

Towards User-Centered Active Learning Algorithms

Jürgen Bernard^{1,2}, Matthias Zeppelzauer³, Markus Lehmann¹, Martin Müller¹, and Michael Sedlmair⁴

¹TU Darmstadt, Germany

²Fraunhofer IGD, Germany

³St. Pölten University of Applied Sciences, St. Pölten, Austria

⁴Jacobs University Bremen, Germany

Abstract

The labeling of data sets is a time-consuming task, which is, however, an important prerequisite for machine learning and visual analytics. Visual-interactive labeling (VIAL) provides users an active role in the process of labeling, with the goal to combine the potentials of humans and machines to make labeling more efficient. Recent experiments showed that users apply different strategies when selecting instances for labeling with visual-interactive interfaces. In this paper, we contribute a systematic quantitative analysis of such user strategies. We identify computational building blocks of user strategies, formalize them, and investigate their potentials for different machine learning tasks in systematic experiments. The core insights of our experiments are as follows. First, we identified that particular user strategies can be used to considerably mitigate the bootstrap (cold start) problem in early labeling phases. Second, we observed that they have the potential to outperform existing active learning strategies in later phases. Third, we analyzed the identified core building blocks, which can serve as the basis for novel selection strategies. Overall, we observed that data-based user strategies (clusters, dense areas) work considerably well in early phases, while model-based user strategies (e.g., class separation) perform better during later phases. The insights gained from this work can be applied to develop novel active learning approaches as well as to better guide users in visual interactive labeling.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Labeling data objects is a fundamental process in machine learning (ML), data mining, and visual analytics (VA). Labeling refers to the task of attaching a certain attribute to an instance in a data set, such as a class label, a relevance score, or a similarity judgment with regard to another instance. The major goals of (interactive) labeling are to acquire knowledge from the user to guide the learning process, to generate “ground-truth” data, and to model the user’s interests and intentions. The labeling of large amounts of data, however, is a time-consuming and tedious task. Especially in the presence of large data sets efficient labeling strategies are required to reduce labeling effort and accelerate learning.

Active learning (AL) is an ML approach that tries to minimize the amount of human interaction for labeling. To this end, AL employs *candidate selection strategies* that query the user only for labels of those samples that the ML model will benefit most from. AL is typically a model-centered approach that selects instances with respect to the model that should be learned. A limitation of AL is that users are not involved in the *selection* but only in the *labeling* of candidate instances. Thereby the potential of exploiting user knowledge (domain knowledge) and the user’s intuition in identifying patterns, clusters, and outliers remains underutilized.

The success of AL thus heavily relies on the quality and suitability of the applied selection strategy.

Labeling is also frequently required in VA where user feedback from labeling is exploited to learn and support the user’s information need. In contrast to AL, in VA the *selection and labeling* of candidates is primarily user-driven, which makes it a user-centered approach and thus complementary to AL. To bridge the gap between AL and VA, previous work has introduced the idea of Visual Interactive Labeling (VIAL) [BZSA17]. VIAL allows for both the user and the model to propose candidates for labeling and thereby combines the strengths of both labeling perspectives into one unified process.

This paper aims at narrowing the gap between AL and VA by comprehensively studying user-based selection strategies. To this end, we perform an in-depth analysis of ten previously identified user-based labeling strategies [BHZ*17]. These strategies comprise different ways of how users go about selecting data points to be labeled from a 2D projection (scatterplot) of the data, and were found through iterative user experiments. Users for instance selected points in dense regions or outliers in the scatterplot (*data-based user strategies*). Other users took information from the visualized ML model into account, for instance by selecting instances for labeling which are close to intersecting border regions of two

classes (*model-based user strategies*). While these strategies have been identified by observation [BHZ*17], their systematic investigation and formalization has not been performed so far. However, this is important to understand *how* users actually select candidate instances, *how* such strategies perform, and *how* knowledge about these methods will support effective labeling. Based on this knowledge novel methods for automated candidate selection in AL may be developed as well as advanced user-inspired methods for visual guidance in VA labeling approaches.

We analytically formalize the ten previously observed user strategies for the selection of instances, identify their building blocks, implement them, and integrate them into an automatic evaluation toolkit. We run simulated labeling experiments with all strategies on different data sets, analyze their performance, and compare them with AL strategies. Additionally, we use two baselines to put the performances of all AL and user-based strategies into context: (i) a random baseline to measure the lower performance limit and (ii) a quasi-optimal selection strategy which always selects the best candidate in a greedy fashion (in a supervised manner based on ground-truth labels) to provide an upper limit of performance (ULoP).

In particular, our experiments seek to answer the following questions: (i) can we formalize and thereby automate user-based selection strategies? (ii) can formalized user-based strategies compete with or even outperform existing AL strategies? And beyond this, can we observe patterns and trends that hold across different data sets and ML tasks and are thus generalizable? (iii) which building blocks are common to the strategies and can they be carved out for future use? (iv) can we improve existing labeling algorithms with a better understanding of human labeling behavior, represented with formalized user strategies and building blocks?

This work represents a first step towards *user-centered* AL algorithms which may in the future facilitate labeling processes in ML, VA, and VIAL. In summary, the major contributions are:

- We propose a formalization and implementation of 10 user strategies to enable a systematic evaluation of user-based visual labeling in comparison with AL.
- We break down these 10 user strategies and propose 11 low-level algorithmic building blocks that in combination completely formalize the 10 high-level user strategies.
- We present the results of a performance analysis of the 10 user strategies and 9 alternative strategies (AL, upper and lower bound), on four data sets. We investigate the quality of each strategy in selecting useful labels as well as in its capabilities in solving the bootstrap problem.

2. Related Work

Related work comes from the different research areas that focus on techniques in the interactive labeling processes: i.e. active learning, interactive visualization, and the combination of the latter in the VIAL workflow. In the following, we briefly review the most important approaches and labeling strategies employed in these areas.

2.1. AL-Centered Labeling Strategies

Active Learning (AL) strategies have been introduced to support the incorporation of user knowledge into the learning process. In AL, an algorithmic model pro-actively asks an oracle (the user) for feedback (labels) to improve the learning model [Set09]. Since user interactions are time-consuming, AL aims at minimizing the amount of required user interaction by querying only that information that will most likely best improve the accuracy of the given model. Different classes of AL strategies have been introduced [Set09, Ols09, TVC*11, WH11], which we partition into five groups: (i) uncertainty sampling, (ii) query by committee, (iii) error reduction schemes, (iv) relevance-based selection, and (v) purely data-centered strategies.

Uncertainty sampling aims at finding instances the learner is most uncertain about. Widely used strategies search for instances near the decision boundary of margin-based classifiers [WKBD06, TVC*11, SDW01], or measure the entropy of instances' class probabilities [JPP09]. *Query by Committee* (QBC) [SOS92] strategies measure the uncertainty of an ensemble of classifiers including the assessment of the committees' disagreement [Mam98, MM04]. *Error reduction schemes* focus on the selection of those instances which may change the underlying classification or optimization model most. Techniques focus either on the impact on the training error (expected model change) [SCR08] or on the reduction of the generalization error (risk reduction [QHR*09], energy reduction [VBF12], and variance reduction [HJL06]). *Relevance-based* [VPS*02] strategies select those instances which have the highest probability to be relevant for a certain class, e.g., based on positive examples for a class [WH11]. Finally, there are purely *data-driven* strategies which are independent of the learning model, such as density- and diversity-based instance selection [WKBD06, BSB*15, BDV*17, DRH06]. Density-based selection of candidates is a promising strategy for initiating an AL process in the case when no labels are available at all (*cold start* problem or *bootstrap problem*) [AP11]. Recently, approaches towards learning candidate selection strategies have been introduced [KSF17]. This requires, however, the availability of a series of previous active learning experiments to draw useful conclusions from it.

2.2. User-Centered Labeling Strategies

User-centered labeling is implicitly used in many interactive visualization and VA approaches to assign users an active role in the learning process. For this purpose, visual interfaces are usually provided that show the data (or feature space) and the state of the learning process (ML model), e.g., by applying dimensionality reduction in combination with scatter plots [BKSS14, SA15, HMdCM17] or by visualizing instances as glyphs arranged by similarity in a 2D spatial layout [BLBC12, BSR*14, BRS*17]. Mamani et al. use an interactive labeling interface to adapt feature spaces, leading to optimized embeddings of image data calculated with dimensionality reduction [MFNP13]. Once the data or feature space is visualized adequately, the user is asked to select individual instances and provides the respective labels [BZSA17]. Labels in this context can be of different type, such as categorical labels [HNNH*12], numerical labels [BSB*15], relevance scores [SSJK16], as well as labels that represent a relation between two instances (e.g. for learning simi-

larity and distance measures) [BRS*17]. The type of the data to be labeled differs among recent studies.

