# Data-driven multi-layer feature analysis method incorporating domain expert knowledge


## 摘要

在一些模式识别任务中，经常会面临着大量高维数据，而特征空间维数的增加会引发“维度灾难”问题，严重影响了模型的预测性能。特征选择方法能够剔除模式中的不相关、冗余、以及噪音信息，进而提高学习器的泛化性能，减少计算成本，增强数据的可解释性。目前的特征选择方法多种多样，大致可以分为过滤式、封装式、嵌入式三种，但它们存在的问题也很明显。一方面它们都能在一定程度上剔除数据中冗余和无关的特征，但它们完全是数据驱动的，导致特征选择结果存在不稳定性，进而可能会加大领域专家认为的关键特征被剔除的风险，影响机器学习算法的预测精度；另一方面特征选择方法存在多样性，每一种方法针对~~某些~~特定的问题都可能拥有其独特的优势，因此领域专家很难从其中快速的挑选出符合自己需求的特征选择方法。为了解决上述问题，本文结合过滤方法、封装方法、嵌入方法的优点，在此基础上提出一种数据和领域专家知识共同驱动的多层级自动式特征分析方法，逐层地消除特征中的稀疏性、不相关性和冗余性，并统计和分析特征与特征之间的关联关系以及特征与决策属性的因果关系（特征重要度（feature imporance），特征影响决策属性的程度）。该方法能够从稀疏性评估、相关性评估和冗余性评估三个方面，分层地对样本数据特征集进行自动地、过滤式的特征筛选；同时，构建专家经验表示与融入方法，采用评分法将专家经验定量化，通过建立特征重要度评分指标将专家经验与特征选择结果结合。在研究中，我们在没有引入领域专家经验和集成专家经验的特征分析方法上分别做了两组实验并进行了对比。实验结果表明融入专家经验的特征分析方法能有效筛选出最优特征子集。



关键字：特征选择，过滤式、封装式、嵌入式方法，领域专家经验，特征重要度



## Abstract

While high dimensionality in datasets is an often occurrence in many pattern recognition tasks, the increase of feature space leads to the curse of dimensionality and hinders the performance of learning algorithms. FS methods are designed accordingly to counter such problem by eliminating the irrelevant, redundant and noisy information in patterns. Such practices improve the generalization performance of the learner, reduce the computational cost, and enhance the interpretability of the dataset. The existing FS approaches are data-driven and can be sorted into three categories including the filter methods, wrapper methods, embedded methods, in which redundancy and irrelevancy in the dataset are removed to some extent. Due to the data-driven approaches however, the results of selected features exhibit instability and objectivity, increasing the risks of removing crucial features and negatively impacting the prediction accuracy. This paper expands upon the advantages of the exsiting three methods and proposes a multi-layer feature analysis method incorporating domain-expert knowledge, whereby the sparsity, irrelevancy and redundancy are filtered out iteratively. The proposed method conducts a multi-layer feature selection

process based on the evaluation of dataset sparsity, relevancy and redundancy. a formal expression of domain expertise and the integration of such domain-expertise are also introduced to quantize domain-expertise via a scoring mechanism and construct an comprehensive model via Importance of Feature(IOF) index considering both domain-expertise and feature selection algorithms.

**Keywords: Feature Selection, Filter, wrapper, embedded method, Domain expert knowledge**

# 1 Introduction

在一些现实应用中，比如数据挖掘、机器学习等，经常会面临着大量高维数据。考虑到数据维数的增加会带来“维度灾难”问题，同时数据中冗余和不相关的信息也会相应的增加，这些信息可能极大降低机器学习算法的泛化能力，提高计算复杂度。为了解决这些问题，特征选择技术通过预先定义评估准则从高维特征集中挑选出最优特征子集，其并不改变变量的原始表示，因而保留了特征的最初语义，能使得领域专家更好的理解数据 [1] 。一个有效的特征选择算法能够在降低数据维数的同时，剔除数据中的无关、冗余、噪音信息，降低模型过拟合的风险 [2] ，提高分类器的泛化性能。

根据评估过程的不同，现有的特征选择方法大致可分为三类：过滤式方法、封装式方法、嵌入式方法 [4] 。图 1 展示了目前的特征选择的分类情况以及相应的算法和技术。过滤式方法是利用统计学理论、信息论知识来挖掘数据本身的内在特征，进而评估特征子集的好坏。它并没有涉及到任何机器学习算法，因此过滤式方法简单有效，但这种方法最大的缺陷在于其特征选择的结果可能并不理想。Relief [5] 算法最早由 Kira 提出，它是是一种特征权重算法，根据各个特征和类别的相关性赋予特征不同的权重，权重小于某个阈值的特征将被移除。由于 Relief 算法比较简单，但运行效率高，并且结果也比较令人满意，因此得到广泛应用，但是其局限性在于只能处理二分类问题。Kononeill [17] 对 Relief 算法进行了扩展，得到了 Relief-F 算法，可以处理多类别问题。Peng [6] 等人基于互信息提出一种过滤式特征选择算法 mRMR，该方法的核心思想是能够最大化特征与分类变量之间的相关性，而最小化特征与特征之间的相关性。Deisy [7] 等人使用了基于信息增益的对称不确定性分析，通过计算所有特征与类别变量熵之间的差异，信息较少的特征可以很容易地识别出来。封装式依赖于预先确定的学习算法以及评价准则对生成的特征子集进行评估，最后从中挑选出满足其最优评价指标的特征子集。它本质上是一种组合优化问题，显而易见，它的计算复杂度高，耗费时间长，但就预测精度来说要远远优于过滤式方法。Hui-Huang [8] 等人介绍了一种混合式特征选择方法，它结合了过滤式和封装式两种特征选择方法的优点。候选特征子集首先通过计算高效的过滤式方法从原始特征集中选择，然后该候选特征子集由更精确的封装式方法进一步调整。Cheng-Lung [9] 使用遗传算法和 SVM 同时优化模型参数和特征子集且不降低 SVM 的分类精度。Gang Chen [10] 介绍了一种新的封装式特征选择方法，即余弦相似度量支持向量机（CSMSVM），通过在支持向量机（SVM）中引入余弦距离来消除分类器构造过程中的不相关或冗余特征。Atsushi [11] 提出了一种融合过滤式和封装式特征选择方法，用进化算法选择最优特征子集，在该方法中，基于相关性的特征选择（CFS）和最小冗余和最大相关性（mRMR）算法被用作过滤式评估方法，二进制遗传算法（BGA）和二进制粒子群优化算法（BPSO）被用作进化搜索算法。嵌入式方法融合了过

