# An Approach to User Feedback Processing in Order to Increase Clustering Results Quality

Pavel V. Dudarin and Nadezhda G. Yarushkina

*Ulyanovsk State Technical University, Ulyanovsk, Russia*

**Abstract**

Dataset clustering could have more than one "right" result depending on a user intention. For example, texts could be clustered according to their topic, style or author. In case of unsatisfactory results, a data scientist needs to re-construct a feature space in order to change the results. The relation between the feature space and the result are often quite complicated. The latter results in building several clustering models to explore useful relations. Interactive clustering with feedback is aimed to cope with this problem. In this paper an approach to user feedback processing during clustering is presented. The approach is based on end-to-end clustering and uses an autoencoder neural network. This technique allows to adjust iteratively the computing clusters without changing  feature space.

**Keywords 1**

Clustering, Interactive Clustering, Mixed-Initiative Clustering, Constrained Clustering, Semi-Supervised Clustering, End-to-End Clustering, Learning to Cluster, Clustering with Intent, Deep Embedding, Deep Representation, Feedback, Neural Networks.

## 1. Introduction

Clustering methods traditionally considered as unsupervised. Unsupervised learning is possible as long as data contains its meta information inside. Clustering methods are aimed to retrieve this meta information and use it to partition or hierarchically organize user data [18]. However in practice data scientists usually have background knowledge or at least a hypothesis about explored dataset. This is true for any kind of domain: economical [11], data collected from sensors, industrial devices [28] or data collected by computer program [15]. Almost in every clustering case an expert assistance is vital for data correction and validation, clustering structure correction or hierarchy changes, or it is useful for significant improvement of clustering result by means of expert knowledge which is not included in the data itself.  An expert assistance is of big importance in text clustering [12]. Clustering of short text is one of the most challenging tasks [14]. It is almost impossible to get a partition without providing additional information about user intent [13]. Apart from the obvious partitioning by topic, many other partitioning could be useful: type of person story, target auditory, legal status or a combination. It is not clear what is the rule to construct clusters. This problem is summarized as follows: "there is no right clustering, but there are useful" [Bae J. et al., 2020]. Expert feedback may be unavailable, but its embedding in the clustering process greatly improves the results and is much in demand by users [27]. But it is important that an expert be involved seamlessly in this process, and internal understanding of algorithm details was not necessary. A clustering algorithm should provide clear connection between expert knowledge and the result.

In [Bae J. et al., 2020] authors show growing interest in interactive clustering i.e., clustering with user feedback. Fig. 1. shows amount of studies and clustering methods used.

Russian Advances in Artificial Intelligence: selected contributions to the Russian Conference on Artificial intelligence (RCAI 2020), October 10-16, 2020, Moscow, Russia
EMAIL: p.dudarin@ulstu.ru (P. Dudarin); jng@ulstu.ru (N. Yarushkina)
ORCID: 0000-0003-2354-6527 (P. Dudarin); 0000-0002-5718-8732 (N. Yarushkina)

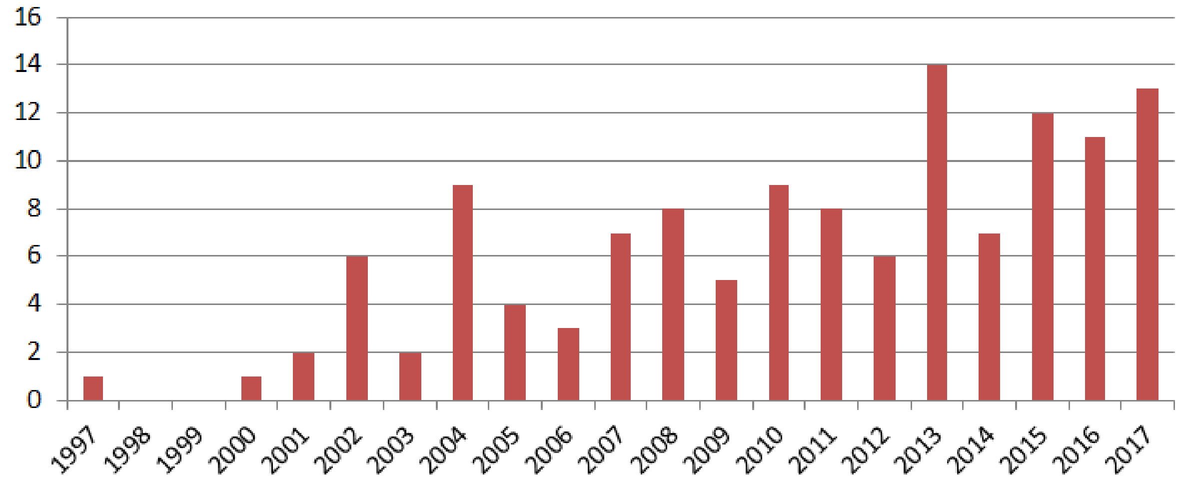


**Figure 1**: Amount of developed interactive clustering methods by years.

In this paper an approach to clustering under expert feedback is proposed. This approach allows to incorporate user feedback into a wide range of clustering methods based on neural networks, for example into DEC algorithm (Unsupervised Deep Embedding for Clustering Analysis) [31].

