Georg Krempl Vincent Lemaire
Edwin Lughofer Daniel Kottke (Eds.)

AL@iKNOW 2016

Workshop on Active Learning: Applications, Foundations and Emerging Trends

International Conference on Knowledge Technologies and Data-driven Business (i-KNOW)

Graz, Austria, October 18, 2016

Proceedings

Preface

Active Learning addresses the intersection between Data Mining/Machine Learning and interaction with humans. Aiming at optimizing this interaction, it bridges the gap between data-centric and user-centric approaches. For example, by requesting the most relevant information or performing the most informative experiment. Facing big volumes of data but limited human annotation and supervision capacities, active approaches become increasingly important for improving the efficiency in interactions.

Active learning is a very useful methodology in on-line industrial applications for reducing efforts for sample annotation and measurements of "target" values (e.g., quality criteria). It further reduces the computation speed of machine learning and data mining tools, as embedded models are only updated based on a subset of samples selected by the implemented active learning technique. This is especially important when performing modeling and mining cycles from on-line data streams, where real-time demands to model updates and inference processing are quite usual.

Various approaches, application scenarios and deployment protocols have been proposed for active learning. However, despite the efforts made from academia and industry researchers alike, there are still gaps between research on theoretical and practical aspects. When designing active learning algorithms for real-world data, some specific issues are raised. The main ones are scalability and practicability. Methods must be able to handle high volumes of data, in spaces of possibly high-dimension, and the process for labeling new examples by an expert must be optimized.

All in all, we accepted five regular papers (5 papers submitted) and one tutorial to be published in these workshop proceedings. The authors discuss approaches, identify challenges and gaps between active learning research and meaningful applications, as well as define new application-relevant research directions.

We thank the authors for their submissions and the program committee for their hard work.

October 2016

Georg Krempl, Vincent Lemaire,
Edwin Lughofer, Daniel Kottke

Organizing Committee

Georg Krempl, University Magdeburg
Vincent Lemaire, Orange Labs France
Edwin Lughofer, University Linz
Daniel Kottke, University Kassel

Program Committee

Niall Adams, Imperial College
Alexis Bondu, AXA France
Hugo Jair Escalente, National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico
Ernesto De Luca, Leibniz Institut Schulbuchforschung
Jose Garcia, University of Alicante, Spain
Michael Granitzer, University Passau
Dino Ienco, IRSTEA
Franz Pernkopf, TU Graz
Kurt Pichler, Linz Center of Mechatronics GmbH
Bernhard Sick, University of Kassel
Christin Seifert, University Passau
Myra Spiliopoulou, University Magdeburg
Jurek Stefanowski, TU Poznan
Sebastian Tschiatschek, ETH Zurich
Martin Znidarsic, Jozef Stefan Institute

Contents

Active Learning: Applications, Foundations and Emerging Trends (Tutorial)

Georg Krempl
University Magdeburg, Germany
georg.krempl@ovgu.de

Vincent Lemaire
Orange Labs, France
vincent.lemaire@orange.com

Edwin Lughofer
University Linz, Austria
edwin.lughofer@jku.at

Daniel Kottke
University Kassel, Germany
daniel.kottke@uni-kassel.de

1 Introduction

Active learning optimizes the interaction between artificial data mining/machine learning systems and humans. For example, its techniques are used for selecting the most relevant information to be requested, or for determining the most informative experiment to be performed. With increasing volumes of data, contrasting limited human supervision capacities, the optimization of this interaction has become even more important. Hence, active sampling and data acquisition techniques could contribute to the design and modeling of highly intelligent learning systems.

This tutorial presents the basic techniques for pool-based and on-line active learning for streams. It contains a summary of the common concepts like version space partitioning, uncertainty sampling and decision theoretic approaches, and shortly mentions the connection between reinforcement learning and active learning. We show how these concepts can be used in data streams and on-line applications, and discuss the main challenges of stream active learning. Finally, we evaluate frameworks for pool-based and stream-based active learning to validate if a method is applicable for a specific demand.

2 Basic Strategies in Active Learning

Adaptive sampling [Sin06] and selective sampling [Roy01] are the two main scenarios to set active learning [Set12]. This tutorial focus on selective sampling. After defining the active learning problem in detail [Set12], we discuss three basic techniques: The selection criterion of methods using *version space partitioning*, is based on the disagreement between hypothesis [Ruf89], e.g., the disagreement within an ensemble of classifiers. *Uncertainty sampling* is an information theoretic approach, selecting instances based on the classifier's uncertainty, e.g., instances near the decision boundary. Decision theoretic approaches are among others *expected error reduction*, which simulate each label outcome for each label candidate, or *probabilistic active learning* [Kre14]. The latter uses local statistics of each candidate to estimate its gain in performance.

3 Reinforcement Learning and Active Learning

(Online) Reinforcement learning tries to improve itself by interacting with the environment while active learning tries to solve the traditional supervised learning problem with a human in the loop. Seeing the human as a part of the environment, as an agent [Kap05], it/he has to find a policy that maps states of the world to the actions to be taken in those states. When the agent has to learn the action to pick the next label candidate, one way to do this is to consider the informativeness of this sample as a form of reward and the current set of labeled and unlabeled data as the observation [Set08].

In: G. Krempl, V. Lemaire, E. Lughofer, and D. Kottke (eds.): Proceedings of the Workshop Active Learning: Applications, Foundations and Emerging Trends, AL@iKNOW 2016, Graz, Austria, 18-OCT-2016, published at http://ceur-ws.org

4 On-line Active Learning for Streams

In this section, we provide an overview on *on-line* active learning concepts for data streams. These algorithms use different reliability concepts for instance selection, e.g., *conflict* and *ignorance* [Lug12]. We point out special challenges for on-line algorithms and stream active learning and discuss possible solutions. Thereby, a specific emphasis will be placed on active learning in connection with *evolving (fuzzy) systems* [Lug16], which are able to expand their structure on the fly to properly react on ignorance cases and thus to reduce uncertainty in the version space. A collection of some successful practical application examples, e.g., from quality control systems, viscose production, or image classification systems, gives an idea of how to use these methods in practice.

5 Evaluation of Active Learning

Comparing active learning methods is challenging, as the results of common evaluation methodologies like 5- or 10-fold cross-validation are often not reliable and therefore insufficient. Hence, we propose to use an evaluation framework using multiple randomly (seed-based) generated train and evaluation sets (typically 50 or more) [Kre14], which also allow pairwise comparisons. The second part of this topic concentrates on stream evaluation and shows why a separation of the temporal and spatial component of a stream active learner is mandatory in order to compare two active approaches [Kot15]. Furthermore, we discuss different forms of visualization like learning curves (performance vs. budget) and performance curves (performance vs. time) and show how the definition of the target function can affect the evaluation.

References

[Kap05] "Reinforcement learning for active model selection." A. Kapoor and R. Greiner. ACM SIGKDD Workshop on Utility-based Data Mining, 2005

[Kot15] "Probabilistic Active Learning in Datastreams." Daniel Kottke, Georg Krempl, Myra Spiliopoulou. International Symposium on Intelligent Data Analysis. Springer International Publishing, 2015.

[Kre14] "Optimised probabilistic active learning (OPAL)." Georg Krempl, Daniel Kottke, Vincent Lemaire Machine Learning 100.2-3 (2015): 449-476.

[Lug12] "Single-pass active learning with conflict and ignorance." Edwin Lughofer. Evolving Systems 3 (4), 251-271, 2012.

[Lug16] "Evolving Fuzzy Systems — Fundamentals, Reliability, Interpretability and Useability." Edwin Lughofer. in: Handbook of Computational Intelligence, World Scientific, pp. 67-135, 2016.

[Roy01] "Toward optimal active learning through sampling estimation of error reduction" Roy, N., McCallum, A. In: Proc. 18th International Conference on Machine Learning, pp. 441-448, 2001

[Ruf89] "What good are experiments?" Ritchey A. Ruff, T. G. Dietterich. Proceedings of the 6th International Workshop on Machine Learning, 1989.

[Set08] "An Analysis of Active Learning Strategies for Sequence Labeling Tasks." Burr Settles, Mark Craven Empirical Methods on Natural Language Processing (EMNLP) 2008

[Set12] "Active Learning." Burr Settles. Number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, (2012).

[Sin06] "Active learning for adaptive mobile sensing networks" Singh, A., Nowak, R., Ramanathan, P. in International conference on Information processing in sensor networks, pp. 60–68, 2006

MapView: Graphical Data Representation for Active Learning

Eva Weigl[1] Alexander Walch[1] Ulrich Neissl[1] Pauline Meyer-Heye[1]
Wolfgang Heidl[1] Thomas Radauer[2] Edwin Lughofer[3] Christian Eitzinger[1]

[1]Profactor GmbH, Steyr-Gleink, Austria
[2]STRATEC Consumables GmbH, Anif, Austria
[3]Johannes Kepler University of Linz, Austria

Abstract

Active learning facilitates the training of classifiers by selectively querying the user in order to gain insights on unlabeled data samples. Until recently, the user had limited abilities to interact with an active learning system: A sub-selection was presented by the system and every sample within had to be annotated. We propose an alternative and graphical solution to active learning called *MapView* where the user may profit from a different interpretation of the underlying data. Experiments underline the usability and advantages of our approach during the training of a classifier from scratch.

Keywords: active learning; graphical data representation; classification; random forests

1 Introduction and Motivation

Active learning is an ongoing research field in the machine learning environment. It enriches and facilitates the training of classifiers by getting rid of enormous amounts of a priori trained samples. This means, instead of forcing a user to manually annotate a large number of samples, the learner pre-selects a subset of *interesting* samples via an algorithm [13]. By presenting these samples to the user, the classifier is able to learn quicker and even explore interpolation or extrapolation areas in the feature space. This avoids tedious annotation times and costs. There are different strategies to select those *interesting* samples [12], with *uncertainty sampling* being the most prominent one which selects the least certain samples according to the currently valid model [13]. Typically, the user then annotates the selected samples, and afterwards the classifier is re-trained or updated based on the newly gained information.

In this paper, we present an alternative approach to user interaction during classifier training within the scope of active learning. Instead of showing the user the sub-set of interesting samples and forcing him/her to annotate all of them, we propose a graphical representation of the data where interesting samples are highlighted within the whole data range. In this way, the user sees all data *at a glance* which results in advantages such as easy identification of clusters as well as outliers.

In the area of *information visualization*, the authors of [1] presented a promising tool for the visualization of probabilistic classification data. A similar technique is presented in [6] or [14]. [2] demonstrate views of clustered graph data. [11] presented a user-based active learning scenario where instead of using classical selection strategies, the user is enabled to choose interesting samples and label them. Our method extends this approach by

In: G. Krempl, V. Lemaire, E. Lughofer, and D. Kottke (eds.): Proceedings of the Workshop Active Learning: Applications, Foundations and Emerging Trends, AL@iKNOW 2016, Graz, Austria, 18-OCT-2016, published at http://ceur-ws.org

providing information on the original features in the graphical visualization as well as confidences and proposed class labels from the classifier. This gives the user a more complete picture of the scenario. Also all samples are presented, allowing relabeling of samples and even adding new classes during training.

Our proposed method is independent from the classifier choice as long as the classifier result contains an additional probability measure about how *certain* the classifier is in its decision. We take advantage of a Random Forest classifier [4] and interpret the trees' votes as probabilities. We chose Random Forests because we start with a small number of pre-labeled samples for active learning where a bagged ensemble approach such as Random Forests is well known to outperform other standard (non-ensembled) classifiers. This is because bagging nicely explores the data space through bootstrapping the samples a multiple times, which is, for instance, more deeply analyzed in [3]. In this way, bagging not only reduces the bias of the classifier, but also its variance.

2 Method

We propose a graphical approach to support the user in training a classifier via active learning. The scenario is multi-class classification with pool-based sampling with uncertainty sampling as query strategy.

The main information displayed in the graphical representation is: current labels, current predicted labels, certainty of the prediction and sample coordinates in feature space. The last providing the user with a feature analysis view. This kind of view is described as enhancing the user's understanding of predictions and trust in the underlying classifier in [1]. To display the multi-dimensional feature coordinates, we need to project them onto a 2-dimensional map. For this purpose, we first take advantage of k-means clustering to find a number of cluster centers among our data (corresponding to the number of classes). These centers are then embedded onto a 2-dimensional map by dimensionality reduction. All remaining samples can then be represented as linear combination of the cluster centers and are added to the map. In the following, we will describe the method in more detail.

As classifier, we select Random Forests (see [4] for more information). However, as mentioned before, any classifier that delivers probability measures for each class label can be selected. In our case, we use the votes of each tree divided by the number of all trees as certainty measure for the sampling of interesting samples: the higher the votes, the more certain the classifier is in its decision. This is similar to a least confidence sampling strategy with aspects of maximum entropy sampling in active learning, where the sample to be labeled by the user, is chosen by the learning algorithm [11]. However, in our approach the user is merely given the information as decision support, thus providing an interactive learning scenario as described in [5].

Within the scope of our classifier, we use the term *event* for any item in our database (this can be e.g., an image). Let $\mathbf{E}_i$ be the i-th event in our database, consisting of $i = 1, ..., N$ samples. Each event $\mathbf{E}_i$ has an F-dimensional feature vector $\mathbf{x}_i = \{x_1, ..., x_F\}$. The goal is to assign one of K class labels to the image. For this purpose, for each event $\mathbf{E}_i$ a class vector $\mathbf{c}_i = \{c_1, ..., c_K\}$ exists, with $c_j \in [0; 1], j = \{1, ..., K\}$ representing the probability of belonging to class j.

Since our aim is to embed the high-dimensional events on a user-friendly 2D map, it is necessary to perform dimensionality reduction. Dimensionality reduction means finding an embedding $e : \mathbb{R}^d \rightarrow \mathbb{R}^c$ that transforms high-dimensional data from its original d-dimensional space into representations with a reduced number of dimensions c (with $c \ll d$) [15]. In our environment, there are two vectors per event which result in two possibilities for the embedding: (i) the feature vector $\mathbf{x}_i$, or (ii) the class vector $\mathbf{c}_i$. The first has the advantage that the resulting map will not change after each classifier training, since the features remain static. Additionally, the resulting map can be used at the very beginning where not a single class label is present in the dataset. On the other side, using the class vector for the embedding better resembles the classifier's view of the data. Nearby located events are regarded as belonging to a similar class, and uncertainties in labeling can quickly be identified (e.g., events lying between two classes). However, the main drawback is that the map will change after every training or incremental update of the classifier. Since our motivation is to facilitate the annotation work for the user, we chose the feature vectors for our embedding in order to avoid confusion over the continuously changing map as it would be in case of the class vectors.

There exist a variety of dimensionality reduction methods (see [15] for a comparative review). For our experiments, we used the Sammon's Mapping (SM) [10] since it attempts to preserve the inherent data structure. The goal is to minimize the following error function

$$E = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}} \tag{1}$$

where d^*_{ij} is the distance between the i-th and j-th event in the original space, and d_{ij} the distance in the embedding. As distance measure the Euclidean distance was selected. The minimization is performed iteratively with a random initialization and a heuristic approach as proposed by [7]. It is a non-linear approach since the embedding is not a linear combination of the original components (as for example, in techniques like the principal component analysis) [8]. In addition, the computational complexity is quadratic (w.r.t. the number of events) which we regarded feasible since the mapping only needs to be computed once as the feature vectors remain static. Additionally, instead of performing the SM on all events from the dataset, we take advantage of k-means clustering as initial step where we compute the locations of K cluster centers (corresponding to the number of K classes). The feature vectors $\mathbf{x_i}$ of all N events of our dataset are then transformed and represented as a linear combination of the K cluster centers $\mathbf{k_j}$ (i.e., as barycentric coordinates):

$$\mathbf{x}_i \approx \sum_{j=1}^{K} \mathbf{k}_j \cdot \lambda_j \tag{2}$$

with $\sum_{j=1}^{K} \lambda_j = 1$. In our application area, the number of cluster centers is typically smaller than the dimension d of the feature vectors. Hence, for each feature vector, we computed a least squares solution of the overdetermined equation system in Eq. 2. In general, the number of cluster centers need to be restricted to $\min(d, K)$ in order to guarantee a unique solution of Eq. 2. By choosing $\lambda_1 = 1 - \sum_{j=2}^{K} \lambda_j$, one can reduce the constraint on the lambdas by solving:

$$\mathbf{x}_i - \mathbf{k}_1 \approx \underbrace{(\mathbf{k}_2 - \mathbf{k}_1, \ldots, \mathbf{k}_K - \mathbf{k}_1)}_{\overline{\mathbf{k}}} \cdot \begin{pmatrix} \lambda_2 \\ \vdots \\ \lambda_K \end{pmatrix} \tag{3}$$

which is equivalent to finding a least squares solution of Eq. 2. Since the matrix $\overline{\mathbf{k}}$ is independent of the feature vectors $\mathbf{x}_i$, its pseudoinverse $\overline{\mathbf{k}}^+$ enables an efficient computation of the barycentric coordinates of all feature vectors:

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_K \end{pmatrix} = \overline{\mathbf{k}}^+ \cdot (\mathbf{x}_i - \mathbf{k}_0) \tag{4}$$

$$\lambda_1 = 1 - \textstyle\sum_{j=2}^{K} \lambda_j \tag{5}$$

After embedding the cluster centers $\mathbf{k}_j$ via SM, the embedded feature vectors $\mathbf{x}^*_i, i = \{1, ..., N\}$ of all events are computed using Eq. 2 by substituting $\mathbf{k}_j$ with the embedded cluster centers $\mathbf{k}^*_j$.

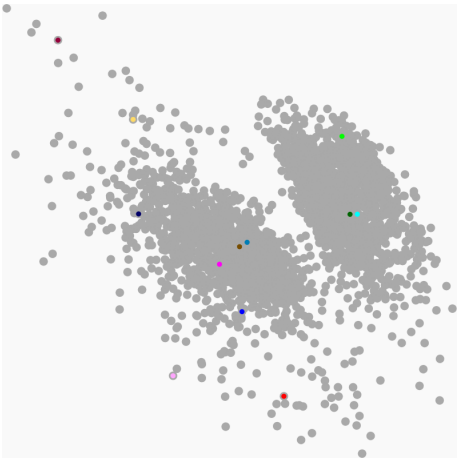
We implemented a simple user interface depicting the proposed MapView of a given dataset (see Fig. 1). In our setting, all events are displayed as points on a 2-dimensional map. Instead of presenting the user the raw view of a pre-selected set of interesting instances (e.g., unlabeled images with pre-computed features), the user gets a graphical overview. Within this view, events are displayed as points with the transparency of the color proportional to the certainty of the classifier – i.e., the more certain a classifier is about classifying a specific event, the less visible (and thus, more transparent) the corresponding point is on the map. This results in a highlighting of interesting samples that is intended to attract the user's attention, while samples with a high classifier certainty are blurred out. We also decided to show both the manually assigned label (area of the circle) as well as the classifier's prediction (circular ring) pretending the label is unknown. The user can interact with the MapView via different ways: (i) zooming in/out and translating, (ii) getting a preview of a selected event (e.g., image thumbnail and features), (iii) assigning a class label to the event (via a pop-up window that shows a preview of the underlying raw image or features), and (iv) applying a (re-)training of the classifier.

