Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.034656
*Article*

# Semi-Supervised Clustering Algorithm Based on Deep Feature Mapping

**Xiong Xu[1], Chun Zhou[2,*], Chenggang Wang[1], Xiaoyan Zhang[2] and Hua Meng[2]**

[1]Southwest China Institute of Electronic Technology, Chengdu, 610036, China
[2]School of Mathematics, Southwest Jiaotong University, Chengdu, 611756, China
*Corresponding Author: Chun Zhou. Email: zc97@my.swjtu.edu.cn
Received: 23 July 2022; Accepted: 13 December 2022

**Abstract:** Clustering analysis is one of the main concerns in data mining. A common approach to the clustering process is to bring together points that are close to each other and separate points that are away from each other. Therefore, measuring the distance between sample points is crucial to the effectiveness of clustering. Filtering features by label information and measuring the distance between samples by these features is a common supervised learning method to reconstruct distance metric. However, in many application scenarios, it is very expensive to obtain a large number of labeled samples. In this paper, to solve the clustering problem in the few supervised sample and high data dimensionality scenarios, a novel semi-supervised clustering algorithm is proposed by designing an improved prototype network that attempts to reconstruct the distance metric in the sample space with a small amount of pairwise supervised information, such as Must-Link and Cannot-Link, and then cluster the data in the new metric space. The core idea is to make the similar ones closer and the dissimilar ones further away through embedding mapping. Extensive experiments on both real-world and synthetic datasets show the effectiveness of this algorithm. Average clustering metrics on various datasets improved by 8% compared to the comparison algorithm.

**Keywords:** Metric learning; semi-supervised clustering; prototypical network; feature mapping

## 1 Introduction

Nowadays, we are facing the challenge of processing a massive amount of data generated by various applications. Data analysis methods are beneficial for uncovering the internal structure of data. Given similarity measure, the clustering algorithm groups a set of data such that samples in the same collection are similar and in different collections are dissimilar [1]. Traditional clustering algorithms include K-means [2], Mean Shift [3], and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [4,5]. These algorithms have been widely used in engineering [6], computer sciences [7], life and medical sciences [8], earth sciences [9], social sciences [10] and economics [11], and many other fields [12].

However, the traditional unsupervised clustering algorithm has two limitations. First, unsupervised clustering algorithms, which do not need prior knowledge, can not work well for clustering data with more complex structures. Not all actual data have no labels. Although manual annotation is expensive, we can still obtain some prior knowledge of data, such as sample labels and paired constraints annotations. Semi-supervised clustering algorithms are the improved generalization methods of machine learning. They can utilize prior knowledge to guide the clustering [13]. Second, there are increasing high-dimensional data such as digital images, financial time series, and gene expression microarrays, from which it is urgent to discover new structures and knowledge. However, it is difficult for traditional unsupervised and semi-supervised clustering algorithms to find an appropriate distance metric in the original high-dimensional feature space [14].

Therefore, some researchers propose a few dimensionality reduction algorithms to map the data into a lower-dimensional space, such as Principal Component Analysis (PCA), Isometric Mapping (Isomap) [15], and Local Linear Embedding (LLE) [16]. PCA is a linear dimensionality reduction algorithm, which is essentially a linear transformation of the data and is less effective in reducing the dimensionality of data that is linearly indistinguishable; LLE attempts to reduce the dimensionality while maintaining linear relationships within the data neighborhood, thus effectively reducing the dimensionality of stream-shaped data. Isomap calculates the geodesic distances between data samples and reduces the dimensionality of the data while maintaining the distances between the samples, preserving the global information of the data, which can also be applied to streamlined data. These methods can reduce data dimension while not destroying the data distribution. But they do not utilize any data prior information and cannot group the samples in the new feature space.

Metric learning can solve this problem with supervised information. Euclidean distance is a representative metric that defines the distance between elements in metric space. K-means uses the Euclidean distance to calculate the distance between a sample with each cluster center. KNN algorithm uses Euclidean distance to find the K nearest neighbors of a sample. Metric learning aims to learn a feature mapping that minimizes the intraclass distance and maximizes the interclass distance, which helps to discriminate different classes of samples in the original metric space [17–19]. The schematic diagram of metric learning is shown in Fig. 1.

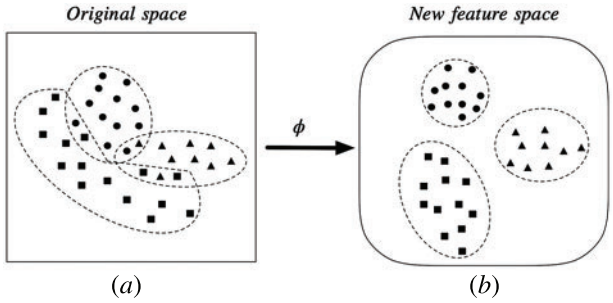



**Figure 1:** Metric learning. Mapping data from the original space to a new, more discriminatory special space

Deep neural networks are often used to construct mappings for metric learning due to their powerful non-linear fitting capacity. The Prototypical Network is a metric learning method that uses neural networks to learn the feature mapping of data with few labels [20]. Prototypical Network can reconfigure the spatial metric and facilitate the classification and clustering of data, therefore it and its variants are widely used in many research fields [21–23]. The Prototypical Networks divide the labeled

data of a category into a Support set and a Query set. Each class center is the mean vector of the embedded support points belonging to its class. Finally, learning proceeds by minimizing a distance function between query points and the class center via Stochastic Gradient Descent (SGD).

To compensate for the shortcomings of unsupervised clustering algorithm that can not use any prior information and hard to find an appropriate distance metric for high-dimensional data, this paper combines the ideas of semi-supervised clustering and metric learning and proposes a semi-supervised clustering algorithm based on deep feature mapping (SSFM). SSFM uses a modified prototype network and a small number of data priors to learn a non-linear mapping that maps all data to a new metric space with higher discrimination, increasing the separability of the samples. The data is then clustered in this metric space. The main contributions of this paper are.

- Applying the ideas of prototypical networks and metric learning to semi-supervised clustering and achieving good results.
- Performing metric learning using neural networks and the learned feature mapping has a closed-form solution that enables feature mapping of unsupervised data as well.
- Designing a new feature mapping method using a modified prototypical network loss function that allows samples of different categories to be better separated in the new feature space, with improved clustering results.
- An algorithm is given for clustering semi-supervised data in a high-dimensional space where Euclidean distance is not applicable.

The remainder of the article is organized as follows. In Section 2, the related works are introduced. In Section 3, the metric-based K-means algorithm and the basic framework of the Prototypical Network algorithm are briefly illustrated. Our SSFM algorithm is presented In Section 4. Further more, Section 5 shows the experimental data, experimental methods, comparative experimental results, feature mapping performances and parameter impact analysis. Section 6 concludes the paper and outlooks future work.

## 2 Related Works

Semi-supervised clustering algorithms usually use pairs of constraints as supervised information, i.e., Must-Link and Cannot-Link constraints [24–26], where Must-Link constraint indicates that two or more samples belong to the same class. In contrast, Cannot-Link constraint indicates that two or more samples do not belong to the same class.