The rest of the paper is organized as follows: Section 2 contains a review and analysis of related work. In Section 3 we give a formal description of the problem. Section 4 introduces the proposed approach. In Section 5 we describe experiments and discuss their results. Section 6 concludes the paper.

## 2. Related work

In order to determine a circle of related work it is necessary to clarify definitions and classification of clustering methods. Modern scientific studies tend to name clustering methods where additional information is used, which is not included in the dataset, as semi-supervised clustering methods or constrained clustering methods [6]. The majority of studies in this field have additional information 'a priory', as input information with explored dataset. Usually, this information is given as pair-wise constraints [9], partially marked labels [5], constraints on a hierarchy structure or background information is given by means of transfer learning [30] (for example, as node weights from neural network trained on related classification problem). Besides, all the constraints and labels could be fuzzy (for example, soft labels) [24].

Meanwhile, there are methods receiving additional information during clustering process. A comprehensive review of these methods is done in [2]. These methods are called as interactive clustering methods. One of the first methods in this area was fuzzy and was proposed in [26]. Depend on interaction character and type of received information all the methods could be divided in groups: active clustering as a part of active learning field [16, 8]; reinforcement clustering, where feedback is received from natural or artificial environment [3]; interactive clustering with user feedback or mixed-initiative clustering, where feedback is received from a user as a reaction, correction command or evaluation of the clustering result. The last mentioned methods allow to reveal latent intentions of the user and obtain really useful clustering, because the user recognizes the right result when looks on it.

Many authors point that there are some types of studies considered as interactive clustering by mistake: methods with interactive visualization of clustering results, methods of interactive choice of clustering algorithms and some others [2].

Methods of assisting clustering also should be mentioned [7]. The leading role in these methods plays a user, which defines clusters, their structure, while algorithm just suggests candidate objects for each cluster. These methods are not widely used.

Interactive clustering methods with user feedback could be grouped according to the type of feedback. The first group, which includes the majority of studies, contains studies where a user iteratively and interactively could directly change algorithm parameters, similarity metric, clustering features [20]. The second group contains methods where a user interacts directly with the result of clustering. The user could point which clusters should be merged or split, directly move elements between clusters or decide what to do with outliers [4] and [2]. An approach proposed in this paper relates to the second group. The user does not need to know the algorithm details in order to use it and could switch between similar methods without changing the main character of its work.

The first step of clustering, evidently, is just an unsupervised clustering. So, all the interactive clustering methods are based on top of some unsupervised clustering algorithm by adding feedback processing into it. According to review [2] the majority of interactive clustering methods are based on: k-means, c-means, agglomerative clustering and graph clustering. A few studies use neural networks with SOM (Kohonen self-organized maps) architecture.

The classical clustering methods are successfully used in many cases and show high results, although the most of best results are shown by modern clustering methods. The majority of modern clustering methods are based on deep neural networks [22, 29, 32, 33]. Theirs advantage could be explained by ability of neural networks to transfer learning and learning to cluster, by complicated non-linear embedding used for feature construction (representation learning, embedding learning) which is more appropriate for clustering methods (for example, dimension reduction is a good way to increase clustering result quality) [35]. But the major contribution of neural network technique into clustering is a way to construct end-to-end clustering methods. There is no division into feature construction phase and partitioning phase in end-to-end clustering [23] and [17]. This allows to learn simultaneously clustering features that suites the best for the current task and to perform partitioning. There are studies where transfer learning abilities are shown: a neural network trained to cluster images in one domain was used to do the same task in another but related domain. There are some semi-supervised methods among those based on a neural network [30, 19], but they just use labels and pair-wise constraints to tune the loss function and these constraints could not be changed during the clustering process.

## 3. Related work

A study by [1] is dedicated to construction of meta-schema that generalizes taxonomy of modern clustering methods based on neural networks. The most of modern methods correspond to this schema (see Fig. 2.) [10, 31, 34]. The schema shows that feedback usage in a loss function allows to simultaneously fine-tune latent feature space according to user intent and perform clustering itself.


**Figure 2**: Generalized schema of clustering methods architecture based on neural networks.

Current study is aimed to construct interactive clustering methods with user feedback based on generalized schema described above. As a basic clustering method for this task method based on neural network with Kullback-Leibler divergence as a common loss function could be used. In particular, in this paper proposed a method based on DEC (Unsupervised Deep Embedding for Clustering Analysis) [31]. Feedback processing could be added to this method due to special properties of loss function based on Kullback-Leibler divergence that intuitively could be seen as a gravity controller between dataset objects and cluster centers. So, slight changes to the gravity between them lead to managed changes in clustering results.

## 4. An approach to user feedback processing

The proposed approach suggests the idea that the clustering result criticism is the most easy and precise feedback. So it allows two types of feedback: "object $X_i$ should be included into cluster $C_j$" and "object $X_i$ should not be included into cluster $C_j$". One portion of feedback could contain any number of constraints. For example, the swapping two elements between clusters implies to establish two constraints of the first type.

