Active Online Learning for Social Media Analysis to Support Crisis Management

Daniela Pohl, Abdelhamid Bouchachia, *Senior Member, IEEE*, and Hermann Hellwagner, *Senior Member, IEEE*

Abstract—People use social media (SM) to describe and discuss different situations they are involved in, like crises. It is therefore worthwhile to exploit SM contents to support crisis management, in particular by revealing useful and unknown information about the crises in real-time. Hence, we propose a novel active online multiple-prototype classifier, called AOMPC. It identifies relevant data related to a crisis. AOMPC is an online learning algorithm that operates on data streams and which is equipped with active learning mechanisms to actively query the label of ambiguous unlabeled data. The number of queries is controlled by a fixed budget strategy. Typically, AOMPC accommodates partly labeled data streams. AOMPC was evaluated using two types of data: (1) synthetic data and (2) SM data from Twitter related to two crises, Colorado Floods and Australia Bushfires. To provide a thorough evaluation, a whole set of known metrics was used to study the quality of the results. Moreover, a sensitivity analysis was conducted to show the effect of AOMPC's parameters on the accuracy of the results. A comparative study of AOMPC against other available online learning algorithms was performed. The experiments showed very good behavior of AOMPC for dealing with evolving, partly-labeled data streams.

Index Terms—Online learning, multiple prototype classification, active learning, social media, crisis management

1 Introduction

The primary task of crisis management is to identify specific actions that need to be carried out before (prevention, preparedness), during (response), and after (recovery and mitigation) a crisis occurred [27]. In order to execute these tasks efficiently, it is helpful to use data from various sources including the public as witnesses of emergency events. Such data would enable emergency operations centers to act and organize the rescue and response. In recent years, a number of research studies [48] have investigated the use of social media as a source of information for efficient crisis management. A selection of such studies, among others, encompasses Norway Attacks [46], Minneapolis Bridge Collapse [34], California Wildfire [62], Colorado Floods [17], and Australia Bushfires [21], [22]. The extensive use of SM by people forces (re)thinking the public engagement in crisis management regarding the new available technologies and resulting opportunities [13].

Our previous work on SM in emergency response focused on offline and online clustering of SM messages. The offline clustering approach [49] was applied to identify sub-events (specific hotspots) from SM data of a crisis for an after-the-fact analysis. Online clustering [47] was used to identify sub-events that evolve over time in a dynamic way. In particular, online feature selection mechanisms were devised as well, so that SM data streams can be accommodated continuously and incrementally.

It is interesting to note that people from emergency departments (e.g., police forces) already use SM to gather, monitor, and to disseminate information to inform the public [20]. Hence, we propose a learning algorithm, AOMPC, that relies on active learning to accommodate the user's feedback upon querying the item being processed. Since AOMPC is a classifier, the query is related to labeling that item.

The primary goal in using user-generated contents of SM is to discriminate valuable information from irrelevant one. We propose classification as the discrimination method. The classifier plays the role of a filtering machinery. With the help of the user, it recognizes the important SM items (e.g., tweets), that are related to the *event* of interest. The selected items are used as cues to identify *sub-events*. Note that an *event* is the crisis as such, while *sub-events* are the topics commonly discussed (i.e., hotspots like flooding, collapsing of bridges, etc. in a specific area of a city) during a crisis. These sub-events can be identified by aggregating the messages posted on SM networks describing the same specific topic [47], [50].

We propose a *Learning Vector Quantization* (LVQ)-like approach based on multiple prototype classification. The classifier operates *online* to deal with the *evolving stream of data*. The algorithm, named *active online multiple prototype classifier* (*AOMPC*), uses unlabeled and labeled data which are tagged through active learning. Data items which fall into ambiguous regions are selected for labeling by the user. The number of queries is controlled by a budget. The requested

- *D. Pohl and H. Hellwagner are with the Institute of Information Technology, Alpen-Adria-Universität Klagenfurt, Universitätsstr. 65-67, Klagenfurt 9020, Austria. E-mail: {daniela, hellwagn}@itec.aau.at.*
- *A. Bouchachia is with the Smart Technology Research Centre, Bournemouth University, Poole BH12 5BB, United Kingdom. E-mail: abouchachia@bournemouth.ac.uk.*

Manuscript received 20 Dec. 2015; revised 1 Jan. 2019; accepted 3 Mar. 2019. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Daniela Pohl.)
Recommended for acceptance by J. Tang.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2019.2906173

items help to direct the AOMPC classifier to a better discriminatory capability. While AOMPC can be applied to any streaming data, here we consider in particular SM data.

The contributions of this paper are as follows:

- An original online learning algorithm, AOMPC, is proposed to handle data streams in an efficient way. It is a multi-prototype LVQ-like algorithm inspired by our previous work [8], [9].
- As part of AOMPC, an active learning strategy is introduced to guide AOMPC towards accurate classification, and in this paper towards sub-event detection. Such a strategy makes use of budget and uncertainty notions to decide when and what to label.
- AOMPC is evaluated on different data: synthetic datasets (synthetic numerical data, generated microblogs, which are geo-tagged) and real-world datasets collected from Twitter related to two crises, Colorado Floods in 2013 and Australia Bushfires in 2013. The choice and the use of all these datasets was motivated by their diversity. That allows to thoroughly evaluate AOMPC because these datasets have different characteristics.
- A sensitivity analysis based on the different AOMPC parameters and datasets is carried out.
- A comparison of AOMPC against well-known online algorithms is conducted and discussed.

The paper has the following structure. Section 2 presents the related work covering streaming and SM analysis. Section 3 introduces the classification algorithm and describes the processing steps, including the active learning facets. Section 4 discusses the empirical evaluation of AOMPC after describing the datasets used. Section 5 concludes the paper.

2 Related Work

The problem addressed in this paper is related to several topics: multiple prototype and Learning Vector Quantization (LVQ) classification, online learning for classification, active learning with budget planning, and social media analysis (i.e., natural language processing). A short overview of these topics is presented in the following.

2.1 Multiple Prototype Classification and LVQ Classification

A prototype-based classification approach operates on data items mapped to a vector representation (e.g., vector space model for text data). Data points are classified via prototypes considering similarity measures. Prototypes are adapted based on items related/similar to them.

A Rocchio classifier [36] is an example of a single prototype-based classifier. It distinguishes between two classes, e.g., "relevant" and "irrelevant". In real world-scenarios, due to the nature of the data, it is often not possible to describe the data with a single prototype-based classifier. Multiple prototype classifiers (i.e., several prototypes) are needed.

Self organizing maps (SOM) introduced by Kohonen [31] are an unsupervised version of prototype-based classification, also known as LVQ. In this case, prototypes are initialized (e.g., randomized) and adapted. SOM was also used for SM analysis in the context of crisis management to identify important hotspots [49].

