Human-Centered Interactive Clustering for Data Analysis

Jason Chuang[*]
Allen Institute for Artificial Intelligence
Seattle, WA
jason@chuang.ca

Daniel J. Hsu
Computer Science, Columbia University
New York, NY
djhsu@cs.columbia.edu

Abstract

Clustering is a critical component of many data analysis tasks, but is exceedingly difficult to fully automate. To better incorporate domain knowledge, researchers in machine learning, human-computer interaction, visualization, and statistics have independently introduced various computational tools to engage users through interactive clustering. In this work-in-progress paper, we present a cross-disciplinary literature survey, and find that existing techniques often do not meet the needs of real-world data analysis. Semi-supervised machine learning algorithms often impose prohibitive user interaction costs or fail to account for external analysis requirements. Human-centered approaches and user interface designs often fall short because of their insufficient statistical modeling capabilities. Drawing on effective approaches from each field, we identify five characteristics necessary to support effective *human-in-the-loop* interactive clustering: ***iterative***, ***multi-objective***, ***local updates*** *that can operate on* ***any initial clustering*** *and a* ***dynamic set of features***. We outline key aspects of our technique currently under development, and share our initial evidence suggesting that all five design considerations can be incorporated into a single algorithm. We plan to demonstrate our technique on three data analysis tasks: feature engineering for classification, exploratory analysis of biomedical data, and multi-document summarization.

1 Introduction

Clustering is a critical component of many data analysis tasks — document exploration in information retrieval systems such as Scatter/Gather [8], community detection in network analysis [19], data preparation for sentiment classification [21], scientific investigations in genetic research [13], or investigative analysis in tools such as JIGSAW [25]. Across these applications, clustering is a part of the larger data analysis workflow. Generating good clusters requires both knowledge about the analysis domain as well as addressing task-specific considerations, making the clustering process exceedingly difficult to fully automate.

In this paper, we first examine interactive clustering approaches in four domains — machine learning, human-computer interaction, visualization, and statistics — where the researchers leverage statistical modeling, user interface design, human cognition, and mathematical properties to incorporate user feedback. We find that existing techniques are often unsatisfactory. Combining best practices from these disciplines and drawing on our own experiences, we then propose five algorithmic characteristics that are necessary to support effective *human-in-the-loop* interactive clustering. We outline key aspects of our technique currently under development, and discuss our plans to demonstrate it on three analysis tasks.

[*]Research done while at Computer Science and Engineering, University of Washington

We would like to take the opportunity at this workshop to seek feedback from other participants on analysis needs that they encounter in their respective domains, and how our ongoing developments may better connect with other statistical modeling, user interface design, or data science research.

2 Background

We consider clustering as the computational problem of partitioning a collection of input data instances into groups, as to maximize the similarity within each group. Users may use such clustering output to support data analysis. Interactive clustering is the user-driven process of refining and optimizing the clusters for subsequent analyses. Distinct from signal processing where a compact representation is often designed to maximize information transmission between machines, here we focus on clustering for human interpretation and comprehension.

K-means clustering [20] is a widely applied and popular technique. Despite its widespread use, it falls under a larger class of automated clustering algorithms [29] and unsupervised learning approaches [4] that do not account for domain knowledge and can deviate from user expectations. When deviations occur, there is typically little recourse available to the users to refine clustering results for their analysis — which limits the utility of these fully automated techniques.

2.1 Constrained Clustering Approaches

In machine learning, efforts to incorporate domain knowledge via semi-supervised learning include constrained k-means clustering approaches by Wagstaff et al. [27], where users express domain-specific information through the specification of instance-level constraints, *must-links* and *cannot-links*, that enforce the association and disassociation between pairs of data points. Klein et al. [17] describe space-level constraints that apply to local neighborhoods surrounding a data point, rather than just the individual data points themselves. A comprehensive review of all constrained approaches to clustering is beyond the scope of this paper, but can be found in the survey by Davidson et al. [9] or the book by Basu et al. [2].