Bernard et al. propose an approach where users play an active role in selecting data instances and assigning numerical well-being scores for medical data to calculate regression models [BSB*15]. Höferlin et al. [HNH*12] propose a system that facilitates interactive classification of surveillance videos with user input based on relevance feedback for video subsequences. A visual-interactive and user-driven search system for text documents is presented by Heimerl et al. [HKBE12], allowing users to label relevant and non-relevant documents to improve the quality of a retrieval component. In a system for interactively labeling human motion capture data, users can select and label human gestures to build a classifier [BDV*17], visually guided by data- and model-centered ML support. A more general approach for the labeling of time series data has been introduced by Sarkar et al. [SSBJ16], including clustering techniques for pattern detection and user guidance. Finally, the user-centered selection and labeling of instances was applied to learn similarity models for mixed data [BSR*14]. In addition, the authors reflect on a series of pitfalls for the design of labeling approaches.

Although the latter approaches enable the user to select instances and learn models from the provided labels, they do not investigate strategies that motivate users to *select* certain instances in a provided visualization. To the best of our knowledge, this area has gained very little attention so far. Seifert and Granitzer [SG10] were the first that sought to simulate and formalize user-picking strategies such as selection. In a small-scale, initial study, they used a star coordinate interface to illustrate the potentially large impact of combining active and user-based learning strategies. Bernard et al. [BHZ*17] built on this initial study and investigated user-based and AL strategies in more details in an experimental user study. In an experiment, the authors identified 10 different user strategies for the selection of candidate instances, observed during three user tasks followed by interviews. Users were asked to label instances in a scatterplot visualization based on dimensionality reduction. Different coloring and convex hull techniques were used to depict data and classifier characteristics and thus support users in the selection of candidates. In this paper, we formalize, implement, and systematically evaluate these strategies.

3. Formalization of User Strategies

We build upon the 10 strategies of Bernard et al. [BHZ*17]. After briefly reiterating on them (Section 3.1), we present how we broke down these strategies into recurring fundamental *building blocks* (Section 3.2). We then use these building blocks to *formalize* the 10 user strategies algorithmically so that they can be implemented for computational uses (Section 3.3). The intuition behind this approach is to integrate the users' abilities to identify patterns directly into automatable labeling algorithms. This allows to simulate how humans go about labeling data and to compare the user strategies to alternative strategies such as those from AL (see Section 4).

3.1. Background

Table 1 (left side) summarizes the 10 user strategies which represent the basis for our further analysis. The strategies were observed

in interactive labeling experiments where users were presented a 2D projection of the original feature space [BHZ*17]. Unlabeled items were represented as data points (crosses) while for labeled items the class label was shown. To visualize class information and the state of the iteratively learned classifier, different visualization techniques were used, such as convex hulls and butterfly plots.

The identified user strategies form two groups: *data-based strategies*, where the user focused mostly on the data itself and its distribution as well as *model-based strategies* where the user rather focused on class characteristics such as predicted boundaries, e.g., depicted as convex hulls. Different visualizations inspired different strategies. As convex hulls of different classes may overlap, strategies trying to reduce this overlap were observed (e.g. “Class Intersection Minimization” and “Class Borders Refinement”). Finally, “Ideal Labels First” is driven by the content of individual instances while the remaining strategies primarily focus on structural information of the data and the class predictions. Note further that “Ideal Labels First” in contrast to all other strategies requires as input one representative instance for each class and is thus not completely unsupervised.

3.2. Building Blocks

We systematically investigated the individual user-based strategies and identified basic recurring units (building blocks) that are shared between strategies. In total, we identified 11 building blocks that form a minimal and complete set of blocks necessary to implement the 10 user strategies. In the following, we describe and specify formally the individual blocks.

To define the building blocks, we use the following notation. Let V_c be the set of all (unlabeled) candidate feature vectors and V_l the set of all already labeled training feature vectors. The entire data set is then $V = V_c \cup V_l$. To measure distances between instances of V we define a distance function as $d : V \times V \rightarrow \mathbb{R}$. Furthermore, we define a subset T with $T \subseteq V_c$ which represents an arbitrary subsets of V_c [†]. Variable $x \in V_c$ refers to a candidate instance from V_c in our notation and is represented by a feature vector of potentially high dimension in the feature space of V . Predicted labels of individual instances are referred to as y^l and stem from the set of possible class labels $y_i \in Y$ with $n = |Y|$ is the number of classes. The identified building blocks are defined the following.

Nearest Spatial Neighbors (NSN) retrieves instances in the neighborhood of a candidate instance, allowing the assessment of local data characteristics around it. It can be implemented in a straight-forward manner based on k -nearest neighbor (kNN) search. Let v_1, \dots, v_n be an ordering of instances $v \in S \subseteq V$ with $x \notin S$, such that $d(x, v_1) \leq \dots \leq d(x, v_n)$. Then the k -nearest neighbors of instance x are represented by function: $kNN(x, S, k) = \{v_1, \dots, v_k\}$. Note that alternatively, the neighborhood around instance x can be specified by defining a radius (orbit epsilon) around x .

Spatial Balancing (SPB) tries to locate instances in so far undiscovered areas of the feature space and thereby tries to uniformly

[†] Note that the definition of building blocks on a subset makes their definition more general which is beneficial for reuse and flexible combination.

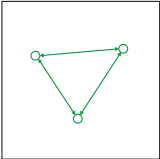
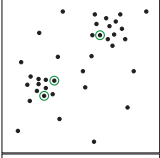
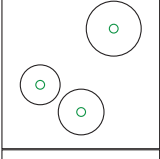
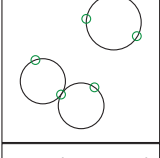
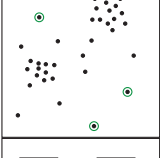
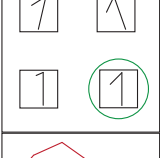
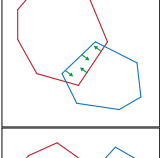
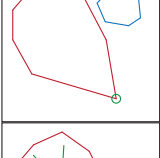
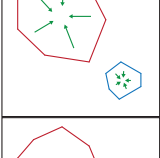
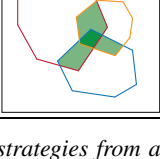
Strategy Name	Illustration	Description Details about the strategies	Formalization In terms of building blocks from Section 3.2.
Equal Spread data-based		Users prefer to label instances in unlabeled areas, i.e., far away from other labeled instances, to distribute labels uniformly across the data.	The implementation builds upon Spatial Balancing (SPB) which identifies instances with maximum distances to the already labeled instances.
Dense Areas First data-based		Users prefer to label instances located in dense regions in the feature space. In contrast to centroid-based strategies instances do not have to be at the center of a dense region.	This strategy combines a spatial Density Estimator (DEN) with Spatial Balancing (SPB), we weight both building blocks equally (50%:50%).
Centroids First data-based		Similar to “Dense Areas First” but users prefer to label instances located at cluster centers uniformly across the feature space.	This strategy combines a Clustering (CLU) algorithm with Spatial Balancing (SPB), we weight both building blocks equally (50%:50%).
Cluster Borders First data-based		Users prefer to label instances located in the border regions of clusters. These areas may potentially represent class boundaries and thus may contain relevant instances for labeling. Additionally, candidates should be spread across the feature space.	Applies a Clustering (CLU) algorithm first. Then, for every cluster, an Outlier Detection (OUT) algorithm is combined with Spatial Balancing (SPB), we weight both building blocks equally (50%:50%).
Outliers First data-based		Users prefer to label instances located in sparsely populated regions, i.e. outliers in different areas across the feature space. This strategy is used to label potentially untypical instances of a class.	This strategy combines an Outlier Detection (OUT) algorithm with Spatial Balancing (SPB), we weight both building blocks equally (50%:50%).
Ideal Labels First data-based		The user selects instances for labeling she believes are most characteristic for a class, by comparing instances to an a priori given example instance, which is typical for the class. We refer such typical instances to as “ideal instances”.	This strategy combines the Ideal Instance Identification (III) building block with Spatial Balancing (SPB), we suggest to weight both building blocks equally (50%/50%).
Class Borders Refinement model-based		Users try to refine the spread of class boundaries in regions with overlapping class borders (in the low-dimensional representation of the data).	This strategy uses the results of a classifier (CL and CP) and then applies Local Class Diversity (LCD) assessment to emphasize border regions.
Class Outlier Labeling model-based		Users select instances for labeling that are far away from the class center. Referring to convex hull visualizations of class borders such outliers are salient spikes of the hull polygon.	This strategy applies classification (CL and CP) first. The individual predicted class distributions are assessed with an Outlier Detection (OUT) algorithm.
Class Distribution Minimization model-based		Users try to minimize the area covered by an individual class distribution. This can, e.g., be achieved with instances that probably help to reduce the size of the convex hull or another visualization of class boundaries.	This strategy applies classification (CL and CP) first. Compactness Estimation (CE) assess the compactness of each class; distances to the class centroid allow weighting of instances within each class.
Class Intersection Minimization model-based		Users try to minimize areas where classes intersect (e.g. represented by convex hulls in scatterplots). Counting the number of overlapping convex hulls is one means to assess class intersection.	This strategy applies classification (CL and CP) first. Then Local Class Separation (LCS) retrieves the Nearest Spatial Neighbors (NSN) per class and assesses the degree of class separation.

