

摘 要

本文提出了一种创新的高维数据集特征选择算法，命名为 SFE (Simple, Fast, and Efficient)。SFE 算法借助搜索代理和两个操作符来执行其搜索过程，即非选择和选择。该算法分为两个关键阶段：勘探和开发。在勘探阶段，非选择算子在整个问题搜索空间中进行全局搜索，识别不相关、冗余、琐碎和受噪声影响的特征，并将这些特征的状态从选择模式切换到非选择模式。而在开发阶段，选择算子则在问题搜索空间中寻找对分类结果影响较大的特征，并将这些特征的状态从非选择模式切换到选择模式。

虽然 SFE 算法在成功应对高维数据集的特征选择方面表现出色，但在数据集维数降低后，其性能提升并不显著。为解决这一问题，我们提出了一种混合算法，即 SFE-PSO (基于粒子群优化)，以在新的、较小的搜索空间中找到更为有效的特征子集。实验结果显示，SFE 和 SFE-PSO 两种算法在性能上明显优于其他算法，可被视为一种高效的高维特征选择算法。这种混合算法的引入为在新的搜索空间中找到最佳特征子集提供了有效的解决方案。在这篇文章的基础上，我们在搜索的两个阶段都对其代码进行了改进，进一步提升了算法的搜索能力。

目录

1	引言	1
2	相关工作	2
3	本文方法	4
3.1	本文创新点	4
3.1.1	创建搜索代理	4
3.1.2	提出了两种算子	4
3.2	搜索过程	4
3.2.1	非选择操作	4
3.2.2	平衡开发和探索	5
3.2.3	开发过程	6
3.3	物理降维	6
3.4	混合 EC 算法框架	7
4	改进思想	7
4.1	对 SFE 算法的改进	7
4.2	对 PSO 算法的改进	8
5	实验级结果分析	8
5.1	实验环境搭建	8
5.2	数据集上的性能提升	8
6	总结与展望	10

基于代理模型的高效特征选择算法

1 引言

计算机系统和新技术带来了大量高维数据集，应用广泛。机器学习用于分类，但高维数据导致维数灾难，增加了算法的时间和空间复杂度 [1]。为应对问题，提出降维方法 [2]，在大数据时代，降维方法助力更可靠的数据分析和决策。

通常，降维方法通常在整个分类的预处理阶段执行，它可以分为特征提取和特征选择 (FS)。特征提取算法的主要目标是根据数据集的主要特征创建新的特征来降低数据集的维数。FS 算法选择包含有关数据的重要信息的相关特征子集，并将不重要的特征从特征集中移 [2], [3]，防止特征过拟合并降低算法的性能。FS 方法保留了特征的物理意义，因为它们直接选择提供正确解释的相关特征。此外，由于高维数据集的特征提取方法的计算复杂度高，特征提取方法在某些应用中可能无法达到预期的性能。由于上述原因，研究并改进 FS 方法 [2], [4] 是一件很有意义的工作。

一般来说，FS 算法可以分为三类：基于过滤器，基于包装器和嵌入式 [2]。典型的基于过滤器的特征选择算法包括两个主要步骤。在第一步中，根据各种标准，如距离，相关性，一致性和信息增益确定每个特征的重要性和等级。在第二步中，具有较低秩的特征然后被过滤 [2], [5]。

基于包装器的 FS 算法使用机器学习分类器来评估所选择的特征。这些方法通常包括两个主要步骤，在第一步中，使用搜索方法搜索特征的最佳子集。在第二步中，用各种分类方法，如最近邻 (KNN) 和支持向量机 (SVM) 评估特征的最佳子集 [6]。重复这两个步骤以实现预期性能 [7]。以往的一些研究 [2] 通过引入混合 FS 方法。在混合 FS 方法中，会利用基于过滤器、基于包装器和嵌入式 FS 算法的组合。一般来说，基于包装器和嵌入式方法比基于过滤器方法更有效，因为它们在 FS 过程中与分类器交互。混合方法虽然利用了不同的算法的优势，但仍然面临着高计算成本的挑战。因此，为了降低计算复杂度，同时能在在 FS 过程中实现更好的效率，本文的重点是提供一种基于包装器的高维数据 FS 方法。