However, recent work finds both modeling and usability issues with these constrained clustering approaches. Wagstaff et al. [28] note a large variation in clustering results when a large number of constraints are present in a system. Davidson et al. [11] find that some constraint sets may even lead to decreased accuracy.

More problematic, from the perspective of developing user-facing analysis tools, are the usability issues. In our own work [6] designing interactive visualizations involving instance-level constraints, we observe that users become exhausted after specifying a large number of pairwise linkages between individual data instances. Davidson et al. [10] criticize these techniques for their lack of a guarantee to preserve existing relationships. Since the algorithm extends the effects of the constraints globally, adding a new constraint can potentially cause significant changes to every single data instance, including ones deemed by the users to be only marginally related to the newly-specified constraint. Therefore, users become frustrated when, after adding a new constraint, the algorithm modifies or removes unrelated clusters. We find [6] that such **global updates** significantly hamper analysis, because users are forced re-inspect every cluster for potential changes after each computer-assisted update.

The design of instance-level constrained clustering may have the good intention of algorithmically capturing domain knowledge from users. In practice, incorporating such algorithms into a user interface creates significant gulf of execution (i.e., specification of a large number of constraints) and substantial gulf of evaluation (i.e., assessing the scope of a cluster refinement in response to a constraint), and violates well-established user interface design principles [22].

2.2 Human-Centered Design

Research in human-computer interaction tends to focus on supporting effective clustering operations by leveraging existing algorithms, rather than developing new ones. For example, Basu et al. [3, 12] introduces the iCluster system to help users sort images into clusters. They accelerate cluster assignment through the use of classifiers to recommend similar images that should go into a known cluster. Seo et al. [24] focus on helping users interactively curate clusters through more semantically mean-

ingful operations such splitting and merging clusters. Their interface is built on top of hierarchical agglomerative clustering output.

However, these studies typically address only piecewise and unrelated components of interactive clustering. Invoking classifiers as a subroutine accelerates cluster assignment, but the technique does not apply to operations such as cluster splits. Hierarchical clustering provides UI designers with the caability to show splits and merges, but the technique is unsupervised and does not allow users to override the machine should it make poor split decisions.

Though insights from these studies are valuable, we believe that realizing truly effective interactive clustering requires more than just user interface design guidelines. We need a fundamentally new clustering algorithm designed with both learning capability as well as usability at its core.

2.3 Interpretability vs. Statistical Expressiveness

Information visualization research typically approaches interactive clustering through the use of techniques that rescale the data space. Lee et al. [18] introduces iVisClustering based on latent Dirichlet allocation [4]. While strictly-speaking only a visual (and not statistical) clustering tool, iPCA by Jeong et al. [16] provides users with the ability to rescale the principal component axes when inspecting data on a two-dimensional plane.

Rescaling techniques have a natural correspondence to 2-dimensional visual displays. While users typically respond favorably to these intuitive tools, such systems typically do not have sufficient statistical power to capture fine-grained domain-specific information or retain them over the course of a long analysis. While Endert et al. [14] studies the semantics of visual clustering, such spatial representations do not provide direct support for high-level clustering operations. These visual clusterings do not produce quantitative output that are typically more suitable for subsequent analyses.

On the opposite end of the spectrum, to maximize statistical expressiveness, Grimmer et al. [15] proposes a framework of computer-assisted clustering by exploring a large number of clusterings, representing all potential partitions of the input data, to help users recognize useful or insightful conceptualizations. Pimentel [23] introduces a meta visualization of the immense space of all potential clusterings.

We argue that an appropriate interactive clustering algorithm must have sufficient learning power, but only when its statistical modeling capabilities can be mapped to effective user interactions and interpretable visual presentations.

2.4 Local Updates for Interactive Clustering

We highlight one additional piece of recent work by Awasthi et al. [1] that is novel in its ability to enable **local updates**.

