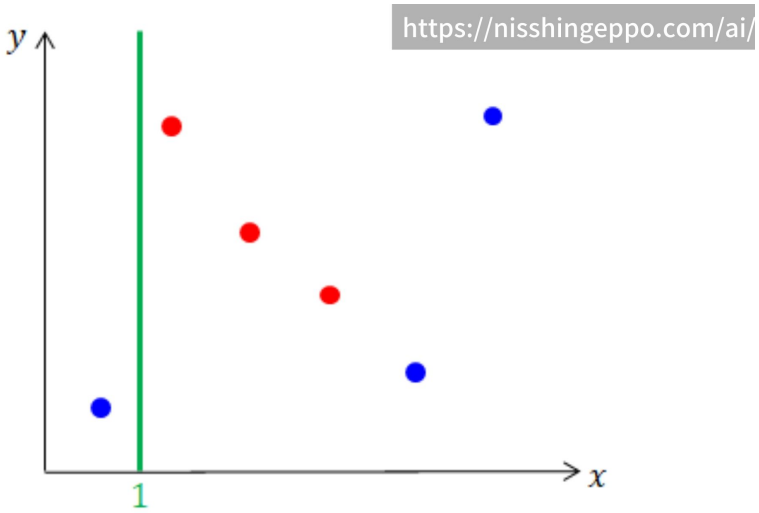


図: x=1 で分割