Semi-supervised DenPeak Clustering (SSDC) is a representative of the Semi-supervised clustering algorithms. It generates some clusters (much more than the actual number of categories) using Density Peak Clustering (DenPeak) [27] without violating the Cannot-Link constraint and then traversing all clusters. The algorithm will contact two clusters if they have sample points that satisfy the Must-Link constraint [28]. SC-Kmeans algorithm [29] is proposed for how to make full use of the efficient prior knowledge in semi-supervised clustering algorithms. The algorithm expands the pairwise constraints by using both pairwise constraints and independent class labels to obtain a new set of ML and CL constraints, which improves the clustering accuracy of K-means method. However, the efficiency of SC-algorithm is not high for processing large scale data. Semi-supervised fuzzy clustering with fuzzy pairwise constraints (SSFPC) [30] extends the traditional pairwise constraint (i.e., Must-Link or Cannot-Link) to fuzzy pairwise constraint and avoids eliminating the fuzzy characteristics. Chen et al. proposed a semi-supervised clustering algorithm (ASGL) using only pairwise relationships, learning both similarity matrices in feature space and label space, exploring the local and global

structures of the data respectively, and obtaining better clustering results [31]. Yan et al. proposed the semi-supervised density peaks clustering algorithm (SSDPC) [32]. Instead of clustering in the original feature space of the data, SSDPC uses the semi-supervised information of pairwise constraints Must-Link and Cannot-Link to learn a linear mapping, where the samples of Must-Link are close to each other, and the samples of Cannot-Link are far from each other in the mapped space. Clustering is done using the classical DenPeak algorithm in the new feature space. SSDPC makes effective use of the prior information of the data and improves the clustering performance, but it still has some limitations. First, the linear mapping method used by SSDPC has difficulty handling data with more complex distributions, such as manifold data. Second, its distance metric is the Euclidean distance, which is not applicable to high-dimensional data. Third, SSDPC uses the eigendecomposition of the matrix to calculate the projection direction, which is inefficient when the amount of data is large.

The above semi-supervised clustering algorithms achieve good clustering results on low-dimensional datasets, but they cannot handle high-dimensional data because the ordinary metrics are not applicable in the high-dimensional space. Snell et al. proposed Prototypical Networks which use a small amount of supervised information and neural network embedders to map high-dimensional data to low-dimensional space, and successfully bring similar samples close to each other in low-dimensional space. But dissimilar samples are not significantly far away from each other due to its loss function. We improve the architecture of the prototype network and its loss function and propose the SSFM algorithm, which has a better feature mapping effect.

## 3 Preliminaries and Motivation

### *3.1 K-means Algorithm and Metric Learning*

Many machine learning algorithms rely on metrics. For example, the K-means algorithm relies on a specific metric to assign a sample to the cluster center nearest to it. Suppose there is a dataset $D = \{X_1, X_2, \ldots, X_n\}$, where the number of samples is $n$, and the dimension of the data is $m$. The K-means algorithm aims to group $n$ samples into a specified $k$ cluster based on similarity. Each point can only be crowded into the nearest cluster center. The algorithmic procedure of K-means is summarized as follows.

Firstly, initializing $k$ cluster centers $\{C_1, C_2, \ldots, C_k\}$, $1 < k \leq n$, and computing the distance $d\left(X_i, C_j\right)$ between samples and each cluster center. If the distance metric is chosen as a Euclidean distance, then

$$d\left(X_i, C_j\right) = \sqrt{\sum\nolimits_{t=1}^{m} \left(X_{it} - C_{jt}\right)^2} \tag{1}$$

where $X_i$ denotes the ith point, $1 \leq i \leq n$; $C_j$ denotes the jth cluster center, $1 \leq j \leq k$; $X_{it}$ denotes the tth feature of the ith sample, $1 \leq t \leq m$; $C_{jt}$ denotes the tth center attribute of the jth cluster. The distance of object to each cluster center is compared in turn, and the object is assigned to the cluster of the nearest cluster center to obtain $k$ clusters $\{P_1, P_2, \ldots, P_k\}$. Recalculate the cluster centers based on the obtained $k$ clusters

$$C_l = \frac{1}{|P_l|} \sum\nolimits_{X_i \in P_l} X_i \tag{2}$$

where $C_l$ denotes the center of the $l$th cluster, $1 \leq l \leq k$, $|P_l|$ denotes the number of objects in the $l$th cluster, $X_i$ denotes the ith object in the $l$th cluster, $1 \leq i \leq |P_l|$. The distance from objects to the center of each cluster is then calculated based on the new cluster centers, and so on, iterating until the cluster centers almost not change.

However, the Euclidean distance is not applicable in all data scenarios. Choosing a good metric can improve the generalization of a model. We want to start with the data itself and find an appropriate metric for that data. Metric learning uses supervised information to map the data into a more discriminative metric space [33], which is beneficial for bringing similar samples close to each other and separating dissimilar samples under the new metric. Combining the K-means algorithm with metric learning can effectively improve the algorithm's performance.

### *3.2 Prototypical Network*

Similar to the semi-supervised clustering scenario, the Prototypical Network also uses a small amount of prior information to learn the essential features of the data. Snell et al. proposed the Prototypical Network structure in 2017 for solving image recognition problems in the few shot dilemmas. Denote the few shot training set as $X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$, which contains $C$ classes of data with only a small number of samples in each class. The samples of each class in the training set are divided into a Support set ($S$) and a Query set ($Q$). The main idea of the prototype network is to map the data into a new feature space through metric learning. A feature prototype is found for each class on the new feature space. When the feature mapping of the image to be tested is obtained, a distance metric is applied to the mapped test image and the feature prototype of each class separately to obtain the prediction result. The class prototypes are defined as follows

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i) \tag{3}$$

where $S_k$ denotes the support set for the $k$th class, consisting of some samples from that class. $|S_k|$ denotes the cardinality of the set $S_k$, and $f_\phi$ represents the non-linear mapping, represented as a four-layer fully convolutional neural network in Snell's article. The similarity between query points and class prototypes is represented by the SoftMax value of the negative Euclidean distance

$$p_\phi(y = k|\mathbf{x}) = \frac{\exp\left(-d\left(f_\phi(\mathbf{x}), \mathbf{c}_k\right)\right)}{\sum_{k'} \exp\left(-d\left(f_\phi(\mathbf{x}), \mathbf{c}_{k'}\right)\right)} \tag{4}$$

Learning proceeds by minimizing the negative log-probability $\mathrm{J}(\phi) = -\log p_\phi(y = k|\mathbf{x})$ of the entire Query set via gradient descent. Fig. 2 shows a schematic representation of the loss calculation of Prototypical Network in the new feature space.

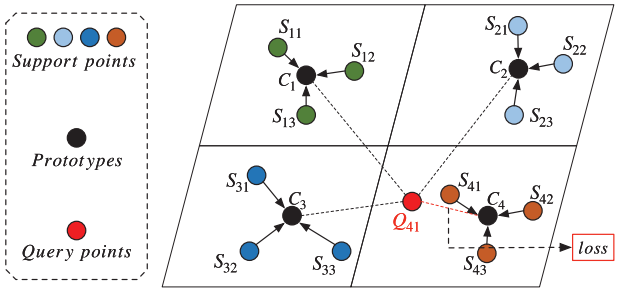



**Figure 2:** Prototypical Network feature space

### *3.3 Motivation*

The semi-supervised data prior information Must-Link and Cannot-Link provide some useful prior knowledge of the data, i.e., which samples are supposed to be similar and which samples are highly divergent. With the idea and framework of the Prototypical Network, we use the prior information to learn an embedding mapping that embeds the data into a new metric space and then clusters them. Although the Prototypical Network performs well in few shot image recognition problems, applied to semi-supervised clustering it has the following shortcomings.

