An Introductory Survey on Reframing in Clustering

Md. Geaur Rahman

ICube Laboratory, University of Strasbourg, France.
Email: grahman@unistra.fr

Abstract. Reframing is an essential task for improving the performance of machine learning and data mining algorithms in the areas where there are context changes between the source and target domains. A major assumption in many reframing algorithms is that the target domain has some labelled data. However, in many real-world applications, this assumption may not hold. For example, we sometimes have a clustering task in one domain of interest, but we only have sufficient source data in another domain of interest, where the latter data may be in a different feature space or follow a different data distribution. Moreover, both source and target data may be unlabelled. In such cases, reframing in clustering, if done successfully, would greatly improve the performance of clustering by avoiding much expensive data labeling efforts. In recent years, reframing in clustering has emerged as a new clustering framework to address this problem. In this paper, we present a review on the state-of-the-art reframing in clustering approaches, and to the best of our knowledge it has never been done in the literature. We give a definition of reframing in clustering. We also explore some potential future issues in this area of research.

Keywords: Reframing, Clustering, Classification, Data Mining, Machine Learning.

1 Introduction

Data mining and machine learning techniques such as clustering, classification and regression play a vital role for discovering interesting information from large datasets. In machine learning, generally models are built based on historical data (i.e. source data) and then deployed on target data where both the source and target data follow a common assumption. The techniques usually perform well when the source and target data are obtained from the same distribution and the same domain space [9, 18]. However, in many real-life applications, the source and target data are drawn from different environments, and the distributions of data may change from one environment to another environment. This phenomenon is known as Dataset Shift [1, 16]. If the distribution of the source and target data changes (that is, dataset shift occurred) then the performances of the techniques may suffer from a large amount of errors. A common solution to increase the performance of a technique is to rebuild (or retrain) a model using the data of target environment. However, in most cases,

the retraining of a model may not be feasible due to insufficient target data and time. Moreover, in many real-life applications, it is expensive or impossible to recollect sufficient data for the purpose of retraining a model. Therefore, it is wise to reduce the necessity of retraining of a model and to avoid the need of recollecting data from the target environment. In such situations, reframing between the domains of interest can be useful.

Many applications in the real world can be found where reframing can truly be beneficial. One such application is web document classification [8], in which the goal is to identify a category (from several predefined categories) of a given web document. As an example, consider a university website in the field of web document classification (see [4]) where the web pages may initially be categorised manually, and then build a web-page classifier by using the categorised web pages. For a classification task, it may not be possible to directly apply the web-page classifier learned on the university website to a newly created website since the data attributes or data distributions of the newly created website may be different. Moreover, in the newly created website, the amount of categorised web pages (i.e. labelled source data) may be insufficient to rebuild a web-page classifier. In such cases, it would be helpful if we could reframe the classification knowledge achieved from the university website into the newly created website.

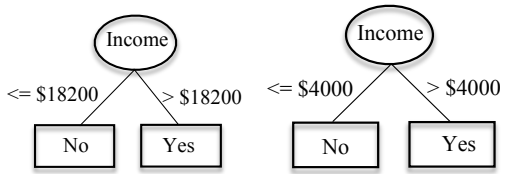
For reframing a number of techniques have been proposed recently [1, 11]. Most of the existing techniques perform a reframing (or transformation) on input attributes or output values. Besides, majority of the techniques require to have labelled data in the target environment. For example, an existing technique [1] handles the dataset shift between source and target data by reframing of continuous input attributes that have significant influences on the dataset shift. However, a user needs to have some labelled target data to apply the technique. So, it can not be applied to an unsupervised task such as web document classification, where the target data may generally be unlabelled. On the other hand, researchers have given less attention to handle dataset shift where target data are not labelled. Only a few number of techniques have been proposed in the area of unsupervised reframing. Therefore, the main objective of this survey paper is to provide a comprehensive overview of unsupervised reframing approaches developed in the field of machine learning and data mining.

The organization of the paper is as follows. In Section 2, we first give a general overview on the basic concept and the types of reframing. We then focus on some state-of-the-art reframing in clustering methods that are presented in Section 3. After that in Section 4, we discuss the potential future issues in the area of research on reframing in clustering, and finally in Section 5, we provide a concluding remark.