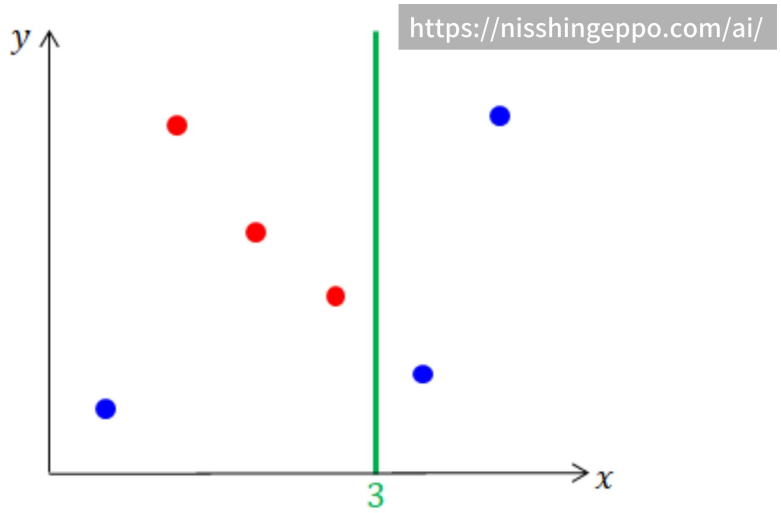
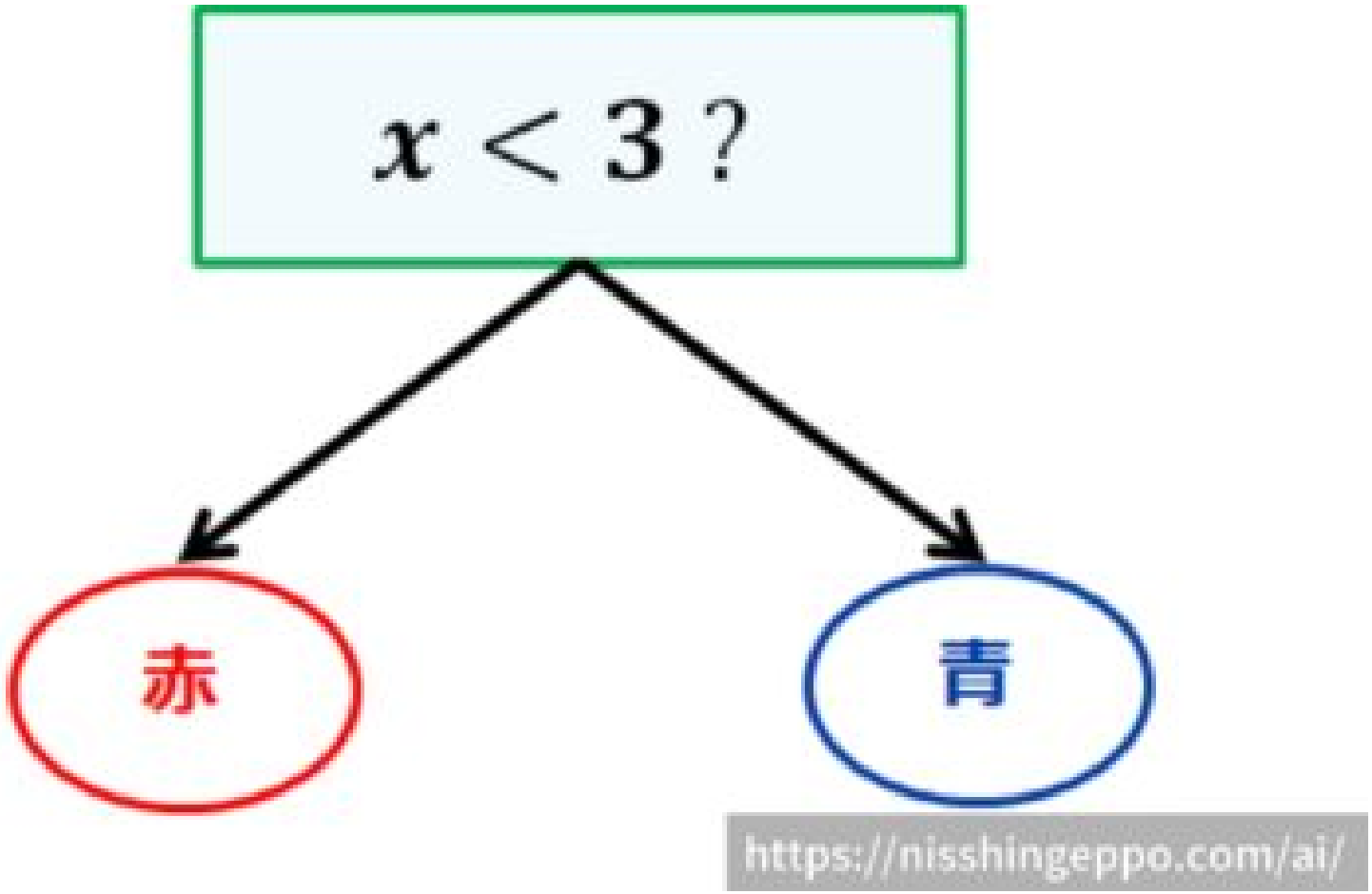