滤式和封装式两种方法的思想，先通过机器学习模型为所有特征打分，例如如线性模型、Lasso [12] 、Randomforest [13] 等学习算法都能获得每个特征的重要度，然后自动选择特征。嵌入式特征选择方法既继承了过滤式方法计算代价低的优点，又能筛选出分类精度高的特征子集，然而它易受到样本数量和维数的干扰造成模型训练时出现开销过大和过拟合问题，进而降低特征选择结果的可靠性和满意度。

近几十年来，大量的研究者和学者花费他们的精力和时间来研究特征选择过程并且提出了各种各样的特征选择算法。然而目前的特征选择方法（filtering，wrapper or embedded）主要存在两方面的问题。一方面特征选择方法具有多样化的特性，人们针对不同的实际问题可以选择不同的算法，这导致领域专家很难在各种各种的算法中快速准确地挑选出合适的。另一方面，现有的特征选择算法都是在数据驱动下，自动地剔除数据集的不相关、冗余特征。但在数据量少的某些应用领域，比如镍基高温合金蠕变寿命预测、锂电池离子电导率预测等，普通的变量选择方法随时可能会把领域专家认为重要的特征剔除，从而一定程度影响了机器学习算法的分类性能。幸运的是，在许多分类领域，专家们可能事先知道一些特征的相关性，这可以被认为一种先验知识或专家经验 [16] ，这些可用的知识，即使只涉及到一小部分特征，也可以作为指导特征选择的手段。一个学科领域的专家在经过多年的训练和实践后通常会在解决问题的过程中形成一些经验法则，并积累对新的、罕见的、复杂的或其他不太了解的现象的见解。正如 Domingos [14] 所表达的那样，使用专家知识可以影响知识的发现，并缓解模型训练中的过拟合问题。Yoon [15] 等人还指出，专家知识可用于描述属性与属性之间的关联关系、属性和属性的类别之间的因果关系。因此，尽管知识密集，基于专家判断的特征选择可能会导致一个有效的特征子集，而不受训练数据集分布的影响。传统的特征选择方法往往忽略了一些特征的先验知识。

本文结合过滤式方法、封装式方法、嵌入式三种方法的优点，在此基础上提出一种新的基于领域专家经验的多层级特征分析方法，分层处理特征集存在的稀疏性、不相关性和冗余性。不像现有的各种特征选择算法，完全是数据驱动的选择过程，而我们提出的方法融入了领域专家知识并将它作为指导特征选择的手段，可以有效筛选出最优特征子集，提高学习算法的分类性能。本文工作的主要贡献如下：融合过滤式、封装式、嵌入式三种方法的优点，提出一种新的多层级特征分析方法，分层地处理原始数据中的稀疏性、不相关、冗余和噪音信息。将领域专家经验融入到数据驱动的多层级特征分析方法中，既减轻了领域专家认为重要的特征被剔除的风险，也提高了特征选择算法的稳定性。构建了专家经验表示与融入方法，采用评分法将专家经验定量化，通过建立特征重要度评分指标将专家经验与特征选择结果结合，提高了最终特征子集的质量，进而模型的泛化性能得到提升。在不同数据集上的实验证实了提出方法的有效性和可行性。

这篇论文的剩余部分如下，在第二节中，我们详细介绍了提出的方法。在第三节中展示了实验和实验结果分析。在第四节中，我们做了总结以及未来的工作展望。

In many real-world applications, such as data mining, machine learning, one is often faced with a large number of high-dimensional data. Considering that the dramatic increase of data demensionality will cause the curse of demensionality, at the same time, redundant and irrelevant information in patterns will increase accordingly. Those information may greatly reduce the generalization ability of machine learning algorithms and increase the computational complexity. In order to solve these issues,

the feature selection techniques are designed to select the optimal feature subset from the high-dimensional feature set by pre-defining the evaluation criteria, which does not change the original representation of the features, thus they preserve the original semantics of the feature and provide better understanding of the datasets for a domain expert [1] . Additionally, An effective feature selection algorithm can reduce the dimension of the data, eliminate the irrelevant, redundant, noise information in the data, reduce the risk of overfitting, and improve the generalization ability of the classifier.

Based on the evaluation procedure, the exsiting feature selection methods can be widely divided into three categories, namely the filter methods, wrapper methods, and embedded methods [2] .