As it was mentioned above, the DEC (Unsupervised Deep Embedding for Clustering Analysis) algorithm has been chosen as a base for the proposed approach, but the same technique could be applied to many other methods, for example to DEPICT [10]. Input dataset $X=\{x_i \mid i \in [0, N)\}$, N – amount of objects in the dataset. The initial vector space of objects with a help of encoder (which is a part of pre-trained autoencoder) projects to another vector space of lower dimensionality: $f_\theta$: $X \rightarrow Z$, where $\theta$ – neural network parameters (weights), Z – latent feature space. Feature space Z is called latent, because it is constructed in unsupervised way during clustering process as a hidden layer of neural network. In this study an encoder of following structure is used: d-50-50-20-k, where d - dimensionality of input dataset, k – amount of clusters. The result of clustering algorithm is a set of cluster centers in a space Z: $\{\mu_j \in Z \mid j \in [0, k)\}$, where k – demanded amount of clusters. Cluster centers initialization is provided by k-means clustering over initial object representation obtained from autoencoder.

The process of cluster centers searching and the process of feature space construction performed simultaneously by means of common loss function. As a similarity measure between object and cluster center a metric based on Student's t-distribution with one degree of freedom (Q) is used.

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2)^{-1}}{\sum_{l=0}^{k}(1 + ||z_i - \mu_l||^2)^{-1}}).$$

Objective function (loss function) is composed as Kullback-Leibler divergence between actual distribution Q and auxiliary target distribution P.

$$L = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

Auxiliary target distribution P is defined as:

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{l=0}^{k} q_{il}^2/f_l}, \qquad f_j = \sum_j q_{ij}.$$

This distribution has some important properties: (1) strengthen predictions (i.e., improve cluster purity), (2) put more emphasis on data points assigned with high confidence, and (3) normalize loss contribution of each centroid to prevent large clusters from distorting the hidden feature space.

It is quite obvious that the loss function is aimed to make $q_{ij}$ greater than $p_{ij}$. Partial derivatives are:

$$\frac{\partial L}{\partial z_i} = 2 \sum_j \left(1 + \left|\left|z_i - \mu_j\right|\right|^2\right)^{-1} * (p_{ij} - q_{ij}) * (z_i - \mu_j),$$

$$\frac{\partial L}{\partial \mu_j} = -2 \sum_i \left(1 + \left|\left|z_i - \mu_j\right|\right|^2\right)^{-1} * (p_{ij} - q_{ij}) * (z_i - \mu_j).$$

It is evident that in the case of a negative value $(p_{ij}\text{-}q_{ij}) < 0$ an object $X_i$ will be pushed out of the cluster $C_j$, even though there is no loss from objective function.

This loss function is used during the first iteration of the algorithm while no feedback was provided. Then the user looking at the clustering result provides a feedback. In order to add feedback processing into this algorithm it is proposed to use following equation for partial derivatives instead:

$$\frac{\partial L}{\partial z_i} = 2 \sum_j \left(1 + \left|\left|z_i - \mu_j\right|\right|^2\right)^{-1} * \left|p_{ij} - q_{ij}\right| * t_{ij} * (z_i - \mu_j),$$

$$\frac{\partial L}{\partial \mu_j} = -2 \sum_i \left(1 + \left|\left|z_i - \mu_j\right|\right|^2\right)^{-1} * \left|p_{ij} - q_{ij}\right| * t_{ij} * (z_i - \mu_j),$$

Using T = {$t_{ij}$} –feedback matrix (user's tips), where :

$$t_{ij} = \begin{cases} > 0\ , to\ include\ object\ i\ into\ cluster\ j \\ < 0, \quad to\ exclude\ object\ i\ out\ of\ cluster\ j \\ 1, \quad otherwise. \end{cases}$$

Absolute value $t_{ij}$ defines a velocity (gravity force) of the objects and cluster centers movements. Also the learning rate of neural network influences this velocity. In experiments performed in this study value $|t_{ij}| = 1000$ was used for all the cases. Although experiments have shown that in cases with moving object out of the cluster it is better to use higher absolute values of $t_{ij}$.

Iteration with a user feedback are performed till the satisfactory result of clustering. Each iteration the user provides new feedback matrix. The neural networks weights and cluster centers are tuned according to received feedback.

## 5. Experiments description and result analysis

To demonstrate usefulness and effectiveness of the proposed approach two types of experiments have been done. Firstly, synthetic generated data set was used. This dataset consists of simple one-hot vectors. It seems that clustering of this data set is trivial, but in practice any partition of one-hot vectors could be right, depending on user intention. It will be shown below how the user could change a wrong partition to make it right. Secondly, an experiment with Fishers' Irises has been done. Fishers' Irises is a common dataset for classification and clustering tasks. There is no standard benchmark dataset for iterative clustering. However by clustering Fishers' Irises dataset the comparison with other clustering methods could be done. An ability to significantly increase clustering result quality will be demonstrated in this experiment.