図: x=3 で分割

$$G_{x=1} = 1/6\{1-(1/1)^2\} + 5/6\{1-\{(3/5)^2 + (2/5)^2\}\} = 0.4 \quad (1)$$

$$G_{x=3} = 4/6\{1-\{(3/4)^2 + (1/4)^2\}\} + 2/6\{1-(2/2)^2\} = 0.25 \quad (2)$$

不純度が一番小さくなるように決定木が決まる



MDI

MDI の方法

1. 各決定木の各ノードにおいて、選択された特徴量は**不純度**が削減されるように入力された部分集合を分割する

2. 決定木ごとに不純度減少量全体のうち各特徴量により削減された量を導き出すことができる

3. そして決定木集合に対し、すべての木に対して削減された減少量を平均し、それに応じて特徴量をランク付けすることができる =MDI

MDI の留意事項

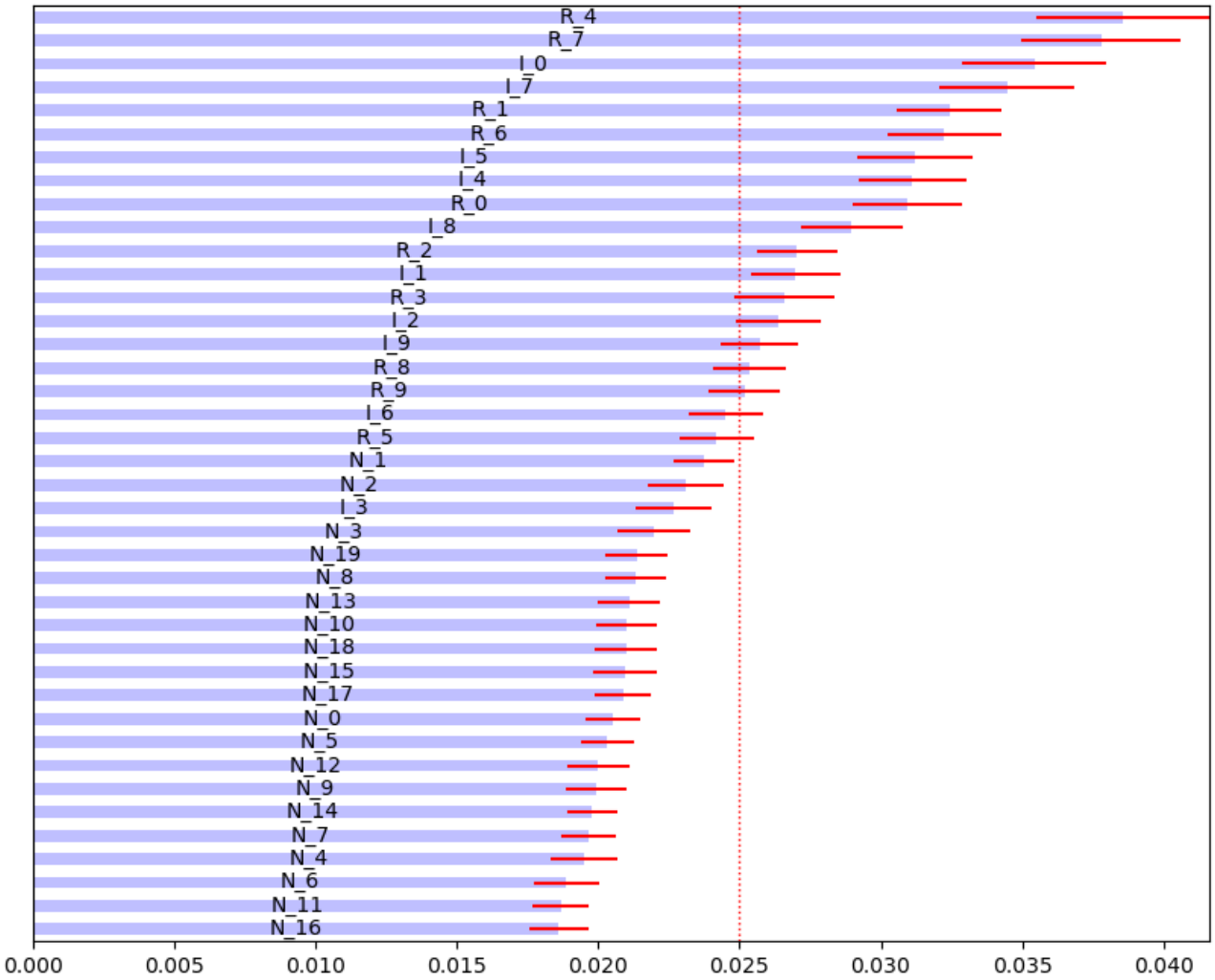
1. マスキング効果 (Masking effects)
   - 分類木がある特徴量をシステマティックに無視し、ほかの特徴量を重視すること
   - 対策：max_features=int(1) に設定することで、階層ごとにランダムなひとつの特徴量のみが選択される
     - ★ 全特徴量はランダムに不純度を削減する機会が与えられる
     - ★ 重要度 0 の特徴量が平均に含まれないことを確認すること
     - ★ 重要度 0 となるのはその特徴量が選択されなかった場合なので、np.nan に置き換える
2. インサンプルな手法なので、予測力が全くなくてもすべての特徴量がある程度の重要度を持つ
3. MDI はツリーベースの分類木にしか使えない