3 Experimental results

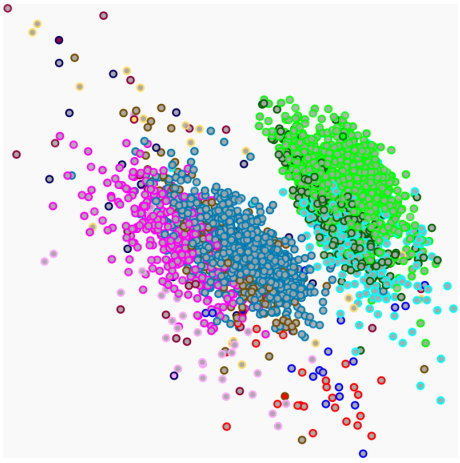
For our experiments, we used an activity recognition dataset containing accelerometer and gyroscope sensor signals collected by waist-mounted smartphones (see [9]). The set consists of 12 classes (representing basic activities and postural transitions of human subjects, such as walking, standing, etc.) with a total of 3162 samples with 561 features each.

We attempted to train a classifier *from scratch* by excluding the class labels for our active learning approach and only using them for labeling the query samples. We started with a random labeling of one sample per

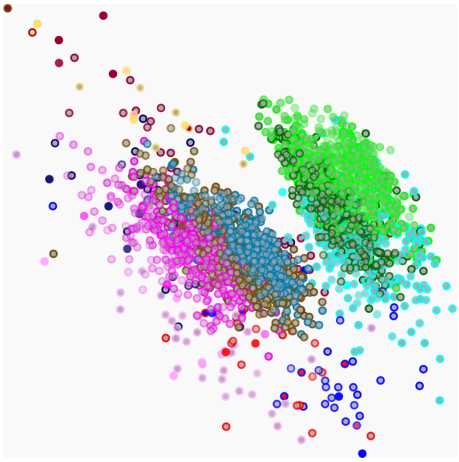
(a) Legend



(b) Initial map



(c) Result after one labeled event per class



(d) Result after 6 labeled events per class

Figure 1: MapView of the dataset. 1(a) Legend of the class labels in the MapView. 1(b) In the beginning, only one event per class is labeled, the remaining events are unlabeled and thus colored gray. 1(c) Result of the classifier training on the initial labeled set (one event per class), and 1(d) after having labeled 5 additional events per class. The more transparent an event occurs, the more confident the classifier is in its decision.

class and a subsequent training of the classifier. This step is depicted in Fig. 1(b) with each color representing a class label (see Fig. 1(a)). The result after training the classifier on these labeled samples is shown in Fig 1(c). Obviously, the certainty of the classifier is very low at this point as nearly all samples are depicted opaque. Afterwards, we labeled five uncertain samples per class and again trained the classifier (the result is in Fig. 1(d)).

The results are easily interpretable and match our assumptions: As one can see, the *active* activities (*walking, walking_down, walking_up*) form a cluster in the lower left region of the map, whereas *passive* activities (*laying, sitting, standing*) are centered on the upper right half of the map, with *laying* being even more separated from the other two classes. Postural transitions (*lie_to_sit, lie_to_stand, sit_to_lie, sit_to_stand, stand_to_lie, stand_to_sit*) are spread in between the other two clusters. The average accuracy as shown in Tab. 1 evaluates the classifier's performance after the initial setup of only one labeled event per class, and after 5 additional samples. We also evaluated the 12 classes by dividing them into passive and active classes as described before, as well as into 3 postural transition classes of the pairs *lie_to_sit/lie_to_stand, sit_to_lie/sit_to_stand, stand_to_lie/stand_to_sit*. Apparently, after labeling only 5 additional events per class, the average accuracy increases up to 60%. When considering the combined groups with only 1 labeled event per (original) class, the accuracy even rises to 97%.

The advantage of our method is that it is faster to calculate the embedding only for the cluster centers instead of all samples of the dataset. In addition, the map also allows a graphical interpretation of the dataset and the classifier's behavior.

Acuracy	1 labeled sample per class	6 labeled samples per class
Per class	0.47	0.60
Per group	0.97	0.98

Table 1: Accuracy of the classification after labeling only one sample per class, and after 5 additionally labeled samples per class. Additionally, we grouped postural transitions as well as active and passive activities and evaluated their average accuracy (last row of table).

4 Conclusion

The proposed method implements a graphical representation of an underlying dataset for classification via active learning. Instead of presenting the user a sub-set of *interesting* samples and requiring their annotation, the samples are depicted as 2D points on a map (called 'MapView') with the color corresponding to the class label. This results in several advantages such as (i) using active learning to avoid the annotation of a large amount of data, (ii) gaining insight into the high-dimensional feature space of the data within the 2D view (e.g., simple identification of cluster samples or outliers), (iii) getting feedback of the classifier's certainty in labeling the samples (samples about which the classifier is uncertain are plotted opaque, whereas the more certain a classifier is, the more transparent the 2D point representation), and (iv) straightforward interpretation of the classifier result and improved understanding of the classifier's decision (e.g., understanding why the classifier made or is uncertain its decision). In the future, we will consider feature selection before the embedding process in order to reduce the high-dimensional space to the most promising features. Additionally, other dimensionality reduction or embedding methods might be worth experimenting with.

Acknowledgements

This work was funded via the projects 'rollerNIL' within the call 'Produktion der Zukunft', and 'MVControl' within the call 'IKT der Zukunft'. Both programs are supplied by the Austrian Research Promotion Agency (FFG), and promoted by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT).

References

[1] Bilal Alsallakh, Allan Hanbury, Helwig Hauser, Silvia Miksch, and Andreas Rauber. Visual methods for analyzing probabilistic classification data. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1703–1712, 2014.

[2] Michael Balzer and Oliver Deussen. Level-of-detail visualization of clustered graph layouts. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, pages 133–140. IEEE, 2007.

[3] P Brazdil, CG Carrier, C Soares, and R Vilalta. *Metalearning: Applications to Data Mining*. Cognitive Technologies. Springer Berlin Heidelberg, 1 edition, 2009.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Benjamin Höferlin, Rudolf Netzel, Markus Höferlin, Daniel Weiskopf, and Gunther Heidemann. Inter-active learning of ad-hoc classifiers for video visual analytics. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 23–32. IEEE, 2012.

[6] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.

[7] T Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer Berlin Heidelberg, 2001.

[8] Boaz Lerner, Hugo Guterman, Mayer Aladjem, I Dinsteint, and Y. Romem. On pattern classification with Sammon's nonlinear mapping an experimental study. *Pattern Recognition*, 31(4):371–381, 1998.

[9] J Reyes-Ortiz, Luca Oneto, A Sama, X Parra, and D Anguita. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing*, 171(1):754–767, 2016.

[10] J.W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.

[11] Christin Seifert and Michael Granitzer. User-Based Active Learning. In *International Conference on Data Mining Workshops*, pages 418–425. IEEE, 2010.

[12] Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin-Madison, Wisconsin, USA, 2010.

[13] Burr Settles. *Active Learning*, volume 18 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool, San Rafael, Calif. (1537 Fourth Street, San Rafael, CA 94901 USA), 2012. OCLC: 799360189.

[14] Alexandru Telea and Ozan Ersoy. Image-Based Edge Bundles: Simplified Visualization of Large Graphs. In *Computer Graphics Forum*, volume 29, pages 843–852. Wiley Online Library, 2010.

[15] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: A comparative Review. Technical Report TiCC-TR-2009-005, Tilburg University, The Netherlands, 2009.

Active Learning with SVM for Land Cover Classification - What Can Go Wrong?

S. Wuttke [1,2,*], W. Middelmann [1], U. Stilla [2]
sebastian.wuttke@iosb.fraunhofer.de
wolfgang.middelmann@iosb.fraunhofer.de
stilla@tum.de

[1] Fraunhofer IOSB, Gutleuthausstr. 1, 76275 Ettlingen, Germany
[2] Technische Universitaet Muenchen, Arcisstr. 21, 80333 Muenchen, Germany
[*] Corresponding author

Abstract

Training machine learning algorithms for land cover classification is labour intensive. Applying active learning strategies tries to alleviate this, but can lead to unexpected results. We demonstrate what can go wrong when uncertainty sampling with an SVM is applied to real world remote sensing data. Possible causes and solutions are suggested.

1 Introduction

The United Nations define: "Land cover is the observed (bio)physical cover on the earth's surface." [DJ00]. It is important to know which land cover class is found in different areas of the earth to make informed political, economical, and social decisions [And76]. In urban planing for example it is important to differentiate between closed and open soil to predict the effects of rainfall. Achieving this at a large scale and high level of detail is impossible without the help of machine learning algorithms. However, these need laborious training which generates high costs in human annotation time and money. Especially in the field of remote sensing, since acquiring ground truth information often involves expensive ground surveys. Therefore only a limited number of training samples can be produced. How to chose which samples should be labelled out of the large amount of unlabelled data samples is the topic of active learning.

There are many approaches for active learning in remote sensing. In general [TRP+09], [TVC+11] as well as with support vector machines (SVMs) [FM04], [BP09]. This paper investigates if the conventional methods can be easily applied to land cover classification on airborne acquired images. Therefore we use a readily available implementation of an SVM and the intuitive uncertainty sampling and query by committee strategies; and apply them to four publicly available real world datasets: Indian Pines, Pavia Centre, Pavia University, and Vaihingen.

The main contributions of this paper are:

- Apply an SVM with uncertainty sampling and query by committee on five real world datasets
- Present the results and discuss possible causes for the underperforming of active learning
- Suggest future actions to alleviate the observed problems

In: G. Krempl, V. Lemaire, E. Lughofer, and D. Kottke (eds.): Proceedings of the Workshop Active Learning: Applications, Foundations and Emerging Trends, AL@iKNOW 2016, Graz, Austria, 18-OCT-2016, published at http://ceur-ws.org

2 Method

The presented method is deliberately kept simple to reduce possible error sources. The results are still expected to demonstrate the advantages of active learning compared to passive learning.

2.1 Pre-Processing

For noise reduction and lowering the amount of data to be processed, segmentation is applied. Here we use the Multi-Resolution-Segmentation algorithm of the eCognition Software [Tri14] with its default parameters. All pixels of a segment are then combined to an average value, which is the new feature. The reasoning behind this is the smoothness assumption [Sch12]. This assumption states that, because of increasing sensor resolution, the probability that two neighbouring pixels belong to the same class, increases. As a result each training sample represents the average spectrum of the materials present in its segment. This step was not applied to the Indian Pines dataset because of its low resolution. Therefore this dataset has an order of magnitude more samples than the others. No further feature extraction was done to keep possible error sources to a minimum. Classes with less then 15 samples were removed to reduce outlier effects.

2.2 Classification Algorithm

The used classification algorithm is the Mathworks MATLAB [Mat15] implementation of a support vector machine. The pre-set "fine Gaussian SVM" was chosen and all kernel parameters set to their default values. As multi-class method the One-vs-All strategy was selected. The chosen SVM uses Error Correcting Output Codes (ECOC) to transform the multi-class problem into multiple two-class problems resulting in the training of multiple SVMs instead of a single one.

2.3 Selection Strategies

Four different training scenarios were implemented. The first is used as a reference the other three are the comparison between active and passive learning:

All at Once uses all available training samples to get the best possible performance. This value can be seen as a reference to which the other strategies are compared.

Random sampling was implemented by choosing the next training samples at random and represents the passive learning approach.

Uncertainty sampling represents an active learning approach and employs the strategy of the same name [LC94]. The certainty measure used is the estimated posterior probability which is included in the MATLAB default implementation.

Query by committee is a different active learning approach originally introduced in [SOS92]. We used a committee size of 5 and vote entropy as disagreement measure. It selects the next query sample x as

$$\underset{x}{\operatorname{argmax}} - \sum_{y} \frac{\operatorname{vote}_{\mathcal{C}}(y,x)}{|\mathcal{C}|} \log \frac{\operatorname{vote}_{\mathcal{C}}(y,x)}{|\mathcal{C}|},$$

where $\operatorname{vote}_{\mathcal{C}}(y,x) = \sum_{\theta \in \mathcal{C}} \mathbf{1}_{\{h_\theta(x)=y\}}$ is the number of "votes" that the committee $\mathcal{C}$ assigns to label y for sample x.

The latter three scenarios start their first training iteration with three samples per class. This is a requirement by the SVM implementation to estimate the posterior probability. The batch size was chosen such that after 30 iterations all training samples were exhausted. This resulted in the following batch sizes: Abenberg: 6, Indian Pines: 134, Pavia Centre: 7, Pavia Uni: 4, Vaihingen: 8. For reasons of computational costs the scenarios "uncertainty sampling"and "query by committee" for the Indian Pines dataset were aborted after 4,000 training samples were selected.

2.4 Accuracy Calculation

The first step of evaluating one method is to split the samples randomly into a training (75%) and a test set (25%). During the following training the evaluated selection strategy is allowed access to the feature data of the training set. Only after the samples for the next iteration are selected the label information is provided to the classification algorithm and the next iteration begins. The test set is never connected to the training process and only used for calculating the performance after each iteration. The performance measure used in this paper is the classification accuracy. This is the ratio of correct classified samples to total samples in the test set. Multiple runs of the whole process are done to get statistically robust results.

2.5 Area Under the Curve

To test if the difference between the three iterative scenarios is statistical significant, the performance of each execution was condensed into a single value. To achieve this the area under the learning curve was chosen (learning curve: accuracy vs. number of training samples). The learning curves were matched to span the same range of training samples. Then the trapezoid method was used to calculate the area.

3 Data

This work uses one internal and four publicly available real world datasets. This section gives a short description of them. For a visual impression of the data see Figure 1. It displays the data with overlayed ground truth. An overview of the datasets after the preprocessing step is given in Table 1.

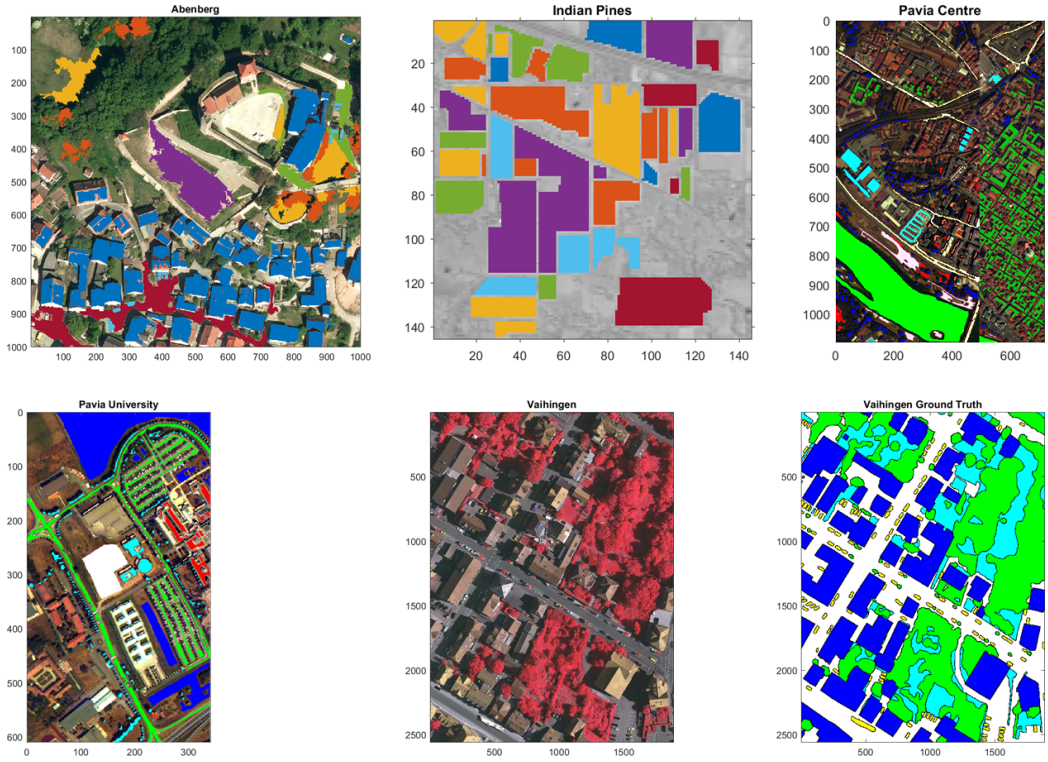


Figure 1: Visual impression of the five datasets overlayed with ground truth information. Vaihingen ground truth is displayed separately for better visualization.

3.1 Abenberg

This dataset is not publicly available. It is an aerial image produced by a survey of Technische Universitaet Muenchen over the Bavarian town of Abenberg in Germany. Each pixel has intensity information for 4 spectral bands: infrared, red, green, and blue. For this work a subset of 1,000 by 1,000 pixel was chosen such that it contains buildings, roads, woodlands, and open soil areas. The ground truth was manually created by this author (8 classes: roof, tree, grass, soil, gravel, car, asphalt, water).

Table 1: Overview of the different datasets after the pre-processing step. All datasets are aerial images of urban and vegetational areas. Each feature is one spectral band.

Dataset	Features	Classes[1]	Samples	SVM Accuracy	Class Distribution [2]
Abenberg	4	8	250	0.86	
Indian Pines	200	16	10,249	0.81	
Pavia (Centre)	102	7	355	0.99	
Pavia (University)	103	8	164	0.86	
Vaihingen (area 30)	3	6	320	0.71	

[1] Contents of original dataset.
[2] Displayed is the final distribution after classes with fewer than 15 samples were removed.

3.2 Indian Pines

This publicly available dataset is an aerial image of the Purdue University Agronomy farm north west of West Lafayette, USA [BBL] and covers different vegetation types. Each pixel is a spectrum containing 200 channels in the 400 to 2,500 nm range of the electromagnetic spectrum. Ground truth is available and contains 16 classes (Alfalfa, Corn-notill, Corn-mintill, Corn, Grass-pasture, Grass-trees, Grass-pasture-mowed, Hay-windrowed, Oats, Soybeans-notill, Soybeans-mintill, Soybeans-clean, Wheat, Woods, Building-Grass-Tree-Drives, Stone-Steel-Towers).

3.3 Pavia Centre & University

These two datasets are also publicly available [Bas11]. They consist of two aerial images of the city Pavia in northern Italy. Contained are urban and vegetation areas with a geometric ground resolution of 1.3 meters. The provided ground truth consists of 9 classes (Water, Trees, Meadows, Self-Blocking Bricks, Bare Soil, Asphalt, Bitumen, Tiles, Shadows), but is mislabelled in the cited datafile (as of submission of this paper). The correct labelling can be found in [Che06], page 494.

3.4 Vaihingen

The Vaihingen dataset stems from the ISPRS Benchmark Test on Urban Object Detection and Reconstruction[1]. It is publicly available and contains multiple aerial images of the town of Vaihingen in Baden-Württemberg, Germany. For each pixel there are intensity values for three channels: infrared, green, and blue. Height information acquired by a LiDAR scanner is also available, but not used in this work. The provided ground truth has six classes (Car, Tree, Low vegetation, Building, Impervious surfaces).

4 Results

Figure 2 shows the learning curves of the methods for the five datasets. They were generated by plotting the classification accuracy (see 2.4) over the number of used training samples. Four of the five datasets show similar performance between active and passive learning with a slight advantage towards passive learning. On the Indian Pines dataset uncertainty sampling greatly underperformes in comparison to random sampling. Each learning curve converges towards the performance of the "All at Once" method which is to be expected, because when all samples are used, the order doesn't matter.

Histograms of the area under the curve for each execution are shown in Figure 3. Those values were used to determine if there are statistically significant differences between the selection strategies. Table 2 lists the p-values of the used statistical test (here: two-sample t-test).

[1]The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [Cra10]: `http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html`.

