2A3-04

Subset-Relief法によるデータマイニングのための属性選択手法

Attribute selection by Subset-Relief method for data mining

三浦輝久[*1]
Teruhisa Miura

[*1](財) 電力中央研究所
Central Research Institute of Electric Power Industory

Attribute selection methods are classified into wrapper methods and filter methods. Generally, wrapper methods are more accurate but have a higher computational complexity than filter methods. It is difficult to directly apply wrapper methods to data mining applications, which has a lot of attributes, because of the considerable computational complexity. In this paper, we propose a context-sensitive filter, which is modified the Relief algorithm and has a low complexity of computation. We propose the approach that selects a good subset of many attributes using the proposed filter, and then applies a wrapper method to its subset. In preliminary experiments, we show that the proposed filter can select effective attributes depending on context. By the proposed approach, we can apply a wrapper method to data mining applications and select effective attributes. We apply the proposed method to large-scale problem(the number of attributes is 500) and verify the effectiveness of our method.

1. はじめに

機械学習やデータマイニングにおいて，属性選択は重要なタスクである．現実の問題において, データセットの属性数はかなり多いが, その中には冗長なものや，無関係なものが多く含まれている．このような冗長／無関係属性は, 機械学習において，精度を落す結果となる．したがって，全体の属性から，重要な属性のみを絞り込む必要がある．これは属性選択 (feature selection) と呼ばれる．属性選択法としては, 属性選択に学習器を用いないフィルター法と学習器を用いるラッパー法がある [Kohavi 97, Das 01]．．一般にラッパー法の方が精度が良いが, データマイニングにおける属性選択では, 属性数が多いため, 計算量が大きくなり, 直接適用するのは困難である．ラッパー法による属性選択を探索問題としてみた場合, 子節点の生成が高い計算量を持つ探索問題としてみなすことができる．

本論文では, ラッパー法により生成する子節点の数を少なくするために, 良い精度の節点を残すことができるフィルターを提案する．この提案フィルターにより, 候補節点を篩にかけ, 篩に残った子節点だけをラッパー法により生成するアプローチを述べる．篩に用いるフィルターは, 現在の文脈を利用し, 効果的に, 高い精度の節点だけを残すことができる．

本論文では，まず従来の 2 つの属性選択法, フィルター法とラッパー法について説明する．次に計算量が軽いオペレータとして，Subset-Relief 法を提案し，次に, いくつかのベンチマーク問題に対する結果を論じ, 最後に大規模問題 (属性数 500) に対しての適用結果を述べる．

2. 先行研究

属性選択の場合, データセットの属性数を N とすると, 状態空間 (部分集合の数) は 2^N となる．つまり，N が大きくなるにつれ指数的に大きくなる．そこで最適な部分集合を探すために，すべての組合せを調査するのは非現実的な選択となる．全組合せを調査することなしに，適当な属性の部分集合を探す方法は，大きくフィルター法とラッパー法に分類される．

連絡先: 三浦輝久 (財) 電力中央研究所 201-8511 東京都狛江市岩戸北 2-11-1 TEL 03-3480-2111, FAX 03-5497-0318, t-miura@criepi.denken.or.jp

Algorithm 1 Relief

▷ *n: the number of samplings*
▷ *N: the number of attributes*
1 **for** $j := 1$ **to** N **do**
  $M_j := 0$
2 **end for**
3 **for** $i := 1$ **to** n **do**
4 sample x_i.
5 query near hit x_h and near miss x_m of x_i.
6 **for** $j := 1$ **to** N **do**
7 $M_j := M_j - \mathrm{diff}_j(x_i, x_h)/n + \mathrm{diff}_j(x_i, x_m)/n$
8 **end for**
9 **end for**

$$\mathrm{diff}_j(x_i, x_l) = \begin{cases} |\, x_{ij} - x_{lj} \,| & (A_i \text{ is numeric}) \\ 0 & x_{ij} = x_{lj} (A_i \text{ is nominal}) \\ 1 & x_{ij} \neq x_{lj} (A_i \text{ is nominal}) \end{cases}$$

