Improving Cluster Analysis by Co-initializations

He Zhang*, Zhirong Yang, Erkki Oja

Department of Information and Computer Science, Aalto University, Espoo, Finland

Abstract

Many modern clustering methods employ a non-convex objective function and use iterative optimization algorithms to find local minima. Thus initialization of the algorithms is very important. Conventionally the starting guess of the iterations is randomly chosen; however, such a simple initialization often leads to poor clusterings. Here we propose a new method to improve cluster analysis by combining a set of clustering methods. Different from other aggregation approaches, which seek for consensus partitions, the participating methods in our method are used consequently, providing initializations for each other. We present a hierarchy, from simple to comprehensive, for different levels of such co-initializations. Extensive experimental results on real-world datasets show that a higher level of initialization often leads to better clusterings. Especially, the proposed strategy is more effective for complex clustering objectives such as our recent cluster analysis method by low-rank doubly stochastic matrix decomposition (called DCD). Empirical comparison with three ensemble clustering methods that seek consensus clusters confirms the superiority of improved DCD using co-initialization.

Keywords: Clustering, Initializations, Cluster ensembles

*Corresponding author.

Email addresses: `he.zhang@aalto.fi` (He Zhang), `zhirong.yang@aalto.fi` (Zhirong Yang), `erkki.oja@aalto.fi` (Erkki Oja)

1. Introduction

Cluster analysis plays an essential role in machine learning and data mining. The aim of clustering is to group a set of objects in such a way that the objects in the same cluster are more similar to each other than to the objects in other clusters, according to a particular objective. Many clustering methods are based on objective functions which are non-convex. Their optimization generally involves iterative algorithms which start from an initial guess. Proper initialization is critical for getting good clusterings.

For simplicity, random initialization has been widely used, where a starting point is randomly drawn from a uniform or other distribution. However, such a simple initialization often yields poor results and the iterative clustering algorithm has to be run many times with different starting points in order to get better solutions. More clever initialization strategies are thus required to improve efficiency.

Many ad hoc initialization techniques have been proposed for specific clustering methods, for example, specific choices of the initial cluster centers of the classical k-means method (see e.g. [1, 2, 3, 4]), or singular value decomposition for clustering based on nonnegative matrix factorization [5, 6]. However, there seems to be no initialization principle that would be commonly applicable for a wide range of iterative clustering methods. Especially, there is little research on whether one clustering method can benefit from initializations by the results of another clustering method.

In this paper, we show experimentally that the clusterings can usually be improved if a set of diverse clustering methods provide initializations for each other. We name this approach *co-initialization*. We present a hierarchy of initializations towards this direction, where a higher level represents a more extensive strategy. At the top are two levels of co-initialization strategies. We point out that despite their extra computational cost, these strategies can often bring significantly enhanced clustering performance. The enhancement is especially significant for more complex clustering objectives, for example, Probabilistic

Latent Semantic Indexing [7], and our recent clustering method by low-rank doubly stochastic matrix decomposition (called DCD) [8].

Our claims are supported by extensive experiments on nineteen real-world clustering tasks. We have used a variety of datasets from different domains such as text, vision, and biology. The proposed initialization hierarchy has been tested using eight state-of-the-art clustering methods. Two widely used criteria, cluster purity and Normalized Mutual Information, are used to measure the clustering performance. The experimental results verify that a higher level initialization in the proposed hierarchy often achieve better clustering performance.

Ensemble clustering is another way to combine a set of clustering methods. It aggregates the different clusterings into a single one. We also compared co-initialization with three prominent ensemble clustering methods. The comparison results show that the improved DCD using co-initializations outperforms these ensemble approaches that seek a consensus clustering.

In the following, Section 2 reviews briefly the recently introduced Data-Cluster-Data (DCD) method. It is a representative clustering method among those that strongly benefit from co-initializations, and will be shown to be overall the best method in the experiments. Then Section 3 reviews related work on ensemble clustering, which is another way of combining a set of base clustering methods. In Section 4, we present our novel co-initialization method and describe the initialization hierarchy. Experimental settings and results are reported in Section 5. Section 6 concludes the paper and discusses potential future work.

2. Clustering by DCD

Some clustering methods such as Normalized Cut [9] are not sensitive to initializations but tend to return less accurate clustering (see e.g. [10], page 8, [8, 11], and Section 5.3). On the other hand, some methods can find more accurate results but require careful initialization. The latter kind of methods can benefit more from our co-initialization strategy, to be introduced in Section 4. Recently we proposed a typical clustering method of the latter kind, which is based on Data-Cluster-Data random walk and thus called DCD [8]. In this section we recapitulate the essence of DCD. It belongs to the class of probabilistic clustering methods. Given n data samples and r clusters, denote by $P(k|i)$ the probability of assigning the ith sample to the kth cluster, where $i = 1, \dots, n$ and $k = 1, \dots, r$.

Suppose the similarities between data items are precomputed and given in an $n \times n$ nonnegative symmetric sparse matrix A. DCD seeks an approximation to A by another matrix $\widehat{A}$ whose elements correspond to the probabilities of two-step random walks between data points through clusters. Let i, j, and v be indices for data points, and k and l for clusters. Then the random walk probabilities are given as

$$\widehat{A}_{ij} = P(i|j) = \sum_k P(i|k)P(k|j) = \sum_k \frac{P(k|i)P(k|j)}{\sum_v P(k|v)}, \tag{1}$$

by using the Bayes formula and the uniform prior $P(i) = 1/n$.

The approximation is given by the Kullback-Leibler (KL-) divergence. This is formulated as the following optimization problem [8]:

$$\underset{P \geq 0}{\text{minimize}} \quad D_{\text{KL}}(A||\widehat{A}) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{\widehat{A}_{ij}} - A_{ij} + \widehat{A}_{ij} \right), \tag{2}$$

where $\widehat{A}_{ij} = \sum_k \frac{P_{ik}P_{jk}}{\sum_v P_{vk}}$ with $P_{ik} = P(k|i)$, subject to $\sum_k P_{ik} = 1,\ i = 1, \dots, n$.