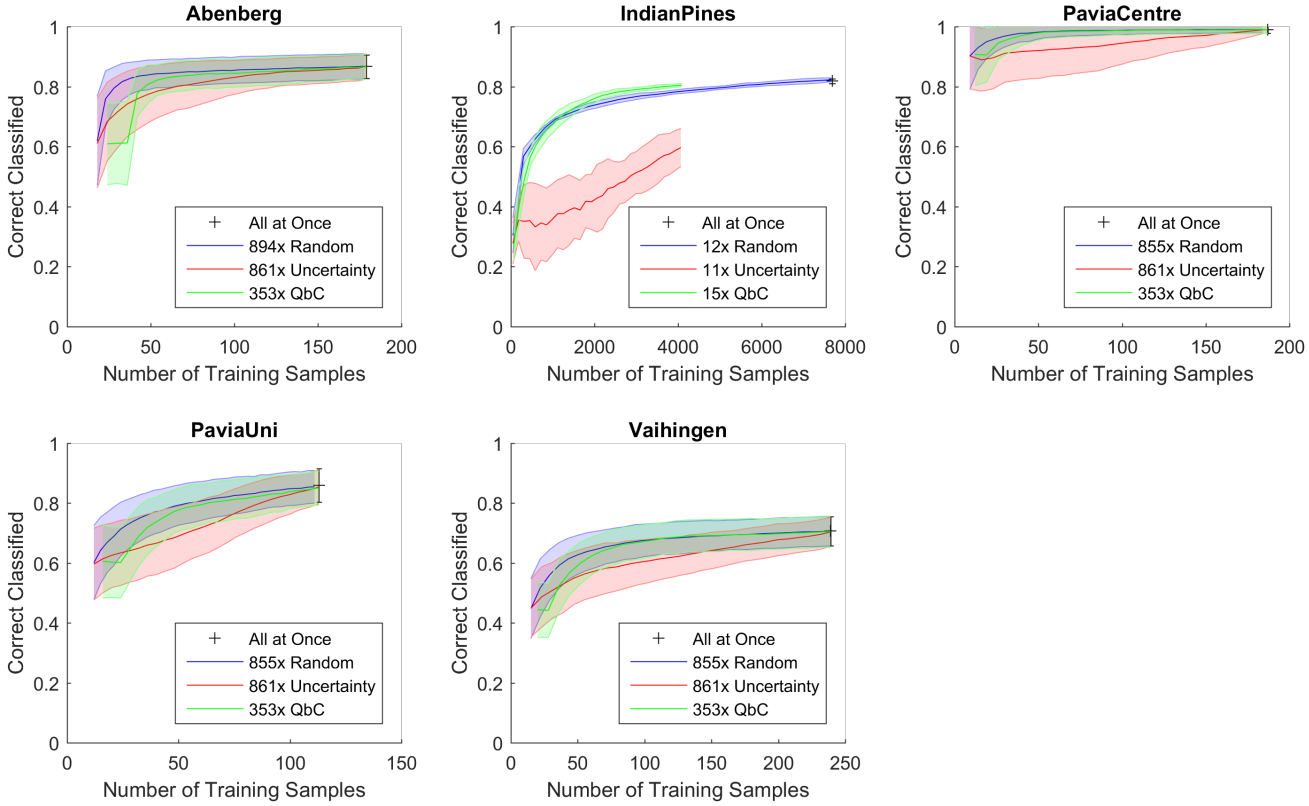


Figure 2: Learning curves for passive ("Random") and active ("Uncertainty", "Query by Committee") learning. "All at Once" is the reference for maximal achievable accuracy. All methods were executed multiple times to reduce the influence of the random splitting into training and test set. The centre line of the graphs is the mean accuracy and the shaded area is the standard deviation.

Table 2: Resulting p-values of the two-sample t-test between the three selection strategies. All combinations except query by committee vs. random sampling on Indian Pines and query by committee vs. uncertainty sampling on Abenberg show p-values nearly equal to zero which indicates a strong statistically significant difference between their performance.

Dataset	Random vs. Uncertainty	Random vs. Query by Committee	Uncertainty vs. Query by Committee
Abenberg	10^{-54}	10^{-32}	0.26
Indian Pines	10^{-9}	0.09	10^{-11}
Pavia (Centre)	10^{-106}	10^{-7}	10^{-39}
Pavia (University)	10^{-109}	10^{-18}	10^{-22}
Vaihingen (area 30)	10^{-104}	10^{-6}	10^{-36}

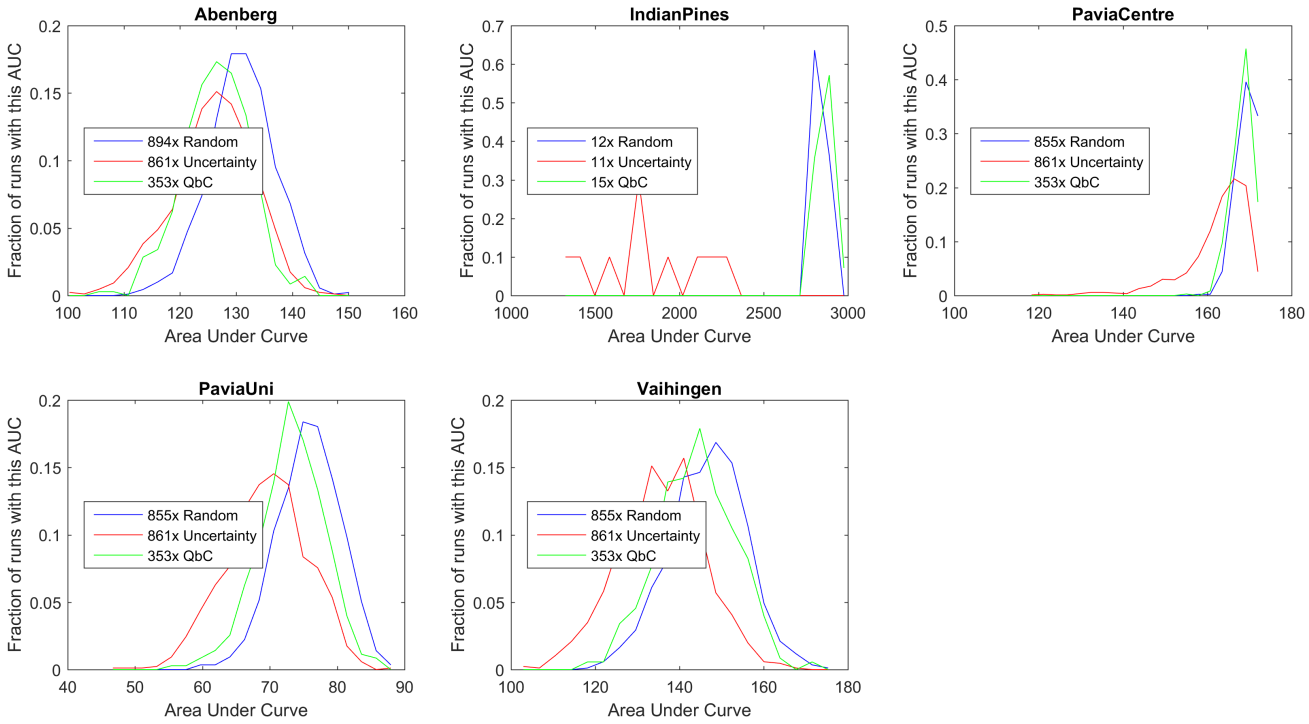


Figure 3: Histograms of the areas under the curve for the different selection strategies and data sets. Each histogram was normalized so that all bins sum to 1.

5 Discussion

Almost all strategies show significant differences for AUC performance. However, the magnitude of the difference measured in correct classified percentage is less then ten percent points which is well within the observed standard deviation. To summarize: the results show that random sampling has a, though small in magnitude, still statistically significant advantage over active learning. This is in stark contrast to most literature. It needs to be determined if this is a problem of the datasets, implementation, or choice of selection strategy. Following we give a list of possible causes, suggest how to test for them, and offer potential solutions.

5.1 Wrong Quality Metric

Cause [RLSKB16] have shown that using a single performance measure can be misleading in an active learning setting.

Test To test this, other measurements such as F_1-Score or area under the receiver operating characteristic curve (AUROC) should be evaluated.

Solution There is no metric that fits every problem. Instead the metric must be chosen to accommodate the domain specific needs. In remote sensing the costs of acquiring more training samples is often higher than the cost of false negatives. However some instances can be weighted opposite for example in the area of Counter-IED (improvised explosive device) detection.

5.2 Uneven Class Distribution

Cause Related to the problem of the wrong quality metric is the problem of uneven class distributions. This is the case if one class is much more common or rarer than others. Random sampling replicates this distribution so that the classification algorithm is trained on the same distribution as it is tested on. In the case of active learning the distribution changes and doesn't match the one from the test data. However it should be noted that it is sometimes argued this bias is the advantage of active learning since it avoids querying redundant samples from overrepresented classes [Mit90].

Test This can be tested by noting which samples are selected during the training process and observing their change of class distribution directly. Also artificially reducing the presence of one class could lead to new insights.

Solution This problem can be alleviated by avoiding use of a classification algorithm that relies on the sample distribution like a Maximum-Likelihood classifier [WMS14]. Instead a non-statistical classifier should be chosen.

5.3 Separability

Cause In remote sensing data the individual pixels often don't contain a single material, but rather a mixture of materials. This leads to overlapping representations in the feature space. If the samples are not separable based on the given features the classifier can't generalize very well.

Test In case of an SVM this could be observed by analysing how many support vectors are used. If the number doesn't increase with more training samples the generalization of the SVM is good. The effect of overlapping classes can be investigated in detail by using specifically generated synthetic datasets or comparing two easily separable classes versus two difficult to separate classes in a two-class setting.

Solution To increase the separability a pre-processing step with feature extraction needs to be introduced. However it remains to be seen if this is an advantage for active learning or just an increase in overall accuracy for both active and passive learning.

5.4 Too Many Samples Per Iteration

Cause The used uncertainty sampling method is based on the estimated posterior probability. To get a good estimate at least three samples per class are needed. Because of this large initial training size the SVM has very good performance from the beginning and shows only very small improvements for the rest of the training so that it is hard to improve by active learning methods. Furthermore for batch sizes with multiple samples the redundant information contained in one batch increases.

Test Observe the classification accuracy of the SVM when initially trained with fewer samples per class. Using smaller batch sizes to reduce the amount of redundant information that is selected in each iteration, should increase the performance.

Solution Use the distance to the hyperplane instead of the estimated posterior probability for the uncertainty sampling method. This alleviates the need for multiple initial training samples per class. The redundant information in one batch can be reduced by adding a second decision criterion like maximising the distance between selected samples in feature space (e.g. density weighted active learning or a diversity criterion [BP09]).

5.5 Using Only Label Information

Cause The presented variant of uncertainty sampling selects samples only based on the state of the learning algorithm. Therefore only information based on the labels of the data is used and information gained from the unlabelled data itself is not utilized.

Solution Applying methods from semi- and unsupervised learning can be beneficial and lead to strategies such as cluster based and hierarchical active learning [LC04], [SOM+12].

References

[And76] J. R. Anderson. *A Land Use and Land Cover Classification System for Use with Remote Sensor Data*. Geological Survey professional paper. U.S. Government Printing Office, 1976. URL: `https://books.google.de/books?id=dE-ToP4UpSIC`.

[Bas11] Basque University. Pavia centre and university: Hyperspectral remote sensing scenes, 2011. URL: `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#Pavia_Centre_and_University`.

[BBL] Marion F. Baumgardner, Larry L. Biehl, and David A. Landgrebe. 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3. URL: `https://purr.purdue.edu/publications/1947/1`, `doi:10.4231/R7RX991C`.

[BP09] Lorenzo Bruzzone and Claudio Persello. Active learning for classification of remote sensing images. In *International Geoscience and Remote Sensing Symposium*, pages III–693–III–696, Piscataway, NJ, 2009. IEEE. `doi:10.1109/IGARSS.2009.5417857`.

[Che06] C. H. Chen. *Signal and Image Processing for Remote Sensing*. CRC Press, 2006. URL: `https://books.google.de/books?id=9CiW0hgiwKYC`.

[Cra10] Michael Cramer. The dgpf-test on digital airborne camera evaluation overview and test design. *PFG Photogrammetrie, Fernerkundung, Geoinformation*, 2010(2):73–82, 2010. URL: `http://dx.doi.org/10.1127/1432-8364/2010/0041`.

[DJ00] Antonio Di Gregorio and Louisa J. M. Jansen. *Land cover classification systems (LCCS): Classification concepts and user manual*. Food and Agriculture Organization of the United Nations, Rome, 2000.

[FM04] Giles M. Foody and Ajay Mathur. Toward intelligent training of supervised image classifications: directing training data acquisition for svm classification. *Remote Sensing of Environment*, 93(1-2):107–117, 2004. `doi:10.1016/j.rse.2004.06.017`.

[LC94] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1994.

[LC04] Sanghoon Lee and M. M. Crawford. Hierarchical clustering approach for unsupervised image classification of hyperspectral data. In *International Geoscience and Remote Sensing Symposium: Proceedings*, pages 941–944, 2004. `doi:10.1109/IGARSS.2004.1368563`.

[Mat15] MathWorks. Matlab, 2015.

[Mit90] Tom M. Mitchell. The need for biases in learning generalizations. In ?, editor, *Readings in Machine Learning*. Morgan Kaufmann, 1990.

[RLSKB16] Maria E. Ramirez-Loaiza, Manali Sharma, Geet Kumar, and Mustafa Bilgic. Active learning: An empirical study of common baselines. *Data Mining and Knowledge Discovery*, 2016. `doi:10.1007/s10618-016-0469-7`.

[Sch12] Konrad Schindler. An overview and comparison of smooth labeling methods for land-cover classification. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4534–4545, 2012. `doi:10.1109/TGRS.2012.2192741`.

[SOM+12] J. Senthilnath, S. N. Omkar, V. Mani, P. G. Diwakar, and Archana Shenoy B. Hierarchical clustering algorithm for land cover mapping using satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(3):762–768, 2012. `doi:10.1109/JSTARS.2012.2187432`.

[SOS92] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Computational Learning Theory*, pages 287–294. ACM, 1992. `doi:10.1145/130385.130417`.

[Tri14] Trimble Navigation Limited. ecognition developer, 2014.

[TRP+09] Devis Tuia, F. Ratle, F. Pacifici, Mikhail F. Kanevski, and William J. Emery. Active learning methods for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2218–2232, 2009. `doi:10.1109/TGRS.2008.2010404`.

[TVC+11] Devis Tuia, Michele Volpi, Loris Copa, Mikhail F. Kanevski, and Jordi Munoz-Mari. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617, 2011. `doi:10.1109/JSTSP.2011.2139193`.

[WMS14] Sebastian Wuttke, Wolfgang Middelmann, and Uwe Stilla. Bewertung von strategien des aktiven lernens am beispiel der landbedeckungsklassifikation. In *34. Wissenschaftlich-Technische Jahrestagung*, volume 2014, 2014. URL: `http://publica.fraunhofer.de/dokumente/N-283921.html`.

Dynamic Parameter Adaptation of SVM Based Active Learning Methodology

Jasmina Smailović[1] Miha Grčar[1] Nada Lavrač[1,2] Martin Žnidaršič[1]

[1]Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia (name.surname@ijs.si)
[2]University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia

Abstract

In this paper we present experimental assessment of a dynamic adaptation of an approach for sentiment classification of tweets. Specifically, this approach enables a dynamic adaptation of the parameters used for three-class classification with a binary SVM classifier. The approach is suited for incremental active learning scenarios in domains with frequent concept alterations and changes. Our target application is in domain of finance and the assessment is partially domain-specific, but the approach itself is not limited to a particular domain.

1 Introduction

The work presented in this paper is aimed at the analysis of sentiment in Twitter messages, which became a very common and well studied problem [KZM14, MGS16, NRR+16]. Our specific focus, though, is on employment of techniques of incremental active learning in a financial domain. The general aim of our work is to develop a methodology that would allow keeping a stock-company focused sentiment classifier up-to-date with minimal human effort. Namely, in case of using informal data sources, like Twitter, and in dynamic target domains such as finance, updating of sentiment classifiers is necessary, as new data features emerge and existing features can change or even reverse their impact on sentiment classification. To take this into account, sentiment lexicon based approaches must update the lexicons, while in machine learning approaches that work with n-grams, the learning processes have to be repeated or an incremental learning algorithm must be employed. In any case, new labeled data is needed, which usually represents the main practical obstacle. Namely, labeling new data in this domain requires human expert effort, thus its frequency is limited (e.g., we cannot get hundreds of new labels per second, even if the cost would not be a constraint) and usually its volume (cost) as well. Therefore, it is beneficial to use an appropriate active learning strategy in order to limit this effort as much as possible.

A particular analysis of active learning strategies that we present in this paper is concerned with the assessment of impacts of a technique for dynamic adaptation of the parameters of an SVM based active learning. Specifically, we elaborate upon the concept of the dynamic neutral zone [Sma14] and present an extended experimental assessment of this approach. The neutral zone is the area around the SVM classifier's hyperplane that distinguishes among the examples that are to be classified as positive and those that are to be classified as negative [SGLŽ13, SGLŽ14, SKG+15]. Definition of such an area allows for three-class classification (negative, neutral, positive) with a binary SVM classifier in cases when only positive and negative learning data is available. An adaptive version of such an area definition, which was recently proposed [Sma14] and denoted as dynamic neutral zone, is able to adapt to the characteristics of new labeled data that becomes available by active learning.

In: G. Krempl, V. Lemaire, E. Lughofer, and D. Kottke (eds.): Proceedings of the Workshop Active Learning: Applications, Foundations and Emerging Trends, AL@iKNOW 2016, Graz, Austria, 18-OCT-2016, published at http://ceur-ws.org

The presented active learning methodology and the approach for examining the relationship between tweet sentiment and stock prices is explained in detail in our previous studies [Sma14, SGLŽ13, SGLŽ14]. In Section 2 we briefly revisit the active learning approach and the concept of the neutral zone. The new extended experiments with the dynamic neutral zone are listed in Section 3 and discussed in Section 4.

2 Methodology

The aim of our experiments is to discover the best combination of parameters of the developed active learning methodology [Sma14, SGLŽ14] for sentiment analysis. The initial sentiment model is trained using the smiley-labeled Twitter messages [GBH09][1] by employing the Support Vector Machine (SVM) [Vap95] algorithm. We measure the model performance (in terms of the F-measure of the positive class) on a simulated stream of tweets by employing the holdout evaluation approach adjusted for dynamic environments [BK09, IGD11], that is, we evaluate the model on each new batch of data from the Twitter data stream. The simulated data stream consists of tweets which discuss Baidu[2] stocks in year 2011. Moreover, active learning is performed, i.e. a selection of tweets from each batch is chosen to be manually labeled and added to the model.

The sentiment model is trained using the positive and negative tweets. However, in the classification phase we adjust the output of the SVM algorithm to detect also the neutral tweets by employing the concept of the neutral zone, that is, examples which are positioned in the neutral zone are marked as neutral. There are various ways of implementing the concept of the neutral zone. For example, the fixed neutral zone is constrained by empirically predefined boundaries [SGLŽ13, SGLŽ14, Sma14] as sketched in Figure 1(a), while the relative neutral zone is a function of positive and negative average distances of training examples [SKG+15, Sma14]. The key idea of the latter approach is the following. Given that an example is projected on the positive side of the SVM hyperplane at distance d and the average distance of positive training examples is $\bar{d}_+$, the first step is to calculate the classification reliability by applying the following formula [SKG+15, Sma14]:

$$R = \frac{d}{2 * \bar{d}_+} \tag{1}$$

If the calculated reliability is greater than 1, it is transformed to $R = 1$. The example is labeled as neutral if its classification reliability is below a predefined reliability threshold R_T. Figure 1(b) presents an example of classifying an instance (at distance d) in this setting. The same approach (with using the average distances of negative training examples) is applied if an example is projected on the negative side of the SVM hyperplane.