- The loss function of the Prototypical Network is mainly considered to bring similar sample points closer, while the constraint on dissimilar sample points is not obvious enough, resulting in dissimilar samples not being clearly distinguished (subsequent experiments will demonstrate the existence of this problem).
- The traditional Prototypical Network is targeted at image data and therefore constructs embedding mappings using convolutional neural networks, which cannot be extended to embedding mappings of general data.

In this paper, we propose a semi-supervised clustering algorithm based on the Prototypical Network with corrected loss, which makes the differences between the embedding of heterogeneous samples more obvious. In particular, we build the network architecture according to the type of dataset so that data of different types can be mapped to the new metric space.

## 4 SSFM Algorithm

The semi-supervised clustering problem is defined as to cluster the entire dataset where a large number of samples are unlabeled, with the help of a small portion of labeled samples. These labeled samples are usually called semi-supervised information. Generally, except for labels on samples, the semi-supervised information could also be given in the form of pairwise constraints or other prior information. In this paper, we mainly focus on the clustering method with pairwise constraints as semi supervised information, where a pairwise constraint has the form $(S_i, S_j)$ and $S_i$ or $S_j$ can be a sample or a collection of samples. All the pairwise constraints are divided into two categories called Must-Link, which specifies that the samples belong to the same class, and Cannot-Link, which specifies that the samples do not belong to the same class. Suppose there are $T$ supervised samples from each class ($T$ typically occupies only 10% to 20% of all samples in each class). Supervised samples of each class constitute a Must-Link set, and supervised samples of every two class constitute a Cannot-Link sets pair. The SSFM algorithm contains three stages.

- Firstly, learning an encoder with supervised information, i.e., Must-Link sets and Cannot-Link sets. The loss function is designed to maximize the distance between two Cannot-Link sets and minimize the distance in a Must-Link set.
- Secondly, feature mapping all the data with the trained encoder to the new feature space.
- Thirdly, Clustering samples in the new feature space using an unsupervised clustering algorithm.

The overall framework of the algorithm is shown in Fig. 3.

### *4.1 Calculate Must-Link Loss*

Denote the set of Must-Link sets as $\mathcal{ML} = \{M_1, M_2, \ldots, M_n\}$, where $M_k = \{u_{k,1}, u_{k,2}, \ldots, u_{k,\sigma_k}\}$ indicates a Must-Link set. The samples in the $M_k$ belong to the same class. $\sigma_k$ is the number of samples in this set. The set of Cannot-Link sets is denoted as $\mathcal{CL} = \{C_1, C_2, \ldots, C_m\}$, where $C_k = \{c_{k,1}, c_{k,2}, \ldots, c_{k,\tau_k}\}$ indicates a Cannot-Link set. Any two sets in $\mathcal{CL}$ do not belong to the same class. $\tau_k$

is the number of samples in this set. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of samples, and the center of this set is defined as

$$Ave\,(X) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5}$$

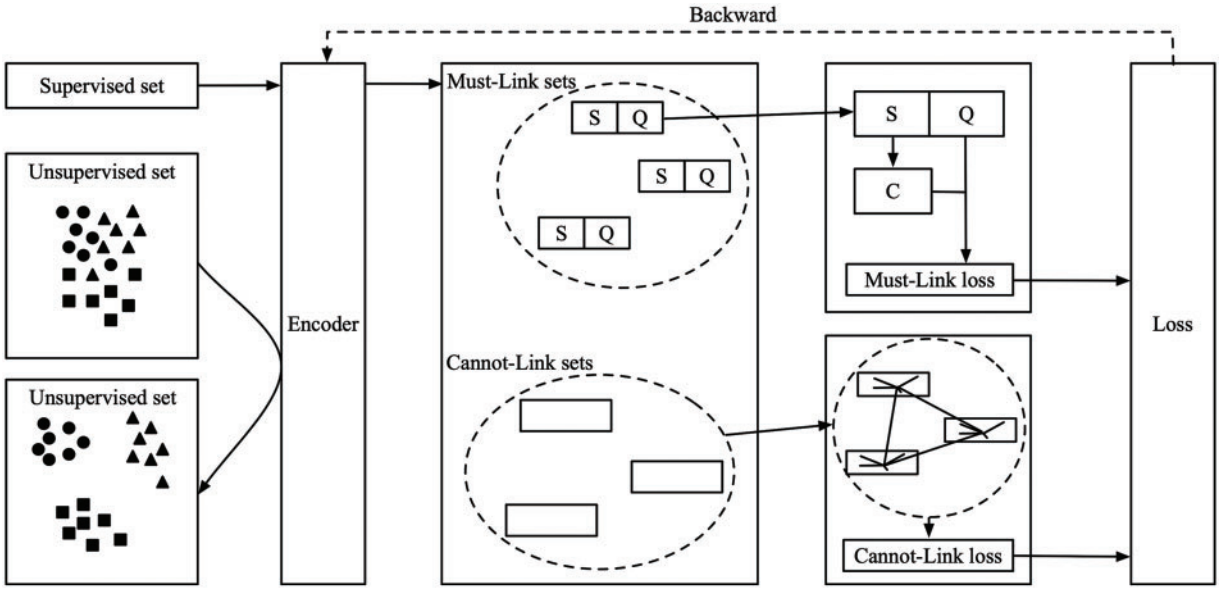



**Figure 3:** Overall framework of SSFM algorithm

In order to keep the distance between samples in the same Must-Link set as small in the new feature space, a Must-Link set is divided into a Support set and a Query set. Let the Support set be encoded to obtain the centers of the Support set in the new feature space, and encode the Query set to obtain its feature mapping. In the new feature space, optimize the following loss function to make each point in the Query set close to the center of its Support set.

**Algorithm 1:** Encoder training loss. $\mathcal{ML} = \{M_1, M_2, \ldots, M_n\}$ is the Must-Link cluster. $\mathcal{CL} = \{C_1, C_2, \ldots, C_m\}$ is the Cannot-Link cluster. $IC$ denotes the mean value of inner distance of Cannot-Link sets. $BC$ denotes the mean value of outer distance of Cannot-Link sets. $T_i$ is the cardinality of $i$th Cannot-Link set. Ave $(\cdot)$ is computed by Eq. (5). D $(\cdot, \cdot)$ denotes Euclidean distance.

**Input:** Must-Link cluster $\mathcal{ML}$, Cannot-Link cluster $\mathcal{CL}$, cardinality of query set of $i$th Must-Link set $q_i, i = 1, 2, \ldots, n$, weight factor $\alpha$, encoder $\phi$

**Output:** Encoder training loss $\mathcal{L}$

1. $Loss_1 \Leftarrow 0$, $IC \Leftarrow 0$, $BC \Leftarrow 0$
2. $q \Leftarrow \sum_{i=1}^{n} q_i$, $\tau \Leftarrow \sum_{i=1}^{m} \tau_i$

**Step1-Compute Must-Link loss ($Loss_1$)**

3. **for** $i = 1:n$ **do**
4. **for** $j = 1:q_i$ **do**
5. $$\text{Loss}_1 \Leftarrow \text{Loss}_1 + \log \frac{\exp\left(-d\left(Ave\,(S_i), Q_{ij}\right)\right)}{\sum_{k=1}^{n} \exp\left(-d\left(Ave\,(S_k), Q_{ij}\right)\right)}$$
6. **end**

(Continued)