Denote $\nabla = \nabla^+ - \nabla^-$ as the gradient of $D_{\text{KL}}(A||\widehat{A})$ with respect to P, where ∇^+ and ∇^- are the positive and (unsigned) negative parts of ∇, respectively. The optimization is solved by a Majorization-Minimization algorithm [12, 13, 14, 15] that iteratively applies a multiplicative update rule:

$$P_{ik} \leftarrow P_{ik} \frac{\nabla^-_{ik} a_i + 1}{\nabla^+_{ik} a_i + b_i}, \tag{3}$$

where $a_i = \sum_l \frac{P_{il}}{\nabla^+_{il}}$ and $b_i = \sum_l P_{il} \frac{\nabla^-_{il}}{\nabla^+_{il}}$.

The preprocessing of DCD employs the common approximation of making A sparse by zeroing the non-local similarities. This makes sense for two reasons: first, geodesics of curved manifolds in high-dimensional spaces can only be approximated by Euclidean distances in small neighborhoods; second, most popular distances computed of weak or noisy indicators are not reliable over long distances, and the similarity matrix is often approximated by the K-nearest neighbor graph with good results, especially when n is large. With a sparse A, the computational cost of DCD is $O(|E| \times r)$ for $|E|$ nonzero entries in A and r clusters. In the experiments we used symmetrized and binarized K-Nearest-Neighbor graph as A ($K \ll n$). Thus the computational cost is $O(nKr)$.

Given a good initial decomposing matrix P, DCD can achieve better cluster purity compared with several other state-of-the-art clustering approaches, especially for large-scale datasets where the data points situate in a curved manifold. Its success comes from three elements in its objective: 1) the approximation error measure by Kullback-Leibler divergence takes into account sparse similarities; 2) the decomposing matrix P as the only variable to be learned contains just enough parameters for clustering; and 3) the decomposition form ensures relatively balanced clusters and equal contribution of each data sample.

What remains is how to get a good starting point. The DCD optimization problem is harder to solve than conventional NMF-type methods based on Euclidean distance in three aspects: 1) the geometry of the KL-divergence cost function is more complex; 2) DCD employs a structural decomposition where P appears more than once in the approximation, and appears in both numerator and denominator; 3) each row of P is constrained to be in the $(r-1)$-simplex. Therefore, finding a satisfactory DCD solution requires more careful initialization. Otherwise the optimization algorithm can easily fall into a poor local minimum.

Yang and Oja [8] proposed to obtain the starting points by pre-training DCD with regularization term $(1-\alpha)\sum_{ik} \log P_{ik}$. This corresponds to imposing

Dirichlet priors over the rows of P. By varying α, the pre-training can provide different starting points for multiple runs of DCD. The final result is given by the one with the smallest DCD objective of Eq. 2. This initialization strategy can bring improvement for certain datasets, whereas the enhancement remains mediocre as it is restricted to the same family of clustering methods. In the remaining, we investigate the possibility to obtain good starting points with the aid of other clustering methods.

3. Ensemble clustering

In supervised machine learning, it is known that combining a set of classifiers can produce better classification results (see e.g. [16]). There have been also research efforts with the same spirit in unsupervised learning, where several basic clusterings are combined into a single categorical output. The base results can come from results of several clustering methods, or the repeated runs of a single method with different initializations. In general, after obtaining the bases, a combining function, called *consensus function*, is needed for aggregating the clusterings into a single one. We call such aggregating methods *ensemble cluster analysis.*

Several ensemble clustering methods have been proposed. An early method [17] first transforms the base clusterings into a hypergraph and then uses a graph-partitioning algorithm to obtain the final clusters. Gionis and Mannila [18] defined the distance between two clusterings as the number of pairs of objects on which the two clusterings disagree, based on which they formulated the ensemble problem as the minimization of the total number of disagreements with all the given clusterings. Fred and Jain [19] explored the idea of evidence accumulation and proposed to summarize various clusterings in a co-association matrix. The incentive of their approach is to weight associations between sample pairs by the number of times they co-occur in a cluster from the set of given clusterings. After obtaining the co-association matrix, they applied the agglomerative clustering algorithm to yield the final partition. Iam-On et al.

[20] introduced new methods for generating two link-based pairwise similarity matrices called connected-triple-based similarity and SimRank-based similarity. They refined similarity matrices by considering both the associations among data points and those among clusters in the ensemble using link-based similarity measures. In their subsequent work [21], Iam-On et al. released a software package called LinkCluE for their link-based cluster ensemble framework. New approaches that better exploit the nature of the co-association matrix have recently appeared (see e.g. [22, 23]).

Despite the rationales for aggregation, the above methods can produce mediocre results if many base clustering methods fall into their poor local optima during their optimization. Seeking a consensus partition of such bases will not bring extraordinary improvement. To overcome this, in the following we present a new technique that enhances the participating clustering methods themselves. In the experimental part we show that our approach outperforms three well-known ensemble clustering methods.

4. Improving clustering by co-initializations

We consider a novel approach that makes use of a set of existing clustering methods. Instead of combining for consensus partitions, the proposed approach is based on two observations: 1) many clustering methods that use iterative optimization algorithms are sensitive to initializations; random starting guesses often lead to poor local optima; 2) on the other hand, the iterative algorithms often converge to a much better result given a starting point which is sufficiently close to the optimal result or the ground truth. These two observations inspired us to systematically study the behavior of an ensemble of clustering methods through *co-initializations*, i.e., providing starting guesses for each other.

The cluster assignment can be represented by an $n \times r$ binary matrix W, indicating the membership of the samples to clusters. Most state-of-the-art cluster analysis methods use a non-convex objective function over the indicator matrix W. The objective is usually optimized by an iterative optimization algorithm with a starting guess of the cluster assignment. The simplest way is to start from a random cluster assignment (*random initialization*). Typically the starting point is drawn from a uniform distribution. To find a better local optimum, one may repeat the optimization algorithm several times with different starting assignments (e.g. with different random seeds). "Soft" clustering has been introduced to reduce the computational cost in combinatorial optimization (see e.g. [24]), where the solution space of W is relaxed to right-stochastic matrices (e.g. [25]) or nonnegative nearly orthogonal matrices (e.g. [26, 14]). Initialization for these algorithms can be a cluster indicator matrix plus a small perturbation. This is in particular widely used in multiplicative optimization algorithms (e.g. [26]).