4. 特徴量重要度の合計が 1 になり、各重要度は 0 ～ 1 の間の値になる
5. 相関を持つ特徴量が存在する際の代替効果を考慮しない
   - 代替効果のある特徴量の重要度を薄めるので、2 つの同一の特徴量がある場合は同じ確率でランダムに選ばれるので、重要度が半分になる
6. 不純度関数は多くのカテゴリを持つ予測変数を不公平に重視するので、いくつかの予測変数に偏る

RandomForest ではデフォルトの特徴量重要度スコアとして MDI を利用している。計算が早いからと思われる。

スニペット 8.2 MDI 特徴量重要性

MDI の結果


R_4
R_7
I_0
I_7
R_1
R_6
I_5
I_4
R_0
I_8
R_2
I_1
R_3
I_2
I_9
R_8
R_9
I_6
R_5
N_1
N_2
I_3
N_3
N_19
N_8
N_13
N_10
N_18
N_15
N_17
N_0
N_5
N_12
N_9
N_14
N_7
N_4
N_6
N_11
N_16
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0.040

横方向のバー：各特徴量に対する全決定木の平均 MDI 値
バーの先端の赤線：標準偏差
MDI による重要度の合計は 1 になるため、仮にすべての特徴量重要度が同じである場合、各重要度の値は 1/40 になる。
赤の縦線：その閾値 1/40 を示している

教科書では、R_5 以外のすべての Informative と Redundant が 1/40 を上回っていたが、
n_estimator=100、n_samples=1000 にして実行した結果では I_3 と R_5 が 1/40 を下回っていた。

多少は結果が違うものの、Noise はすべて 1/40 を下回っていたので、MDI はうまく機能していたと言える。

8.3.2 平均正解率減少量 (Mean Decrease Accuracy)