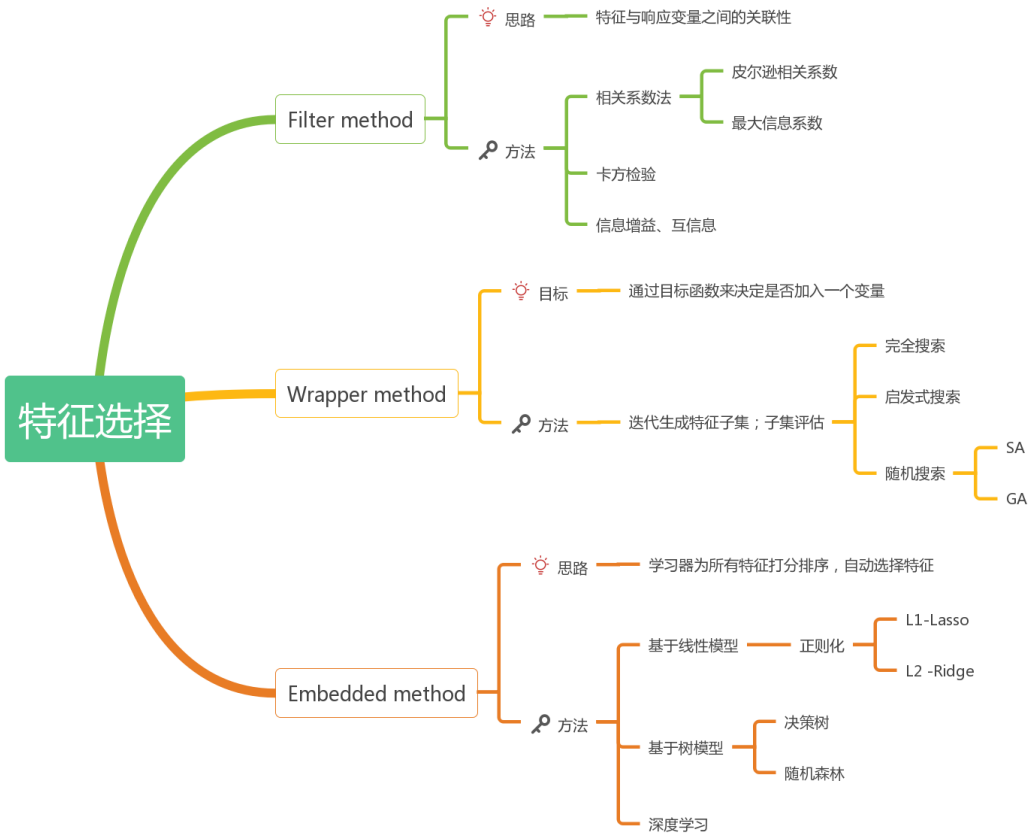


图 1 特征选择方法分类

## 2 The mechanism of the proposed novel method

为解决原始特征集存在的稀疏性、不相关性、冗余性等问题以及如何表示与融入专家经验，如图 2 所示，本节提出了融合专家经验的多层级特征分析方法，该方法结合了过滤式、封装式、嵌入式特征选择方法的优点，逐层处理原始数特征集中存在的稀疏性、不相关性和冗余性问题，同时建立专家经验的定量化表示方法，通过建立特征重要度评分指标将专家经验与特征选择结果结合，在每一层次对获得的特征子集进行模型验证，评价每一层生成的特征子集的优劣。本节中将从专家经验的表示方法、多层级特征分析方法以及两者的协同策略研究三个方面进行介绍。

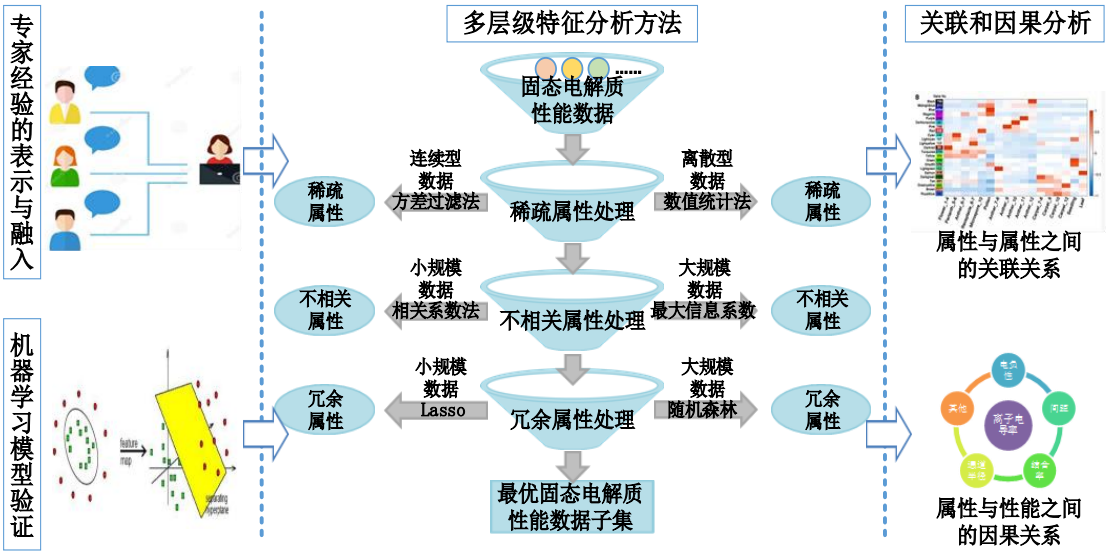


图 2 融合专家经验的多层级特征选择方法

## 2.1 The presentation method of domain expertise

在考虑将融合专家经验的多层级特征分析方法应用于实际的机器学习研究中的情况下，算法的使用专家可分为两类：当前用户与历史用户。此外，用户的身份也可能是材料专家、计算机专家等。不同的专家由于专业领域的不同，因此，在给与特征重要度时，权重也应有所不同。因此，本节中专家经验评分 $s$ 包括下列两大类指标：