基于包装器的 FS 方法使用搜索算法来找到特征的最佳子集。在数据集中搜索最优特征子集的最简单方法是搜索整个问题搜索空间并评估每个可能子集的性能，这被称为穷举搜索方法。对于大规模的搜索，使用完整的搜索来找到最佳的特征子集几乎是不可能的，即使是今天最强大的计算机 [6], [8]。出于这个原因，研究人员没有使用穷举搜索算法，而是采用了启发式算法，如顺序向前选择 (SFS) [9]，顺序向后选择 (SBS) [10] 等。虽然这些算法可以显著降低完全搜索的时间复杂度，但由于它们在搜索过程中的一致策略，它们可能会陷入局部最优点 [6]。在这方面，近年来，研究人员使用另一类搜索算法，称为进化计算 (EC) 方法进行特征选择。进化计算算法通常具有良好的全局搜索能力，能够在搜索空间中找到全局最优或者较好的解。进化计算不依赖于线性关系，因此可以有效地捕捉非线性关系且具进化计算方法具有很高的灵活性，能够适应不同类型

的问题。虽然这些方法对于低维数据集的 FS 的高效率已经被证明，但是增加数据集中的特征数量将导致算法陷入局部最优点并经历早期收敛。原因是问题搜索空间的指数增长和问题搜索空间中添加了许多局部最优点 [8]。此外，计算量大，内存消耗大，分类效率低也是现存算法所面临的重大挑战。

因此，本文的目的是提供一个有效的 FS 算法的高维数据集，以减少 FS 在这样的数据集的问题。我们希望该算法能够达到最高的分类精度，具有更少的功能和更少的计算成本。

2 相关工作

近年来，研究人员广泛采用进化计算 (EC) 方法来进行高维数据集的特征选择。基于进化计算的特征选择方法通常将庞大的搜索空间划分为较小的子空间，并将进化计算方法应用于这些子空间。在众多 EC 方法中，粒子群优化 (PSO) 是其中受到最多关注的方法之一，其优势在于计算速度快、高效、易于实现、参数较少，并在特征选择中证明了其有效性 [11]。下文简要回顾了之前针对高维数据集提出的一些特征选择方法。

Tran et.al [12] 在 PSO 算法的基础上提出了一种变长粒子的 FS 算法。采用变长粒子的思想是通过使用较短长度的粒子来缩小高维数据集的搜索空间，以降低计算成本和内存消耗。使用基于过滤器的 FS 算法，称为对称不确定性 (SU) 方法，每个特征的秩被确定用于初始化种群。此外，提出了一种称为长度变化的机制，以避免粒子群算法的搜索空间的局部最优点。该方法的效率高度依赖于 SU 算法的效率和长度改变机制。

Song 等人 [13] 提出了一种基于可变大小合作协同进化 PSO 的高维数据集 FS 算法。在该算法中，每个特征的重要性是确定使用对称不确定性算法，以减少在高维数据的搜索空间的维数。在形成更小的子空间后，使用多群 PSO 算法搜索每个子空间。在搜索过程中，采用多种方法去除重复粒子，增加新粒子，以增加种群的多样性。最后，使用交叉方法，将每个子群得到的解合并，得到最终解。该算法存在一定的不足，包括不同群体的计算代价高，以及不同群体下相关特征之间缺乏相互作用而可能陷入局部最优。

Tan 等人 [3] 提出了一种称为 DimReM 的新方法，用于使用 EC 方法降低搜索空间的高维数。该方法旨在从数据集中物理删除一些不重要的特征，以减少问题搜索空间。由于 EC 方法的性质，如果群体中的最佳成员陷入局部最优点，则群体中的其他成员也可能陷入局部最优点。此外，由于减小搜索空间大小是缓慢的，因此实现期望的解需要高计算成本。

在 [11] 中提出了一种用于高维数据集的基于进化多任务的 FS 算法，称为 PSO-EMT。在该方法中，两个任务具有不同的搜索空间进行检查。第一个任务采用重要特征，其重要性由 ReliefF 算法 [5] 确定，第二个任务采用所有特征。该算法使用交叉算子来共享任务之间的信息。此外，还使用了另外两种机制来减少搜索空间和保持种群多样性。ReliefF 算法的性能较差，导致算法停在局部最优点。此外，因为所有的功能都存在

于第二个任务，一个高的计算成本花费在评估的 PSO 算法获得的解决方案。此外，使用多种群优化算法执行搜索过程具有高计算成本。

类似地，在 [14] 中提出了一种基于进化多任务的 FS 算法和一种用于高维数据的 PSO 算法，称为 MTPSO。该算法采用基于 ReliefF 算法的方法，将大搜索空间转化为多个小搜索空间的子任务。然后，使用多种群优化算法搜索每个任务的子空间。为了在搜索过程中实现更高的效率，使用一种称为知识转移的方法，在每个子集之间交换关于特征的重要信息。