### 5.1. Experiments with synthetic generated dataset

There were generated synthetic dataset consisted of 400 items of one-hot vectors, as follows:

1. As a base 4 classes of one-hot vectors {(1,0,0,0); (0,1,0,0); (0,0,1,0); (0,0,0,1)} have been defined.
2. Random noise from uniform distribution U[0, 1/10] was added to each vector to generate 125 vector variations just in order to augment data.
3. Projection into latent feature space for the first 12 vectors will be shown as a clustering result. 3 samples from each group. This is done to make figures clear without uninformative details.

A clustering process aimed to produce 2 clusters in this dataset was performed. Clustering results are shown in Table 1. Final value of autoencoder loss function was 0.000341. Figure 3 shows a distribution of first 12 vectors (in a latent feature space) from dataset on the plane. From this starting point 3 experiments were conducted according to different possible user feedback cases: vector $X_1$ should be included in $C_1$; vector $X_1$ should be moved out from $C_1$; complex feedback with a command to swap vectors $X_2$ and $X_3$. All the experiments were performed from one starting point in order to reduce amount of figures, but sequential user feedback processing is also possible without any limitations.

**Table 1**
Sample vectors list of synthetic generated dataset

| X | Coordinates | | | | Cluster |
|---|---|---|---|---|---|
| $X_0$ | **1.0191519** | 0.06221088 | 0.04377278 | 0.07853585 | $C_0$ |
| $X_1$ | 0.07799758 | **1.0272592** | 0.02764643 | 0.08018722 | $C_0$ |
| $X_2$ | 0.09581394 | 0.08759326 | **1.0357817** | 0.05009951 | $C_1$ |
| $X_3$ | 0.06834629 | 0.07127021 | 0.03702508 | **1.0561196** | $C_1$ |

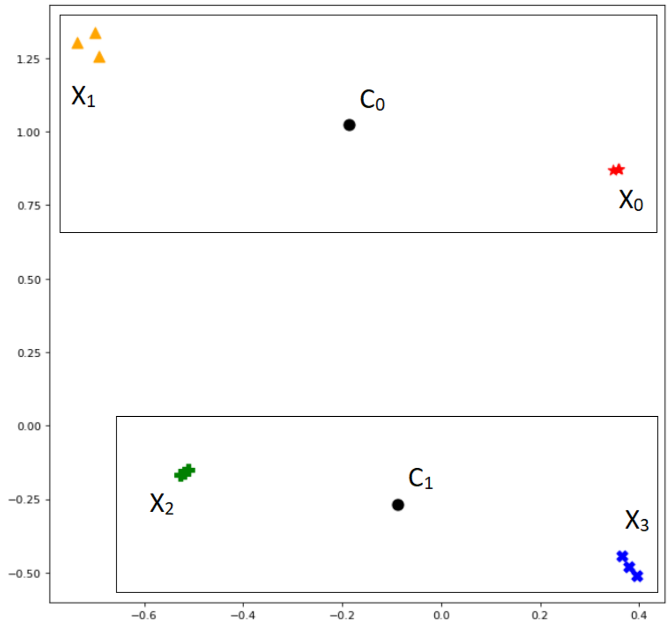


**Figure 3**: Clustering result (12 first vectors are shown).

The first experiment suggests that a user knows that vector $X_1$ is semantically closer to vectors $X_2$ and $X_3$ that to $X_1$. According to this, a feedback demanding that vector $X_1$ should be included into the cluster $C_1$ in a form of feedback matrix was provided: $T_{[500,4]} = \{t_{ij} \mid i \in [0, 500), j \in [0, 4)\}$, where

$$t_{ij} = \begin{cases} 1000, & i = 1, \quad j = 1; \\ 1, & otherwise. \end{cases}$$

Result of 100 epochs of clustering is shown on Fig. 4. As it could be seen all the 125 vectors from the second group have been moved with $X_1$ into $C_1$ (feedback has been provided joust for 1 vector). Also it is worth pointing out that semantic distance between the third and the fourth classes has been preserved, the same as for mutual arrangement of vectors inside each class.

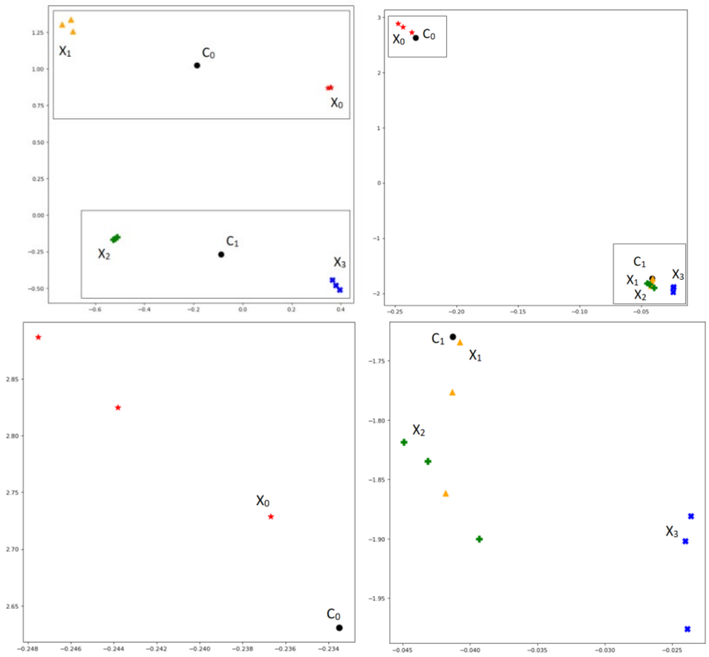


**Figure 4 (a,b,c,d)**: Clustering results of moving X1 into C1. 4a – result of the first stage of clustering, 4b – results after a user feedback provided, 4c and 4d – zoomed in resulting clusters.