図 1: The Relief Algorithm

2.1 Filter Method

フィルター法は機械学習，データマイニングを始める以前に, データセットの一般的特徴に基づいて, 使用する部分集合をふるいにかける手法である．見込みのある属性集合をフィルタリングするため，フィルター法と呼ばれる．フィルター法の代表的手法である Relief[Kira 92a, Kira 92b] は各属性に関連性の重みを割り当てる．アルゴリズムは図 1 の通りである．ここで near hit とは, 選択されたサンプルと最も近い同クラスのサンプル，near miss とは最も近い異クラスのサンプルであり，距離は全属性を用いたユークリッド距離により計算される．最終的な M_j の値が属性 A_j の重要度となる．つまりより大きな値を持つ属性ほど, 関連性が高い属性となる．ユーザが与えた閾値を越える重要度を持つ属性を最終的に使う属性として用いる．

2.2 Wrapper method

ラッパー法 [Kohavi 97] では，属性の部分集合の評価を機械学習手法自身により行なう．どの部分集合の評価をどのような順序で行なうかは，探索アルゴリズムに依存する．代表的な探索アルゴリズムとしては，空集合から始めて, 1 つずつ属性

Algorithm 2 Wrapper Method

▷ S: 現在の部分集合
▷ $acc(S)$: 学習による部分集合 S の評価値
▷ a: 属性
▷ $maxacc$: これまでの最大評価値

```
1  S := ∅
2  maxacc := 0
3  for each a ∉ S do
4      calculate acc(a ∪ S)              ▷ 節点の生成
5  end for
6  a' := arg max_a acc(a ∪ S)
7  if acc(a' ∪ S) > maxacc then
8      maxacc := acc(a' ∪ S)
9      S := S ∪ a'
10     Go to 3
11 else
12     return maxacc
13 end if
```

図 2: 前向き探索によるラッパー法

を追加していく, 前向き探索．全属性から始めて，1 つずつ集合から属性を取り除いていく後向き探索がある．次にどの部分集合を評価するかを決定するために, 生成されたすべての節点 (属性の部分集合) に関して, 学習アルゴリズムを適用するため， 計算量が非常に大きくなるという問題点がある．つまり，探索問題としてみると，節点の生成に計算量がかかり，属性数が多い場合, 節点の展開に膨大な計算量がかかる問題と見ることができる．一般に属性数が大きいほど, 機械学習の計算時間がかかるため, 計算量の観点から見れば, 後向き探索より, 前向き探索の方が現実的である．前向き探索によるラッパー法のアルゴリズムを図 2 に示す.

2.3 フィルター法とラッパー法の優劣

フィルター法はデータマイニングや機械学習の前に 1 度実行すれば良いという計算量の点での利点がある．一方, 精度に関しては，一般にラッパー法の方が良いといわれている．このため, 属性選択にはラッパー法を適用することが望まれる．けれども, ラッパー法は, 属性数が多くなった時に学習アルゴリズムの実行回数が膨大になり計算量的に実行が非現実的になるという問題を持つ.

3. 提案手法

提案手法は, ラッパー法における子節点の評価の前に, より計算の軽いフィルターを用いる．このフィルターにより見込みのある節点を篩にかけ，残った少数の節点に対してラッパー法を適用することで，節点の展開にかかる計算量を軽くする．提案するフィルターは図 3 の通りである.

3.1 アルゴリズム

図 3 で，near $\mathrm{hit}_S(x_h)$,near $\mathrm{miss}_S(x_m)$ は現在用いている属性 S で計算された, 最近傍同符号サンプルと異符号サンプルである．従来のフィルター法は, データマイニングの前処理として実行されているため, 現在どの節点 (部分集合 S) を評価しているかという文脈が入っていなかった．提案アルゴリズムでは, 以下の点で, 現在いる節点の文脈を取り込むことにより, 各節点 (部分集合) における, 未使用属性の評価を行なっている.