The authors describes a local algorithm for interactive clustering. At initialization, this algorithm constructs an agglomerative hierarchical clustering of the input data. When a user requests cluster split or merge operations, the algorithm identifies potential data instances that belong to the newly formed clusters. By limiting the effects of the inference step to a subtree within the hierarchical clustering structure, the authors demonstrate that they can localize their inference algorithm and therefore cluster reassignment.

Since all cluster reassignments are based on the output of an agglomerative hierarchical clustering algorithm, this technique does not support multi-objective clustering. In comparison to aforementioned work by Seo et al. [24], this technique can be viewed as a means to allow users to partially accept, rather than completely adhere to, the output of hierarchical clustering.

2.5 Data Analysis Workflow

As clustering is typically only an interim step within a larger data analysis workflow, we point out two additional factors that many current existing techniques and tools fail to take into account.

First, in our own experiences, many domain-specific datasets come with sufficient metadata that users can typically generate initial clusterings that are better than random cluster assignments, a

common initial condition for existing techniques. We argue that a properly-designed interactive clustering algorithm must be able to accept any user-supplied clustering as its input.

Second, again drawing on our own experiences but echoed in numerous conversations, cluster generation — assigning a cluster label to every data instance — is typically only a means to support analysis and rarely a goal in itself. A common use of clustering output is to enable predictive analysis — examining how a set of input factors affect observable output states and measuring any population effect that arises when we partition the input data into various clusters. In this regard, we argue that an effective interactive clustering algorithm must provide explanatory power, including on how input features (used to compute item-to-item similarity measures and construct the clusters) contribute to the final clustering. The algorithm should also provide the capability for users to dynamically refine the set of input features, so that they may directly inspect any effect on the clusters and downstream observable states.

3 Interactive Clustering for Data Analysis

From our assessment of best practices in machine learning, human-computer interaction, visualization, statistics, and data analysis practices, we distill five design considerations for our interactive clustering algorithm. We outline our algorithm, which is still under active development, and discuss how it can contribute to two ongoing visualization projects in feature engineering and biomedical research and one potential text analysis application.

3.1 Design Considerations

We argue that an effective interactive clustering algorithm should be **iterative** and **multi-objective**, support **local updates**, and can operate on **any initial clustering** and a **dynamic set of features**.

Iterative The algorithm must allow users to iterative refine the partition of the input data, in a direct manner through semantically meaningful operations. The converse should also hold. The algorithm should not modify cluster membership without direct intervention from the users. (*Reducing the gulf of execution*)

Multi-Objective The algorithm must have sufficient statistical modeling capabilities to learn the different potential user-defined partitioning of the data, and accelerate the clustering process. (*Sufficient statistical expressiveness*)

Local Updates The algorithm must allow users to localize inference operations, so that any computer-assisted clustering operations applies only to relevant data. (*Reducing the gulf of evaluation*)

Initial Clustering The algorithm must be able to accept any user-supplied clustering as its input.

Dynamic Feature Set The algorithm should allow users to assess the contribution of input features on the output clusters and dynamically modify the input feature set while maintaining the current cluster memberships.

3.2 Our Algorithm

We outline key aspects of our technique, and share our initial evidence suggesting that all design considerations specified above can be actualized in a single algorithm.

At initialization and the during cluster refinement process, we internally maintain multiple hierarchical clusterings of the input data, based on bootstrap-sampled features. Similar to the work by Awasthi et al. [1], such clusterings allow our algorithm to perform local updates. As our algorithm only performs cluster reassignment in response to user operations, it can accept any arbitrary clustering as input and meets our iterative design criteria. By maintaining multiple trees and continuously updating them to reflect the most likely user-defined partition, our algorithm is both multi-objective and can adapt to any dynamic feature sets. We are currently exploring the appropriate algorithm, visual, and interaction designs to enable effective active learning.