Table 1: Ten user strategies from a previous experiment [BHZ*17]. The strategies can be partitioned into data-based and model-based strategies. The data in the original study was represented with dimensionality reduction in a 2D scatterplot. Class boundaries in model-based strategies were visualized among others by convex hulls of the labels predicted by the underlying classifier.

distribute the labeling candidates across the space. For each sample $x \in V_c$ first the minimum distance B to the labeled samples $t \in V_l$ is determined: $B(x) = \min_{t \in V_l} d(x, t)$. Second, the sample x with maximum distance to a labeled sample is selected: $SPB(V_c) = \operatorname{argmax}_{x \in V_c} (B(x))$.

Clustering (CLU) partitions a set of instances into disjoint groups or clusters C_1, \dots, C_n of similar instances. Clustering provides a meta-structure on the original data and is a common building block for several selection strategies where it facilitates the selection of candidates at e.g. cluster centroids, cluster border areas, or at spatially close clusters. Clustering can be implemented in many different ways [Jai10]. To stay general we define a generic clustering function CLU on the set of unlabeled instances V_c as: $CLU(V_c) = \min(f_{cost}(\{C_1, \dots, C_n\}))$ such that all instances are assigned to exactly one cluster, i.e., $\forall x \in T : \exists C_k : x \in C_k$ and $\forall x \in T : (x \in C_k \wedge x \in C_l) \implies k = l$. Function f_{cost} represents the cost function to be minimized during clustering.

Density Estimation (DEN) identifies maximally dense areas in the feature space. It can be used to select instances that seem typical for the data set and that have many similar neighbors. Density estimation is defined by a scoring function $DEN(x) = score_{DEN}(x)$ which is applied to all instances $x \in V_c$ to find candidates in maximally dense areas. Density can be estimated by different approaches, such as averaging the distances to the k -nearest neighbors of x : $score_{DEN}(x) = -\sum_{v \in kNN(x, V_c, k)} \frac{d(x, v)^2}{k}$

Outlier Detection (OUT) In contrast to density estimation (above), outlier detection tries to find instances in sparsely populated regions. It can be used to select instances with untypical characteristics and helps to better sample the variability that exists in the data and its classes, respectively. Outliers can be identified by using different outlier scoring functions: $OUT(T) = \{score_{OUT}(v, T) \geq t : v \in T\}$, where $t \in \mathbb{R}$ is a decision threshold. Outlier detection methods can be partitioned into classification-based, clustering-based, nearest neighbor, and density-based [KN98] approaches. A straight-forward implementation would be to use the negative density $score_{OUT}(v, T) = -DEN(v)$.

Compactness Estimation (CE) determines the compactness of groups of instances. Groups may be clusters obtained from clustering or sets of instances with the same predicted class. This measure can be used to favor instances for labeling in either compact or spatially distributed groups. A compactness function has the following general signature: $CE(T) = score_{CE}(T)$. A typical realization of CE is the variance of the instances in a group: $CE(T) = \frac{1}{|T|} \sum_{x \in T} (d(x, m_i))^2$, where m_i is the average of the group.

Ideal Instance Identification (III) requires a set of representative instances as input which a user considers “ideal” for certain classes. This makes it structurally different from the other building blocks. It requires the user first to specify which instances are considered ideal. This can be formalized as follows. The function U specifies which samples are considered ideal (by the user in the loop):

$$U(x) = \begin{cases} 1, & \text{if } x \text{ is considered ideal by the user} \\ 0, & \text{otherwise} \end{cases}$$

In a second step, this building block estimates the relevance of any instance $x \in V_c$ with respect to the set of representative instances. The score for an instance x is the minimal distance to one of the “ideally” labeled instances: $III(x) = \min_{u \in \{v \in V : U(v)=1\}} d(x, u)$.

Class Likelihood (CL) represents the likelihoods l provided by a given (pre-trained) classifier for an unlabeled instance x as: $CL(x) = l$, with $l \in \mathbb{R}^{|Y|}$ is a vector of probabilities for each of the classes in Y .

Class Prediction (CP) estimates the most likely class y' for set of class predictions l : $y' = CP(x) = \operatorname{argmax}(CL(x))$.

Local Class Diversity (LCD) assesses the diversity of class predictions in the neighborhood of an instance x . Thus, each instance needs to have a most likely class y' assigned by a classifier. Given an instance x and a class label $y_i \in Y$, we can compute the portion p_i of neighbors with the class prediction $y' = y_i$ as: $p_i(x, y_i) = \frac{|\{v \in kNN(x, V_c, k) : CP(v) = y_i\}|}{k}$. The local class diversity can then be estimated by a diversity measure div as follows: $LCD(x) = div(p)$, where p is the vector of all portions p_i for the n classes: $p = (p_1, \dots, p_n)$. The entropy of p is one possible realization of function div .

Local Class Separation (LCS) is similar to Local Class Diversity but estimates how well the predicted classes around a given instance are separated from each other. It can be used to identify regions with high class uncertainties. We define this building block based on subsets C_i , $i = 1, \dots, n$, which are the k -nearest neighbors of instance x found for each of the n classes: $C_i = kNN(x, \{v \in V : CP(v) = y_i\}, k)$. Local Class Separation LCS is then defined as a separation function f_{sep} over all subsets C_i : $LCS(x) = f_{sep}(C_1, \dots, C_n)$. Such a scoring function could be based on Dunn-like indices [Dun74], the Davies-Bouldin Index [DB79], or Silhouettes [Rou87].

3.3. Formalized User Strategies

Having identified the building blocks from Section 3.2, we can now formalize the user strategies from Section 3.1. To implement the user strategies, we first implemented the underlying building blocks. For implementation details, we refer the interested reader to the supplemental material. In a next step, we constructed the original user strategies from the building blocks according to the formalizations in Table 1 (column “Formalization”). To better understand how the implementation was performed, consider the example of “Centroids First” strategy. “Centroids First” focuses on cluster centers on the one hand and thus requires building block “Clustering”. On the other hand, users at the same time tried to select and label centroids spatially distributed, i.e. from previously unlabeled clusters before re-visiting already labeled clusters, which refers to “Spatial Balancing”. The scores of both building blocks can be combined by taking their average. The result is a strategy that selects cluster centers which are distributed across the entire space. In all cases where several building blocks are used to vote for instances, we use equal weights to keep the number of free parameters in the evaluation limited. Other weightings will be subject to future work.

Building Blocks → User Strategy ↓	NSN	SPB	CLU	DEN	OUT	CE	III	CL	CP	LCD	LCS
Equal Spread		x									
Dense Areas First	x	x		x							
Centroids First		x	x								
Cluster Borders First		x	x								
Outliers First		x			x						
Ideal Labels First	x	x		x			x				
Class Borders Refinement								x	x	x	
Class Outlier Labeling					x			x	x		
Class Distribution Min.						x		x	x		
Class Intersection Min.	x			x				x	x		x

Table 2: Mapping between user strategies and building blocks. Most user strategies build upon two or more building blocks. From a building blocks’ perspective Spatial Balancing is included in six user strategies, Nearest Spatial Neighbor, Density Estimation, Class Likelihood and Class Prediction in four.