**Algorithm 1:** Continued

7. **end**
8. $Loss_1 \Leftarrow -\frac{1}{q} Loss_1$

**Step2-Compute Cannot-Link loss ($Loss_2$)**

9. **for** $i = 1:m$ **do**
10. **for** $j = 1:\tau_i$ **do**
11. $IC \Leftarrow IC + \frac{1}{\tau} d\left(Ave\left(\phi\left(C_i\right)\right), \phi\left(C_{ij}\right)\right)$
12. **end**
13. **for** $j = 1:m$ **do**
14. $\mathrm{BC} \Leftarrow BC + \frac{1}{m^2} \mathrm{d}\left(Ave\left(\phi\left(C_i\right)\right), Ave\left(\phi\left(C_j\right)\right)\right)$
15. **end**
16. **end**
17. $Loss_2 \Leftarrow \frac{IC}{BC}$
18. $\mathcal{L} \Leftarrow Loss_1 + \alpha * Loss_2$
19. **Return** $\mathcal{L}$

$$Loss_1 = -\frac{1}{\sum_{i=1}^{n} q_i} \sum_{i=1}^{n} \sum_{j=1}^{q_i} \log \frac{\exp\left(-d\left(Ave\left(S_i\right), Q_{ij}\right)\right)}{\sum_{k=1}^{n} \exp\left(-d\left(Ave\left(S_k\right), Q_{ij}\right)\right)} \tag{6}$$

where $S_i, Q_i, i = 1, 2, \ldots, n$ are the encoded Support set and Query set of the $i$th Must-Link set. $Q_{ij}$ denotes the $j$th sample of the $i$th Query set. $q_i$ is the cardinality of the $i$th Query set, and $d(\cdot, \cdot)$ denotes the Euclidean distance. Optimising $Loss_1$ can simultaneously adjust the embedding of samples in S and Q so that samples of the same class are embedded in a cluster.

### *4.2 Calculate Cannot-Link Loss*

For the data from different classes to be separated further away in the new feature space, the intraclass distance of any two Cannot-Link sets should be as large as possible compared to the interclass distance. Cannot-Link loss function is designed as

$$Loss_2 = \frac{\frac{1}{\sum_{i=1}^{m} \tau_i} \sum_{i=1}^{m} \sum_{j=1}^{\tau_i} d\left(Ave\left(\phi\left(C_i\right)\right), \phi\left(C_{ij}\right)\right)}{\frac{1}{m^2} \sum_{i,j=1}^{m} d\left(Ave\left(\phi\left(C_i\right)\right), Ave\left(\phi\left(C_j\right)\right)\right)} \tag{7}$$

where $\phi$ indicates the feature mapping function. $C_{ij}$ denotes the $j$th sample of the $i$th Cannot-Link set, and $\tau_i$ denotes the cardinality of the $i$th Cannot-Link set.

The final loss function of the algorithm $\mathcal{L}$ is defined as

$$\mathcal{L} = Loss_1 + \alpha * Loss_2 \tag{8}$$

where $\alpha$ is a weighting factor to balance the contribution of the two components. Optimize the loss via gradient descent and adjust the parameters in the encoder until the loss falls steady. This process is described in Algorithm 1.

### *4.3 Encoder Designs*

Different encoders apply to diverse semi-supervised clustering data. For low-dimensional data, a fully connected neural network is adequate. However, a multi-layer convolutional neural network is more suitable for high-dimensional data, such as image data. Convolutional neural networks have the advantage of local connectivity and shared parameters, reducing the trained parameters. Furthermore, convolutional neural networks can use the original image as input, avoiding a complex feature extraction process. CNN perceives the local pixels of the image, which can effectively learn the corresponding features from many samples [34]. The data is input into the trained encoder and crowded into various clusters in the new feature space via an unsupervised clustering algorithm e.g., K-means.

## 5 Experimental Analysis

### *5.1 Experimental Datasets*

We conducted experiments on the simulation datasets, the University of California Irvine (UCI) datasets, and the image datasets, respectively. The simulation datasets include Aggregation (Aggr) and Jain. The UCI real-world datasets include Iris, Wine, Synthetic Control, Glass Identification, Balance Scale and Letter-Recognition (only data of the letter D, O and Q). In addition, we also run experiments on three image datasets, including the MNIST dataset (selecting 1,000 samples from each class), the CIFAR-10 dataset (selecting 500 samples from each class), and the ORL face image dataset (selecting 10 samples from each people). Table 1 shows the specific number of samples, feature dimensions, and categories for each dataset.

**Table 1:** Datasets details

| Data name | Instances | Attributes | Classes |
|---|---|---|---|
| Jain | 373 | 2 | 2 |
| Aggregation | 788 | 2 | 7 |
| Iris | 150 | 4 | 3 |
| Wine | 178 | 13 | 3 |
| Letter DOQ | 450 | 16 | 3 |
| Synthetic | 600 | 60 | 6 |
| Glass | 214 | 10 | 6 |
| Balance | 625 | 4 | 3 |
| MNIST | 10,000 | $28 * 28$ | 10 |
| CIFAR-10 | 5,000 | $32 * 32$ | 10 |
| ORL | 100 | $112 * 92$ | 10 |

The UCI real-world datasets can be found at https://archive.ics.uci.edu/ml/datasets.php.

The MNIST dataset can be found at http://yann.lecun.com/exdb/mnist/.

The CIFAR-10 dataset can be found at https://www.cs.toronto.edu/~kriz/cifar.html.

The ORL dataset can be found at http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

### *5.2 Model Evaluation Indicators*

Various metrics have been proposed for the evaluation of clustering algorithms, such as the Adjusted Mutual Information (AMI) [35] and the Adjusted Rand Index (ARI) [36]. The AMI and the ARI have $[-1, 1]$ value range. The higher values indicate a better clustering result. In addition, we used the 1-nearest neighbor (1NN) classification accuracy to evaluate the performance of feature mapping. A high 1NN classification accuracy means that the original data are easier to distinguish after feature mapping. We compared our algorithm with some classical unsupervised clustering and semi-supervised clustering algorithms, including K-means, DBSCAN, DenPeak, SSDC, SSDPC, SC-Kmeans and Proto-Net.

### *5.3 Experimental Method*

The supervised information is obtained by randomly selecting a small fraction (typically 20%) of data from each class to form the Must-Link sets and the Cannot-Link sets. To ensure the amount of supervised information is identical in different algorithms, we converted all the samples in the Must-Link sets and the Cannot-Link sets into pairs constraints, which have the same number as other semi-supervised clustering algorithms.

The encoder was designed as a fully connected neural network with only one hidden layer for the low-dimensional datasets, including Aggregation, Jain, Iris, Wine, Letter DOQ, Synthetic, Balance, and Glass. In addition, we designed the encoder as a four-layer fully convolutional neural network for the image datasets. The convolutional kernel size was $3 \times 3$ and the max-pooling kernel size was $2 \times 2$. We use Rectified Linear Unit (ReLU) as an activation function, Batch Normalization (BN) as regularization method and use the Adam algorithm to optimize the loss. The model settings are presented in Table 2.

**Table 2:** Model settings

| Dataset | Encoder |
|---|---|
| Simulation and UCI datasets | Dense, BN1D, ReLU, Dense |
| MNIST | Conv2D($1 \times 64 \times 3 \times 3$), BN2D, ReLU, MaxPool2D($2 \times 2$), $3 \times$ [Conv2D($64 \times 64 \times 3 \times 3$), ReLU, BN1D, MaxPool2D($2 \times 2$)] |
| CIFAR-10 | Conv2D($3 \times 64 \times 3 \times 3$), BN2D, ReLU, MaxPool2D($2 \times 2$), $3 \times$ [Conv2D($64 \times 64 \times 3 \times 3$), BN2D, ReLU, MaxPool2D($2 \times 2$)] |
| ORL | Conv2D($1 \times 64 \times 3 \times 3$), BN2D, ReLU,MaxPool2D($2 \times 2$), $3 \times$ [Conv2D($64 \times 64 \times 3 \times 3$), BN2D, ReLU, MaxPool2D($2 \times 2$)] |

### *5.4 Experimental Results*

Each algorithm was performed 10 times for each dataset. The average ARI and AMI are shown in Tables 3 and 4 The best results on each dataset are bolded.

