Kernelized Evolutionary Distance Metric Learning

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2019

Wasin KALINTHA

Abstract

Recently, clustering has played an important role in data mining and machine learning. Semi-supervised clustering is an extension of conventional clustering technique by integrating background information in the clustering, e.g., pairwise constraints or class labels. The conventional way to do semi-supervised clustering is Mahalanobis-based distance metric learning which penalizes objective function using the constraints satisfactory in order to find a suitable metric. Although, state-of-the-art semi-supervised clustering has a rich performance to improve the clustering accuracy by utilizing the class information from human intervention; however, it is reported that hard pairwise constraints, i.e., instance-level constraints, sometimes destroy the clustering quality, depending on relationship between the constraints and the data distribution and there is no monotonicity to the number of constraints, that is the improvement of cluster quality is not guaranteed by adding constraints. These drawbacks are critical issues in practice. Evolutionary distance metric learning (EDML) has been proposed to address the problem of instance-level constraints by directly improve cluster validity index, however, it is categorized as a linear distance metric learning, which yields a small benefit when the data is not linearly separable, like many other distance metric learning techniques. Even though many researchers proposed non-linear distance metric learning, it could not get away from the problem of instance-level constraints.

This study proposes a distance metric learning method which addresses the problem of non-linearly separable data and the problem of instance-level constraints simultaneously. Hence, this research provides an integration of kernelization technique with evolutionary distance metric learning called kernelized evolutionary distance metric learning (K-EDML). The proposed methods are able to handle either class labels or pairwise constraints and directly improve any clustering index as an objective function and can also perform a non-linear distance metric simultaneously. It can be viewed as utilizing cluster-level soft constraints, unlike other instance-level hard constraints which sometimes collapse the clustering. This research demonstrates the performance of the proposed method on UCI dataset compare with other well-known clustering and distance metric learning technique. As a result, the proposed method empirically overcomes other methods in many datasets and secure the highest average ranking in all dataset both in training and test sample. Moreover, the results demonstrate the benefit of kernelization in distance metric learning on the real-world dataset. The advantage of directly optimize the cluster validity index is illustrates by the improvement of cluster quality in EDML and K-EDML from

baseline and also state-of-the-art distance metric learning technique. In addition, the proposed method demonstrates generalize performance over the evaluation environment which different from training scheme. Finally, the proposed method maintains neighbor relation of clusters and can lead to a better visualization of the clustering result. Thus, it can be used as a novel cluster analysis technique that analyzes both class label and features sample simultaneously as a human-centered computing. This method is applied to the real-world problem of facial images and recipes data. The analysis provided promising insights, i.e., more intelligible cluster structure with neighbor relations can be obtained, and a particular cluster structure can be obtained according to the purpose of analysis.

Acknowledgement

The dissertation would not have been possible without supports and helps of my professors, family, and friends. I would like to express my very great appreciation to Professor Masayuki Numao for giving me the opportunities to study in Japan and the continuous support of my study and providing me with an excellent atmosphere for doing research. I would also like to express my sincere gratitude to my advisor Associate Professor Ken-ichi Fukui for his patience guidance, motivation, enthusiastic encouragement, immense knowledge, insightful comments, corrected my writing, and supported me through this research work.

My gratitude also goes out Professor Yasumasa Fujisaki and Professor Jun Tanida, the dissertion committee, for their great comments and contributions toward my doctoral degree.

I would also like to offer my special thanks to Associate Professor Satoshi Ono and Associate Professor Koichi Moriyama for his valuable and constructive suggestions throughout this research work. I would like to acknowledge the laboratory secretaries Ms. Misuzu Yuuki, Ms. Megumi Tanabe, Ms. Azusa Hirabayashi, Ms. Mika Kusakabe, Ms. Mitsuyo Otsuka, Ms. Akiko Yamamoto, and Ms. Makiko Kuroki for their accommodating and assisting me with all of the university affairs.

I would also like to thank The Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan for their continuous support in term of Japanese Government (MEXT) scholarship.

My sincere thanks also go to Dr. Nattapong Thammasan, Dr. Kamalas Udomlamlert, Dr. Songpol Chaunchaiyakul, Juan Lorenzo Hagad, Sirawit Sopchoke, Noppayut Sriwatanasakdi, Ekasit Phermphoonphiphat, Bassel Ali, ex- and present students under Professor Numao's laboratory, for their encouragement, guidance, and insightful comments during the work. I would also like to thank Evolutionary Distance Metric Learning research group for continuous supports in term of research funding and discussion.

Finally, I wish to thank Asawin Kalintha, Chintana Kalintha, Wisana Kalintha, Wisanee Kalintha, my family, friends, and Thai students' association in japan under the royal patronage (TSAJ) for their support and encouragement throughout my study and life in Japan.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human has the ability to divide objects into groups (clustering) and assign an object to a particular group (classification) by their common senses. This concept has gained much attention in data mining and machine learning, in order to automatically find clusters or assign a particular object to groups.

Recently, clustering [54] has played an important role in data mining and machine learning. Clustering divides data into clusters (group) according to their similarities between each other. Depending on the purpose of clustering, it could obtain meaningful groups (e.g., capture the nature structure of the data), beneficial group, which is a useful initialization for another approach, or sometimes both. It has been extended to a wide varieties of practical problems, such as, to find groups of similar functions genes in biology [52, 31], to use for information retrieval [74, 7, 75] in search engine even in Google[1][13], which is the current best search engine in the world, to find the pattern of atmosphere and ocean to predict the significant climate impact [48, 23], and to cluster patients who have similar symptom into subgroups to indicate the disease [59, 4, 27, 1] and so on. Clustering algorithms are generally introduced only in an unsupervised learning. In real-world application domains, the data usually contain some background information, e.g., pairwise constraint or class information (label). However, the traditional clustering algorithms have no way to make use of this information even it exists. Hence, the cluster structure is constructed only in the learning perspective, which often considers incomprehensible to human.

Semi-supervised clustering [5, 77, 33] is proposed to take advantage of this information by integrating background information in the clustering. Traditionally, to overcome this weakness, Xing et al. have attempted to learn a similarity metric from side information [77], such as constraints on which pairs of documents must or must not appear in the same cluster [66], so that

[1] https://www.google.com

the ideal clustering can be produced. Semi-supervised clustering has emerged as an interesting alternative in the last years. These algorithms improve the clustering quality through external knowledge conveyed in the form of constraints. These constraints are used to guide the clustering process and can be directly derived from original data (using partially labeled data) or provided by a user, trying to adapt clustering results to his/her expectations [18].

The idea to learn a similarity metric from side information is expanded to other fields not limited to only clustering. Many research well validates the metric, both empirically and theoretically, that the definition of distance between two data points significantly affects clustering and classification tasks. Recently, varieties of distance metric learning (DML) methods have been proposed by learning a distance metric from a dataset [79, 68, 49]. Example approaches include nearest neighbor classification [73, 40], clustering [77, 33, 51], and data visualization [30, 39].

Meanwhile, conventional semi-supervised clustering methods [66, 77, 11] try to improve clustering based on pairwise constraints—i.e., *must-links* that identify pairs of data that must be in the same class and *cannot-links* that identify pairs that must not be in the same class. These pairwise constraints can be written in binary, and can therefore be specified as a matrix; furthermore, the optimization problem is solved in such forms as semi-definite programming (SDP).

However, the conventional semi-supervised clustering methods have the following drawbacks. First, it is reported that hard pairwise constraints, i.e., instance-level constraints, sometimes destroy the clustering quality [19, 67], depending on a relationship between the constraints and the data distribution. Second, there is no monotonicity to the number of constraints, that is the improvement of cluster quality is not guaranteed by adding constraints. These drawbacks are critical issues in practice. Instead of including the pairwise constraints into the objective function as a penalty, Fukui *et al.* [29] propose evolutionary distance metric learning (EDML), a methodology to directly improve a cluster validity index such as purity, F-measure, and entropy. The EDML can be regarded as based on cluster-level soft constraints.

In general, a function of the clustering index is massively multimodal when distance metrics vary. Hence, EDML framework utilizes an evolutionary algorithm for this multimodality problem to search a sufficiently optimal metric transformation. The advantages of an evolutionary algorithm (EA) are as follows: (1) EAs can provide a solution even to problems that are hard to formulate using mathematical programming; (2) EAs sometimes heuristically discover unexpected solutions; and (3) EAs are highly parallelizable and can therefore make use of recent computational resources as multicore CPUs or PC clusters. Classic optimization methods such as gradient descent require a differentiable objective function, thus it addressed this by differential evolution (DE) [63] algorithm since DE does not require the optimization problem to be differentiable. Thus, their work utilizes differential evolution with self-adapting control parameters and generalized opposition-based learning (GOjDE) [69] for a real-valued optimization problem, which has a high search-ability without requiring parameter adjustments. Koloseni *et al.* [43] proposed the DML method using DE [63] for classification. EDML is for clustering, while their work supports the capability of applying DE to DML.

Although EDML provides outstanding results over other semi-supervised clustering in many datasets [36, 38, 37], it can only perform a linear transformation, like most of the DML techniques, which yields small benefit to non-linearly separable data because they are based on Mahalanobis distance. Also, non-linear approaches to metric learning have not managed to replicate this success [40]. Even though, many kernel-based distance metric learning approaches have been proposed to address non-linearly separable [81, 61, 14, 45], the drawbacks of conventional semi-supervised clustering methods are remained. Thus, this study proposes *kernelized evolutionary distance metric learning (K-EDML)*, a DML method which provides an integration of kernelization technique to address the problem of non-linear transformation while maintaining all properties of EDML simultaneously. Therefore, the non-linear transformation of the distance metric can be performed while maintaining the optimized cluster validity index by an evolutionary algorithm.

Moreover, this study takes advantage of the overall cluster structure evaluation by smoothed cluster validity index in the proposed method which refines neighboring cluster for better visualization, and the data points of the same class tend to locate in neighboring clusters in order to address the limitation of state-of-the-art cluster analysis, i.e., it cannot preserve class information as a neighborhood relation. Thus, this study proposed a cluster analysis technique that simultaneously visualizes the cluster structure by considering the class label, which is available from human intervention, simultaneously with the features to guide clustering to be constructed in the human point of view. By taking advantage of EDML, it allows us to propose this novel cluster analysis easily because of its capability that not only preserve the neighbor cluster relation, but also directly improve cluster accuracy in term of cluster validity index score.

This dissertation is organized as follows. In Chapter 2, the related literatures about basis of the kernelized evolutionary distance metric learning are reviewed, i.e., distance metric learning, semi-suprevised clustering, kernelization, cluster validity indices with neighborhood smoothing, and differential evolution. Moreover, literatures about cluster analysis and human-certered computing are covered in this chapter. Then, the overview of evolutionary distance metric learning framework is presented in Chapter 3. In Chapter 4, the kernelized evolutionary distance metric learning is proposed. Start from the cluster structure learning, then unifying the kernelization technique into the evolutionary distance metric learning. The proposed method is evaluated by varieties of real-world data from UCI. Moreover, the proposed method is compared with the state-of-the-art distance metric learning technique both linear and non-linear method to show the effectiveness of the proposed method. Then, in Chapter 5, the proposed method is applied to the real-world application: facial images dataset and recipe dataset. Cluster structure is visualized by a K-means clustering with K-nearest neighbor centroids graph in order to explain how the proposed method aid the cluster analysis. The analysis covers the relation between class information and features of the data, comprehensiveness of cluster structure, neighborhood relations of cluster boundary, and changing the direction of the cluster analysis. The dissertation end with a conclusion and introduced some further study in Chapter 6.

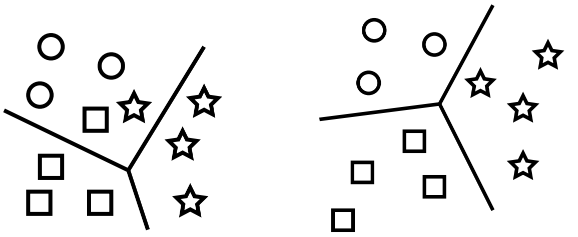
Chapter 2

Literature Review

2.1 Distance Metric Learning

Distance metric learning (DML) [79] attempts to optimize a metric to improve classification or clustering. Example approaches include nearest neighbor classification [73, 40], clustering [77, 33, 51], image ranking [15, 56, 47, 84, 82], and data visualization [30, 39].

Fig. 2.1a shows data points with three classes, i.e., circles, squares, and stars, and three initial partitions (or clusters) in Euclidean space. Note that one of the clusters has data points in all three classes. To cluster all data points correctly, the data space transformation stretches the partitions as shown in Fig. 2.1b



(a) Euclidean space (b) After transformation

Figure 2.1: Conceptual diagram of distance metric transformation

Distance metric learning can be divided into unsupervised, supervised, and semi-supervised learning by the amount of label information:

- **Unsupervised DML** attempts to identify geometric relationships in the Euclidean data space. Normally, unsupervised DML methods are viewed as dimensional reduction or projection into low-dimensional space while preserving neighbor relations of data points. The classical method of multidimensional scaling (MDS) fall into this category. ISOMAP [64], local linear embedding (LLE) [58], and Laplacian eigenmaps [8]. They can also be called *manifold learning*, which is an approach to learning the nonlinear structure of the data distribution.

- **Supervised DML** attempts to learn a distance metric transform function based on auxiliary information, including class labels and pairwise constraints of must-links and cannot-links. So far, a lot of supervised DML algorithms have been proposed, such as Mahalanobis distance learning model which first proposed by Xing et al. [77], Distance Metric Learning for Large Margin Nearest Neighbor Classification(LMNN) [73], linear DML for ranking (LDMLR) [76], Online Algorithm for Scalable Image Similarity learning (OASIS) [15], DML using dropout [56], and Geometric Mean Metric Learning (GMML) [84].
- **Semi-supervised DML** combines an advantage of supervised and unsupervised DML, which attempts to use unlabeled data to help supervised metric learning which has limited of auxiliary information to learn an appropriate metric such that it satisfies the constraints. Example algorithms are MPC-Kmeans [11], Information-Theoretic Metric Learning (ITML) [21], Hierarchical Confidence-based Active Clustering with Metric learning [51], An Intrinsic Approach for Semi-supervised Distance Metric Learning [82].

Moreover, DML can be viewed in another perspective as global and local DML: *Global distance metric learning* [77, 6, 30, 21, 73, 85, 15, 10, 56, 45, 84, 51, 82] has a common metric transformation in a whole data space, and attempts to learn the optimal transformation by preserving all the elements of classes close to each other while separating different classes. Conversely, *local distance metric learning* [80, 73, 71] attempts to locally satisfy the constraints rather than simultaneously satisfy all constraints. This locality is particularly useful for information retrieval and *k*-nearest neighbor classifiers. Although local DML methods have rich representation capability, they also have a tendency to over-fit owing to the high dimensionality to learn [80] while the global DML methods have relatively high constraints and are resistant to over-fitting.

In addition, there are several nonlinear methods which learn more flexible metrics in order to fit into non-linearly separable data [81], for instance, the kernelization technique aid the linear learning algorithm by a implicit nonlinear mapping function, e.g., a nonparametric kernel matrix [61], Semi-supervised Kernel k-means (SS-K-KMN) [44], and Kernel-Based Distance Metric Learning for Ranking [45]. Moreover, a nonlinear distance metric could be learned via nonlinear Gradient Boosting Regression Trees (GBRT) [40] or a deep feedforward neural network [72].

The proposed K-EDML has a distance metric learning as a basis and it is categorized as a semi-supervised non-linear global DML.

2.2 Semi-supervised Clustering

Semi-supervised clustering tries to improve clustering quality aiding by external knowledge, mostly pairwise constraints. COP-Kmeans [66] is the first attempt to introduce pairwise constraints to K-means clustering. The cluster assignments of data points are forcibly modified to satisfy the constraints (hard-constraints), and the centroids are updated based on the modified assignments. MPC-Kmeans [11] uses soft-constraints to allow violating some constraints, and integrates with DML as well. Then, Information-Theoretic Metric Learning (ITML) [21] uses the LogDet divergence regularization which will later be used in several other Mahalanobis distance learning methods. Hierarchical Confidence-based Active Clustering with Metric Learning (HCAC-ML) [51] is one of the successor of ITML which generate the constraints from hierarchical information and feed to ITML. In addition, there are several nonlinear methods which learn more flexible metrics, for example, a nonparametric kernel matrix [61], Semi-Supervised Kernel k-means (SS-K-KMN) [44].

As mentioned earlier, the instance-level constraints sometimes collapse the clustering. Davidson et al. [19] introduced *Coherence* that is the degree of agreement between the constraints to measure the property of a given set of constraints. Constraints with low coherence have contradictions in the data space. Thus it will be difficult to fully satisfy the constraints and can lead the clustering to undesirable result. Meanwhile, our EDML and K-EDML utilizes cluster-level constraints, which tries to satisfy the constraints as much as possible guided by a clustering validity index.