Table 2 shows the exact mapping between building blocks and strategies and shows which blocks contribute to which strategies and beyond this, which blocks appear how often and are thus of particular importance. “Spatial Balancing” for example appears most often. Nine of ten user strategies build upon combinations of building blocks. Only the “Equal Spread” consists of a single block (“Spatial Balancing”). Furthermore, the table reveals which user strategies share similar blocks and are thus similar, e.g. “Cluster Borders First” and “Class Borders Refinement” or “Class Distribution Minimization” and “Class Intersection Minimization”. The representation in Table 2 further shows which combinations of building blocks are used by previously identified strategies. Finally, we note that, the set of possible building blocks may be much larger than the 11 building blocks needed to formalize the investigated user strategies. This set opens up a design space for novel strategies not yet observed.

4. Experiments & Results

After having formalized the user strategies, we now can algorithmically compare different labeling strategies. Our overall goal was to investigate the different performances of the 10 user strategies for different classification tasks, compared to AL algorithms as well as to lower and upper bound baselines. In addition, we investigate the performance of the individual building blocks based on the performance of the user strategies. We broke down the analysis into the following three research questions:

- **RQ₁** How well do the 10 formalized user strategies perform in the very early bootstrap phase of the labeling process; compared to lower and upper baselines, as well as AL strategies?
- **RQ₂** How well do these strategies perform after the bootstrapping phase when at least one instance per class was labeled?
- **RQ₃** How do the low-level building blocks contribute to the performance of the user strategies under different conditions?

4.1. Candidate Selection Strategies

Overall, we conduct performance analysis for 19 strategies, with an emphasis on comparing 10 user strategies with each other and to compare them to 7 AL strategies and two baselines (upper and lower limit of performance).

10 User Strategies. We implemented the 10 user strategies according to the formalization described above (in Java on a standard desktop computer). We use kMeans for the CLU building block (k = label cardinality), the number of instances to be retrieved with nearest neighbor operations (NSN, DEN, LCD, LCS) is $\min(\max(5; V_c \times 0.05); 50)$. CLS adopts the Silhouettes Index [Rou87] for cluster/class separation, Entropy [Sha48] is used to measure diversity (LCD). To simulate our experiments, we apply the user strategies in the original feature space (instead of in the projected low-dimensional space). This allows the formalized strategies to fully exploit the potential of high-dimensional data structures, and allows for a fair comparison to AL and baseline strategies that operate in the same space.

7 Active Learning (AL) Strategies. We integrate 7 AL strategies into the experiments aiming for robustness, generalizability, and variability. Techniques include Smallest Margin [SDW01, WKBD06], Entropy-Based Sampling [SC08], Least Significant Confidence [CM05] representing simple, fast, and frequently applied *uncertainty sampling* strategies [Set12]. In addition, Average Kullback Leibler Divergence [MN98], Probability Distance (distance of classifiers’ likelihoods), Vote Comparison (diversity of predicted labels), and Vote Entropy [Sha48, DE95] build the set of *Query-By-Committee* (QBC) [Mam98] strategies. AL techniques based on error/variance reduction are computationally expensive [Set12] as classifiers need to be trained for every instance and iteration. Hence, we do not include these AL techniques since they do not allow interactive execution [MPG*14].

2 Baseline Strategies. We also include a lower and upper-bound baseline. As the lower bound, we use a series of trials conducted with randomly selected instances (*Random Baseline*). As an upper bound, we pre-compute the quasi optimum performance that can be achieved by always selecting the instance with a maximum performance gain. A simple algorithm that fulfills this criterion is a Greedy search using the ground truth data (Upper Limit of Performance *ULoP*). In every iteration, the algorithm trains and tests the model for every single candidate instance, and finally chooses the winning instance. Note that Random Baseline and ULoP are additional strategies allowing the comparison and assessment of performance [Caw11]. Neither Random Baseline nor ULoP are user strategies and thus, do not consist of the building blocks described in Section 3.2.

4.2. Data Sets

The concrete results of the experiment depend on the data set of choice. We thus compare the strategies with different data sets, selected according to the following considerations:

- Numerical attributes/features
- A size of at least several thousand instances
- Public availability
- Intuitiveness, wide distribution in research community

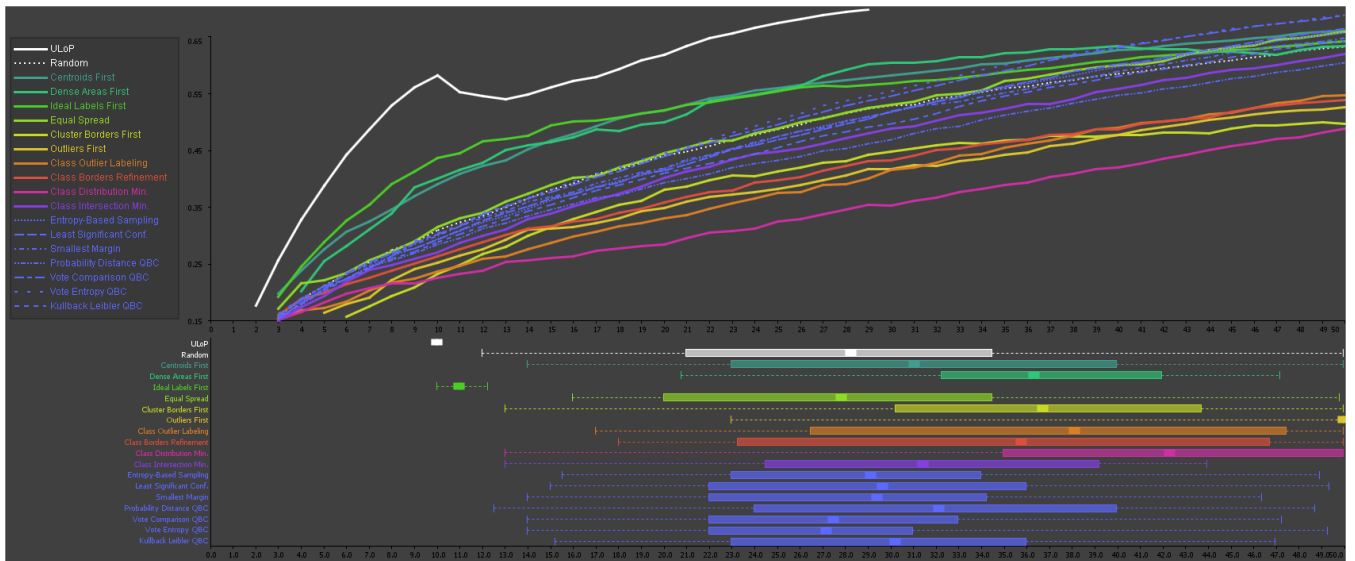


Figure 1: Average performance of strategies in the first 50 iterations of the labeling process (MNIST data set). Most data-based user strategies perform particularly well in the very first iterations (Ideal Labels First, Centroids First, Dense Areas First). Most model-based user strategies perform below Random. Focusing on the distinction between density-based and outlier-based strategies, the density-based strategies perform particularly well, while outlier-based strategies perform poorly. The performance of AL strategies is rather low at start, but increases with more iterations. Boxplots at the bottom show the distribution of the iteration numbers when strategies produced at least one label for each class. The Ideal Labels First strategy visits all class labels remarkably early (all 10 labels in 11 iterations on average), thus, it uses the set of a-priori given ideal representatives as expected. Model-based strategies require more iterations to see each label at least one time. As a general rule, strategies with good accuracies also visited every class label earlier. We conclude that data-based strategies are better suited to tackle the bootstrap problem than the other strategies. At the beginning of the labeling process the class boundaries generated in model-based strategies seem to be less expressive (i.e. they may jump around chaotically) and thus it is more robust to go for data and structure.

Finally, we aimed at covering a broad range of data-based characteristics, such as (1) binary classification versus multi-class classification, (2) equally-balanced label distribution versus unbalanced distribution, (3) proof-of-concept versus real-world complexity, as well as (4) few versus many outliers. These considerations led us to four data sets. The supplemental material contains a detailed overview of the data sets including visualizations of low-dimensional projections.

