The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Interactive Machine Learning at Scale with CHISSL

Dustin Arendt,[1] Emily Grace,[2] Svitlana Volkova[2]
[1]Visual Analytics, [2]Data Science and Analytics, Pacific Northwest National Laboratory
902 Battelle Blvd, Richland, WA 99354

Abstract

We demonstrate CHISSL a scalable client-server system for real-time interactive machine learning. Our system is capable of incorporating user feedback incrementally and immediately without a pre-defined prediction task. Computation is partitioned between a lightweight web-client and a heavyweight server. The server relies on representation learning and off-the-shelf agglomerative clustering to find a dendrogram, which we use to quickly approximate distances in the representation space. The client, using only this dendrogram, incorporates user feedback via transduction. Distances and predictions for each unlabeled instance are updated incrementally and deterministically, with $O(n)$ space and time complexity. Our algorithm is implemented in a functional prototype, designed to be easy to use by non-experts. The prototype organizes the large amounts of data into recommendations. This allows the user to interact with actual instances by dragging and dropping to provide feedback in an intuitive manner. We applied CHISSL to several domains including cyber, social media, and geo-temporal analysis.

Motivation and Background

Often analysts are handed huge piles of data such as news articles, images, or social media posts that need to be summarized, organized, or triaged. Machine learning can help with this task, but only when quality labeled training data is available. Such data can be difficult to obtain – labeling cannot be crowdsourced, for example, if expertise is limited to a few individuals. Experienced data scientists know that real-world systems require a human-in-the-loop to provide labels for instances and to correct prediction errors (Amershi, Fogarty, and Weld 2012). However, current human-in-the-loop systems have scalability limitations for both machine and human components, e.g., they are slow to incorporate user feedback and overwhelm the user by showing all the data as points in abstract statistical visualizations.

Our contributions in this paper are performance and design improvements for CHISSL (Arendt, Komurlu, and Blaha 2017), a framework for Computer-Human Interaction for Semi-Supervised Learning. These contributions allow CHISSL to overcome the aforementioned scalability issues. The primary task we support with CHISSL is to perform transductive learning by allowing the user to explore and organize large amounts of unlabeled instances into groups that need not be defined *a priori*. In future work, we plan to allow the user to export their grouping as a training set for a classification model to facilitate inductive learning.

There are several approaches for human-in-the-loop transductive learning including spatialization, interactive clustering, and active learning. Spatialization is a visual analytics technique where the user's mental model of a dataset is learned and projected into a two dimensional space. Brown et al. 2012 pioneered variants of this approach where model parameters were learned directly from user interactions e.g., dragging an instance on the screen. A disadvantage of these techniques is that they require the user to interpret groups from a 2-D projection of the data, and such projections can be misleading (Chuang et al. 2012). Interactive clustering addresses a similar problem by allowing a user to guide a clustering algorithm with rejection (Srivastava, Zou, and Sutton 2016) or providing linkage constraints (Bilenko, Basu, and Mooney 2004). Likewise, supervised and semi-supervised learning has been implemented in systems that elicit user feedback and perform label induction or transduction to update predicted labels (Kulesza et al. 2014). Furthermore, instances are usually represented abstractly in statistical plots, making it unclear about what instances may be in need of correction. Finally, the above techniques are not scalable because they require too much time to incorporate user feedback or they show all instances on the screen, overwhelming the user.

Approach

CHISSL is designed for non-experts, allowing users to drag and drop instances to organize their data into groups. While CHISSL scales to large amounts of data, we keep the user interface simple by limiting what the user sees to a few recommendations of similar instances for each group. The model updates after each user interaction, finding new recommendations and allowing the user to correct classification errors. This is made possible with our implementation of a novel transductive learning algorithm, which was motivated by the limitations experienced using existing semi-supervised learning techniques, specifically label propagation (Delalleau, Bengio, and Le Roux 2005). While this algorithm is described as "efficient," it can to take several seconds to re-

train for datasets with more than a few thousand instances, which is not ideal for an interactive system where the user expects the system to respond within miliseconds.

To address these issues, we designed a client-server algorithm with strict space and time requirements, specifically 1) the representation matrix is kept on the server and is never sent to the client; and 2) the client incorporates user feedback without consulting the server. Meeting these requirements helps the system scale to larger datasets and support many concurrent users. Our solution hinges on the assumption that the shortest path distance between instances in a dendrogram is an effective approximation of thier distance in the representation space.