2.3 Kernelization

2.3.1 Kernel Function

Kernel trick is a technique to map the feature space to a higher dimensional feature space using a nonlinear function. This technique benefits the clustering task which is non-linearly separable. Given a dataset $\mathscr{D} = \{\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,v})^t \in \mathbb{R}^v\}_{i=1}^N$. By mapping points to feature space using basis function $\phi(\boldsymbol{x}_i)$, then replace a dot product $\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)$ with kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j) \tag{2.3.1}$$

For example the polynomial kernel function is as follows:

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\gamma \boldsymbol{x}_i{}^t \boldsymbol{\xi}_j + c)^d \tag{2.3.2}$$

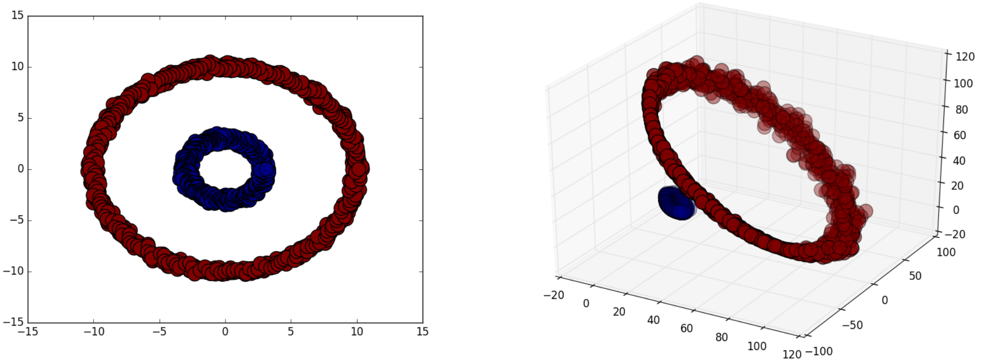
In two dimensional space, given $\gamma = 1$, $c = 0$ and $d = 2$, the basis function $\phi(\boldsymbol{x}_i)$ can derived as follows:

$$\begin{aligned} K(\boldsymbol{x}_i,\boldsymbol{x}_j) &= (\boldsymbol{x}_i{}^t \boldsymbol{x}_j)^2 \\ &= (x_{i,1}^2, \sqrt{2}x_{i,1}x_{i,2}, x_{i,2}^2) \cdot (x_{j,1}^2, \sqrt{2}x_{j,1}x_{j,2}, x_{j,2}^2) \\ &= \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j) \end{aligned} \tag{2.3.3}$$

Thus, the mapping function is

$$\phi(\boldsymbol{x}_i) = \phi(x_{i,1}, x_{i,2}) = (x_{i,1}^2, \sqrt{2}x_{i,1}x_{i,2}, x_{i,2}^2) \tag{2.3.4}$$

Fig. 2.2 shows the visualization of data space on synthetic data. Each point denotes the data points and color denotes the class of each point. In Fig. 2.2a, two classes are not linearly separable. On the other hand, Fig. 2.2b shows the transformed data space using polynomial mapping function in Eq. (2.3.4), this visualization shows the linearly separations of two classes.



(a) Original space

(b) Kernel space

Figure 2.2: Example of the mapping function of polynomial kernel on synthetic data

In addition, their are many works that proposed a kernel functions[62]. The kernel function utilized in this research are linear kernel, polynomial kernel, radial basis function (RBF) kernel, laplacian kernel, and hyperbolic tangent (sigmoid) kernel. The linear kernel is the simplest kernel function, it is given by inner product of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ plus an optional constants c. The kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is as follows:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = x_i^t x_j + c \tag{2.3.5}$$

For the radial basis function (RBF) kernel, an example of RBF kernel is a gaussian kernel. The RBF kernel function is

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\| x_i - x_j \|^2}{2\sigma^2}) \tag{2.3.6}$$

The adjustable parameter σ need to be carefully tune to make sure that it will not lose its non-linear power or sensitive to noise in training data. The laplacian kernel is similar to the RBF kernel with the square of the norm left out and it is less sensitive for change in the adjustable parameter σ. So the laplacian kernel can be written as follows:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\| x_i - x_j \|}{\sigma}) \tag{2.3.7}$$

Finally, Hyperbolic Tangent kernel or sigmoid kernel, which is oftened used as an activation function in Neural Networks field is as follows:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\alpha x_i^t x_j + c) \tag{2.3.8}$$

There are two adjustable parameters in the this kernel, the slope α and the intercept constant c. An inverse of number of dimension is a common value for α.

2.3.2 Kernel K-means Clustering (K-KMN)

Kernel k-means clustering (K-KMN) [25] is an enhancement of K-means clustering (KMN) that can extract non-linearly separable clusters in the original data space by applying a proper nonlinear mapping function (kernel) to a higher dimensional feature space. Given a dataset $\mathscr{D} = \{\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,v})^t \in R^v\}_{i=1}^N$ with cluster set $\mathbf{C}$, let the k^{th} cluster $C_k \in \mathbf{C}$. Using the non-linear function $\phi(\boldsymbol{x})$, the objective function of K-KMN is defined as:

$$\text{Minimize} \sum_{C_k \in \mathbf{C}} \sum_{\boldsymbol{x}_i \in C_k} \| \boldsymbol{\pi}_k - \phi(\boldsymbol{x}_i) \|_2^2 \tag{2.3.9}$$

Note that $\boldsymbol{\pi}_k$ denotes a centroid of cluster C_k on the mapped space. Then, the $\boldsymbol{\pi}_k$ is as follow:

$$\boldsymbol{\pi}_k = \frac{\sum_{\boldsymbol{x}_i \in C_k} \phi(\boldsymbol{x}_i)}{| C_k |}. \tag{2.3.10}$$

Here $\mid C_k \mid$ denotes the number of data points in cluster C_k. Since the basis function $\phi(\mathbf{x_i})$ is hard to obtain, kernel function $K(\mathbf{x_i},\mathbf{x_j}) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$ is calculated instead.

$$\begin{aligned}
\parallel \pi_k - \phi(\boldsymbol{x}_i) \parallel_2^2 &= \parallel \frac{\sum_{\boldsymbol{x}_j \in C_k} \phi(\boldsymbol{x}_j)}{\mid C_k \mid} - \phi(\boldsymbol{x}_i) \parallel_2^2 \\
&= \frac{\sum_{\boldsymbol{x}_j,\boldsymbol{x}_l \in C_k} \phi(\boldsymbol{x}_j) \cdot \phi(\boldsymbol{x}_l)}{\mid C_k \mid^2} \\
&- \frac{2\sum_{\boldsymbol{x}_j \in C_k} \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)}{\mid C_k \mid} + \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_i) \\
&= \frac{\sum_{\boldsymbol{x}_j,\boldsymbol{x}_l \in C_k} K(\boldsymbol{x}_j, \boldsymbol{x}_l)}{\mid C_k \mid^2} \\
&- \frac{2\sum_{\boldsymbol{x}_j \in C_k} K(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\mid C_k \mid} + K(\boldsymbol{x}_i, \boldsymbol{x}_i)
\end{aligned} \tag{2.3.11}$$

2.4 Cluster Validity Indices

Validation of clustering results is a fundamental but difficult issue because clustering often involves unsupervised learning and is essentially finding latent clusters in observed data[78]. This paper focuses on using external criteria, i.e., using class label, for evaluation of a clustering result.

Up until now, various validity measures have been proposed, and they are mainly separated into the following two types: (1) measures using internal criteria evaluate compactness and separability[22] of clusters based only on the distance between samples (e.g., Dunn-index[26], DB-index[20], and CDbw[32]) and (2) measures using external criteria evaluate how accurately the correct or desired clusters are formed in the clusters based on the class labels (i.e., categories) of samples, e.g., purity, entropy, F-measure, and mutual information[65, 78].

Set-Based Indices

Given dataset $\mathscr{D}$ with cluster set $\mathbf{C}$ and class set $\mathbf{T}$, let $N_{s,i}$ be the number of data points with class $s \in \mathbf{T}$ in the i^{th} cluster $C_i \in \mathbf{C}$; $N_{s,i} = \#\{\boldsymbol{x}_k | t(k) = s, c(k) = C_i\}$, where # denotes the number of elements, and $c(k)$ and $t(k)$ denote the cluster/class assignment for $\boldsymbol{x}_k$. N_i denotes the number of data points in cluster C_i; $N_i = \#\{\boldsymbol{x}_k | c(k) = C_i\}$, N is the total number of data points; $N = \#\{\boldsymbol{x}_k | \boldsymbol{x}_k \in \mathscr{D}\}$. These basic values are smoothed by a weighting function $h_{i,j}$ as follows:

$$N'_{s,i} = \sum_{C_j \in \mathbf{C}} h_{i,j} N_{s,j}, \tag{2.4.1}$$

$$N'_i = \sum_{s \in \mathbf{T}} N'_{s,i} = \sum_{s \in \mathbf{T}} \sum_{C_j \in \mathbf{C}} h_{i,j} N_{s,j}, \tag{2.4.2}$$

$$N' = \sum_{C_i \in \mathbf{C}} N'_i = \sum_{C_i \in \mathbf{C}} \sum_{s \in \mathbf{T}} \sum_{C_j \in \mathbf{C}} h_{i,j} N_{s,j}. \tag{2.4.3}$$

Here, the smoothing function $h_{i,j}$ can be any monotonically decreasing function, we use Gaussian function; $h_{i,j} = \exp(-d^c_{i,j}/\sigma^2)$, where $d^c_{i,j}$ denotes inter-cluster distance such as distance between cluster centroids and $\sigma (> 0)$ is a smoothing (neighborhood) radius.

By using eqs. (2.4.1) through (2.4.3) instead of the original values, any set-based clustering index, such as purity, F-measure, and entropy, can be extended as follows:

weighted purity (wPUR)

$$\mathrm{wPUR}(\mathbf{C}, \mathbf{T}) = \frac{1}{N'} \sum_{C_i \in \mathbf{C}} \max_{s \in \mathbf{T}} N'_{s,i} \tag{2.4.4}$$

weighted F-measure (wFME)

$$\mathrm{wFME}(\mathbf{C}, \mathbf{T}) = \sum_{s \in \mathbf{T}} \frac{N_s}{N} \max_{C_i \in C} F(s, C_i), \tag{2.4.5}$$

$$F(s, C_i) = \frac{2 \cdot Prec(s, C_i) \cdot Rec(s, C_i)}{Prec(s, C_i) + Rec(s, C_i)}, \tag{2.4.6}$$

where $Prec(s, C_i) = N'_{s,i}/N'_i$, $Rec(s, C_i) = N'_{s,i}/N_s$, and $N_s = \#\{\boldsymbol{x}_k | t(k) = s \in \mathbf{T}\}$.

weighted entropy (wENT)

$$\mathrm{wENT}(\mathbf{C}, \mathbf{T}) = 1 - \frac{1}{|\mathbf{C}|} \sum_{C_i \in \mathbf{C}} Entropy(C_i), \tag{2.4.7}$$

$$Entropy(C_i) = -\frac{1}{\log N'} \sum_{s \in \mathbf{T}} \frac{N'_{s,i}}{N'_i} \log \frac{N'_{s,i}}{N'_i}. \tag{2.4.8}$$

Pairwise-Based Index

Given class and cluster assignment of a data point $\boldsymbol{x}_i$ denoted as $t(i)$ and $c(i)$. Table 2.1 shows a class and cluster confusion matrix of data pairs, where a, b, c, d are the number of data pairs that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ do or do not belong to the same class/cluster.

Here, Fukui and Numao [28] introduced $likelihood(c(i) = c(j))$ indicating a degree that

Table 2.1: Class and cluster confusion matrix of data pairs

	$t(i) = t(j)$	$t(i) \neq t(j)$
$c(i) = c(j)$	a	b
$c(i) \neq c(j)$	c	d

a data pair $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belongs to the same cluster instead of the actual number of data pairs. The likelihood is given by a weighting function based on inter-cluster distance of the data pair; $likelihood(c(i) = c(j)) = h_{c(i),c(j)}$.

Then, a, b, c, d are replaced by summation of the likelihoods as follows:

$$a' = \sum_{\{i,j|t(i)=t(j)\}} h_{c(i),c(j)}, \tag{2.4.9}$$

$$b' = \sum_{\{i,j|t(i)\neq t(j)\}} h_{c(i),c(j)}, \tag{2.4.10}$$

$$c' = \sum_{\{i,j|t(i)=t(j)\}} \left(1 - h_{c(i),c(j)}\right) = a + c - a', \tag{2.4.11}$$

$$d' = \sum_{\{i,j|t(i)\neq t(j)\}} \left(1 - h_{c(i),c(j)}\right) = b + d - b'. \tag{2.4.12}$$

With these extended a', b', c' and d', weighted pairwise F-measure is defined as follows:

weighted pairwise F-measure (wPFM)

$$\text{wPFM}(\mathbf{C}, \mathbf{T}) = \frac{2 \cdot P \cdot R}{P + R}, \tag{2.4.13}$$

where $P = a'/(a' + b')$ is a weighted precision and $R = a'/(a' + c')$ is a weighted recall. The conventional precision is a ratio of the data pairs belonging to the same class within the same cluster. Likewise, the conventional recall is the data pairs belonging to the same cluster within the same class. The weighted precision and recall are extended to calculate the degree of belonging to the cluster/class by neighborhood relation of clusters.

2.5 Differential Evolution

Differential evolution (DE) [63] is a population-based meta-heuristics approach for solving real-valued optimization problems. Fig. 2.3 shows the population-based probabilistic search on the landscape of EDML fitness function. Each circle represents individual population in the search space. DE requires less user's interaction by requiring minimal gene selection operator(s) and control parameter adjustments. Furthermore, DE performs better than real-valued genetic algorithms or evolution strategies and is therefore applied to various optimization problems [55].

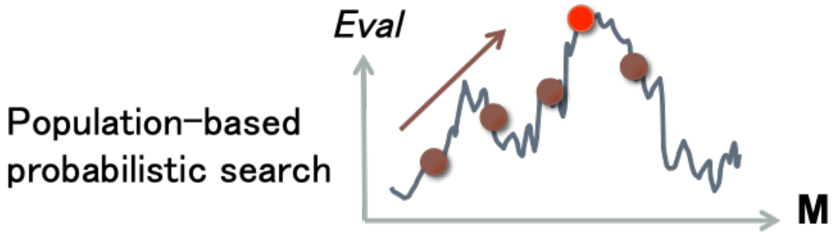


Figure 2.3: Conceptual diagram of population-based probabilistic search on the landscape of EDML fitness function

The proposed method makes use of a variant of DE, generalized opposition-based jDE (GOjDE) [69], which is an extension of self-adaptive differential evolution (jDE) [12]. They do not require any adjustment to crossover rate *CR* and scale factor *SF* by randomization of the parameters while searching. Because each individual has its own *CR* and *SF* values, GOjDE/jDE allows individuals that have better values to lead to better individuals in the next generation owing to their higher survivability rates. Also, jDE showed the best performance in “Evolutionary Computation in Dynamic and Uncertain Environments” in CEC2009 [46].

GOjDE employs generalized opposition-based learning (GOBL) for population initialization and for population jumping during the optimization. GOBL improves the search performance of DE for functions whose global optimum is around the center of its search space, in particular [57, 70]. Because most non-diagonal elements of optimal solutions in EDML become zero, which implies the solutions located around the center of the search space, GOjDE is suitable for this problem. The GOjDE algorithm is summarized as follows:

Step 1: Initialization

Randomly generate N_P individuals with m-dimensional vector $\boldsymbol{p}_{i,g} = \{p_{1,i,g}, p_{2,i,g}, \ldots, p_{m,i,g}\}(i = 1,2,...,N_P)$ within each domain of the definition, and set the generation number as $g = 0$. Then, create an opposition population using GOBL by the following equations:

$$p^*_{j,i,g} = k(a_{j,g} + b_{j,g}) - p_{j,i,g} \ (j = 1,2,\ldots,m) \tag{2.5.1}$$

$$a_{j,g} = \min_i(p_{j,i,g}),\ b_{j,g} = \max_i(p_{j,i,g}) \tag{2.5.2}$$

where $p^*_{j,i,g}$ denotes an opposite point calculated from a reference point $k(a_{j,g} + b_{j,g})$. In initialization, $a_{j,0}$ and $b_{j,0}$ are regarded as the min and the max of the defined range of j^{th} variable, respectively, and $k = 1$. Next, evaluate the fitness (i.e., one of the smoothed clustering indices mentioned above) for individuals in the original and opposite populations, and select top N_P individuals to the next population.

Step 2: Termination determination

When a termination condition is satisfied, the process terminates.

Step 3: Operation selection

Select GOBL operation (go to Step 4) with the probability τ_o, otherwise perform jDE operations (go to Step 6).

Step 4: GOBL

Randomly determine k within the range of $[0,1]$. Then, create opposition population by eq. (2.5.1). The change of k produces various reference points, allowing the population to jump to another place in the search space; the closer to $k=1$, the opposition population is generated inside the population in previous generation, while the closer to $k=0$, the opposition population jumped to farther area from the previous population. If $p^*_{j,i,g}$ exceeds the defined domain range, $p^*_{j,i,g}$ is redetermined by $rand(a_{j,g}, b_{j,g})$.

Step 5: Evaluation and selection (GOBL)

Evaluate the fitness of individuals in the opposition population, and then select top N_P individuals from a union of previous and opposite populations to the next generation. Update the generation no. $g \rightarrow g+1$, and go back to Step 2.

Step 6: Control parameter update

Update scale factor SF_i and crossover rate CR_i of i^{th} individual by the following equations:

$$SF_{i,g} = \begin{cases} SF_l + rand_1 \cdot SF_u & \text{if } rand_2 < \tau_1 \\ SF_{i,g-1} & \text{otherwise} \end{cases} \tag{2.5.3}$$

$$CR_{i,g} = \begin{cases} rand_3 & \text{if } rand_4 < \tau_2 \\ CR_{i,g-1} & \text{otherwise} \end{cases} \tag{2.5.4}$$

where $rand_j$ $(j \in 1,2,3,4)$ are uniform random values ranging in $[0,1]$, τ_1 and τ_2 are probabilities changing $SF_{i,g}$ and $CR_{i,g}$, respectively, and SF_l and SF_u determines the range of scale factor values.

