本发明公开了一种自组织映射的权重粒子群均值聚类算法—SOM&WPK（Self Organizing Feature Maps and Weight Particle K-means）,该方法弥补了传统K-means聚类算法难以确定聚类中心，划分样本困难以及易陷入局部最优的缺点。本发明首先使用SOM对输入层的样本向量通过竞争层学习样本分布，实现不同特征的样本聚类划分，经过多轮迭代直到聚类结果不再改变，我们计算每个聚类簇的聚类均值暂时作为聚类中心，借鉴近邻思想将聚类均值映射到最近的样本，该样本作为该类簇的聚类中心，同时避免了聚类中心的类标分配问题；接着使用WPSO（Weight Particle Swarm Optimization）对SOM的粗聚类结果进行细聚类，权重粒子群算法（WPSO）的粒子位置由样本权重表示，聚类中心作为“食物”，粒子与所属聚类的中心的均方差作为适应度函数，每经过WPSO优化都进行一次K-means算法进行聚类中心和样本的重新选择和划分，直到适应度最小时停止迭代，计算最终的聚类结果准确率。该发明可以有效提高聚类性能。



1. 一种自组织映射的权重粒子群均值聚类算法，其特征在于，包括：

数据预处理模块，用于将原始文本处理成聚类算法能够识别的向量形式。

SOM粗聚类模块，用于输入向量经过竞争学习获得获胜神经元，输入层与竞争层的神经元通过权值连接，调整获胜神经元及相邻神经元的连接权值，使样本分布最终趋于稳定的聚类划分。

WPK细聚类模块，用于对SOM的粗聚类结果进行细聚类划分，利用SOM的聚类结果初始化WPSO算法，通过聚类中心对粒子的“牵引”改变粒子位置，使粒子向聚类中心的位置移动，每次迭代粒子都向聚类中心的位置移动，通过不断迭代直至适应度值及聚类结果不再变化。

聚类结果输出模块，用于对最终的聚类结果作可视化处理，将不同类内的样本统计，计算样本聚类的准确率，刻画样本迭代过程中的适应度曲线，使用Davies-Bouldin index、Dunn's index和Silhouette coefficient等评价指标来衡量算法好坏。

2. 如权利要求1所述的自组织映射的权重粒子群均值聚类算法，其特征在于，包括：

步骤S1：本发明的数据集使用UCI的公开数据Iris、Wine、Glass，三个数据集的样本都是向量形式。Iris数据集包含150个数据样本，分为3类，每类50个数据，每个数据包含4个属性，通过花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类。Wine数据集包含来自3种不同起源的葡萄酒的共178条记录，13个属性是葡萄酒的13种化学成分，通过化学分析可以来推断葡萄酒的起源。Glass数据集包含214个数据样本，分为3类，每个数据包含9个属性。

步骤S2：对样本使用SOM粗聚类，SOM的输入层和竞争层神经元全部归一化处理：

寻找获胜神经元：

输入一个模式时，竞争层的所有神经元对应的权值与输入模式进行相似性比较，将相似性最大的权向量判为获胜神经元。通过对获胜神经元及其周围的兴奋神经元的权值进行调节，以增加它们对输入信号的判别函数值，随着权值的不断调整，获胜神经元对与相似的输入信号会有更强的响应。

步骤S3：使用SOM的聚类结果初始化WPK算法，WPK算法是由WPSO和K-means算法组合而成，通过使用SOM的聚类中心和聚类划分初始化WPSO算法，然后类似于传统PSO算法进行迭代，只不过WPSO的每个粒子是由每个样本取代的，初始化的每个粒子位置是不同的，由聚类中心牵引直到适应度值最小时停止迭代。

步骤S4：统计最终的聚类结果，以样本类标的众数来给划分的类标记，计算每个类的准确率，使用不同的评价指标来评价算法的优劣。

3. 如权利要求2所述的WPK细聚类模块，其特征在于，包括：

WPK模块是由WPSO和K-means算法组合而成，WPSO的每个粒子的位置是由所使用的数据样本代替的，粒子的适应度函数为每个粒子与该粒子所属簇的类中心的欧式距离，粒子编码方式如下：

|  |  |  |
| --- | --- | --- |
| *x1,x2,x3,……,xn* | *v1,v2,v3,……,vn* | *fitness(x,c)* |