Song 等人 [15] 提出了一种用于高维数据集的 FS 算法，称为 HFS-C-P。该算法由三个重要阶段组成。在第一阶段，对称不确定性算法用于去除不重要的特征。在第二阶段中，使用快速相关性引导的特征聚类方法对第一阶段中剩余的特征进行聚类。最后，在第三阶段，粒子群算法的粒子初始化使用的特征，并执行搜索过程，以获得一个最佳的子集。对称不确定性算法在第一阶段的性能不佳以及错误地去除一些相关特征会导致算法在某些数据集上的性能不佳。

Wang 等人 [16] 提出了一种称为 SaWDE 的 FS 算法，该算法基于加权差分进化算法并利用自适应机制。多种群机制增加了种群的多样性，多个子种群搜索问题搜索空间以找到最优的特征子集。两个自适应机制已被用来提高 DE 算法的效率，在不同的数据集的特征选择。提出了选择合适的变异算子和 DE 算法参数值的机制。此外，权重模型用于确定每个特征的重要性。他们提出的方法的主要缺点是计算成本高。

Cheng 等人 [17] 提出了一种基于转向矩阵的多目标进化算法，称为 SM-MOEA，用于高维数据集的 FS。在该算法中，使用基于搜索代理的转向矩阵计算特征的重要性。因此，它被用来引导人口走向更好的解决方案。此外，在该方法中，两个运营商，降维和个别修复，被用来删除不重要的功能，从搜索代理以及搜索空间的问题。该方法的缺点之一是从具有大量特征的数据集创建转向矩阵时的高内存消耗。此外，使用降维机制的相关和重要的特征的错误去除，可以导致该算法停止在局部最优点和算法的性能降低。

综上所述，以上方法的目标是通过采用进化计算方法在低维数据集中搜索来减小问题的搜索空间，并在高维数据集中达到期望的性能。然而，其中一些方法（如 [16] 和 [17]）存在内存消耗大、计算成本高的问题，而另一些方法（如 [15]）虽然计算成本较低，但由于使用基于滤波器的降维方法，可能会在某些数据集中陷入局部最优点，导致错误地过滤掉相关特征。这可能降低分类效率，尤其是当相关特征未正确识别时。总体而言，现有特征选择算法面临的挑战主要包括：高维搜索空间中算法陷入局部最优点的风险、大量计算和内存需求、以及由于滤波器方法引起的分类效率低下。因此，为应对这些挑战，本文提出了一种旨在在高维数据集中高效选择特征的新算法 SFE。

3 本文方法

3.1 本文创新点

3.1.1 创建搜索代理

EC 方法从高维数据集中选择特征的高计算成本和高内存消耗是因为将搜索代理存储在主存中，然后对搜索代理获得的解进行计算。使用搜索代理可以减少计算成本和内存消耗。FS 是一个二元问题，其中特征有两种模式：选择和非选择。在该算法中，搜索代理 X 是一个解决方案的 FS 问题，显示每个功能的模式。因此，代理 X 的表示如图 1 所示。

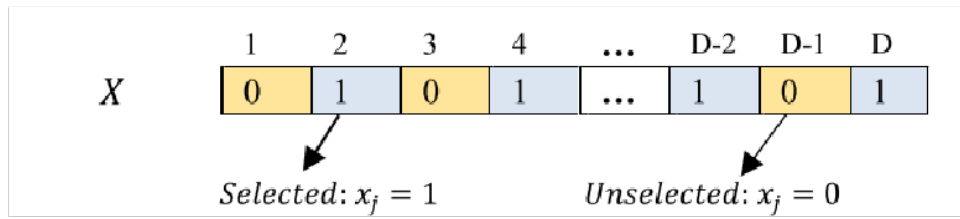


图 1 代理搜索的编码

3.1.2 提出了两种算子

上述文本提到的方法基于两个简单的算子：选择算子和非选择算子。这两个算子用于在搜索过程中改变特征的状态，从而影响搜索代理的行为。选择算子将特征的状态从非选择模式改变为选择模式，而非选择算子将特征的状态从选择模式改变为非选择模式。在 SFE (Selective Feature Evolution) 算法的搜索过程中，为了获得更优解，对搜索算子进行选择和非选择操作。选择算子的使用取决于问题搜索空间的条件和问题搜索空间中的位置。换句话说，在 SFE 算法的搜索过程中的每个阶段，都会应用非选择运算符，以便在搜索过程中获得更优的解。然而，在某些数据集中，选择运算符可能从搜索过程的一开始就被使用，而在其他一些数据集中，它可能直到搜索过程完成时才会被使用。