**（1） 用户重要度评分**

用户特征重要度评分包括当前用户重要度评分 $su$ 以及历史用户重要度评分 $sp$，分别代表了用户凭借自己的经验对该特征相较于决策属性的重要度的评分情况，其定量表示如公式(1)所示：

$$su, sp = \begin{cases} 0 \\ 0.5 \\ 1 \end{cases} \tag{1}$$

其中，0 表示用户认为该特征不重要，0.5 表示用户不确定该特征的重要程度，1 表示用户认为该特征非常重要。

**（2） 用户评分权重**

用户评分权重包括当前用户以及历史用户的评分权重 $d$，其定量表示如公式(2)所示：

$$d=\begin{cases}1\\1.5\\2\end{cases} \tag{2}$$

其中，2 表示用户是材料专家，1.5 表示用户是计算机专家，1 表示用户既不是材料专家又不是计算机专家。

## 2.2 The data-driven multi-layer feature analysis method

多层级特征分析方法主要是针对数据原始特征集存在的稀疏性、不相关性、冗余性问题提出的，其思想是采用分层级方式进行特征筛选，确保最终的特征子集中特征与响应变量之间具有较强的相关性，特征与特征之间相关性较弱。该方法结合了过滤式、封装式、嵌入式三种特征选择方法的思想。横向上，对于每一层的筛选结果使用模型验证的方法进行评估，依据评估的结果再次筛选，直到得出本层能挑选出的最优特征子集；纵向上，依据求解最优特征子集的思路严格规定层次的先后顺序，每一层的结果作为下一层的输入，首先，第一层是消除稀疏性，主要是对原始特征集中单个特征的稀疏性进行评估，筛除稀疏属性；第二层是消除不相关性，对特征与目标变量之间的相关性进行评估，筛除与响应变量不相关的特征；第三层是消除冗余性，对特征与特征之间的冗余性进行评估，筛除冗余属性。主要思路如算法 1 所示。

**Algorithm 1：多层级特征分析方法**

**输入：**原始样本集 $S=\{x_1,x_2,...,x_m,y\}_{i=1}^{n}$，交叉验证次数 $k$，验证方法 $M$，稀疏性阈值 $\varepsilon$，子集搜索策略 PTA

**输出：**最优特征子集 $S_{optimal}$

**算法流程:**

//归一化处理

1. For each $x_i$ in $S$,

$$x_i^{'}=\frac{x_i-x_{\min}}{x_{\max}-x_{\min}}$$ //对第 $r$ 个特征进行归一化

Add $x_i^{'}$ to $S_1$

End For

$P_{error\_1}$=*ModelValidation*($S_1$, $k$, $M$) //模型验证，具体算法见**错误!未找到引用源。**节

//第一层：消除原始特征集中的稀疏属性

2. $S_{temp}$=*SparsityEvaluate*($S_1$,$\varepsilon$) //稀疏性评估，具体算法见**错误!未找到引用源。**节

   $P_{error\_2}$=*ModelValidation*($S_1$-$S_{temp}$, *k*, *M*)

   If ($S_{temp}$=*null* || $P_{error\_2}$<$P_{error\_1}$) //模型预测精度增加或不存在稀疏属性

   $S_2$=$S_1$-$S_{temp}$

   Else

   $\varepsilon$=$\varepsilon$±$\Delta t_1$ //调整稀疏性评估阈值 $\varepsilon$

   goto Step 2

//第二层：消除不相关属性

3. Spea_coor, Smic_coor =*CorrelationEvaluate*($S_2$) //相关性评估，具体算法见**错误!未找到引用源。**节

4. Repeat：

   Pea_ subset=PTA(Sper_coor)

   $P_{error\_3}$=*ModelValidation*(Pea_ subset, *k*, *M*)

   If $P_{error\_3}$ < = $P_{error\_2}$

   $P_{error\_2}$ = $P_{error\_3}$

   goto Step 4

   Else

   break

   Pea_final_subset=Pea_subset //输出最终特征子集（pearson 相关系数）

   同样的，Mic_final_subset=Mic_subset //输出最终特征子集（最大互信息系数）

   S3=Mic_fianal_subset ∩ Pea_final_subset //两种相关性分析方法特征选择结果取交集

   $P_{error\_4}$ = *ModelValidation*(S3, *k*, *M*)

//第三层：消除冗余属性

5. $S_{temp}$=*RedundancyEvaluate*($S_3$) //冗余性评估，具体算法见**错误!未找到引用源。**节

$P_{error\_4}$=*ModelValidation*($S_3$-$S_{temp}$, *k*, *M*)

If ($S_{temp}$=*null* || $P_{error\_4}$<$P_{error\_3}$)