Step 7: Mutation

Let a *target vector* be the i^{th} individual $\boldsymbol{p}_{i,g}$ to be operated on. Select a *base vector* $\boldsymbol{p}_{b,g}$

from individuals, and generate a *mutant vector* $\boldsymbol{v}_{i,g}$ by

$$\boldsymbol{v}_{i,g} = \boldsymbol{p}_{b,g} + SF_{i,g} \cdot (\boldsymbol{p}_{r1,g} - \boldsymbol{p}_{r2,g}), \tag{2.5.5}$$

where $b \neq r_1 \neq r_2$ ($r_1, r_2 \in \{1, \cdots, N_P\}$ are randomly selected), and scale factor $SF_{i,g}(0 \leq SF_{i,g} \leq 1)$ is an important parameter to determine the search range.

Step 8: Crossover

Generate *trial vector* $\boldsymbol{u}_{i,g}$ by a crossover operation between target vector $\boldsymbol{p}_{i,g}$ and mutant vector $\boldsymbol{v}_{i,g}$. Eq. (2.5.6) shows a binomial crossover. Each element of $u_{j,i,g} \in \boldsymbol{u}_{i,g}$ $(j = 1, 2, \cdots, m)$ is determined with crossover rate $CR_{i,g}$ $(0 \leq CR_{i,g} \leq 1)$ and a randomly selected index j_{rand} $(1 \leq j_{rand} \leq m)$ as:

$$\boldsymbol{u}_{j,i,g} = \begin{cases} v_{j.i,g} & \text{if } rand[0,1] \leq CR_{i,g} \text{ or } j = j_{rand}, \\ p_{j,i,g} & \text{otherwise,} \end{cases} \tag{2.5.6}$$

where *rand*[0,1] is a uniformly distributed random numbers in the range [0,1].

Step 9: Repair

When the trial vector involves a violation against the constraint, the repair operation is executed in order to maintain the individual back on track, e.g., the repair operation of individual in EDML and K-EDML are explained in section 3.6.

Step 10: Evaluation and selection (DE)

Evaluate the fitness of the trial vectors and compare with each target vector, and then select the more fit vector as an individual to the next generation. Update the generation no. $g \rightarrow g+1$, and go back to Step 2.

2.6 Cluster Analysis

In data mining and machine learning, cluster analysis [54] has received much attention in recent years due to a capability to a wide variety of fields. Cluster analysis attempts to analyze the cluster structure according to clustering purposes: clustering for understanding and clustering for utility. In cluster analysis, data are identified and grouped according to the similarities; similar data group together in the cluster and vice versa. Moreover, cluster analysis has become a standard method for many practical problems. In biology, Biologists spent many years and obtain a few patterns of cell [52]. Guzzi *et al.* [31] combine cluster analysis with Bioinformatics analysis to help understanding cell mechanisms by explaining the relationship among genes and related molecular processes. In information retrieval, nowadays many data are flooded in the internet, obtain the beneficial information is a challenging task. Hence, many approaches using cluster analysis have been proposed [74, 7, 75], even in search engine like Google [13], cluster analysis helps produce much more satisfying search results than existing systems. Learning the nature of the natural disaster can result in prediction and saving human being. Liu and Roy [48] applied the cluster analysis in spatio-temporal nature of weather data to identify and preserve interesting phenomena in the weather data. Dey *et al.* [23] utilized clustering in the generic methodology for weather forecasting. Knowing the disease beforehand can result in prevent people in dead, many works in medical applied cluster analysis to cluster patients into subgroups that have similar symptom to indicate the disease [59, 4, 27, 1]. Or in the business, it can discover the insight (e.g., merchandises that customers usually buy together) to tailor marketing strategies [9].

2.7 Human-centered Computing

Computer takes an important role in most of the computing task nowadays. However, most current methodologies only focus on a particular technological domain while overlooking human domain yield unnatural results and difficult to use. Thus, only the experts who dedicate amounts of their time can take advantage of this technology. Human-centered computing (HCC) [35] aims to decrease the gap between multiple disciplines that are concerned both with understanding human beings and with the design of computational artifacts by studies the design, development, and deployment of mixed-initiative human-computer systems. HCC has been facilitated in many aspects (i.e., personal, social [41], and cultural [34]) and addresses many problems, such as data analysis [16], the interaction between computer-human or human-human. In this study, evolutionary distance metric learning [29] is utilized as HCC for cluster analysis in Chapter 5.

Chapter 3

Evolutionary Distance Metric Learning

3.1 Overview

Evolutionary Distance Metric Learning (EDML) is originally proposed by Fukui *et al.* in 2013 [29]. EDML is an efficient framework that applies an evolutionary algorithm (EA) to firmly search a sufficiently optimal distance metric transformation matrix, also known as, a distance metric learning method (DML). EDML is based on a clustering index with neighbor relation that simultaneously evaluates inter- and intra-clusters to improve clustering quality. In contrast to other semi-supervised clustering, which formulate a penalty function for constraints into an objective function, EDML, however, directly improves the cluster validity index, such as purity, F-measure, or entropy, depending on the clustering purpose, as an objective function, when class information is available. Moreover, the cluster validity index is smoothed by neighbor relations which refines neighboring cluster for better visualization and the data points of the same class tend to locate in neighboring clusters.

3.2 Global Distance Metric Learning

In this work, a Mahalanobis-based distance is used just as in the case of many global DML methods. Given a dataset $\mathscr{D} = \{\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,v})^t \in \mathbb{R}^v\}_{i=1}^N$, the Mahalanobis-based distance can be defined as:

$$d_{i,j}^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^t \mathbf{M} (\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{3.2.1}$$

where $\mathbf{M} = (m_{k,l})$ is a $v \times v$ matrix. In the original Mahalanobis distance, $\mathbf{M}$ is given by the inverse of the variance-covariance matrix of the input data, i.e., $\mathbf{M} = \boldsymbol{\Sigma}^{-1}$. While in DML, the elements of $\mathbf{M}$ are variables to be learned that represent a transformation of the input data, in this case, $\mathbf{M}$ must be a symmetric positive semi-definite matrix to satisfy the distance propositions. For further understanding, eq. (3.2.1) can be rewrited as follows:

$$d_{i,j}^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^t \mathbf{M} (\boldsymbol{x}_i - \boldsymbol{x}_j) = \sum_{k,l} m_{k,l} (x_{i,k} - x_{j,k})(x_{i,l} - x_{j,l}), \tag{3.2.2}$$

in which diagonal elements of $\mathbf{M}$ (where $k = l$) indicate scaling for each dimension, whereas non-diagonal elements indicate correlation between different dimensions. Obviously, when $\mathbf{M}$ is a unit matrix, the Mahalanobis-based distance is equivalent to the Euclidean distance.

3.3 Cluster Structure Learning

EDML could be applied to any partition-based clustering with neighbor relation. Originally, self-organizing map (SOM) [42] has been used as a base clustering method as well as for visualizing DML results in EDML (EDML-SOM). For this work, however, K-means clustering with K-nearest neighbor centroids graph (KMN-KNN) was chosen instead, indicated by EDML-KMN, in order to preserve internal cluster validity index namely sum of the squared error (SSE), which is the K-means clustering (KMN) objective function. As a result, EDML-KMN can either optimize internal and external criteria simultaneously. Hence, an entire the data structure can be easily studied via inter-cluster connectivity and also visualize neighbor relations among cluster centroids.

3.3.1 K-means Clustering

K-means clustering (KMN) is one of the most frequently used clustering method because of its simplicity. The algorithm attempts to assign the data point to the nearest centroid in order to minimize sum of the distance, which in this study is calculated according to Eq. (3.2.1), between each data point and its nearest centroid. First, K initial centroids are randomly generated, namely, the number of clusters. Each point is assigned to the closest centroid, and the points assigned to the identical centroid form one cluster, totally K clusters. The centroid is updated based on the point. The algorithm repeatedly assigns the data point and recalculate the centroid until all data points and centroids converge. In other words, it tries to minimize the within-cluster SSE as the following equation:

$$\text{Minimize} \sum_{i=1}^{K} \sum_{x_j \in C_i} d_{x_j,\mu_i}^2 \tag{3.3.1}$$

where x_j is the data point that belong to the cluster i and μ_i is the cluster centroid.

3.3.2 Mutual K-nearest Neighbor Centroids Graph

Mutual K-nearest neighbor centroids graph (KNN) is a technique to obtain K nearest mutual neighborhood relations of each cluster centroid. The algorithm calculates the distance between every centroid, obtained from KMN, and selects k nearest centroids for each centroid. If each centroid pair has a mutual relation, then they form a mutual neighborhood relation. Hence, the mutual neighborhood relation can be represented by this adjacency matrix $\mathbf{A} = (a_{i,j})$ as follows:

$$a_{i,j} = \begin{cases} 1 & \text{if } C_j \in O(C_i) \text{ and } C_i \in O(C_j) \\ 0 & \text{otherwise} \end{cases} \tag{3.3.2}$$

where $O(C_i)$ denotes a set of k-nearest neighbor clusters from cluster i^{th} centroid C_i. Then, distance matrix is calculated based on adjacency matrix $\mathbf{A}$. For mutual neighborhood centroids ($a_{i,j} = 1$), the distance between them can be given by Euclidean distance, Mahalanobis distance, or 1 (number of hops). The distance between every nonmutual neighborhood centroids can be calculated by topological distance. In this case, topological distance is the shortest path between cluster centroids, and the shortest path of all pairs are calculated by Floyd-Warshall algorithm. In this research, the distance between mutual neighborhood centroids is given by a number of hops in order to normalize the distance between centroids in every data space.

This graph is simultaneously generated during the cluster structure learning because it is utilized in the neighborhood smoothing in cluster validity index. Unlike, another clustering method which cannot visualize cluster structure during the clustering.

3.3.3 Visualization

K-nearest neighbor centroids graph has been used with K-means clustering denotes KMN-KNN in order to easily analyze obtained cluster structures, especially in the case of micro-clusters or cluster number is larger than class number. With this, the entire data structures can be studied via the inter-cluster connectivity and also visualize neighbor relations among cluster centroids. The KNN is visualized using Cytoscape[1], the open source platform for complex network analysis and visualization. Cytoscape core distribution provides a basic set of features for data integration, analysis, and visualization. the KNN centroids are imported to Cytoscape using edge-weighted spring embedded layout.

[1] http://www.cytoscape.org/

3.4 Clustering validity index with neighborhood smoothing

EDML focuses on using external criteria, that is provided by human interpretation of data. It is more beneficial to use external criteria because class labels are available in this study. This cluster validity index is used as the evaluation of objective function in EDML.

This study focuses on using extended F-measure or F_1score by using the class label as external criteria for evaluation of a clustering result. In order to evaluate the overall cluster structure, this research used neighborhood smoothing in the cluster validity index by adding a weighting function $h_{i,j}$, unlike conventional clustering validity that can evaluate individual cluster quality, which is proposed by Fukui and Numao [28]. Weighted Pairwise F-measure (*wPFM*) in Eq. 2.4.13 is mainly used in this study for the fair evaluation to other comparison methods which utilize only pairwise constraints. The parameters for neighborhood smoothing are as follows. The weighting function $h_{i,j}$ is a Gaussian function. Therefore, $h_{i,j} = \exp(-r_{i,j}/\sigma)$, where $r_{i,j}$ denotes the inter-cluster distance between C_i and C_j, and $\sigma(>0)$ is a smoothing (neighborhood) radius.

3.5 Objective Function

EDML approach optimizes a cluster validity index *Eval* as follows:

$$\begin{aligned} \text{Maximize} \quad & Eval(Clustering(d^2_{i,j})), \\ \text{s.t.} \quad & |m_{k,k}| \geq \sum_{l(k\neq l)} |m_{k,l}|,\ 0 < m_{k,k} \leq 1,\ -1 \leq m_{k,l} \leq 1\ (k \neq l), \end{aligned} \tag{3.5.1}$$

where $Clustering(d^2_{i,j})$ denotes a clustering result by using a distance metric $d^2_{i,j}$—i.e., $Clustering() : \boldsymbol{x} \mapsto c \in \mathbf{C}$, where $\mathbf{C}$ is a set of cluster identifier. For constraint condition, in order to satisfy the proposition of distance metric, a condition of matrix $\mathbf{M}$ is set to be a weak *diagonally dominant matrix*—i.e., $|m_{i,i}| \geq \sum_{j(i\neq j)} |m_{i,j}|$ and the diagonal elements must be positive, to ensure that $\mathbf{M}$ is a positive semi-definite matrix.

3.6 Evolutionary Algorithm

EDML is focused on applying to real-world problems, which typically involve a gigantic high-dimensional data. High-dimensional global optimization is one such high-complexity problem. Therefore, self-adapting control parameters and generalized opposition-based differential evolution (GOjDE) [69], which explained in section 2.5, is used to optimize the objective function

(3.5.1) in order to manage the quality of candidate solutions. Here an upper bound on the generation number or fitness evaluation number value is used as a termination condition in GOjDE. The matrix **M** is an individual, and elements in a triangular matrix of **M** correspond to a gene, for example, in the two-dimensional case, the individual vector for **M** is $(m_{1,1}, m_{1,2}, m_{2,2})$. The repair process in Step 9 of GOjDE is occurred during the mutation, the variables in the vector are uniformly repaired unless the diagonally dominant matrix condition in function (3.5.1) is satisfied. The following eqation is a repair process for the individual in EDML.

$$m_{i,j}^{repair} = \frac{m_{i,i}}{\sum_j |m_{i,j}|} m_{i,j}, \ (i \neq j). \tag{3.6.1}$$

3.7 Evolutionary Distance Metric Learning Framework

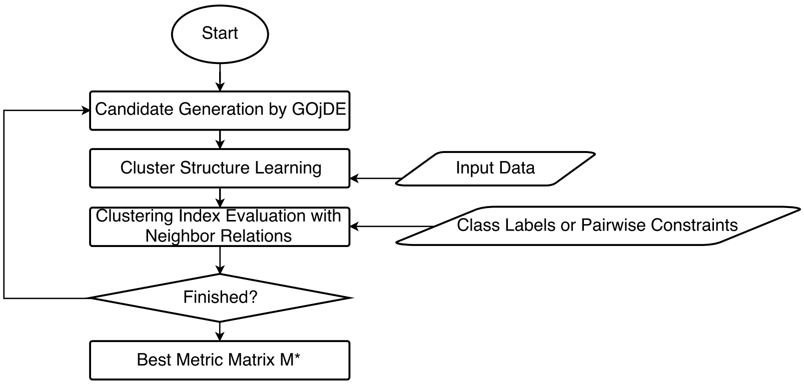

Start
Candidate Generation by GOjDE
Cluster Structure Learning
Input Data
Clustering Index Evaluation with Neighbor Relations
Class Labels or Pairwise Constraints
Finished?
Best Metric Matrix M*

Figure 3.1: Flowchart of the evolutionary distance metric learning (EDML)

The EDML framework is summarized in Fig. 3.1. First, the candidates of metric matrix **M** are generated by evolutionary algorithm, e.g., DE, jDE, GOjDE. Next, the cluster structure—i.e., clusters with neighbor relations—is obtained with a distance metric transformed by eq. (3.2.1). Here the cluster structure can be obtained by any partition-based clustering technique with neighborhood relation, such as k-means with a k-nearest neighbor graph of cluster centroids, or vector quantization with topology preservation by Self-Organizing Map (SOM). After obtaining the cluster structure with a transformed distance metric, the quality of the clusters and neighbor relations is evaluated with class labels or pairwise constraints via the smoothed clustering index; one of the weighted purity, weighted F-measure, weighted entropy, or weighted pairwise F-measure indices are used. Next, the evaluation value is fed back into GOjDE as the fitness for the candidate metric matrix. GOjDE selects individuals for the next generation on the basis of the fitness and generates the next candidates by mutation and crossover with certain probabilities. These steps are repeated until the termination condition is satisfied. The output is the best metric matrix $\mathbf{M}^*$ in terms of the smoothed clustering index among the overall generations of candidates.

Algorithm 3.7.0.1 shows the pseudocode of EDML with the following configuration.

- Global distance metric: Mahalanobis-based distance metric
- Evolutionary algorithm: GOjDE
- Clustering algorithm: K-means clustering
- Cluster validity index: Weighted Pairwise F-measure ($wPFM$)

Algorithm 3.7.0.1 Evolutionary Distance Metric Learning (EDML)

```
    Input: $\mathscr{D}$: dataset, $\mathbf{T}$: class labels or pairwise constraints
    Output: $\mathbf{M}^*$: best metric matrix
$g \leftarrow 0$.
$maxEval \leftarrow 0$
Initialize candidate metric population $\boldsymbol{P}_g$ via GOjDE.
while $g < itr_{max}$ or $maxEval = 1$ do
    for $\forall \boldsymbol{p}_{c,g} \in \boldsymbol{P}_g$ do
        $\mathbf{M} \leftarrow \boldsymbol{p}_{c,g}$
        if $|m_{k,k}| \geq \sum_{l(k \neq l)} |m_{k,l}|, 0 < m_{k,k} \leq 1,\ -1 \leq m_{k,l} \leq 1\ (k \neq l)$ then // weak diagonally dominant matrix
            Repair $\mathbf{M}$ using (3.6.1).
        end if
        $d_{i,j}^2 \leftarrow (\boldsymbol{x}_i - \boldsymbol{x}_j)^t \mathbf{M} (\boldsymbol{x}_i - \boldsymbol{x}_j)$ using (3.2.2).
        $Clustering(d_{i,j}^2) \leftarrow$ k-means clustering using $d_{i,j}^2$
        $eval \leftarrow$wPFM($Clustering(d_{i,j}^2)$ , $\mathbf{T}$) using (2.4.13).
        if $eval > maxEval$ then
            $maxEval \leftarrow eval$
            $\mathbf{M}^* \leftarrow \mathbf{M}$
        end if
    end for
    Crossover and Mutation $\boldsymbol{P}_{g+1}$ using (2.5.5) and (2.5.6) via GOjDE.
    $g \leftarrow g + 1$.
end while
return $\mathbf{M}^*$
```

Chapter 4

Kernelized Evolutionary Distance Metric Learning

4.1 Overview

Theoretically, the evolutionary distance metric learning (EDML) yields insignificant results in non-linearly separable data. Therefore, this study proposed an improvement of EDML, namely kernelized evolutionary distance metric learning (K-EDML) which is an integration of kernelization technique and EDML. While this proposed method maintains all the properties of EDML, unlike other kernelized DML [49] which are formulating a penalty function for constraints, i.e., must-link and cannot-link, into an objective function. Obviously, the K-EDML is equivalent to the EDML when the linear kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^t \boldsymbol{x}_j)$ is used as a kernel function.