The server computes the dendrogram using an off-the-shelf agglomerative clustering algorithm and sends it to the client as a memory-efficient parent pointer array. Using this, the client classifies each unlabeled instance with the class of the closest labeled instance. Label transduction is performed for all instances in two phases with $O(n)$ time complexity by iterating over the parent pointer array in forwards and then reverse order. The ordering of the parent pointer array guarantees the first pass visits all children before parents, and the second pass visits all parents before children. Each time a node is visited in either phase, if the current distance to that node plus its distance to its parent is less than the current distance to its parent, then the distance to the parent and its class are updated.

The dendrogram is also used to determine recommendations by partitioning the tree into separate components based on the predicted class label. These components are then split into sub-trees by cutting a few top-level nodes, and a single representative instance is chosen from each subtree. For each user-defined group, CHISSL shows one user-labeled example and the recommended instances in a table row in the user interface. Dragging an example from one group/row to another labels the dragged instance with the dropped group's label, and instances are double clicked to create new groups.

Applications

We applied CHISSL to several different domains including social media images and text (24K images, 8K messages); cloud resource utilization[1] (42K sequences); and as shown in Figure 1, an insider threat scenario[2] (15K entity-day pairs) and to geo-temporal analysis[3] to summarize patterns of life (500 vehicle-day pairs). Our user interface performed responsively in all cases.

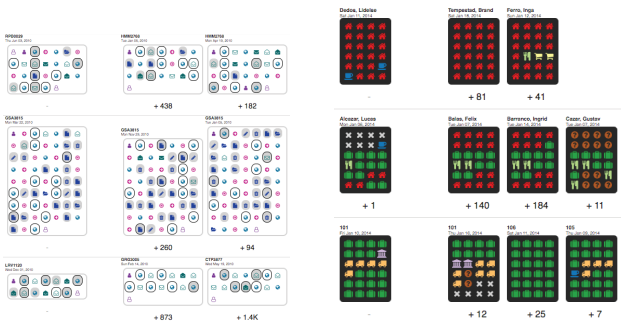
Impact and Future Work

CHISSL is fast, easy to use, and generalizable. We have tested CHISSL on a variety of application domains including social media, image analysis, geo-temporal analysis, and cybersecurity. CHISSL can incorporate user feedback at interactive speeds. Future work will focus on empirical evaluation of its accuracy and usability using ground truth datasets.

[1]https://wiki.openstack.org/wiki/Telemetry#Ceilometer

[2]https://www.cert.org/insider-threat/tools/index.cfm

[3]http://www.vacommunity.org/VAST+Challenge+2014



(a) (b)

Figure 1: CHISSL allows users to flexibly create groups by example. Each group is a row in the user interface. Examples are shown in the left column, and a few recommendations of similar instances are shown to the right. (a) Activities in the CERT dataset: instances are blocks of icons showing user actions within a day. (b) Vehicle movements in the VAST Challenge 2014 dataset: instances are blocks of icons showing a vehicle's hourly activities within a day.

Acknowledgments

The research described in this paper is part of the Analysis in Motion Initiative at Pacific Northwest National Laboratory. It was conducted under the Laboratory Directed Research and Development Program at PNNL, a multi-program national laboratory operated by Battelle for the U.S. Department of Energy.

References

Amershi, S.; Fogarty, J.; and Weld, D. 2012. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of SIGCHI*, 21–30. ACM.

Arendt, D.; Komurlu, C.; and Blaha, L. M. 2017. Chissl: A human-machine collaboration space for unsupervised learning. In *Conference on Augmented Cognition*, 429–448.

Bilenko, M.; Basu, S.; and Mooney, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of ICML*. ACM.

Brown, E. T.; Liu, J.; Brodley, C. E.; and Chang, R. 2012. Dis-function: Learning distance functions interactively. In *Proceedings of VAST*, 83–92. IEEE.

Chuang, J.; Ramage, D.; Manning, C.; and Heer, J. 2012. Interpretation and trust: Designing model-driven visualizations for text analysis. In *Proceedings of SIGCHI*, 443–452.

Delalleau, O.; Bengio, Y.; and Le Roux, N. 2005. Efficient non-parametric function induction in semi-supervised learning. In *Proceedings of AISTATS*.

Kulesza, T.; Amershi, S.; Caruana, R.; Fisher, D.; and Charles, D. 2014. Structured labeling for facilitating concept evolution in machine learning. In *Proceedings of SIGCHI*, 3075–3084. ACM.

Srivastava, A.; Zou, J.; and Sutton, C. 2016. Clustering with a reject option: Interactive clustering as bayesian prior elicitation. In *KDD IDEA Workshop*. ACM.