In the active learning environment we dynamically update not only the sentiment model, but also the parameters of the relative neutral zone, i.e. the average training distances. The positive average distance is updated by applying the following formula [Sma14]:

$$\bar{d_+}' = (1 - \alpha) * \bar{d}_+ + \alpha * \bar{d}_b \tag{2}$$

where $\bar{d_+}'$ is an updated average distance, $\bar{d}_+$ is the current one, and $\bar{d}_b$ is the average distance of the positive examples in the currently processed batch b, which were used for updating the model. Parameter α controls the influence of the new and previous tweets. If α is set to 0, the average distance of initial training examples does not get dynamically updated. Equation 2 is applied accordingly for dynamically updating the negative average distance.

We experimented with the following active learning query strategies [Sma14, SGLŽ14] to select the most suitable examples from each batch of data for manual labeling:

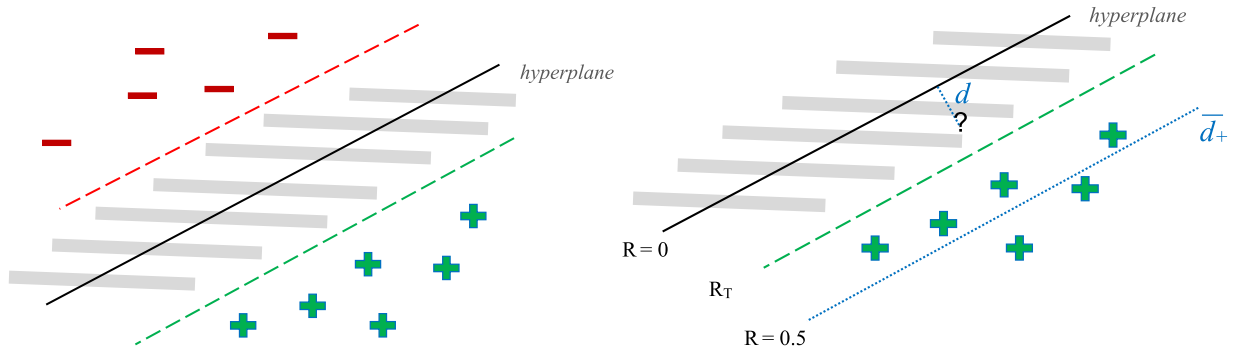
1. Closest to the neutral zone: the algorithm chooses a selection of tweets whose classification reliability is closest to the reliability threshold. The number of positive/negative examples (according to the classifier's labeling) must not exceed half of the allocated manual labels.

2. Random: the algorithm randomly selects tweets for manual labeling.

3. Combined approach: combination of two previous approaches, i.e. a certain percentage of tweets is chosen randomly, while the rest of the tweets are chosen according to the "Closest to the neutral zone" strategy.

[1] The dataset was obtained from the Sentiment140 Web page, section "For Academics" (`http://help.sentiment140.com/for-students`).

[2] `http://www.baidu.com/`.



(a) Fixed Neutral Zone (b) Relative Neutral Zone

Figure 1: Two Approaches to the Concept of the Neutral Zone

Additionally, we evaluated the scenario without active learning, i.e. without updating the sentiment model or neutral zone.

The Friedman test [Dem06, Fri37, Fri40], the Iman-Davenport improvement [ID80], and the Nemenyi post-hoc test [Nem63] were used to rank a selection of the evaluated active learning settings and to find statistically significant differences between them.

The implementation of the methodology uses elements of several libraries: Pegasos SVM [SSSS07] from the sofia-ml library[3] [Scu10], SWIG[4] to connect sofia-ml C++ implementation with C# programming language, and the LATINO library[5] for preparing the features. The learning algorithm for the initial model training in sofia-ml was adapted by implementing sampling which takes examples in succession.

3 Experiments

In this study we extend the experimental setting from [Sma14] and test the following parameters and their values (the parameters used already in [Sma14] are also included):

- Alpha values: 0, 0.05, 0.1, 0.3, 0.5.
- Five active learning querying strategies and one without active learning.
- Two batch selection strategies: select 10 of 100 (select 10 examples for manual labeling out of 100 examples in a batch) and select 10 of 50.
- Reliability threshold: 0, 0.1, 0.2, 0.3, 0.4, 0.5.

The results of experimental assessment for all combinations of the above parameters are presented in Table 1.

From Table 1 it is not straightforward to conclude which combination of parameters is the best one. For that reason, we present the average results of active learning strategies over all values of reliability threshold in Figure 2, where lighter color corresponds to lower values and darker color corresponds to higher (better) values. We exclude the "AL closest to NZ" strategy since the results in Table 1, which are marked with asterisk(s), indicate that this strategy is unreliable as many batches did not have positively classified tweets, which caused missing values of F-measure (see [Sma14] for more details on this phenomenon). The Figure 2 indicates that both 0.1 and 0.3 are reasonable values for the parameter α. However, we focus on $\alpha = 0.3$, since we already performed the analysis of $\alpha = 0.1$ in our previous study [Sma14].

[3] `https://code.google.com/p/sofia-ml/`.
[4] `http://www.swig.org/`.
[5] `https://github.com/LatinoLib/LATINO`.

Table 1: Average F-measure of the Positive Class ± Std. Deviation for Different Active Learning and Batch Selection Strategies, *Alpha* Values, and Reliability Thresholds

			$\alpha = 0$			
Rel. threshold	0	0.1	0.2	0.3	0.4	0.5
Select 10 of 100						
AL closest to NZ	0.5512±0.12	0.5396±0.12	0.5282±0.11	0.5165±0.11	0.5018±0.11	0.4802±0.11
AL comb. 20% r.	0.5512±0.12	0.5396±0.12	0.5281±0.11	0.5164±0.11	0.5017±0.11	0.4803±0.11
AL comb. 50% r.	0.5513±0.12	0.5398±0.12	0.5283±0.11	0.5165±0.11	0.5016±0.11	0.4803±0.11
AL comb. 80% r.	0.5512±0.12	0.5396±0.12	0.5281±0.11	0.5165±0.11	0.5017±0.11	0.4803±0.11
AL 100% rand.	0.5514±0.12	0.5399±0.12	0.5281±0.11	0.5169±0.11	0.5017±0.11	0.4804±0.11
No AL	0.5500±0.12	0.5389±0.12	0.5277±0.11	0.5162±0.11	0.5004±0.11	0.4787±0.10
Select 10 of 50						
AL closest to NZ	0.5466±0.14	0.5342±0.14	0.5221±0.14	0.5103±0.14	0.4956±0.13	0.4756±0.13
AL comb. 20% r.	0.5466±0.14	0.5339±0.14	0.5220±0.14	0.5103±0.14	0.4957±0.13	0.4757±0.13
AL comb. 50% r.	0.5465±0.14	0.5340±0.14	0.5219±0.14	0.5104±0.14	0.4957±0.13	0.4758±0.13
AL comb. 80% r.	0.5468±0.14	0.5340±0.14	0.5218±0.14	0.5103±0.14	0.4959±0.13	0.4756±0.13
AL 100% rand.	0.5466±0.14	0.5341±0.14	0.5222±0.14	0.5109±0.14	0.4963±0.13	0.4762±0.13
No AL	0.5444±0.14	0.5329±0.14	0.5213±0.14	0.5094±0.14	0.4938±0.13	0.4731±0.13
			$\alpha = 0.05$			
Rel. threshold	0	0.1	0.2	0.3	0.4	0.5
Select 10 of 100						
AL closest to NZ	0.5512±0.12	0.5484±0.12	0.5749±0.11**	0.5688±0.11**	0.5667±0.11**	0.5547±0.11**
AL comb. 20% r.	0.5513±0.12	0.5460±0.12	0.5408±0.12	0.5329±0.11	0.5298±0.11	0.5024±0.11
AL comb. 50% r.	0.5512±0.12	0.5424±0.12	0.5344±0.11	0.5221±0.11	0.5084±0.11	0.4545±0.10
AL comb. 80% r.	0.5515±0.12	0.5377±0.11	0.5250±0.11	0.5126±0.11	0.4821±0.11	0.4435±0.11
AL 100% rand.	0.5514±0.12	0.5340±0.11	0.5168±0.11	0.4971±0.11	0.4640±0.11	0.4296±0.11
No AL	0.5500±0.12	0.5389±0.12	0.5277±0.11	0.5162±0.11	0.5004±0.11	0.4787±0.10
Select 10 of 50						
AL closest to NZ	0.5471±0.14	0.5613±0.14	0.5394±0.14	0.5341±0.14	0.5985±0.12**	0.5920±0.12**
AL comb. 20% r.	0.5466±0.14	0.5406±0.14	0.5365±0.14	0.5278±0.14	0.5243±0.14	0.5126±0.14
AL comb. 50% r.	0.5465±0.14	0.5367±0.14	0.5258±0.14	0.5155±0.14	0.4939±0.13	0.4518±0.13
AL comb. 80% r.	0.5467±0.14	0.5331±0.14	0.5185±0.14	0.4979±0.13	0.4713±0.13	0.4445±0.13
AL 100% rand.	0.5466±0.14	0.5295±0.14	0.5149±0.14	0.4961±0.13	0.4740±0.13	0.4491±0.13
No AL	0.5444±0.14	0.5329±0.14	0.5213±0.14	0.5094±0.14	0.4938±0.13	0.4731±0.13
			$\alpha = 0.1$			
Rel. threshold	0	0.1	0.2	0.3	0.4	0.5
Select 10 of 100						
AL closest to NZ	0.5512±0.12	0.5808±0.12**	0.5800±0.10**	0.5923±0.10**	0.5356±0.11**	0.5765±0.10**
AL comb. 20% r.	0.5530±0.12	0.5463±0.12	0.5432±0.11	0.5375±0.11	0.5289±0.11	0.5102±0.11
AL comb. 50% r.	0.5513±0.12	0.5415±0.12	0.5320±0.12	0.5246±0.11	0.5116±0.11	0.4831±0.11
AL comb. 80% r.	0.5512±0.12	0.5384±0.11	0.5279±0.11	0.5128±0.11	0.4833±0.11	0.4531±0.10
AL 100% rand.	0.5514±0.12	0.5335±0.11	0.5164±0.11	0.4961±0.11	0.4638±0.11	0.4323±0.11
No AL	0.5500±0.12	0.5389±0.12	0.5277±0.11	0.5162±0.11	0.5004±0.11	0.4787±0.10
Select 10 of 50						
AL closest to NZ	0.5766±0.15**	0.5682±0.14*	0.6349±0.12**	0.6348±0.11**	0.6250±0.11**	0.5114±0.14*
AL comb. 20% r.	0.5466±0.14	0.5398±0.14	0.5382±0.14	0.5328±0.14	0.5237±0.14	0.5172±0.14
AL comb. 50% r.	0.5464±0.14	0.5359±0.14	0.5262±0.14	0.5173±0.14	0.4957±0.14	0.4690±0.13
AL comb. 80% r.	0.5463±0.14	0.5303±0.14	0.5188±0.14	0.5020±0.14	0.4756±0.13	0.4359±0.13
AL 100% rand.	0.5466±0.14	0.5299±0.14	0.5153±0.14	0.4967±0.14	0.4751±0.13	0.4521±0.13
No AL	0.5444±0.14	0.5329±0.14	0.5213±0.14	0.5094±0.14	0.4938±0.13	0.4731±0.13

$\alpha = 0.3$						
Rel. threshold	0	0.1	0.2	0.3	0.4	0.5
Select 10 of 100						
AL closest to NZ	0.5499±0.12*	0.5734±0.11**	0.5394±0.11**	0.5320±0.12**	0.5322±0.11	0.6219±0.09**
AL comb. 20% r.	0.5617±0.12	0.5442±0.11	0.5533±0.11	0.5420±0.11	0.5356±0.11	0.5244±0.11
AL comb. 50% r.	0.5497±0.12	0.5431±0.12	0.5284±0.11	0.5160±0.12	0.5173±0.11	0.4806±0.12
AL comb. 80% r.	0.5554±0.12	0.5356±0.11	0.5311±0.11	0.5022±0.11	0.4817±0.11	0.4639±0.12
AL 100% rand.	0.5514±0.12	0.5337±0.11	0.5161±0.11	0.4951±0.11	0.4701±0.11	0.4337±0.12
No AL	0.5500±0.12	0.5389±0.12	0.5277±0.11	0.5162±0.11	0.5004±0.11	0.4787±0.10
Select 10 of 50						
AL closest to NZ	0.5557±0.15*	0.5307±0.15*	0.5293±0.14*	0.5210±0.14*	0.5298±0.15*	0.5124±0.14
AL comb. 20% r.	0.5470±0.14	0.5398±0.14	0.5446±0.14	0.5305±0.14	0.5193±0.14	0.5162±0.13
AL comb. 50% r.	0.5467±0.14	0.5348±0.14	0.5267±0.14	0.5230±0.14	0.5029±0.14	0.4654±0.14
AL comb. 80% r.	0.5467±0.14	0.5343±0.14	0.5199±0.14	0.5035±0.14	0.4703±0.14	0.4411±0.14
AL 100% rand.	0.5455±0.14	0.5291±0.14	0.5134±0.14	0.4922±0.14	0.4730±0.13	0.4517±0.14
No AL	0.5444±0.14	0.5329±0.14	0.5213±0.14	0.5094±0.14	0.4938±0.13	0.4731±0.13
$\alpha = 0.5$						
Rel. threshold	0	0.1	0.2	0.3	0.4	0.5
Select 10 of 100						
AL closest to NZ	0.5706±0.11	0.5442±0.12*	0.5308±0.12**	0.5251±0.11*	0.5058±0.11**	0.5299±0.12**
AL comb. 20% r.	0.5577±0.11	0.5506±0.12	0.5466±0.11	0.5427±0.12	0.5282±0.12	0.4902±0.12
AL comb. 50% r.	0.5496±0.12	0.5381±0.11	0.5310±0.11	0.5253±0.11	0.5017±0.12	0.4488±0.13
AL comb. 80% r.	0.5479±0.12	0.5392±0.12	0.5171±0.12	0.4998±0.11	0.4629±0.12	0.4539±0.11
AL 100% rand.	0.5514±0.12	0.5339±0.11	0.5178±0.11	0.4941±0.11	0.4693±0.12	0.4362±0.12
No AL	0.5500±0.12	0.5389±0.12	0.5277±0.11	0.5162±0.11	0.5004±0.11	0.4787±0.10
Select 10 of 50						
AL closest to NZ	0.5414±0.14*	0.5234±0.14*	0.5101±0.14*	0.5043±0.15*	0.5299±0.14	0.4995±0.14
AL comb. 20% r.	0.5370±0.15	0.5359±0.14	0.5336±0.14	0.5269±0.14	0.5188±0.14	0.5076±0.14
AL comb. 50% r.	0.5463±0.14	0.5356±0.14	0.5287±0.14	0.5109±0.14	0.4998±0.14	0.4673±0.14
AL comb. 80% r.	0.5486±0.14	0.5304±0.14	0.5185±0.14	0.5008±0.13	0.4692±0.14	0.4397±0.15
AL 100% rand.	0.5440±0.14	0.5273±0.14	0.5108±0.14	0.4900±0.13	0.4696±0.13	0.4467±0.14
No AL	0.5444±0.14	0.5329±0.14	0.5213±0.14	0.5094±0.14	0.4938±0.13	0.4731±0.13

** sample contains less than 50% of data batches from which positive F-measure could be calculated.
* sample contains less than 70% of data batches from which positive F-measure could be calculated.

The results of the Friedman test with the Iman-Davenport improvement and the Nemenyi post-hoc test for $\alpha = 0.3$ are presented in Figure 3. The strategies which are not significantly different are connected with a red line. From the figure it follows that the best active learning settings are: "Select 10 of 100 with AL comb. 20% random","Select 10 of 50 with AL comb. 20% random" and "Select 10 of 100 with AL comb. 50% random".

Finally, we analyze the relationship between sentiment in tweets and stock closing prices of the discussed company. We apply the Granger causality test [Gra69] for different time lags and time periods on two time series: daily change of the positive sentiment probability and daily return in stock closing price [Sma14, SGLŽ13, SGLŽ14]. This statistical test indicates whether one time series is useful for predicting the values of another one. The results for top three active learning settings for $\alpha = 0.3$ are shown in Table 2. The significant results, after applying the Bonferroni correction [Abd07], are marked in bold (which corresponds to values lower than 0.025).

4 Discussion and conclusions

We presented an extended experimental assessment of the active learning methodology with dynamic neutral zone in which we were particularly interested in experimenting with the parameter α which dynamically updates the neutral zone as new examples arrive from the data stream. The conclusions of this extended study are in agreement with our previous one [Sma14]. The indications about the characteristics of the neutral zone are now strengthened, but for many aspects still lack a decisive statistical significance.

For $\alpha = 0$ all the active learning strategies are better than the strategy without active learning. However, the differences between the strategies are not so prominent (see Figure 2). On the other hand, the results with the dynamic neutral zone ($\alpha > 0$) are more diverse between different strategies and the new results in this setting show that besides completely random query strategy, also combinations with a strong random component (80%) are even worse than not applying active learning at all. In general, the "Select 10 of 100" batch selection seems to be somewhat better than "Select 10 of 50" selection, which is not intuitive, but might be partly caused by partitioning of the batches. Moreover, in larger batches the querying strategies might be more effective as they operate on larger number of different examples. Discovery of the exact cause would be a possible direction for further work. Regarding the query strategies, the combined seem to be the best ones, but the differences among them are usually not significant (see Figure 3). The Granger causality analysis showed that there is a relationship between sentiment in tweets and stock prices in specific time periods, mostly June-August, as already shown in [Sma14]. The relationship also depends on choosing an appropriate active learning setting and the value of reliability threshold.