我们的目标是寻找最优聚类中心，而不是粒子的最优位置，粒子的适应度函数如下：

粒子的适应度值表示各个类内数据对象之间的相似度，适应度值越小则表明类内数据对象的结合程度越紧密，聚类效果越好。WPSO更新完粒子位置和速度后计算后，进行K-means聚类，计算聚类均值，重新聚类划分，利用K-medoids思想寻找均值最近的样本点作为聚类中心，大幅度降低噪音点对算法的影响，也减少了空簇产生的概率，加快了算法收敛的速度。

**自组织映射的权重粒子群均值聚类算法**

**技术领域**

本发明属于数据挖掘领域，特别是在无监督学习中对未标记的数据样本进行聚类划分。

**背景技术**

近年来，大数据和人工智能备受追捧，热度空前，信息也呈现出爆炸式的增长，我们面对的将是海量的文本、视频、图片和音频数据，如何从规模庞大的数据中挖掘出蕴含现实价值的信息也逐渐变成计算机研究领域的重要课题。传统的数据挖掘，主要依靠人工经验和团队协作来完成，但是这这种方法无疑增加了时间和人力成本，甚至最终的结果不尽人意。因此，数据挖掘技术随之诞生，旨在帮助人们从海量无序的数据中提取潜在的、有价值的信息。聚类算法作为数据挖掘领域的重要分支，在许多领域都得到了广泛应用，包括机器学习、模式识别、图像分析、信息检索、计算机视觉等，高效的聚类算法能够提高工作效率，改善工作质量。

聚类算法是根据样本特征相似性将样本划分成不同分布的聚类簇，类内的样本具有较高的文本特征相似性，类间的样本分布距离较远，特征差异较大。作为数据挖掘的分支，聚类分析已被研究多年，大量聚类算法涌现而出，但由于不同的数据拥有自身的形式和维度，没有一个算法对数据具有普适性，各种算法都是针对特定的数据采取特定的聚类方案。例如，一些聚类算法对中低维度数据有显著的聚类结果，但在高维数据上表现不佳；一些聚类算法只能针对特殊分布结构的数据而不能很好处理其它分布的数据。这些特点要求算法具有可伸缩性，能处理不同类型的数据，还要可以发现各种形状的簇，解决“噪音”和孤立点问题。传统的聚类算法，已不能解决上述问题，有学者将群智能优化算法应用到传统算法中，发现基于群智能优化的聚类算法比传统算法具有更好的聚类效果。

基于划分的聚类算法是将样本集按照距离规则划分为多个不相交的簇，通过迭代直到目标函数最小停止聚类划分。该方法易于实现，收敛速度快，但其复杂度与样本规模、样本维度和聚类中心呈线性关系。由MacQueen提出的K均值算法是一种经典的基于划分的聚类算法，该算法集计算简单和快速收敛于一身，采用距离作为样本相似性的评价指标。但是K均值聚类算法存在固有缺点：（1）聚类簇数K的确定，根据什么指标选择K值会直接影响聚类准确率（2）初始聚类中心的选择，聚类效果依赖于聚类中心的初始化。融合的聚类算法有时候会比单一算法的准确率和收敛度上更好，一些学者也对聚类算法进行了组合，这个被称为聚类继承的过程能够在不同邻域和数据提供更强的鲁棒性和稳定性的解决方法。有学者整合粒子群优化和K均值的数据聚类算法，结合两者优点改善了聚类算法的质量，但是依然没有解决K值的问题。

综上所述，现有的技术问题为：

传统聚类算法难以确定聚类簇数，聚类簇数的选择会直接影响聚类结果的准确性，聚类簇数选择过大，则会导致产生聚类结果冗余，有可能一个类的样本被划分为两个类，或者两个不同的类的样本交叉划分；初始聚类中心的选择也会影响最终的聚类结果，若初始聚类中心选择过于密集，则会导致聚类划分中产生多个类的样本交叉，不能很好的给划分的类标记，若初始聚类中心选择距离过远，则有可能选择的中心点是边缘点或者异常点，导致聚类结果无参考价值，所以聚类中心和聚类簇数的选择要合理。组合的聚类算法一般存在易陷入局部最优的缺点，若达到好的适应度值则会停止聚类算法的迭代进行，从而不能寻找全局最优。