The second experiment suggests that a user knows that vector $X_2$ is semantically could not belong to the same cluster with $X_3$. In this case user does not point out the cluster where $X_2$ should be included. According to this, a feedback demanding that vector $X_2$ should be moved out of the cluster $C_1$ in a form of feedback matrix was provided: $T_{[500,4]} = \{t_{ij} \mid i \in [0, 500), j \in [0, 4)\}$, where

$$t_{ij} = \begin{cases} -1000, & i = 2, \quad j = 1; \\ 1, & otherwise. \end{cases}$$

Result of 100 epochs of clustering is shown on Fig. 5. As it could be seen vector $X_2$ and all the vectors from the third class have been moved out of $C_1$ and were included into $C_0$. It worth pointing out that as long as "push out" constraint was used, the third class is located at the farthest possible position regarding to $C_1$. Also this experiment has shown lower velocity of convergence that the previous one.

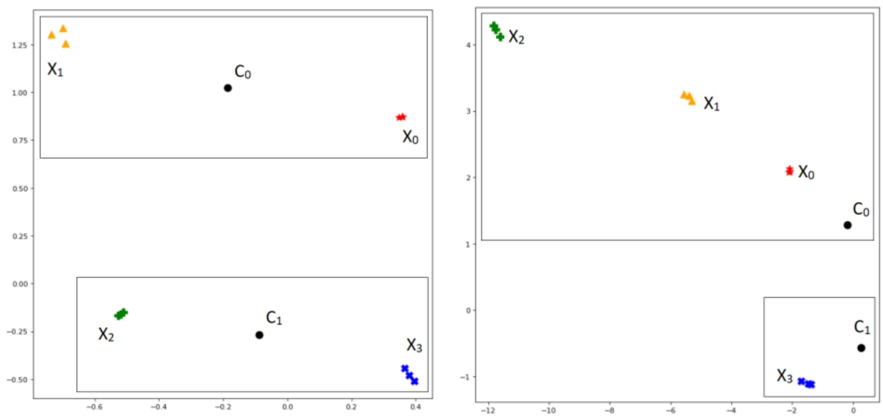


**Figure 5 (a,b)**: Clustering results of moving vector X2 out of C1. 5a – the first clustering stage results, 5b – results of iteration with a user feedback.

The third experiment suggests that a user knows that vector $X_1$ and $X_2$ should be swapped. According to this, a feedback matrix was provided: $T_{[500,4]} = \{t_{ij} \mid i \in [0, 500), j \in [0, 4)\}$, where

$$t_{ij} = \begin{cases} 1000, & i = 1, \quad j = 1; \\ 1000, & i = 2, \quad j = 0; \\ 1, & otherwise. \end{cases}$$

Result of 100 epochs of clustering is shown on Fig. 6. Not only the vectors $X_1$ and $X_2$ has been swapped, but the whole classes 2 and 3 too.

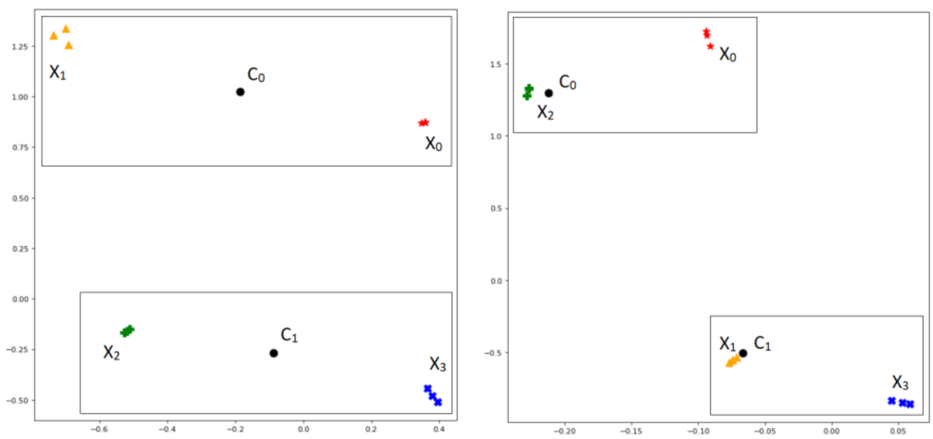


**Figure 6 (a,b)**: Clustering results of swapping two vectors (X1 and X2). 6a – the first clustering stage results, 6b – results of iteration with a user feedback.