$S_4$=$S_3$-$S_{temp}$

6. Output $S_{optimal}$

第一层筛除稀疏属性的过程中，为了解决离散型变量和连续型变量中存在的稀疏性问题，分别采用数值统计法和方差分数（计算属性值的方差大小）进行处理。首先判断每个特征的数值类型，然后选择对应的稀疏性评估方法。如果变量的数据类型是离散的，则采用数值统计法来筛除离散型特征，计算每个特征的稀疏性评估值，将各特征的稀疏性评估值与数值统计法的阈值进行比较，如果稀疏性评估值大于过滤阈值，则筛除该条件属性；否则，保留该属性。如果特征的数据类型是连续的，使用方差分数来筛除连续型特征，对每个特征进行稀疏性评估，将各特征的连续稀疏性评估值与方差过滤法的阈值进行比较，如果稀疏性评估值小于阈值，则筛除该特征；否则，保留该属性。之后通过模型验证的方法来判断此时筛选出的特征子集结果是否较筛选前的特征子集更优秀，这里使用性能普遍比较好的 SVR 作为用来验证的模型，如果评估得出筛选后的更优，自适应地调整阀值，依据调整后的阀值再次进行筛选评估，直到筛选结果不优于筛选之前的特征子集，此时更优的特征子集则为本层算法得出的特征子集，并作为下一层的输入。具体算法如算法 2 所示。

**算法 2：**稀疏性评估 *SparsityEvaluate*

**输入：**样本数据集 $S_1$，离散型稀疏性阈值 $\varepsilon_1$，连续型稀疏性阈值 $\varepsilon_2$

**输出：**待删除样本数据集 $S_1'$

**算法流程:**

1：初始化 $S_1'$ 为空

2：For each $x_i$ in $S_1$

If ( $x_i.type = bool$)

$$score1_i = \frac{1}{n}\sum_{i=1}^{n} x_i$$ //计算 $x_i$ 的离散稀疏性评估值

*E*lse

$$score2_i = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \quad \text{//计算 } x_i \text{ 的方差分数}$$

If ($score1_i < \varepsilon_1$ and $score2_i < \varepsilon_2$) //$\varepsilon_1$ 为离散型稀疏性阈值，$\varepsilon_2$ 为连续型稀疏性阈值

Add $x_i$ to $S_1'$

EndFor

3：Output $S_1'$

第一层消除稀疏属性之后，得到数据集 S2，这时数据中可能存在一些不相关特征。因此在第二层对不相关属性进行分析和筛除。由于数据集中特征与目标变量之间可能同时存在线性相关性和非线性相关性，所以在本层同时使用皮尔逊相关系数和最大互信息系数来评估特征和响应变量的相关性。皮尔森相关系数是一种最简单的，能帮助理解特征和响应变量之间关系的方法，该方法只能衡量变量之间的线性相关性，其结果的绝对值表明变量之间的相关程度，值越大，相关性越强，反之亦然。最大互信息系数能够很好的衡量两个变量之间的非线性关系。本层首先分别计算每个特征与类别属性之间的皮尔逊相关系数和最大互信息系数，接着根据相关系数对特征进行降序排列，得到两个按相关系数降序排序的原始特征集，然后采用 PTA 搜索策略生成特征子集，同样选用 SVR 的预测精度作为评估准则，选出两个最终的特征子集。为了保证特征与响应变量之间同时存在线性和非线性关系，最优特征子集取两个特征子集的交集。具体算法如算法 3 所示。

**算法 3：**相关性评估 *CorrelationEvaluate*

**输入：**样本数据集 $S_2$

**输出：**按降序排序的每个特征的皮尔逊相关系数和最大互信息系数

**算法流程:**

1：For each $x_i$ in $S_2$

$$pea\_coor_i = P(x_i) = P = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad \text{//Pearson 相关系数}$$

$$mic\_coor_i = \mathrm{MIC}(x_i; y) = \max_{|x||y|<B} \frac{I[x; y]}{\log_2(\min(|x|, |y|))}$$ //最大互信息系数

2： $sort\_pea\_coor_i = sorted(pea\_coor_i)$ //Pearson 相关系数降序排列

$sort\_micoor_i = sorted(mic\_coor_i)$ //最大互信息系数降序排列

EndFor

4：Output Spea_coor, Smic_coor

在第二层消除不相关特征后，数据集 S3 中特征与响应变量之间都具有较强的相关性，此时 S3 中特征与特征之间可能还存在较强的相关性，称之为冗余性。虽然特征与特征之间的冗余性同样可以通过相关性度量和信息度量等方法来衡量，但在特征数目较多的情况下，这些方法计算复杂度高且难以实现，所以一般不采用。目前还没有一种方法是专门针对冗余性提出的，而是以赋予特征权重的方式来表现特征的重要度，并且依据给予特征权重时，当一个特征权重得分较高时，其他与它有冗余关系的特征权重会急剧下降的特性，因此可以有效地筛除冗余属性。目前，筛除冗余属性常用的方法有 Lasso、Randomforest 等，其中 Lasso 在高维海量和高维小样本是容易出现开销过大或过学习问题，而 RandomForest 在该情况下能够保持较好的稳定性。本层中特征过滤的主要思想是，首先对样本数据集的数目 m 和维数 n 进行分析。如果 m 大于 4000 或 n>40 时，采用随机森林作为特征权重评估方法[57]，否则则采用 Lasso，具体算法可参考文献[58]中 LARS 算法所实现的 Lasso 特征选择。根据特征权重评估方法计算各个特征的权重。最后比较特征权重值和冗余性过滤阈值，如果权重值低于冗余性过滤阈值，则从样本数据集中删除该属性数据；否则，保留该属性。之后通过模型验证的方法来判断此时筛选出的特征子集结果是否较筛选前的特征子集更优秀，这里使用 SVR 作为用来验证的模型，如果评估得出筛选后的更优，则根据算法特性一定程度地调整阀值，依据调整后的阀值再次进行筛选评估，直到筛选结果不优于筛选之前的特征子集，此时更优的特征子集则为本层算法得出的特征子集，并作为下一层的输入。具体算法如算法 4 所示。