LVQ has been applied to several areas, e.g., robotics, pattern recognition, image processing, text classification etc. [19], [31], [60]. LVQ - in the context of similarity representation, rather then vector-based representation - is analyzed by Hammer et al. [24]. Mokbel et al. [39] describe an approach to learn metrics for different LVQ classification tasks. They suggest a metric adaptation strategy to automatically adapt metric parameters.

Bezdek et al. [6] review several offline multiple prototype classifiers, e.g., LVQ, fuzzy LVQ, and the deterministic Dog-Rabbit (DR) model. The latter limits the movement of prototypes and is similar to our approach. However, in contrast to our approach, DR uses offline adaptation of the learning rate. The time-based learning rate of our algorithm considers concept drift (i.e., changes of the incoming data) directly during the update of the prototypes.

In contrast to the previous approaches, Bouchachia [8] proposes an incremental supervised LVQ-like competitive algorithm that operates online. It consists of two stages. In the first stage (learning stage), the notions of winner reinforcement and rival repulsion are applied to update the weights of the prototypes. In the second stage (control stage), two mechanisms, *staleness* and *dispersion* are used to get rid of dead and redundant prototypes. A summary of different prototype based learning approaches can be found in Biehl et al. [7].

In this study, we deal with online real-time classification and we propose a multi-prototype quantization algorithm, where the winning prototype is adapted based on the input. In particular, the algorithm relies on online learning and active learning.

2.2 Online Learning and Active Learning (with Budget Planning)

Online learning receives data items in a continuous sequence and processes them once to classify them accordingly [64]. Bouchachia and Vanaret [10], [11] use Growing Gaussian Mixture Models for online classification. Compared to the algorithm proposed in this work, there is a difference in adapting the learning rate and representing the prototypes. Reuter et al. [53] use multiple prototypes representing an event. New incoming items are assigned to the most similar events (by using an offline-trained SVM) or otherwise new events are created.

Another important topic in streaming analysis is active learning to improve results of classification with an amount of labeled data actively asked by the system [55]. Ienco et al. [28] use a pre-clustering step to identify relevant items to be labeled by the user. In Smailović et al. [57] active learning is used to improve the sentiment analysis of incoming tweets as an indicator for stock movements. Hao et al. [26] design two active learning algorithms (Active Exponentially Weighted Average Forecaster and Active Greedy Forecaster) which includes feedback of experts for labeling. The approach considers confidence of labels from the classifier compared to a set of experts. Hao et al. [25] also introduce online active learning considering second order information, e.g., based on covariance matrix. Ma et al. [35] combine decision trees with active learning. This approach improves the learning step for decision trees. Bouguelia et al. [12] use instance weighting for active online learning. They consider the weight that must be changed to cause the classifier changing its prediction. If only a small change in weight changes the original classification, then the classifier is highest uncertain about the item. Mohamad et al. [38] introduce an active learning algorithm for data streams with concept evolution. In addition, they suggest a bi-criteria active learning algorithm by including both label uncertainty and density of the underlying distribution [37].

Monzafari et al. [40] study different batch-based active learning approaches and define two uncertainty strategies to query labels from crowdsourcing platforms. In addition, the authors also define a budget or goal constraint to limit labeling. Žliobaitė et al. [63] use active learning combined with streaming data. They suggest several processing mechanisms to identify uncertainty regions especially for handling data drifts. It is also important to minimize the number of queries, asking an expert for labels. Žliobaitė et al. [63] include a moving average over the incoming items and the amount of already labeled items to estimate the budget. We adopted this mechanism together with the uncertainty strategies.

Based on categorization of active learning approaches by Settles et al. [55], our implementation is classified as a stream-based selective sampling approach, considering different strategies to request instances for labeling. In addition, we use an online feature selection approach described later.

2.3 Social Media Analysis for Crisis Management

Recent research studies SM from several technical perspectives. Due to space limitations, we describe existing SM analysis frameworks mostly in the context of crisis management, although there are several frameworks in other contexts, e.g., Twitterbeat [56] and HarVis [2]. Backfried et al. [3] describe an analysis approach based on visual analytics for combining information from different sources with a specific focus on multilingual issues. Vieweg and Hodges [29], [61] describe the Artificial Intelligence for Disaster Response (AIDR) platform, where persons annotate incoming tweets (similar to Amazon Mechanical Turk). The tweets are then used to train classifiers to identify more relevant tweets. AIDR allows to classify incoming tweets based on different information categories, e.g., damage report, casualties, advises, etc. Chen et al. [15] analyse tweets related to Flu to identify topics for predicting the Flu-peak. Neppalli et al. [41] perform sentiment analysis based on social media related to Hurricane Sandy. The work shows that sentiment of users is related to the distance of the Hurricane to the users. Twitcident described by Abel et al. [1] is a framework to search and filter Twitter messages through specific profiles (e.g., keywords). Terpstra et al. [59] show the usage of Twitcident in crisis management. Tweak-the-Tweet introduced by Starbird et al. [58] defines a grammar which can be easily integrated in tweets and therefore automatically parsed. Also, TEDAS described by Li et al. [33] is a system to detect high-level events (e.g., all car accidents in a certain time period) using spatial and temporal information. Yin et al. [65], [66] design a situational awareness platform for SM. Tweets are analyzed based on bursty keywords to identify emergent incidents. Ragini et al. [51] combine several techniques to identify people in danger. They examined rule based classification and several machine learning approaches, like SVM, for hybrid classification.

Additional information on social media analysis in different crises can be found in Reuter and Kaufhold [52]. Due to the importance of SM, it is our aim to support emergency management when using the content of SM platforms. Currently, there are systems with crowd-sourcing platform characteristics, but no procedure (like active learning) is available to directly involve emergency management personnel in filtering relevant information.

3 Active Online Multiple Prototype Classifier (AOMPC)

Due to the fact that SM data is noisy, it is important to identify relevant SM items for the crisis situation at hand. The idea is to find an algorithm that performs this classification and also handles ambiguous items in a reasonable way. Ambiguous denotes items where a clear classification is not possible based on the current knowledge of the classifier. The knowledge should be gained by asking an expert for feedback. The algorithm should be highly self-dependent, by asking the expert only labels for a limited number of items. Therefore, we propose an original approach that combines different aspects - such as online learning and active learning - to build a hybrid classifier, AOMPC. AOMPC learns from both, labeled and unlabeled data, in a continuous and evolving way. In this context, AOMPC is designed to distinguish between relevant and irrelevant SM data related to a crisis situation in order to identify the needs of individuals affected by the crisis. AOMPC relies on active learning. It implies the intervention of a user in some situations to enhance its effectiveness in terms of identifying relevant data and the related event in the SM stream of data (see Fig. 1). The user is asked to label an item if there is a high uncertainty about the classification as to whether it is relevant or irrelevant. The classifier assigns then the item (be it actively labeled or unlabeled) to the closest cluster or uses it to create a new cluster. A cluster - in this case - represents either relevant (i.e., specific information about the crisis of interest) or irrelevant information (i.e., not related to the crisis). The process flow and the steps of AOMPC are shown in Fig. 1.