Random initialization is easy to program. However, in practice it often leads to clustering results which are far from a satisfactory partition, even if the clustering algorithm is repeated with tens of different random starting points. This drawback appears for various clustering methods using different evaluation criteria. See Section 5.3 for examples.

To improve clusterings, one can consider more complex initialization strategies. Especially, the cluster indicator matrix W may be initially set by the output of another clustering method instead of random initialization. One can use the result from a fast and computationally simple clustering method such as Normalized Cut (NCUT) [9] or k-means [27] as the starting point. We call the clustering method used for initialization the *base method* in contrast to the *main method*, used for the actual consequent cluster analysis. Because here the base method is simpler than the main clustering method, we call this strategy *simple initialization*. This strategy has been widely used in clustering methods with Nonnegative Matrix Factorization (e.g. [26, 24, 28]).

We point out that the clusterings can be further improved by more considerate initializations. Besides NCUT or k-means, one can consider any clustering methods for initialization, as long as they are different from the main method. The strategy where the base methods belong to the same parametric family is called *family initialization*. That is, both the base and the main methods use

Algorithm 1 Cluster analysis using heterogeneous initialization. We denote $W \leftarrow \mathcal{M}(\mathcal{D}, U)$ a run of clustering method $\mathcal{M}$ on data $\mathcal{D}$, with starting guess U and output cluster indicator matrix W. $\mathcal{J}_{\mathcal{M}}$ denotes the objective function of the main method.

1: Input: data $\mathcal{D}$, base clustering methods $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_T$, and main clustering method $\mathcal{M}$
2: Initialize $\{U_t\}_{t=1}^T$ by e.g. random or simple initialization
3: **for** $t = 1$ to T **do**
4: $\quad V \leftarrow \mathcal{B}_t(\mathcal{D}, U_t)$
5: $\quad W_t \leftarrow \mathcal{M}(\mathcal{D}, V)$
6: **end for**
7: Output: $W \leftarrow \arg\min_{W_t} \{\mathcal{J}_{\mathcal{M}}(\mathcal{D}, W_t)\}_{t=1}^T$.

the same form of objective and metric but only differ by a few parameters. For example, in the above DCD method, varying α in the Dirichlet prior can provide different base methods [8]; the main method ($\alpha = 1$) and the base methods ($\alpha \neq 1$) belong to the same parametric family. Removing the constraint of the same parameterized family, we can generalize this idea such that any clustering methods can be used as base methods and thus call the strategy *heterogeneous initialization*. Similar to the strategies for combining classifiers, it is reasonable to have base methods as diverse as possible for better exploration. The pseudocodes for heterogeneous initialization is given in Algorithm 1.

Deeper thinking in this direction gives a more comprehensive strategy called *heterogeneous co-initialization*, where we make no difference from base and main methods. The participating methods can provide initializations to each other. Such cooperative learning can run for more than one iteration. That is, when one algorithm finds a better local optimum, the resulting cluster assignment can again serve as the starting guess for the other clustering methods. The loop will converge when none of the involved methods can find a better local optimum. The convergence is guaranteed if the objective functions are all bounded. A special case of this strategy was used for combining NMF and Probabilistic Latent Semantic Indexing [29]. Here we generalize this idea to any participating clustering methods. The pseudo-code for heterogeneous co-initialization is given in Algorithm 2.

Algorithm 2 Cluster analysis using heterogeneous co-initialization. $\mathcal{J}_{\mathcal{M}_i}$ denotes the objective function of method $\mathcal{M}_i$.

Input: data $\mathcal{D}$ and clustering methods $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_T$
$\mathcal{J}_t \leftarrow \infty, t = 1, \ldots, T$.
Initialize $\{W_t\}_{t=1}^T$ by e.g. random or simple initialization
repeat
    bContinue←False
    for $i = 1$ to T **do**
        for $j = 1$ to T **do**
            if $i \neq j$ **then**
                $U_i \leftarrow \mathcal{M}_i(\mathcal{D}, W_j)$
            end if
        end for
        $\mathcal{J} \leftarrow \min\limits_{U_j} \{\mathcal{J}_{M_j}(\mathcal{D}, U_j)\}_{j=1}^T$
        $V \leftarrow \arg\min\limits_{U_j} \{\mathcal{J}_{M_j}(\mathcal{D}, U_j)\}_{j=1}^T$
        if $\mathcal{J} < \mathcal{J}_i$ **then**
            $\mathcal{J}_i \leftarrow \mathcal{J}$
            $W_i \leftarrow V$
            bContinue←True
        end if
    end for
until bContinue=False or maximum iteration is reached
Output: $\{W_t\}_{t=1}^T$.

By this level of initialization, each participating method will give their own clusterings. Usually, methods that can find accurate results but require more careful initialization will get more improved than those that are less sensitive to initialization but give less accurate clusterings. Therefore, if a single clustering is wanted, we suggest the output of the former kind. For example, DCD can significantly be improved by using co-initialization. We thus select its result as the single clustering as output of *heterogeneous co-initialization* in the experiments in Section 5.4.

In Table 1, we summarize the above initialization strategies in a hierarchy. The computational cost increases along the hierarchy from low to high levels. We argue that the increased expense is often deserved for improving clustering quality, which will be justified by experiments in the following section. Note that the hierarchy was mentioned in our preliminary work [11].