Alpha	0	0.05	0.1	0.3	0.5
Select 10 of 100					
AL comb. 20% rand.	0.5196	0.5339	0.5365	0.5435	0.5360
AL comb. 50% rand.	0.5196	0.5188	0.5240	0.5225	0.5158
AL comb. 80% rand.	0.5196	0.5087	0.5111	0.5117	0.5035
AL 100% rand.	0.5197	0.4988	0.4989	0.5000	0.5005
No AL	0.5187	0.5187	0.5187	0.5187	0.5187
Select 10 of 50					
AL comb. 20% rand.	0.5140	0.5314	0.5331	0.5329	0.5267
AL comb. 50% rand.	0.5140	0.5117	0.5151	0.5166	0.5148
AL comb. 80% rand.	0.5141	0.5020	0.5015	0.5026	0.5012
AL 100% rand.	0.5144	0.5017	0.5026	0.5008	0.4981
No AL	0.5125	0.5125	0.5125	0.5125	0.5125

Figure 2: Averaged F-measure Results from Table 1 for All Active Learning Strategies (Except "AL Closest to NZ") Over All Values of Reliability Threshold

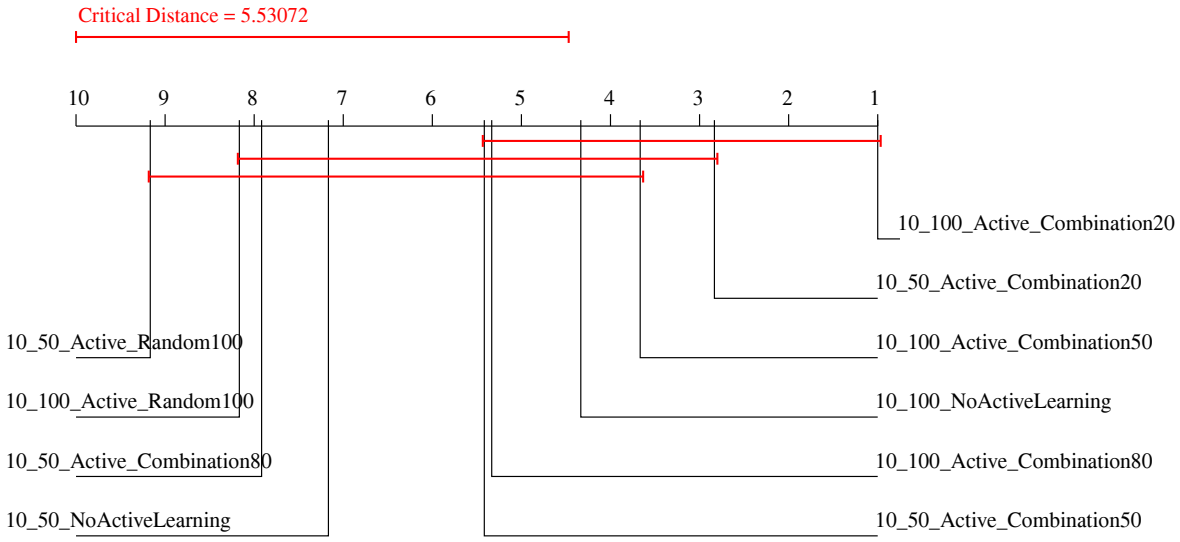


Figure 3: The Results of the Friedman Test With the Iman-Davenport Improvement and the Nemenyi Post-hoc Test for $\alpha = 0.3$

Table 2: Granger Causality Results (p-values) Between Daily Change of the Positive Sentiment Probability and Daily Return in Stock Closing Price for Baidu. The Results of Three Active Learning Query Strategies for $\alpha = 0.3$ are Shown. Statistically Significant Results are Marked in Bold

Reliability threshold	Lag	0	0.1	0.2	0.3	0.4	0.5
Select 10 of 100, comb. 20% rand.							
9 months	1	0.874	0.785	0.190	0.170	0.630	0.773
March - May	1	0.247	0.469	0.507	0.657	0.815	0.887
June - August	1	0.152	0.396	0.282	0.044	0.451	0.331
September - November	1	0.416	0.696	0.212	0.604	0.906	0.625
9 months	2	0.705	0.837	0.199	0.240	0.497	0.352
March - May	2	0.148	0.244	0.644	0.369	0.860	0.534
June - August	2	0.292	0.713	0.093	0.033	0.147	0.175
September - November	2	0.698	0.683	0.350	0.384	0.849	0.272
9 months	3	0.696	0.946	0.328	0.358	0.586	0.331
March - May	3	0.269	0.444	0.805	0.540	0.916	0.541
June - August	3	0.384	0.413	**0.024**	0.026	0.099	0.069
September - November	3	0.822	0.324	0.439	0.342	0.864	0.322
Select 10 of 50, comb. 20% rand.							
9 months	1	0.565	0.099	0.415	0.666	0.856	0.912
March - May	1	0.545	0.915	0.934	0.227	0.352	0.510
June - August	1	0.537	0.034	0.719	0.276	0.660	0.125
September - November	1	0.269	0.450	0.159	0.532	0.169	0.400
9 months	2	0.904	0.150	0.681	0.417	0.992	0.975
March - May	2	0.067	0.941	0.927	0.339	0.537	0.734
June - August	2	0.807	0.042	0.584	0.037	0.442	0.096
September - November	2	0.157	0.639	0.300	0.664	0.352	0.292
9 months	3	0.321	0.131	0.719	0.395	0.996	0.985
March - May	3	0.068	0.295	0.970	0.515	0.783	0.855
June - August	3	0.345	0.066	0.608	0.121	0.373	0.094
September - November	3	0.320	0.538	0.314	0.524	0.553	0.517
Select 10 of 100, comb. 50% rand.							
9 months	1	0.531	0.076	0.232	0.124	0.118	0.793
March - May	1	0.286	0.960	0.438	0.881	0.909	0.836
June - August	1	0.089	0.042	0.026	0.028	**0.021**	0.046
September - November	1	0.783	0.709	0.854	0.745	0.790	0.258
9 months	2	0.405	0.129	0.334	0.049	0.142	0.130
March - May	2	0.541	0.572	0.708	0.309	0.293	0.139
June - August	2	0.107	0.039	0.041	**0.019**	**0.025**	0.056
September - November	2	0.646	0.742	0.894	0.813	0.945	0.058
9 months	3	0.424	0.172	0.183	0.053	0.184	0.232
March - May	3	0.710	0.766	0.274	0.545	0.506	0.282
June - August	3	0.134	0.037	0.060	0.049	**0.007**	0.064
September - November	3	0.278	0.456	0.550	0.366	0.285	**0.021**

Acknowledgements

This work was partially funded by the European Commission in the context of the FP7 projects FIRST and FOC (Grant No. 257928 and 255987), by the Slovenian Research Agency through the research program Knowledge Technologies under (Grant P2-0103) and the project Influence of formal and informal corporate communications on capital markets (Grant No. J5-7387). We are grateful to Dragi Kocev for his help in the statistical evaluation of the results and Martin Saveski for his help with the implementation of the active learning algorithms.

References

[Abd07] Herv Abdi. Bonferroni and Šidák corrections for multiple comparisons. In Neil Salkind, editor, *Encyclopedia of Measurement and Statistics*, pages 103–107. Thousand Oaks (CA): Sage, 2007.

[BK09] Albert Bifet and Richard Kirkby. Data stream mining: A practical approach. 2009.

[Dem06] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[Fri37] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

[Fri40] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[GBH09] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[Gra69] Clive W.J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[ID80] Ronald L. Iman and James M. Davenport. Approximations of the critical region of the friedman statistic. *Communications in Statistics-Theory and Methods*, 9(6):571–595, 1980.

[IGD11] Elena Ikonomovska, João Gama, and Sašo Džeroski. Learning model trees from evolving data streams. *Data Mining and Knowledge Discovery*, 23(1):128–168, 2011.

[KZM14] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

[MGS16] Igor Mozetič, Miha Grčar, and Jasmina Smailović. Multilingual Twitter Sentiment Classification: The Role of Human Annotators. *PloS one*, 11(5):e0155036, 2016.

[Nem63] Peter B. Nemenyi. *Distribution-free Multiple Comparisons*. PhD thesis, Princeton University, 1963.

[NRR+16] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), San Diego, US (forthcoming)*, 2016.

[Scu10] David Sculley. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 979–988. ACM, 2010.

[SGLŽ13] Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. Predictive sentiment analysis of tweets: A stock market application. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, Lecture Notes in Computer Science Volume 7947, pages 77–88. Springer Berlin Heidelberg, 2013.

[SGLŽ14] Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. Stream-based active learning for sentiment analysis in the financial domain. *Information Sciences*, 285:181–203, 2014.

[SKG+15] Jasmina Smailović, Janez Kranjc, Miha Grčar, Martin Žnidaršič, and Igor Mozetič. Monitoring the Twitter sentiment during the Bulgarian elections. In *Proc. IEEE Intl. Conf. on Data Science and Advanced Analytics*, pages 1–10. IEEE, 2015.

[Sma14] Jasmina Smailović. *Sentiment analysis in streams of microblogging posts*. PhD thesis, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, 2014.

[SSSS07] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *Proceedings of the 24th International Conference on Machine Learning*, pages 807–814, 2007.

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

Investigating Exploratory Capabilities of Uncertainty Sampling using SVMs in Active Learning

Dominik Lang Daniel Kottke Georg Krempl Myra Spiliopoulou

dominik.lang@st.ovgu.de, {daniel.kottke, georg.krempl}@ovgu.de, myra@iti.cs.uni-magdeburg.de
KMD Group, Otto von Guericke University Magdeburg, Germany

Abstract

Active learning provides a solution for annotating huge pools of data efficiently to use it for mining and business analytics. Therefore, it reduces the number of instances that have to be annotated by an expert to the most informative ones. A common approach is to use uncertainty sampling in combination with a support vector machine (SVM). Some papers argue that uncertainty sampling performs badly due to missing exploration, others report good results using an SVM. This paper investigates whether uncertainty sampling is able to explore the data space due to the kernel trick used by the SVMs. Hence, we evaluate this on multiple synthetic and real datasets and the effects of parameter tuning and kernel selection for different evaluation criteria.

1 Introduction

In classification tasks, active learning methods intelligently select unlabeled instances to be labeled by an expert. The aim of active learning is to request labels from those instances that improve the classifier's performance the most [22]. To select the most useful instances, active algorithms should (1) explore the data space to find regions with unexpected labels and (2) do exploitation, i.e., to refine the classifier's decision boundary [8, 18].

One of the most commonly used methods is uncertainty sampling (US) which preferably samples instances near the decision boundary. This behavior is often considered as pure exploitation, without exploration in the strict sense. Some articles argue that this lack of exploration is the main drawback of US [4] and they claim that this behavior of US may explain its inferiority to the purely exploratory random sampling [11, 22]. Some authors therefore propose exploratory components for US [2, 20], while others capitalize the exploratory behavior of SVMs, letting US sample instances near the SVM decision boundary [12]. These characteristics suggest that the combination of SVMs and US might be promising. The rationale behind this is, that since SVMs use the kernel trick to learn a linear separation of two classes, sampling close to this decision boundary should be sufficient and exploration gets unnecessary. But is this truly the case?

In this paper, we investigate to what extend US, when combined with differently tuned SVMs, covers the original data space to learn a classification model. Furthermore, we propose an evaluation framework to measure the influence of this coverage (exploration) on the classification performance.

To demonstrate the interplay of US and a classifier in Fig. 1, we depict an exemplary one-dimensional dataset, where we show how the behavior of US is affected by the behavior of the classification algorithm. We see a dataset with two classes distributed across three clusters. A Bayesian classifier or even a very simple linear classifier would probably detect the boundary, whereupon US would *not* select instances from the cluster at the right. An SVM may, depending on the fit of its hyperparamters, place instances from the cluster at the right

In: G. Krempl, V. Lemaire, E. Lughofer, and D. Kottke (eds.): Proceedings of the Workshop Active Learning: Applications, Foundations and Emerging Trends, AL@iKNOW 2016, Graz, Austria, 18-OCT-2016, published at http://ceur-ws.org

inside the decision area, whereupon US would readily consider them. Since the behavior of an SVM depends on the chosen kernel and the SVMs hyperparameters, we investigate the dependencies between the exploratory capabilities of US with SVMs and the chosen kernel, also w.r.t. its tuning.

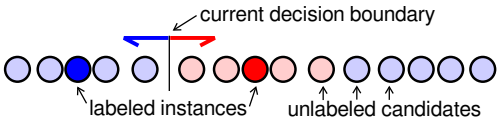


Figure 1: Two-class dataset with three clusters and a linear classifier.

The following section briefly summarizes the background and related work, followed by a description of the experimental framework in section 3 and the conducted experiments in section 4. Finally, we provide a discussion and conclude our results.

2 Background and Related Work

Active learning (AL) is a special area of machine learning, more precisely semi-supervised learning. A pool-based active learner successively selects and removes an instance $x \in \mathcal{U}$ from a large pool of unlabeled instances $\mathcal{U}$. An expert annotates this instance with a class label y. This information is added to the labeled set of instances $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x, y)\}$ which forms the training basis for a classifier [22].

Different strategies exist to determine which instances are chosen. A simple but naive approach is selecting instances at random, which has the benefit of potentially sampling a well distributed set of instances. A common approach is called uncertainty sampling (US), its underlying rationale being that the learner should select those instances the classifier is most uncertain about. For SVMs this might be the distance to the decision boundary (simple margin [23]), probabilistic classifiers might use the posterior estimates.

Uncertainty sampling is solely focused on exploitation of the data [4], i.e. it acquires the labels of instances that are useful to refining the decision boundary already assessed by the classifier. If the decision boundary proposed by the model is close to the actual decision boundary for the data, this process of refinement is likely to perform well. However, if there are unexplored regions in the data space with wrongly predicted labels far from the decision boundary, they will not be found. One approach to overcome this problem is to alternate between US for exploitation and an exploration component like random sampling [18, 16] or a semi-supervised learning method[13, 12]. Another approach is to improve the US selection criterion either by improving the uncertainty measure [23] or adding additional components to the criterion like expected model change [3, 12], representativeness and diversity [6, 10, 12] or confidence [15].

A Support Vector Machine (SVM) is a supervised learning model performing binary classification in a kernel-induced feature space [12]. For the binary classification problem (+/-), an SVM learns a decision hyperplane that separates the classes in the kernel-induced topological space by a maximal margin. Therefore, an SVM defines an objective function, where the sign indicates the predicted class of an instance x. The absolute value of this objective function represents the distance of an instance x to the SVM decision hyperplane. Some of the most commonly used kernel functions for support vector machines are the Polynomial, RBF, Sigmoid and Laplacian kernels [21].

In this context, the goal of an SVM is to achieve an optimal separation in the kernel-induced space. As the hyperplane is gradually 'refined', exploration becomes less and less important. Indeed, Tong and Koller [23] assume a quick reduction of the version space size using US since the current hypothesis of an SVM is roughly in the center of the version space. Hence, choosing an instance near the center should approximately split the version space in halves [23, 18]. There have been arguments against this, suggesting that the simple margin implementation of US fails to achieve this approximate halving of the version space in some cases [17, 12]. An aspect that has to be considered in the discussion about US with SVMs and exploration is the question of how the SVM hyperparameters are chosen, since it has been shown that they strongly influence the performance of the active learner [5, 14]. A problem with using common methods to determine appropriate SVM hyperparameters, such as tuning them through grid search, in the context of AL is that the required labeled data needed for such approaches is not available at the start of the AL process. However the most commonly used kernel function for SVM AL of those mentioned above is RBF [6, 7, 13, 16, 18], which requires only one parameter. This popularity might be due to the fact that the RBF kernel has been shown to learn concepts well even with few available training examples [17] (as is usually the case in active learning scenarios) and tends to explore the feature space

more than its alternatives, although it can be more sensible to noise in the data [5]. Lin and Lin recommend using the RBF kernel for most applications, but also suggest that the Sigmoid kernel can behave similar to RBF given certain hyperparameters [14].

3 Evaluation Framework

In our experiments, we use pool-based active learning as described in Sec. 2 and stop learning after 50 label acquisitions (budget $B = 50$). For uncertainty sampling (US), we use the most commonly used simple margin (sm) implementation for SVMs by [23]. To compare the exploratory behavior, we additionally use a random sampler with each classifier (SVM+kernel+tuning). Since an SVM requires at least one instance of each of the two classes for training the labeled set $\mathcal{L}$ set was initialized accordingly.

In Alg. 1, we show our framework to evaluate our active learning experiments. For a given data model $\mathcal{M}$, we perform US using SVMs with different kernels as defined in [21], namely a polynomial kernel, a radial basis function kernel, a sigmoid kernel, and a Laplacian kernel.

To tune the hyperparameters of each classifier (SVM+kernel), we perform a grid search on a separate tuning set D_{tune}. This is generated by selecting B labeled instances from the data model $\mathcal{M}$, to optimize the classifier according to the final number of labels. The labels in the D_{tune} set are used solely for tuning and validating the hyperparameters. To avoid a bias of the model due to overfitting, the labels in D_{tune} are not used in the training or testing of the active learner. Based on the hyperparameter optimization, we select for each kernel (1) the best parameter setting and (2) a parameter setting that achieved medium results. Surely, the classification performance will probably decrease by selecting a non-optimal parameter setting. Here, we want to investigate the influence of the parameter choice for doing exploitation. Hence, we get two different classifiers for each kernel $C_{k,t}$. The search space of the hyperparameter optimization is given in Tab. 1: γ denotes the kernel coefficient, C is the penalty parameter of the error term of the SVM, d is the degree of the polynomial kernel and $coef_0$ is the independent term of the kernel function [19].

Parameters	Value Space
γ	$\{1e^{-5}, 1e^{-3}, 1e^{-2}, 1e^{-1}, 0.2, 0.4\}$
C	$\{1, 10, 100, 200, 400, 1000\}$
d	$\{1, 2, 3, 4\}$
$coef_0$	$\{0.0, 0.1, 0.2, 0.3\}$

Table 1: Search space for hyperparameter tuning

The experiments are conducted based on a set of 200 different seeds that were used for generating resp. splitting the datasets. For each seed, we generate a test set D_{test} which we use for evaluation and a training set $\mathcal{U}$ consisting of solely unlabeled instances. As mentioned above, the labeled set $\mathcal{L}$ is initialized with one instance from each of the two classes. Then, the active learner chooses B instances successively to be labeled. The unlabeled instance is removed from $\mathcal{U}$ and added (with the corresponding label) to $\mathcal{L}$.

As the model $\mathcal{M}$ is not explicitly given for the real world datasets, we split B instances from the real datasets for hyperparameter tuning. Then, half of the remaining instances is used for testing, the other half for training.

In the evaluation step, we focus on two aspects: (1) the classifier's performance, and (2) the amount of exploration. To address the classifier's performance, we determine the hold-out accuracy on the test set D_{test} for each budget step resulting in a learning curve for each active learner and each classifier. The exploration is addressed by a score motivated by [6]. Here, we determine the average euclidean distance from each instance $x \in D_{test}$ to the nearest labeled instance $x_l \in \mathcal{L}$ (see Eq. 1). This score (avg. min. distance) is determined after each acquisition, which again results in a learning curve. A low value indicates a good coverage of the data space, high values indicate that labeled instances are far away. Per definition, this score is decreasing over time, as more labeled instances are added subsequently to the labeled set.

$$\frac{1}{|D_{test}|} \sum_{x \in D_{test}} \min_{x_l \in L} ||x - x_l||_2 \tag{1}$$

Regarding the expected results of the experiments, it is difficult to make predictions as to the differences in performance and exploration between the different kernel functions. Generally, it is to be expected that the rbf and laplacian kernel will show a similar but not identical behavior based on their similarity. Among the artifical datasets the ones referred to as 'Three Cluster Datasets' (Sec. 4.1) are designed with an expected result in mind:

Data: data model $\mathcal{M}$, seeds S, budget $B = 50$, SVM $\mathcal{C}$, active method US
begin
  for $k \in \{\text{RBF}, \text{Poly}, \text{Sigm}, \text{Laplace}\}$ **do**
    for $t \in \{\text{best}, \text{middle}\}$ **do**
      $D_{tune} \leftarrow \text{getLInst}(\mathcal{M}, B)$;
      $p_{k,t} \leftarrow \text{gridSearch}(\mathcal{C}(k), D_{tune}, t)$;
      $C_{k,t} \leftarrow \text{initializeClassifier}(\mathcal{C}(k), p_{k,t})$;
      for $s \in S$ **do**
        $D_{test} \leftarrow \text{getLInst}(\mathcal{M}, 350, s)$;
        $\mathcal{L} \leftarrow \text{getOneLInstPerClass}(\mathcal{M}, s)$;
        $\mathcal{U} \leftarrow \text{getUInst}(\mathcal{M}, 350, s)$;
        for $i \in \{1, \ldots, B\}$ **do**
          $C^* \leftarrow \text{trainClassifier}(C, \mathcal{L})$;
          $x^* \leftarrow \text{selectInst}(\text{US}, \mathcal{U}, C^*)$;
          $y^* \leftarrow \text{getLabel}(\mathcal{M}, x^*)$;
          $\mathcal{U} \leftarrow \mathcal{U} \setminus \{x^*\}$;
          $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x^*, y^*)\}$;
          evaluate$(C^*, \mathcal{L}, D_{test})$
        end
      end
    end
  end
end

Algorithm 1: Evaluation framework

since one of the three clusters is unknown to the active learner at the beginning because only two labels are given in the initial L set, we expect that learners using US with a linear classifier will find this cluster very late. However, as these clusters are generated with Gaussian distributions, it is likely that both the rbf and laplacian kernel will perfom better by using this implicit information.