For the fourth experiment noise level was 10 times increased using another uniform distribution U[0, 1]. This is done to show general clustering abilities of proposed method. Four samples of the input dataset are shown in Table 2. Fig.7a demonstrates the first 12 vectors partitioned by the first clustering iteration. Vectors $X_0$ and $X_2$ were partitioned incorrectly. To correct this result a feedback matrix was constructed: $T_{[500,4]} = \{t_{ij} \mid i \in [0, 500), j \in [0, 4)\}$ where

$$t_{ij} = \begin{cases} 1000, & i = 0, \quad j = 0; \\ 1000, & i = 2, \quad j = 1; \\ \quad 1, & otherwise. \end{cases}$$

Result of 100 epochs of clustering is shown on Figure 7b, just the wrong portioned vectors were moved, while earlier correctly partitioned vectors have saved their cluster labels. This shows how the user feedback gently corrects results without dramatic changes in partitioning itself.

**Table 2**
Vector samples with increased noise

| X | Coordinates | | | | Cluster |
|---|---|---|---|---|---|
| $X_0$ | **1.191519** | 0.6221088 | 0.4377278 | 0.7853585 | $C_0$ |
| $X_1$ | 0.7799758 | **1.272592** | 0.2764643 | 0.8018722 | $C_0$ |
| $X_2$ | 0.9581394 | 0.8759326 | **1.357817** | 0.5009951 | $C_1$ |
| $X_3$ | 0.6834629 | 0.7127021 | 0.3702508 | **1.561196** | $C_1$ |

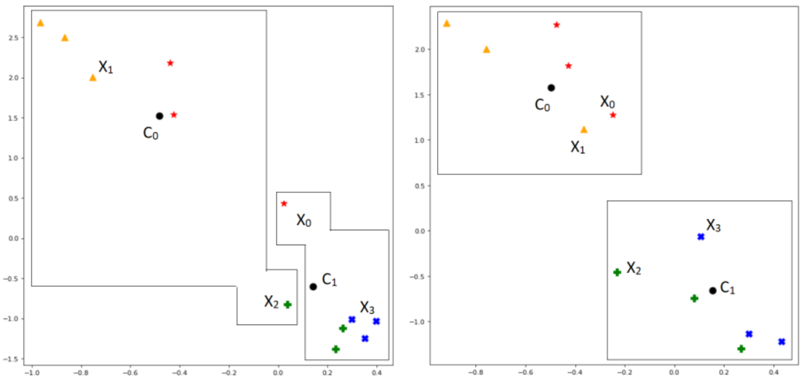


**Figure 7 (a,b)**: Clustering results of highly noised vectors. 7a – the first clustering stage results, 7b – results of iteration with a user feedback.

## 5.2. Experiments with synthetic Fishers' Irises dataset

Fishers' Irises dataset is a standard dataset for classification and clustering tasks [21]. It has petal and sepal length and width as features. This dataset has 3 classes 'setosa', 'versicolor', 'virginica'. Table 3 shows vector samples for each class. As long as Fishers' Irises dataset has 150 items only, in order to augment data slight random noise from uniform distribution U[0, 1/10] has been added to each vector to multiply this data set in 4 times. In total 600 items have been obtained.

**Table 2**
Vector samples with increased noise

| X | Coordinates | | | | Cluster |
|---|---|---|---|---|---|
| $X_0$ | 5.1191519 | 3.5622108 | 1.4437727 | 0.2785358 | $C_0$ (setosa) |
| $X_1$ | 7.0779975 | 3.2272592 | 4.7276464 | 1.4801872 | $C_1$ (versicolor) |
| $X_2$ | 6.3958139 | 3.3875932 | 6.0357817 | 2.5500995 | $C_2$ (virginica) |

Figure 8a demonstrates the result of the first clustering iteration. Cluster for the first class '*setosa*' has been detected without any errors. However other two classes have 13 and 16 incorrectly partitioned flowers respectively. Classes '*versicolor*' and '*virginica*' are actually close to each other and even classification algorithms could not distinguish them without any errors. However, let the user knows the real classes just for two items (in this example vectors with numbers 7 and 50) that have been partitioned incorrectly, so they should be swapped. Feedback matrix form the user provided: $T_{[600,3]} = \{t_{ij} \mid i \in [0, 600), j \in [0, 3)\}$, where

$$t_{ij} = \begin{cases} 1000, & i = 7, \quad j = 2; \\ 1000, & i = 50, \quad j = 1; \\ 1, & otherwise. \end{cases}$$

Result of 100 epochs of clustering is shown on Figure 8b (to project 3-dimensional vectors from latent feature space on a plane the t-SNE algorithm from python sklearn library has been used [25]). Incorrectly partitioned items 7 and 50 were moved to their classes, besides more errors were corrected by clustering method in unsupervised manner. After the 100-th epoch of the second iteration just 8 and 2 items partitioned incorrectly according to ground truth in 2-nd and 3-rd classes respectively. An accuracy of the resulting partitioning is 0.98(3). To compare, the average result of clustering performance state-of-the-art unsupervised clustering methods is 0.85 and for the state-of-the-art classification method is 0.971. Many clustering algorithms are not able to distinguish 2-nd and 3-rd classes at all [21].

