




# Supplemental Material: Interactive Visual Explanation of Incremental Data Labeling

R. Beckmann<sup>1,2</sup>, C. Blaga<sup>1,2</sup>, M. El-Assady<sup>2</sup>, M. Zeppelzauer<sup>3</sup> and J. Bernard<sup>1,4</sup>

<sup>1</sup>University of Zürich, Switzerland    <sup>2</sup>Swiss Federal Institute of Technology Zürich, Switzerland  
<sup>3</sup>St. Pölten University of Applied Sciences, Austria    <sup>4</sup>Digital Society Initiative, Zürich, Switzerland

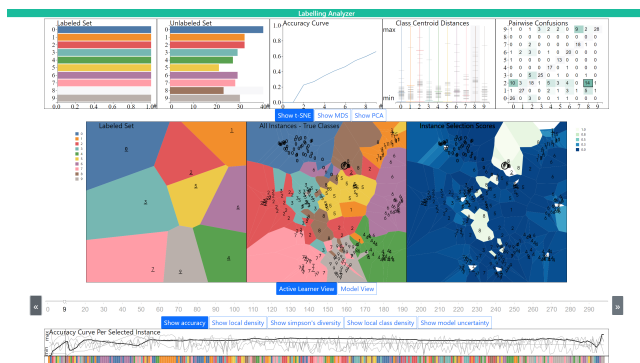


Figure 1: State at iteration 9. Active Learner View.

## 1. Model Specifications

Due to its robustness and scalability, a support vector classifier with a Gaussian kernel was chosen as the underlying model. It was implemented using scikit-learn [PVG\*11], and the standard parameters were kept.

## 2. Dataset Specifications

For this experiment, the MNIST dataset [LBBH98] was used. For interpretability and to avoid too much overplotting, 300 samples of the dataset were selected to form the working dataset. Since the MNIST dataset contains picture data, it is high-dimensional, with 28x28 features per instance. Each feature represents a pixel in the instance.

## 3. Case Study: Interesting Findings

This section summarizes some of the most interesting findings in the tool at hand. Here, we will make use of the parametrization power of the All Instances view. When the Active Learner View is active, all instances are shown with their ground truth coloring. When the Model View is active, the predictions by the model are shown, and are colored if those predictions are wrong.

An interesting finding of iteration 9 can be seen in Figure 1. In the Instance Selection Scores view, only the instances with corresponding class label 8 are given a high score by the ISS. This is because the only label which doesn't have an instance in the labeled dataset yet is the label 8.

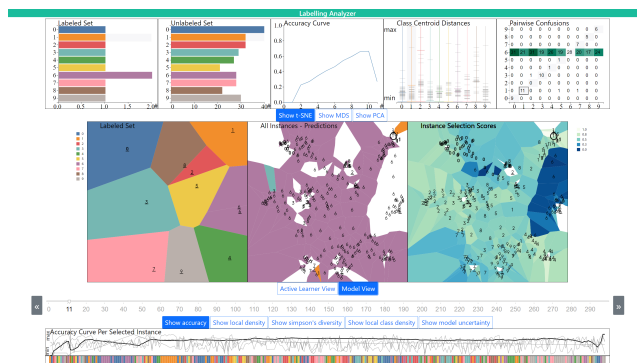


Figure 2: State at iteration 11. Model View.

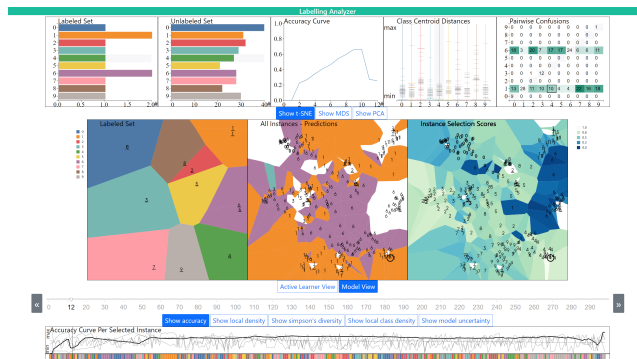


Figure 3: State at iteration 12. Model View.