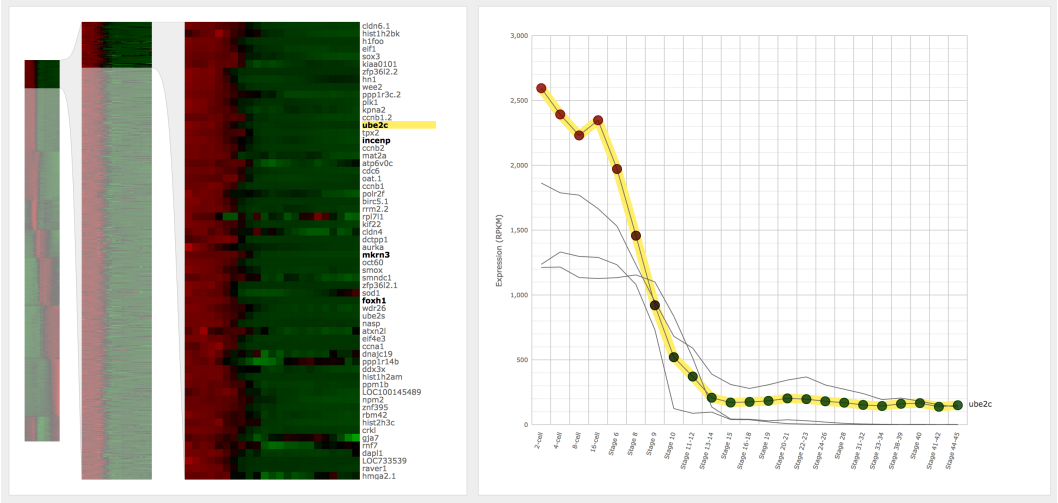

Figure 1: Our visualization for exploring clustered frog gene data, as described in the paper by Tan et al. [26], available online at `http://jason.chuang.ca/research/xenopus`

4 Applications

We discuss two ongoing visualization projects and a potential text analysis application, that can immediately benefit from the development of our interactive clustering algorithm. We welcome feedback on other potential applications or user interface designs that may help inform our design considerations, algorithm development, or evaluations.

4.1 Feature Engineering

We previously applied visual analysis to help machine learning researchers design state-of-the-art sentiment classifiers [7]. Interactive visualizations contribute to many parts of the project, from initial data exploration through model design and development. However, we find a general lack of interactive algorithms to support the design of visualizations for feature engineering. When tuning a classifier, clustering misclassified instances into groups helps model builders recognize systematic errors. Model builders can then inspect features common to the observed misclassifications, and respond with complementary features. We plan to apply our interactive clustering algorithm to support human-centered boosting for feature engineering.

4.2 Exploratory Analysis in Biomedical Research

We previously created interactive visualizations (Figure 1) to help genetics researchers explore clusterings of frog gene data [26], and are currently collaborating with bioinformatics researchers to visualize clinical patient data. We plan to incorporate our interactive clustering algorithm into these visualizations to support exploratory analysis.

4.3 Multi-Scale Text Summarization

An open research question in natural language processing is identifying an effective means to help users explore a large corpus of documents. Christensen et al. [5] introduces a technique for generating multi-scale text summarization, that can provide users with a summary of multiple documents aggregated at multiple levels of details. Currently, their algorithm pre-processes the input corpus by partitioning all documents into clusters via hierarchical agglomerative clustering. While their approach can generate good summarization for a given cluster, the quality of the generated summaries ultimately depends on the clustering quality. Users are limited to exploring the corpus only along the predefined clusters. We believe our interactive clustering algorithm can enable novel text summarization and exploratory text analysis tools.

References

[1] Pranjal Awasthi, Nina Balcan, and Konstantin Voevodski. Local algorithms for interactive clustering. 2014.