Table 1: Summary of the initialization hierarchy for cluster analysis

level	name	description
0	random initialization	uses random starting points
1	simple initialization	initialize by a fast and computationally simple method such as k-means or NCUT
2	family initialization	uses base methods from a same parameterized family for initialization
3	heterogeneous initialization	uses any base methods to provide initialization for the main method
4	heterogeneous co-initialization	run in multiple iterations; in each iteration all participating methods provide initialization for each other

5. Experiments

We provide two groups of empirical results to demonstrate that 1) clustering performance can often be improved using more comprehensive initializations in the proposed hierarchy and 2) the new method outperforms three existing approaches that aggregate clusterings. All datasets and codes used in the experiments are available online[1].

5.1. Data sets

We focus on clustering tasks on real-world datasets. Nineteen publicly available datasets have been used in our experiments. They are from various domains, including text documents, astroparticles, face images, handwritten digit/letter images, protein. The sizes of these datasets range from a few hundreds to tens of thousands. The statistics of the datasets are summarized in Table 2. The data sources and descriptions are given in the supplemental document. For fair comparisons, we chose datasets whose ground truth classes are known.

The datasets are preprocessed as follows. We first extracted vectorial features for each data sample, in particular, scattering features [30] for images and Tf-Idf features for text documents. In machine learning and data analysis, the vectorial data often lie in a curved manifold, i.e. most simple metrics such as the Euclidean distance or cosine (here for Tf-Idf features) is only reliable in a small

[1] `http://users.ics.aalto.fi/hezhang/Clustering_co_init/`

Table 2: Statistics of the data sets.

DATASET	# SAMPLES	# CLASSES
ORL	400	40
MED	696	25
VOWEL	990	11
COIL20	1440	20
SEMEION	1593	10
FAULTS	1941	7
SEGMENT	2310	7
CORA	2708	7
CITESEER	3312	6
7SECTORS	4556	7
OPTDIGITS	5620	10
SVMGUIDE1	7089	2
ZIP	9298	10
USPS	9298	10
PENDIGITS	10992	10
PROTEIN	17766	3
20NEWS	19938	20
LET-REC	20000	26
MNIST	70000	10

neighborhood. We employed K-Nearest-Neighbor (KNN) graph to encode such local information. The choice of K is not very sensitive for large-scale datasets. Here we fix $K = 10$ for all datasets. We symmetrized the affinity matrix A: $A_{ij} = 1$ if i is one of the K nearest neighbors of j, or vice versa, and $A_{ij} = 0$ otherwise.

5.2. Evaluation criteria

The performance of cluster analysis is evaluated by comparing the resulting clusters to ground truth classes. We have adopted two widely used criteria:

- *purity* (e.g. [26, 8]), computed as

$$\text{purity} = \frac{1}{n} \sum_{k=1}^{r} \max_{1 \le l \le q} n_k^l, \tag{4}$$

  where n_k^l is the number of vertices in the partition k that belong to ground-truth class l;

- *normalized mutual information* [17], computed as

$$\mathrm{NMI} = \frac{\sum_{i=1}^{r}\sum_{j=1}^{r'} n_{i,j}\log\left(\frac{n_{i,j}n}{n_i m_j}\right)}{\sqrt{\sum_{i=1}^{r} n_i \log\left(\frac{n_i}{n}\right)\sum_{j=1}^{r'} m_j \log\left(\frac{m_j}{n}\right)}}, \tag{5}$$

  where r and r' respectively denote the number of clusters and classes; $n_{i,j}$ is the number of data points agreed by cluster i and class j; n_i and m_j denote the number of data points in cluster i and class j respectively; and n is the total number of data points in the dataset.

For a given partition of the data, all the above measures give a value between 0 and 1. A larger value in general indicates a better clustering performance.

5.3. Clustering with initializations at different levels

In the first group of experiments, we have tested various clustering methods with different initializations in the hierarchy described in Section 4. We focus on the following four levels: *random initialization*, *simple initialization*, *heterogeneous initialization*, and *heterogeneous co-initialization* in these experiments, while treating *family initialization* as a special case of *heterogeneous initialization*. These levels of initializations have been applied to six clustering methods, which are

- Projective NMF (PNMF) [31, 32, 28],
- Nonnegative Spectral Clustering (NSC) [24],
- Symmetric Tri-Factor Orthogonal NMF (ONMF) [26],
- Probabilistic Latent Semantic Indexing (PLSI) [33],
- Left-Stochastic Matrix Decomposition (LSD) [25],
- Data-Cluster-Data random walks (DCD) [8].

For comparison, we also include the results of two other methods based on graph cut:

- Normalized Cut (NCUT) [9],
- 1-Spectral Ratio Cheeger Cut (1-SPEC) [34].

We have coded NSC, PNMF, ONMF, LSD, DCD, and PLSI using multiplicative updates and ran each of these programs for 10,000 iterations to ensure their convergence. Symmetric versions of PNMF and PLSI have been used. We adopted the 1-SPEC software by Hein and Bühler[2] with its default setting. Following Yang and Oja [8], we employed four different Dirichlet priors ($\alpha = 1, 1.2, 2, 5$) for DCD, where each with a different prior is treated as a different method in *heterogeneous initialization* and *heterogeneous co-initialization*.

For *random initialization*, we ran the clustering methods with fifty starting points, each with a different random seed, and then record the result with the best objective. For *simple initialization*, we employed NCUT to provide initialization for the six non-graph-cut methods. Precisely, their starting (relaxed) indicator matrix is given by the NCUT result plus 0.2. This same scheme is used in *heterogeneous initialization* and *heterogeneous co-initialization* where one method is initialized by another. For *heterogeneous co-initialization*, the number of co-initialization iterations was set to 5, as in practice we found that there is no significant improvement after five rounds. For any initialization and clustering method, the learned result gives an objective no worse than the initialization.

Table 3 shows the clustering performance comparison. For clarity, only the DCD results using $\alpha = 1$, i.e., with a uniform prior, are listed in the table, while the complete clustering results including three other Dirichlet priors are given in the supplemental document.

There are two types of methods: NCUT and 1-SPEC are of the first type and they are insensitive to starting points, though their results are often mediocre when compared to the best in each row, especially for large datasets. The second type of methods include the other six methods, whose performance depends on

[2] `http://www.ml.uni-saarland.de/code/oneSpectralClustering/oneSpectralClustering.html`

initializations. Their results are given in cells with quadruples. We can see that more comprehensive initialization strategies often lead to better clusterings, where the four numbers in most cells monotonically increase from left to right.