3.2 搜索过程

3.2.1 非选择操作

在高维数据集中存在大量不重要的特征。与此同时，特征选择的目标是用尽可能少的特征实现最高的分类精度。为了达到这一关键目标，有必要在整个问题的搜索空间中进行全局搜索，以识别不重要的特征并将其模式转换为非选定模式。为了在 SFE 算法中执行全局搜索，我们引入了非选择运算。在任何搜索过程中，该运算符通过不选择大

量不重要的特征，使搜索代理更接近问题的最佳点。应用非选择运算符时，需要计算应用该运算符的要素数量。对于不同的数据集，可以考虑不同的值。在 SFE 算法中，等式 (1) 用于确定非选择运算符的运算生成的伪随机数的数量。

$$UN = \lceil UR \cdot nvar \rceil \quad (1)$$

上述公式中， $nvar$ 是搜索空间的维数或数据集的特征数。 UR 是非选择运算符比率，可以是 0 到 1 之间的数字。具体的搜索过程如下图所示：首先，生成 UN 个随机数。然后，根据这些随机数对应的索引，将相应的特征状态由 1 变为 0。这一过程实现了对特征的非选择操作，使得在整个搜索空间中发生全局搜索，找到不重要的特征并将其模式转换为非选定模式。这有助于在探索阶段中更全面地搜索问题空间，提高算法的鲁棒性和性能。

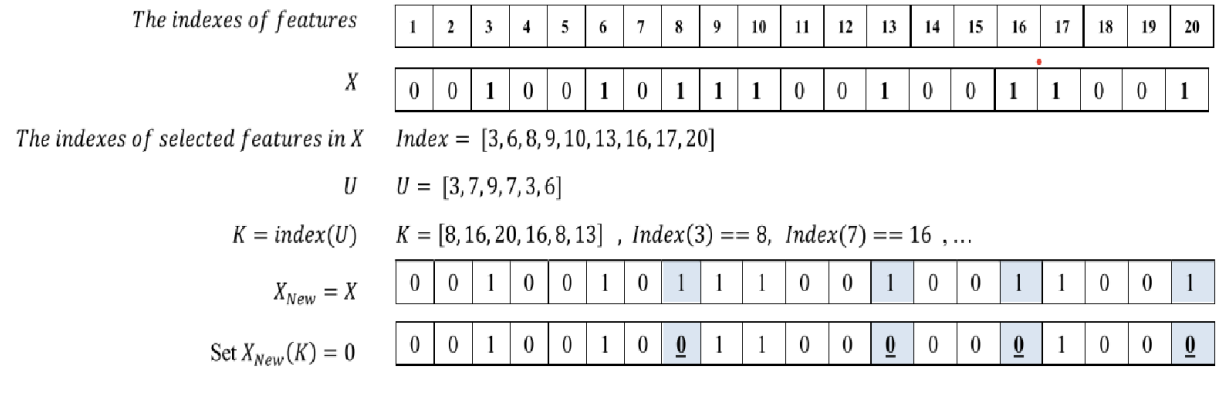


图 2 非选择操作的图示

3.2.2 平衡开发和探索

如果选择了较大的值作为 UR ，则在搜索过程的每个阶段，大量特征的状态都会被这个运算符改变为非选择模式。然而，在搜索的后期，每次优化都删除大量特征可能会阻碍局部优化的效果。为了在这个阶段实现更细粒度的优化，需要 UR 的值较小。为了满足这一要求，作者采用了一种线性递减的策略，逐渐减小 UR 的值。如果选择运算符的值在几个搜索阶段之后仍然较大，则在除了非选择运算符之外，还将使用选择运算符。因此，在搜索的初期，非选择算子用于全局搜索，算法处于探索阶段。在接下来的搜索步骤中，除了探索阶段之外，算法还执行开发阶段。为了在 SFE 算法的探索阶段和开发阶段之间创建适当的平衡，可以使用从大值开始逐渐减小为小值的线性减小的阈值。对于 SFE 算法，使用以下等式来确定当前 UR 值：

$$UR = UR_{\min} + \frac{(UR_{\max} - UR_{\min}) \times (MAX_Fes - Fes)}{MAX_Fes} \quad (2)$$