Tables 2 and 3 show that SSFM outperforms other clustering algorithms for high-dimensional data and also perform well in low-dimensional data, especially on the DOQ dataset. For the five low-dimensional datasets of Iris, Wine, DOQ, Synthetic and Glass Balance, the clustering performance of SSFM is significantly better than the other seven clustering algorithms. For the Jain and Aggregation datasets, density-based clustering algorithms such as DBSCAN and DenPeak achieved better clustering results because these two datasets are non-convex datasets. Density-based clustering algorithms

have better clustering results for such irregularly structured datasets. SSFM also has better clustering results on two high-dimensional image datasets, MNIST and ORL, due to the fact that the metric loss designed in SSFM optimizes the parameters of the convolutional neural network constructed for the image data, extracting features of the images that are conducive to clustering. However, the SSFM and the Proto-Net algorithms do not perform well for the CIFAR-10 dataset. The reason may be that the encoder only uses the simplest four-layer fully convolutional neural network, which cannot extract many valuable features for three-channel images.

**Table 3:** ARI (%)

| | Synthetic | | UCI real world | | | | | | Image | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | Jain | Aggr | Iris | Wine | DOQ | Synth | Glass | Balance | MNIST | CIFAR-10 | ORL |
| K-means [2] | 32.14 | 76.00 | 73.20 | 37.11 | 0.81 | 56.92 | 54.17 | 16.05 | 35.29 | 4.02 | 62.22 |
| DBSCAN [4] | **90.50** | **80.93** | 52.06 | 42.28 | 1.48 | 48.10 | 41.12 | 0.01 | 0.00 | 0.00 | 0.00 |
| DenPeak [27] | 14.40 | 77.97 | 86.83 | 24.93 | 11.67 | 55.33 | 38.97 | 0.62 | N/A | N/A | N/A |
| SSDC [28] | 72.16 | 71.89 | 69.72 | 61.99 | 15.48 | 47.09 | 47.22 | 10.26 | N/A | N/A | N/A |
| SSDPC [32] | 68.21 | 72.20 | 73.78 | 74.15 | 28.66 | 62.98 | 44.84 | 12.24 | N/A | N/A | N/A |
| SC-Kmeans [29] | 72.30 | 80.02 | 82.74 | 70.32 | 36.75 | 75.80 | 59.11 | 10.37 | N/A | N/A | N/A |
| Proto-Net [20] | 69.60 | 71.09 | 81.79 | 78.30 | 60.76 | 50.11 | 46.17 | 43.65 | 81.64 | **18.10** | **70.09** |
| SSFM (Ours) | 70.55 | 75.21 | **92.22** | **88.10** | **74.91** | **80.63** | **59.39** | **52.63** | **83.17** | 16.59 | 69.30 |

**Table 4:** AMI (%)

| | Synthetic | | UCI real world | | | | | | Image | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | Jain | Aggr | Iris | Wine | DOQ | Synth | Glass | Balance | MNIST | CIFAR-10 | ORL |
| K-means [2] | 36.77 | 87.51 | 75.51 | 42.27 | 0.65 | 72.29 | 72.26 | 14.03 | 48.57 | 7.93 | 59.15 |
| DBSCAN [4] | **76.60** | 82.41 | 59.90 | 52.08 | 12.52 | 67.71 | 66.35 | 0.01 | 0.00 | 0.00 | 0.00 |
| DenPeak [27] | 40.20 | **89.48** | 85.54 | 38.44 | 15.50 | 72.65 | 69.28 | 0.72 | N/A | N/A | N/A |
| SSDC [28] | 61.59 | 75.92 | 73.80 | 64.11 | 33.50 | 67.87 | 62.70 | 11.87 | N/A | N/A | N/A |
| SSDPC [32] | 70.06 | 77.21 | 84.16 | 73.33 | 31.42 | 67.72 | 55.40 | 11.07 | N/A | N/A | N/A |
| SC-Kmeans [29] | 70.55 | 76.29 | 80.22 | 68.80 | 40.64 | **83.39** | 63.26 | 12.44 | N/A | N/A | N/A |
| Proto-Net [20] | 61.71 | 82.81 | 79.25 | 75.73 | 60.20 | 82.20 | 59.25 | 52.18 | 80.33 | 25.57 | 78.48 |
| SSFM (Ours) | 64.39 | 84.50 | **91.33** | **85.89** | **72.49** | 81.90 | **72.60** | **56.28** | **82.23** | **28.33** | **81.13** |

### *5.5 Feature Mapping Results*

Let $\gamma = 0.2$ and $\alpha = 0.5$. The SSFM algorithm map the four datasets, including Iris, Wine, Letter DOQ, and Synthetic, into the three-dimensional feature space. Fig. 4 demonstrates the performance of feature mapping. It shows that the SSFM algorithm effectively maps similar samples to one cluster and heterogeneous samples separate from each other in the new feature space, which is more beneficial for unsupervised clustering algorithms to group data. In addition, Figs. 5 and 6 show the results of the image dataset ORL and MNIST projected to a two-dimensional feature space by the SSFM algorithm. SSFM successfully mapped the photos of different persons' faces to different clusters in the feature space. The clusters of person1, person3 and person9 have a little overlap occurring, while the clusters of the remaining seven people can be clearly separated. The images of each category of the MNIST dataset are also mapped into distinct clusters that are clearly differentiated.