- 現在の学習結果で, 誤判別のサンプルだけを利用して，計算を行なう (アルゴリズム 3 行目).

Algorithm 3 Subset-Relief

▷ S: 現在の属性の部分集合
▷ n: 誤判別サンプル数
▷ k: 未使用属性数

```
1  for j := 1 to k do
       W_j := 0
2  end for
3  for すべての誤判別サンプル x_i に対して
4      x_i の near hit_S(x_h) and near miss_S(x_m) を検索する.
5      for each a ∉ S に対して
6          W_a := W_a − diff_a(x_i − x_h)/n + diff_a(x_i − x_m)/n
7      end for
8  end for
```

図 3: Subset-Relief Algorithm

- near hit, hear miss の計算に, 現在使用している属性のみを用いる (アルゴリズム 4 行目).

誤判別サンプル数が 0 の場合は，全サンプルを使うか, Relief のようにランダムにサンプリングを行なう．本論文では全サンプルを用いて計算を行なった．このフィルターは現在誤判別しているサンプルと同じクラスのサンプルが近くに存在する属性には高い得点を与え, 逆に異なるクラスのサンプルが近くに存在する属性には低い得点を与える．提案手法では，この未使用属性に与えられた得点を, 属性追加の望ましさとして解釈し，実際に学習アルゴリズムで属性の評価をすることなしに，子節点に関する評価を行なう．アルゴリズムは 1 回の実行で，全未使用属性の評価を行なえるので，属性数が大きい場合，ラッパー法に比べて, 非常に少ない計算量で子節点の評価が実行できる.

3.2 Subset-Relief の基本性能

実際に Subset-Relief がラッパー法とどの程度一致するか．ラッパー法の代わりとなり得るのかについて, 基本的性能を調べる．調査に際しては以下の 2 つの問題を用いた [Newman 98].

データ名	属性数	サンプル数	状態空間中の節点数
Breast Cancer Wisconsin	10	699	1024
Pima Indian Diabetes	8	768	256

ラッパー法の機械学習アルゴリズムとしては Support Vector Machine を用い, 節点の評価は 10-fold クロスバリデーションで行なった.

3.21 Breast Cancer Wisconsin:次に示すのは，それぞれの節点で，ラッパー法で実際に 1 位になる子節点を Subset-Relief では何位と予測したかを表す表である.

1 位	2 位	3 位	4 位	5 位	6 位	7 位	8 位	9 位以下
594	166	132	73	43	11	3	1	0

また, Subset-Relief が 1 位に予測した節点の実際の順位は平均 1.789834 位であった．Subset-Relief は多くの場合に実際の 1 位を 1 位と予測し，そうでない場合にも, 比較的上位に予測していることがわかる．図 4 は各節点での子節点のフィルターによる順位とラッパー法による子節点の順位の間の spearman の順位相関値 (ρ) の頻度分布である．この図から明らかなように実際の順位をある程度うまく予測していることがわかる.

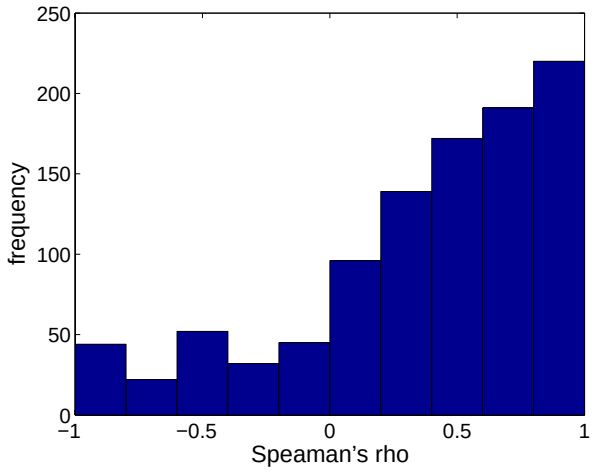

図 4: Spearman's rho(Breast Cancer Wisconsin)