AOPMC is described in Algorithm 1. The used symbols are defined in Table 1. CT and LTU are updated in batch-mode due to the feature selection method used (see Section 3.3 for details). The algorithm could also be used in item-wise mode.

The general idea of this algorithm is that the longer a prototype is stale (not updated), the slower it should move to a new position. The learning rate α is a function of the last time the prototype was a *winner* (i.e., α can be seen as a *forgetting factor*). The winning prototype is computed based on the learning rate (steps 5-6). If there is an uncertainty detected (see Section 3.2) and enough budget is available (see Section 3.1), the label is queried (steps 7-11). Otherwise (e.g., not enough budget) the winning prototype defines the label (step 16). When a prototype wins the competition among all other neighboring prototypes based on the queried label, it is updated to move in the direction of the new incoming item (steps 17-20). In case the new input comes with new features, the prototype's feature vector is extended to cover those new

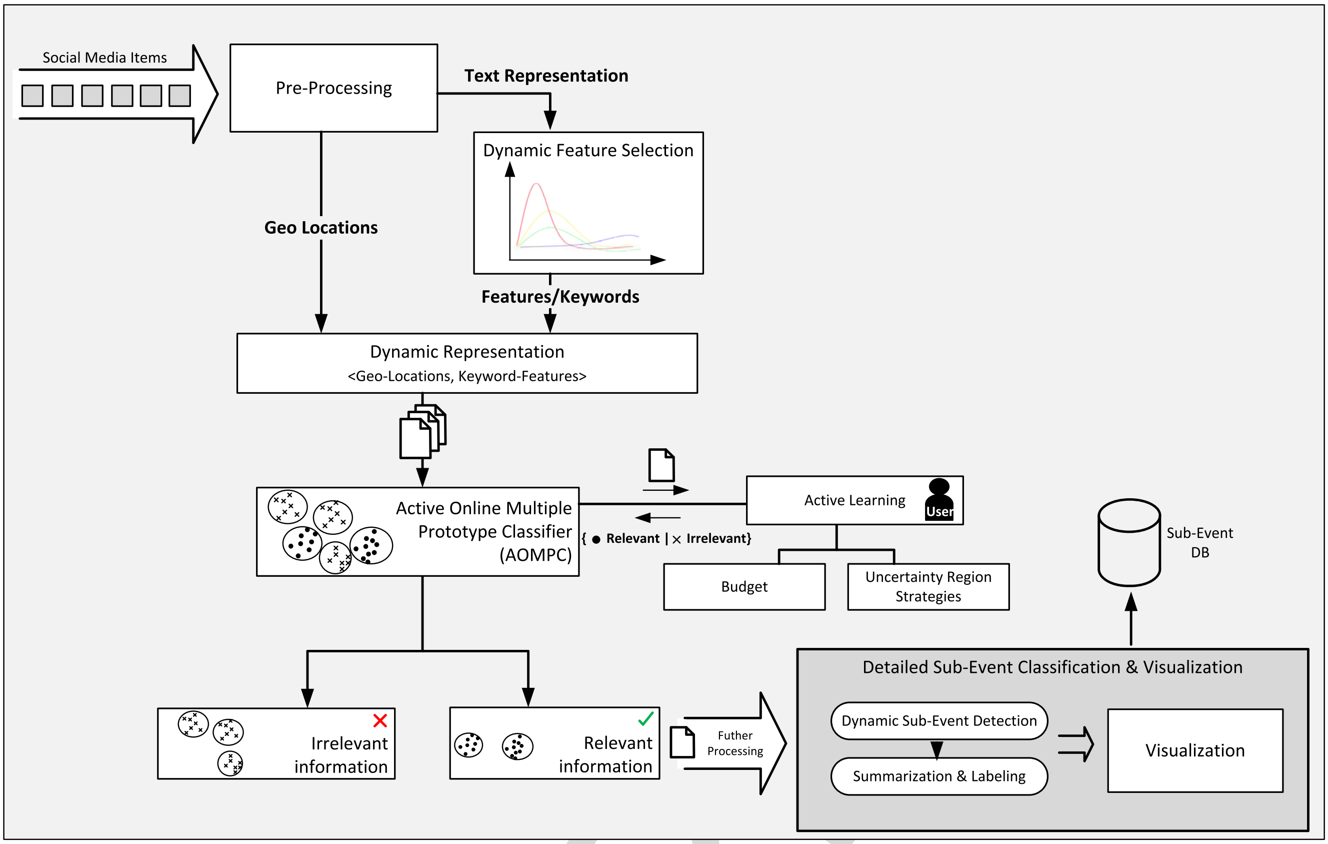


Fig. 1. Processing steps.

textual features (see step 20). In general, AOMPC is capable of accommodating new features. In the case of textual input, like in this study, the evolution of the vocabulary over time is captured. When no prototype is sufficiently close to the new item (step 22), a new prototype is created to accommodate that item (steps 26-28).

Algorithm 1 relies on the computation of the distance between the input and the existing prototypes (e.g., Euclidean distance in Algorithm 2). Because the SM items usually consist of a textual description (c.f., tweets), we apply the Jaccard coefficient [36] as a text-based distance ($dist\_text$) (see Algorithm 2, steps 2-3). If the social media items consist of two parts, the body of the message and the geo-location that indicates where the message was issued in terms of coordinates, then we apply a combined distance measure $(dist\_text + dist\_geo)/2$. Specifically, $dist\_text$ refers to the Jaccard coefficient, while $dist\_geo$ is the Haversine distance [5], [54] described in Algorithm 2, steps 4-7. The coordinates are expressed in terms of latitude and longitude.

Moreover steps 4-12 of Algorithm 1 are related to the active learning part. The algorithm starts by checking whether the new input item lies in the uncertainty region between the relevant and irrelevant prototypes and whether there is enough budget for labeling this item. More details follow in the next section.