Handwritten Digits The *MNIST* data set [LBBH98] perfectly meets all requirements to the data. It consists of thousands of raw images showing handwritten digits, each represented by a 28x28 image (784 dimensional vector). For faster classification, we use a descriptor that extracts slices in horizontal, vertical, and diagonal direction, yielding feature vectors with 42 numerical dimensions.

IRIS The *IRIS* data set [Lic13] does not fulfill the criterion of thousands of instances, rather it consists of three classes with 50 instances each. However, we consider *IRIS* as a low-dimensional and easy to interpret proof-of-concept dataset.

Gender Voices The *Gender Recognition by Voice* data set [Bec16] contains acoustic properties of voices to identify the gender of speakers. This data set consists of 3,168 instances which are pre-processed by acoustic analysis with an analyzed frequency range of 0-280 Hz. The outcome is a 21 dimensional vector of acoustic properties; many instances have a tendency to be outliers.

Fraud Detection The *Credit Card Fraud* data set [PCJB15] con-

tains transactions of credit cards recorded in Europe in two days in September 2013. Overall, 492 frauds are included in the 284,807 transactions (0.172%), we use the data set to assess a highly unbalanced classification problem. The data consists of 28 numerical features; many instances have a tendency to be outliers.

4.3. Data Analysis

We use classification *accuracy* as the main dependent measure for our performance comparisons of balanced data sets (percentage of correctly predicted instances compared to ground truth data [FHOM09]). For unbalanced data sets we compute the class-wise *f1* measure from recall and precision [SWY75] and average the *f1* scores over all classes (in a macro averaging fashion). We measure performances each time a new instance has been selected by a certain strategy for labeling, and in doing so, observe how performance changes over the labeling process. For our experiments, we use 50 labeling iterations. The labels are set automatically according to the ground truth that we have for all our data sets. We average the results of an ensemble of classifiers to achieve robust predictions, including Naive Bayes [DHS*73], Random Forest [Bre01] (RF), Multilayer Perceptron [HSW89] (MP), and Support Vector Machine [CV95] (SVM).

We examine the performance of iterative labeling with 19 strategies (cf. Section 4.1) with all four data sets (cf. Section 4.2). All strategies are executed in an automated batch process. To

Building Block	MNIST	IRIS	GEND.	FRAUD
Nearest Spatial Neighb.	++	++	++	+
Spatial Balancing	++	++	++	++
Clustering	++	++	∅	++
Density Estimation	++	++	++	∅
Outliers Detection	--	--	-	+
Compactness Estim.	--	∅	--	-
Ideal Instance Ident.	++	++	+	-
Class Likelihood	-	∅	∅	+
Class Prediction	-	∅	∅	+
Local Class Diversity	-	--	-	+
Local Class Separation	∅	∅	∅	+

Table 3: Analysis of the bootstrap phase at the granularity of building blocks. The performance of building blocks (implemented in user strategies) is depicted with discrete ordinal scores from very bad to very good (--, -, ∅, +, ++). Blocks employing data characteristics help to tackle the bootstrap problem.

achieve robust results the calculation of each strategy is repeated 50 times [KCH*17]. The order of candidate instances is randomized in each trial with a constantly incrementing seed, in order to have distinct and reproducible repetitions.

To analyze these results, we use superimposed line-charts that depict the strategies’ performance progressions over the labeling process. The large number of trials allows for robust average performance curves for the strategies. To keep the number of colors manageable, we use dashed lines to distinguish ULoP and Random Baseline, and the AL strategies. The ULoP and Random baseline are white; AL strategies are blue; and user strategies have categorical colors from green to purple (data-based to model-based).

Finally, we assess the performance of individual building blocks. To that end, we use the mapping of building blocks to formalized user strategies depicted in Table 2 as a basis. To assess the performance of a building block, we are interested in the maximum performance a building block can provide. Thus, we assess the maximum performance within a set of user strategies which all implement a particular building block. We use a discrete ordinal scale to depict the performance of building blocks (--, -, ∅, +, ++), ranging from very bad to very good.

4.4. Results

4.4.1. RQ₁: Analysis of the Bootstrap Problem

The first part focused on the bootstrap (cold start) problem which is well-known in machine learning field. The challenge here is to start the learning in the very first phase without any labels at all [MN98]. We are interested in strategies that are able to cope well with the bootstrap problem, i.e. that sample at least one item of each class as early as possible in the labeling process. This is essential for the performance of further learning progression.

User Strategies Figure 1 shows the result of the MNIST data set. Figures with results for the remaining three data sets are provided in the supplemental material. We start the result analysis with three generalizable patterns, each occurred in at least three of the four

Building Block	MNIST	IRIS	GEND.	FRAUD
Nearest Spatial Neighb.	++	+	+	++
Spatial Balancing	++	+	++	-
Clustering	++	+	--	--
Density Estimation	++	+	++	++
Outliers Detection	-	∅	--	-
Compactness Estim.	--	+	--	--
Ideal Instance Ident.	++	+	+	-
Class Likelihood	∅	++	-	++
Class Prediction	∅	++	-	++
Local Class Diversity	--	-	-	-
Local Class Separation	-	+	-	++

Table 4: Analysis of building blocks after the bootstrap. The performances of building blocks differ from Table 3, e.g. blocks capturing model characteristics show a stronger potential for the later phases of the labeling process (Class Prediction, Class Separation). Data characteristics are still beneficial for the labeling performance.

data sets. First, data-based strategies (Centroids First, Dense Areas First, Ideal Labels First) show particularly good performances in the very early phase of the process. Furthermore, Ideal Labels First manages to visit every available class label considerably earlier than Random and AL. We infer that data-based strategies are a valuable means to tackle the bootstrap problem. However, in the unbalanced FRAUD dataset these strategies failed; we assume that the strategies predominantly selected instances of the dominating class while neglecting the smaller class leading to a weak performance. Second, most model-based strategies approximately perform at the pace of AL and Random in early iterations. Model-based strategies seem to be less appropriate to address the bootstrap problem. Third, labeling outliers or border regions (Outliers First, Cluster Borders First, Class Outliers) cannot compete with Random and AL in most cases and seem to be unsuitable for very early labeling passes.

The result is also reflected in Figure 1 (MNIST). For data-based strategies, we observe a steep increase from the start, which flattens out after about 20 iterations. AL strategies start to outperform data-based strategies beginning with the 35th iteration. Surprisingly, before this point Centroids First, Dense Areas First, and Ideal Labels First strongly outperform AL strategies. Interestingly, Ideal Labels First is the best strategy to visit each class as soon as possible (see, e.g., the boxplot in Figure 1). This raises the question how future VIAL approaches may provide good overviews of the candidates data to allow labeling ideal labels at first. In this connection, the notion of “ideal” may also be subject to semantical meanings or subjective judgment.

Implications of Building Blocks We reflect on the results of the user strategies and bring the analysis to the granularity of building blocks (RQ₃). A compact overview of the building-blocks’ performances is presented in Table 3. The analysis of the bootstrap performance reveals the following insights. First, data-based building blocks perform particularly well. Second, cluster and density-based support is more valuable than outlier detection. Third, model-based building blocks can compete with data-based building blocks in the outlier-prone FRAUD data set.