4.2 Integrating Kernelization technique in K-EDML

The kernelization technique is unified in the cluster structure learning process. K-EDML could be applied to any partition-based kernel clustering with neighbor relation. In this study, kernel K-means clustering (K-KMN) [25], which is introduced in section 2.3.2, has been used as a base clustering method.

In order to integrate the kernelization technique into EDML, a symmetric positive semi-definite matrix $\mathbf{M}$ in Eq. (3.2.2) can be decomposed into $\mathbf{M} = \mathbf{L}^t\mathbf{L}$ by Cholesky decomposition, where $\mathbf{L}$ denotes an upper triangular matrix. Eq. (3.2.1) can therefore be rewritten as:

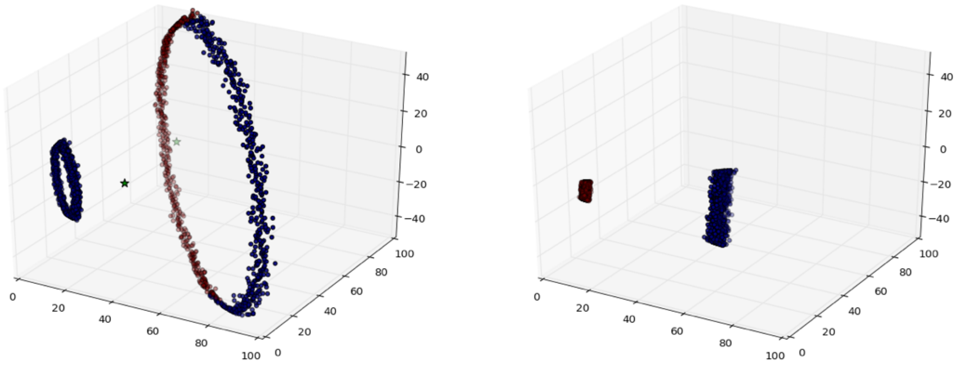
$$\begin{aligned} d_{i,j}^2 &= (\boldsymbol{x}_i - \boldsymbol{x}_j)^t \mathbf{M} (\boldsymbol{x}_i - \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^t (\mathbf{L}^t\mathbf{L})(\boldsymbol{x}_i - \boldsymbol{x}_j) \\ &= (\mathbf{L}\boldsymbol{x}_i - \mathbf{L}\boldsymbol{x}_j)^t (\mathbf{L}\boldsymbol{x}_i - \mathbf{L}\boldsymbol{x}_j) = \|\mathbf{L}\boldsymbol{x}_i - \mathbf{L}\boldsymbol{x}_j\|_2^2. \end{aligned} \tag{4.2.1}$$

Hence, the Mahalanobis-based distance can be viewed as the Euclidean distance after a linear transformation by $\mathbf{L}$. So, each decomposed obtained candidate $\mathbf{L}$ is individually multiplied to the original data, thus the input data $\boldsymbol{x}_i$ is substituted with $\mathbf{L}\boldsymbol{x}_i$. The transformed input is utilized into the mapping function $\phi(\boldsymbol{x}_i)$ in K-KMN objective function in Eq. (2.3.9). Let $\phi'(\boldsymbol{x}_i)$ denote the mapping function of the transformed data $\phi(\mathbf{L}\boldsymbol{x}_i)$, the K-KMN objective function utilized transformed data is as follows:

$$\text{Minimize} \sum_{C_k \in \mathbf{C}} \sum_{\boldsymbol{x}_i \in C_k} \| \pi_k - \phi'(\boldsymbol{x}_i) \|_2^2 . \tag{4.2.2}$$

4.3 Simple Example of K-EDML

Fig. 4.1 visualizes the concept of the proposed K-EDML. The original data is visualized in Fig. 2.2a, each color denotes each class. In order to see the difference in kernel data space, the original data to the kernel data space is mapped using the Eq. (2.3.4) as in Fig. 2.2b. Fig. 4.1a and Fig. 4.1b present the result of K-KMN and K-EDML respectively. Since the K-KMN cannot make use of provided class labels, also only the minimization of the distance between centroid and data points in Eq. (2.3.9) is preserved. Thus, K-KMN cannot correctly cluster the data even it has a linear separation which can be seen from the mixing of two clusters on the outer circle. In contrast, K-EDML can take advantage of class labels to preserve the clusters with the same class label by stretch the data space, i.e., the same class data move close together, and the different class moves apart which present in Fig. 4.1b.



(a) Kernel KMN

(b) K-EDML

Figure 4.1: Visualization of clustering results on kernel space

4.4 Kernelized Evolutionary Distance Metric Learning Framework

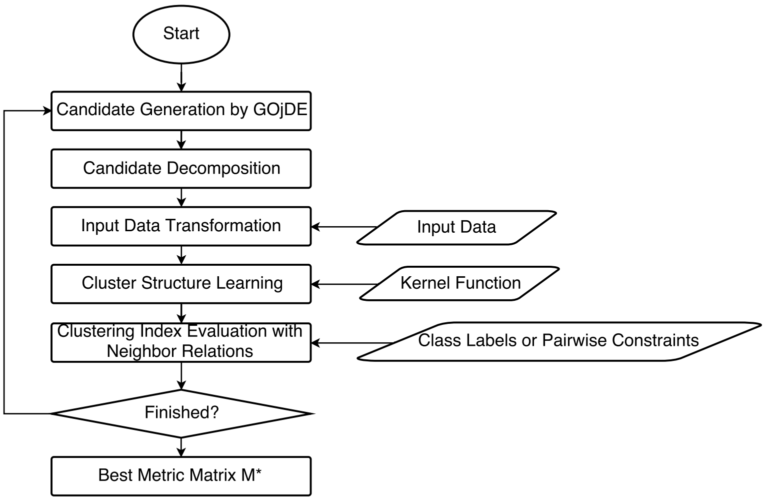

Start
Candidate Generation by GOjDE
Candidate Decomposition
Input Data Transformation
Input Data
Cluster Structure Learning
Kernel Function
Clustering Index Evaluation with Neighbor Relations
Class Labels or Pairwise Constraints
Finished?
Best Metric Matrix M*

Figure 4.2: Flowchart of the kernelized evolutionary distance metric learning (K-EDML)

Although the framework of K-EDML is similar to EDML, there are some modification in each step in order to integrate the kernelization technique. Fig. 4.2 summarised K-EDML framework. First, candidates of metric transform matrix $\mathbf{M}$ are generated using GOjDE. Further, the symmetric PSD matrix $\mathbf{M}$ can be decomposed into $\mathbf{M} = \mathbf{L}^t\mathbf{L}$ by Cholesky decomposition, where $\mathbf{L}$ denotes an upper triangular matrix. Mahalanobis distance in Eq. (3.2.1) can be rewritten as Eq. (4.2.1). Next, each decomposed obtained candidate $\mathbf{L}$ is individually multiplied to the original data, thus the input data $\boldsymbol{x}_i$ is substituted with $\mathbf{L}\mathbf{x}_i$. This transformed input is utilized into the K-KMN objective function Eq. (2.3.9) and used as $Clustering()$ in Eq. (3.5.1) Then, class labels are utilized in order to evaluate the quality of the cluster structure through the neighborhood smoothing in the clustering index. This is followed by feeding the evaluated values back into GOjDE as the fitness for each candidate $\mathbf{M}$. GOjDE selects candidates based on the fitness to evolve and generate the next candidates by mutation and crossover with certain probabilities. These steps are repeated until the termination condition (e.g., the iteration limit) is satisfied. Finally, the optimal metric transform matrix $\mathbf{M}^*$ is obtained in terms of the most smoothed clustering index among the overall generations of candidates.

Algorithm 4.4.0.1 shows the pseudocode of EDML with the following configuration.

- Global distance metric: Mahalanobis-based distance metric
- Evolutionary algorithm: GOjDE
- Clustering algorithm: Kernel k-means clustering

- Cluster validity index: Weighted Pairwise F-measure ($wPFM$)

Algorithm 4.4.0.1 Kernelized Evolutionary Distance Metric Learning

```
Input: $\mathscr{D}$: dataset, $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$: kernel function, $\mathbf{T}$: class labels or pairwise constraints
Output: $\mathbf{M}^*$: best metric matrix
$g \leftarrow 0$.
$maxEval \leftarrow 0$
Initialize candidate metric population $\boldsymbol{P}_g$ via GOjDE.
while $g < itr_{max}$ or $maxEval = 1$ do
    for $\forall \boldsymbol{p}_{c,g} \in \boldsymbol{P}_g$ do
        $\mathbf{M} \leftarrow \boldsymbol{p}_{c,g}$
        if $|m_{k,k}| \geq \sum_{l(k \neq l)} |m_{k,l}|, 0 < m_{k,k} \leq 1,\ -1 \leq m_{k,l} \leq 1\ (k \neq l)$ then // weak diagonally dominant matrix
            Repair $\mathbf{M}$ using (3.6.1).
        end if
        $\mathbf{L}^t\mathbf{L} \leftarrow \mathbf{M}$
        $d_{i,j}^2 \leftarrow \|\mathbf{L}\boldsymbol{x}_i - \mathbf{L}\boldsymbol{x}_j\|_2^2$ using (4.2.1).
        $Clustering(d_{i,j}^2) \leftarrow$ kernel k-means clustering using $d_{i,j}^2$ and $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
        $eval \leftarrow$wPFM($Clustering(d_{i,j}^2)$ , $\mathbf{T}$) using (2.4.13).
        if $eval > maxEval$ then
            $maxEval \leftarrow eval$
            $\mathbf{M}^* \leftarrow \mathbf{M}$
        end if
    end for
    Crossover and Mutation $\boldsymbol{P}_{g+1}$ using (2.5.5) and (2.5.6) via GOjDE.
    $g \leftarrow g + 1$.
end while
return $\mathbf{M}^*$
```

4.5 Experimental Design

The experiment used the following ten open datasets from the well-known UCI machine learning repository [24]; Iris, Glass, Wine, Segment, Vehicle, Balance, Pima, Yeast, Ionosphere, and Musk. Basic statistics of these datasets are summarized in Table 4.1. The attribute values were normalized such that each average is equal to zero and standard deviation is equal to one. Note that for the datasets that the number of attributes is greater than 10 are indicated by adding an asterisk (*) after their names in Table 4.1. Principal component analysis (PCA) was adopted to reduce the input dimension to 10.

Table 4.1: The basic statistics of UCI datasets

dataset	# samples	# attributes	# classes
Glass	214	9	6
Iris	150	4	3
Wine*	178	13	3
Vehicle*	846	18	4
Segment*	2310	19	7
Ionosphere*	351	34	2
Pima	768	8	2
Musk*	625	166	2
Balance	625	4	3
Yeast	1484	8	10

Table 4.2: Settings for K-means with K-nearest neighbor graph (KMN-KNN) and Kernel K-means kernel function, and the evolutionary algorithms (EA)

	KMN-KNN	#individuals in EA	#generations in EA
dataset	#clusters, #neighbors	full	
Glass	20, 5	90	6,000
Iris	20, 5	30	2,000
Wine	20, 5	182	4,000
Vehicle	20, 5	171	3,000
Segment	20, 5	190	2,000
Ionosphere	20, 5	165	2,000
Pima	20, 5	108	2,000
Musk	20, 5	165	2,000
Balance	20, 5	30	2,000
Yeast	20, 5	108	2,000

Table 4.2 shows the settings of the parameters for each dataset. The number of clusters was set to 20 in every dataset. In this experiment, it is not necessary to determine an appropriate number of clusters. The number of individuals in an evolutionary algorithm (EA) was determined depending on the size of search space. It sets 1D, where D is the dimension number of variables, for larger number of variables; Wine, Vehicle, and Segment with a full matrix, and the rest of cases use 2D to 5D. Also, the generation limit in EA was determined by checking convergence and by computational time consumption. As mentioned earlier, the parameters in GOjDE, *CR*

and SF are automatically adjusted while searching. Moreover, the optimal neighborhood radius in the smoothing function $h_{i,j}$ within a cluster validity index was also automatically determined in advance of performing EDML for each dataset and a type of validity index. The smoothing radius that maximizes $Eval$ compared to the randomized neighbor relations is assumed to be an optimal one; $\sigma^* = \arg\max_\sigma |Eval - \overline{Eval}_{rnd(n)}|$, where $\overline{Eval}_{rnd(n)}$ denotes an average of $Eval$s when inter-cluster distances $\{\boldsymbol{r}_{i,j}\}$ are n times shuffled.

In this experiment, the proposed EDML and K-EDML, and state-of-the-art distance metric learning methods are tested, as well as several baseline clustering. The comparison methods are as follows:

- Baseline
  - K-means clustering (KMN)
  - Kernel K-means clustering (K-KMN) [25]
- Distance Metric Learning
  - Information-Theoretic Metric Learning (ITML) [21]
  - Distance Metric Learning for Large Margin Nearest Neighbor Classification (LMNN) [73]
  - Geometric Mean Metric Learning (GMML) [84]
  - Gradient Boosted Large Margin Nearest Neighbors (GB-LMNN) [40]

Moreover, some popular semi-supervised clustering, i.e., COP-Kmeans [66], and clustering with distance metric learning methods, i.e., DML [77] and MPC-Kmeans [11], are omitted since other comparison methods overcome these baseline clustering and DML methods [21, 73, 40].

Fig. 4.3 summarizes framework of this experiment. For the fair comparison, the experiments performed under five-fold cross-validation. In the training process, each method produces 5 metrics matrix in each fold, the number of clusters and neighbors equal to 20 and 5 respectively. Label sampling rate for training data is set to 30%. During the kernel selection and hyper-parameter tuning, grid search with five-fold cross-validation was performed to achieve the suitable kernel and hyper-parameters. The suitable kernel was selected among polynomial kernel, radial basis function (rbf) kernel, laplacian kernel and sigmoid kernel. We omitted the linear kernel from K-EDML because the K-EDML with a linear kernel is analogous to EDML. Therefore, the results of K-EDML with a linear kernel can be view as EDML. For hyper-parameters tuning range in each kernel are listed as follows:

- Polynomial kernel:
  - degree $d = \{-10, -9, \cdots, 9, 10\}$
  - gamma $\gamma = \{-10, -9, \cdots, 9, 10\}$, $\{-1/10, -1/9, \cdots, 1/9, 1/10\}$, $\{10^{-10}, 10^{-9}, \cdots, 10^{9}, 10^{10}\}$
  - coefficient $c = \{-10, -9, \cdots, 9, 10\}$, $\{-1/10, -1/9, \cdots, 1/9, 1/10\}$, $\{10^{-10}, 10^{-9}, \cdots, 10^{9}, 10^{10}\}$
- Radial basis function (rbf) kernel:
  - sigma $\sigma = \sqrt{1/2\gamma}$
    where gamma $\gamma = \{-10, -9, \cdots, 9, 10\}$, $\{-1/10, -1/9, \cdots, 1/9, 1/10\}$, $\{10^{-10}, 10^{-9}, \cdots, 10^{9}, 10^{10}\}$
- Laplacian kernel:
  - sigma $\sigma = \{-10, -9, \cdots, 9, 10\}$, $\{-1/10, -1/9, \cdots, 1/9, 1/10\}$, $\{10^{-10}, 10^{-9}, \cdots, 10^{9}, 10^{10}\}$
- Sigmoid kernel:
  - alpha α = inverse of number of attibutes
  - coefficient $c = \{-10, -9, \cdots, 9, 10\}$, $\{-1/10, -1/9, \cdots, 1/9, 1/10\}$, $\{10^{-10}, 10^{-9}, \cdots, 10^{9}, 10^{10}\}$

Table 4.3 shows the selected kernel function and its tuned hyper-parameter in K-EDML for each dataset.

Table 4.3: The selected kernel function and tuned hyper-parameter used in K-EDML obtain from grid search with five-fold cross-validation

dataset	kernel function	degree	γ	σ	c
Glass	polynomial	3	10	-	10^{2}
Iris	laplacian	-	-	$1/4$	-
Wine	rbf	-	10^{-1}	-	-
Vehicle	polynomial	9	10^{4}	-	10^{4}
Segment	polynomial	3	10^{-7}	-	10^{4}
Ionosphere	polynomial	5	10^{6}	-	10^{5}
Pima	polynomial	5	10^{3}	-	10^{4}
Musk	polynomial	7	1	-	1
Balance	polynomial	9	10^{8}	-	1
Yeast	polynomial	3	10^{4}	-	10^{5}

Product from each method, i.e., cluster centriods in KMN and K-KMN and $\mathbf{M}^*$ in ITML, LMNN, GMML, GB-LMNN, EDML and K-EDML were carried out to the evaluation process afterwards.