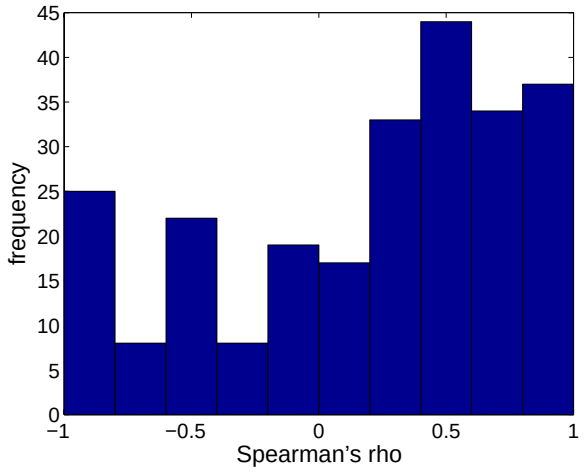

図 5: Spearman's rho(Pime Indians Diabete)

3.22 Pima Indians Diabetes:次に示すのは，それぞれの節点で, ラッパー法で実際に 1 位になる子節点を Subset-Relief では何位と予測したかを表す表である．

1 位	2 位	3 位	4 位	5 位	6 位	7 位以下
126	62	40	21	2	4	0

一方, Subset-Relief が 1 位に予測した節点の実際の順位は平均 2.0039216 位であった．図 5 は各節点でのフィルターによる順位とラッパー法による順位の間の spearman の順位相関値 (ρ) の頻度分布である．

上記の 2 つの表, 図が示す通り, 提案フィルターによる順位は, ある程度, 実際の精度を反映している．したがって，属性数が多く，全節点を生成し，展開するのが困難な場合に, Subet-Relief を用いて, 精度が高い節点のみを生成する候補として残し，実際にラッパー法により生成することで，部分的に展開することが可能となる．

表 1: 最良 k 前向き探索 (問題 Breast)

Breast	精度	生成節点数
best-1	0.972182	6
best-2	0.972182	12
best-3	0.976574	24
	0.973646	24
best-4	0.976574	29
	0.973646	29
best-5	0.973646	29
best-6	0.973646	33
best-7	0.973646	36
best-8	0.973646	38
forward search	0.973646	39

表 2: 最良 k 前向き探索 (問題 Pima)

Pima	精度	生成節点数
best-1	0.773438	8
best-2	0.774740	14
best-3	0.778646	18
best-4	0.778646	23
best-5	0.778646	27
best-6	0.778646	30
best-7	0.778646	32
forward search	0.778646	33

4. フィルターによる生成節点の制限

4.1 探索方法

まず前向き探索に対して，Subset-Relief で節点の望ましさを計算し，上位 k 節点だけを生成し，そのうち最高の精度を持つ節点を探索していくという方法が考えられる．これをここでは最良 k 前向き探索と呼ぶ．k が属性数 m と一致すると前向き探索と全く同じ動きとなる．本論文では， 機械学習手法としては Support Vector Mashine を用い, 部分集合の評価は 10-fold クロスバリデーションにより行なった．

4.2 実験

最良 k 前向き探索を Breast, Pima の 2 問題に適用する．適用結果は表 1,2 の通りである．同じ欄に複数の結果があるのは，途中でタイブレークが入るためである．

4.3 結果

まず当然のことだが，k を小さい数にすることによって, 生成節点数を少なくすることができる．節点生成の計算量は大きいので，これは計算量の大きな削減になる．一部数値が大きい場所は, 探索が続くためである．

もう一点, k が小さいからと言って，必ずしも結果が悪いわけではないと言うことがわかる．逆に単純な前向き探索に比べ，より良い解を発見している場合もある (breast $k = 3$)．これは節点を展開し, 全生成節点から最良の節点へと探索を進める前向き探索が必ずしも良い結果を生み出さないことを示している．つまり適用問題の探索空間では最良子節点に沿って精度が単調に増加していく経路が最良の経路ではないのである．