技术问题不能有效解决的原因：

不同的聚类算法都是针对特定的数据而适用的，没有一种聚类算法可以适用于所有数据，由于属于无监督学习，词向量的训练直接影响到聚类结果，若没有好的语料，聚类算法将变得毫无意义。除此之外，好的聚类算法能有效对样本分布进行划分，使最终的聚类结果变得有意义。

解决的难度在于：

由于K-均值的目标函数不是凸函数，可能包含许多局部最小值，因此，K-均值算法主要存在两个固有缺点：（1）对于随机的初始值选取可能会导致不同的聚类结果，甚至存在着无解的情况；（2）该算法是基于目标函数的算法，通常采用梯度法求解极值，由于梯度法的搜索方向是沿着能量减少的方向进行，使得算法很容易陷入局部极值。这两大缺陷大大限制了它的应用范围。

解决的意义在于：

在分析K-均值聚类算法存在不足的基础上，本发明提出了一种新的聚类算法，实验结果证明，该算法有很好的全局收敛性，不仅有效地克服了传统的K-均值算法易陷入局部极小值和对初始值敏感的问题，而且具有较快的收敛速度。

**发明内容**

本发明提出一种自组织映射的粒子群均值聚类算法，解决了传统聚类算法陷入局部极小值和对初始值敏感的问题，对无标记的样本能够实现很好的聚类效果，提高了对样本的适用性。

本发明所采用的方案是，自组织映射的粒子群均值聚类算法，包括：

数据预处理模块，用于将原始文本处理成聚类算法能够识别的向量形式。

SOM粗聚类模块，用于输入向量经过竞争学习获得获胜神经元，输入层与竞争层的神经元通过权值连接，调整获胜神经元及相邻神经元的连接权值，使样本分布最终趋于稳定的聚类划分。

WPK细聚类模块，用于对SOM的粗聚类结果进行细聚类划分，利用SOM的聚类结果初始化WPSO算法，通过聚类中心对粒子的“牵引”改变粒子位置，使粒子向聚类中心的位置移动，每次迭代粒子都向聚类中心的位置移动，通过不断迭代直至适应度值及聚类结果不再变化。

聚类结果输出模块，用于对最终的聚类结果作可视化处理，将不同类内的样本统计，计算样本聚类的准确率，刻画样本迭代过程中的适应度曲线，使用Davies-Bouldin index、Dunn's index和Silhouette coefficient等评价指标来衡量算法好坏。

本发明所采用的另一种技术方案是，自组织映射的粒子群均值聚类算法，具体如下：

步骤S1：本发明的数据集使用UCI的公开数据Iris、Wine、Glass，三个数据集的样本都是向量形式。Iris数据集包含150个数据样本，分为3类，每类50个数据，每个数据包含4个属性，通过花萼长度，花萼宽度，花瓣长度，花瓣宽度4个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类。。Wine数据集包含来自3种不同起源的葡萄酒的共178条记录，13个属性是葡萄酒的13种化学成分，通过化学分析可以来推断葡萄酒的起源。Glass数据集包含214个数据样本，分为3类，每个数据包含9个属性。

步骤S2：对样本使用SOM粗聚类，SOM的输入层和竞争层神经元全部归一化处理：

寻找获胜神经元：

输入一个模式时，竞争层的所有神经元对应的权值与输入模式进行相似性比较，将相似性最大的权向量判为获胜神经元。通过对获胜神经元及其周围的兴奋神经元的权值进行调节，以增加它们对输入信号的判别函数值，随着权值的不断调整，获胜神经元对与相似的输入信号会有更强的响应。

步骤S3：使用SOM的聚类结果初始化WPK算法，WPK算法是由WPSO和K-means算法组合而成，通过使用SOM的聚类中心和聚类划分初始化WPSO算法，然后类似于传统PSO算法进行迭代，只不过WPSO的每个粒子是由每个样本取代的，初始化的每个粒子位置是不同的，由聚类中心牵引直到适应度值最小时停止迭代。

步骤S4：统计最终的聚类结果，以样本类标的众数来给划分的类标记，计算每个类的准确率，使用不同的评价指标来评价算法的优劣。

进一步的，所述步骤S2包括：

自组织映射网络（Self Organization Map，SOM）是由芬兰Helsinki大学的Kohonen T.教授在1981年提出的一种竞争型网络，该网络是通过模拟人类大脑皮层对信号的自组织映射特性得到的。SOM的主要目的是将任意维度的输入信号通过计算映射转变为一维或者二维的离散映射，并且以自适应的方式实现这个过程。当外部信号输入时，SOM网络会经过竞争学习决定哪个信号得到处理，这个过程时自组织完成的。