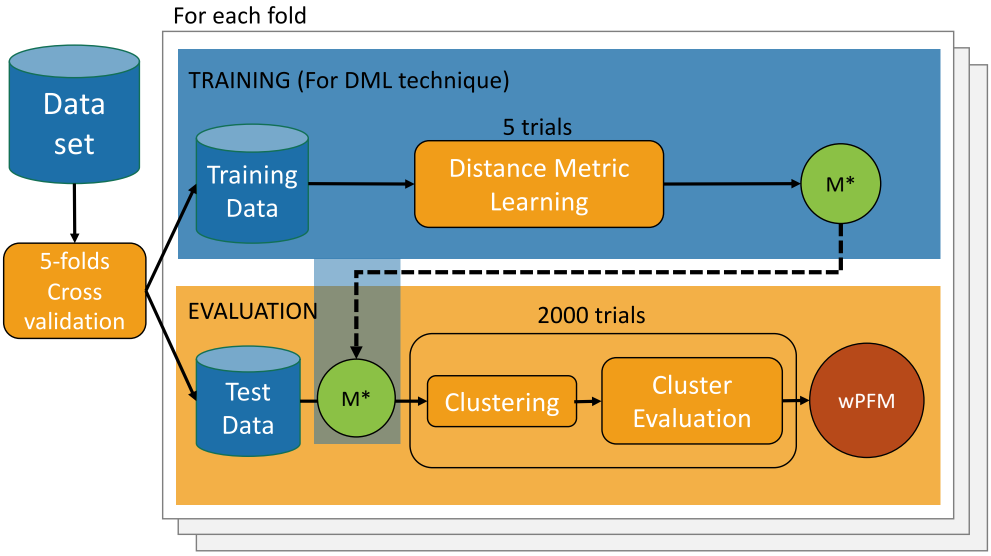


Figure 4.3: Framework of the experiment

4.6 Experimental Results

4.6.1 Computational time

This experiment conducted on Intel Xeon E5-2690v4 14-Core 2.6GHz x2. Since the EDML and K-EDML are compatible with parallel computing, they run with 56 threads. Table 4.4 shows the training time to obtain the $\mathbf{M}^*$ for each fold. Note that KMN and K-KMN are omitted because they do not need a training process. The EDML and proposed K-EDML are based on an evolutionary algorithm; therefore, they inherit drawback of an evolutionary algorithm in high complexity. As a result, both EDML and K-EDML are definitely cost more computational time compared to other methods which based on mathematical optimization; however, in the relatively large dataset, e.g., Vehicle, Segment, and Yeast, EDML can train faster than the mathematical optimization methods, due to parallel computing. Meanwhile, the proposed method is much slower than the EDML due to the kernel calculation.

4.6.2 Comparison with Other Semi-supervised Clustering Methods

Each method is evaluated base on their category, i.e., linear and non-linear distance metric learning. This research adopted k-means with k-nearest neighbor graph for linear technique

Table 4.4: The computational training time (per fold)

	ITML	LMNN	GMML	GB-LMNN	EDML	K-EDML
Iris	7.26s	0.37s	0.36s	7.30s	7.80s	70.63s
Wine	9.94s	0.17s	0.41s	10.86s	28.82s	198.27s
Glass	15.95s	0.17s	0.50s	18.92s	26.62s	259.29s
Vehicle	215.63s	0.27s	1.72s	197.19s	101.44s	73 hours 39 minutes
Segment	1611.93s	0.90s	8.06s	498.08s	496.35s	33 hours 53 minutes
Balance	108.38s	0.28s	1.09s	5.50s	17.39s	12 hours 20 minutes
Pima	177.81 s	0.30s	1.77s	102.42s	62.77s	35 hours 4 minutes
Yeast	658.70s	1.64s	4.16s	674.17s	168.63s	144 hours 15 minutes
Ionosphere	41.28s	0.14s	0.69s	46.06s	36.65s	9 hours 0 minute
Musk	68.41s	0.10s	0.94s	45.73s	47.68s	23 hours 30 minutes

(KMN, ITML, LMNN, GMML, EDML) and kernel k-means clustering with k-nearest neighbor graph with an identical trained kernel for non-linear methods (K-KMN, GB-LMNN, K-EDML). Weighted pairwise F-measure (wPFM) with the same configurations as in training process, the number of cluster and nearest neighbor equals to 20 and 5, was used to evaluating the clustering results. For simplicity, let wPFM@a/b denotes the wPFM which evaluated using the number of the cluster assignment equals to a and the number of nearest neighbor centroid equals to b. wPFM@20/5 is used as an objective function for EDML and K-EDML to optimize as in Eq. (3.5.1).

Table 4.5 and Table 4.6 present the five-fold cross-validation evaluation results in average and standard deviation of each clustering algorithm from 2000 trials. Observation indicates as follows: 1) ITML, LMNN, GMML, GB-LMNN, and the proposed method improve the clustering performance from the baseline clustering, i.e., KMN and K-KMN, due to the benefit of distance metric learning. 2) The proposed method overcome or at least comparable to other clustering methods in 8 datasets and obtain the highest clustering score in 5 datasets. These results secure the properties of the proposed method which utilize class label in order to directly improve the objective function Despite highest results from other methods in some dataset, they can only perform well in specific data, while the proposed method performs well in many datasets. The reason for the susspricious performance in Wine data of K-EDML is the selected kernel maybe not fully suitable for this data set because GMML which is a non-linear technique still obtains the highest score in this data. 3) The benefit of kernelization technique can be seen from the pair of the results, i.e., KMN and K-KMN, LMNN and GB-LMNN, and EDML and K-EDML, the kernel integrated technique yield higher result than the one without kernelization because the properties of the data that is non-linearly separable. However, EDML is still comparable to other methods. 4) EDML obtains higher wPFM than K-EDML in in some dataset, e.g., vehicle, pima, and balance dataset, because linear kernel is suitable for these dataset. These results of EDML can be viewed as the K-EDML with linear kernel. Even though ITML obtains the highest wPFM in these data set due to its superior performance in linearly separable data, K-EDML achieves the comparable score.

Since EDML can be viewed as a special case of K-EDML when using the linear kernel, the EDML is then merged to the K-EDML. The ranking is presented as Table 4.7. As a result, K-EDML secures the top 5 in all dataset and overcomes all other unsupervised and semi-supervised clustering methods in this paper with average ranking 1.8 and 2.1 for training and test sample. These results clearly illustrated the performance of the proposed method.

4.6.3 Evaluation via Standard Evaluation Criteria

Lastly, to make this experiment more practical, the number of clusters is set to equal the number of classes in an evaluation process. Then, standard pairwise F-measure (PFM) which can be seen equivalent to wPFM@#class/0 is used as the measurement criterion. Then the trained distance metric $\mathbf{M}^*$ from the training process is evaluated again. Table 4.8 and Table 4.9 present the evaluation results in average and standard deviation of each clustering algorithm. Table 4.10 shows the ranking of comparison results of selected K-EDML and other methods. The selected K-EDML still achieved similar results like in the previous evaluation at wPFM@20/5. The proposed method empirically overcomes other methods with the lowest average rank of 2.6 and 2.5 in both training and test dataset, also it still places in the top 4 in all dataset except Balance dataset. Since Balance dataset has an imbalance of the class distribution, this is possibly a reason why K-EDML is not performed well in this dataset. Even the number of the cluster in training and evaluation process is not identical which is the properties of the proposed method. These results clearly affirmed the performance and robustness of the proposed method, the benefit of neighborhood smoothing in cluster validity index which is an objective function and kernelization technique. Meanwhile, determining the number of clusters is still a hot topic nowadays.

Table 4.5: The average wPFM@20/5 with standard deviation of EDML, K-EDML and their comparison clustering methods on training data

	KMN	ITML	LMNN	GMML	K-KMN	GB-LMNN	EDML	K-EDML
Glass	0.416±0.01	0.418±0.01	0.416±0.01	0.422±0.01	0.471±0.02	0.443±0.02	0.425±0.02	**0.477±0.03**
Iris	0.568±0.08	0.619±0.09	0.644±0.09	0.641±0.09	0.675±0.09	0.606±0.09	0.642±0.09	**0.737±0.04**
Wine	0.520±0.02	0.638±0.13	0.628±0.12	0.552±0.07	0.556±0.07	**0.688±0.16**	0.553±0.07	0.574±0.09
Vehicle	0.395±0.00	**0.406±0.00**	0.396±0.00	0.396±0.00	0.399±0.00	0.396±0.00	0.400±0.01	0.399±0.00
Segment	0.261±0.01	0.294±0.02	0.271±0.02	0.286±0.05	0.450±0.06	0.318±0.04	0.338±0.06	**0.514±0.07**
Ionosphere	0.654±0.01	0.683±0.01	0.655±0.01	0.690±0.02	0.591±0.03	0.662±0.02	0.689±0.01	**0.704±0.01**
Pima	0.642±0.01	**0.664±0.01**	0.646±0.01	0.577±0.15	0.496±0.12	0.630±0.01	0.661±0.02	0.570±0.21
Musk	0.552±0.06	0.614±0.01	0.546±0.06	0.565±0.05	**0.666±0.02**	0.560±0.06	0.585±0.04	0.639±0.05
Balance	0.561±0.00	**0.601±0.01**	0.583±0.01	0.570±0.00	0.507±0.02	0.591±0.01	0.569±0.01	0.546±0.04
Yeast	0.366±0.00	0.369±0.00	0.363±0.01	0.370±0.00	0.364±0.00	0.371±0.02	0.383±0.03	**0.396±0.01**

Table 4.6: The average wPFM@20/5 with standard deviation of EDML, K-EDML and their comparison clustering methods which on test data

	KMN	ITML	LMNN	GMML	K-KMN	GB-LMNN	EDML	K-EDML
Glass	0.410±0.02	0.406±0.02	0.410±0.02	0.413±0.02	0.458±0.04	0.419±0.02	0.417±0.02	**0.469±0.04**
Iris	0.553±0.08	0.622±0.10	0.655±0.11	0.604±0.08	0.647±0.1	0.567±0.08	0.659±0.11	**0.704±0.08**
Wine	0.527±0.05	0.563±0.07	0.562±0.07	0.540±0.05	0.489±0.05	**0.565±0.07**	0.531±0.04	0.538±0.07
Vehicle	0.392±0.00	**0.403±0.00**	0.392±0.00	0.393±0.00	0.388±0.02	0.393±0.00	0.393±0.01	0.386±0.02
Segment	0.262±0.01	0.301±0.03	0.274±0.02	0.298±0.06	0.450±0.06	0.293±0.03	0.306±0.05	**0.504±0.07**
Ionosphere	0.652±0.03	0.686±0.04	0.652±0.03	**0.687±0.04**	0.621±0.12	0.656±0.03	0.685±0.04	0.671±0.06
Pima	0.655±0.03	0.665±0.02	0.657±0.03	0.659±0.03	0.500±0.17	0.639±0.03	**0.665±0.02**	0.617±0.18
Musk	0.558±0.06	**0.612±0.01**	0.561±0.06	0.571±0.05	0.570±0.10	0.570±0.05	0.589±0.04	0.530±0.11
Balance	0.557±0.01	**0.593±0.02**	0.583±0.01	0.569±0.01	0.499±0.06	0.586±0.01	0.564±0.01	0.531±0.07
Yeast	0.367±0.02	0.370±0.02	0.364±0.02	0.370±0.02	0.361±0.03	0.363±0.01	0.371±0.02	**0.397±0.02**

Table 4.7: The rank of average wPFM@20/5 of proper kernel selection on K-EDML and their comparison clustering methods on training/test data

	KMN	ITML	LMNN	GMML	K-KMN	GB-LMNN	K-EDML
Glass	6/5	5/7	7/6	4/4	2/2	3/3	1/1
Iris	7/7	5/4	3/2	4/5	2/3	6/6	1/1
Wine	7/6	2/2	3/3	6/4	5/7	1/1	4/5
Vehicle	7/6	1/1	6/5	5/3	3/7	4/4	2/2
Segment	7/7	4/3	6/6	5/4	2/2	3/5	1/1
Ionosphere	6/6	3/2	5/5	2/1	7/7	4/4	1/3
Pima	4/5	1/2	3/4	6/3	7/7	5/6	2/1
Musk	6/7	3/1	7/6	4/3	1/4	5/5	2/2
Balance	6/6	1/1	4/3	5/5	7/7	2/2	3/4
Yeast	5/4	4/2	7/5	3/3	6/7	2/6	1/1
Rank	6.1/5.9	2.9/2.5	5.1/4.5	4.4/3.5	4.2/5.3	3.5/4.2	1.8/2.1

Table 4.8: The average standard PFM with standard deviation of EDML, K-EDML and their comparison clustering methods on training data

	KMN	ITML	LMNN	GMML	K-KMN	GB-LMNN	EDML	K-EDML
Glass	0.456±0.06	0.456±0.05	0.483±0.05	0.475±0.05	0.452±0.03	**0.624±0.11**	0.469±0.05	0.477±0.04
Iris	0.733±0.03	0.901±0.10	**0.917±0.10**	0.843±0.11	0.771±0.04	0.825±0.10	0.915±0.04	0.895±0.08
Wine	0.884±0.09	0.952±0.08	**0.958±0.08**	0.924±0.08	0.930±0.07	0.944±0.11	0.915±0.08	0.934±0.05
Vehicle	0.331±0.04	**0.416±0.03**	0.343±0.03	0.392±0.04	0.399±0.00	0.356±0.03	0.351±0.04	0.399±0.00
Segment	0.359±0.09	0.548±0.04	0.417±0.10	0.439±0.10	0.563±0.03	0.565±0.08	0.580±0.04	**0.602±0.04**
Ionosphere	0.601±0.01	**0.807±0.03**	0.601±0.01	0.715±0.05	0.669±0.01	0.622±0.02	0.700±0.06	0.700±0.01
Pima	0.600±0.03	0.654±0.01	0.588±0.03	0.602±0.03	**0.702±0.01**	0.580±0.05	0.635±0.02	0.652±0.06
Musk	0.545±0.04	0.607±0.03	0.520±0.03	0.542±0.04	**0.667±0.00**	0.517±0.03	0.558±0.04	0.664±0.01
Balance	0.463±0.03	0.613±0.04	0.624±0.03	0.628±0.03	0.572±0.00	**0.688±0.06**	0.570±0.05	0.554±0.03
Yeast	0.372±0.03	0.264±0.02	0.374±0.03	0.404±0.03	0.390±0.01	0.337±0.03	0.348±0.04	**0.414±0.02**

Table 4.9: The average standard PFM with standard deviation of EDML, K-EDML and their comparison clustering methods which on test data

	KMN	ITML	LMNN	GMML	K-KMN	GB-LMNN	EDML	K-EDML
Glass	0.436±0.06	0.450±0.05	0.424±0.06	0.448±0.05	0.431±0.05	**0.464±0.06**	0.449±0.06	0.446±0.05
Iris	0.741±0.07	0.863±0.10	0.895±0.10	0.801±0.12	0.738±0.09	0.771±0.08	**0.904±0.08**	0.842±0.11
Wine	0.771±0.13	0.896±0.10	**0.906±0.09**	0.851±0.12	0.722±0.12	0.891±0.10	0.820±0.13	0.858±0.11
Vehicle	0.321±0.04	**0.420±0.04**	0.338±0.03	0.378±0.04	0.394±0.00	0.349±0.03	0.335±0.04	0.394±0.00
Segment	0.372±0.09	0.546±0.05	0.434±0.11	0.465±0.11	0.563±0.04	0.514±0.11	0.564±0.05	**0.629±0.04**
Ionosphere	0.594±0.05	**0.794±0.07**	0.593±0.05	0.699±0.07	0.679±0.03	0.613±0.04	0.682±0.08	0.701±0.03
Pima	0.609±0.05	0.648±0.02	0.603±0.04	0.607±0.04	**0.703±0.03**	0.592±0.05	0.637±0.04	0.656±0.06
Musk	0.537±0.04	0.583±0.06	0.524±0.04	0.543±0.04	**0.661±0.01**	0.516±0.04	0.558±0.04	0.657±0.02
Balance	0.467±0.05	0.623±0.06	0.631±0.06	0.620±0.06	0.585±0.01	**0.668±0.06**	0.552±0.05	0.554±0.03
Yeast	0.382±0.03	0.278±0.03	0.369±0.03	0.399±0.03	0.383±0.02	0.316±0.03	0.352±0.04	**0.403±0.03**

Table 4.10: The rank of average standard PFM of proper kernel selection on K-EDML and their comparison clustering methods with the number of clusters is equal to the number of classes on training/test data

	KMN	ITML	LMNN	GMML	K-KMN	GB-LMNN	K-EDML
Glass	5/5	6/2	2/7	4/4	7/6	1/1	3/3
Iris	7/6	3/3	1/2	4/4	6/7	5/5	2/1
Wine	7/6	2/2	1/1	6/5	5/7	3/3	4/4
Vehicle	7/7	1/1	6/6	4/4	3/2	5/5	2/3
Segment	7/7	4/3	6/6	5/5	3/2	2/4	1/1
Ionosphere	7/6	1/1	6/7	2/3	4/4	5/5	3/2
Pima	5/4	2/3	6/6	4/5	1/1	7/7	3/2
Musk	4/5	3/3	6/6	5/4	1/1	7/7	2/2
Balance	7/7	4/3	3/2	2/4	5/5	1/1	6/6
Yeast	5/3	7/7	4/5	2/2	3/4	6/6	1/1
Rank	6.1/5.6	3.3/2.8	4.1/4.8	3.8/4.0	3.8/3.9	4.2/4.4	2.7/2.5

Chapter 5

Human-Centered Cluster Analysis via EDML

5.1 Overview

The proposed K-EDML directly improves the cluster validity index as an objective function and the cluster validity index is smoothed by neighbor relations which refines neighboring cluster relation and the data points of the same class tend to locate in neighboring clusters. Accordingly, a rich representation capability and empirically high performance are obtained in the proposed method. This research takes advantage of these properties to address state-of-the-art cluster analysis limitation, i.e., it cannot preserve class information as a neighborhood relation by considering both features and class labels, provide from a human point of view to guide clustering to be constructed correspondingly with human intuition based on evolutionary distance metric learning. Thus, this study proposed a cluster analysis technique that simultaneously visualizes the cluster structure by considering the class labels, which are available from human intervention, simultaneously with the features to guide clustering to be constructed in the human point of view. By taking advantage of EDML, it allows us to propose this novel cluster analysis easily because of its capability that not only preserve the neighbor cluster relations but also directly improve cluster accuracy in term of cluster validity index score.