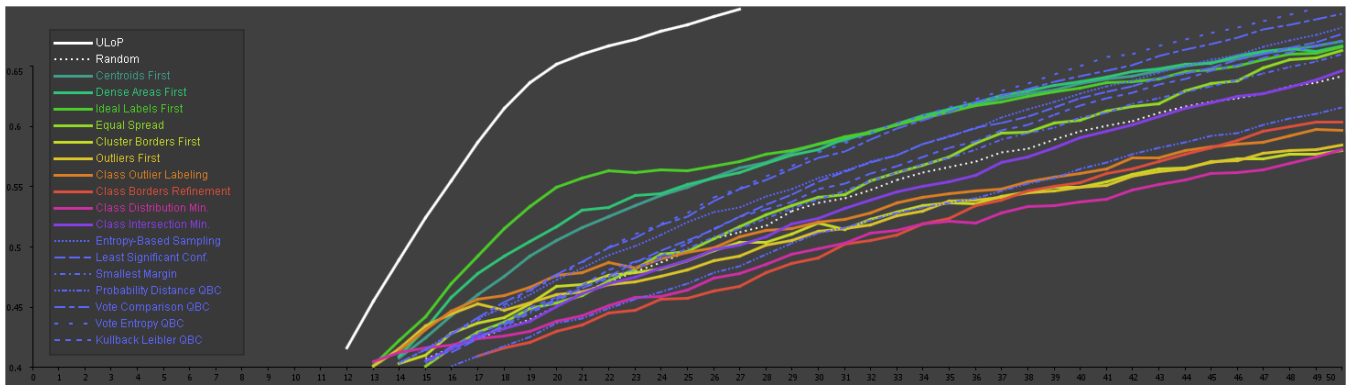


Figure 2: Average performance of strategies after the initialization with one labeled instance per class (MNIST data set). The ULoP still outperforms remaining strategies significantly. Three data-based user strategies (Centroid First, Dense Areas First, Ideal Labels First) perform considerably good at start. AL strategies start at a moderate level, but achieve particularly good performances in later phases. Using the aforementioned data-based user strategies and the AL strategies, we assess a break-even point in the performance at the 35th iteration. Class Intersection is at random level, remaining model-based strategies perform below Random. In general, data-based strategies with a focus on dense areas perform particularly well.

4.4.2. RQ₂: Performance Comparison after Bootstrap

We investigate the performance of all strategies in the phase of the labeling process after bootstrapping is solved. For this purpose the bootstrap problem is resolved in advance with an initial set of instances including one training label per class. Thus, all strategies start with the same representatives. One core question is how model-based strategies that intrinsically suffer from the bootstrap problem will perform in this later phase, compared to the remaining strategies. Again, we evaluate all strategies on all data sets.

User Strategies Figure 2 shows the result of the MNIST data set, Figure 3 provides insight about the GENDER voice data set. Figures of the results for the two remaining data sets are provided in the supplemental material.

We identify three generalizable aspects. First, some data-based strategies still tend to outperform the performance of model-based and AL strategies at the beginning. However, the performance of data-based strategies heavily depends on the characteristics of the data and the labeling task. AL strategies are more robust than data-based strategies and start to surpass the performance of the latter in the course of the labeling process (MNIST: 35th iteration, IRIS 5th iteration, GENDER voice: 14th iteration). The in-depth investigation of identifying sweet spots where labeling should automatically switch to AL strategies may be an interesting subject to future research. Second, the performance of Class Intersection Minimization is the best-performing model-based strategy in all four data sets. Still, Class Intersection Minimization does not outperform most AL strategies. Third, we observe that model-based strategies are more robust to outliers.

We make a specific finding when observing the very small IRIS data set: the performance of Centroids First and Dense Areas First decreases considerably after 20 iterations. We assume that small data sets do not require labeling many centroid-like instances. It is even possible that there are no centroid-like instances left in the candidate set. Figure 2 confirms another insight: instances being outliers or located in border regions contribute less to the per-

formance. We hypothesize that these strategies may have their strengths in very late phases of the labeling process when robust class boundaries have been found which just have to be further refined. We further observe that outliers in the GENDER voice data set strongly degrade the performance of Centroids First. Another problem observed for data-based strategies is caused by unbalanced label distributions. The comparatively weak performance may originate from clustering, i.e., both clusters (in case of a binary classification problem) may be located in regions of the same, dominating class (observed for the FRAUD data set, depicted in the supplemental material).

Implications of Building Blocks Again we analyze the individual building blocks (RQ₃) and observe interesting patterns in the results, see Table 4. *First*, algorithms focusing on data characteristics (Nearest Spatial Neighbors, Spatial Balancing, Clustering, Density Estimation) are in particular valuable for this particular phase of the labeling process. *Second*, the analysis of outliers seems less relevant in this phase. *Third*, model-based building blocks indicate their potentials and may be more tightly coupled with building blocks used in AL. From a data set perspective, we identify problems for GENDER voice and FRAUD. Clustering in particular suffered from outliers and class imbalance.

5. Discussion and Future Work

Our experiments provide an in-depth analysis of different user-based candidate selection strategies. The long-term goal of our work is twofold. First, the development of better AL algorithms that also include user-centered aspects. Second, the use of these algorithms for better guidance of users in future visual-interactive labeling systems. During our experiments, we identified a number of issues relevant to the realization of our goal and which we discuss in the following.

From User Strategies to Parameterizable Building Blocks. The implementation of the identified building blocks requires the specification of a number of parameters and choices to be made,

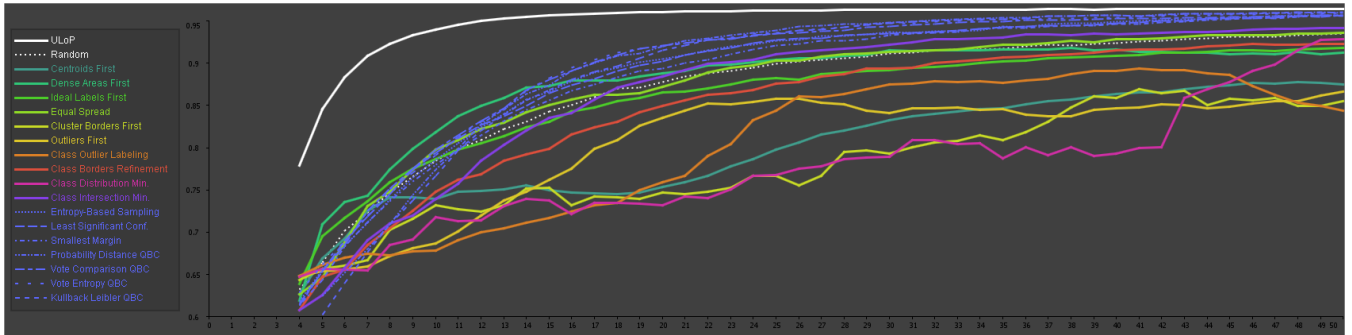


Figure 3: Average performance of strategies after the initialization with one labeled instance per class (GENDER VOICE). ULoP shows the best performance by far. Dense Areas First, Ideal Labels First, and Equal Spread start strong, but get outperformed by AL strategies between the 10th and 20th iteration. Class Intersection Minimization is the best model-based strategy that almost competes with AL. The remaining model-based strategies perform below Random. Centroid First shows particularly weak performance, potentially suffering from outliers.

such as the selection of k in K-NN or the selection of a suitable clustering algorithm. In addition, we also want to note that, beyond the 11 building blocks, other building blocks might exist to formalize other strategies. While the formalization of user strategies may reduce the amount of interaction needed from the user, it opens up large design space for the implementation of the strategies and thereby shifts the users' responsibility from selecting instances rather to selecting parameters for strategy automation. The automatic selection of appropriate parameters is an interesting direction of future research. In terms of the VA principle our proposed formalization opens up new possibilities. A better understanding of user strategies allows for the development of better visual guidance methods. Furthermore, it may become possible to automate the candidate selection process as a whole.

Dependency on Characteristics of Data Sets. We used four different data sets for the assessment of strategies, varying in cardinality, size, class balance, and outlier rate. The performance differences between the user strategies depend on these factors. Data-based user strategies (Centroid First, Dense Areas First, and Ideal Labels) performed well for MNIST which is large, pattern-rich (10 classes), and cleansed. The potential of these strategies decreased quickly for the small IRIS data set. Additionally, data-based strategies (especially Centroids First) had problems in the presence of outliers (GENDER voice and FRAUD detection). In contrast, model-based strategies with their potentials in later phases of the labeling process seemed to be more robust against outliers. However, model-based strategies had problems with data sets with many classes, which requires further research. In general, taking local data characteristics into account (particularly density estimation, as well as class prediction) can help to support the labeling process. A challenge that remains for all strategies are very large datasets, which are difficult to tackle by VIAL approaches. For such datasets, on the one hand classifiers are required with a large capacity to enable the modeling of the large-scale data. On the other hand, user strategies may be useful in this context to accelerate the labeling process by finding better candidate instances for labeling in less time making the overall labeling task more efficient.