3.2.3 开发过程

当使用非选择运算符执行全局搜索后，可能导致数据集中的所有特征都被转换为非选择模式。在这种情况下，为了避免丢失先前搜索阶段的进展，我们需要进行一步回退，将当前特征子集还原为上一步的状态。然后，我们引入选择运算符，其核心目标在于引入随机性，从而有效地避免陷入局部最优解。通过在搜索过程中动态切换非选择和选择模式，算法能够在全局和局部搜索之间保持平衡，提高对问题空间的探索性，从而更为有效地解决复杂的优化问题。这一策略不仅维护了搜索的多样性，也增强了算法应对问题复杂性的适应能力。详细的过程如下图：

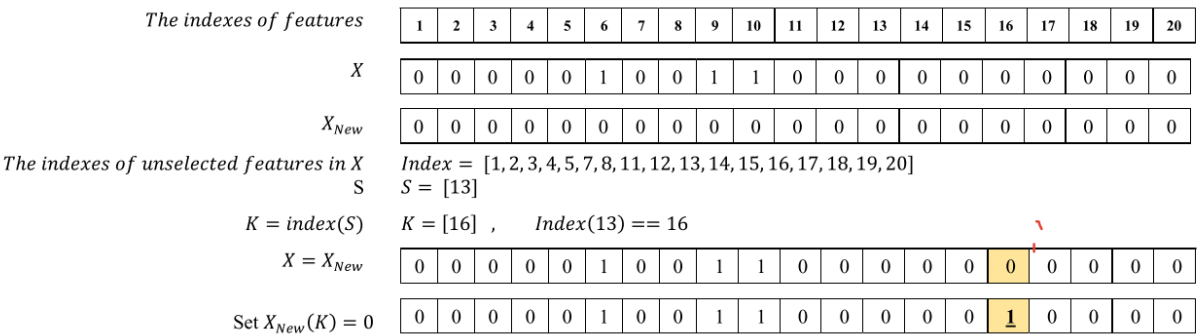


图 3 . 选择操作的图示

3.3 物理降维

分类精度用于评估在搜索过程的每个步骤中获得的新解：

$$\text{fit} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \times 100\%$$

(3)

基于上述评估函数，当我们得到分类精度最佳的解后，可以通过从数据集中实际移除不重要的特征来减少高维数据集中的问题搜索空间。具体的操作步骤如下图：

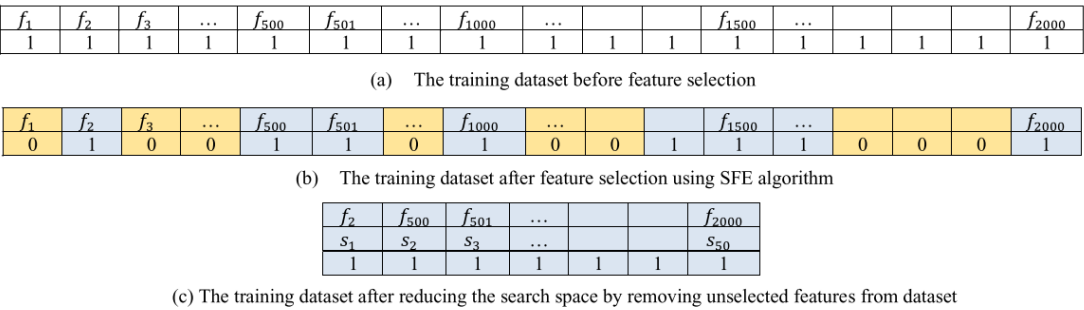


图 4 降维过程图示

3.4 混合 EC 算法框架

通过前述步骤，我们已成功缩减了问题的搜索空间，现在我们可以充分利用进化计算 (EC) 方法的强大潜力，这些方法在低维数据集的 FS 中的高效率已被证明 [18], [19] 所以我们结合进化算法为了取得更好地效果。

因此，混合算法的步骤流程可总结如下：

全局搜索阶段 (SFE 算法)：在整个庞大搜索空间中，初期采用自适应特征选择 (SFE) 算法进行全局搜索。SFE 算法通过对特征的动态选择，有助于找到初步的优秀解决方案。