**算法 4：** 冗余性评估 *RedundancyEvaluate*

**输入：** 样本数据集 $S_3$，冗余性阈值 $\gamma$

**输出：** 待删除样本数据集 $S_3'$

**算法流程:**

1：初始化 $S_3'$ 为空

2：$S_3$→$(m, n)$ //$m$ 为样本数据集 $S_3$ 的样本数，$n$ 为 $S_3$ 的维度

3：If ($m$>5000 || $n$>40)

*Random_Forest*($S_3$)　//计算各特征的权重 *x.weight*，具体算法实现见文献[57]

Else

*LARS*($S_3$)　//使用 LARS 为文献[59]中所实现的 Lasso 特征选择算法

4：For each $x_i$ in $S_3$

If (*x.weight*<$\gamma$)　//$\gamma$ 为冗余性阈值

Add $x_i$ to $S_3'$

EndFor

5：Output $S_3'$

结果验证是指通过机器学习方法验证所选择的特征子集的有效性。在多层级过滤式特征选择中，为了保证特征选择的有效性，将对每一层特征选择后得到的特征子集采用机器学习方法进行验证。在特征子集模型验证中，机器学习方法由使用者依据具体学习问题确定，通常采用 SVM 作为验证方法。由于机器学习方法的参数对样本的维度和大小都具有一定的敏感性，因此，采用目前常用网格搜索对其参数进行寻优。此外，为了保证实验验证结果的准确性，以交叉验证的方式对模型的预测精度进行验证，交叉验证次数 k 主要是依据样本数的大小而确定，通常设置 k=5。在模型预测精度评估中，所采用的误差评估方法 ErrorEvaluate 有平均绝对误差 MAPE、均方根误差 RMSE 和判定系数 R2。通常采用 RMSE 来度量，RMSE 可以有效地反映预测值误差 Perror 的变化，同时对预测值最大或最小误差非常敏感。具体算法如算法 5 所示。

**算法 5：**模型验证 *ModelValidation*

**输入：**样本数据集 *S*，交叉验证数 *k*，学习器 *M*

**输出：**模型预测结果的 $P_{error}$

**算法流程:**

1：初始化 $P_{error}$=0

2：$S=\{S_1,S_2,...,S_k\}$　//将样本数据集 *S* 按随机抽样的方式划分为 *k* 份

For *i*: 1 to *k*

$S_{train}$=*S*-$S_i$; $S_{test}$=*S*-$S_i$

*Train a regression model* $M'$ *by learning machine M from* $S_{train}$

*Test* $S_{test}$ *by* $M'$ *and obtained prediction value* $\{y'\}_{S_{test}}$

$P_{error}$+=*ErrorEvaluate*($y, y'$) //记录交叉验证的累计预测误差

EndFor

3：Output $P_{error}= P_{error} /k$

## 2.3 Collaborative strategy research

在使用多层级特征分析方法进行特征筛选时，对特征的筛选判断通过特征选，其定量表示如公式(3)所示：

$$sa = \begin{cases} 0 \\ 1 \end{cases} \tag{3}$$

其中，0 表示筛除该特征，1 表示保留该特征。

结合小节中对专家经验的评价指标，特征重要度评分(Importance of Feature, IoF)包括专家经验评分 *s* 及特征分析方法重要度评分两部分，其定量表述如公式(4)所示：

$$IoF = s + sa = su * \sqrt{1-\frac{3(m-n)^2}{4m^2}} + \frac{\sum sp * d}{\sum d} * (1-\sqrt{1-\frac{3(m-n)^2}{4m^2}}) + sa \tag{4}$$

其中，*n* 表示与当前用户特征打分相同的历史用户人数，*m* 表示为特征打过分的历史总用户数。当 IoF>=1 时，保留特征；当 IoF<1 时，筛除特征。

特征重要度评分 IoF 包含了下列 7 个核心思想：

1) 需要当前用户经验，历史用户经验以及特征选择方法的结果来共同判定是否筛选该特征；
2) 专家重要度评分考虑到了专家的领域问题，并设立了权值来划分各领域的区别，能够提高打分的可信度；
3) 当前用户评分和历史用户评分的权值和为 1，可以融合当前用户的经验与历史用户的经验，并且当前用户评分的权值在 0.5~1，历史用户评分的权值在 0~0.5，既保证了当前用户经验的主导性，又可以一定程度上融入历史用户的经验；
4) 当前用户与历史用户评分的权值基于历史用户评分的可信度；

5) 特征选择方法得出该特征不会被筛除的结果，那么该特征就不会被筛除；

6) 当前用户认为特征非常重要，那么该特征不会被筛除；

7) 当前用户认为该特征不重要，但结合其他专家的经验，该特征有被保留的可能性。

# 3 Experimental analysis

本节介绍了实验所使用的多组数据集、实验参数设置，并对实验结果进行了分析。

## 3.1 Datasets

为了验证多层级过滤式特征选择方法的有效性，通过 6 组数据集进行实验验证。实验数据集包括 3 组材料数据集和 3 组 UCI 公共数据集。数据集的具体信息如 Table 1 所示。