In particular, improvement brought by co-initializations is more often and significant for PLSI and DCD. For example, Table 3 (top) shows that DCD for `USPS` dataset, the purity of *heterogeneous co-initialization* is 5% better than *heterogeneous initialization*, 10% better than *simple initialization*, and 34% better than *random initialization*. The advantage of co-initialization for PLSI and DCD is because these two methods are based on Kullback-Leibler (KL-) divergence. This divergence is more suitable for sparse input similarities due to curved manifolds [8]. However, the objective using KL-divergence involves a more sophisticated surface and is thus difficult to optimize. Therefore these methods require more considerate initializations that provide a good starting point. In contrast, objectives of PNMF, NSC, ONMF and LSD, are relatively easier to optimize. These methods often perform better than PLSI and DCD using lower levels of initializations. However, more comprehensive initializations may not improve their clusterings (see e.g. ONMF for `OPTDIGITS`). This can be explained by their improper modeling of sparse input similarities based on the Euclidean distance such that more probably a better objective of these methods may not correspond to better clustering performance.

The improvement pattern becomes clearer when the dataset is larger. Especially, PLSI and DCD achieve remarkable 0.98 purity for the largest dataset `MNIST`. Note that purity corresponds to classification accuracy up to permutation between clusters and classes. This means that our unsupervised cluster analysis results are already very close to state-of-the-art supervised classification results[3]. A similar purity was reported in DCD using *family initialization*. Our experiments show that it is also achievable for other clustering methods given co-initializations, with even better results by PLSI and DCD.

[3] see `http://yann.lecun.com/exdb/mnist/`

5.4. Comparison with ensemble clustering

Our approach uses a set of clustering methods and outputs a final partition of data samples. There exists another way to combine clustering algorithms: ensemble clustering, which was reviewed in Section 3 . Therefore, we have compared our co-initialization method with three ensemble clustering methods in the second group of experiments: the BEST algorithm [18], the co-association algorithm (CO) [19], and the link-based algorithm (CTS) [20]. We coded the BEST and CO algorithms by ourselves and ran the CTS algorithm using LinkCluE package [21]. For fair comparison, the set of base methods (i.e. same objective and same optimization algorithm) is the same for all compared approaches: the 11 bases are from NCUT, 1-SPEC, PNMF, NSC, ONMF, LSD, PLSI, DCD1, DCD1.2, DCD2, and DCD5 respectively. The data input for a particular base method is also exactly the same across different combining approaches. The final partitions for CO and CTS were given by the complete-linkage hierarchical clustering algorithm provided in [21].

Different from the other compared approaches, our method actually does not average the clusterings. Each participating clustering method in co-initializations gives their own results, according to the *heterogeneous-co-initialization* pseudocode in Algorithm 2. Here we chose the result by DCD for the comparison with the ensemble methods, as we find that this method benefits the most from co-initializations.

The comparison results are shown in Table 4. We can see that DCD wins most clustering tasks, where it achieves the best for 16 out of 19 datasets in terms of purity, and 18 out of 19 in terms of NMI. The superiority of DCD using co-initializations is especially distinct for large datasets. DCD clearly wins for all but the smallest datasets.

6. Conclusions

We have presented a new method for improving clustering performance through a collection of clustering methods. Different from conventional combining scheme that seeks consensus as clustering, our method tries to find better starting points for the participating methods through their initializations for each other. The initialization strategies can be organized in a hierarchy from simple to complex. By extensive experiments on real-world datasets, we have shown that 1) higher-level initialization strategies in the hierarchy can often lead to better clustering performance and 2) the co-initialization method can significantly outperform conventional ensemble clustering methods that average input clusterings.

Our findings reflect the importance of pre-training in cluster analysis. There could be more future steps towards this direction. Currently the participating methods are chosen heuristically. A more rigorous and computable diversity measure between clustering methods could be helpful for more efficient co-initializations. A meta probabilistic clustering framework might be also advantageous, where the starting points are sampled from more informative priors instead of the uniform distribution.

The proposed co-initialization strategy shares similarities with evolutionary algorithms (EA) or genetic algorithms (GA) that use different starting points and combine them to make new solution [35, 36]. In the future work, it would be interesting to find more precise connection between our approach with EA/GA, which could in turn generalize co-initializations to a more powerful framework.

References

[1] P. Bradley, U. Fayyad, Refining initial points for k-means clustering, in: International Conference on Machine Learning (ICML), 1998, pp. 91–99.

[2] A. Likas, N. Vlassis, J. J Verbeek, The global k-means clustering algorithm, Pattern Recognition 36 (2) (2003) 451–461.

[3] S. Khan, A. Ahmad, Cluster center initialization algorithm for k-means clustering, Pattern Recognition Letters 25 (11) (2004) 1293–1302.

[4] M. Erisoglu, N. Calis, S. Sakallioglu, A new algorithm for initial cluster centers in k-means algorithm, Pattern Recognition Letters 32 (14) (2011) 1701–1705.

[5] Z. Zheng, J. Yang, Y. Zhu, Initialization enhancer for non-negative matrix factorization, Engineering Applications of Artificial Intelligence 20 (1) (2007) 101–110.

[6] Y. Kim, S. Choi, A method of initialization for nonnegative matrix factorization, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2007, pp. 537–540.

[7] T. Hofmann, Probabilistic latent semantic indexing, in: International Conference on Research and Development in Information Retrieval (SIGIR), 1999, pp. 50–57.

[8] Z. Yang, E. Oja, Clustering by low-rank doubly stochastic matrix decomposition, in: International Conference on Machine Learning (ICML), 2012, pp. 831–838.

[9] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[10] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Transactions on Pattern Analysis Machine Intelligence 33 (5) (2011) 898–916.

[11] Z. Yang, T. Hao, O. Dikmen, X. Chen, E. Oja, Clustering by nonnegative matrix factorization using graph random walk, in: Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1088–1096.