2 A Brief Overview on Reframing

Traditional data mining and machine learning algorithms make use of previously collected source data to build models and then make predictions on the target data using the models [1]. For example, a decision tree algorithm such as C4.5 [19] builds a classifier using source data in order to classify target records. Moreover, it predicts the label of a record if the record does not have any label.

Most of the traditional algorithms assume that the source and target data are drawn from the same distributions and the same domain space. However, it is natural that the distributions of the source and target data may be different if the data are collected from different locations. When the distributions of the source and target data change, the algorithms may not perform well or may produce misleading information. Let us consider a real life scenario where a decision function can be used to decide whether a person needs to pay tax or not. Figure 1 shows two decision functions on incomes for the people of two countries namely Australia and Bangladesh. An Australian person needs to pay tax if the income of him/her is higher than AU$18200, whereas in Bangladesh a person needs to pay tax if the income is higher than AU$4000 ($\approx$ BDT250000). Now if a model is built on Australian tax payment data and deployed in Bangladesh then we can see majority of the people of Bangladesh do not need to pay tax. On the other hand, if a model is built on Bangladeshi tax payment data and deployed in Australia then we can see majority of Australian need to pay tax. Besides, it can be seen from the figure that an accurate decision function can be achieved if a model is retrained on the deployment data. However, in many real world applications the retraining of a model may not be possible due insufficient target data [1].



(a) Australia (b) Bangladesh

Fig. 1. Simple decision functions to decide whether a person needs to pay tax or not.

Reframing, in contrast, is an alternative approach which deals with context changes between source and target environments [1,16]. The common context changes include dataset shift, task change and representation change [1, 16]. Let M be the model built from the source data, θ be the context such as dataset shift, X be the target data and D_a be the additional data (labelled or unlabelled) that may be available during deployment. If Y is the expected output then reframing can be defined as a function $R(.)$ as follows [14].

$Y \leftarrow R(X, M, \theta, D_a)$

The reframing approaches of the existing techniques can be categorised into output reframing, input reframing and structural reframing. The reframing approaches can further be categorised into supervised reframing and unsupervised reframing. The techniques belonging to the supervised reframing category generally handle the cases where both source and target data are labelled, unlike the techniques belonging to the unsupervised reframing category deal with the cases where both source and target data are unlabelled. Reframing in clustering falls in the category of unsupervised reframing.

3 Reframing in Clustering

Before we discuss the methods of reframing in clustering, we first provide an overview about clustering itself. Clustering is a process of grouping records of a dataset into a number of groups called clusters in such a way that the records belonging to a cluster are similar to each other and the records belonging to different clusters are dissimilar to each other [9, 12, 20]. Typically, a clustering algorithm requires the user to enter the number of clusters k, which is greater than or equal to 2 [10]. Each cluster is represented by a center and therefore, there are k centers ($\{V_1, V_2, \ldots, V_k\}$) for k clusters ($\{C_1, C_2, \ldots, C_k\}$). Clustering has a wide range of real-world applications including medical data analysis, business and marketing data analysis, and social network data analysis [9, 12].

Many clustering algorithms have been proposed for grouping the records of a dataset [9, 10, 12]. A commonly used clustering algorithm is k-Means [9, 12] which initially selects k records randomly from the dataset as the centers for k clusters. A record r_i of the dataset is assigned to a cluster C_k if V_k is the cluster center with minimal distance to the record r_i. Once all records are assigned to the clusters, in the next iteration the technique calculates the cluster centers again based on the records of each cluster. After that all records are reorganised such that a record r_i is assigned to the cluster C_k the center V_k of which has the minimum distance with r_i. The process of reorganising records and finding new centers continues recursively until a termination conditions is satisfied. Generally, the number of iterations and a minimum difference between the centers are considered as termination condition.

While the simplicity is an advantage of k-Means, the technique requires the user to provide a value for k [12]. Additionally, the performance of the technique depends on the size of a data set. The technique may not perform well in a very small dataset. It is reported that a clustering algorithm applied on a small dataset having less than 500 records may group the records incorrectly [15, 17].

Although a clustering algorithm does not produce a model (which is built by a classification or a regression algorithm) from the source data, it can produce a prototype which can be deployed on the target data in order to group them [21]. For example, a center of

a cluster of the k-Means clustering algorithm can be the representative (or prototype) of the cluster. Therefore, the target data can be grouped based on the k representatives of the k-Means clustering algorithm.