[2] Sugato Basu, Ian Davidson, and Kirk Wagstaff. *Clustering with Constraints: Algorithms, Applications and Theory*. Chapman Hall/CRC Press, 2008.

[3] Sumit Basu, Danyel Fisher, Steven M. Drucker, and Hao Lu. Assisting users with clustering tasks by combining metric learning and classification. In *AAAI*, 2010.

[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(1):993–1022, 2003.

[5] Janara Christensen, Stephen Soderland, Gagan Bansal, and Mausam. Hierarchical summarization: Scaling up multi-document summarization. In *ACL*, 2014.

[6] Jason Chuang, Yuening Hu, Ashley Jin, John D Wilkerson, Daniel A McFarland, Christopher D Manning, and Jeffrey Heer. Document exploration with topic modeling: Designing interactive visualizations to support effective analysis workflows. In *NIPS Workshop on Topic Models: Computation, Application, and Evaluation*, 2013.

[7] Jason Chuang and Richard Socher. Interactive visualizations for deep learning. In *VAST Workshop on Visualization for Predictive Analytics*, 2014.

[8] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *SIGIR*, pages 318–329, 1992.

[9] Ian Davidson and Sugato Basu. A survey of clustering with instance-level constraints. *ACM Transactions on Knowledge Discovery from Data*, (1):1–40, 2007.

[10] Ian Davidson, S. S. Ravi, and Martin Ester. Efficient incremental constrained clustering. In *KDD*, pages 240–249, 2007.

[11] Ian Davidson, Kirk Wagstaff, and Sugato Basu. Measuring constraint-set utility for partitional clustering algorithms. In *ECML/PKDD*, 2006.

[12] Steven M. Drucker, Danyel Fisher, and Sumit Basu. Helping users sort faster with adaptive machine learning recommendations. In *Interact 2011*, 2011.

[13] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, (25):14863–14868, 1998.

[14] Alex Endert, Seth Fox, Dipayan Maiti, and Chris North. The semantics of clustering: analysis of user-generated spatializations of text documents. In *AVI*, pages 555–562, 2012.

[15] Justin Grimmer and Gary King. General purpose computer-assisted clustering and conceptualization. *Proceedings of the National Academy of Sciences*, 108(7):2643–2650, 2011.

[16] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, and Remco Chang. iPCA: An interactive system for pca-based visual analytics. In *EuroVis*, pages 767–774, 2009.

[17] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML*, pages 307–314, 2002.

[18] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John T. Stasko, and Haesun Park. ivisclustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, (3):1155–1164, 2012.

[19] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *WWW*, pages 631–640, 2010.

[20] James MacQueen. Some methods for classification and analysis of multivariate observations, 1967.

[21] Wanting Mao, Lu Xiao, and Robert Mercer. *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, chapter The Use of Text Similarity and Sentiment Analysis to Examine Rationales in the Large-Scale Online Deliberations, pages 147–153. Association for Computational Linguistics, 2014.

[22] D.A. Norman. *The Psychology of Everyday Things*. Basic Books, 1988.

[23] Samuel D. Pimentel. Choosing a clustering: an a posteriori method for social networks. *Journal of Social Structure*, 15, 2014.

[24] Jinwook Seo and Ben Shneiderman. Interactively exploring hierarchical clustering results. *Computer*, 35(7):80–86, 2002.

[25] John Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: Supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008.

[26] Meng How Tan, Kin Fai Au, Arielle Yablonovitch, Andrea Wills, Jason Chuang, Julie Baker, Wing Hung Wong, and Jin Billy Li. Rna sequencing reveals diverse and dynamic repertoire of the xenopus tropicalis transcriptome over development. *Genome Research*, (1):201–216, 2013.

[27] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *ICML*, pages 577–584, 2001.

[28] Kiri L. Wagstaff, Sugato Basu, and Ian Davidson. When is constrained clustering beneficial, and why. In *in AAAI*, 2006.

[29] Rui Xu and II Wunsch, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.