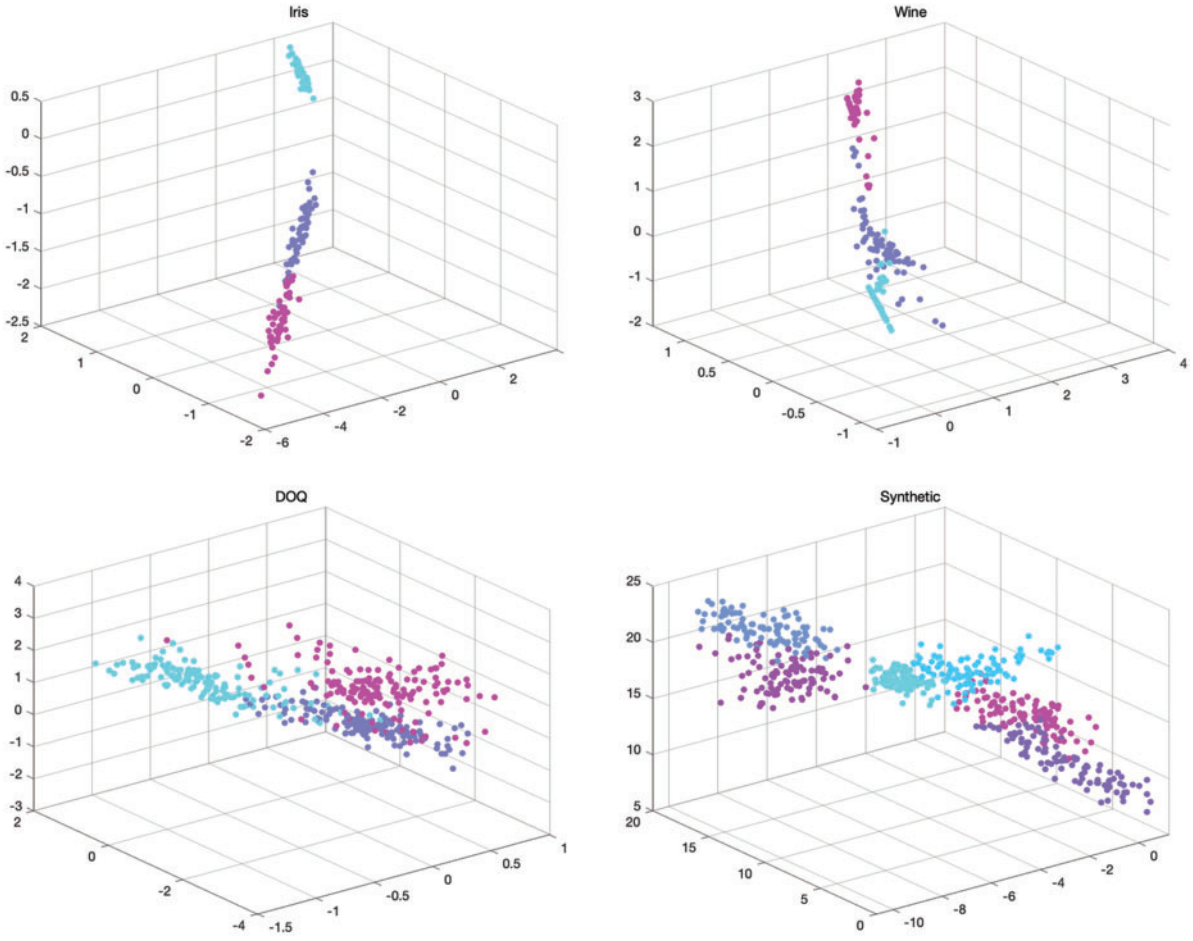



**Figure 4:** Mapping of Iris, Wine, Letter DOQ, Synthetic datasets in the 3D feature space

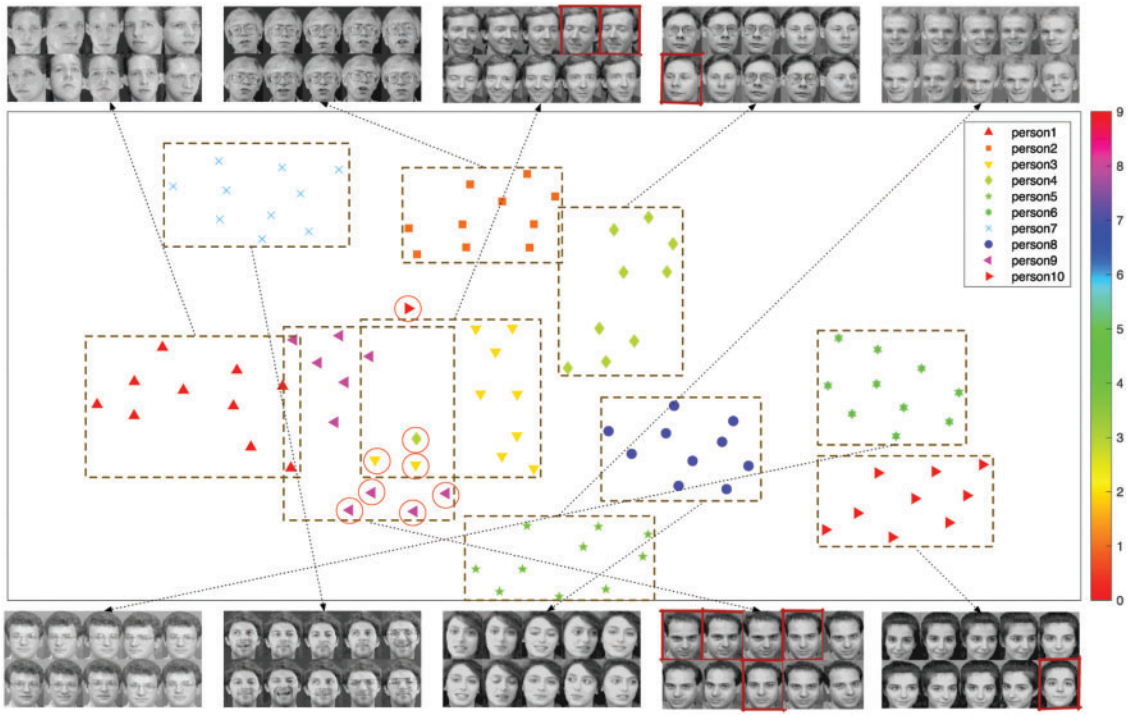
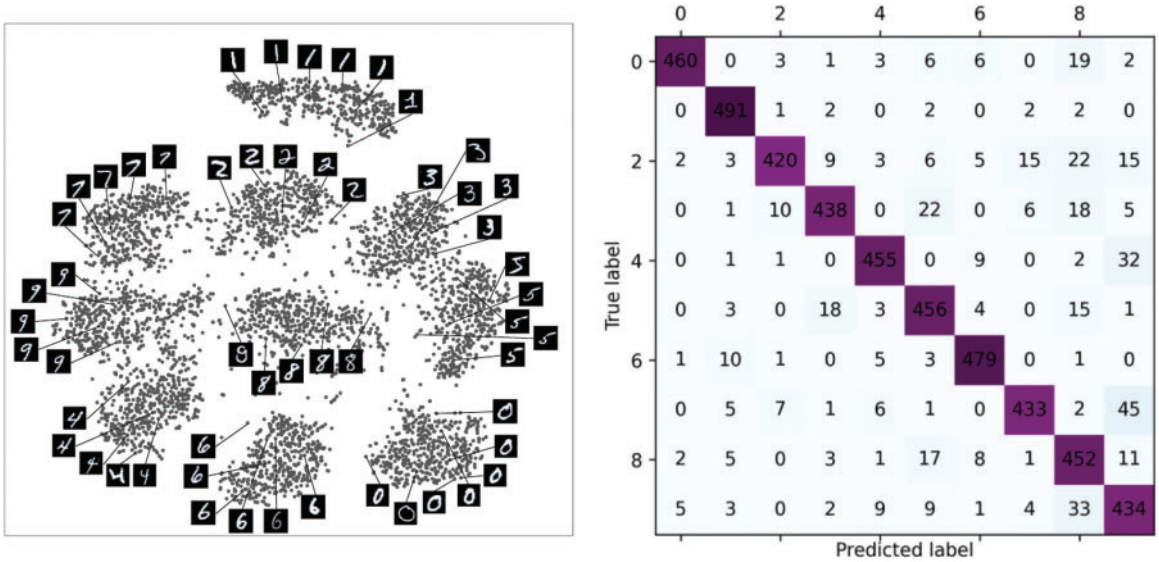



**Figure 5:** 2D features of the ORL face dataset by SSFM feature mapping. The outliers have been circled in red and the corresponding face images have been marked with red boxes




(*a*) (*b*)

**Figure 6:** (a) 2D features of the MNIST dataset after SSFM feature mapping. (b) Confusion matrix of labels predicted by clustering in new feature space and true labels

In addition, we demonstrate the data feature mapping performance of SSFM algorithm, Proto-Net algorithm and PCA on two datasets, including Wine and DOQ. Fig. 7 shows the mapping results. The performance of PCA was poor because it did not use the supervised information at all. SSFM and Proto-Net both mapped different classes of data into different clusters. However, SSFM can separate the data of different classes further than Proto-Net and be more able to reflect the differences between the data in the new feature space. The reason for this is that the loss of SSFM considers separating the Cannot-Link of the sample sets in the embedding feature space from each other, whereas Proto-Net does not consider it.