Now consider a scenario where we have source data from one environment and target data from another environment, and both source and target data are unlabelled. Additionally, only a few number of data are available in the target environment. Can we design a method to group the unlabelled insufficient target data correctly?

Since both source and target data are unlabelled we can not build models through applying a classification or regression algorithm on the source data. In addition, due to insufficient target data, a clustering algorithm may not be able to group the data correctly. Moreover, since the training and test data are drawn from different environment, the prototypes built (by a clustering algorithm) on the source data can not directly be applied on the target data. Therefore, a reframing of the prototypes would be desirable to group the target data correctly. This phenomenon of reframing of prototypes can be referred to as reframing in clustering.

A definition of reframing in clustering can be given as follows. *Given a source domain D_S with a clustering task C_S, a target domain D_T and a corresponding clustering task C_T , reframing in clustering aims to improve the quality of clustering of the deployment clustering function $R_C(.)$ in D_T using the knowledge in D_S and C_S, where $C_S \neq C_T$ and the label of data of the source domain Y_S and the label of data of the target domain Y_T are not observable*.

Based on the definition of the reframing in clustering, no labeled data are observable in the source and target domains in training. So far, a little research work has been done on this category. However, research on reframing in clustering has attracted more and more attention recently in different names: self-taught clustering [5], online clustering [2], incremental clustering [7], and mean shift clustering [3]. Among these, a closely related technique to reframing in clustering is the self-taught clustering [5], which makes use of the common features between the source and target data in order to group the target data. We now discuss some state-of-the-art reframing in clustering techniques.

Self-taught clustering (STC) [5] aims to cluster a small collection of unlabelled data in the target domain with the help of a large amount of unlabelled data in the source domain. The basic idea of STC is to learn a common feature space between source and target domains, which helps in clustering in the target domain. Let X_S and X_T be the source and target domain data, respectively and Z be the common feature space between X_S and X_T. Moreover, consider that there exist three clustering functions $C_{XT} : X_T \rightarrow \tilde{X}_T$, $C_{XS} : X_S \rightarrow \tilde{X}_S$ and $C_Z : Z \rightarrow \tilde{Z}$, where $\tilde{X}_T$, $\tilde{X}_S$ and $\tilde{Z}$ are corresponding clusters of X_T, X_S and Z, respectively. Now if $I(.,.)$ is the mutual information between two random variables, the objective function $R(\tilde{X}_T, \tilde{X}_S, \tilde{Z})$ of STC is given as follows [5] .

$$R(\tilde{X}_T, \tilde{X}_S, \tilde{Z}) = I(X_T, Z) - I(\tilde{X}_T, \tilde{Z}) + \lambda \left[I(X_S, Z) - I(\tilde{X}_S, \tilde{Z}) \right] \quad (1)$$

where λ is a user-defined parameter to balance the influence between the target data and source data. In Equation(1), we see that the two different co-clustering functions $I(X_T, Z) - I(\tilde{X}_T, \tilde{Z})$ and $I(X_S, Z) - I(\tilde{X}_S, \tilde{Z})$ share the same clustering function $\tilde{Z}$, which is acting as the bridge in reframing the knowledge between the source and target data. The technique finally finds the clusters $\tilde{X}_T$ of the target data X_T by solving the optimization problem [5].

$$\underset{\tilde{X}_T, \tilde{X}_S, \tilde{Z}}{arg\,min} \quad R(\tilde{X}_T, \tilde{X}_S, \tilde{Z}) \quad (2)$$

Experimental results indicate that STC performs better than several existing clustering techniques. However, the techniques requires a user to provide several inputs including the common feature space Z and the value of the trade-off parameter λ, which could be difficult for a user to know in advance for a real-world application.

While STC assumes that both source data and target data are present during clustering, an existing technique called COBWEB [7] that considers a stream of records which are acquired one at a time. This approach is known as incremental clustering. The technique clusters the records by following three steps. First, COBWEB assigns the first record into a cluster. Second, the technique considers the next record and assigns it either to one of the existing clusters or to a new cluster. The procedure of assignment is done based on some criterion. For instance, the distance between the new record and the centers of the existing clusters determines the cluster in which the new record falls. Moreover, the technique uses a heuristic evaluation measure to ensure the quality of the clusters. A new record is assigned to a cluster without affecting the existing clusters significantly. Third, the technique repeats the second step till all the records are clustered. In terms of time and space complexity, COBWEB is found to be less expensive since it does not require to store all the records (belonging to the clusters) in the memory [13], and therefore, can be successfully used in engineering applications.