Comparison with AL Strategies. Data-based user strategies demonstrated their potential in early phases of the process, e.g.,

to support AL strategies suffering from the bootstrap problem. The experiments also showed that AL strategies become increasingly strong in the course of the labeling process. This raises questions about the sweet spot between these two classes of algorithms, which poses an interesting subject to future work. Figure 2 demonstrates such a sweet spot visually, at iteration 35 when AL strategies start to outperform the user strategies. We confirm the superior performance of AL to model-based user strategies in average. Given the potentials provided with the underlying building blocks (cf. Tables 3 and 4), the improvement of model-based strategies in combination with AL strategies poses another interesting future direction. Leverage points that may foster the performance are using the Class Likelihood building block to incorporate label uncertainty and Equal Spread to better respond to data characteristics.

Complexity of User Actions. With the variety of the four data sets, we observed that many formalized user strategies have strengths, but also weaknesses. Considering that the performance of users in a recent experiment [BZSA17] was as good as AL or better, we conclude that users often applied more than one particular strategy during the labeling process. More formally, we assume that no single formalized user strategy can replace real users in the VIAL process. This implies several aspects for future investigation. First, research in the observation of users may reveal other strategies that can be formalized. Second, the observation of changes in user strategies in the course of the labeling process may yield new insights and potentials. Finally, the combination of existing user strategies may improve the performance and come closer towards the ULoP.

Relations between Strategies. A similar aspect is the analysis of relations and complementing strengths between strategies as well as between building blocks. Tighter coupling of AL strategies with user strategies may lead to novel powerful selection strategies. Another dimension in the analysis is the temporal domain of the labeling process itself. Which strategy is best suited at a given phase in the labeling process? Further insights are necessary to select, combine or detach strategies for different points in time in the labeling process. The analysis of the very early phase (bootstrap phase) and the later phase brings a first evidence that the series of user strate-

gies and building blocks require individual weighting in the course of the labeling process.

Exploratory Interactive Labeling. Our experiments are based on data sets with a known number of classes, known class cardinalities (balance), and we only observed instances which, by no doubt, were assignable to a single true label. In our experiment, we identified that knowledge about the label alphabet and the expected distribution of labels in the data are key criteria for effective labeling. For real world cases the ability to formalize a given labeling task becomes more difficult. We expect that VIAL combining visual-interactive interfaces with AL and ML-support are a beneficial means to address such complex real-world learning tasks. In such situations, we assume that users in the loop are, and remain, indispensable to conduct effective and efficient labeling.

6. Conclusions

We presented the first formalization of user strategies for selecting instances in the labeling process. The formalization of these 10 high-level user strategies builds upon 11 low-level building blocks resembling techniques from data mining, machine learning, statistics, and information retrieval. We assessed their performance with four data sets varying in size, cardinality, label balance, and degree of outliers. The results showed that data-centered user strategies work considerably well in early phases of the labeling process, while model-based strategies (and active learning strategies) in this phase suffer from the bootstrap problem. Furthermore, cluster- and density-based algorithms outperformed outlier-based strategies for early candidate selection. Model-based user strategies performed better during later phases of the labeling process, possibly indicating room for further improvement. In general, we conclude that 1) no single strategy is consistently outperforming others for the different data sets we used, 2) formalized user strategies partly outperform active learning strategies, and 3) formalized user strategies may not replace real users selecting instances.

Follow-up research may include the analysis of relations between strategies and the combination of user strategies to leverage complementary strengths. We believe that both the machine learning and the visualization/visual analytics community can benefit from the proposed approach of formalizing user strategies. On the one hand, our approach might help to shift the development towards user-centered active learning approaches, which leverage user knowledge. On the other hand, it might foster designing better user-guidance concepts for visual interactive labeling approaches.

References

[AP11] ATTENBERG J., PROVOST F.: Inactive learning?: Difficulties employing active learning in practice. *SIGKDD Explor. Newsl.* 12, 2 (Mar. 2011), 36–41. doi:10.1145/1964897.1964906. 2

[BDV*17] BERNARD J., DOBERMANN E., VÖGELE A., KRÜGER B., KOHLHAMMER J., FELLNER D.: Visual-interactive semi-supervised labeling of human motion capture data. In *Visualization and Data Analysis (VDA)* (2017). doi:https://doi.org/10.2352/ISSN.2470-1173.2017.1.VDA-387. 2, 3

[Bec16] BECKER K.: Gender recognition by voice – identify a voice as male or female, 2016. Accessed: 2017-12-05. 7

[BHZ*17] BERNARD J., HUTTER M., ZEPPELZAUER M., FELLNER D., SEDLMAIR M.: Comparing visual-interactive labeling with active

learning: An experimental study. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* (2017). 1, 2, 3, 4

[BKSS14] BEHRISCH M., KORKMAZ F., SHAO L., SCHRECK T.: Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *IEEE Visual Analytics Science and Technology (VAST)* (2014), pp. 43–52. 2

[BLBC12] BROWN E. T., LIU J., BRODLEY C. E., CHANG R.: Dysfunction: Learning distance functions interactively. In *IEEE Visual Analytics Science and Technology (VAST)* (2012), IEEE, pp. 83–92. 2

[Bre01] BREIMAN L.: Random forests. *Machine Learning* 45, 1 (2001), 5–32. 7

[BRS*17] BERNARD J., RITTER C., SESSLER D., ZEPPELZAUER M., KOHLHAMMER J., FELLNER D.: Visual-interactive similarity search for complex objects by example of soccer player analysis. In *IVAPP, VIS-GRAPP* (2017), pp. 75–87. doi:10.5220/0006116400750087. 2, 3

[BSB*15] BERNARD J., SESSLER D., BANNACH A., MAY T., KOHLHAMMER J.: A visual active learning system for the assessment of patient well-being in prostate cancer research. In *VIS Workshop on Visual Analytics in Healthcare* (2015), ACM, pp. 1–8. doi:10.1145/2836034.2836035. 2, 3

[BSR*14] BERNARD J., SESSLER D., RUPPERT T., DAVEY J., KUIJPER A., KOHLHAMMER J.: User-based visual-interactive similarity definition for mixed data objects-concept and first implementation. *Journal of WSCG* 22 (2014). 2, 3

[BZA17] BERNARD J., ZEPPELZAUER M., SEDLMAIR M., AIGNER W.: Vial – a unified process for visual-interactive labeling. *The Visual Computer (TVCJ)* TVCJ-D-17-00413 (2017). 1, 2, 10

[Caw11] CAWLEY G.: Baseline methods for active learning. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010* (Sardinia, Italy, 16 May 2011), Guyon I., Cawley G., Dror G., Lemaire V., Statnikov A., (Eds.), vol. 16 of *Proceedings of Machine Learning Research*, PMLR, pp. 47–57. URL: <http://proceedings.mlr.press/v16/cawley11a.html>. 6

[CM05] CULOTTA A., MCCALLUM A.: Reducing labeling effort for structured prediction tasks. In *Conference on Artificial Intelligence (AAAI)* (2005), AAAI Press, pp. 746–751. 6

[CV95] CORTES C., VAPNIK V.: Support-vector networks. *Machine Learning* 20, 3 (1995), 273–297. 7

[DB79] DAVIES D. L., BOULDIN D. W.: A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 1, 2 (Feb. 1979), 224–227. doi:10.1109/TPAMI.1979.4766909. 5

[DE95] DAGAN I., ENGELSON S. P.: Committee-based sampling for training probabilistic classifiers. In *International Conference on Machine Learning (ICML)* (1995), Morgan Kaufmann, pp. 150–157. 6

[DHS*73] DUDA R. O., HART P. E., STORK D. G., ET AL.: *Pattern classification*, vol. 2. Wiley New York, 1973. 7

[DRH06] DAGLI C. K., RAJARAM S., HUANG T. S.: Leveraging active learning for relevance feedback using an information theoretic diversity measure. In *Conference on Image and Video Retrieval* (2006), Springer, pp. 123–132. doi:10.1007/11788034_13. 2

[Dun74] DUNN J. C.: Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics* 4, 1 (1974), 95–104. doi:10.1080/01969727408546059. 5

[FHOM09] FERRI C., HERNÁNDEZ-ORALLO J., MODROIU R.: An experimental comparison of performance measures for classification. *Pattern Recognition Letters* 30, 1 (2009), 27–38. doi:10.1016/j.patrec.2008.08.010. 7

[HJL06] HOI S. C., JIN R., LYU M. R.: Large-scale text categorization by batch mode active learning. In *World Wide Web* (2006), ACM, pp. 633–642. doi:10.1145/1135777.1135870. 2