[12] D. R. Hunter, K. Lange, A tutorial on MM algorithms, The American Statistician 58 (1) (2004) 30–37.

[13] Z. Yang, E. Oja, Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization, IEEE Transactions on Neural Networks 22 (12) (2011) 1878–1891.

[14] Z. Yang, E. Oja, Quadratic nonnegative matrix factorization, Pattern Recognition 45 (4) (2012) 1500–1510.

[15] Z. Zhu, Z. Yang, E. Oja, Multiplicative updates for learning with stochastic matrices, in: The 18th conference Scandinavian Conferences on Image Analysis (SCIA), 2013, pp. 143–152.

[16] E. Alpaydin, Introduction to Machine Learning, The MIT Press, 2010.

[17] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2003) 583–617.

[18] A. Gionis, H. Mannila, P. Tsaparas, Clustering aggregation, in: International Conference on Data Engineering (ICDE), IEEE, 2005, pp. 341–352.

[19] A. Fred, A. Jain, Combining multiple clusterings using evidence accumulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 835–850.

[20] N. Iam-On, T. Boongoen, S. Garrett, Refining pairwise similarity matrix for cluster ensemble problem with cluster relations, in: International Conference on Discovery Science (DS), Springer, 2008, pp. 222–233.

[21] N. Iam-On, S. Garrett, Linkclue: A matlab package for link-based cluster ensembles, Journal of Statistical Software 36 (9) (2010) 1–36.

[22] S. Rota Bulò, A. Lourenço, A. Fred, M. Pelillo, Pairwise probabilistic clustering using evidence accumulation, in: International Workshop on Statistical Techniques in Pattern Recognition (SPR), 2010, pp. 395–404.

[23] A. Lourenço, S. Rota Bulò, N. Rebagliati, A. Fred, M. Figueiredo, M. Pelillo, Probabilistic consensus clustering using evidence accumulation, Machine Learning, in press (2013).

[24] C. Ding, T. Li, M. Jordan, Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding,

in: IEEE International Conference on Data Mining (ICDM), IEEE, 2008, pp. 183–192.

[25] R. Arora, M. Gupta, A. Kapila, M. Fazel, Clustering by left-stochastic matrix factorization, in: International Conference on Machine Learning (ICML), 2011, pp. 761–768.

[26] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: International Conference on Knowledge Discovery and Data Mining (SIGKDD), ACM, 2006, pp. 126–135.

[27] S. Lloyd, Last square quantization in pcm, IEEE Transactions on Information Theory, special issue on quantization 28 (1982) 129–137.

[28] Z. Yang, E. Oja, Linear and nonlinear projective nonnegative matrix factorization, IEEE Transactions on Neural Networks 21 (5) (2010) 734–749.

[29] C. Ding, T. Li, W. Peng, On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing, Computational Statistics & Data Analysis 52 (8) (2008) 3913–3927.

[30] S. Mallat, Group invariant scattering, Communications on Pure and Applied Mathematics 65 (10) (2012) 1331–1398.

[31] Z. Yuan, E. Oja, Projective nonnegative matrix factorization for image compression and feature extraction, in: Proceedings of 14th Scandinavian Conference on Image Analysis (SCIA), Joensuu, Finland, 2005, pp. 333–342.

[32] Z. Yang, Z. Yuan, J. Laaksonen, Projective non-negative matrix factorization with applications to facial image processing, International Journal on Pattern Recognition and Artificial Intelligence 21 (8) (2007) 1353–1362.

[33] T. Hofmann, Probabilistic latent semantic indexing, in: International Conference on Research and Development in Information Retrieval (SIGIR), ACM, 1999, pp. 50–57.

[34] M. Hein, T. Bühler, An inverse power method for nonlinear eigenproblems with applications in 1-Spectral clustering and sparse PCA, in: Advances in Neural Information Processing Systems (NIPS), 2010, pp. 847–855.

[35] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, A. P. L. F. De Carvalho, A survey of evolutionary algorithms for clustering, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39 (2) (2009) 133–155.

[36] M. J. Abul Hasan, S. Ramakrishnan, A survey: hybrid evolutionary algorithms for cluster analysis, Artificial Intelligence Review 36 (3) (2011) 179–204.

Table 3: Clustering performance of various clustering methods with different initializations. Performances are measured by (top) Purity and (bottom) NMI. Rows are ordered by dataset sizes. In cells with quadruples, the four numbers from left to right are results using *random*, *simple*, and *heterogeneous initialization* and *heterogeneous co-initialization*.