Unlike STC and COBWEB, an online clustering technique [2] first finds the clusters centers called prototypes based on source data and then uses the prototypes to group the target data. The technique allows the prototypes to learn online. It then iteratively updates the prototypes as follows.

$$V_k^{new} = V_k + \zeta(x_i - V_k) \quad (3)$$

where $x_i \in X_T$ is the target data, V_k in the prototype of the k-th cluster on the source data, V_k^{new} is the estimated prototype of the k-th cluster on the target data, and ζ is a learning rate usually set to be a small positive number (e.g., 0.05). The learning rate can also gradually decrease during the learning process.

Another algorithm called mean shift clustering [3] groups the records into clusters without any user input such as the number of clusters and the shape of the clusters. The basic idea behind mean shift clustering is to consider the records in the d-dimensional feature space as an empirical probability density function where dense regions in the feature space correspond to the local maxima or modes of the underlying distribution. For each record in the feature space, one performs a gradient ascent procedure on the local estimated density until convergence. The potential cluster centers of this procedure represent the modes of the distribution. Furthermore, the records associated with the same cluster center are considered members of the same cluster.

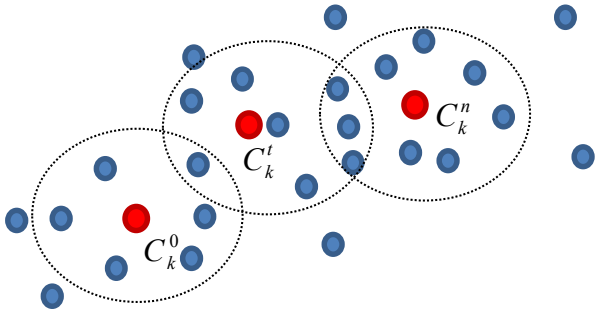


Fig. 2. An illustration of mean shift clustering [6].

An illustration of the procedure of mean shift clustering is presented in Fig. 2 [6]. In the figure, we can see that the clustering is started with the initial cluster C_k^0, where the superscript denotes the mean shift iteration and subscript denotes the k-th cluster. The blue and red circles (small) denote the input records and successive window centers, respectively, and the dotted circles denote the density estimation windows. In the next iteration, it then run the mean shift procedure to find the potential cluster center of the density function. After the t-th iteration, the technique find the cluster C_k^t. The iterative procedure is finished when the technique find the final cluster C_k^n.

Recently, a technique called Transferred Discriminative Analysis (TDA) [22] makes use of the discriminative analysis for clustering unlabelled target data. TDA first generates pseudo-class labels for the target unlabelled data by applying a clustering algorithm. A dimensionality reduction method is then applied to the target data and labelled source data to reduce the dimensions. The technique runs these two steps iteratively to find the best clusters for the target data.

Although it is reported that the existing reframing in clustering approaches perform well over the baseline clustering algorithms, however, the existing techniques have limitations and have scope for further improvement.

4 Potential Future Issues

In this section we discuss some potential future issues in the area of research on reframing in clustering.

Real-life datasets: To the best our knowledge, most of the existing techniques have been evaluated either on synthetic datasets or on the datasets that are not relevant in the context of reframing. The use of real-life datasets to evaluate the techniques could be a better motivation for improvements.

Automatic tuning of parameters: Most of the existing techniques require the user to provide input which could be difficult for a user to know in advance in real-world applications. For example, an existing technique called STC [5] requires a user-defined value for the trade-off parameter λ that balances the influence between the target data and source data. Similarly, the mean shift clustering [3] technique requires the user to provide the radius of the circle that represents the density region in the dataset. On the other hand, online clustering [2] requires the user to give the learning rate ζ for the quick convergence of the learning process.

Feature representation: For achieving a good clustering result on the target data, it is important to have better feature representation between the source and target data. Moreover, sometimes it is required by a user to provide the common feature. For example, STC [5] requires the user to provide the common feature space Z. However, it could be difficult for a user to know Z in advance in real-world applications. So, it would be useful if a technique can automatically find the best common features between the source and target data.