Table 1 : Statistics of the Datasets

| | 样本数 | 条件属性数 | 决策属性数 | 描述 |
|---|---|---|---|---|
| 数据集一 | 128 | 6 | 1 | 离子电导率预测 |
| 数据集二 | 5619 | 47 | 1 | 有机材料的密度预测 |
| 数据集三 | 669 | 18 | 1 | 有机材料的粘度预测 |
| 数据集五 | 1302 | 5 | 1 | 晶格常数预测 |
| Wine Quality | 1599 | 11 | 1 | 红酒质量预测 [1] |
| Residential Building | 372 | 107 | 2 | 住宅建筑数据 [2] |
| Energy efficiency | 768 | 8 | 2 | 能源效应数据 [3] |

1. http://archive.ics.uci.edu/ml/datasets/Residential+Building+Data+Set
2. http://archive.ics.uci.edu/ml/datasets/Wine+Quality
3. http://archive.ics.uci.edu/ml/datasets/Energy+efficiency

## 3.2 Experimental settings

Table 1 中前 3 组材料属性数据集中每个特征的专家重要度评分记录可以通过向不同的材料领域专家咨询而得知。而后三组数据集属于公共数据集，领域专家

较难给出特征重要度评分。所以本文分别采用融合领域专家经验的多层级特征分析方法和多层级特征分析方法对前 3 组材料数据集进行特征选择实验，比较两种方法在每一层特征筛选保留的特征以及模型的分类性能。而对后三组公共数据集采用仅采用多层级特征分析方法进行实验，验证我们提出的多层级特征分析方法的有效性和可行性。对于 6 组数据集，分别按层次进行特征选择实验，对于每层特征选择后获得的特征子集将作为下层特征选择的输入，并且在该层使用支持向量回归（SVR）进行模型验证。

由于实验参数因样本数据的不同，差异较大，且对特征选择及其模型验证结果影响较大。因此，本节实验参数设置策略如下：特征选择方法参数设置将根据具体样本进行寻优；模型验证参数将根据每一层学习样本进行寻优。具体实验参数设置为。

针对 6 组数据集，本节中的实验参数设置如下：

（1） 特征分析方法的选择及其参数设置

特征选择分为三层。第一层，将主要采用方差过滤法进行稀疏属性的筛除。方差过滤参数设置为，初始化方差过滤阈值 $\varepsilon$ =0.01，即当样本中属性值波动或数值比例小于 0.01 时，则认为该属性为稀疏属性。由于样本容量大小不同，稀疏属性对模型的预测精度影响也有所不同，所以在进行方差过滤时结合模型验证，对阈值 $\varepsilon$ 采用 ±0.01的波动进行调整，直至模型预测精度不再提高。第二层，相关性评估，将采用 Pearson 相关系数法，Pearson 相关系数 $\varphi$ 阈值为 0.4，对阈值 $\varphi$ 采用 ±0.1的波动进行调整，直至模型预测精度不再提高。第三层，冗余性消除。本层采用 Random Forest 和 Lasso 进行特征选择，筛除冗余属性。其中，Random Forest 最大深度 *depth* 依据寻优结果确定，寻优范围为[1,*n*]，其中 *n* 为特征数，Lasso 惩罚参数由网格搜索确定。

（2） 模型验证参数设置

本章主要采用 SVR 对特征子集的预测精度进行评估，针对不同数据集，在原始特征下将分别采用网格搜索进行参数寻优。在分别得到最优参数后，采用这些参数进行后续特征选择的模型验证。

（3） 交叉验证参数设置

本章实验样本数据均大于 100 条，样本量较为充足，将采用 k=10 次交叉验

证。每次验证中，9/10 样本用于训练，超过了总样本数的 2/3，能够较为完整地反映决策属性的变化规律，1/10 用于测试。

（4） 模型评估标准设置

模型评估主要从预测精度和训练时间两个方面进行。预测精度主要采用均方根误差 *RMSE*、平均绝对百分误差 *MAPE* 和拟合优度 $R^2$，如公式(5)-(7)所示；训练时间主要以秒（s）进行度量。

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y'_i-y_i\right)^2} \tag{5}$$

$$MAPE=\frac{1}{n}\sum_{i=1}^{n}\frac{\left|y'_i-y_i\right|}{y_i}*100\% \tag{6}$$

$$R^2=\frac{[\sum_{i=1}^{n}(y_i-\bar{y})(y'_i-\bar{y}')]^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2\cdot\sum_{i=1}^{n}(y'_i-\bar{y}')^2}*100\% \tag{7}$$

其中，$y_i$代表原始值，$y_i'$代表预测值，$\bar{y}$表示原始值的平均值，$\bar{y}'$表示预测值的平均值。*MAPE* 对所有的误差进行了平均，对预测值中最大或最小误差不敏感，*RMSE* 对一组预测值中的最大或最小误差反映非常敏感，$R^2$ 反映了模型对样本观测值的拟合程度。

各种特征选择方法均由 Python 实现，并调用 scikit-learn 工具包[60]

## 3.3 Experimental results and analysis

### 3.3.1 Multi-layer feature analysis method

**Private datasets**

Table 2 dataset 2

| | | RMSE | MAE | $R^2$ | feature_numbers |
|---|---|---|---|---|---|
| 原始特征集 | Layer0 | 0.062006 | 0.051037 | 0.671098 | 47 |
| 稀疏属性分析 | Layer1 | 0.061156 | 0.050319 | 0.699479 | 34 |
| 相关属性分析 | Layer2 | 0.056459548 | 0.04571785 | 0.741822 | 31 |
| 冗余属性分析 | Layer3 | 0.056459548 | 0.04571785 | 0.741822 | 31 |