4 Experimental Evaluation

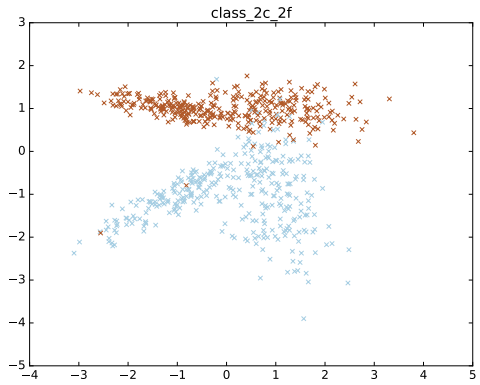
In the experimental evaluation, we use 10 different data models resp. datasets to investigate the exploratory capabilities of uncertainty sampling and SVMs (see Fig. 2a-2f). Therefore, we used a cluster system running the NeuroDebian system [9]. First, we discuss the results on an artificial dataset consisting of three clusters and investigate, if uncertainty sampling acquires labels in every cluster. We then try to generalize the findings on further artificial datasets from a standard library and on real data.

4.1 Three Cluster Dataset

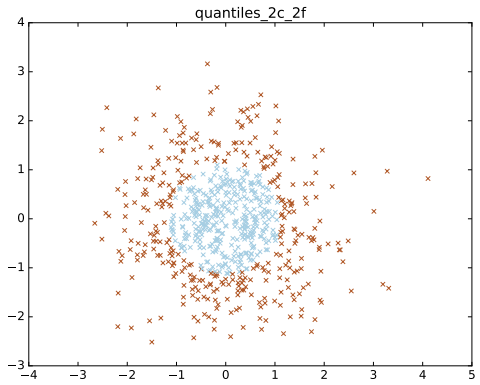
The first model consists of three Gaussian clusters, two clusters of class one and the other cluster of the second class. To get more expressive results, we use three different standard deviations (equal for all clusters) to have well separated clusters as well as overlapping ones (high standard deviation (overlapping): gaussian_2dp0, medium: gaussian_2dp1, low (well-separated): gaussian_2dp2). Since 2 of the 3 clusters are sampled in the initialization step, the learner's task is to discover the remaining cluster in order to perform well.

If a learner-classifier combination was able to find the remaining cluster is summarized in Tab. 2. In each cell, we show the percentage that the active learner found the unknown cluster within the first B label acquisitions across all 200 trials. Every classifier has a specific kernel and a set of hyperparameters coming from the grid search tuning. Here, we choose the best performing hyperparameters (best) as well as non-optimal ones (middle).

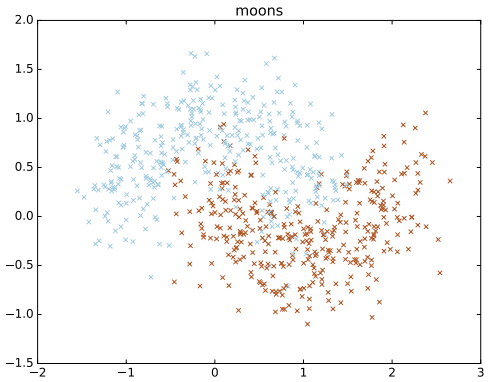
Choosing the best set of parameters, the SVMs with a laplacian and an rbf kernel are able to discover the remaining cluster reliably, whereas the discovery rate of those using a polynomial or sigmoid kernel is less than 25%. Applying not-optimal parameters, the detection rate of an SVM with an rbf decreases, but the rate of polynomial SVMs increases to approx. 50% and of sigmoid SVMs to higher than 80%. This exploratory behavior is also indicated by the average minimal distance score in Fig. 3b, 3d. Here, a high discovery rate correlates with a low value for the avg. min. distance. In this specific dataset, there is a clear correlation of the exploration score, resp. the discovery rate, and the learning curves in Fig. 3a. The previously mentioned expectation of the



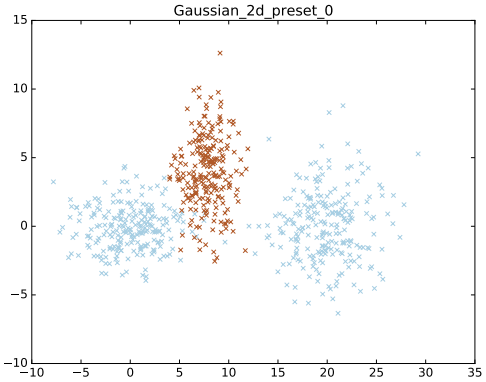
(a) Example of the 'classification' dataset



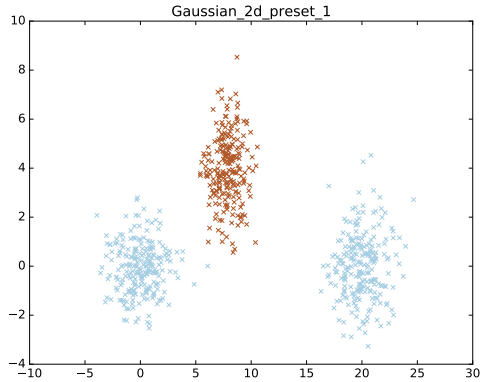
(b) Example of the 'quantiles' dataset



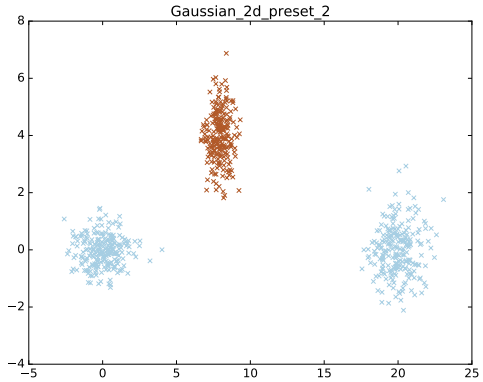
(c) Example of the 'moons' dataset



(d) Example of the 3 cluster 'Gaussian' dataset, preset 0

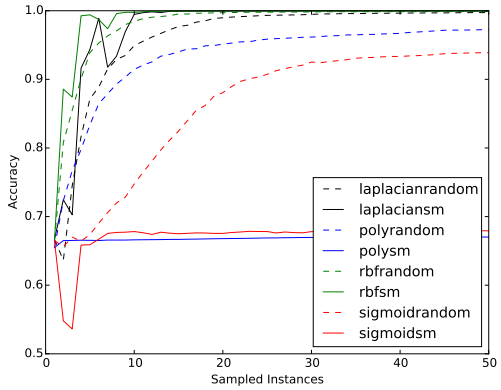


(e) Example of the 3 cluster 'Gaussian' dataset, preset 1

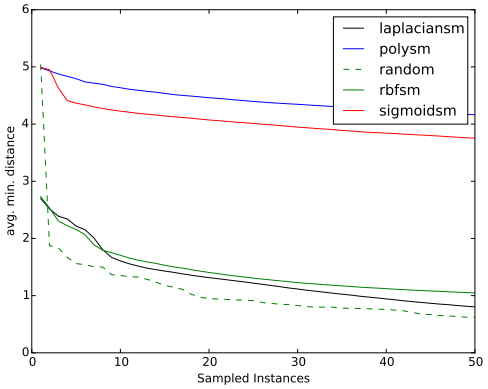


(f) Example of the 3 cluster 'Gaussian' dataset, preset 2

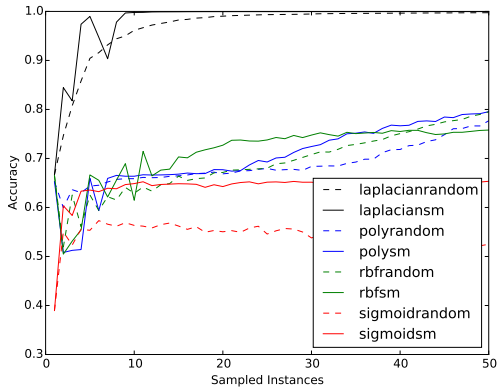
Figure 2: Examples for the various artificial datasets



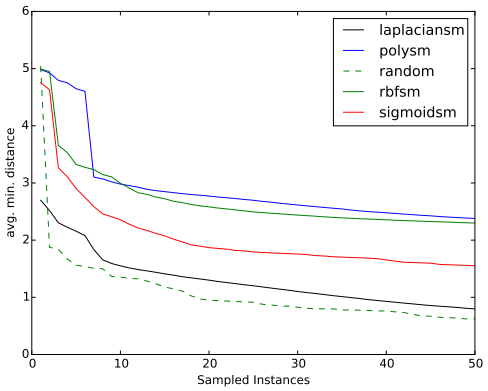
(a) Learning curve with best parameters



(b) Avg. min. dist curve with best parameters

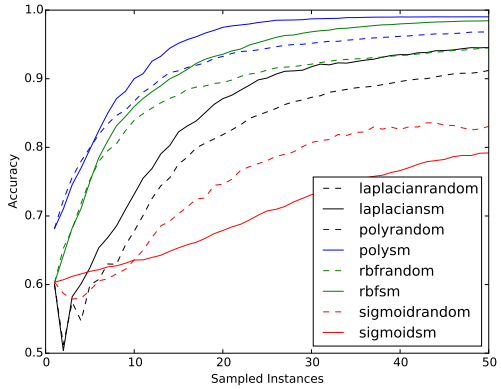


(c) Learning curve with middle parameters

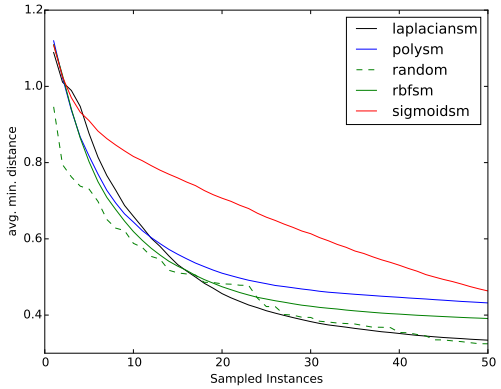


(d) Avg. min. dist curve with middle parameters

Figure 3: Results for the Gaussian_2dp1 dataset

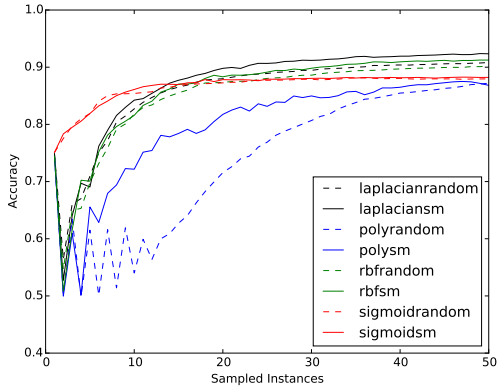


(a) 'Quantiles' learning curve

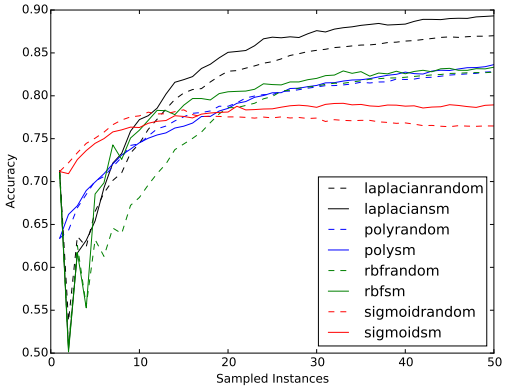


(b) 'Quantiles' avg. min. dist curve

Figure 4: Results using best parameters on the generated data using the scikit-learn functions

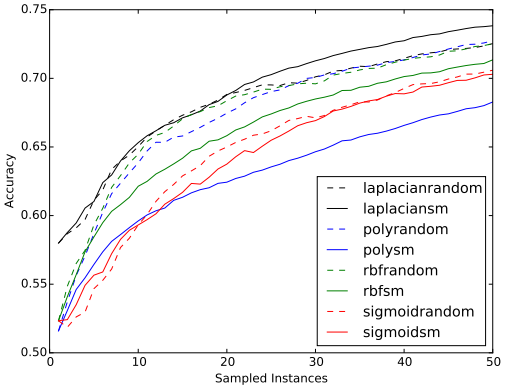


(a) 'Classification' learning curve

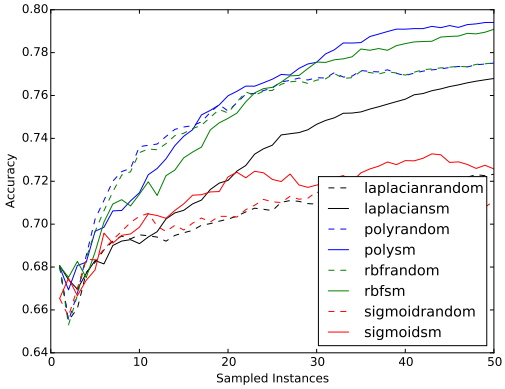


(b) 'Moons' learning curve

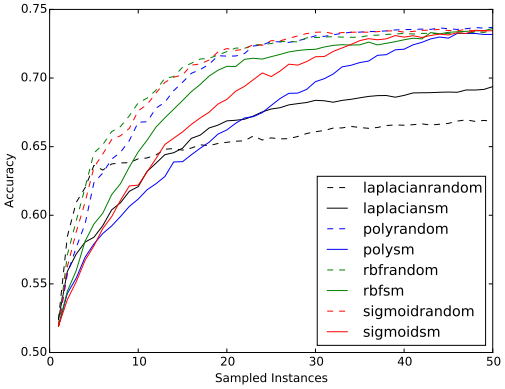
Figure 5: Results using best parameters on the generated data using the scikit-learn functions



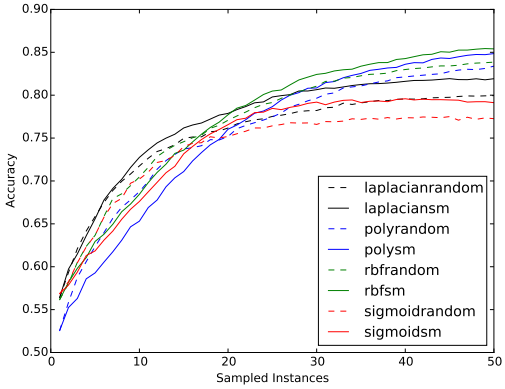
(a) Abalone



(b) Mammo



(c) Haberman



(d) Vertebral

Figure 6: Learning curves on the real-world datasets with best parameters

Params	Kernel/Data	p0	p1	p2
best	laplace	100%	100%	100%
	poly	23%	1%	0%
	rbf	100%	100%	100%
	sigmoid	18%	7%	22%
middle	laplace	100%	100%	100%
	poly	67%	49%	56%
	rbf	64%	56%	56%
	sigmoid	98%	94%	82%

Table 2: Discovery rate of the unknown cluster on the gaussian_2d dataset for the different presets and learner configurations using uncertainty sampling.

rbf and laplacian kernel learners to perform better based on the Gaussian distributions of the data was shown to be correct. The accuracy curves of the random learners can also be used to find out, if the tuned SVM is able to fit the decision boundary. Using the best hyperparameters, all random active learners achieve a performance higher than 0.9 accuracy which means that these are generally able to perform well. Here, a lack of exploration also leads to bad performance.

Using non-optimal parameters, the discovery rate might increase, but the ability to find an appropriate boundary decreases (except for the laplacian kernel), which is not surprising. Hence, an increase of exploration by changing the kernel does not necessarily increase the accuracy of a classifier, as the classifier is not able anymore to fit the decision boundary accordingly. Interestingly, uncertainty sampling now performs better with non-optimal hyperparameters compared to random in general. This observation could be critical to active learning research: In active learning research, the main focus is to compare different active learning algorithms. Hence, they fix a classifier and a hyperparameter setting (however this is determined) and compare the active methods with each other. This experiment shows that random and US methods using the same SVM with the same kernel can change their order by just varying their hyperparameters.

Note that these experiments are based on normal distributions. We expect circular shapes which might favors the performance of rbf or laplace kernels. Hence, we perform more experiments using less structured data.

4.2 Synthetic Datasets

The results of the synthetic datasets are given in Fig. 4a-4b and Fig. 5a-5b. All three datasets are provided by the scikit-learn library [19], namely 'quantiles', 'classification' and 'moons'. Here, we only show the results for the best tuning hyperparameters.

In the exemplary Fig. 4a-4b, a fast decreasing avg. min. distance (high exploration) in the early steps indicates a fast improvement in terms of accuracy. This tendency is also visible on the other datasets which we provide at our companion website[1]. This is also indicated by the observation that the random sampler is similar or better in the early steps (except for sigmoid on Quantiles) but beaten later by the uncertainty sampling (simple margin sm) method. This has also been shown by various other authors that an exploration phase in the beginning is beneficial.

Finding the best SVM+AL combination remains difficult: On Quantiles, the polynomial SVM in combination with simple margin is superior. On Classification and Moons, the winner is the laplacian SVM + simple margin.

4.3 Real-world datasets

The real-world datasets are chosen from the UCI machine learning repository [1] and include the 'Abalone', 'Haberman', 'Mammo' and 'Vertebral'. Their characteristics are summarized in Tab. 3. We transformed nominal attributes into multiple binary attributes, numerical attributes were normalized to $[0, 1]$.

On these datasets, we choose to show the learning curves of the well-tuned SVMs in Fig. 6a-6d as one would do this in practice. The winning kernel differs very much across the datasets. On Abalone, the laplacian kernel was superior; on Mammo it was the polynomial kernel and on Haberman and Vertebral, the rbf kernel was best.

On Mammo and Vertebral, we observe the same situation as in Sec. 4.2: random sampling outperforms simple margin in the early learning stage and the latter catches up or surpasses random sampling later on. The results on Abalone in Fig. 6a show that the laplacian kernel is beneficial. Even more interesting is that random outperforms

[1] `http://kmd.cs.ovgu.de/res/explore-us/`

Name	Attributes	Size
Abalone	8	4177
Haberman	3	306
Mammo	11	830
Vertebral	6	310

Table 3: Characteristics of the real-world datasets

the simple margin strategy for every other kernel. The same appears on Haberman in Fig. 6c. Only the laplacian kernel single margin strategy outperforms its random competitor, but the performance of both are far less then all others.

To summarize, there are quite a lot of cases where a solely exploratory strategy (random) outperforms the uncertainty sampling approach which was combined with a pre-tuned SVM to potentially add some exploration.

5 Discussion and Conclusion

In this paper, we investigated the exploratory capabilities of uncertainty sampling (US) in combination with different support vector machines (SVMs). We described an evaluation framework and tested multiple synthetic and real datasets using this framework. Furthermore, we proposed to use the average minimum distance as an indicator for exploration.

Although SVM and US are seen as a promising combination for active learning, it does not mitigate the lack of exploration to which the inferior performance of US is accredited to. The exploiting behavior of US in the kernel-induced space ends up looking similar to exploratory behavior in the feature space, yet it does not perform actual exploration in a strict sense.

If non-optimal SVM hyperparameters are used, the exploitation that US performs in the kernel-induced space becomes less precise which can lead to the behavior showing more exploratory characteristics. However, in a strict sense this is more a misbehavior of the exploitation than real exploration of the data space. Hence, we conclude that merely choosing a SVM to perform US does not replace a dedicated exploratory component.

The experimental results affirm that exploration in the beginning of the active learning process is indeed beneficial to the classification performance. Furthermore, they indicate that the hyperparameter tuning is critical to classification performance. We propose to validate multiple hyperparameters in an evaluation of active methods to get rid of the bias induced.

Acknowledgements

We thank Michael Hanke and Alex Waite for support by providing the cluster for our computations and Pawel Matuszyk for the discussions on that topic.

References

[1] Arthur Asuncion and David J. Newman. UCI machine learning repository, 2015.