搜索空间降维阶段：在完成 SFE 算法的全局搜索步骤后，通过实际移除未选择的模式特征，降低了搜索空间的维度。这个步骤有助于聚焦具有更高信息含量的特征。

局部优化阶段 (PSO 算法)：利用粒子群算法 (PSO) 对经过降维的搜索空间进行进一步的搜索。PSO 算法的使用旨在在更小、更关键的搜索空间中进一步优化解决方案，提高了算法的局部搜索能力。

这一混合算法的设计思路旨在充分发挥 SFE 和 PSO 两种算法的优势，通过逐步缩小搜索空间的维度，更加精准地定位问题的最优解。这样的流程有望有效克服高维问题所带来的挑战，提高算法的效率和性能。实验结果显示，SFE-PSO 混合算法在效率上明显优于独立使用的 SFE 算法。

4 改进思想

4.1 对 SFE 算法的改进

对 SFE 算法的改进思路涉及引入互信息的概念，以应对其随机选择带来的较大不确定性。这种改进旨在提高算法的准确性和效率，特别是在高维数据集的特征选择任务中，所以我们可以通过以下步骤改进 SFE 算法。

引入互信息概念：考虑互信息作为特征之间关系的度量，将高度相关的特征划分到同一组。互信息在这里的使用旨在量化特征之间的关联程度。互信息 (Mutual Information) 衡量了两个离散随机变量之间的相互依赖程度。对于两个离散随机变量 X 和 Y ，它们之间的互信息 $I(X; Y)$ 的定义如下：

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \cdot \log \left(\frac{P(x, y)}{P(x) \cdot P(y)} \right) \quad (4)$$

聚类操作：聚类操作的核心在于运用聚类方法，将具有高互信息的特征有机地归类到同一组。这一步骤的目标是揭示数据中的潜在关联性结构，使同一组内的特征更具相似性。通过形成更有序、结构化的特征组，我们为后续的分析 and 处理提供了更清晰、更可操作的特征集。

特征组内 SFE 操作：在形成的不同特征组内，特征组内的 SFE 操作进一步细化了特征的选择过程。通过在每个组内分别运行 SFE 算法，我们能够更有针对性地选择特

征，减小冗余，从而提高整体特征选择的效果。这种策略通过优化每个组内的特征选择，有望在高维数据集中实现更精准的特征组合，为进一步的分析和应用提供更强有力的支持。

通过以上改进，我们期望在 SFE 算法中引入互信息概念和特征聚类，以提高特征选择的精确度和效率。这种改进方法有望更好地应对高维数据集中特征选择中的不确定性和冗余性问题。

4.2 对 PSO 算法的改进

因为 SFE 已经很高效，快速的降维了，所以我们在改进 PSO 时，就需要增加其收敛性，和多样性，从而增加精度。所以我们进行如下改进有助于增强粒子群优化 (PSO) 算法的性能和适应性：

增加竞争因子以提高多样性：引入竞争因子是为了应对 PSO 算法可能陷入局部最优解的问题。当所有粒子朝向相同的方向移动时，容易导致整个群体陷入局部最优。通过增加竞争因子，引入了更多的随机性和多样性，使得粒子更有可能跳出局部最优解，更广泛地搜索整个空间。这种改进有助于提高 PSO 算法的全局搜索性能，减少陷入局部最优的风险。

切换到环形拓扑结构：将拓扑结构从全局切换到环形有助于改善信息传递和搜索效率。在全局拓扑结构中，每个粒子与所有其他粒子相连，信息传递过于密集，可能导致群体过早收敛。而在环形拓扑结构中，每个粒子只与其相邻的两个粒子通信，使得信息在整个群体中更为均匀地传递。这样的转变有助于平衡全局搜索和局部搜索，增加了算法对搜索空间的全面覆盖，提高了搜索的效率和鲁棒性。

综合来看，这些改进使得 PSO 算法更能够在复杂、高维的搜索空间中表现出更强大的搜索和优化能力。增加竞争因子和采用环形拓扑结构是在应对 PSO 的一些局限性的同时，提高其适用性和性能表现的有效手段。

5 实验级结果分析

5.1 实验环境搭建

使用的电脑 CPU 为 CPUIntel(R) Core(TM) i7-7700，GPU 为 NVIDIA GeForce GT 720，使用的操作系统为 Windows10，选择的编程语言是 matlab。

5.2 数据集上的性能提升

在 11_{Tumor} 数据集的性能提升对比如下：我们观察到，不论是在最差情况下、最好情况下的精度，还是平均精度方面，改进算法相对于原算法都呈现出一定程度的提升。