Table 2

Table 3 dataset 1

| | | RMSE | MAE | R2 | feature_numbers |
|---|---|---|---|---|---|
| 原始特征集 | Layer0 | 0.161104 | 0.13102 | -3.075166 | 6 |
| 稀疏属性分析 | Layer1 | 0.161104 | 0.13102 | -3.075166 | 6 |
| 相关属性分析 | Layer2 | 0.130246 | 0.103612916 | -0.8496589 | 4 |
| 冗余属性分析 | Layer3 | 0.130246 | 0.103612916 | -0.8496589 | 4 |

Table 3

Table 4 dataset3

| | | RMSE | MAE | $R^2$ | feature_numbers |
|---|---|---|---|---|---|
| 原始特征集 | Layer0 | 0.196473 | 0.138197 | 0.071446 | 18 |
| 稀疏属性分析 | Layer1 | 0.19527 | 0.142096 | 0.025499 | 17 |
| 相关属性分析 | Layer2 | 0.188615489 | 0.133037876 | 0.202605 | 15 |
| 冗余属性分析 | Layer3 | 0.191765718 | 0.131543705 | 0.144872 | 12 |

Table 4

**Public available datasets**

Table 5 Residential Building dataset

| | | RMSE | MAE | $R^2$ | feature_numbers |
|---|---|---|---|---|---|
| 原始特征集 | Layer0 | 0.051782 | 0.041465 | 0.814994 | 107 |
| 稀疏属性分析 | Layer1 | 0.051782 | 0.041465 | 0.814994 | 107 |
| 相关属性分析 | Layer2 | 0.051362505 | 0.041149 | 0.817324 | 106 |
| 冗余属性分析 | Layer3 | 0.051362505 | 0.041149 | 0.817324 | 106 |

Table 5

### 3.3.2 Feature analysis method incorporating domain expertise

# 4 Conlusion and future work

本文针对数据中存在的稀疏性、不相关性、冗余性以及目前特征算法的不稳定性等问题，提出了数据和领域专家知识共同驱动的多层级特征分析方法。该方法能够从稀疏性评估、相关性评估和冗余性评估三个方面，分层地对样本数据特征集进行自动地、过滤式的特征筛选；同时，构建专家经验表示与融入方法，采用评分法将专家经验定量化，通过建立特征重要度评分指标将专家经验与特征选择结果结合。实验结果证明，融合专家经验的多层级特征分析方法在充分融入专家经验的同时，可以有效地筛选出最优特征子集，提高模型预测精度。

# Acknowledgement

# References


[1] Gui J, Sun Z, Ji S, et al. Feature Selection Based on Structured Sparsity: A Comprehensive Study[J]. IEEE Transactions on Neural Networks & Learning Systems, 2017, 28(7):1490-1507.

[2] Guyon, Isabelle, Elisseeff, et al. An introduction to variable and feature selection[J]. Journal of Machine Learning Research, 2003, 3(6):1157-1182.

[3] Maldonado S, Weber R. A wrapper method for feature selection using Support Vector Machines[J]. Information Sciences, 2009, 179(13):2208-2217.

[4] Chen G, Chen J. A novel wrapper method for feature selection and its applications[M]. Elsevier Science Publishers B. V. 2015.

[5] Kira, Kenji, Rendell, et al. A practical approach to feature selection[J]. Machine Learning Proceedings, 1992, 48(1):249-256.

[6] Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy[J]. IEEE Trans Pattern Anal Mach Intell, 2005, 27(8):1226-1238.

[7] Deisy C, Subbulakshmi B, Baskar S, et al. Efficient Dimensionality Reduction Approaches for Feature Selection[C]// International Conference on Conference on Computational Intelligence and Multimedia Applications. IEEE, 2007:121-127.

[8] Hsu H H, Hsieh C W, Lu M D. Hybrid feature selection by combining filters and wrappers[J]. Expert Systems with Applications, 2011, 38(7):8144-8150.

[9] Huang C L, Wang C J. A GA-based feature selection and parameters optimizationfor support vector machines[J]. Expert Systems with Applications, 2006, 31(2):231-240.

[10] Chen G, Chen J. A novel wrapper method for feature selection and its applications[J]. Neurocomputing, 2015, 159(1):219-226.

[11] Kawamura A, Chakraborty B. A hybrid approach for optimal feature subset selection with evolutionary algorithms[C]// International Conference on Awareness Science and Technology. 2017:564-568.

[12] Efron B, Hastie T, Johnstone I, et al. Least angle regression[J]. Annals of Statistics, 2004, 32(2):407-451.

[13] Hapfelmeier A, Ulm K. A new variable selection approach using Random Forests[M]. Elsevier Science Publishers B. V. 2013.

[14] Domingos P. The Role of Occam's Razor in Knowledge Discovery[C]// Data Mining & Knowledge Discovery. 1999:págs. 409-425.

[15] Yoon S C, Henschen L J, Park E K, et al. Using domain knowledge in knowledge discovery[C]// Eighth International Conference on Information and Knowledge Management. ACM, 1999:243-250.

[16] Ben Brahim A, Limam M. New prior knowledge based extensions for stable feature selection[C]// Soft Computing and Pattern Recognition. IEEE, 2015:306-311.

[17] Kononenko I. Estimating attributes: Analysis and extensions of RELIEF[C]// European Conference on Machine Learning on Machine Learning. Springer-Verlag New York, Inc. 1994:171-182.