By applying K-EDML as a human-centered computing for cluster analysis, a novel cluster analysis technique which improves interpretation of cluster analysis by utilizing class information is proposed. This technique is demonstrated by analyzing the cluster structure result of real-world dataset: facial images and food recipes, to present the usefulness of proposed method which not only preserves the similar features, but also utilizes the class label from the data while learning a distance metric learning simultaneously, unlike any other semi-supervised clustering algorithm. Then the cluster structure is visualized using a K-means clustering with K-nearest neighbor centroids graph (KMN-KNN) [38, 37]. Thus, it helps us to investigate comprehensiveness of cluster analysis, the relationship between class and features, neighborhood relation of cluster boundary and specify the direction of cluster analysis by constructing a particular structure according to the class category (e.g., age or gender), unlike any other semi-supervised clustering algorithm which is impossible to investigate and analyze these properties. For simplicity, the K-EDML with a linear kernel is utilized in this chapter and it is denoted by EDML.

5.2 Experimental Design on Facial Images Data

This experiment performed on facial images from MIT face recognition project[1]. The original facial images are monochrome facial images with 99 features, calculated by coefficients measuring. A principal component analysis (PCA) is applied for dimension reduction. Each face image contains a descriptor of each face, e.g., age, gender, skin color, facial expression, and property on their face. In this experiment, age, namely Child, Teen, Adult and Senior, was used as a class label for DML. Gender was also used as a class label, but only in the last experiment. 200 images were selected in total, 50 images from each age category. Fig. 5.1 shows some examples of facial images. Each row from top to bottom presents an example from each class from Child to Senior. First 3 columns are male and the rest are female.

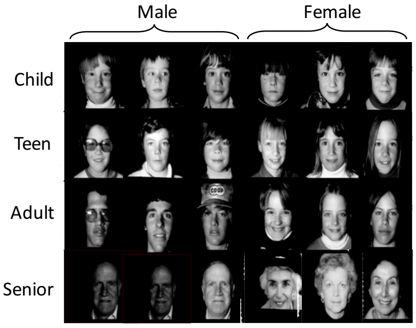


Figure 5.1: Example of monochrome facial images from each category

Note that the number of dimension D depends on the number of variable in the distance transform matrix $\mathbf{M}$ in Eq. (3.2.1). In this dataset, the diagonal representation with 55 dimensions was used. Consequently, PCA was applied to reduce the features to 55. Table 5.1 presents the dataset information.

In addition, population size was set to 5 times the number of dimensions. This experiment

[1] http://courses.media.mit.edu/2004fall/mas622j/04.projects/faces/

Table 5.1: Facial image dataset information

dataset	DML representation	#samples	#attributes	#classes (Age)	#classes (Gender)
faceR-D55C4	diagonal	200	55	4	2

was conducted for five trials with 10,000 iterations (generations) per trial. K-means clustering with K-nearest neighbor centroids graph was utilized with EDML and without EDML for further analysis on the neighborhood relation of clusters. 20 clusters and 5 mutual neighborhoods were employed to obtain cluster structure. For better visualization, all labeled data are provided to EDML in this experiment. The cluster structure was evaluated using weighted F-measure ($wFME$).

The KMN-KNN structures were individually visualized by Cytoscape[2] using edge-weighted spring embedded layout. Cluster analysis was done and summarized interesting things in the following subsections.

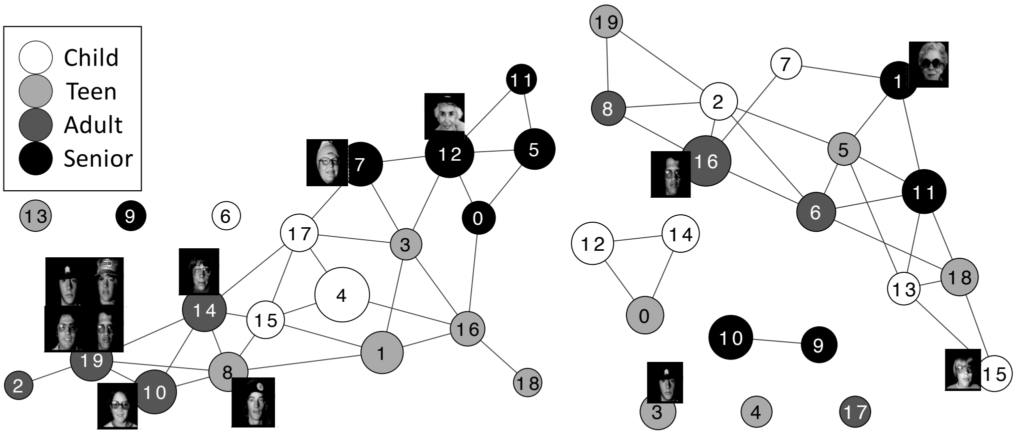
5.3 Experimental Results on Facial Images Data

Before the visualization, the numerical result is examined first. EDML archived about 61% accuracy in terms of $wFME$, obviously yielding an improvement of 10% over the one without EDML. The clustering result by EDML shows that the images in the same age category are either in the same or neighboring clusters more than without EDML, and different age category are in the distant clusters.

5.3.1 Comprehensiveness of Cluster Structure

A visualization of KMN-KNN on the dataset results is visulized in Fig. 5.2 in order to investigate the micro-cluster. Let C_i denotes the i^{th} cluster. C_i is represented by a node, where the edge between two clusters indicates a mutual neighborhood relation. The class label of each cluster is determined by the majority class of the samples in the cluster. The class label of each cluster is indicated by the brightness of the node, the brighter the younger, and vice versa. As well as the node's size indicates the number of majority class's sample. In Fig. 5.2a which is the visualization of EDML, overall distribution can be viewed as a clique, most of the cluster with the same class are intensely connected and appears to be neighbors because of the benefit of preserving age category while cluster is constructed. For example, in Fig. 5.2a Child (C_4, C_{15} and C_{17}), Teen (C_1, C_3, C_8, C_{16} and C_{18}), Adult (C_2, C_{10}, C_{14} and C_{19}) and Senior (C_0, C_5, C_7, C_{11} and

[2] http://www.cytoscape.org/



(a) with EDML (b) without EDML

Figure 5.2: Example of facial images in cluster

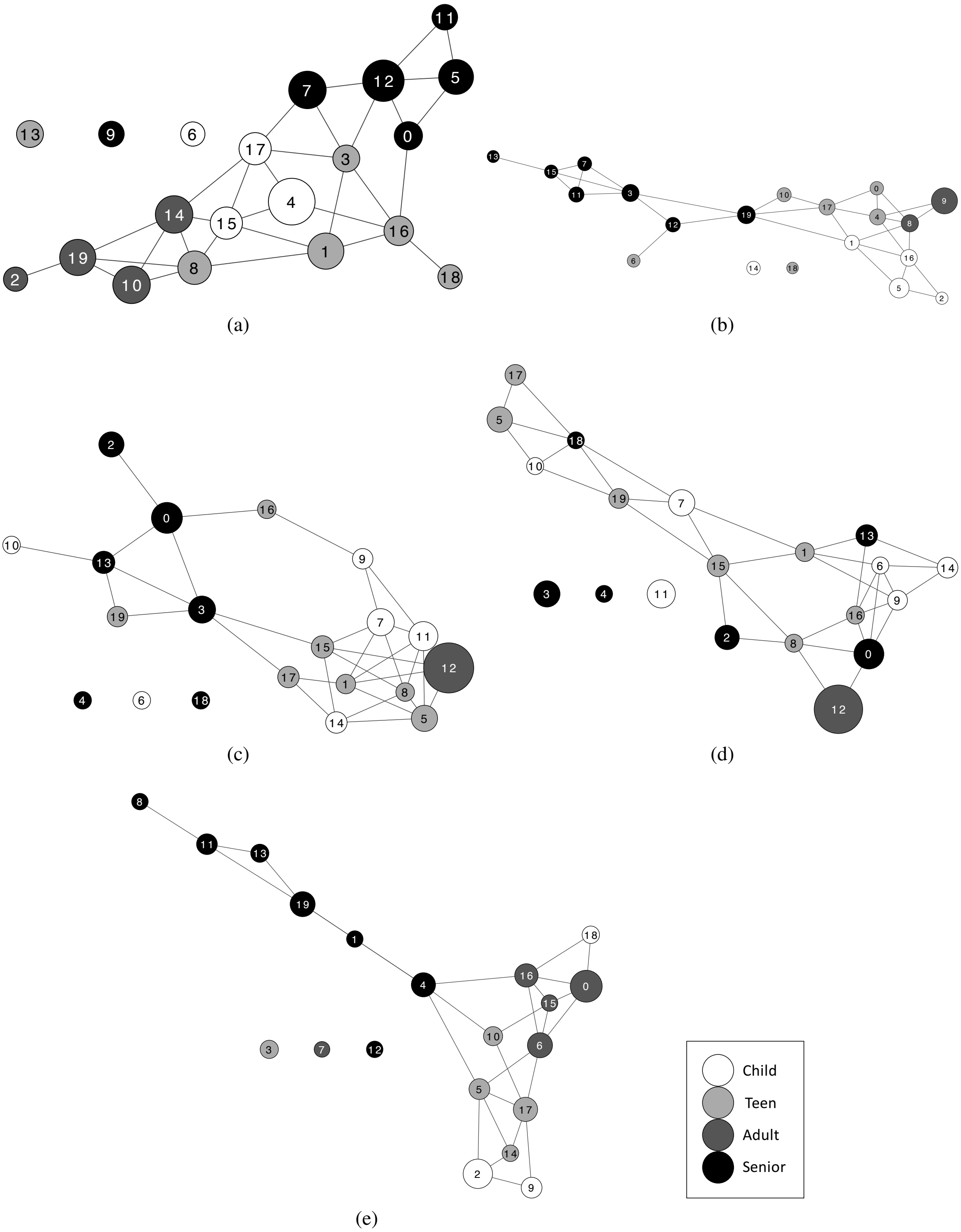
C_{12}) class individually group together and has strong relation between the cluster with the same class than one with the different class. Also only a few individual nodes, no neighbor relation, appear here. Contrary, the without EDML results in Fig. 5.2b, the structures are complicated, the cluster with the same class divided into many parts, for instance, Child class is separated. Unlike the EDML results, many individual nodes appear in Fig. 5.2b, in which EDML is not applied.

Moreover, the number of majority class's sample in each micro-cluster, which is indicated by node's size is also investigated. In without EDML case, each cluster size is similar to each other, while they are varieties of cluster size in EDML. Numerically, the standard deviation of number of majority class sample in each figure is calculated to represent the distribution of the cluster size. The average standard deviation of cluster size in EDML is 4.849 while without EDML is 3.128 which means that EDML can provide a distinct structure of the cluster distribution.

Then the overall cluster distribution is further investigated. In order to confirm the generality of visualization, 4 more trials (5 trials in total) are done in each dataset both with and without EDML, and visualized in in Fig. 5.4 and Fig. 5.3. Sub-figures (a) to (e) in each figure denote different trials. Similar trend can be obtained in every trial.

Therefore, utilizing EDML to KMN-KNN provides a better view of cluster structure, for example, identical class clusters are intensely connected and appear to be neighbors, reduce the number of individual clusters, and affect distribution of samples. Because the cluster analysis using EDML could preserve class information in each cluster.

5.3.2 Investigating Relation Between Class and Features



(a)

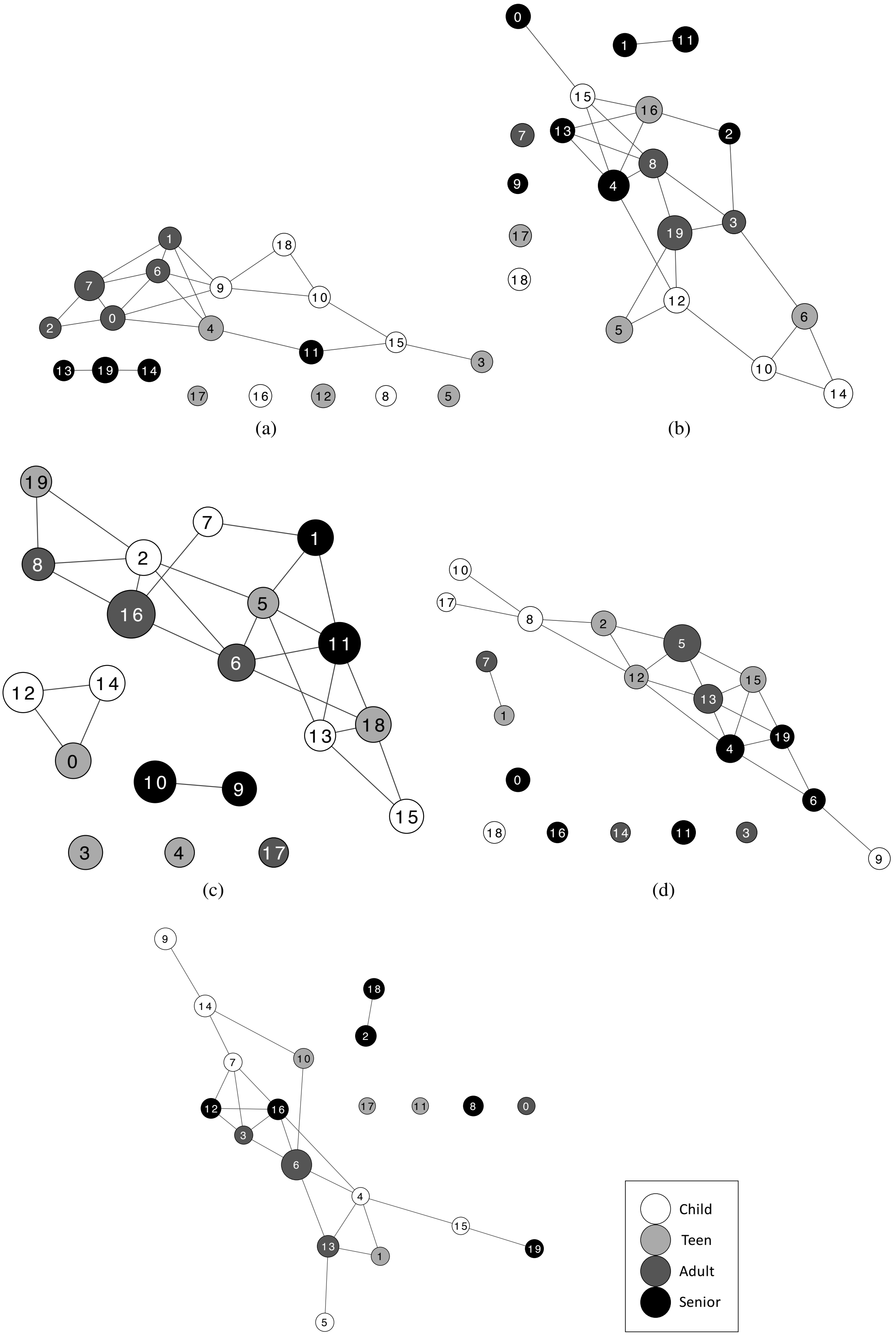
(b)

(c)

(d)

(e)

Figure 5.3: KMN-KNN visualization with EDML



(a)

(b)

(c)

(d)

(e)

Figure 5.4: KMN-KNN visualization without EDML

A unique pattern facial images such as persons wearing glasses and wearing a hat are attached to the node in Fig. 5.2. By comparing Fig. 5.2a, in which EDML was applied, with Fig. 5.2b, in which did without EDML, and analyze the cluster structure. As a result, a pattern that images which contain a unique pattern are obtained such as persons wearing glasses are in neighborhood clusters (C_{10} and C_{19}) and wearing a hat are neighbor (C_8 and C_{14} , and C_7 and C_{12}) in Fig. 5.2a. In contrast to Fig. 5.2b that these pictures are located in the different region of the cluster structure. This is because of the benefit of EDML which can utilize class information (age category) in the clustering. Moreover, it preserves the neighbor relation between the same age category, as a result, it groups the clusters with the same age category close together while it preserves the similar image features, e.g. people wear glasses or hat. Therefore, clusters with the same age category are grouped together, and the similar feature images are gathered together in these groups as well, e.g., in Fig. 5.2a people who wearing glasses are in C_{10}, C_{14} and C_{19} or people who wearing hat in C_7 and C_{12}. These patterns cannot be occurred when without EDML (Fig. 5.2b).

5.3.3 Investigating Neighborhood Relation of Cluster Boundary

In order to investigate another benefit of this work that can preserve the neighborhood relations and provide the cluster boundary. Let assume that the cluster with same label (nodes) which has a neighbor relation merge into a big cluster. Fig.5.3b is selected and defined big clusters indicated by a dash line circle as in Fig. 5.5.