[2] Christian Beyer, Georg Krempl, and Vincent Lemaire. How to select information that matters: A comparative study on active learning strategies for classification. In *Proc. of the 15th Int. Conf. on Knowledge Technologies and Data-Driven Business (i-KNOW 2015)*. ACM, 2015.

[3] Wenbin Cai, Ya Zhang, Siyuan Zhou, Wenquan Wang, Chris Ding, and Xiao Gu. Active learning for support vector machines with maximum model change. In *Machine Learning and Knowledge Discovery in Databases*, pages 211–226. Springer, 2014.

[4] Gavin C Cawley. Baseline methods for active learning. In *Active Learning and Experimental Design@ AISTATS*, pages 47–57, 2011.

[5] Nicolas Cebron. *Aktives Lernen zur Klassifikation großer Datenmengen mittels Exploration und Spezialisierung*. PhD thesis, 2008.

[6] Gang Chen, Tian-jiang Wang, Li-yu Gong, and Perfecto Herrera. Multi-class support vector machine active learning for music annotation. *International Journal of Innovative Computing, Information and Control*, 6(3):921–930, 2010.

[7] Husheng Guo and Wenjian Wang. An active learning-based svm multi-class classification model. *Pattern Recognition*, 48(5):1577–1597, 2015.

[8] Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov, editors. *Active Learning Challenge*, volume 6 of *Challenges in Machine Learning*. Microtome Publishing, 2011.

[9] Yaroslav O Halchenko and Michael Hanke. Open is not enough. let's take the next step: an integrated, community-driven computing platform for neuroscience. *Frontiers in neuroinformatics*, 6, 2012.

[10] Tianxu He, Shukui Zhang, Jie Xin, Pengpeng Zhao, Jian Wu, Xuefeng Xian, Chunhua Li, and Zhiming Cui. An active learning approach with uncertainty, representativeness, and diversity. *The Scientific World Journal*, 2014, 2014.

[11] Daniel Kottke, Georg Krempl, Dominik Lang, Johannes Teschner, and Myra Spiliopoulou. Multi-class probabilistic active learning. In *ECAI 2016*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 586 – 594, 2016.

[12] Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):313–326, 2014.

[13] Yan Leng, Xinyan Xu, and Guanghui Qi. Combining active learning and semi-supervised learning to construct svm classifier. *Knowledge-Based Systems*, 44:121–131, 2013.

[14] Hsuan-Tien Lin and Chih-Jen Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. *submitted to Neural Computation*, pages 1–32, 2003.

[15] Pabitra Mitra, CA Murthy, and Sankar K Pal. A probabilistic active support vector learning algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(3):413–418, 2004.

[16] Nir Nissim, Mary Regina Boland, Robert Moskovitch, Nicholas P Tatonetti, Yuval Elovici, Yuval Shahar, and George Hripcsak. An active learning framework for efficient condition severity classification. In *Artificial Intelligence in Medicine*, pages 13–24. Springer, 2015.

[17] Nir Nissim, Robert Moskovitch, Lior Rokach, and Yuval Elovici. Detecting unknown computer worm activity via support vector machines and active learning. *Pattern Analysis and Applications*, 15(4):459–475, 2012.

[18] Thomas Osugi, Deng Kim, and Stephen Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Tobias Reitmaier and Bernhard Sick. Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4ds. *Information Sciences*, 230:106 – 131, 2013. Mobile and Internet Services in Ubiquitous and Pervasive Computing Environments.

[21] Matthias Rupp. Machine learning for quantum mechanics in a nutshell. *International Journal of Quantum Chemistry*, 115(16):1058–1073, 2015.

[22] Burr Settles. *Active Learning*. Number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[23] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

Active Subtopic Detection in Multitopic Data

Benjamin Bergner
benjamin.bergner@st.ovgu.de

Georg Krempl
georg.krempl@ovgu.de

Knowledge Management & Discovery, Otto von Guericke University Magdeburg, Germany

Abstract

Subtopic detection is a useful text information retrieval tool to create relations between documents and to add descriptive information to them. For the task of detecting subtopics *with user guidance*, clustering by intent (CBI) has recently been proposed. However, this approach is limited to single-topic environments. We extend this approach for interactive subtopic detection in multi-topic environments, and for the incorporation of positive and *negative* user feedback. Our multi-topic clustering by intent (MCBI) approach iteratively constructs so-called similarity sets of documents within the same topic, derives candidates for new subtopics and actively queries feedback from the user, which is then used to refine the subtopic and similarity sets in the next iteration. For evaluation, we construct a corpus of the Wikipedia articles for the 4309 most common English nouns, comprising a broad range of different topics. Our MCBI approach is compared with the recently proposed CBI approach and random sampling. Each approach is evaluated based on the number of subtopics that are found in the same predefined, closed topic (countries). The results show that MCBI finds up to 137% and 445% percent more correct subtopics than random term selection or CBI, respectively.

1 Introduction

A business user might intend to group text documents, such as support logs or customer reviews, into meaningful subsets that correspond to the different subtopics addressed in the texts. Or, as further illustrative example, consider a user wants to cluster documents by using the different types of sport as subtopics. As pointed out in [7], it might be too tedious or even impossible to provide a priori an exhaustive list of possible subtopics, to label documents manually, or to experiment with different parameters until the desired clustering is obtained. However, it is comparatively easy for the user to illustrate the intended clustering by providing one (or a few) exemplary subtopics, for example by giving a cluster of tennis-related documents. Given the documents and initial subtopic(s), the clustering algorithm's task is to construct candidates for new subtopics, to actively seek the user's confirmation or rejection of these candidates as members of the intended topic, and to extend the clustering accordingly. Thus, this problem, recently described as *clustering by intent* in [7], is a combination of an active, incremental clustering task with very weak, interactive supervision.

A first approach for this task is provided in [7]. However, this approach is limited to single-topic environments. That is, documents unrelated to the intended topic might confuse the approach, like for example food-related

In: G. Krempl, V. Lemaire, E. Lughofer, and D. Kottke (eds.): Proceedings of the Workshop Active Learning: Applications, Foundations and Emerging Trends, AL@iKNOW 2016, Graz, Austria, 18-OCT-2016, published at http://ceur-ws.org

documents when the intent is deriving animal-related subtopics. Furthermore, the approach is focused on the user's confirmation of candidates, neglecting negative feedback by the user.

In this paper, we extend this approach to the interactive subtopic detection in multi-topic environments, and for incorporating positive and *negative* user feedback. Our **M**ulti-topic **C**lustering **B**y **I**ntent (MCBI) approach iteratively constructs so-called similarity sets, which comprise solely documents of the same topic. Then, it derives candidates for new subtopics, for which it actively queries feedback from the user. Information on the confirmed as well as on the rejected subtopics is then incorporated into the clustering model and used in subsequent iterations.

The remainder of this paper is structured as follows: in the next Section 2, we provide a more detailed background of clustering by intent and review the related work. Our method is presented in Section 3, followed by the experimental evaluation in Section 4. The paper closes with concluding remarks and an outlook to future work in Section 5.

2 Background and Related Work

We first provide a more formal definition of clustering by intent, in order to facilitate the subsequent discussion of the related work. Clustering by intent [7] corresponds to an interactive clustering task, where a given set of documents D should be partitioned into clusters. Each document $d \in D$ is represented by its bag of word feature vector $\vec{w}$. Each cluster c_{st} is defined and labelled by its corresponding subtopic st. The clusterer has neither access to an explicitly given similarity function, nor to an expert willing to perform a tedious tuning of such a function, nor to single similarity measurements between objects. Instead, the clusterer is provided with one (or few) exemplary clusters as an indication of the human *intent*. Its task is then to actively propose the user further clusters that should correspond to the same intent. Documents belonging to clusters that the user has already confirmed constitute the labeled set $L \subseteq D$. The unlabelled set $U \subseteq D$ comprises all remaining documents. This is illustrated in Fig. 1 a, where the two subtopics c_{Dog} and c_{Cat} as well as their corresponding documents (blue and yellow dots) have already been identified, while other documents (black dots) remain unlabelled. The approach proposed in [7] for CBI iteratively (1) trains a probabilistic multi-class classifier that discriminates between the known subtopics, (2) computes for each unlabelled document its confidence as the margin between its highest and second highest posterior estimate, (3) builds a so-called residual set that comprises the low-confidence documents, (4) selects the terms with highest precision in separating L from U as subtopic candidates, (5) queries their membership to the intent from the user, updates the sets and restarts the process again. This is illustrated in Fig. 1 b, where the most uncertain documents are used to form a residual set. The pseudocode for our implementation of this clustering by intent algorithm is given in Fig. 5 in the appendix. This clustering by intent is related to areas in constraint clustering and active learning, as well as to novel class detection, subgroup discovery, and topic detection, for which we will briefly review the most related literature below.

Within the rich literature on clustering, constraint clustering is of particular relevance. A recent survey on constrained (also denoted semi-supervised) clustering is provided in [5]. As an example, text clustering is given, where the objective is to group texts according to similarities in e.g. their content or their authors. The expert might provide must-link constraints for documents that are deemed to be similar, or cannot-link constraints for documents that are different. In [5], three categories are named, search based (constraint-based), where the solution space is modified e.g. by penalizing violations of constraints, distance based (similarity based), where the similarity function is modified to consider the constraints, or by hybrids of both. Active approaches, e.g. for selection of informative document pairs for which user feedback is queried [1], and more recently also noisy constraints have been researched [18]. However, in contrast to [7, page 33], constrained clustering assumes the similarity measure to be a priori specified, and the pairwise constraints require to query information on the document level, rather than on the subtopic level. In (inter)active clustering (e.g. [9, 6, 11]), not all similarities (or distances) are known a priori. Rather, the approach actively selects similarity measurements and queries them from an oracle. Recent works comprise the so-called interactive (hierarchical) clustering proposed in [11] and the active hierarchical clustering in [6]. However, these (inter)active clustering approaches require the similarity measure to be known a priori or similarity measurements to be provided by the user.

Many works in active learning literature addresses classification, where labels for instances are queried from the user [15]. Although some of these works address active learning on text data (e.g. sentiment classification in [10]), they assume the set of labels to be known and are therefore not applicable here. Similarly, in [14], active learning in text categorization is studied. The authors propose a two-fold process, where active learning is done

both on the document-level (querying document's category, i.e. its label), and on the term level. The latter corresponds to asking the oracle about the importance of features, i.e. asking for the most predictive words. However, the text categorization problem is posed as one-versus-the-rest classification problem, where the single category of interest is initially known. In contrast, in clustering by intent, the categories are not known a priori but rather need to be learned on the way. Nevertheless, the approach in [14] might be used as post-processing for labelling all documents, once the set of categories has been determined by a clustering-by-intent approach.

Another related field of supervised machine learning research is subgroup discovery ([17], see e.g. [2] for a recent survey). Given a population of individuals, its objective is finding as large as possible subgroups that show a distributional unusualness with respect to a certain property of interest [17]. The interestingness of a pattern is measured by a quality function [2], which is in general exploiting the target concepts' distribution. A common exemplary quality function is to combine a measure for the subgroup's size with its from the whole population in the target concept value. There exist interactive subgroup discovery frameworks (e.g. [8] that allow the expert to affect the attributes that are used for learning. However, the quality function and the target concept (i.e. class attribute) need to be specified a priori.

Furthermore, novel class detection (e.g. [12]) in data streams is a related area of research. There, the objective is to detect novel classes, whose instances differ from those of already known classes. Some approaches are passive (e.g. [12]), while others actively query labels from the user (e.g. [13, 3]). However, similar to clustering these approaches rely on an a priori specified similarity function. Furthermore, these approaches focus on finding emerging topics, requiring a chronological ordering of instances. In contrast, in our setting all documents are provided at once, before starting to detect subtopics interactively. Likewise, one-class classification, outlier detection and anomaly detection approaches are not applicable here [7], as their objective is the detection of abnormal, rare data points that differ from the data available at training time. Thus, they detect small rather than large groups, and do not aim to provide a meaningful clustering.

In topic detection and modelling, the aim is to discover the most relevant topics in text documents. An example is [4], where frequent topics in twitter messages are discovered. Another, more recent, is the discovery of the most interesting topics (and subtopics) in students' messages in a learning management system [16]. However, these approaches rely on temporal information. For example, the detected "emerging topics" in [4] and the "spike topics" in [16] correspond to terms have gained in frequency. More related is the detection of "chatter topics" in [16]. These chatter topics are sustained discussion topics, which are the most frequent words that are not in a set of so-called "DumpTerms". However, this requires providing a list of DumpTerms, which is unfeasible for large vocabularies.

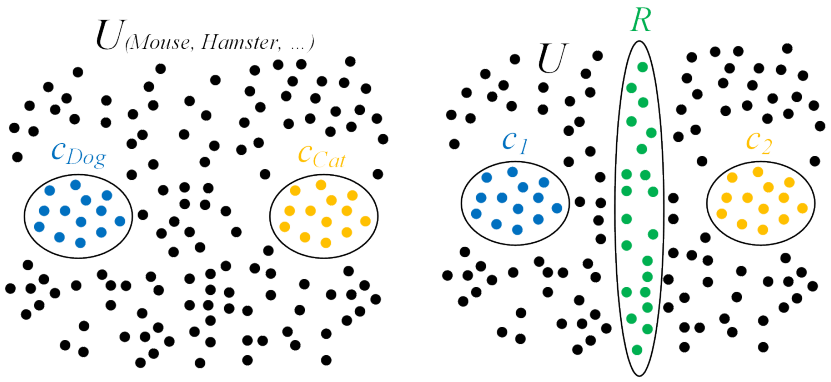


Figure 1: Dots represent single documents, some assigned to clusters of initially given subtopics. **(a) Left:** Example topic *animals*; c_{Dog}, c_{Cat} contain documents with subtopics *dog* and *cat*, respectively, remaining documents are unlabeled and might contain further animal-related subtopics. **(b) Right:** Building the residual set R of documents who's classification is uncertain.

3 Multi-Topic Clustering by Intent

We first explain the main ideas of our approach, before providing details. Similar to CBI [7], we use an uncertainty-related measure to construct a *residual set* of documents that are not certainly classified into one known subtopic. However, our aim is building a residual set consisting solely of documents from the intended topic. Thus, for constructing the residual set, we include a threshold on the document's *similarity* with respect to the union of known subtopics.

Having this residual set of documents, we extract the words that are used therein. In contrast to CBI, we exclude all words that have previously been rejected as subtopics. Furthermore, some terms are not specific for a subtopic, but are rather related to the topic as a whole. Therefore, we also exclude such words that occur frequently in all subtopics. We then compute a score for the remaining words. Similar to CBI, we consider the discriminatory power of a word, but we also consider the frequency of its co-occurrence with rejected words. The highest-ranking words in this combined score are proposed as new subtopics. Upon querying the users decision, the lists of rejected words and accepted subtopics are updated.

The pseudocode of our **M**ulti-Topic **C**lustering **b**y **I**ntent (MCBI) approach is given in Figure 3. Its input are lists of unlabelled U as well as labelled L documents. Using a bag-of-words representation over the vocabulary V, each document d therein has a feature vector $\vec{d}$. Furthermore, the current clustering C is given as a set of subtopics $s_1, s_2, \cdots$, each subtopic consisting of one or more words. Finally, a list of previously rejected subtopics V_{REJ} is given. These four lists are updated and returned by the algorithm.

The first step of the algorithm (lines 2–12) is the construction of the *residual set*. This is done by first training a Multinomial Naive Bayes Classifier (MNB) to classify documents into the different known subtopics (line 2). Iterating over each document u in the unlabelled set U (lines 3–12), we use this classifier to obtain a posterior-based score $p_{s|u}$, normalized by the vocabulary size, for each known subtopic s (line 6). For simplicity and speed, we consider in subsequent calculations solely the scores of the best (s) and second-best (s') subtopic (line 6). For better comparison with CBI, our algorithm uses the same uncertainty-based strategy to select the most ambiguous unlabelled documents. Thus, MCBI calculates for each unlabelled instance the difference between the score of its best s and second best s' subtopic (line 7)[1]:

$$uncertainty_u = p_{s|u} - p_{s'|u} \tag{1}$$

In a multi-topic corpus, one might distinguish three types of documents in the residual set: first, documents belonging to one of the *known subtopics*. Second, documents belonging to a *new subtopic* of the intended topic, and third, documents belonging to a *different topic*. Our aim is to identify documents of the second kind. For illustration, consider the example of the intended topic "sports", with known subtopics "soccer" and "tennis", the yet undiscovered subtopic "hockey", and the unrelated topic "building" (see Fig. 2 (a)). A classifier using the frequency of words like "soccer" or "tennis" might differentiate documents from the first type from the rest. However, it fails in distinguishing between the second and the third type, as both contain equally likely soccer- or tennis-specific words (see Fig. 2 (b)). Nevertheless, the second type of sports-related documents contain words that occur in both known subtopics, as for example the word "athlete". Such topic-specific words are more frequent in the second (and first) type, than they are in the third (see Fig. 2 (c)). Thus, the score over all subtopics will be greater for topic-related documents. We exploit this to remove topic-unrelated documents from the residual set by applying a threshold on the sum of scores (which we denote as *similarity*, line 8):

$$similarity_u = p_{s|u} + p_{s'|u} \tag{2}$$

Finally, all documents that are ambiguous with respect to being classified into one existing subtopic (high uncertainty) and are sufficiently similar to the union of the two subtopics in question (high similarity), are selected for the residual set R (lines 9–11 in Figure 3), i.e. R is a subset of the one computed with equation 1 by applying the similarity condition in order to guide pre-selected documents to the desired topic.

Having the residual set of topic-related documents, the next task is selecting candidate words for new subtopics, for which the user's feedback is then queried. Thus, we iterate over each word w occurring in a document of the residual set (lines 15–24). However, the most frequent words in the residual set might be specific for the topic, but not for a subtopic. Continuing the sports-topic above, an exemplary word is "athlete". Excluding such *topic-specific* words from being suggested as subtopic candidates saves annotation time and prevents them from being added to the list of rejected terms V_{REJ}. Such topic-specific words are frequent in the documents of the

[1] Here, entropy might be considered as another uncertainty measure.

labelled set L, too. Thus, we exclude any word that occurs in more than half of the labelled documents[2] (line 16). Here, the subfunction $getDocumentsWithWords(D, W)$ returns all documents in D that contain words W. Furthermore, our algorithm makes use of negative feedback in different ways. First, by excluding w in case it corresponds to a previously proposed but rejected subtopic (line 17). Second, by computing a score that considers how often this word co-occurs with rejected words. Thereby, we aim to exclude words that are specific to unrelated topics. For each rejected subtopic $n \in V_{REJ}$, we compute the co-occurence frequency of w and n and divide it with the frequency of n. Then, we use the maximum of this relative co-occurence frequencies as a *reject score* in line 20. Next, we compute a score reflecting the discriminatory power of the word w, similarly to [7]. This *discriminative score* corresponds to the difference between the number of unlabelled documents that are classified by the word w, subtracted by the number of labelled documents that are (wrongly) classified by w as discriminative feature (line 21). Finally, the best scoring among the candidate words are selected and added to the query list Q (line 25).

The user's feedback on these subtopic candidates is queried (line 27), which might result in an *accept* or *reject* decision. With the accepted words the list of subtopics and clustering is updated C (line 28), and their corresponding documents are moved from U to L (lines 29–30). Rejected words are simply added to the rejected word list V_{REJ} (line 31), and the resulting lists U,L,C,V_{REJ} are returned.