MDA はアウトオブサンプル (OOS) での予測における重要度 (predictive-importance) を計る低速な方法

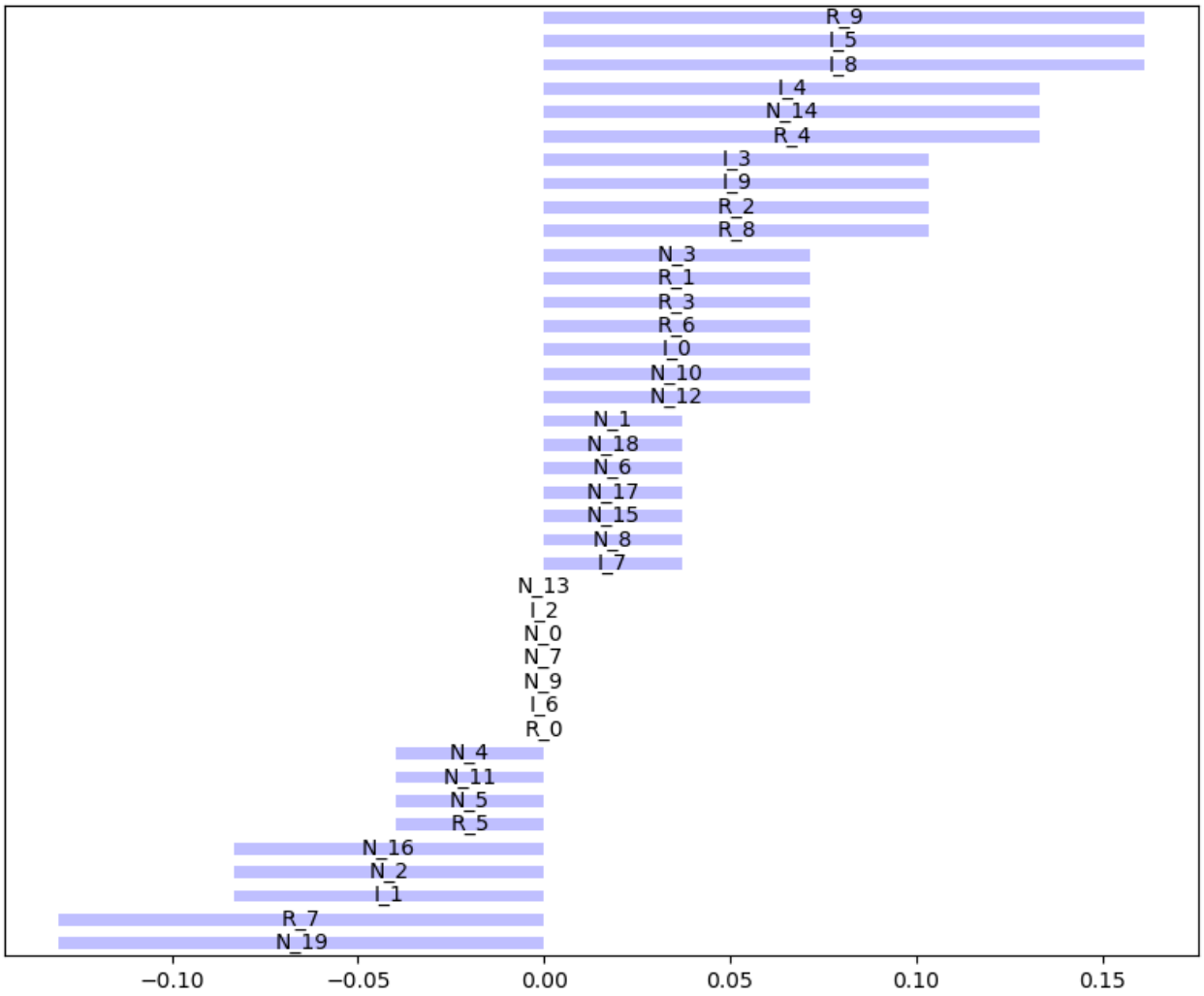
MDA の方法

1. まず、分類器を学習させる
2. 何らかのパフォーマンススコア（正解率や負の対数損失など）により OOS でのパフォーマンスを算出する
3. そして、特徴量行列（X）を一度に 1 列ランダムに並び替え、並べ替え後のアウトオブサンプルでのパフォーマンスを算出する
4. MDA はその列の並べ替えによるパフォーマンス低下量の関数として測定する

MDA の留意事項

1. 分類木だけでなく、すべての分類器に適用できる
2. MDA におけるにおけるパフォーマンススコアは正解率に限定されない。
   - たとえば F1 などが使われることもある
   - MDA よりも並べ替え重要度 (permutation importance) という名前のほうが適切かも？
   - スコアリング関数が距離空間の性質を満たしていない場合は MDA の結果はランキングとして使用される (相対的に評価する)
3. MDI と同様、相関を持つ特徴量が存在すると代替効果の影響を受けやすい
   - 2 つの同一の特徴量がある場合、MDA は常に一方の特徴量が他方に対して冗長であるとみなす。
4. アウトオブサンプルなので、すべての特徴量が重要ではないと判断する可能性がある
5. CV に対してパージとエンバーゴを適用しないといけない

MDA の結果


R_9
I_5
I_8
I_4
N_14
R_4
I_3
I_9
R_2
R_8
N_3
R_1
R_3
R_6
I_0
N_10
N_12
N_1
N_18
N_6
N_17
N_15
N_8
I_7
N_13
I_2
N_0
N_7
N_9
I_6
R_0
N_4
N_11
N_5
R_5
N_16
N_2
I_1
R_7
N_19
−0.10
−0.05
0.00
0.05
0.10
0.15

ほとんどの Informative と Redundant の MDA が Noise に比べて高いことがわかる
一部 R_7 や I_1 など低いランクになっているものもあるが、おそらく代替効果によるものだと考えられる

標準偏差はこの図には乗せることができなかったが、図 8-3 を見ると、標準偏差がやや高くなっている。これは MDA のネガティブな特徴である。しかし、パージつき k 分割 CV における分割数を増加させることによって対処できる