When examined these big clusters, the features in the border node should be similar to the neighboring clusters of the other class. Result in the distribution of the sample in the big cluster that can divide a relatively younger or older in the class which can be seen from the sample of facial images in Fig. 5.5. If it connects to the older class, that cluster tends to be a relatively older face in that class, for example, considering C_{10} (Teen) and C_{19} (Senior), the average age in C_{10} is relatively higher than the other cluster with the class label, and C_{19} are younger in the senior class. Moreover, C_4 and C_{17} has high probabilities to be a younger teen because it is connected to Child class. On the other hand, C_1 clustered the elder child, as it has neighbor relations with Teen, Adult and Senior class. Furthermore, when Senior class was considered, each individual sub cluster could be analyzed, and show that C_{13} is relatively oldest senior because it has only relation with the same class (Senior) and has no relation to another class. Similarly to C_2 in Child class and C_9 in Adult class. Note that image features do not reflect only age, other possibilities are such as similar hair style, face shape, or they are families. Importantly, these boundaries cannot be found without EDML (Fig. 5.2b).

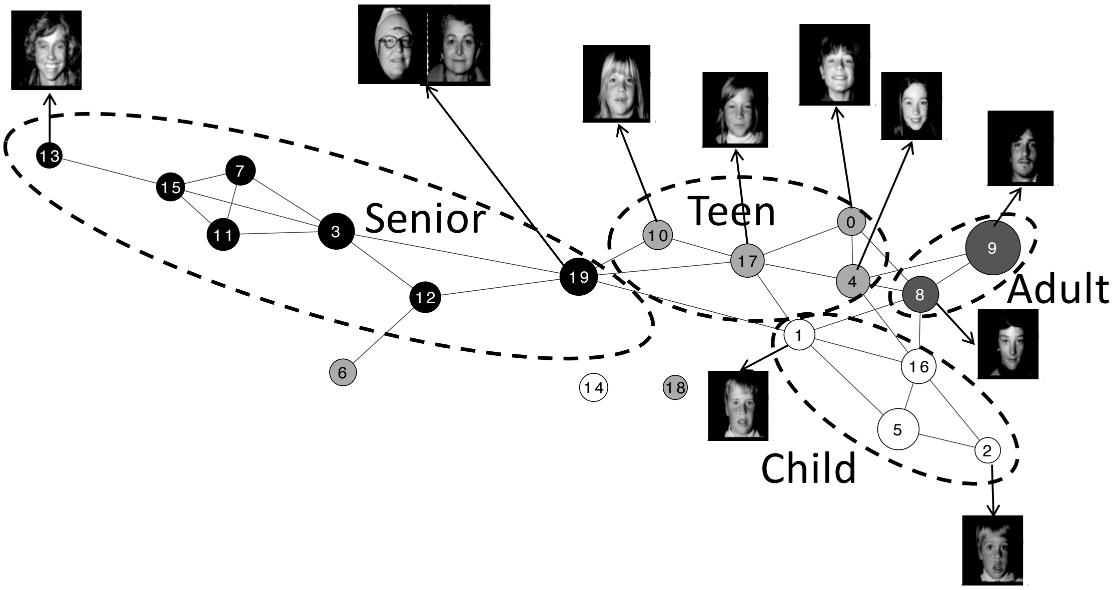


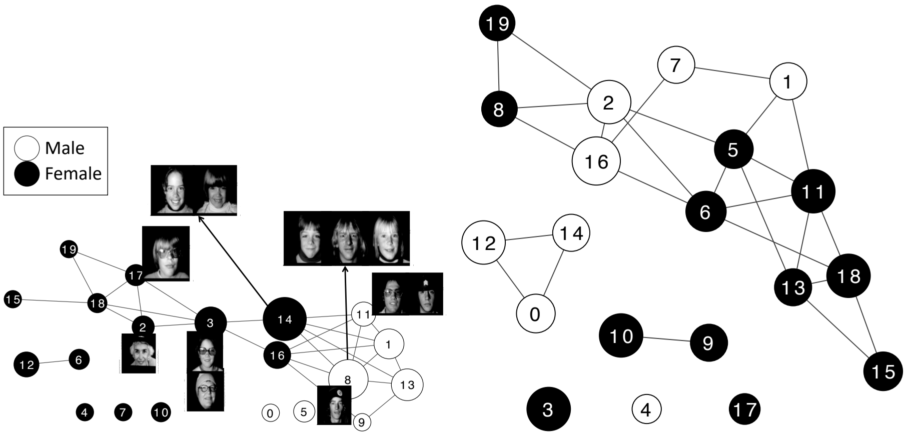
Figure 5.5: Cluster structure boundary

5.3.4 Changing the Viewpoint of Cluster Analysis

When the target of cluster analysis is changed, it is better to construct a new cluster structure according to the target class category. Since EDML has an ability to optimize class information, this method makes use of it to select a different class category (e.g., age or gender) to optimize and change the viewpoint of cluster analysis. For example, using age category aids cluster analysis together with image features, and similarly in gender category.

In this experiment, the KMN-KNN was visualized according to the gender of sample: male and female, denoted by node's color (i.e., white and black respectively). The proposed method can provide a particular optimization for the gender analysis, which can be visualized as shown in Fig. 5.6a. Conversely, without EDML, as seen in Fig. 5.6b changes is noticed only in the labels of the micro-clusters; however, the cluster structure is identical to the one in the age analysis in Fig. 5.2b.

Fig. 5.6a, KMN-KNN preserves the neighbor relations and forms the clique in between the cluster with samples of identical majority class. Also, the border properties of clusters representing males with long hair are shown in close proximity with females who have short hair, shown in the figure by the dash line, due to their similar visual features. Lastly, the unique patterns can be recognized in each class: male wearing glasses C_{11}, male wearing hat (C_8 and C_{11}), women wearing glasses (C_3 and C_{17}) and women wearing hat (C_2 and C_3) which are located close together.



(a) with EDML (optimized by gender category) (b) without EDML

Figure 5.6: KMN-KNN visualization on gender category

5.4 Experimental Design on Recipe Data

This experiment applied EDML to a larger dataset by performing on food recipes data from Cookpad[3], the Japan's largest site for sharing their original recipes. The dataset is provided from collaboration between Cookpad Co., Ltd and National Institute of Informatic (NII) to reseacher[4] [17]. The recipes are all in Japanese, each recipe contains ingredients and how to cook. First, recipes is selected from 10 preselected food categories (Seaweed, Vegetable, Fish, Noodle, Pasta, Egg, Salad, Soup, Sauce & Dressing, and Meat), which come from today's recipe category total 19236 recipes and 8398 ingredients. A Japanese cooking ontology [50] is applied to merge various ingredient names since the users freely write their ingredients. Then, random sampling was used to select about 220 to 400 recipes from each category to resolve the imbalance of the data. Consequently, total 3732 recipes and 129 ingredients were selected to use in this experiment. The features are extracted from ingredients of each recipe using term frequency-inverse document frequency (TF-IDF) and PCA is appllied to reduce dimension to 10. The food categories is corresponded to data class label. As previous experiment *wFME* was used as an objective function. Unlike in facial images data set, cluster size is set to 50, and full matrix representation is applied instead, due to the limitation of computational time.

[3] http://cookpad.com/
[4] https://www.nii.ac.jp/dsc/idr/cookpad/cookpad.html

5.5 Experimental Results on Recipe Data

The goodness of the cluster structure is preliminary evaluated by *wFME*. Although, EDML archived 0.261 of *wFME* in this data, it yielded an improvement of 30% over the clustering without EDML. The improvement of *wFME* clearly illustrated the distribution of the same class recipes which are either in the same cluster or neighbor clusters more than the one without EDML. Conversely, the different recipe category seems to be located farther away.

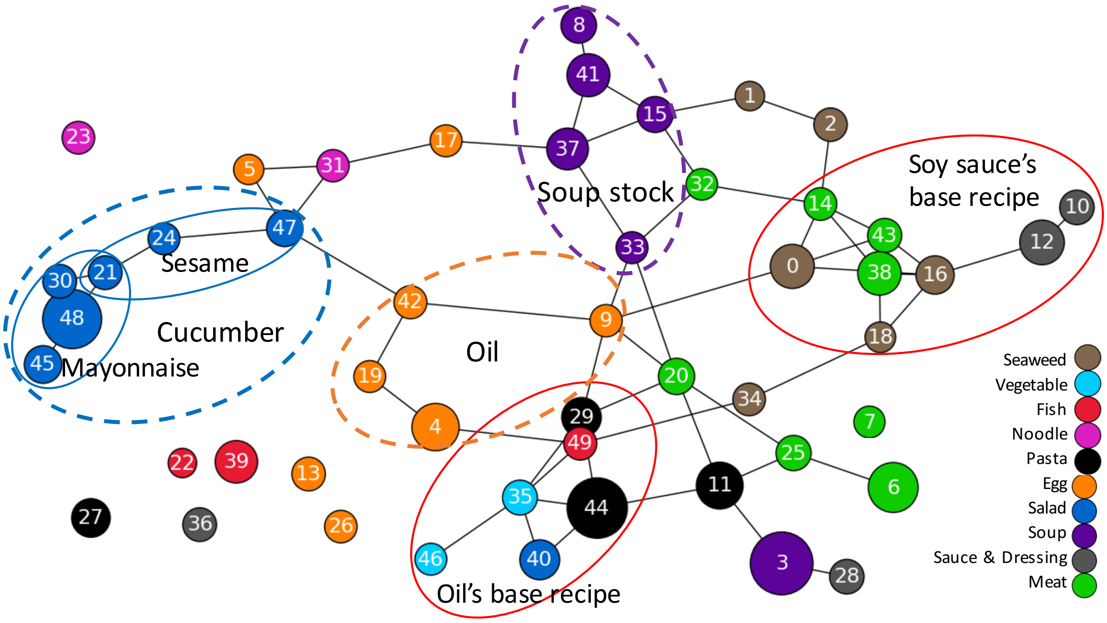


Figure 5.7: KMN-KNN visualization on cookpad recipe's data set

Then, the KMN-KNN visualization is shown in Fig. 5.7. From the visualization, EDML still archived similar results like in facial image dataset. First, comprehensiveness of cluster structure is preserved. It is evident from the visualization that preserving class category, while construct the cluster, benefit most of the clusters with the same class to be intensely connected and appears to be neighbors. These results are indicated by dash line ellipse in Fig. 5.7, e.g., Salad (C_{21}, C_{24}, C_{30}, C_{45}, C_{47} and C_{48}), Soup (C_8, C_{15}, C_{33}, C_{37} and C_{41}), and so on.

In order to investigate relationship between classes and features, the most occurrence ingredients in each cluster are investigated as shown in Table 5.2 (unnecessary clusters are omitted). These results indicated that not only the same category recipes tend to locate close together, but recipes that contain similar ingredients (feature) are also located nearby either in the same or neighbor clusters. These clusters are denoted inside the red ellipse in Fig. 5.7. These could help the user to select the recipe according to the remaining ingredients, or what kind of taste they want, for example, soy sauce's base recipe: "Basil Chicken" (C_{38}), oil's base recipe: "Napolitan spaghetti" (C_{44}), and so on.

Lastly, the big cluster is indicated as in a dash line ellipse in order to explore the cluster boundary. The features in the border node is discovered again as in the previous experiments that it should be similar to the neighboring big clusters, as can be seen from the relationship between C_9 (Egg) and C_{33} (Soup) that the intersection of these two clusters is "Green onion". While in their own big clusters are "Oil" for Egg (C_4, C_9, C_{19} and C_{42}) and "Soup stock" for Soup ($C_8, C_{15}, C_{33}, C_{37}$ and C_{41}) big clusters respectively. Moreover, we surprisingly discovered that the border is possible to occur in an individual big cluster. The Salad big cluster (C_{21}, C_{24}, C_{30}, C_{45}, C_{47} and C_{48}), indicated inside the blue dash line ellipse, was carefully investigated and found that C_{21} is the inner border of their own big cluster. By dividing the big cluster into two groups using C_{21}, indicated by blue ellipse, the similar features in C_{45}, C_{48}, C_{30} change to mayonnaise, and another group is sesame instead of cucumber. Thus, border between big clusters can be occasionally extended to the neighbor cluster of the border of the big cluster.

Table 5.2: Top five most occurance ingredients in each cluster with number of occurances (Translated from Japanese)

Cluster	Rank1	Rank2	Rank3	Rank4	Rank5
0	Soy sauce(97)	Soup stock(74)	Sugar(56)	Mirin(55)	Oil(53)
4	Egg(105)	Salt(74)	Oil(67)	Sugar(28)	Cheese(27)
8	Miso(42)	Soup stock(23)	Eggplant(7)	Fried tofu(5)	Cabbage(4)
9	Green onion(52)	Oil(39)	Soy sauce(33)	Vinegar(24)	Soup stock(20)
10	Soy sauce(66)	Sugar(58)	Vinegar(51)	Mirin(47)	Ginger(11)
12	Soy sauce(87)	Vinegar(80)	Sugar(77)	Oil(69)	Sesame(33)
14	Soy sauce(37)	Sugar(36)	Sake(33)	Mirin(27)	Miso(25)
15	Miso(57)	Soup stock(28)	Oil(27)	Green onion(14)	Egg(13)
16	Soy sauce(97)	Oil(79)	Sugar(62)	Mirin(41)	Vinegar(38)
18	Soy sauce(52)	Sliced dried bonito(15)	Kombu(14)	Mirin(14)	Radish(10)
19	Egg(58)	Oil(26)	Noodle Soup base(19)	Wheat flour(19)	Cheese(12)
21	Mayonnaise(37)	Sesame(35)	Cucumber(18)	Sugar(11)	Soy sauce(10)
24	Sesame(50)	Cucumber(20)	Oil(20)	Ponzu(18)	Sugar(16)
29	Oil(127)	Salt(108)	Soy sauce(50)	Garlic(42)	Pepper(39)
30	Mayonnaise(65)	Tuna(22)	Radish(20)	Green onion(18)	Soy sauce(16)
33	Water(61)	Soup stock(53)	Salt(40)	Green onion(34)	Soy sauce(33)
35	Oil(95)	Salt(80)	Garlic(28)	Pepper(27)	Parsley(13)
37	Soup stock(52)	Green onion(51)	Miso(46)	Carrot(41)	Radish(41)
38	Soy sauce(96)	Sake(85)	Mirin(48)	Ginger(47)	Chicken(41)
40	Salt(88)	Oil(75)	Vinegar(58)	Sugar(42)	Sesame(35)
41	Miso(54)	Soup stock(45)	Water(26)	Green onion(25)	Fried tofu(16)
42	Oil(83)	Perilla(29)	Soy sauce(24)	Seaweed(23)	Noodle Soup base(23)
43	Soy sauce(87)	Water(82)	Sugar(72)	Mirin(52)	Sake(40)
44	Oil(231)	Salt(193)	Garlic(141)	Pepper(128)	Pasta noodles(106)
45	Mayonnaise(82)	Egg(45)	Salt(28)	Soy sauce(22)	Pepper(19)
47	Sesame(29)	Ponzu(27)	Cucumber(21)	Noodle Soup base(19)	Tomato(16)
48	Mayonnaise(175)	Salt(124)	Pepper(62)	Onion(53)	Potato(47)
49	Soy sauce(73)	Salt(72)	Oil(60)	Pepper(30)	Mayonnaise(25)

Chapter 6

Conclusion and Future work

6.1 Summary

Many well-known semi-supervised clustering is suffered from monotonicity to the number of constraints and constraint satisfactory objective function, which sometimes destroy the cluster quality. Moreover, conventional distance metric learning techniques are based on Mahalanobis distance which yields small benefit to non-linearly separable data. This research addresses these problems by unifying the state-of-the-art evolutionary distance metric learning (EDML) and kernelization technique. The proposed method not only overcomes other methods in term of cluster validity. It can also preserve class information as a neighborhood relation by considering both features and class labels to guide clustering to be constructed correspondingly with human intuition based on evolutionary distance metric learning.

The first contribution to this dissertation is a novel kernelized evolutionary distance metric learning (K-EDML) for semi-supervised clustering, an integration of kernelization, and evolutionary distance metric learning technique wherein any set-based or pairwise-based clustering validity index can be optimized utilizing a differential evolution with self-adapting control parameters and generalized opposition-based learning (GOjDE) algorithm in Chapter 4. In this experiments, the proposed method is compared to baseline clustering, i.e., k-means clustering and kernel k-means clustering, and well-known distance metric learning techniques, i.e., information-theoretic metric learning, distance metric learning for large margin nearest neighbor classification, Geometric Mean Metric Learning, and Gradient Boosted Large Margin Nearest Neighbors, in ten datasets from the UCI machine learning repository. The cross-validation comparison results illustrate that the proposed K-EDML obtain the highest accuracy in five datasets in both training and test data, which confirms the benefit of the evolutionary distance metric learning (EDML) in K-EDML that directly optimize the cluster validity index as an objective function. Despite highest results from other methods in some dataset, they can only perform well in specific data, while the proposed method performs well in many datasets, which can be seen from the highest average ranking in all dataset. Moreover, K-EDML addressed the drawback of EDML in non-linearly separable input space and demonstrate the benefit of kernel function in real-world dataset not only to the proposed K-EDML method but also other kernelized methods due to its superior results improving from its couple methods. Finally, the proposed method demonstrates its generalization capability by overcoming other competitor methods in standard cluster evaluation, even the evaluation criteria is different from the training scheme.