SOM神经网络采用无监督的学习方式进行迭代训练，除了能够别样本分布的拓扑结构，还能够随着输入模式（刺激）的变化选择性的进行权值调整。SOM网络将任意维度的输入模式映射到低维空间，既降低了向量维度，又减轻了迭代训练的计算复杂度，同时也保持着样本的原始拓扑结构。

典型的SOM网络结构有两层构成：输入层和竞争层。输入层主要负责外界信息的接收，输入层的每个神经元都与竞争层的神经元进行权值连接，然后将外界信息传递到竞争层；竞争层主要负责输入信息的分析，通过竞争学习获取获胜神经元，同时抑制近邻神经元的兴奋。图1是二维SOM网络模型图。

所述步骤S3包括：

粒子群优化（Particle Swarm Optimization，PSO）算法是在1995年由Kennedy和Eberhart提出的迭代智能优化算法，该算法源于对鸟群觅食行为的模拟，通过个体的简单行为，以及个体的协作和信息交互实现问题的最优解。PSO的优势在于操作简单，不需要设置过多的参数，收敛速度快等方面。

在PSO算法中，粒子群的每个粒子都代表N维空间的一个可行解，所有粒子都具有两个属性：位置和速度。位置代表粒子移动的方向，速度代表粒子移动的快慢，在每次迭代中，粒子根据自己的历史最佳位置和种群最优位置决定自己下次更新的位置和速度。粒子的位置和速度更新公式如下：

速度：

位置：

上述两式中，（t+1）表示第i个粒子在t+1时刻的速度，pbest代表个体粒子经过的历史最佳位置，gbest代表粒子群的历史最佳位置，c1、c2代表学习因子，一般取值为2，w代表惯性因子，r1、r2为两个（0,1）之间的随机数。

1967年，MacQueen提出了一种基于划分的经典聚类算法—K-means算法，K-means算法计算简单，收敛速度快。该算法随机选择K个样本点作为初始聚类中心，然后根据近邻原则将其它样本划分到离K个样本最近的簇中，每次迭代完成后，重新计算聚类中心，也就是聚类簇中所有样本的均值。该算法停止的条件是，当然数据集所有的样本分配的距其最近的簇不在发生变化时，就停止分配。

本发明研究了传统的PSO聚类算法，发现PSO聚类算法旨在优化聚类中心，聚类中心虽然在移动，但是却忽略了样本权值的优化，本文在PSO聚类算法的基础上，在优化聚类中心的同时又优化样本权值，使样本权值朝着最近邻聚类中心靠近，粒子根据自身经验寻找最优位置。每次迭代重新划分类簇和计算准确率或适应度值，我们设置一个比较次数的阈值，比较准确率或适应度值在阈值范围内有没有达到最优，若达到最优，则后期迭代都是此刻粒子位置，算法收敛，若没有达到最优则继续迭代。

WPSO算法仍需要设定初始聚类中心和聚类簇数，传统的聚类簇数由人工选择，聚类簇数是个很棘手的问题，而初始聚类中心的选择往往是随机选择或基于密度、距离选择，甚至初始聚类中心可能是孤立点、边界点，聚类算法极易陷入局部最优，甚至出现空簇问题。SOM算法具有降维的作用，能有效处理孤立点问题，不含复杂的求导、积分等操作，但SOM算法也存在训练时间长，竞争学习可能出现“死神经元”等缺点。本文将融合SOM、WPSO和K-means算法，在缺点上进行互补，优势上进行加强。

为避免SOM训练时间过长，SOM首先对数据进行粗聚类，使数据的聚类情况处于不饱和状态，SOM不断迭代，网络会学习到数据分布情况，待达到迭代次数后，SOM网络会返回迭代训练的权值，这个权值是通过对数据的竞争学习得到的，然后将原始数据与权值进行分析比较，会得到获胜神经元。

假设训练集，i=1,2,3,……,m，每个样本的维度是n，竞争层采用矩阵神经元阵列形式，输出矩阵大小为n\*k，每个竞争层神经元的维度为k（j=1,2,……,n）。SOM算法步骤如下：