Outline

8.4 代替効果を除いた特徴量重要度

MDI や MDA では実際には重要な特徴量であっても、代替効果により冗長であると判断され破棄される可能性がある

予測の時にはそんなに問題にはならないが、モデルを理解、改善、単純化しようとしているときに誤った結論へ導かれる可能性がある

2 つの解決策

- **単一特徴量重要度 (Single Feature Importance)** によって MDI と MDA を補完する
- MDI と MDA を適用する前に特徴量を直交化する

8.4.1 単一特徴量重要度 (Single Feature Importance)

SFI は、クロスセクションで、OOS での予測における重要度 (predictive importance) を測る手法

各特徴量単体の OOS におけるパフォーマンススコアを計算する

SFI の留意事項

1. ベースツリー分類器だけでなく、あらゆる分類器に適用できる
2. パフォーマンススコアとしては正確度以外も使える
3. MDI や MDA とは異なり、代替効果は発生しない
4. MDA と同様に、OOS な CV によって評価されるため、すべての特徴量が重要でないと判断されることがある

SFI の主な欠点は、2 つの特徴量をもつ分類器が、2 つの単一特徴量による分類器のバギングよりも優れたパフォーマンスを発揮しうることを捉えられないことである。

たとえば以下の 2 つの効果が失われる

1. **結合効果 (joint effects)**: 特徴量 B は特徴量 A との組み合わせにおいてのみ有用かもしれない
2. **階層的重要度 (hierarchical importance)**: 特徴量 B だけでは不正確な場合でも、特徴量 B は特徴量 A による分割効果を説明するのに役立つかもしれない。

代替案として、特徴量の部分集合から OOS のパフォーマンススコアを計算することだが、特徴量が増えるに連れてその計算は複雑になる

スニペット 8.4:SFI の実装

SFI の結果

図 8-4 によると SFI もまともに機能している

しかし、おそらく結合効果によっていくつかの重要な特徴量がノイズより低いランクになっている。
ラベルは特徴量の組み合わせによる関数であり、それらを別々に予測しようとすると結合効果が見逃される。

それでも、SFI は代替効果の影響を受けないため、MDI と MDA を補完するものとして有用である。

8.4.2 特徴量の直交化

PCA のような直交化手法によってすべての代替効果が消える訳では無いが、少なくとも線形代替効果の影響は軽減されるはず。

PCA の方法: **特徴量を標準化して、分散共分散行列の固有ベクトルをとる**

観測値 $t = 1, ..., T$ と変数 $n = 1, ..., N$ をもつ、定常な特徴量行列 $\{X_{t,n}\}$ を考える

まず、$Z_{t,n} = \sigma_n^{-1}(X_{t,n} - \mu_n)$ により**標準化**特徴量行列 Z を計算する
μ_n : $\{X_{t,n}\}_{t=1,...,T}$ の平均、　　σ_n : $\{X_{t,n}\}_{t=1,...,T}$ の標準偏差

標準化

データを平均 0、標準偏差 1 になるように調整する

次に、$Z'ZW = W\Lambda$ を満たす固有値行列 Λ と固有ベクトル行列 W を計算する

$Z'Z$: Z の分散共分散行列 $(N \times N)$
Λ：降順にソートした固有値を対角成分に持つ $N \times N$ の対角行列
W：固有ベクトルを並べた $N \times N$ の正規直交行列
直交行列：$W' = W^{-1}$
正規行列：$W'W = WW'$、対角化可能

直交化された特徴量は $P = ZW$ として求められる
$P'P = W'Z'ZW = W'W\Lambda W'W = \Lambda$
となるので、特徴量が直交化されていることがわかる

対角化を X ではなく Z(標準化特徴量行列) を用いて行う理由

1. データを中心化することにより、第一主成分が観測値の主方向へ正しく方向づけられる
   - 線形回帰で切片を追加するのと同じような意味
2. データを再スケーリングすることにより、PCA は分散よりも相関の説明に焦点を当てることになる
   - 再スケーリングをしないと、第一主成分は X のなかで最大の分散を持つ列によって左右されてしまい、変数間の構造や関係について学習できなくなる