DATASET	KM	NCUT	1-SPEC	PNMF	NSC	ONMF	LSD	PLSI	DCD
ORL	0.70	0.81	0.81	0.81 0.81 0.81 0.81	0.81 0.81 0.81 0.81	0.53 0.78 0.80 0.80	0.82 0.82 0.82 0.82	0.65 0.81 0.83 0.83	0.67 0.81 0.83 0.83
MED	0.59	0.57	0.53	0.57 0.57 0.56 0.56	0.57 0.59 0.57 0.57	0.51 0.57 0.56 0.56	0.57 0.57 0.57 0.57	0.57 0.57 0.57 0.58	0.57 0.57 0.57 0.58
VOWEL	0.40	0.35	0.34	0.38 0.35 0.37 0.38	0.35 0.35 0.35 0.35	0.36 0.35 0.37 0.38	0.34 0.37 0.38 0.40	0.34 0.36 0.38 0.40	0.28 0.36 0.36 0.40
COIL20	0.63	0.71	0.67	0.67 0.71 0.71 0.71	0.73 0.71 0.72 0.72	0.63 0.71 0.72 0.72	0.71 0.68 0.68 0.68	0.58 0.75 0.69 0.70	0.62 0.75 0.69 0.70
SEMEION	0.68	0.64	0.66	0.60 0.66 0.60 0.60	0.66 0.66 0.66 0.66	0.62 0.60 0.60 0.60	0.76 0.72 0.74 0.75	0.67 0.65 0.74 0.77	0.68 0.65 0.75 0.77
FAULTS	0.42	0.40	0.40	0.42 0.45 0.45 0.45	0.44 0.39 0.38 0.38	0.42 0.45 0.42 0.42	0.39 0.44 0.43 0.43	0.35 0.41 0.44 0.44	0.35 0.40 0.43 0.44
SEGMENT	0.59	0.61	0.55	0.49 0.54 0.49 0.53	0.39 0.61 0.69 0.71	0.49 0.51 0.53 0.53	0.30 0.64 0.61 0.65	0.26 0.62 0.64 0.65	0.26 0.61 0.61 0.65
CORA	0.53	0.39	0.36	0.41 0.36 0.41 0.41	0.36 0.36 0.36 0.36	0.34 0.36 0.43 0.43	0.48 0.51 0.51 0.54	0.41 0.45 0.52 0.55	0.41 0.45 0.52 0.55
CITESEER	0.61	0.30	0.31	0.28 0.29 0.29 0.28	0.26 0.28 0.25 0.25	0.31 0.32 0.28 0.28	0.38 0.43 0.45 0.47	0.35 0.44 0.44 0.48	0.37 0.44 0.44 0.48
7SECTORS	0.39	0.25	0.25	0.29 0.29 0.29 0.29	0.26 0.25 0.25 0.25	0.24 0.29 0.29 0.29	0.27 0.37 0.40 0.35	0.27 0.37 0.40 0.35	0.30 0.37 0.40 0.35
OPTDIGITS	0.72	0.74	0.76	0.70 0.68 0.68 0.68	0.66 0.77 0.77 0.77	0.68 0.68 0.68 0.68	0.71 0.76 0.82 0.87	0.51 0.72 0.76 0.85	0.57 0.76 0.71 0.85
SVMGUIDE1	0.71	0.75	0.93	0.68 0.68 0.68 0.68	0.82 0.77 0.79 0.81	0.68 0.68 0.68 0.68	0.70 0.78 0.91 0.91	0.59 0.77 0.90 0.91	0.58 0.78 0.90 0.91
ZIP	0.49	0.74	0.74	0.54 0.70 0.72 0.68	0.65 0.74 0.74 0.74	0.55 0.70 0.67 0.68	0.72 0.84 0.83 0.85	0.33 0.74 0.84 0.84	0.36 0.76 0.84 0.84
USPS	0.74	0.74	0.74	0.67 0.80 0.75 0.68	0.72 0.74 0.74 0.74	0.62 0.80 0.75 0.68	0.80 0.79 0.84 0.85	0.48 0.73 0.80 0.85	0.51 0.75 0.80 0.85
PENDIGITS	0.72	0.80	0.73	0.79 0.79 0.79 0.79	0.49 0.79 0.73 0.73	0.63 0.79 0.79 0.79	0.66 0.88 0.89 0.89	0.24 0.82 0.88 0.89	0.25 0.84 0.88 0.89
PROTEIN	0.46	0.46	0.46	0.46 0.46 0.46 0.46	0.46 0.46 0.46 0.46	0.46 0.46 0.46 0.46	0.46 0.46 0.48 0.50	0.46 0.51 0.51 0.50	0.47 0.46 0.51 0.50
20NEWS	0.07	0.43	0.36	0.39 0.39 0.39 0.38	0.39 0.43 0.40 0.40	0.27 0.38 0.38 0.38	0.41 0.48 0.48 0.49	0.22 0.44 0.49 0.49	0.23 0.45 0.49 0.50
LET-REC	0.29	0.21	0.15	0.36 0.37 0.34 0.35	0.17 0.21 0.21 0.21	0.29 0.35 0.35 0.34	0.31 0.31 0.37 0.37	0.16 0.26 0.32 0.38	0.17 0.25 0.32 0.38
MNIST	0.60	0.77	0.88	0.57 0.87 0.73 0.57	0.74 0.79 0.79 0.79	0.57 0.75 0.65 0.57	0.93 0.75 0.97 0.97	0.46 0.79 0.97 0.98	0.55 0.81 0.97 0.98