TABLE 1
List of Symbols Used

Variable	Description
$\mathbf{x}$	Input (one item) received by the data stream X with bt_{CT} batches
V	Set of currently known prototypes
α	A parameter used in Algorithm 1 to compute the staleness of a prototype. It is given as: $\alpha = e^{\frac{-log2}{\beta}}$, where β is the half-life span, denoted hereafter as $(1/2)$-life-span, described in [30] that refers to the amount of time required for a quantity to fall to half its value as measured at the beginning of the time period.
I	Set of indices i indicating the prototypes $\mathbf{v}_i$
$dist$	Appropriate distance measure; see Algorithm 2
UT	Threshold used to identify uncertainty
CT	Current time
LTU	Last time the prototype was updated (i.e., the winner)
S	List of nearest prototypes in ascending order to the current input $\mathbf{x}$
$label$	Labels are: $relevant$, $irrelevant$, and $unknown$

3.1 Definition of Budget

The idea of active learning is to ask for user feedback instead of labeling the incoming data item automatically. To limit the number of interventions of the user, a so called *budget*, is defined. Budget can be understood as the maximum number of queries to the user. We adapt the method presented in [63] to implement active learning in the context of online multiple prototype classification. In step 7 of Algorithm 1, the method *within_budget()* checks if enough budget is available for querying the user. The consumed budget after k items, b_k is defined in [63] as follows:

$$u_k = u_{k-1}\lambda + labeling_k;\ \lambda = (w-1)/w;\ b_k = \frac{u_k}{w}, \tag{3}$$

where u_k estimates the amount of labels already queried by the system in the last w steps. The window w acts as memory [63] (e.g., last 100 item steps) described by λ. Hence, λ describes the fraction of including value u_{k-1}. $labeling_k$ updates u_k based on the requested label (i.e., $labeling_k = 0$ if no label was queried and $labeling_k = 1$ if there was a label requested) for the current item k.

Algorithm 1: Steps of AOMPC

Input: Data stream X
Output: List of prototypes V
1: CT=1; LTU=CT;
2: Let CT and LTU indicate the current time and the last time a prototype was updated respectively
3: **for** batch bt_{CT} of X **do**
4: **for** incoming input **x** of bt_{CT} **do**
5: Compute distance φ_i between **x** and all prototypes $\mathbf{v}_i$, $i = 1 \cdots |V| = I$, as follows:

$$\textbf{if } (inaction(\mathbf{v}_i) > 0)\ \varphi_i = inaction(\mathbf{v}_i) \cdot dist(\mathbf{v}_i, \mathbf{x}) \quad \textbf{else } \varphi_i = dist(\mathbf{v}_i, \mathbf{x}) \textbf{ endif} \tag{1}$$

such that $inaction(\mathbf{v}_i) = 1 - \alpha^{(CT - \mathbf{v}_i.LTU)}$
6: Compute list of nearest prototypes S based on sorted index I such that $S = createSortedList(I,\ (x,y)) : (\varphi_x \leq \varphi_y)$
7: check = *uncertainty*(**x**) and *within_budget()*;
8: **if** check = true **then**
9: Query the label of **x**
10: **else**
11: $\mathbf{x}.label = unknown$
12: **end if**
13: **if** $S \neq \{\}$ **then**
14: Let j be the index of the closest prototype: $j = S(1)$
15: **if** $\mathbf{x}.label = unknown$ **then**
16: Assign the data item to $\mathbf{v}_j$
17: **else**
18: **if** $\mathbf{x}.label = \mathbf{v}_j.label$ **then**
19: Reinforce $\mathbf{v}_j$ with **x** using only the common features: $\mathbf{v}_j = \mathbf{v}_j + \alpha^{CT-LTU}(\mathbf{x} - \mathbf{v}_j)$
20: Add the non-common features of **x** to $\mathbf{v}_j$: $\mathbf{v}_j.feature = \alpha^{CT-LTU}$(x.feature)
21: **else**
22: Go to line 26
23: **end if**
24: **end if**
25: **else**
26: Initialize a new prototype: $\mathbf{v}_{new}$=**x**
27: $\mathbf{v}_{new}.label = \mathbf{x}.label$; $\mathbf{v}_{new}.LTU = CT$
28: $V = V \cup \{\mathbf{v}_{new}\}$
29: **end if**
30: **end for**
31: Update winning clusters in bt_{CT} with $LTU = CT$
32: CT = CT + 1;
33: **end for**

An upper bound B is defined describing the maximum number of requested labels. B is the fraction of data from window w that can be labeled (i.e., $B = 0.2$ are 20 percent). At each step, one input is processed. The *within_budget()* procedure in Algorithm 1 checks if enough budget is available (i.e., $b_k < B$). If so, the algorithm queries the label of the ambiguous input.

Algorithm 2: $dist(\mathbf{v}, \mathbf{x})$

Input: Prototype **v** , input **x**
Output: Distance of (**v**,**x**)
1: **if** the input is a social media item **then**
2: Compute the textual distance (Jaccard) as follows:

$$\begin{aligned} dist\_text =& 1 - jaccard, \text{ where:} \\ jaccard =& |A \cap B| / |A \cup B|; \end{aligned}$$

3: $distance = dist\_text;$
4: **if** the input is a composed social media item **then**
5: Compute the geo-location distance as follows:

$$\begin{aligned} dist\_geo =& 1 - H(\mathbf{v}.geo\_co, \mathbf{x}.geo\_co)/\pi \\ &\text{where:} \\ H(\mathbf{x}_1, \mathbf{x}_2) =& 2 \cdot atan2(\sqrt{\phi}, \sqrt{1-\phi}) \\ \phi =& sin^2(\frac{\Delta lat}{2}) + cos(\mathbf{x}_1.lat) \cdot \\ & cos(\mathbf{x}_2.lat) \cdot sin^2(\frac{\Delta lon}{2}) \\ \Delta lat =& \mathbf{x}_2.lat - \mathbf{x}_1.lat, \\ \Delta lon =& \mathbf{x}_2.lon - \mathbf{x}_1.lon \end{aligned}$$

6: $distance = (dist\_geo + dist\_text)/2;$
7: **end if**
8: **else**
9: Note: the input is no social media item
10: Compute the Euclidean distance as follows:

$$dist\_Euclidean(\mathbf{v}, \mathbf{x}) = \sqrt{\sum_{i=1}^{M} (\mathbf{v}_i - \mathbf{x}_i)^2} \tag{2}$$

11: **end if**

3.2 Which Data Items to Query?

In active learning, before querying the label, one has to decide which data points to query. Obviously one has to find those points, for which the classifier is not confident about the assignment decision (see Algorithm 1, step 7). In this paper, we use a simple mechanism based on the neighboring prototype proximity and labels. An input **x** is queried if its two most closest prototypes, $\mathbf{v}_i$ and $\mathbf{v}_j$ with distances φ_i and φ_j, respectively, and where $i = S(1)$ and $j = S(2)$, have different labels. Eq. (4) below formalizes the test which is called *simple conflicting neighborhood (SCN)* hereafter.

$$uncertainty(\mathbf{x}) = \begin{cases} 1 & \text{if } (|S| < 2) \text{ or} \\ & (|\varphi_i - \varphi_j| < UT \text{ and} \\ & \mathbf{v}_i.label \neq \mathbf{v}_j.label) \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

However, to make the selection more constrained, a second variant is introduced. In fact, it is worthwhile to look at the border area of the inter-class uncertainty regions, where the labels are very important/useful. This border area could be used to track concept drift.