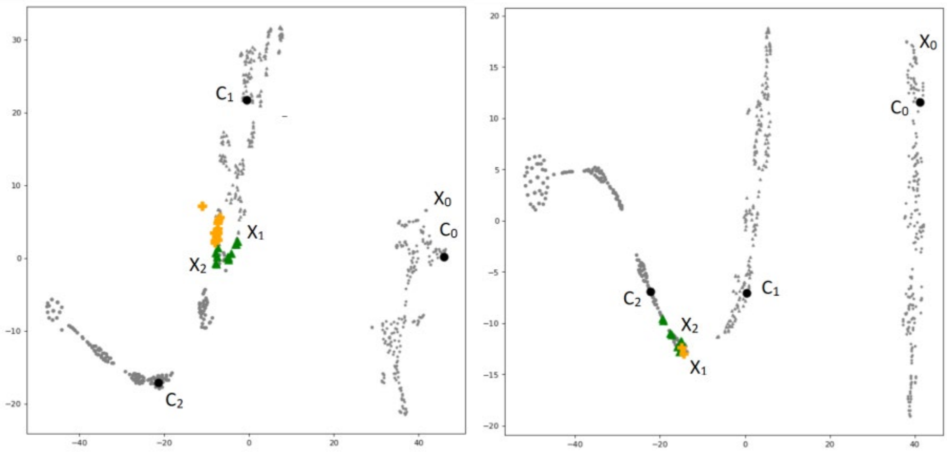


**Figure 8 (a,b)**: Clustering results for the Fishers' Irises dataset. Amount of incorrectly partitioned items has been decreased from 26 (8a) to 10(8b) due to feedback provided.

## 6. Conclusion

In this paper the recent methods of interactive clustering with user feedback are discussed. There are a lot of modern unsupervised clustering methods demonstrating as good results as the state-of-the-art approaches, but a few methods could process user feedback. To compensate the lack of interactive methods an approach to user feedback processing as interactive clustering technique with user feedback was proposed. This approach is based on neural network and the DEC algorithm was used as a base to demonstrate the core idea of proposed technique. Experiments have shown usability and effectiveness of the proposed approach. Although, lack of sensibility to "moving out of the cluster" operation was detected.

Future studies will be dedicated to investigation of possible alternatives to auxiliary target distribution and further exploring its properties. Also it is planned to add more types of feedback to the approach, for example, pair-wise feedback or structure of desired hierarchy.

## Acknowledgements

The report study was funded by RFBR and the government of Ulyanovsk region according to the research project Num. 18-47-00019.

## References


[1] Aljalbout E., Golkov V., Siddiqui Y., Strobel M., Cremers D., Clustering with Deep Learning: Taxonomy and New Methods, 2018. arXiv:1801.07648.

[2] Bae J., Helldin T., Riveiro M. Nowaczyk S., Bouguella M., Falkman G., Interactive Clustering: A Comprehensive Review, in: ACM Comput. Surv., 2020, Vol. 53, No. 1.

[3] Bagherjeiran A., Eick C. F., Chen C.-S., Vilalta R., Adaptive clustering: obtaining better clusters using feedback and past experience, in: Fifth IEEE International Conference on Data Mining (ICDM'05), Houston, TX, 2005.

[4] Balcan M.F., Blum A., Clustering with Interactive Feedback, in: Freund Y., Györfi L., Turán G., Zeugmann T. (eds) Algorithmic Learning Theory. Lecture Notes in Computer Science, vol 5254. Springer, Berlin, Heidelberg, 2008.

[5] Basu S., Banerjee A., Mooney R., Semi-supervised Clustering by Seeding., in: Proceedings of 19th International Conference on Machine Learning, 2002.

[6] Basu S., Davidson I., Wagstaff K., Constrained Clustering: Advances in Algorithms, Theory, and Applications, CRC Press, 2008.

[7] Basu S., Fisher D., Drucker S.M., Lu H., Assisting Users with Clustering Tasks by Combining Metric Learning and Classification, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010.

[8] Dasgupta S., Ng V., Which Clustering Do You Want? Inducing Your Ideal Clustering with Minimal Feedback, 2014. arXiv:1401.5389. URL: https://arxiv.org/abs/1401.5389.

[9] Demiriz A., Bennett K.P., Embrechts M.J., A Genetic Algorithm Approach for Semi-Supervised Clustering, in: International Journal of Smart Engineering System Design, 2002, vol. 4.

[10] Dizaji K.G., Herandi A., Deng C., Cai W., Huang H., Deep Clustering via Joint Convolutional Autoencoder Embedding and Relative Entropy Minimization, in: IEEE International Conference on Computer Vision (ICCV), Venice, 2017.

[11] Dudarin P., Pinkov A., Yarushkina N., Methodology and the algorithm for clustering economic analytics object, in: Automation of Control Processes. 2017. Vol. 47, № 1. P. 85-93.