Step 1：初始化，设定输入数据集、网络迭代次数iteration、竞争层矩阵大小n\*k，随机生成竞争层神经元的权值。



Step 2：竞争过程，归一化输入样本和神经元权值，计算归一化输入样本和竞争层神经元的内积，内积最大的维度的索引为获胜神经元索引winner。



样本向量和权值归一化：



Step 3：定义邻域函数\*(t)，优胜邻域随着时间的减少逐渐收缩。优胜邻域用邻域半径表示：



Step 4：调整权值，更新获胜神经元的值以及优胜邻域的权值。

式（8）中σ(t,N)代表学习率，学习率也是随时间减少的。

Step 5：循环迭代，若达到迭代次数iteration，返回网络训练的权值w，若未达到迭代次数，重复Step 2，Step 3，Step4。

Step 6：输出最终聚类划分和聚类中心。计算原始数据和训练权值的内积，内积最大的位置为获胜索引，将相同的获胜索引规划为一类，然后使用K-means计算聚类中心。

我们在PSO聚类算法中设置了聚类中心，使用SOM竞争学习到的聚类中心初始化PSO聚类中心，样本的数量即为粒子数量，粒子位置即为样本权值，粒子的适应度函数为每个粒子与该粒子所属簇的类中心的欧式距离，我们的目标是寻找最优聚类中心，而不是粒子的最优位置，PSO更新完权值和速度后进行K-means聚类，计算聚类均值，重新聚类划分，利用K-medoids思想寻找均值最近的样本点作为聚类中心，大幅度降低噪音点对算法的影响，也减少了空簇产生的概率，加快了算法收敛的速度。

WPK的算法框架如下：

Step 1：种群初始化：使用SOM获取的聚类中心初始化PSO算法设置的聚类中心，以样本权值初始化粒子位置，然后计算粒子的适应度值，从而选择出个体的局部最优位置向量和种群的全局最优位置向量Gbest。

Step 2：迭代设置：设置迭代次数gmax，并令当前迭代次数g=1，设置适应度对比比较次数的阈值；

Step 3：速度更新：根据公式（1）更新每个粒子的速度向量；

Step 4：位置更新：根据公式（2）更新每个粒子的位置向量；

Step 5：局部位置向量和全局位置向量更新：更新每个个体的和种群的Gbest；

Step 6：利用K-means算法重新进行聚类划分，计算新的聚类均值，通过K-medoids寻找近邻样本作为新的聚类中心。

K-means算法步骤如下所示：

input：包含n个数据对象的数据集合D以及聚类簇的个数K。

output：满足聚类准则函数收敛的k个聚类簇的集合。

从数据集D中随机选择k个样本作为初始聚类中心，j=1,2,3,……,k。

计算数据集的每个样本与k个聚类中心的距离distance（，），i=1,2,3,……,n 。

按照最近邻原则划分样本到最近的类中，即满足distance（，）=min{ distance（，），j=1，2，3，……，k}，则ϵ。

重新计算聚类中心：

计算类内聚：

反复执行（2）、（3）、（4）、（5）步，直到聚类中心不在变化为止。

Step 7：终止条件判断：判断迭代次数时都达到gmax，如果满足，输出Gbest；否则继续进行迭代，跳转至Step 3。

SOM&WPK整个算法步骤和流程图如下所示：

步骤一：初始化SOM的迭代次数和网络神经元的矩阵大小，经过迭代获得最终的聚类划分。

步骤二：基于SOM聚类划分使用K-means算法求出聚类簇的均值，采用K中心点思想求离均值最近的点作为聚类中心。

步骤三：使用SOM的聚类中心初始化WPSO的粒子位置，每个样本代表一个粒子，样本数量即为粒子群规模，样本权值即为粒子位置。

步骤四：WPSO聚类算法开始迭代，计算适应度值，适应度值的计算是每个样本与非样本类中心的欧式距离的倒数，每次迭代都要经过K-means算法的重新聚类得到聚类划分和聚类簇均值，使用PAM获得映射到样本上的聚类中心。

步骤五：判断是否达到迭代次数，否则，重复步骤三、步骤四。

算法流程图如图2所示。

传统的PSO聚类算法优化的总是聚类中心，本文通过优化样本权值，使用样本权值代替粒子位置，使用聚类中心作为“食物”，使粒子趋向聚类中心的方向进行优化，待达到最高准确率和最佳适应度值时算法收敛。