Eq. (5) shows the constraint by multiplying the threshold UT by a random number m that has a uniform distribution in unit interval [0,1] ($m \sim U(0,1)$) [63]. This variant is called *controlled variable conflicting neighborhood (CVCN)*.

$$uncertainty(\mathbf{x}) = \begin{cases} 1 & \text{if } (|S| < 2) \text{ or} \\ & (|\varphi_i - \varphi_j| < (UT * m) \\ & \quad \text{and } \mathbf{v}_i.label \neq \mathbf{v}_j.label \\ & \quad \text{where } m \sim U(0,1)) \\ 0 & \text{otherwise} \end{cases}. \quad (5)$$

Moreover, the threshold UT can be continuously updated, as proposed in [63], according to the following rule:

$$\begin{cases} uncertainty(\mathbf{x}) = \begin{cases} 1 & \text{if } (|S| < 2) \text{ or} \\ & (|\varphi_i - \varphi_j| < UT \text{ and} \\ & \mathbf{v}_i.label \neq \mathbf{v}_j.label) \\ 0 & \text{otherwise} \end{cases}, \\ UT = UT + (-1)^{uncertainty} * step \end{cases} \quad (6)$$

where $step$ is set to 0.01 as suggested in [63]. We name this variant *dynamic conflicting neighborhood (DCN)*. In the given equation it is combined with the *SCN* strategy. Additionally, we combined it with the *CVCN* strategy given above.

As a baseline for comparison, we implement a *random* version (see Eq. (7)). We name this variant *random conflicting neighborhood (RCN)*.

$$uncertainty(\mathbf{x}) = \begin{cases} 1 & \text{if } (|S| < 2) \text{ or} \\ & (|\varphi_i - \varphi_j| < r \\ & \quad \text{and } \mathbf{v}_i.label \neq \mathbf{v}_j.label \\ & \quad \text{where } r \sim U(0,1) \text{ is a} \\ & \quad \text{random variable}) \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

We also implemented another version, called Random (R) that assumes a fixed uncertainty given by UT as shown in Eq. (8).

$$uncertainty(\mathbf{x}) = \begin{cases} 1 & \text{if } (|S| < 2) \text{ or} \\ & (r < UT) \\ & \quad \text{where } r \sim U(0,1) \text{ is a} \\ & \quad \text{random variable}) \\ 0 & \text{otherwise} \end{cases}. \quad (8)$$

We ignore an absolute *pure random* version $r < B$, because it would increase the number of queries drastically compared to the other uncertainty variants.

3.3 Dynamic Representation of Social Media Stream

The SM items considered in our work are textual documents and therefore their representation will rely on the standard *tf-idf* [36], [47]. The pre-processing step pointed out in Fig. 1, as part of the workflow, makes use of feature extraction which is sufficiently discussed in our previous work [47]. This step also includes the identification of word synonymy using WordNet [47]. Similar words (e.g., "car" and "automobile") are reduced to one root word. In this case, a document is represented as a bag-of-words. However, because social media documents arrive online and are processed as batches, *tf-idf* should be adapted to meet the streaming requirement [47]. Basically, the importance of a word is measured based on the number of incoming documents containing that word. Thus, the evolution of a term's importance should be reflected in the formulation of *tf-idf*. Here, we use a factor that scales *tf-idf* so that the importance increases and decreases according to the term's presence in the incoming batches

$$scaled\_tf\_idf_{t,d} = importance_{t,\tau} \cdot tf_{t,d} \cdot idf_t. \quad (9)$$

The importance factor $importance_{t,\tau}$ of term t is calculated over batches (windows) marked by time τ. The length of the batch is defined by the user (e.g., 30 minutes). It depends on the nature of the crisis. Slow evolution of the crisis may require longer windows, while fast evolution requires short windows. Terms with low importance value are removed from the index. For instance, if importance < 0.2, then 80 percent of the term's importance is lost. The importance of a term is computed as follows:

$$importance_{t,\tau} = g_{t,\tau}/g\_max_t, \quad (10)$$

where $g_{t,\tau}$ is the weight of term t obtained at time τ. The weight $g_{t,\tau}$ is refreshed based on intermediate sampling intervals (i.e., sub-batches, like every 10 minutes). g_max_t is the maximum weight the term t reached. $g_{t,\tau}$ is expressed as follows:

$$g_{t,\tau} = \begin{cases} (1-\gamma) \cdot u_{t,\tau} + \gamma \cdot g_{t,\tau-1} & \text{if } u_{t,\tau} > g_{t,\tau-1} \\ (1-\delta) \cdot u_{t,\tau} + \delta \cdot g_{t,\tau-1} & \text{otherwise} \end{cases}, \quad (11)$$

where $u_{t,\tau}$ describes the incoming SM items containing t till time τ and $g_{t,\tau-1}$ is the weight of term t of the previous sampling interval $\tau - 1$. Case 1 of Eq. (11) shows how fast terms are learned (i.e., a smaller γ corresponds to faster increase of importance). Case 2 of Eq. (11) shows how fast terms should be forgotten (i.e., a higher δ corresponds to slower forgetting or decrease of importance). The values γ and δ are empirically set by the user. We suggest that $\gamma < \delta$ so that terms are learned faster, compared to forgetting them again.

4 Evaluation

In the following we present the experimental setting including the datasets and the metrics we used. We then describe the experiments and their outcomes.

4.1 Synthetic Datasets

To evaluate AOMPC, we use two synthetic datasets. The first one is a 2-dimensional numerical dataset and the second one is a collection of SM messages artificially generated by a tool. These datasets allow to observe the behavior of the algorithm, especially because it simulates data drift. The artificial SM data is used to evaluate the online classifier on geo-tagged textual data which is close to the real-world data.

The simple 2-dimensional synthetic dataset is based on Gaussian data (GD). GD consists of 4 batches (see Fig. 2) which are sequentially presented to AOMPC. Each batch consists of 200 points, generated by two Gaussians which actually represent two clusters. The upper clusters (100

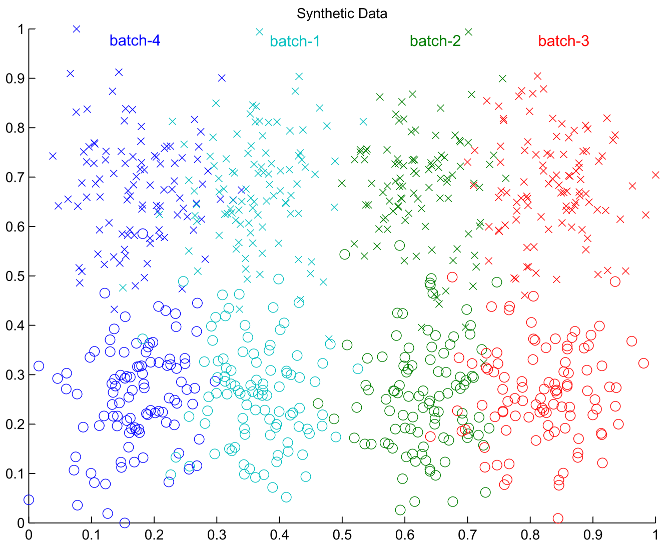


Fig. 2. GD dataset to simulate the stream appearing in the order batch-1, batch-2, batch-3, and batch-4.

points each), denoted as ’x’, are assumed “irrelevant”, while the lower clusters, denoted as ’o’, are assumed “relevant”. Batch-4 given in Fig. 2 contains a virtual or temporary drift caused by abrupt changes of the feature values [23].