直交特徴量を使うメリット

直交特徴量を使うことには、代替効果への対処に加えて、2 つの利点がある

1. 直交化により、小さな値の固有値に関する特徴量を落とし、特徴量行列 X の次元を減らすことができる
   - 機械学習アルゴリズムの収束を早める
2. **データ構造を説明するために設計された特徴量に対して分析を行える**

2 つ目が特に重要
本書を通して懸念事項となっているのは、**オーバーフィッティング**のリスクであり、その可能性を検証できる

直交特徴量によるオーバーフィッティングのリスクの検証

偽りの重要度: 機械学習は統計的なまぐれで何らかのパターンを見つけることがある

PCA 分析は、ラベル情報なしに（教師なし学習)、ある特徴量が他の特徴量よりも「主要」であると判断する。
つまり、PCA は分類という観点でいっさいオーバーフィッティングすることなく、特徴量をランク付けしている。

PCA が主成分として選択したものと同じ特徴量が、MDI、MDA、または SFI 分析で最も重要なものとして（ラベル情報を用いて）選択されたとき、
これは機械学習によって識別されたパターンが**全くオーバフィッティングしていない**という確証になる

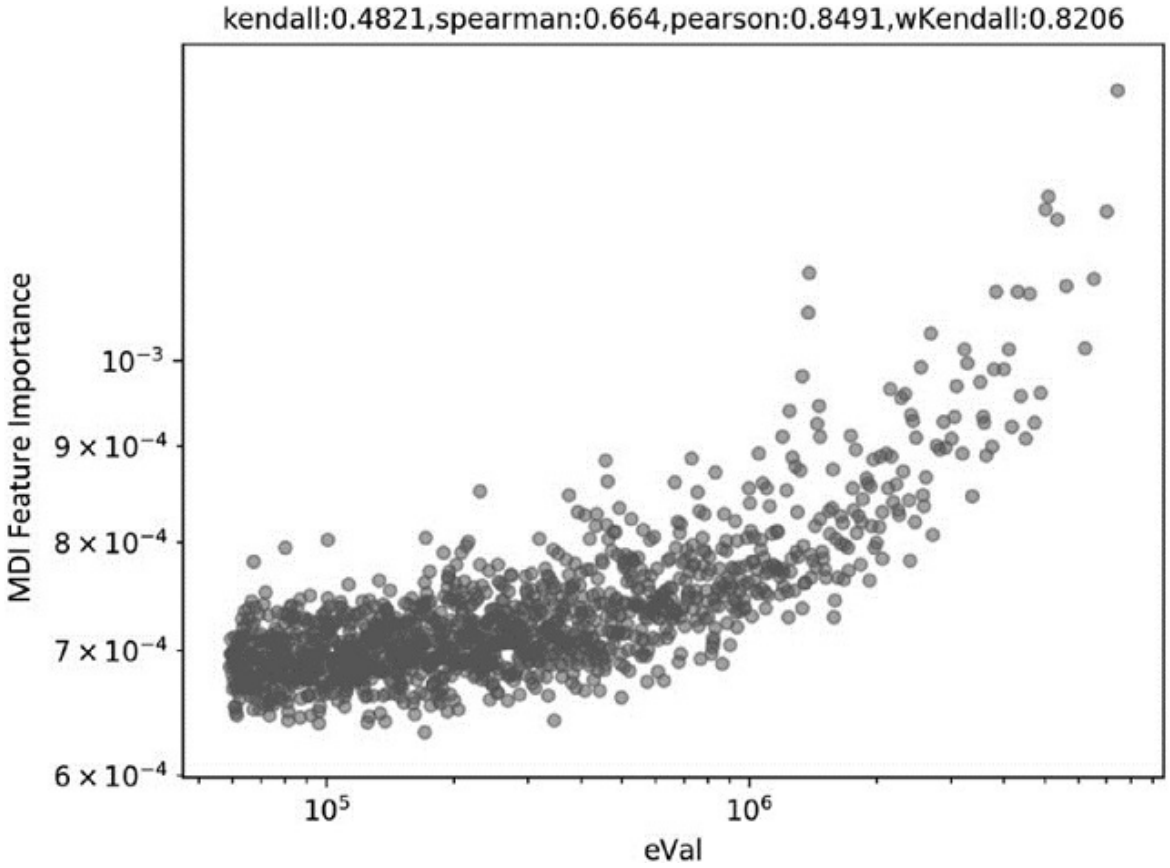


図: 8.1
y 軸：ある固有ベクトルに関する特徴量の MDI
x 軸：固有ベクトルに関する固有値、つまり直交化特徴量

図 8.1 では PCA が有益な特徴を識別し、オーバーフィッティングすることなく正しくランク付けしたことが証明されている

特徴量重要度とその特徴量に関連する固有値（逆 PCA ランクとも言う）の間の**ケンドールの加重タウ (順位相関係数)** が指標になる
この値が 1 に近いほど、PCA ランキングと特徴量重要度ランキングとの間の一貫性が強いことを示す

図 8-1 のケンドールの双曲 (hyperbolic) 加重タウは 0.8206 である

ケンドールの多重タウ

標準的なケンドールのタウよりもケンドールの加重タウを用いるべき理由の一つは、最も重要度の高い特徴量間でのランクが一致することを優先したいためである
重要度の低い（＝おそらくノイズの多い）特徴量の間のランクの一致性についてはあまり気にしない。

スニペット 8.6：Scipy でケンドールの加重タウを計算する方法

Outline

8.5 特徴量重要度の並列計算 VS スタック計算

特徴量重要度のリサーチアプローチは少なくとも 2 種類存在する

1. 並列計算
2. 特徴量スタッキング

並列計算