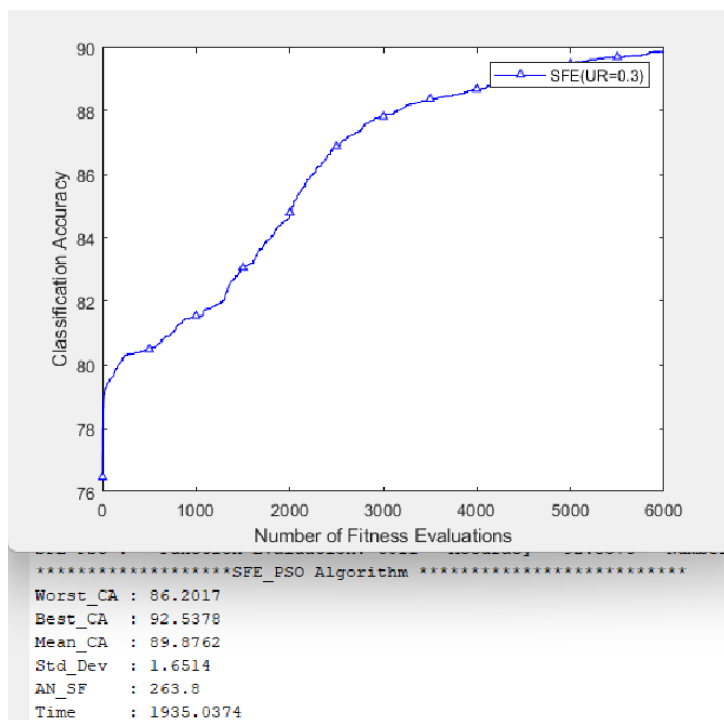


图 5 原算法的运行结果

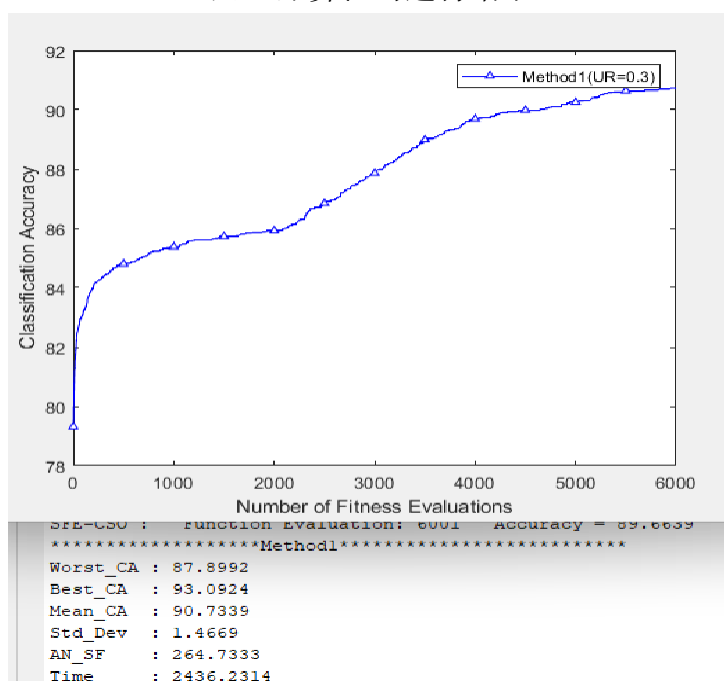


图 6 改进算法的运行结果

下表是原算法与改进算法在不同数据集上的精度对比，我们发现在大多数数据集上算法的精度都有所提升，尤其是当数据集比较大提升效果越明显。

表 1 改进算法与原算法的精度对比

	Method	SFE_PSO
9_Tumor	73.0182	72.8333
11_Tumors	90.7339	89.8762
Brain_Tumor_2	91.3333	90.4375
Brain_Tumor_1	94.0125	94.0238
Breast_Cancer	84.3123	83.6351
CLL_SUB_111	87.0123	87.0629
Colon	90.9615	90.1028
Lung_Cancer	97.9812	98.0932
nci9	73.4372	72.2586
Yale	75.2386	74.3283

6 总结与展望

综合以上对 SFE 和 PSO 算法的改进思路，我们通过引入互信息概念和特征聚类，以及增加 PSO 算法的竞争因子和切换拓扑结构的方式，尝试提高特征选择的准确度和效率。这些改进的目标在于更好地解决高维数据集中特征选择中的不确定性和冗余性问题，以及增强 PSO 算法的全局搜索和局部搜索性能。