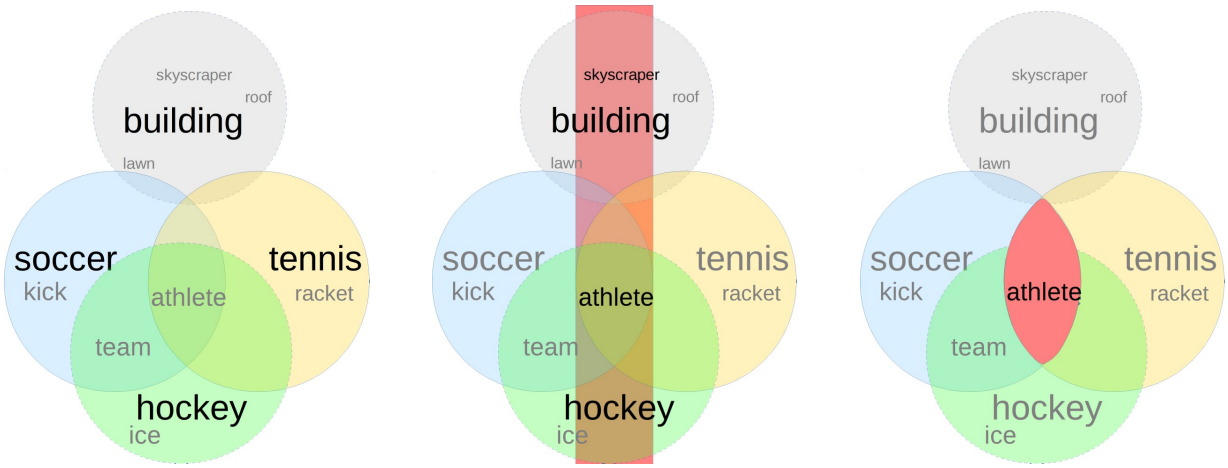


Figure 2: Illustration of subtopic's word bags. **(a) Left:** Overlap of word sets of the sport-subtopics "soccer" (given), "tennis" (given), "hockey" (unknown), with unrelated topic "building" (unknown). **(b) Center:** When considering solely ambiguity with respect to classification into "soccer" and "tennis", the residual set (in red) contains "building" and "hockey"-related documents. **(c) Right:** Considering also the similarity to the topic-common word "athlete", only "sport"-related documents like "hockey" overlap.

4 Evaluation

There are several questions that will be answered for one composed dataset to evaluate an active learning topic detection algorithm. How many subtopics will be found in comparison to random term selection? How many subtopics will be found over all subtopics in D? What is the round distribution, i.e. how many subtopics will be found over every round? Finally, how many unique subtopics will be found over rounds with different initial subtopics? An ideal algorithm would detect all subtopics that are available in D. In fact, it would detect them all in the first round without needing much time of a human oracle. Furthermore, it will indicate fast when it is not able anymore to find new subtopics and runs reliable, such that its performance is equal over differing starting values.

4.1 Dataset and Evaluation Design

For constructing a **dataset** with multiple topics, a list of 4309 common English nouns[3] has been parsed to search for Wikipedia articles which build up the document set D. In case of getting more than one search result on

[2]This threshold value $IGN = 0.5$ allows the algorithm to differentiate already with two documents in L, and also showed the best performance in experiments.

[3]Available at `http://www.desiquintans.com/nounlist`

function MCBI(U,L,C,V_{REJ})
  $MNB \leftarrow trainClassifier(L, C)$ ▷ train Multinomial Naive Bayes
  $R \leftarrow \emptyset$ ▷ **build residual set**
  for $u \in U$ **do**
    $p_{\cdot|u} \leftarrow posteriorScores(MNB, u)$ ▷ compute scores
    $(p_{best}, p_{2nd}) \leftarrow maxN(p_{\cdot|u}, 2)$ ▷ get the two highest scores
    $uncertainty_u \leftarrow p_{best} - p_{2nd}$ ▷ compute their difference
    $similarity_u \leftarrow p_{best} + p_{2nd}$ ▷ compute their sum
    if $(uncertainty_u \leq \tau_m) \wedge (similarity_u \geq \tau_s)$ **then**
      $R \leftarrow R \cup \{u\}$ ▷ add document to residual set
    end if
  end for

  $i \leftarrow 0$ ▷ **get words and their scores**
  for $w \in words(R)$ **do**
    if $|getDocumentsWithWords(L, w)| < 0.5 \cdot |L|$ **then** ▷ no topic term
      if $w \notin V_{REJ}$ **then** ▷ no rejected term
        $i \leftarrow i + 1$
        $word_i \leftarrow w$ ▷ compute scores
        $rejscore_i \leftarrow \max_{n \in V_{REJ}} \left(\frac{|getDocumentsWithWords(U \cup L, \{w,n\})|^2}{|getDocumentsWithWords(U \cup L, n)|} \right)$
        $discscore_i \leftarrow |getDocumentsWithWords(U, w)| - |getDocumentsWithWords(L, w)|$
      end if
    end if
  end for
  $Q \leftarrow getBestScores(word, discscore, rejscore)$ ▷ select candidates

  $(Q_{accepted}, Q_{rejected}) \leftarrow queryUser(Q)$ ▷ **query user's feedback**
  $C \leftarrow C \cup Q_{accepted}$ ▷ update clustering
  $L \leftarrow L \cup getDocumentsWithWords(U, Q_{accepted})$ ▷ update labelled set
  $U \leftarrow U \setminus getDocumentsWithWords(U, Q_{accepted})$ ▷ update unlabelled set
  $V_{REJ} \leftarrow V_{REJ} \cup Q_{rejected}$ ▷ update reject list
return U,L,C,V_{REJ}
end function

Figure 3: The MCBI Algorithm

Wikipedia, the first choice has been considered. Stop words have been removed, followed by lemmatization. On D, *tf-idf* with a threshold of 5 words per document was applied to keep solely the most important words. The resulting vocabulary set V has a size of 6.594 unique terms, which were considered in the bag-of-words representation. In order to enable a fair and automatic evaluation, a topic with known subtopics as ground truth was required. We have chosen the topic *countries*, as it is closed and identifiable. This means, that a complete list of all countries' names (as subtopics) is obtainable. We also wanted to count in *languages* [4] and *denonyms* (naming of a country's natives) [5] because of their high relatedness. As in most natural language applications some words are ambiguous, e.g. the country *Georgia* which is also an US state. For simplicity, such ambiguous terms were considered as valid subtopics. Furthermore, countries like *Marshall Islands* and *New Zealand* were transformed into single terms like *Marshall* and *Zealand* to reduce ambiguity.

We compared our MCBI approach (denoted as *MCBI, NF=on*) against three **baselines**: *CBI* [7][6], which in the paper [7] was already shown to compare favourably against other clustering-based approaches. *Random* selection, and finally a variant of MCBI without using negative feedback (denoted as *MCBI, NF=off*). For CBI's residual set size $|R|$, different parameter values were used (see first column in CBI's results table 1a). On this dataset, we run *CBI* and the *MCBI* algorithms 20 times, while *random* was run 1000 times to get reliable results.

[4] Available at `http://www.infoplease.com/ipa/A0855611.html`
[5] Available at `http://geography.about.com/library/weekly/aa030900a.htm`
[6] We reimplemented this approach, as the original source code is not available.

Each of those iterations consisted of 20 sampling rounds, where in each round 20 subtopics where queried. For the experiments we used 2, respectively 4, initial given subtopics. Within iterations, initial subtopics changed between countries (e.g. *Germany* & *Portugal*). When constructing *MCBI*'s residual set R, we iteratively lowered the thresholds τ_s and τ_m until the residual set's size was equal or greater than one percent of the unlabelled set's size. We also evaluated the effect of choosing not 0.5 as threshold IGN for ignored topic terms in line 16, which confirmed the our choice.

4.2 Results and Discussion

We first provide the aggregated results in Fig. 4, where we compare the four algorithms in terms of their average (over all iterations) number of found subtopics (ordinate axis) in each round (abscissa). In these experiments, *Random* yields a nearly uniform performance over the rounds. This is as expected, as due to the large size of V there is little dependence between rounds (i.e. it corresponds to sampling without replacement from a very large urn). *Random* performs on this dataset consistently better than *CBI*. This is in contrast to single-topic environments where CBI originates from. This evaluation shows the necessity of adaptations for handling multi-topic environments. Furthermore, *Random* performs worse than *MCBI* in most rounds. The performance of *MCBI* increases over the first rounds, before starting to decline towards the last quarter of the rounds. In particular the *MCBI* variant using negative feedback improves at the beginning. This indicates that the negative feedback helps in determining the relevance of subtopics. However, in the last quarter both *MCBI* variants perform similarly. This indicates that at this point negative feedback might start to be too restricting, although this requires further investigation.

When relating the maximum average found subtopics 26.20 to the total number of available subtopics (183) in D, ca. 14% are discovered, with standard deviation 4.69 over all iterations. Another interesting information is the unique count of subtopics in relation to the total availability over all iterations within highest scoring settings ($\#subtopics = 2, 4; NF = on; IGN\ Share\ variable$) which ranges between 60 and 66. Since we already detect up to 26.20 subtopics in one iteration, it seems not worth to restart with differing initial values depending on application.

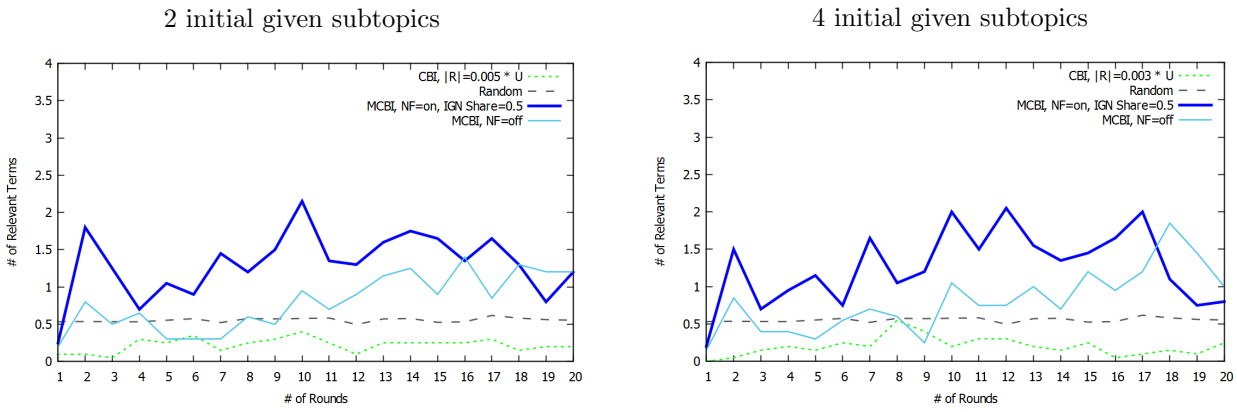


Figure 4: Average number of found subtopics over 20 rounds with original CBI, 2 & 4 initial subtopics and varying size of R

More detailed results for *CBI* and random are given in Table 1a. In the first column, settings are listed to compare different numbers of initial given subtopics and the document size of R. In the other columns, the average number of found countries, languages & denonyms and the count of subtopic and related words over all iterations with different initial given subtopics are depicted. Selecting words at random gives much better results than applying CBI in its original form. When composing R, documents are rated best for which it is uncertain how to associate them to the given subtopic clusters. Since we work in a multitopic environment, R will mostly imply documents that do not share any words with those from L, posteriors for given clusters and $u \in U$ are nearly equal which will result in small margins. For new subtopic proposals, those words are preferred that occur most often in R and least often in L. Since R is multitopical, most frequent words from R with a high unrelatedness to the searched topic are proposed. The size of R is set very low to limit the number of unrelated documents and therefore to achieve best results under these critical circumstances. The more documents we consider for R, the worse gets performance neglecting variance.

Detailed results for *MCBI* on different configurations (number of initial subtopics, negative feedback and ignoring shares) are given in Table 1b. *MCBI* without using negative feedback finds 44% more subtopics than *random*. By incorporating negative feedback without factoring in ignored words, we again make a jump with a total improvement of 117%. Furthermore, the choice of setting the IGN threshold to 0.5 yields the best results, as explained in Section 3.

Iteration Setting # subtopics, $\|R\|$	Average subtopics found		
	Count.	Lang. & Denonym	Total
Random	5.52	5.60	11.12
$2, 0.003 \cdot \|U\|$	1.2	3.25	4.45
$2, 0.005 \cdot \|U\|$	0.9	3.9	4.8
$2, 0.01 \cdot \|U\|$	0.4	3.15	3.55
$2, 0.05 \cdot \|U\|$	0.05	2.4	2.45
$2, 0.1 \cdot \|U\|$	0.0	2.3	2.3
$4, 0.003 \cdot \|U\|$	1.05	2.95	4.0
$4, 0.005 \cdot \|U\|$	0.9	2.8	3.7
$4, 0.01 \cdot \|U\|$	0.95	2.85	3.8
$4, 0.05 \cdot \|U\|$	0.0	1.45	1.45
$4, 0.1 \cdot \|U\|$	0.1	2.15	2.25

(a) Detailed evaluation results for CBI and random.

Iteration Setting # subtopics, NF, IGN	Average subtopics found		
	Count.	Lang. & Denonym	Total
Random	5.52	5.60	11.12
$2, off, -$	7.75	8.20	15.95
$2, on, 0$	15.40	8.60	24.00
$2, on, 0.1$	15.55	8.9	24.45
$2, on, 0.3$	15.35	8.9	24.25
$2, on, 0.5$	15.90	10.30	26.20
$2, on, 0.7$	15.60	10.15	25.75
$2, on, 1.0$	10.15	7.05	17.20
$4, off, -$	9.00	7.10	16.10
$4, on, 0$	15.40	7.90	23.30
$4, on, 0.1$	15.90	8.00	23.90
$4, on, 0.3$	15.95	8.45	24.40
$4, on, 0.5$	16.30	9.05	25.35
$4, on, 0.7$	15.90	9.35	25.25
$4, on, 1.0$	9.00	6.90	15.90

(b) Detailed evaluation results for $MCBI$ and random.

Table 1: Detailed results

5 Conclusion

This work addressed the clustering by intent scenario recently introduced in [7]. For this scenario, an active approach for detecting subtopics was presented. This approach extends the CBI algorithm to multi-topic environments, and to the incorporation of positive and *negative* user feedback. It iteratively constructs so-called residual sets of documents within the same topic. Based on this residual sets, it derives candidates for new subtopics. Then, feedback is actively queried from the user on these candidates. Finally, this is used to refine the subtopic and residual sets in the next iteration. The approach was evaluated on a corpus of Wikipedia articles for the 4309 most common English nouns, comprising a broad range of different topics. In our experiments, MCBI provides an improvement of up to 137% against random sampling, and 445% against CBI. While these first results are promising, a more extensive experimental evaluation is planned for the future. Because of the unsupervised learning task, a way to automatically tune parameters like uncertainty and similarity for constructing the residual set as well as IGN with differing datasets is desired. Furthermore, the role of negative feedback in such a system seems worth to be investigated further.

Acknowledgements

We would like to thank Andreas Nürnberger, Daniel Kottke, and George Forman for insightful discussions on this topic.

References

[1] An active learning framework for semi-supervised document clustering with language modeling. *Data and Knowledge Engineering*, 68(1):49–67, 2009.

[2] Martin Atzmüller. Subgroup discovery. *WIREs Data Mining Knowledge Discovery*, 5(1):35–49, 2015.

[3] Mohamed-Rafik Bouguelia, Yolande Belaïd, and Abdel Belaïd. Efficient active novel class detection for data stream classification. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, number 3, pages 2826–2831, 2014.

[4] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on twitter based on temporaland social terms. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, pages 4:1–4:10, 2010.

[5] Derya Dinler and Mustafa Kemal Tural. *A Survey of Constrained Clustering*, pages 207–235. Springer International Publishing, Cham, 2016.

[6] Brian Eriksson, Gautam Dasarathy, Aarti Singh, and Robert Nowak. Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15 of *JMLR Workshop and Conference Proceedings*, pages 260–268, 2011.

[7] George Forman, Hila Nachlieli, and Renato Keshet. Clustering by intent: A semi-supervised method to discover relevant clusters incrementally. In Albert Bifet, Michael May, Bianca Zadrozny, Ricard Gavalda, Dino Pedreschi, Francesco Bonchi, Jaime Cardoso, and Myra Spiliopoulou, editors, *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part III*, pages 20–36, Cham, 2015. Springer International Publishing.

[8] Dragan Gamberger, Nada Lavra, and Goran Krstai. Active subgroup mining: a case study in coronary heart disease risk group detection. *Artificial Intelligence in Medicine*, 28:27–57, 2003.

[9] Thomas Hofmann and Joachim M. Buhmann. Active data clustering. In *Proceedings of the 10th Conference on Advances in Neural Information Processing Systems*, NIPS '97, pages 528–534. MIT Press, 1998.

[10] Janez Kranjc, Jasmina Smailović, Vid Podpečan, Martin Grčar, Miha Žnidaršič, and Nada Lavrač. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing and Management*, 51:187–203, 2014.

[11] Akshay Krishnamurthy. *Interactive Algorithms for Unsupervised Machine Learning*. PhD thesis, 2015.

[12] Mohammad Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani M. Thuraisingham. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Trans. on Knowl. and Data Eng.*, 23(6):859–874, June 2011.

[13] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, and Bhavani Thuraisingham. Classification and novel class detection in data streams with active mining. In *Proceedings of the 14^{th} Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part II*, PAKDD 2010, pages 311–324, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.

[15] Burr Settles. *Active Learning*. Number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[16] Llanos Tobarra, Antonio Robles-Gmez, Salvador Ros, Roberto Hernndez, and Agustn C. Caminero. Discovery of interest topics in web-based educational communities. In *Proceedings of the International Symposium on Computers in Education (SIIE)*, pages 87–92. IEEE, 2012.

[17] Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proc. of the 1st Europ. Symposium on Principles of Data Mining and Knowledge Discovery*, pages 78–87. Springer, 1997.

[18] Xiatian Zhu, Chen Change Loy, and Shaogang Gong. Constrained clustering with imperfect oracles. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1345–1357, 2015.

1: **function** CBI(U,L,C)
2: $MNB \leftarrow trainClassifier(L, C)$ ▷ train Multinomial Naive Bayes
3: $R \leftarrow \emptyset$ ▷ **build residual set**
4: **for** $u \in U$ **do**
5: $p_{\cdot|u} \leftarrow posteriorScores(MNB, u)$ ▷ compute scores
6: $(p_{best}, p_{2nd}) \leftarrow maxN(p_{\cdot|u}, 2)$ ▷ get the two highest scores
7: $uncertainty_u \leftarrow p_{best} - p_{2nd}$ ▷ compute their difference
8: **if** $(uncertainty_u \leq \tau_m)$ **then**
9: $R \leftarrow R \cup \{u\}$ ▷ add document to residual set
10: **end if**
11: **end for**
12:
13: $i \leftarrow 0$ ▷ **get words and their scores**
14: **for** $w \in words(R)$ **do**
15: $i \leftarrow i + 1$
16: $word_i \leftarrow w$ ▷ compute scores
17: $discscore_i \leftarrow |getDocumentsWithWords(U, w)| - |getDocumentsWithWords(L, w)|$
18: **end for**
19: $Q \leftarrow getBestScores(word, discscore)$ ▷ select candidates
20:
21: $(Q_{accepted}) \leftarrow queryUser(Q)$ ▷ **query user's feedback**
22: $C \leftarrow C \cup Q_{accepted}$ ▷ update clustering
23: $L \leftarrow L \cup getDocumentsWithWords(U, Q_{accepted})$ ▷ update labelled set
24: $U \leftarrow U \setminus getDocumentsWithWords(U, Q_{accepted})$ ▷ update unlabelled set
25: **return** U,L,C
26: **end function**

Figure 5: The Implemented CBI Algorithm (based on [7]).