Evolutionary Spanning Clustering (ESC)

Soroush Oskouei

Abstract

This paper introduces the Evolutionary Spanning Clustering (ESC) algorithm, a novel approach to clustering data using a genetic algorithm in order to find the minimum number of spanning trees that would cover all data points within a specified distance threshold. ESC uses genetic algorithm to optimize the clustering by constructing spanning forests that respect the distance constraints, thereby forming naturally separated clusters. The effectiveness of ESC through experiments on synthetic datasets has been established, highlighting its ability to handle complex clustering tasks where traditional methods might fail, provided that a reasonable distance threshold is given.

1 Introduction

Clustering is a critical task in data analysis and machine learning, aiming to partition similar data points into distinct clusters. Conventional clustering algorithms, such as K-means and hierarchical clustering, typically depend on predefined distance metrics and specific assumptions regarding data distribution. In contrast, the Evolutionary Spanning Clustering (ESC) algorithm employs a genetic algorithm to determine the minimum number of spanning trees necessary to encompass the dataset. This approach offers a more flexible and adaptive method for clustering.

The ESC algorithm addresses the limitations of conventional methods by incorporating a distance threshold D, which ensures that each edge in the spanning trees does not exceed this threshold. This constraint allows ESC to form clusters based on the natural structure of the data, rather than imposing artificial boundaries.

2 Methodology

The ESC algorithm operates by evolving a population of candidate spanning forests, where each forest is a collection of spanning trees that cover all data points. The algorithm seeks to minimize the number of spanning trees, thereby optimizing the clustering solution.

2.1 Problem Formulation

Given a dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ and a distance threshold D, the goal is to partition the dataset into k clusters such that:

(a) Each cluster forms a connected component in a spanning forest.

(b) The maximum distance between any two connected points in a cluster does not exceed D.

(c) The number of clusters k is minimized.

The problem can be formulated as an optimization problem:

$$\min_{F} \quad k(F)$$

subject to:

$$d(\mathbf{x}_i, \mathbf{x}_j) \leq D, \quad \forall (\mathbf{x}_i, \mathbf{x}_j) \in E(F)$$

where F is a spanning forest, $k(F)$ is the number of connected components (clusters) in F, and $E(F)$ is the set of edges in F.

2.2 Genetic Algorithm Framework

The ESC algorithm employs a genetic algorithm to search for the optimal spanning forest. The genetic algorithm consists of the following components:

- **Population Initialization**: Generate an initial population of random spanning forests.
- **Fitness Function**: Evaluate the fitness of each individual based on the number of clusters.
- **Selection**: Select individuals for reproduction based on their fitness.
- **Crossover**: Combine pairs of individuals to produce offspring by merging their spanning forests.
- **Mutation**: Introduce random modifications to offspring to maintain diversity.
- **Termination**: Repeat the process for a specified number of generations or until convergence.

3 Algorithm

The ESC algorithm can be described using the pseudocode shown in Algorithm 1.

Algorithm 1 Evolutionary Spanning Clustering (ESC)

Input: Dataset $\mathbf{X}$, distance threshold D, population size P, number of generations G, mutation rate μ
Output: Optimal spanning forest F^*
Initialize population $\mathcal{P}$ with P random spanning forests
for generation $g = 1$ to G **do**
    for each individual $F \in \mathcal{P}$ **do**
        Compute fitness $f(F) = k(F)$
    end for
    Select parents from $\mathcal{P}$ based on fitness
    Apply crossover to parents to produce offspring
    Apply mutation to offspring with probability μ
    Replace worst individuals in $\mathcal{P}$ with offspring
end for
$F^* \leftarrow \arg\min_{F \in \mathcal{P}} f(F)$
return F^*

4 Scaling ESC for Large Datasets

When dealing with a large number of data points, the computational complexity of clustering algorithms, including the ESC algorithm, can become prohibitive. To address this challenge, we propose a grid-based approximation technique that significantly reduces the computational overhead while preserving the effectiveness of the ESC algorithm.