[12] Dudarin P.V., Yarushkina N.G., An Approach to Fuzzy Hierarchical Clustering of Short Text Fragments Based on Fuzzy Graph Clustering, in: Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'17). IITI 2017. Advances in Intelligent Systems and Computing. 2018. vol 679. Springer. Cham.

[13] Dudarin P., Samokhvalov M., Yarushkina N., An Approach to Feature Space Construction from Clustering Feature Tree, in: Kuznetsov S., Osipov G., Stefanuk V. (eds) Artificial Intelligence. RCAI 2018. Communications in Computer and Information Science, vol 934. Springer, Cham, 2018.

[14] Dudarin P.V., Tronin V.G., Svyatov K.V., A Technique to Pre-trained Neural Network Language Model Customization to Software Development Domain, in: Kuznetsov S., Panov A. (eds) Artificial Intelligence. RCAI 2019. Communications in Computer and Information Science, vol 1093. Springer, Cham, 2019.

[15] Dudarin P.V., Tronin V.G., Svatov K.V., Belov V.A., Shakurov R.A., Labor intensity evaluation technique in software development process based on neural networks, in: Proceedings of the Second International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'19). Advances in Intelligent Systems and Computing, Springer. Cham, 2020.

[16] Fatehi K., Bozorgi A., Zahedi M.S., Asgarian E., Improving semi-supervised constrained k-means clustering method using user feedback, in: Journal of Computing and Security, 2014, Volume 1, Number 4.

[17] Greff K., van Steenkiste S., Schmidhuber J., Neural Expectation Maximization, in: Advances in Neural Information Processing Systems 30, 2017.

[18] Hastie T., Tibshirani R., Friedman J., The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition, in: Springer Series in Statistics book series, 2009.

[19] Hoffer E., Ailon N., Deep Metric Learning Using Triplet Network, in: Feragen A., Pelillo M., Loog M. (eds) Similarity-Based Pattern Recognition. Lecture Notes in Computer Science, vol 9370. Springer, Cham, 2015.

[20] Huang Y. Mixed-Iterative Clustering, PhD thesis at Language Technologies Institute School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, 2010.
[21] Leela V., Sakthipriya K., Manikandan R., Comparative Study of Clustering Techniques in Iris Data Sets, in: World Applied Sciences Journal 29 (Data Mining and Soft Computing Techniques), 2014.
[22] Li L., Kameoka H., Deep Clustering with Gated Convolutional Networks, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, 2018.
[23] Meier B.B., Elezi I., Amirian M., Dürr O., Stadelmann T. Learning Neural Models for End-to-End Clustering, in: Artificial Neural Networks in Pattern Recognition edited by Pancioni L., Schwenker F., Trentin E., Lecture Notes in Computer Science, vol 11081. Springer, Cham, 2018.
[24] Nebu C.M., Joseph S., Semi-supervised clustering with soft labels, in: International Conference on Control Communication & Computing India (ICCC), Trivandrum, 2015.
[25] Pedregosa F., Varoquaux G, Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay É., Scikit-learn: Machine Learning in Python, in: Journal of Machine Learning Research, 2011, vol. 12.
[26] Pedrycz W., Algorithms of fuzzy clustering with partial supervision, in: Pattern Recognition Letters, Volume 3, 1985.
[27] Shelekhova N.V., Rimareva L.V., Management of Technological Processes of Production of Alcohol Products with the Application of Information Technology, in: Storage and processing of agricultural raw materials, Moscow, 2017(3).
[28] Shelekhova N.V., Polyakov V.A., Serba E.M., Shelekhova T.M., Veselovskaya O.V., Skvortzova L.I., Information technology in the analytical quality control of alcoholic beverage, in: Food Industry, Moscow, 2018(12).
[29] Suresh T., Meena Abarna K.T., LSTM Model for Semantic clustering of user-generated content using AI Geared to wearable Device, in: Semanticscholar.org Corpus ID: 212585860, 2017. URL: https://www.semanticscholar.org/paper/LSTM-Model-for-Semantic-clustering-of-content-using-Suresh-Abarna/7b72349284b78803fe2581a041e5c7a19a081bdc
[30] Wang Z., Mi H., Ittycheriah A., Semi-supervised Clustering for Short Text via Deep Representation Learning, in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, 2016.
[31] Xie J., Girshick R., Farhadi A., Unsupervised deep embedding for clustering analysis, in: ICML'16: Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2002.
[32] Xu J., Xu B., Wang P., Zheng S., Tian G., Zhao J., Self-Taught Convolutional Neural Networks for Short Text Clustering, in: IEEE Neural Networks, 2017, Volume 88.
[33] Yang C., Shi X., Jie L., Han J., I Know You'll Be Back: Interpretable New User Clustering and Churn Prediction on a Mobile Social Application, in: the 24th ACM SIGKDD International Conference, 2018.
[34] Yang J., Parikh D., Batra D., Joint Unsupervised Learning of Deep Representations and Image Clusters, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016.
[35] Yang B., Fu X., Sidiropoulos N.D., Hong M., Towards K-means-friendly spaces: Simultaneous deep learning and clustering, in: Proceedings of the 34th International Conference on Machine Learning, Volume 70, 2017.