投資ユニバース $i = 1, ..., I$ 内の各証券 i に対して、データセット (X_i, y_i) を作成し、並列的に特徴量重要度を導き出す手法

証券 i における特徴量 j の基準 k による重要度を $\lambda_{i,j,k}$ で表す。
投資ユニバース全体での結果を集約することにより、特徴量 j の基準 k による結合重要度 $\Lambda_{j,k}$ を導き出すことができる。

様々な証券にわたって高い重要度となる特徴量は、裏付けとなる事象が存在する可能性が高いと考えられる。
特にこれらの特徴量重要度が各基準感において高い順位相関を示す場合はその可能性がさらに高い
これらの特徴量が予測力を持つ理論的メカニズムを詳しく調べると良いかもしれない

並列計算の利点、欠点

利点

- 計算を並列化できるため高速

欠点

- 代替効果により、重要である特徴量のランクが証券感で交換され、推定した $\lambda_{i,j,k}$ の分散が大きくなる可能性がある
  - 十分な数の投資ユニバースで $\lambda_{i,j,k}$ を平均すれば、この欠点は比較的小さくなる

特徴量スタッキング

すべてのデータセット $\{\tilde{X}_i, y_i\}_{i=1,\ldots I}$ を単一の結合データセット (X, y) にスタックする方法
$\tilde{X}_i$ は X_i の変換（例えば、移動ウィンドウにおける標準化）

この変換の目的は、ある程度の分布の均一性 $\tilde{X}_i \sim X$ を保証すること

投資対象全体が実際には単一の証券であるかのように、
分類器はどの特徴量が高いかをすべての証券において同時に学習することになる

特徴量スタッキングの利点、欠点

利点

1. 分類器は並列化アプローチよりもはるかに大きいデータセットに対して推定を行える
2. 重要度が直接導出されるため、結果の結合のウェイトづけの仕組みを必要としない
3. 結果は一般化されており、外れ値やオーバーフィッティングによるバイアスが少ない
4. 重要度スコアは証券間で平均化されていないため、代替効果によってこれらのスコアが弱まることがない

欠点

1. 大量のメモリとリソースを消費する可能性がある
   - HPC 技術が役立つ（20 22 章）

特徴量の並列計算 VS 特徴量スタッキング

筆者は、分類器が分析対象の集合において推定を行える場合はいつでも特徴量スタッキングを好んでいる
推定器が特定の証券や小さなデータセットに対してオーバーフィッティングを起こす可能性が低くなるから