Another contribution to this dissertation a novel cluster analysis technique that simultaneously analyzes both class label and features sample by utilizing an evolutionary distance metric learning in Chapter 5. This research demonstrated the method in the real-world dataset called facial images from the MIT face recognition project and food recipe dataset from Cookpad. Cluster structures of these images were constructed using the proposed method and without the proposed method then visualized KMN-KNN by Cytoscape. The experimental results have shown the advantage of the proposed cluster analysis method which provide a more understanding cluster structure. The proposed method can help to investigate the relationship between class and features, for example, the same class clusters are grouped together in order to preserve the class and neighbor relations while features are also preserved. Thus, Adult (class) who wear glasses (feature) is located in the same or neighbor clusters. Moreover, overall cluster structure is well-organized, identical majority class groups together and reduce the number of individual clusters, confirm by both visualization and numerical analysis with higher standard derivation from the one without the proposed technique. In addition, an approach could analyze each individual sub cluster, for example, elder teen and older teen, which can investigate from the boundary of the big clusters. Lastly, EDML can specify the direction of cluster analysis by constructing a particular structure according to the class category (e.g., age or gender), unlike other methods which can construct only one structure for the dataset in the same environment.

6.2 Future work

Although, K-EDML has a rich representation capability and empirically high performance; however, it suffers from computational time. Thus, this research aims to improve the computational efficiency of K-EDML in the higher dimensionality problems, e.g., eigenvalue optimization [83], reinforcement learning [2, 3] and aggregated DML [47]. Moreover, in more complex dataset which data are gathered from multiple sources, this research aims to utilize more label information, e.g., multiple-kernel learning [45], utilize hierarchical information [51], and integrate deep learning technique to learn a distance metric [53, 60, 72]. Moreover, applying the proposed methods to other field is also a candidate goal, e.g., classification and image retrieval. Lastly, this research aims to apply the proposed method to many more real-world dataset.

Bibliography

[1] A. Aktas, D. Walsh, and L. Rybicki. Review: Symptom clusters: myth or reality? *Palliative Medicine*, 24(4):373–385, 2010.

[2] B. Ali, K. Fukui, W. Kalintha, K. Moriyama, and M. Numao. Reinforcement learning based distance metric filtering approach in clustering. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1328–1335, Nov 2017.

[3] B. Ali, W. Kalintha, K. Moriyama, M. Numao, and K. Fukui. Reinforcement learning based distance metric filtering approach in clustering. In *2018 The Genetic and Evolutionary Computation Conference (GECCO)*, pages 155–156, July 2018.

[4] M. B. Andrea. The elusive concept of the symptom cluster. *Oncology nursing forum*, 34(5):971–980, 2007.

[5] E. Bair. Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(5):349–361, 2013.

[6] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proc. the 20th International Conference on Machine Learning (ICML-03)*, pages 11–18, 2003.

[7] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407–416, 2000.

[8] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Processing Systems (NIPS)*, pages 585–591, 2002.

[9] M. J. Berry and G. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Support*. John Wiley & Sons, Inc., New York, NY, USA, 1997.

[10] W. Bian and D. Tao. Learning a distance metric by empirical loss minimization. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 1186–1191, 2011.

[11] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of the International Conference on Machine Learning (ICML-04)*, pages 81–88, 2004.

[12] J. Brest, S. Greinero, B. Boskovic, M. Mernik, and V. Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657, 2006.

[13] S. Brin and L. Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18):3825 – 3833, 2012.

[14] R. Chatpatanasiri, T. Korsrilabutr, P. Tangchanachaianan, and B. Kijsirikul. A new kernelization framework for mahalanobis distance learning algorithms. *Neurocomputing*, 73(10-12):1570–1579, June 2010.

[15] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *J. Mach. Learn. Res.*, 11:1109–1135, Mar. 2010.

[16] J. Chuang and D. J. Hsu. Human-centered interactive clustering for data analysis. In *Workshop on Human-Propelled Machine Learning held in Conference on Neural Information Processing Systems (NIPS)*, 2014.

[17] L. Cookpad Co. Cookpad dataset, 2015.

[18] S. Dasgupta and V. Ng. Which clustering do you want? inducing your ideal clustering with minimal feedback. *CoRR*, abs/1401.5389, 2014.

[19] I. Davidson, K. L. Wagstaff, and S. Basu. Measuring constraint-set utility for partitional clustering algorithms. In *Proc. The 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-06)*, pages 115–126, 2006.

[20] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1(4):224–227, 1979.

[21] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 209–216. ACM, 2007.

[22] L. J. Deborah, R. Baskaran, and A. Kannan. A survey on internal validity measure for cluster validation. *International Journal of Computer Science & Engineering Survey (IJCSES)*, 1(2):85–102, 2010.

[23] R. Dey, S. Chakraborty, and L. Dey. Weather forecasting using convex hull & k-means techniques an approach. In *1st International Science and Technology Congress(IEMCON)*, volume 1, pages 288–295. Elsevier, 2014.

[24] D. Dheeru and E. Karra Taniskidou. UCI machine learning repository, 2017.

[25] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: Spectral clustering and normalized cuts. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 551–556. ACM, 2004.

[26] J. C. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104, 1974.

[27] G. Fan. Symptom clusters in cancer patients: A review of the literature. *Current Oncology*, 14(5), 2007.

[28] K. Fukui and M. Numao. Neighborhood-based smoothing of external cluster validity measures. In *Proc. the 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, PAKDD '12, pages 354–365. Springer, 2012.

[29] K. Fukui, S. Ono, T. Megano, and M. Numao. Evolutionary distance metric learning approach to semi-supervised clustering with neighbor relations. In *Proc. of 2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, ICTAI '13, pages 398–403, 2013.

[30] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, pages 513–520, 2004.

[31] P. H. Guzzi, E. Masciari, G. M. Mazzeo, and C. Zaniolo. *Information Technology in Bio- and Medical Informatics*, chapter A Discussion on the Biological Relevance of Clustering Results, pages 30–44. Springer International Publishing, 2014.

[32] M. Halkidi and M. Vazirgiannis. Clustering validity assessment using multi representatives. In *Proc. 2nd Hellenic Conference on Artificial Intelligence*, pages 237–248, 2002.

[33] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *Proc. the 21st International Conference on Machine Learning (ICML-04)*, pages 393–400, 2004.

[34] A. Jaimes and N. Dimitrova. Human-centered multimedia: culture, deployment, and access. *MultiMedia, IEEE*, 13(1):12–19, 2006.

[35] A. Jaimes, D. Gatica-Perez, N. Sebe, and T. Huang. Human-centered computing–toward a human revolution. *Computer*, 40(5):30–34, 2007.

[36] W. Kalintha, K. Fukui, S. Ono, T. Megano, K. Moriyama, and M. Numao. Semi-supervised evolutionary distance metric learning for clustering. In *The 29th Annual Conference of the Japanese Society for Artificial Intelligence*, JSAI '15, 2015.

[37] W. Kalintha, K. Fukui, S. Ono, T. Megano, K. Moriyama, and M. Numao. Integrating class information and features in cluster analysis based on evolutionary distance metric learning. In *Intelligent and Evolutionary Systems*, pages 165–181. Springer International Publishing, 2017.

[38] W. Kalintha, T. Megano, S. Ono, K. Fukui, and M. Numao. Cluster analysis of face images and literature data by evolutionary distance metric learning. In *Proc. of the 35th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, AI '15, pages 301–315. Springer, 2015.

[39] S. Kaski and J. Sinkkonen. Principle of learning metrics for exploratory data analysis. *The Journal of VLSI Signal Processing Systems for Signal Image and Video Technology*, 37:177–188, 2004.

[40] D. Kedem, S. Tyree, F. Sha, G. R. Lanckriet, and K. Q. Weinberger. Non-linear metric learning. In *Advances in Neural Information Processing Systems 25*, pages 2573–2581. 2012.

[41] R. Kling and L. Star. Human centered systems in the perspective of organizational and social informatics. In *Computer and Society*, pages 22–29, 1998.

[42] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, 1995.

[43] D. Koloseni, J. Lampinen, and P. Luukka. Optimized distance metrics for differential evolution based nearest prototype classifier. *Expert Systems with Applications*, 39(12):10564–10570, 2012.

[44] B. Kulis, S. Basu, I. Dhillon, and R. Mooney. Semi-supervised graph clustering: a kernel approach. *Machine Learning*, 74(1):1–22, Jan 2009.

[45] C. Li, Q. Liu, J. Liu, and H. Lu. Ordinal distance metric learning for image ranking. *IEEE Transactions on Neural Networks and Learning Systems*, 26(7):1551–1559, July 2015.

[46] C. Li, S. Yang, T. T. Nguyen, E. L. Yu, X. Yao, Y. Jin, H. g. Beyer, and P. N. Suganthan. Benchmark generator for cec'2009 competition on dynamic optimization, 2008.

[47] J. Li, X. Lin, X. Rui, Y. Rui, and D. Tao. A distributed approach toward discriminative distance metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, 26:2111–2122, 2015.

[48] Z. Liu and R. George. *Fuzzy Modeling with Spatial Information for Geographic Problems*, chapter Mining Weather Data Using Fuzzy Cluster Analysis, pages 105–119. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[49] P. Moutafis, M. Leng, and I. A. Kakadiaris. An overview and empirical comparison of distance metric learning methods. *IEEE Transactions on Cybernetics*, 47(3):612–625, 2016.

[50] H. Nanba, Y. Doi, M. Tsujita, T. Takezawa, and K. Sumiya. Construction of a cooking ontology from cooking recipes and patents. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 507–516. ACM, 2014.

[51] B. M. Nogueira, Y. K. B. Tomas, and R. M. Marcacini. Integrating distance metric learning and cluster-level constraints in semi-supervised clustering. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4118–4125, May 2017.

[52] R. Nugent and M. Meila. *Statistical Methods in Molecular Biology*, chapter An Overview of Clustering Applied to Molecular Biology, pages 369–404. 2010.

[53] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[54] V. K. P.-N. Tan, M. Steinbach. *Cluster Analysis: Basic Concepts and Algorithms*. Addison-Wesley, 2006.

[55] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution A Practical Approach to Global Optimization*. Natural Computing Series. Springer-Verlag, Berlin, Germany, 2005.

[56] Q. Qian, J. Hu, R. Jin, J. Pei, and S. Zhu. Distance metric learning using dropout: A structured regularization approach. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 323–332, New York, NY, USA, 2014. ACM.

[57] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama. Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation*, 12(1):64–79, 2008.

[58] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[59] M. J. Sewitch, K. Leffondré, and P. L. Dobkin. Clustering patients according to health perceptions: Relationships to psychosocial characteristics and medication nonadherence. *Journal of Psychosomatic Research*, 56(3):323 – 332, 2004.

[60] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1857–1865. Curran Associates, Inc., 2016.

[61] M. Soleymani Baghshah and S. Bagheri Shouraki. Kernel-based metric learning for semi-supervised clustering. *Neurocomput.*, 73(7-9):1352–1361, Mar. 2010.

[62] C. R. Souza. Kernel functions for machine learning applications., 2010.

[63] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[64] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[65] C. Veenhuis and M. Koppen. *Data Swarm Clustering*, chapter 10, pages 221–241. Springer-Verlag, 2006.

[66] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proc. of the International Conference on Machine Learning (ICML-01)*, pages 577–584, 2001.

[67] K. L. Wagstaff. Value, cost, and sharing: Open issues in constrained clustering. In *Proc. the Fifth International Workshop on Knowledge Discovery in Inductive Databases (KDID 2006)*, pages 1–10, 2007.

[68] F. Wang and J. Sun. Survey on distance metric learning and dimensionality reduction in data mining. *Data Min. Knowl. Discov.*, 29(2):534–564, Mar. 2015.

[69] H. Wang, S. Rahnamayan, and Z. Wu. Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems. *Journal of Parallel and Distributed Computing*, 73(1):62–73, 2013.

[70] H. Wang, Z. Wu, and S. Rahnamayan. Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. *Soft Computing*, 15(11):2127–2140, 2011.

[71] Q. Wang, J. Wan, and Y. Yuan. Locality constraint distance metric learning for traffic congestion detection. *Pattern Recognition*, 75:272 – 281, 2018. Distance Metric Learning for Pattern Recognition.

[72] W. Wang, C. Chen, W. Chen, P. Rai, and L. Carin. Deep metric learning with data summarization. In *European Conference on Machine Learning and Knowledge Discovery in Databases - Volume 9851*, ECML PKDD 2016, pages 777–794, Berlin, Heidelberg, 2016. Springer-Verlag.

[73] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)*, 10:207–244, 2009.

[74] R. Weiss, B. Vélez, and M. A. Sheldon. Hypursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *Proceedings of the the Seventh ACM Conference on Hypertext*, pages 180–193, 1996.

[75] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Clustering user queries of a search engine. In *Proceedings of the 10th International Conference on World Wide Web*, pages 162–168, 2001.

[76] B. Xiao, X. Yang, Y. Xu, and H. Zha. Learning distance metric for regression by semidefinite programming with application to human age estimation. In *Proceedings of the 17th ACM International Conference on Multimedia*, MM '09, pages 451–460, New York, NY, USA, 2009. ACM.

[77] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems (NIPS)*, pages 505–512, 2002.

[78] R. Xu and D. Wunsch. *CLUSTER VALIDITY*, chapter 10, pages 263–278. Computational Intelligence. IEEE Press, 2008.

[79] L. Yang. Distance metric learning : A comprehensive survey. Technical Report 16, Michigan State Universiy, 2006.

[80] L. Yang, R. Jin, R. Sukthankar, and Y. Liu. An efficient algorithm for local distance metric learning. In *Proc. the National Conference on American Association for Artificial Intelligence (AAAI-06)*, pages 543–548, 2006.

[81] D. Y. Yeung and H. Chang. A kernel approach for semisupervised metric learning. *IEEE Transactions on Neural Networks*, 18:141–149, 2007.

[82] S. Ying, Z. Wen, J. Shi, Y. Peng, J. Peng, and H. Qiao. Manifold preserving: An intrinsic approach for semisupervised distance metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):2731–2742, July 2018.

[83] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. *J. Mach. Learn. Res.*, 13:1–26, Jan. 2012.

[84] P. H. Zadeh, R. Hosseini, and S. Sra. Geometric mean metric learning. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2464–2471, New York, New York, USA, 20–22 Jun 2016. PMLR.

[85] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua. Robust distance metric learning with auxiliary knowledge. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 1327–1332, 2009.

Appendix A

List of Publication

A.1 Journal

- Wasin Kalintha, Satoshi Ono, Masayuki Numao, and Ken-ichi Fukui. "Kernelized Evolutionary Distance Metric Learning for Semi-Supervised Clustering", Intelligent Data Analysis, 23 (6). 2019. (in Press)

A.2 International Conference/Symposium (with review)

- Wasin Kalintha, Satoshi Ono, Masayuki Numao, and Ken-ichi Fukui. "Kernelized Evolutionary Distance Metric Learning for Semi-supervised Clustering", Proc. 31st AAAI Conference on Artificial Intelligence, Student Abstract and Poster Program, pp. 4945-4946, San Francisco, USA, Feb. 2017. (Poster)
- Wasin Kalintha, Satoshi Ono, Masayuki Numao, and Ken-ichi Fukui. "Integrating Class Information and Features in Cluster Analysis based on Evolutionary Distance Metric Learning", Proc. the 20th Asia-Pacific Symposium on Intelligent and Evolutionary Systems (IES-2016), pp. 165-181, Canberra, Australia, Nov. 2016.
- Wasin Kalintha, Taishi Megano, Satoshi Ono, Ken-ichi Fukui, and Masayuki Numao. "Cluster Analysis of Face Images and Literature Data by Evolutionary Distance Metric Learning", Proc. Thirty-fifth SGAI International Conference on Artificial Intelligence (AI-2015), pp. 301-315, Cambridge, UK, Dec. 2015.
- Bassel Ali, Wasin Kalintha, Koichi Moriyama, Masayuki Numao, and Ken-ichi Fukui. "Reinforcement Learning for Evolutionary Distance Metric Learning Systems Improvement", Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2018),

pp.155-156, Kyoto, Jul. 2018. (Poster)

- Bassel Ali, Ken-ichi Fukui, Wasin Kalintha, Koichi Moriyama, and Masayuki Numao. “Reinforcement Learning based Distance Metric Filtering Approach in Clustering”, Proc. 2017 IEEE Symposium Series on Computational Intelligence (SSCI 2017), pp. 1328-1335, Honolulu, USA, Nov. 2017. (Poster)

A.3 International Symposium (without review)

- Bassel Ali, Ken-ichi Fukui, Koichi Moriya, Wasin Kalintha, and Masayuki Numao. “Machine Learning for Distance Metric Learning Systems Improvement”, Proc. 21th SANKEN International The 16 SANKEN Nanotechnology Symposium, Osaka, Jan. 2018. (Poster)

A.4 Domestic Workshop (without review)

- Wasin Kalintha, Satoshi Ono, Masayuki Numao, and Ken-ichi Fukui. “Kernelized Evolutionary Distance Metric Learning for Semi-supervised Clustering”, IT連携フォーラムOACIS 第33回シンポジウム, Osaka, Dec. 2017. (Poster)

- Wasin Kalintha, Satoshi Ono, Masayuki Numao, and Ken-ichi Fukui. “Proposition of Kernelized Evolutionary Distance Metric Learning for Semi-supervised Clustering”, 人工知能学会第109回知識ベースシステム研究会, pp. 56-61, Tokyo, Nov. 2016. (人工知能学会研究会優秀賞)

- Wasin Kalintha, Ken-ichi Fukui, Taishi Megano, Satoshi Ono, and Masayuki Numao. “Semi-supervised evolutionary distance metric learning for clustering”, Proc. the 29th Annual Conference of the Japanese Society for Artificial Intelligence (JSAI -2015), Hakodate, May 2015.