4.1 Grid-based Approximation Technique

The key idea is to partition the data space into a grid of equal-sized cells and use the average of points within each grid cell as a representative node for that cell. By clustering these average points instead of the entire dataset, we reduce the problem size and computational cost. Once the average points are clustered, each original data point is assigned the label of the nearest average point, effectively labeling the entire dataset.

4.2 Steps of the Grid-based Approximation

1. Grid Partitioning: Divide the data space into a grid of equal-sized cells. The size of the grid cells is determined based on the desired level of approximation and computational efficiency.

2. Calculate Average Points: For each grid cell, compute the average point (centroid) of all data points within that cell. These average points act as representative nodes for constructing spanning trees.

3. Apply ESC on Average Points: Run the ESC algorithm on the average points to find the optimal spanning forest, ensuring that the clusters formed respect the given distance threshold.

4. Label Data Points: Assign each original data point the label of the average point closest to it. This step labels the entire dataset based on the clustering of average points.

4.3 Algorithm: Grid-based ESC for Large Datasets

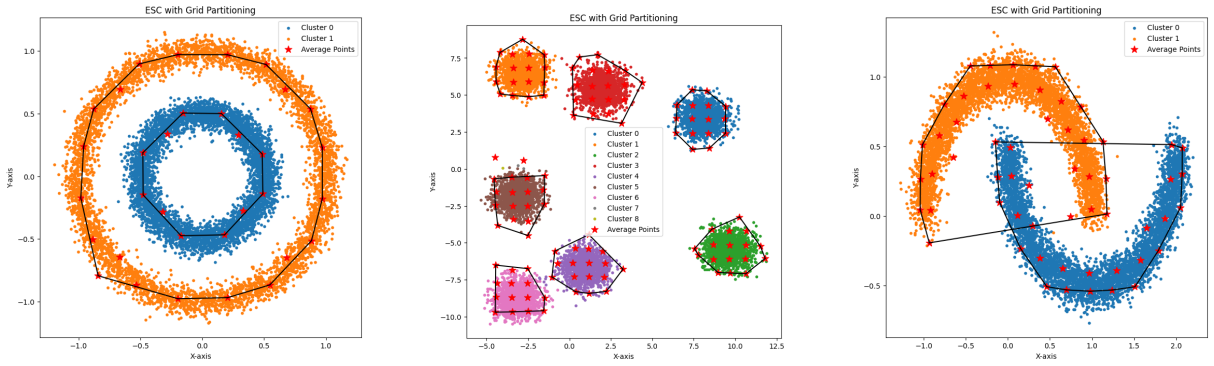
Below is the pseudocode for the grid-based ESC approach:

Algorithm 2 Grid-based Evolutionary Spanning Clustering (Grid-ESC)

Input: Dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$, distance threshold D, grid size g
Output: Cluster labels for each data point in $\mathbf{X}$
Divide the data space into a grid with cell size g
Initialize an empty list of average points $\mathbf{A}$
for each grid cell C **do**
    Calculate the average point $\mathbf{a}_C$ of all points in C
    Add $\mathbf{a}_C$ to $\mathbf{A}$
end for
Apply ESC to the average points $\mathbf{A}$ to form clusters
for each data point $\mathbf{x}_i \in \mathbf{X}$ **do**
    Find the closest average point $\mathbf{a}_{C_j} \in \mathbf{A}$
    Assign $\mathbf{x}_i$ the label of $\mathbf{a}_{C_j}$
end for
return Cluster labels for each data point in $\mathbf{X}$

5 Experimental Results

Algorithm's performance is evaluated on three large synthetic datasets consisting ten thousand data points using the suggested grid-based approach.



(a) ESC clustering on the circles dataset.

(b) ESC clustering on the blobs dataset.

(c) ESC clustering on the moons dataset.

Figure 1: Comparison of ESC clustering on different synthetic datasets.

6 Open-Source Code

The code used for experiments is available at
shttps://github.com/SoroushOskouei/Evolutionary-Spanning-Clustering-ESC-