The geo-tagged text collection, *synthetic social media dataset (SSMD)*, was generated using a tool[1] we originally developed for integrating SM into emergency exercises (i.e., training of first responders). We generated microblogs using a data generation tool we developed and which is based on a set of predefined text snippets that describe sub-events like “vehicles and garbage dumps on fire”, “police attacked by rioters”, and ”shop on fire nearby” (see Fig. 3a). The randomly generated data follows the timeline of the UK riots (see [4]) described as an XML file (see Fig. 3b). This way we generate data which describes incidents close to what happened in reality. The XML file covers the different phases and particularly the sub-events of the UK riots which are marked as relevant or irrelevant using a tag (*relevant*) to provide the ground truth for the experiments. Irrelevant sub-events in the data are represented by real-world tweets collected from Twitter in relation to a given location (e.g., London), while relevant sub-events are based on the text snippets. On the other hand, additional data, in the form of textual annotations, was collected from Flickr and YouTube and was labeled based on the real-world sub-events of the riots (see [49]).

In total, we used a collection of 1227 messages, mostly covering London districts. The data collected over 28 hours (’2011-08-06 19:44:00’ to ’2011-08-07 23:44:00’) covers several calm periods during the riots. The data is split into 30-minutes batches to observe the behavior of AOMPC. The number of messages relevant to the riots is 312, with 116 distinct text messages. Furthermore, there are 915 irrelevant messages with 789 distinct messages. In all, the dataset contains approximately 322 repetitions of text messages. Repetition refers to messages that are very similar and correspond to retweets.

1. http://www.bridgeproject.eu/content/bridge_information_intelligence_flyer.pdf, [Accessed: August 2014]

4.2 Real-World Datasets

The CrisisLexT26 collection [42] was recently made available to the community. It consists of Twitter data related to 26 crises around the world. Each crisis is described by 1,000 items which were randomly selected and labeled through a crowdsourcing platform. The class labels of the items were assigned by the majority of three crowdsourcing workers. Four categories are available: *related to the crisis and informative*, *related to the crisis - but not informative*, *not related* and *not applicable*. In our case, we have considered items *relevant* only when they are labeled as *related to the crisis and informative*. Otherwise, they are considered irrelevant.

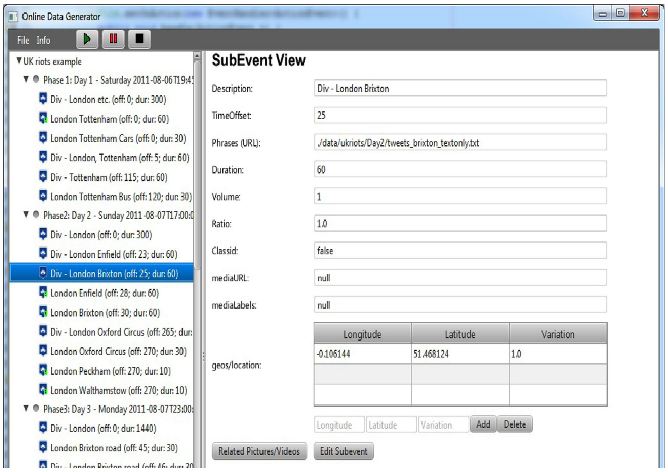
We selected two datasets from the CrisisLexT26 collection: Colorado Floods (CF) and Australia Bushfires (AB) which are dated but not geo-tagged. CF data is from the period ’2013-09-12 07:00:00’ - ’2013-09-29 10:00:00’. The data is somewhat imbalanced, the number of relevant items is larger than that of the irrelevant ones. CF data consists of 751 relevant items and 224 irrelevant items and approximately 189 repetitions. Considering the number of relevant and irrelevant items of SSMD, CF has an opposite, but very similar, distribution. AB data is from the period ’2013-10-17 05:00:00’ - ’2013-10-29 12:30:00’. It consists of 645 relevant, 408 irrelevant items and approximately 385 retweets.

4.3 Evaluation Measures

Because AOMPC combines clustering and classification, we developed a combined performance measure, called *combined quality measure* (CQM), to evaluate the algorithms. It is defined as follows:

$$\begin{aligned} CQM = & \left[0.3 * \frac{\sum_{i=1}^{|Bt|} vm_i}{|Bt|}\right] \\ & + \left[0.5 * \frac{\sum_{i=1}^{|Bt|}(1 - er_i/100)}{|Bt|}\right] \\ & + [0.2 * (1 - (Q/\#items))]. \end{aligned} \tag{12}$$

It refers to two other known measures, namely the validity measure (VM) and the error-rate (ER) measure (see Appendix A for details, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2019.2906173). CQM contains VM as a cluster evaluation measure and ER as classification specific measure. A high VM value indicates a good clustering, whereas a high value of (1-ER) unveils satisfactory labelling. The technical details of VM and ER are given in Appendix A, available in the online supplemental material. In terms of active learning budget B, the number of queries (Q) has been taken into account. In Eq. (12), Bt is the set of batches ($Bt = \{bt_1, \ldots, bt_{|Bt|}\}$) and vm_i and er_i are the values of VM and ER for batch bt_i respectively. *#items* is the number of items. As shown in Eq. (12), the measures are weighted based on their importance. ER is weighted with a factor of 0.5 due to its high importance, followed by VM with weight 0.3. Finally, the number of queries is weighted with 0.2. In conclusion high values of CQM indicate high quality of clustering and classification.