[HKBE12] HEIMERL F., KOCH S., BOSCH H., ERTL T.: Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 18, 12 (2012), 2839–2848. 3

- [HMdCM17] HUANG L., MATWIN S., DE CARVALHO E. J., MINGHIM R.: Active learning with visualization for text data. In *ACM WS on Exploratory Search and Interactive Data Analytics (ESIDA)* (2017), ACM, pp. 69–74. doi:10.1145/3038462.3038469. 2
- [HNH*12] HÖFERLIN B., NETZEL R., HÖFERLIN M., WEISKOPF D., HEIDEMANN G.: Inter-active learning of ad-hoc classifiers for video visual analytics. In *IEEE Visual Analytics Science and Technology (VAST)* (2012), IEEE, pp. 23–32. doi:10.1109/VAST.2012.6400492. 2, 3
- [HSW89] HORNIK K., STINCHCOMBE M., WHITE H.: Multilayer feed-forward networks are universal approximators. *Neural Networks* 2, 5 (1989), 359–366. doi:10.1016/0893-6080(89)90020-8. 7
- [Jai10] JAIN A. K.: Data clustering: 50 years beyond k-means. *Pattern Recognition Letters* 31, 8 (2010), 651–666. 5
- [JPP09] JOSHI A. J., PORIKLI F., PAPANIKOLOPOULOS N.: Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition (CVPR)* (2009), IEEE, pp. 2372–2379. 2
- [KCH*17] KOTTKE D., CALMA A., HUSELJIC D., KREMPL G., SICK B.: Challenges of reliable, realistic and comparable active learning evaluation. In *Workshop and Tutorial on Interactive Adaptive Learning (IAL), co-located with ECML PKDD* (2017), pp. 2–14. 8
- [KN98] KNORR E. M., NG R. T.: Algorithms for mining distance-based outliers in large datasets. In *Conference on Very Large Data Bases (VLDB)* (San Francisco, CA, USA, 1998), Morgan Kaufmann Publishers Inc., pp. 392–403. 5
- [KSF17] KONYUSHKOVA K., SZNITMAN R., FUA P.: Learning active learning from data. In *Advances in Neural Information Processing Systems* (2017), pp. 4226–4236. 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. 7
- [Lic13] LICHMAN M.: UCI machine learning repository, 2013. URL: <http://archive.ics.uci.edu/ml/>. 7
- [Mam98] MAMITSUKA N. A. H.: Query learning strategies using boosting and bagging. In *International Conference on Machine Learning (ICML)* (1998), vol. 1, Morgan Kaufmann Pub. 2, 6
- [MFNP13] MAMANI G. M. H., FATORE F. M., NONATO L. G., PAULOVICH F. V.: User-driven feature space transformation. *Computer Graphics Forum (CGF)* 32, 3 (2013), 291–299. doi:10.1111/cgf.12116. 2
- [MM04] MELVILLE P., MOONEY R. J.: Diverse ensembles for active learning. In *International Conference on Machine Learning (ICML)* (Banff, Canada, July 2004), ACM, pp. 584–591. doi:10.1145/1015330.1015385. 2
- [MN98] MCCALLUM A., NIGAM K.: Employing em and pool-based active learning for text classification. In *International Conference on Machine Learning (ICML)* (San Francisco, CA, USA, 1998), Morgan Kaufmann Pub., pp. 350–358. 6, 8
- [MPG*14] MÜHLBACHER T., PIRINGER H., GRATZL S., SEDLMAIR M., STREIT M.: Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 20, 12 (2014), 1643–1652. doi:10.1109/TVCG.2014.2346578. 6
- [Ols09] OLSSON F.: *A literature survey of active machine learning in the context of natural language processing*. Tech. rep., Swedish Institute of Computer Science, 2009. 2
- [PCJB15] POZZOLO A. D., CAELEN O., JOHNSON R. A., BONTEMPI G.: Calibrating probability with undersampling for unbalanced classification. In *Computational Intelligence and Data Mining (CIDM)* (2015), IEEE, pp. 159–166. doi:10.1109/SSCI.2015.33. 7
- [QHR*09] QI G.-J., HUA X.-S., RUI Y., TANG J., ZHANG H.-J.: Two-dimensional multilabel active learning with an efficient online adaptation model for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 31, 10 (2009), 1880–1897. doi:10.1109/TPAMI.2008.218. 2
- [Rou87] ROUSSEUW P.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20, 1 (Nov. 1987), 53–65. doi:10.1016/0377-0427(87)90125-7. 5, 6
- [SA15] SEDLMAIR M., AUPETIT M.: Data-driven evaluation of visual quality measures. *Computer Graphics Forum (CGF)* 34, 3 (2015), 201–210. doi:10.1111/cgf.12632. 2
- [SC08] SETTLES B., CRAVEN M.: An analysis of active learning strategies for sequence labeling tasks. In *Empirical Methods in Natural Language Processing* (2008), Computational Linguistics, pp. 1070–1079. 6
- [SCR08] SETTLES B., CRAVEN M., RAY S.: Multiple-instance active learning. In *Advances in neural information processing systems* (2008), pp. 1289–1296. 2
- [SDW01] SCHEFFER T., DECOMAIN C., WROBEL S.: Active hidden markov models for information extraction. In *Conference on Advances in Intelligent Data Analysis (IDA)* (London, UK, UK, 2001), Springer-Verlag, pp. 309–318. doi:10.1007/3-540-44816-0_31. 2, 6
- [Set09] SETTLES B.: *Active Learning Literature Survey*. Tech. Report 1648, Univ. of Wisconsin–Madison, 2009. 2
- [Set12] SETTLES B.: *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012. doi:10.2200/S00429ED1V01Y201207AIM018. 6
- [SG10] SEIFERT C., GRANITZER M.: User-based active learning. In *IEEE International Conference on Data Mining Workshop* (2010), pp. 418–425. doi:10.1109/ICDMW.2010.181. 3
- [Sha48] SHANNON C.: A mathematical theory of communication. *Bell system technical journal* 27 (1948). 6
- [SOS92] SEUNG H. S., OPPER M., SOMPOLINSKY H.: Query by committee. In *Worksh. on Comput. Learning Theory (COLT)* (New York, NY, USA, 1992), ACM, pp. 287–294. doi:10.1145/130385.130417. 2
- [SSBJ16] SARKAR A., SPOTT M., BLACKWELL A. F., JAMNIK M.: Visual discovery and model-driven explanation of time series patterns. In *Visual Languages and Human-Centric Computing (VL/HCC)* (2016), IEEE, pp. 78–86. doi:10.1109/VLHCC.2016.7739668. 3
- [SSJK16] SEEBACHER D., STEIN M., JANETZKO H., KEIM D. A.: Patent Retrieval: A Multi-Modal Visual Analytics Approach. In *EuroVis Workshop on Visual Analytics (EuroVA)* (2016), Eurographics, pp. 013–017. 2
- [SWY75] SALTON G., WONG A., YANG C. S.: A vector space model for automatic indexing. *Communications of the ACM* 18, 11 (Nov. 1975), 613–620. doi:10.1145/361219.361220. 7
- [TVC*11] TUIA D., VOLPI M., COPA L., KANEVSKI M., MUNOZ-MARI J.: A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing* 5, 3 (2011), 606–617. 2
- [VBF12] VEZHNEVETS A., BUHMANN J. M., FERRARI V.: Active learning for semantic segmentation with expected change. In *Computer Vision and Pattern Recognition (CVPR)* (2012), IEEE, pp. 3162–3169. 2
- [VPS*02] VENDRIG J., PATRAS I., SNOEK C., WORRING M., DEN HARTOG J., RAAIJMAKERS S., VAN REST J., VAN LEEUWEN D. A.: Trec feature extraction by active learning. In *TREC* (2002). 2
- [WH11] WANG M., HUA X.-S.: Active learning in multimedia annotation and retrieval: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 2 (2011), 10:1–10:21. doi:10.1145/1899412.1899414. 2
- [WKBD06] WU Y., KOZINTSEV I., BOUGUET J.-Y., DULONG C.: Sampling strategies for active learning in personal photo retrieval. In *IEEE Int. Conference on Multimedia and Expo* (2006), IEEE, pp. 529–532. doi:10.1109/ICME.2006.262442. 2, 6