提案した Subset-Relief 法は大規模な属性数のデータマイニングに適用することを目的に開発された．しかし, 適用結果が示す通り, 大規模属性データだけでなく, 小規模データへの以下のような適用も考えられる．

表 3: Madelon result

アルゴリズム	属性数	テスト精度	生成節点数
best-1	1	0.5811	2
best-2	5	0.7139	12
best-3	6	0.7733	21
best-5	2	0.6206	15
best-10	2	0.5739	30
best-20	6	0.8622	140
best-50	9	0.8650	500
forward search	9	0.8394	5000

1. 前向き探索を行なう.
2. $k = 1, 2, 3$ と徐々に前向き k 探索の k を増やしていき, 前向き探索と同じ経路をたどる結果が出るまで続ける.

上記のようにすることで, 単純な前向き探索では探索されなかった部分の探索が実行でき, より良い解を発見する可能性が増加する.

5. 大規模問題への適用

前向き k 探索法を用いて実際に大規模なデータマイニング問題への適用を行なった. 使用したデータは MADELON であり, 属性数 500, 訓練事例数 1000, 検証事例数 300, テスト事例数 900 という特徴を持つ. NIPS2003Workshop on Feature Extraction の feature selection challenge[NIPS 03] で使用されたものである. ラッパー法のアルゴリズムとして Support Vector Machine を用い, 節点の評価は検証事例集合で行なった. 適用の結果は表 3 の通り. ここでも, forward search よりも best-20,best-50 の方が良い精度の節点へ到達できた. best-20 では forward search の 2.8%の節点数で同等の結果に到達でき, 大幅に計算量が削減できることがわかる.

6. まとめ

ラッパー法で属性数が多い場合に, 節点展開の計算量を軽くするための近似フィルターとして, Subset-Relief を提案した. Subset-Relief は実際の節点の精度による順位をうまく予測することができる. したがって, Subset-Relief を用いることで, 実際に節点を生成することなく, 節点の良さを推定することができ, 無駄な計算を省くことができる. データマイニングのように考慮すべき属性数が膨大な場合には, 未使用属性の価値を一度に評価できる Subset-Relief を用いて, 上位 k 属性のみを実際に機械学習手法を用いて評価することで, 近似的に前向き探索が実行可能となる. また, 属性数が少ない場合でも, 上位 k 前向き探索法を k を増やしつつ実行することで, 単純な前向き探索では探索されなかった空間を探索することができ, 発見する解の改善が期待できる.

謝辞

議論をしていただき, 有益なコメントをいただいた京都大学石田亨先生, 大阪大学元田浩先生, 鷲尾隆先生, 電力中央研究所小野田崇氏に感謝致します.

参考文献

[Newman 98] D.J. Newman, S. Hettich, C.L. Blake and C.J. Merz, UCI Repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Sciences, http://www.ics.uci.edu/∼mlearn/MLRepository.html, (1998)

[NIPS 03] Feature Selection Challenge, NIPS 2003 Workshop on Feature Extraction, http://nipsfsc.ecs.soton.ac.uk/, (2003)

[Kira 92a] K. Kira and L. Rendell, A Practical approach to feature selection, Proceedings of International Conference on Machine Learning, pp. 249–256, (1992)

[Kira 92b] K. Kira and L. Rendell,The feature selection problem: traditional methods and new algorithm, Proceedings of the 10th National Conference on Artificial Intelligence(AAAI-92), pp. 129–134,(1992)

[Kohavi 97] Ron Kohavi and George H. John: Wrappers for Feature Subset Selection, Artificial Intelligence, Vol 97, No 1-2, pp. 273-324, (1997)

[Das 01] , Sanmay Das, Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection, Proceedings of the Eighteenth International Conference on Machine Learning, pp. 74–81, (2001)