(a) Data Generation Tool GUI

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
- <Exercise generalPicPath="./data/CodedEntries.xml" name="UK riots example" startdate="2011-08-06T19:45:00Z"
  xmlns="urn:bridge:datagen:2013">
  - <Phase name="Phase 1: Day 1 - Saturday" date="2011-08-06T19:45:00Z">
    - <SubEvent relevant="0" classid="0" ratio="0.0" volume="1" duration="300"
      phrases="./data/ukriots/Day1/tweets_uk_london_tottenham_textonly.txt" desc="Div - London etc."
      timeOffset="0">
        <GeoLocation variation="10" longitude="-0.123024" latitude="51.50917"/>
      </SubEvent>
    - <SubEvent relevant="1" ratio="1.0" volume="2" duration="60"
      phrases="./data/ukriots/Day1/London_Tottenham.txt" desc="London Tottenham" timeOffset="0"
      pictures_labels="70, 71, 72, 73">
        <GeoLocation variation="1" longitude="-0.072191" latitude="51.605784"/>
      </SubEvent>
    - <SubEvent relevant="1" ratio="1.0" volume="1" duration="30"
      phrases="./data/ukriots/Day1/London_Tottenham_cars.txt" desc="London Tottenham Cars"
      timeOffset="0">
        <GeoLocation variation="0.3" longitude="-0.070777" latitude="51.591763"/>
      </SubEvent>
    - <SubEvent relevant="0" classid="0" ratio="0.0" volume="1" duration="60"
      phrases="./data/ukriots/Day1/tweets_london_tottenham_textonly.txt" desc="Div - London, Tottenham"
      timeOffset="5">
        <GeoLocation variation="10" longitude="-0.123024" latitude="51.50917"/>
      </SubEvent>
    - <SubEvent relevant="0" classid="0" ratio="0.0" volume="1" duration="60"
      phrases="./data/ukriots/Day1/tweets_tottenham_textonly.txt" desc="Div - Tottenham"
      timeOffset="115">
        <GeoLocation variation="10" longitude="-0.123024" latitude="51.50917"/>
      </SubEvent>
```


(b) UK riots stream in XML format

Fig. 3. Data generation tool.

4.4 Experiments and Results

We conducted extensive analysis. In particular, we did a sensitivity analysis to observe the effect of the algorithm's parameters: α, β, the threshold UT (see Algorithm 1 and Table 1), and the budget B (see Section 3.1). In this section, we describe the outcome of the experiments on the datasets using different settings as shown in Table 2. We focus on the performance of the different uncertainty strategies using CQM. The α-*setting* represents the fixed and variable α settings.

Gaussian Dataset (GD). Considering the most sensitive parameters, namely B and α (see Appendix B, available in the online supplemental material), the effect of active learning methods is illustrated in Fig. 4. The other parameters B and UT are discussed in Appendix B, available in the online supplemental material. In general it can be seen that the uncertainty strategy R yields the lowest CQM value and that RCN tends to query more often, since the pure random threshold r varies between 0 and 1 (see Section 3.2). For example, SCN has a query ratio of 0.14 and RCN a ratio of 0.2 to achieve a similar ER value (SCN with ER=1.250 and RCN with ER=1.370). On average, SCN variants show the most stable results, while the CVCN variants slightly increase *CQM* for small values of B (i.e., $B \leq 0.2$), because they focus on concept drift near to the uncertainty boundary.

TABLE 2
Evaluation Parameters

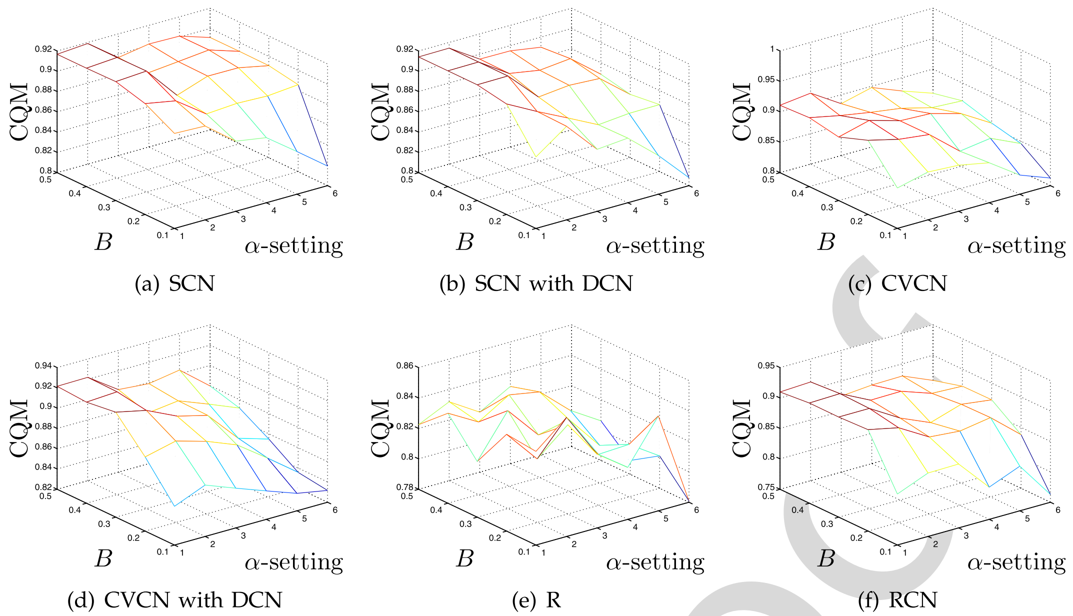
Parameter	Values/Instances
B	$B = 0.1, 0.2, \ldots 0.5$ with $w = 100$
UT	$0.1, 0.2, 0.3$
β	$1, 2, 3, 4$
fixed α	0.01 and 0.03
variable α	$\alpha = e^{\frac{-log(3)}{\beta}}$ as $(1/3)$-life-span $\alpha = e^{\frac{-log(2)}{\beta}}$ as $(1/2)$-life-span $\alpha = e^{\frac{log(2/3)}{\beta}}$ as $(2/3)$-life-span $\alpha = e^{\frac{log(7/8)}{\beta}}$ as $(7/8)$-life-span
Active Learning Method	SCN, CVCN, SCN with DCN, CVCN with DCN, R, and RCN
α-setting #1	equals to 0.01 (fixed α)
α-setting #2	equals to 0.03 (fixed α)
α-setting #3	equals to $(1/3)$-life-span (var. α)
α-setting #4	equals to $(1/2)$-life-span (var. α)
α-setting #5	equals to $(2/3)$-life-span (var. α)