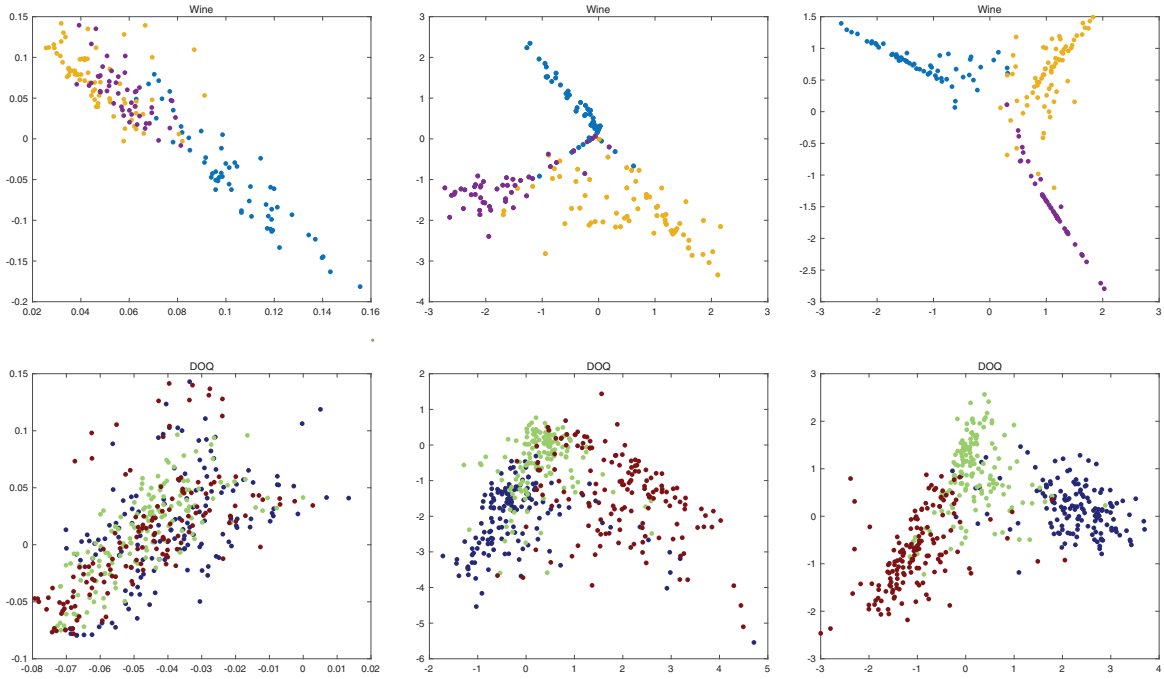



**Figure 7:** Wine, Letter DOQ datasets mapped by PCA (left), Proto-Net (middle) and SSFM (right) for 2D features

### *5.6 Parameter Impact Analysis*

Two parameters, including the loss function weight factor α and the ratio of supervised information to the total data γ, influence the performance of feature mapping. We analyze their effect on different datasets. Firstly, fix the proportion of supervised information $\gamma = 0.2$ and adjust the value of the weighting factor $\alpha$. Secondly, fix the value of the weighting factor $\alpha = 0.5$ and adjust the proportion of supervised information $\gamma$. The classification accuracy of the 1NN classifier was chosen as the metric of feature mapping effectiveness. Specifically, the whole data embedded into the new feature space was divided into a training set and a test set on a 7:3 ratio. The 1NN classifier was trained to classify the test set. Fig. 8 shows the impact of the parameters on Iris, Wine, Letter DOQ, and Synthetic datasets.

$\alpha = 0$ means not considering maximizing the distance between the Cannot-Link sets. The results show that the Cannot-Link loss is effective. Moreover, even using a small amount of supervised information, the feature mapping performance is still better than that of the model using more supervised information by choosing an appropriate $\alpha$. More supervised information will result in better feature mapping if the loss function is consistent.