5 Conclusion

In machine learning and data mining applications, it is natural to have source data from one environment and target data from another environment, and both source and target data are unlabelled. Moreover, in case of insufficient data in the target environment, it is required to reframe the knowledge achieved from the source environment into the target environment. This process is known as reframing. In this paper, we present a survey on the state-of-the-art clustering algorithms that can reframe prototypes from a source environment to a target environment. However, most of the existing technique have limitations. For example, an existing technique called STC [5] requires a user to provide several inputs including the common feature space Z and the value of the trade-off parameter λ, which could be difficult for a user to know in advance for a real-world application. Furthermore, the existing techniques have been evaluated on small-scale of applications. In addition, they were not evaluated on any real-world applications. Therefore, existing techniques have room for fur-

ther improvement. In the future, we aim to develop a new algorithm by addressing issues and evaluate the technique on a real world application.

Acknowledgements

I thank Dr. Nicolas Lachiche of ICube, University of Strasbourg, France, for his helpful comments to improve the paper. I also thank Dr. Pierre Ganarski of ICube, University of Strasbourg, France, for his assistance with the online clustering and incremental clustering. I would also like to express deep gratitude to the anonymous reviewers of this paper. Their useful comments have played a significant role in improving the quality of this work.
This work was supported by the REFRAME project granted by the European Coordinated Research on Long-term Challenges in Information and Communication Sciences & Technologies ERA-Net (CHIST-ERA).

References

1. Ahmed, C.F., Lachiche, N., Charnay, C., Braud, A.: Reframing continuous input attributes. In: Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence. pp. 31–38. IEEE (2014)
2. Barbakh, W., Fyfe, C.: Online clustering algorithms. International Journal of Neural Systems 18(03), 185–194 (2008)
3. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(8), 790–799 (1995)
4. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: Proceedings of the 24th international conference on Machine learning. pp. 193–200. ACM (2007)
5. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Self-taught clustering. In: Proceedings of the 25th international conference on Machine learning. pp. 200–207. ACM (2008)
6. Derpanis, K.G.: Mean shift clustering. Lecture Notes. http://www. cse. yorku. ca/~ kosta/CompVis_Notes/mean_shift. pdf (2005)
7. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Machine learning 2(2), 139–172 (1987)
8. Fung, G.P.C., Yu, J.X., Lu, H., Yu, P.S.: Text classification without negative examples revisit. Knowledge and Data Engineering, IEEE Transactions on 18(1), 6–20 (2006)
9. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann (2006)
10. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Applied statistics pp. 100–108 (1979)
11. Hernández-Orallo, J., Flach, P., Ferri, C.: A unified view of performance metrics: Translating threshold choice into expected classification loss. The Journal of Machine Learning Research 13(1), 2813–2869 (2012)
12. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc. (1988)

13. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM computing surveys (CSUR) 31(3), 264–323 (1999)
14. Jose, H.O., Ricardo, B.P., Kull, M., Flach, P., Chowdhury, F.A., Lachiche, N., Martiynez-Uso, A.: Reframing in context: A methodology for model reuse in machine learning. AI Communications (submitted) (2015)
15. Marston, L., Peacock, J.L., Yu, K., Brocklehurst, P., Calvert, S.A., Greenough, A., Marlow, N.: Comparing methods of analysing datasets with small clusters: case studies using four paediatric datasets. Paediatric and perinatal epidemiology 23(4), 380–392 (2009)
16. Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition 45(1), 521–530 (2012)
17. Nikitinsky, N., Sokolova, T., Pshehotskaya, E.: Practical issues of clustering relatively small text data sets for business purposes. In: The International Conference on Digital Security and Forensics (DigitalSec2014). pp. 15–22. The Society of Digital Information and Wireless Communication (2014)
18. Pan, S.J., Yang, Q.: A survey on transfer learning. Knowledge and Data Engineering, IEEE Transactions on 22(10), 1345–1359 (2010)
19. Quinlan, J.R.: Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research 4, 77–90 (1996)
20. Rahman, M.G., Islam, M.Z.: Missing value imputation using a fuzzy clustering-based em approach. Knowledge and Information Systems pp. 1–34 (2015)
21. Tan, P.N., Steinbach, M., Kumar, V.: Data mining cluster analysis: Basic concepts and algorithms (2013)
22. Wang, Z., Song, Y., Zhang, C.: Transferred dimensionality reduction. In: Machine learning and knowledge discovery in databases, pp. 550–565. Springer (2008)