DATASET	KM	NCUT	1-SPEC	PNMF	NSC	ONMF	LSD	PLSI	DCD
ORL	0.85	0.90	0.92	0.89 0.90 0.89 0.89	0.89 0.90 0.90 0.90	0.76 0.88 0.89 0.89	0.90 0.90 0.90 0.90	0.84 0.90 0.91 0.91	0.83 0.90 0.91 0.91
MED	0.55	0.57	0.52	0.56 0.55 0.55 0.55	0.57 0.58 0.57 0.57	0.51 0.57 0.56 0.56	0.56 0.57 0.57 0.57	0.56 0.56 0.57 0.58	0.56 0.57 0.57 0.58
VOWEL	0.43	0.40	0.38	0.39 0.36 0.37 0.39	0.37 0.38 0.37 0.37	0.37 0.36 0.37 0.39	0.35 0.38 0.40 0.40	0.32 0.40 0.38 0.41	0.28 0.39 0.38 0.40
COIL20	0.77	0.79	0.77	0.75 0.79 0.79 0.79	0.81 0.79 0.80 0.80	0.74 0.79 0.79 0.79	0.79 0.78 0.77 0.77	0.71 0.80 0.80 0.80	0.74 0.80 0.80 0.80
SEMEION	0.57	0.61	0.62	0.58 0.62 0.58 0.58	0.63 0.63 0.63 0.63	0.59 0.57 0.57 0.57	0.67 0.63 0.65 0.66	0.59 0.61 0.66 0.68	0.61 0.61 0.67 0.68
FAULTS	0.10	0.08	0.09	0.09 0.11 0.11 0.11	0.10 0.07 0.07 0.07	0.10 0.11 0.09 0.09	0.06 0.11 0.11 0.11	0.03 0.08 0.12 0.11	0.02 0.08 0.11 0.11
SEGMENT	0.58	0.55	0.58	0.43 0.48 0.43 0.49	0.25 0.56 0.62 0.63	0.38 0.46 0.44 0.44	0.13 0.53 0.51 0.58	0.08 0.55 0.53 0.58	0.07 0.55 0.53 0.58
CORA	0.34	0.16	0.14	0.14 0.13 0.17 0.17	0.14 0.14 0.14 0.14	0.11 0.13 0.17 0.17	0.22 0.24 0.23 0.25	0.15 0.20 0.24 0.25	0.15 0.20 0.24 0.25
CITESEER	0.34	0.10	0.12	0.07 0.07 0.07 0.07	0.07 0.08 0.07 0.07	0.10 0.12 0.07 0.07	0.13 0.18 0.20 0.20	0.10 0.17 0.19 0.21	0.11 0.17 0.18 0.21
7SECTORS	0.17	0.04	0.05	0.05 0.04 0.04 0.04	0.05 0.04 0.05 0.05	0.01 0.04 0.04 0.04	0.04 0.10 0.14 0.11	0.04 0.07 0.13 0.11	0.04 0.08 0.13 0.11
OPTDIGITS	0.70	0.72	0.80	0.67 0.68 0.68 0.67	0.66 0.77 0.78 0.78	0.67 0.68 0.67 0.67	0.69 0.72 0.78 0.83	0.40 0.71 0.73 0.82	0.51 0.74 0.69 0.82
SVMGUIDE1	0.31	0.35	0.65	0.27 0.27 0.27 0.27	0.34 0.39 0.41 0.44	0.27 0.27 0.27 0.27	0.12 0.25 0.60 0.60	0.02 0.38 0.59 0.59	0.02 0.40 0.59 0.59
ZIP	0.40	0.78	0.79	0.54 0.67 0.65 0.64	0.61 0.78 0.78 0.78	0.56 0.66 0.62 0.64	0.66 0.78 0.80 0.81	0.18 0.77 0.79 0.81	0.21 0.78 0.79 0.81
USPS	0.62	0.77	0.80	0.66 0.75 0.71 0.66	0.71 0.78 0.78 0.78	0.62 0.75 0.71 0.66	0.75 0.77 0.81 0.82	0.40 0.75 0.77 0.81	0.46 0.76 0.77 0.81
PENDIGITS	0.68	0.81	0.78	0.78 0.78 0.78 0.78	0.51 0.79 0.79 0.78	0.63 0.78 0.77 0.77	0.61 0.83 0.86 0.86	0.10 0.81 0.83 0.86	0.10 0.81 0.83 0.86
PROTEIN	0.00	0.01	0.01	0.02 0.01 0.01 0.02	0.01 0.01 0.01 0.01	0.00 0.01 0.01 0.00	0.01 0.00 0.02 0.04	0.01 0.04 0.04 0.04	0.02 0.01 0.04 0.04
20NEWS	0.05	0.54	0.52	0.36 0.36 0.36 0.34	0.48 0.54 0.52 0.52	0.24 0.34 0.34 0.34	0.36 0.43 0.44 0.44	0.14 0.44 0.44 0.45	0.14 0.45 0.44 0.45
LET-REC	0.35	0.38	0.26	0.43 0.43 0.42 0.43	0.21 0.38 0.37 0.37	0.35 0.43 0.43 0.42	0.39 0.41 0.45 0.45	0.17 0.37 0.42 0.46	0.18 0.36 0.42 0.46
MNIST	0.51	0.81	0.89	0.59 0.82 0.72 0.59	0.73 0.84 0.84 0.84	0.58 0.75 0.64 0.59	0.87 0.76 0.93 0.93	0.34 0.81 0.92 0.94	0.48 0.80 0.92 0.93

Table 4: Clustering performance comparison of DCD using *heterogeneous co-initialization* with three ensemble clustering methods. Rows are ordered by dataset sizes. Boldface numbers indicate the best. The 11 bases are from NCUT, 1-SPEC, PNMF, NSC, ONMF, LSD, PLSI, DCD1, DCD1.2, DCD2, and DCD5 respectively.

	Purity				NMI			
DATASET	BEST	CO	CTS	DCD	BEST	CO	CTS	DCD
ORL	0.81	0.81	0.80	**0.83**	0.90	0.90	0.90	**0.91**
MED	**0.59**	0.58	0.58	0.58	**0.58**	0.57	0.57	**0.58**
VOWEL	0.35	0.33	0.36	**0.40**	0.38	0.36	0.37	**0.40**
COIL20	**0.73**	0.69	0.72	0.70	**0.80**	0.77	0.79	**0.80**
SEMEION	0.65	0.61	0.65	**0.77**	0.61	0.56	0.62	**0.68**
FAULTS	0.40	0.39	0.41	**0.44**	0.08	0.08	0.09	**0.11**
SEGMENT	0.63	0.61	0.63	**0.65**	0.55	0.55	0.55	**0.58**
CORA	0.45	0.41	0.42	**0.55**	0.20	0.17	0.18	**0.25**
CITESEER	0.43	0.34	0.35	**0.48**	0.18	0.12	0.15	**0.21**
7SECTORS	**0.36**	0.27	0.25	0.35	0.08	0.04	0.04	**0.11**
OPTDIGITS	0.76	0.63	0.71	**0.85**	0.74	0.68	0.71	**0.82**
SVMGUIDE1	0.78	0.82	0.78	**0.91**	0.40	0.46	0.40	**0.59**
ZIP	0.74	0.62	0.76	**0.84**	0.78	0.70	0.80	**0.81**
USPS	0.75	0.65	0.73	**0.85**	0.76	0.69	0.78	**0.81**
PENDIGITS	0.84	0.84	0.81	**0.89**	0.81	0.81	0.82	**0.86**
PROTEIN	0.46	0.46	0.46	**0.50**	0.01	0.01	0.01	**0.04**
20NEWS	0.45	0.28	0.40	**0.50**	0.45	0.38	**0.47**	0.45
LET-REC	0.26	0.23	0.24	**0.38**	0.37	0.35	0.39	**0.46**
MNIST	0.96	0.57	0.76	**0.98**	0.92	0.68	0.84	**0.93**