本文的粒子群算法采用基于样本权值的编码格式，即每个粒子的位置由每个样本权值代表，粒子除了位置之外，还有速度、适应度值，速度的维度与样本维度相同。粒子编码方式如下：

|  |  |  |
| --- | --- | --- |
| x1,x2,x3,……,xn | v1,v2,v3,……,vn | fitness(x,c) |

适应度函数的选择直接影响了聚类算法的收敛速度和能否找到最优解，对聚类划分有一个整体的认识以及对聚类结果正确率的判断，将WPSO算法引入K-means算法中，可将评价聚类质量的准则函数作为粒子群的适应度函数。选用类内紧密程度MSE来表示聚类质量的好坏，MSE越小，聚类效果越好。

粒子的适应度值表示各个类内数据对象之间的相似度，适应度值越小则表明类内数据对象的结合程度越紧密，聚类效果越好。适应度函数可以表示为：

虽然粒子的飞行方向受聚类中心的牵引，但是每次迭代完成无法落到具体的样本点上，这就增加了聚类中心选择的难度。为降低噪音点的影响，本文借助K-中心点的思想，在每次K-means聚类完成后，选择与聚类均值最近的样本点作为聚类中心，不仅加快了收敛速度，还可以防止空簇的发生。

此公式表示在第t次迭代过程中，第i个聚类平均值与所有样本的距离，距离最近的那个样本的索引即为聚类中心索引。

为了验证算法的有效性和可行性，本文采用UCI的Iris数据集、Wine数据集和Glass数据集进行实验，为验证本文算法的可行性，本文采用purity、适应度值、收敛度曲线、Davies-Bouldin index、Dunn's index和Silhouette coefficient作为聚类结果评判标准。

为验证算法的聚类纯度，本文将集成算法拆分成单一算法模型SOM、PSO/K-means、K-means与本文算法在纯度上进行比较，在每组数据集上重复实验20次，取准确率最高值进行比较，如图3所示。

除此之外，本发明与PSO&K-means算法进行了对比，如表1、表2所示。

实验还将从Davies-Bouldin index、Dunn's index和Silhouette coefficient方面进行比较。DB越小意味着类内距离越小，同时类间距离越大，即DB越小证明聚类效果越好。DI越大意味着类内距离越小，类间距离越大，聚类效果越好。轮廓系数Silhouette coefficient的值是介于 [−1,1] ，越趋近于1代表内聚度和分离度都相对较优。对比如表3，4、5所示。

本文为验证算法的收敛度，将与pso-km算法在Iris数据集上进行对比，如图4、图5所示。

表1 聚类准确率比较

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | 准确率 | K-means | pso-km | SOM&WPK |
|  | 最高 | 89.33 | 89.33 | 99.33 |
| Iris | 最低 | 51.33 | 88.67 | 90.67 |
|  | 平均 | 80.17 | 89.13 | 96.08 |
|  | 最高 | 70.79 | 70.79 | 99.44 |
| Wine | 最低 | 51.69 | 70.79 | 79.77 |
|  | 平均 | 68.88 | 70.79 | 87.85 |
|  | 最高 | 57.47 | 55.14 | 86.91 |
| Glass | 最低 | 48.59 | 35.51 | 48.59 |
|  | 平均 | 44.93 | 47.13 | 64.95 |

表2 适应度值对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | 适应度值 | K-means | pso-km | SOM&WPK |
|  | 最高 | 123.8498 | 97.6575 | 101.9808 |
| Iris | 最低 | 97.3259 | 97.2221 | 97.2725 |
|  | 平均 | 103.7300 | 97.3901 | 100.9184 |
|  | 最高 | 18776.94 | 16960.21 | 16932.2505 |
| Wine | 最低 | 16960.20 | 16940.28 | 16704.7616 |
|  | 平均 | 17067.07 | 16944.78 | 16770.1669 |
|  | 最高 | 215.7317 | 227.9053 | 238.7814 |
| Glass | 最低 | 213.4705 | 222.3460 | 233.2041 |
|  | 平均 | 214.9835 | 224.3496 | 235.9792 |

表3 Iris数据集比较

|  |  |  |  |
| --- | --- | --- | --- |
|  | K-means | pso-km | SOM&WPK |
| DBI | 0.8116 | 0.6937 | 0.6680 |
| DVI | 04347 | 2.6776 | 2.8772 |
| SC | 0.4876 | 0.4927 | 0.4999 |