虽然所提出的方法在实验中取得了显著的成功，但仍存在改进的空间。对于 SFE 算法，未来的研究可以考虑设计自适应参数机制，以适应不同数据集的特点。此外，非选择算子和选择算子的改进也是一个值得深入研究的方向，以进一步提高算法的性能。

对于 SFE-PSO 算法，未来的工作可以考虑使用更先进的 PSO 算法或其他进化计算方法，以提高算法的效率和性能。此外，还可以进一步探索 SFE 算法与其他进化计算方法的结合方式，以达到更高水平的性能。

总体而言，本研究为高维数据中的特征选择提供了一种有效的方法，并在 SFE 和 PSO 算法的改进方向上展示了可行性。未来的研究可以在这些方面进行更深入的优化和拓展，以适应更广泛的应用场景。

参考文献

- [1] Jie Cai, Jiawei Luo, Shulin Wang, and Sheng Yang. Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79, 2018.
- [2] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys*

(*CSUR*), 50(6):1–45, 2017.

- [3] Ping Tan, Xin Wang, and Yong Wang. Dimensionality reduction in evolutionary algorithms-based feature selection for motor imagery brain-computer interface. *Swarm and Evolutionary Computation*, 52:100597, 2020.
- [4] Hai Huang and Huan Liu. Feature selection for hierarchical classification via joint semantic and structural information of labels. *Knowledge-Based Systems*, 195:105655, 2020.
- [5] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53:23–69, 2003.
- [6] Bach Hoai Nguyen, Bing Xue, and Mengjie Zhang. A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation*, 54:100663, 2020.
- [7] Mohamed Abdel-Basset, Doaa El-Shahat, Ibrahim El-Henawy, Victor Hugo C De Albuquerque, and Seyedali Mirjalili. A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Systems with Applications*, 139:112824, 2020.
- [8] Yu Xue, Bing Xue, and Mengjie Zhang. Self-adaptive particle swarm optimization for large-scale feature selection in classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(5):1–27, 2019.
- [9] A Wayne Whitney. A direct method of nonparametric measurement selection. *IEEE transactions on computers*, 100(9):1100–1103, 1971.
- [10] Thomas Marill and D Green. On the effectiveness of receptors in recognition systems. *IEEE transactions on Information Theory*, 9(1):11–17, 1963.
- [11] Ke Chen, Bing Xue, Mengjie Zhang, and Fengyu Zhou. An evolutionary multitasking-based feature selection method for high-dimensional classification. *IEEE Transactions on Cybernetics*, 52(7):7172–7186, 2020.
- [12] Binh Tran, Bing Xue, and Mengjie Zhang. Variable-length particle swarm optimization for feature selection on high-dimensional classification. *IEEE Transactions on Evolutionary Computation*, 23(3):473–487, 2018.
- [13] Xian-Fang Song, Yong Zhang, Yi-Nan Guo, Xiao-Yan Sun, and Yong-Li Wang. Variable-size cooperative coevolutionary particle swarm optimization for feature se-

- lection on high-dimensional data. *IEEE Transactions on Evolutionary Computation*, 24(5):882–895, 2020.
- [14] Ke Chen, Bing Xue, Mengjie Zhang, and Fengyu Zhou. Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 26(3):446–460, 2021.
- [15] Xian-Fang Song, Yong Zhang, Dun-Wei Gong, and Xiao-Zhi Gao. A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data. *IEEE Transactions on Cybernetics*, 52(9):9573–9586, 2021.
- [16] Xubin Wang, Yunhe Wang, Ka-Chun Wong, and Xiangtao Li. A self-adaptive weighted differential evolution approach for large-scale feature selection. *Knowledge-Based Systems*, 235:107633, 2022.
- [17] Fan Cheng, Feixiang Chu, Yi Xu, and Lei Zhang. A steering-matrix-based multiobjective evolutionary algorithm for high-dimensional feature selection. *IEEE transactions on cybernetics*, 52(9):9695–9708, 2021.
- [18] Mohammad Tubishat, Norisma Idris, Liyana Shuib, Mohammad AM Abushariah, and Seyedali Mirjalili. Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Systems with Applications*, 145:113122, 2020.
- [19] Pei Hu, Jeng-Shyang Pan, and Shu-Chuan Chu. Improved binary grey wolf optimizer and its application for feature selection. *Knowledge-Based Systems*, 195:105746, 2020.