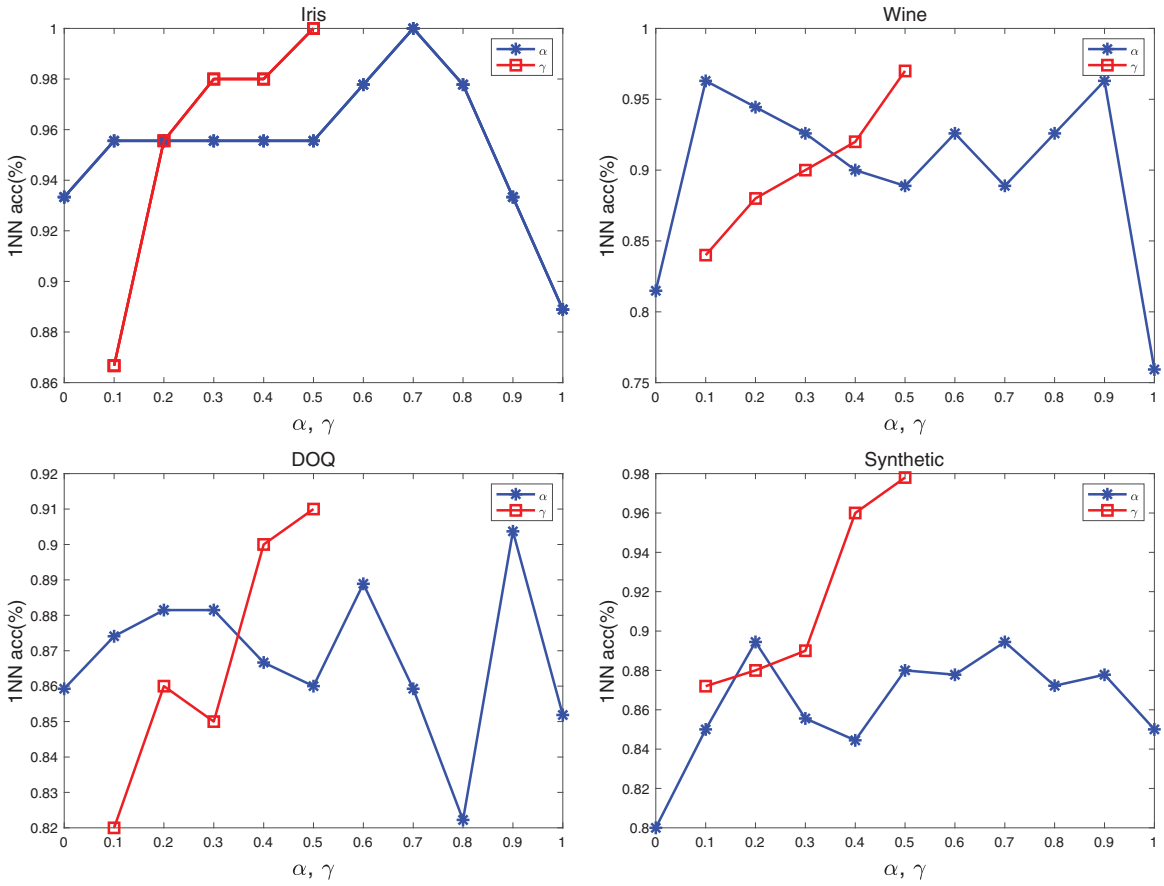



**Figure 8:** The effect of loss weighting factor $\alpha$ and supervisory information proportion $\gamma$

## 6 Conclusion

This paper presents a semi-supervised clustering algorithm (SSFM) based on metric learning and prototype networks. By designing an appropriate network structure and loss function, the algorithm can learn a feature mapping using semi-supervised information (Must-Link, Cannot-Link). This mapping embeds the original data into a new metric space and allows samples of the same type to be placed close to each other and samples of different types to be separated, thus making the data easier to cluster. Experiments were conducted on synthetic and real-world data (including low and high dimensional data), and the mapped data were passed to 1NN classification and K-means clustering. The experimental results show that the mapped data have significantly better classification and clustering results compared with the original data. Furthermore, by comparing classical unsupervised and semi-supervised clustering algorithms on a variety of datasets using common clustering metrics (ARI, AMI), the experimental results validate the effectiveness and robustness of the SSFM algorithm. SSFM algorithm requires a small amount of supervised information for each class of data, otherwise it will lead to undesirable clustering results, which is a limitation of the algorithm. One solution is to first label the data with a small number of pseudo-labels by an unsupervised clustering algorithm. Re-utilizing the semi-supervised information of the data in the new feature space to assist clustering is a future work to be accomplished.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks, Materials & Continua*, vol. 16, no. 3, pp. 645–678, 2005.
[2] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the Fifth Berkeley Symp. on Mathematical Statistics and Probability*, Oakland, CA, USA, vol. 1, 1967.
[3] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
[4] M. Ester, H. P. Kriegel, J. Sander and X. xu, "A Density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. of the Second Int. Conf. on Knowledge Discovery and Data Mining*, Portland, Oregon, pp. 226–231, 1996.
[5] Y. Xie, S. Shekhar and Y. Li, "Statistically-robust clustering techniques for mapping spatial hotspots: A survey," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1–38, 2022.
[6] W. Ma, X. Tu, B. Luo and G. Wang, "Semantic clustering based deduction learning for image recognition and classification," *Pattern Recognit.*, vol. 124, pp. 108440, 2022.
[7] A. Nazir, M. N. Cheema, B., Sheng, P. Li, H. Li *et al.,* "ECSU-Net: An embedded clustering sliced U-net coupled with fusing strategy for efficient intervertebral disc segmentation and classification," *IEEE Trans. Image Process.*, vol. 31, pp. 880–893, 2022.
[8] G. Gilam, E. M. Gramer, K. A. Webber, M. S. Ziadni, M. C. Kao *et al.,* "Classifying chronic pain using multidimensional pain-agnostic symptom assessments and clustering analysis," *Science Advances*, vol. 7, no. 37, pp. eabj0320, 2021.
[9] Y. Xie, D. Feng, X. Shen, Y. Liu, J. Zhu *et al.,* "Clustering feature constraint multiscale attention network for shadow extraction from remote sensing images," *IEEE Trans. Geosci. Remote. Sens.*, vol. 60, pp. 1–14, 2022.
[10] C. Mao, H. Liang, Z. Yu, Y. Huang and J. Guo, "A clustering method of case-involved news by combining topic network and multi-head attention mechanism," *Sensors*, vol. 21, no. 22, pp. 7501, 2021.
[11] Y. Cheng, M. Cheng, T. Pang and S. Liu, "Using clustering analysis and association rule technology in cross-marketing," *Complex.*, vol. 2021, pp. 9979874:1–9979874:11, 2021.
[12] B. S. Everitt, S. Landau and M. Leese, "Cluster analysis arnold," *A Member of the Hodder Headline Group, London*, pp. 429–438, 2001.
[13] Y. Yu, G. Yu, X. Chen and Y. Ren, "Semi-supervised multi-label linear discriminant analysis," in *Neural Information Processing*, Cham, Springer International Publishing, pp. 688–698, 2017.
[14] Z. Yu, P. Luo, J. You, H. S. Wong, H. Leung *et al.,* "Incremental semi-supervised clustering ensemble for high dimensional data clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 701–714, 2016.
[15] J. B. Tenenbaum, V. de Silva and J. C. Langford, "A global geometric framework for non-linear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[16] S. T. Roweis and L. K. Saul, "Non-linear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[17] Y. Feng, Y. Yuan and X. Liu, "Person reidentification via unsupervised cross-view metric learning," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 1849–1859, 2021.
[18] Y. Yuan, J. Lu, J. Feng and J. Zhou, "Deep localized metric learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2644–2656, 2018.

[19] G. J. Liu, H. Chen, L. Y. Wang and D. Zhu, "Metric learning based similarity measure for attribute description identification of energy data," in *2020 Int. Conf. on Machine Learning and Cybernetics*, Online, pp. 219–223, 2020.
[20] J. Snell, K. Swersky and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, Long Beach, pp. 4077–4087, 2017.
[21] Y. Li, Z. Ma, L. Gao, Y. Wu, F. Xie *et al.,* "Enhance prototypical networks with hybrid attention and confusing loss function for few-shot relation classification," *Neurocomputing*, vol. 493, pp. 362–372, 2022.
[22] H. Tang, Z. Huang, Y. Li, L. Zhang and W. Xie, "A multiscale spatial–Spectral prototypical network for hyperspectral image few-shot classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.
[23] W. Fu, L. Zhou and J. Chen, "Bidirectional matching prototypical network for few-shot image classification," *IEEE Signal Processing Letters*, vol. 29, pp. 982–986, 2022.
[24] S. Xiong, J. Azimi and X. Z. Fern, "Active learning of constraints for semi-supervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 43–54, 2014.
[25] N. M. Arzeno and H. Vikalo, "Semi-supervised affinity propagation with soft instance-level constraints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 5, pp. 1041–1052, 2015.
[26] H. Zeng and Y. Cheung, "Semi-supervised maximum margin clustering with pairwise constraints," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 926–939, 2012.
[27] A. Rodriguez and A. Liao, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
[28] Y. Ren, X. Hu, K. Shi, G. Yu, D. Yao *et al.,* "Semi-supervised DenPeak clustering with pairwise constraints," in *PRICAI 2018: Trends in Artificial Intelligence*, Cham, Springer International Publishing, pp. 837–850, 2018.
[29] Z. Chen, H. Wang and M. Hu, "An active semi-supervised clustering algorithm based on seeds set and pairwise constraints," *J. Jilin Univ.(Sci. Ed.)*, vol. 55, no. 3, pp. 664–672, 2017.
[30] Z. Wang, S. Wang, L. Bai, W. Wang and Y. Shao, "Semisupervised fuzzy clustering with fuzzy pairwise constraints," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 9, pp. 3797–3811, 2022.
[31] L. Chen and Z. Zhong, "Adaptive and structured graph learning for semi-supervised clustering," *Information Processing & Management*, vol. 59, no. 4, pp. 102949, 2022.
[32] S. Yan, H. Wang, T. Li, J. Chu and J. Guo, "Semi-supervised density peaks clustering based on constraint projection," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 140–147, 2021.
[33] Y. Li, X. Fan and E. Gaussier, "Supervised categorical metric learning with schatten p-norms," *IEEE Transactions on Cybernetics*, vol. 52, no. 4, pp. 2059–2069, 2022.
[34] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[35] A. Amelio and C. Pizzuti, "Correction for closeness: Adjusting normalized mutual information measure for clustering comparison," *Computational Intelligence*, vol. 33, no.3, pp. 579–601, 2017.
[36] N. X. Vinh, J. Epps and J. Bailey, "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?," in *Proc. of the 26th Annual Int. Conf. on Machine Learning*, New York, NY, USA, pp. 1073–1080, 2009.