表4 Wine数据集比较

|  |  |  |  |
| --- | --- | --- | --- |
|  | K-means | pso-km | SOM-WPSO |
| DBI | 0.5479 | 0.5313 | 0.5283 |
| DVI | 1.9032 | 1.9370 | 2.0675 |
| SC | 0.5883 | 0.5921 | 0.5611 |

表5 Glass数据集比较

|  |  |  |  |
| --- | --- | --- | --- |
|  | K-means | pso-km | SOM-WPSO |
| DBI | 1.0989 | 0.9246 | 1.1960 |
| DVI | 0.5741 | 0.7507 | 0.7270 |
| SC | 0.5180 | 0.6846 | 0.2564 |



图1 二维SOM网络模型图



图2 算法流程图

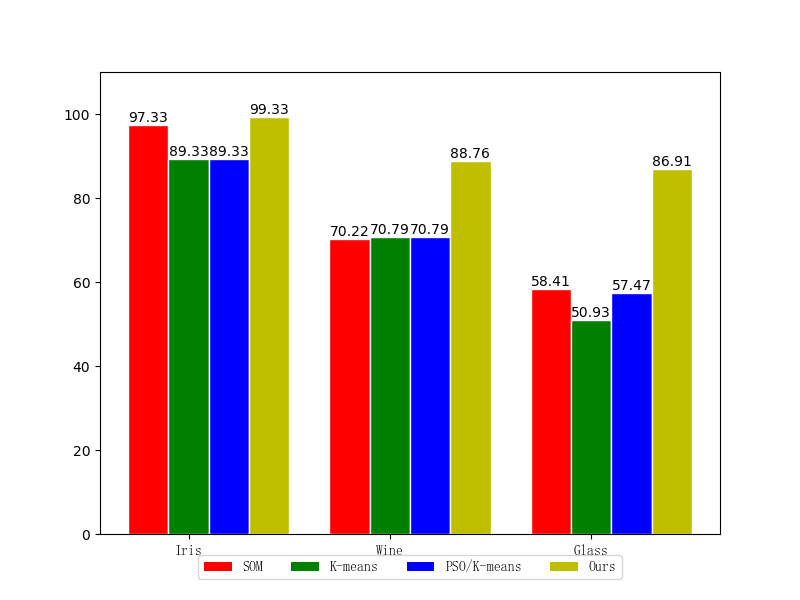
****

图3 算法纯度对比

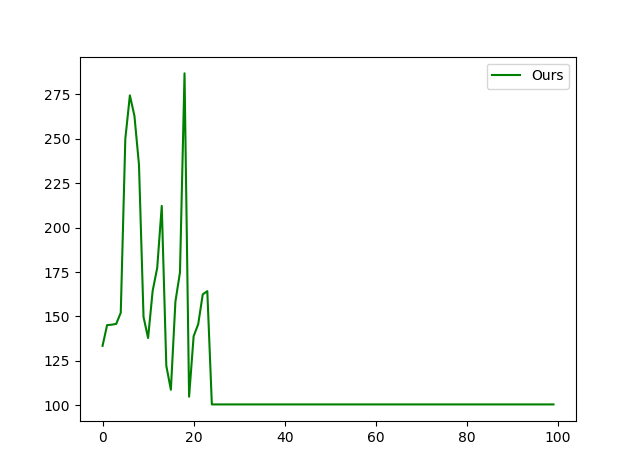


图4 算法收敛图

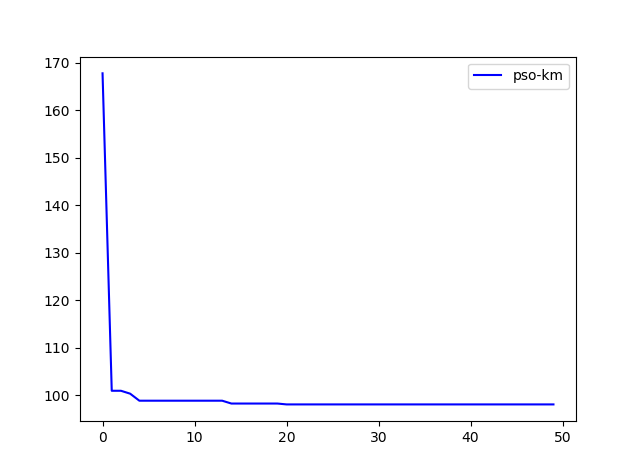


图5 pso-km收敛图