Figure 2 shows that at iteration 11, the ISS was forced to label one and only one class twice. This means that the dataset is now highly unbalanced, as there are now twice as many instances with the label 6 as there are instances with another label. In the All Instances view, and in the Pairwise Confusions view, one can see that the model, due to the imbalance of the labeled dataset, now almost only predicts the label 6, and is wrong in most cases.

Along these lines, an expected behaviour can be explored in Figure 3. It shows iteration 12, after adding one more instance to the labeled dataset compared to Figure 2. Now the class imbalance favors both labels 6 and 1, which can be seen in both the Pairwise Confusions, as well as in the All Instances view.

For the finding at iteration 17, some more features of the tool can be used. For this, the reader's attention is brought to Figure 4. By

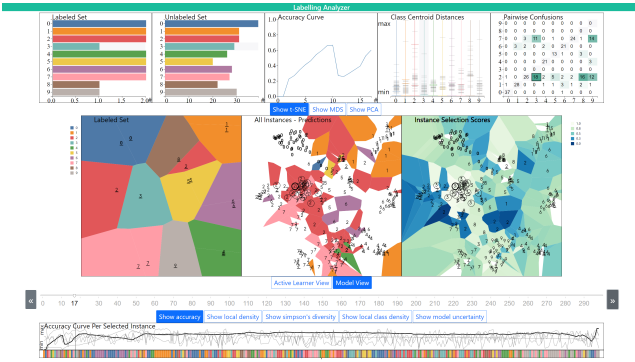


Figure 4: State at iteration 17. Model View.

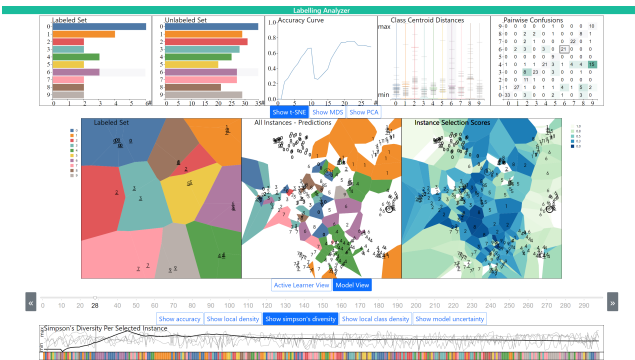


Figure 5: State at iteration 28. Model View.

hovering the mouse over the component of the Pairwise Confusions view which the next selected instance belongs to, all instances that belong to the same component are highlighted with a thin circle in the main plots. In this iteration, the ISS not only tackles the largest current source of class confusion, but it also selects an instance in a particularly dense area of confusion.

Figure 5 proves that the visualizations and metrics at hand sometimes do not suffice to fully explain the behaviours of the Instance Selection Strategy at hand. Following the observations made before, one might expect the ISS to propose one of the 15 instances with label 9 which have actually been classified as 4 by the model. Hovering over the corresponding component in the Pairwise Confusions view also reveals that the corresponding instances are packed in a rather dense area. To make things worse, the scores assigned to those instances by the ISS are not even particularly high. Instead, for the next few iterations, the ISS decides to select instances in the dense regions of class 6 (which are already correctly classified anyways), which also drastically brings up the Simpson’s Diversity metric.

#### 4. Implementation

The tool at hand is implemented with a client-server architecture based on the Flask framework. The client side is written with HTML, CSS, and Javascript, where D3.js [BOH11] is used for the (interactive) visualizations. The server side is written with Python.

#### References

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D<sup>3</sup> data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309. doi:10.1109/TVCG.2011.185. 2

[LBBH98] LECUN Y., BOTTOU L., BENGIO Y., HAFFNER P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324. doi:10.1109/5.726791. 1

[PVG\*11] PEDREGOSA F., VAROQUAUX G., GRAMFORT A., MICHEL V., THIRION B., GRISEL O., BLONDEL M., PRETTENHOFER P., WEISS R., DUBOURG V., VANDERPLAS J., PASSOS A., COURNAPÉAU D., BRUCHER M., PERROT M., DUCHESNAY É.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. 1