α-setting #6	equals to $(7/8)$-life-span (var. α)

Synthetic Social Media Dataset (SSMD). The active learning strategies (SCN, CVCN, SCN with DCN and CVCN with DCN) given in Fig. 5 show that they outperform the random method R. Again, RCN shows good performance due to the higher variety of the threshold. For *CVCN with DCN* 0.22 queries and RCN 0.24 queries out of $B = 0.3$ are requested, reaching an ER of 7.3225 and 7.4984, respectively. A high value of B increases the overall quality of the results independently of the method (i.e., more labeled data is available to build the classification model). The *CVCN* options performs best for high values of B for the different α settings. In general, the active learning options *SCN with DCN* and *CVCN with DCN* perform best. This might indicate that concept drift appears along the uncertainty region border as those *"with DCN"* methods vary the border by changing UT. This behavior is expected, since data varies in a small range, i.e., geo-data within London area with similar incidents (damages caused by riots).

Colorado Floods (CF). Fig. 6 illustrates the outcome of AOMPC on the CF data for the different active learning strategies.

The results of CF indicate good performance for the fixed α values and especially for a low budget B. The results corresponding to variable α are better than those obtained with fixed α. Note that higher α leads to fast update of the AOMPC prototypes and that variable α requires less queries (see Table 5). Based on the Levenshtein distance ($ldis$) ([32], for calculating similarity between character strings), there exist 105 items with similar text (i.e., $ldis \leq 0.2$) in CF, which is a quite small number. This also indicates that the length of the repeating text fragments are very small (105 versus 189 repetitions of text). Therefore, the small number of similar items for this long period of the crisis and the performance related to the variable α with a fast adaptation are an indication that there are drifts in CF not near the inter-class border as defined by UT.

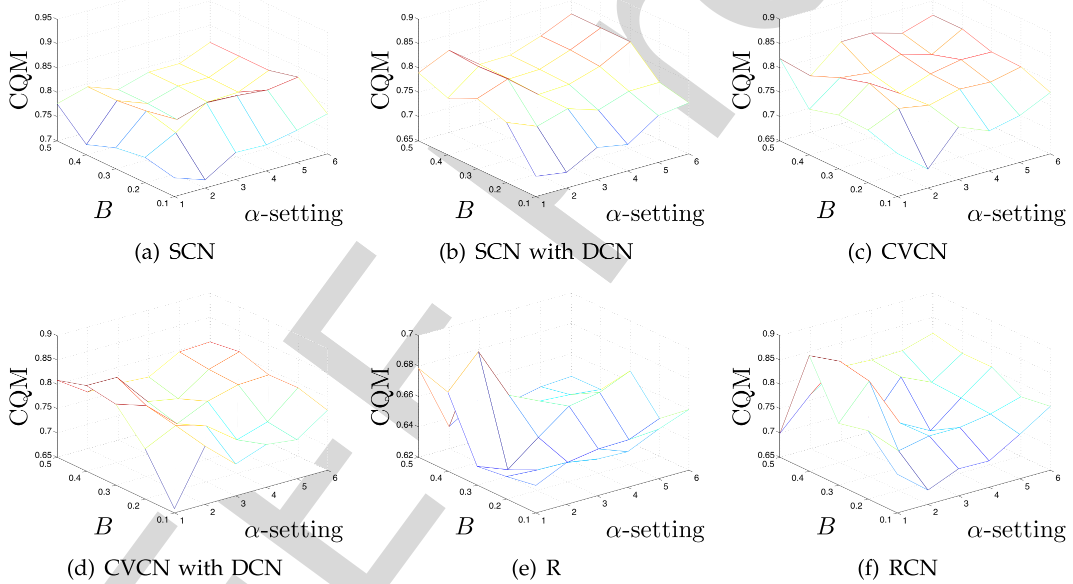
Australian Bushfires (AB). AOMPC's results on AB are illustrated in Fig. 7. The variable α shows nearly the same



(a) SCN (b) SCN with DCN (c) CVCN

(d) CVCN with DCN (e) R (f) RCN

Fig. 4. Results of the different active learning methods using the Gaussian data (GD) and the CQM measure.



(a) SCN (b) SCN with DCN (c) CVCN

(d) CVCN with DCN (e) R (f) RCN

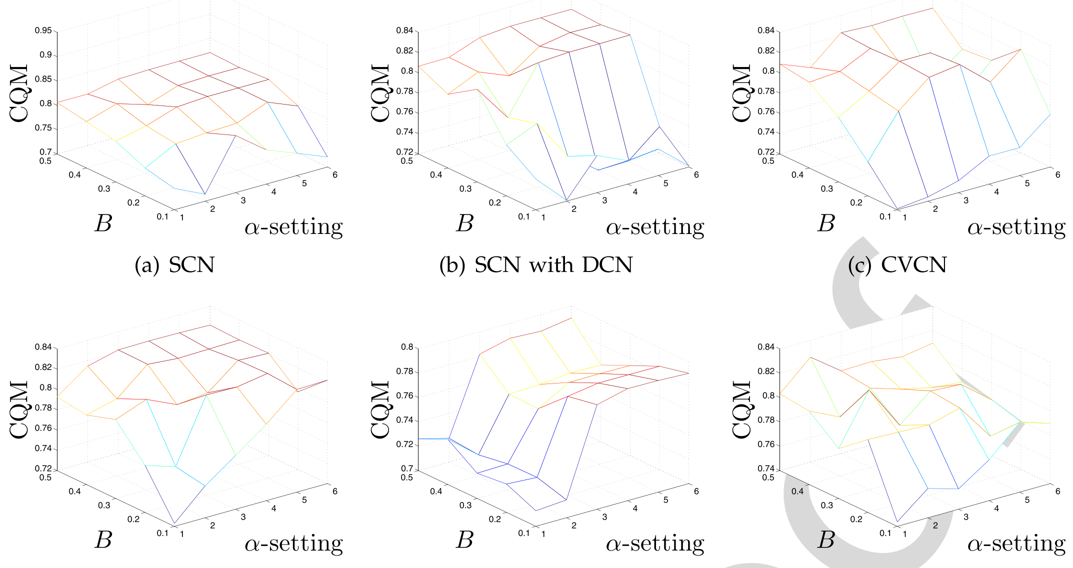
Fig. 5. Results of the different active learning methods using the synthetic social media dataset (SSMD) and the CQM measure.

performance, but this time it is worse compared to the values obtained on CF. The AB dataset has a high amount of similar items, which is 582 (items with $ldis \leq 0.2$). This high amount of similar items is an indicator that changes in data are more common around the boundary, because similar vocabulary within the items is used. AOMPC shows the best performance with a fixed α value for all budget settings. Due to the high similarity between items combined with conflicting labels, it is more difficult to distinguish between relevant and irrelevant items. Consider the following example, which shows the same tweet, but labeled differently [42] (*Related-and-informative* and *Not-related*):

- Wed Oct 16 17:12:46 +0000 2013: ”RT @Xxxxx: A dog has risked its life to save a litter of newborn kittens from a house fire in Melbourne, Australia http://t.co/Gz..”,Eyewitness,Affected individuals, *Related and informative*
- Wed Oct 16 17:13:57 +0000 2013: ”RT @Xxxxx: A dog has risked its life to save a litter of newborn kittens from a house fire in Melbourne, Australia http://t.co/Gz...”,Not labeled,Not labeled,*Not related*

AB is an interesting dataset for testing the algorithms under various conditions. Fixed α provides much better quality on AB compared to other α-settings as shown in Fig. 7.

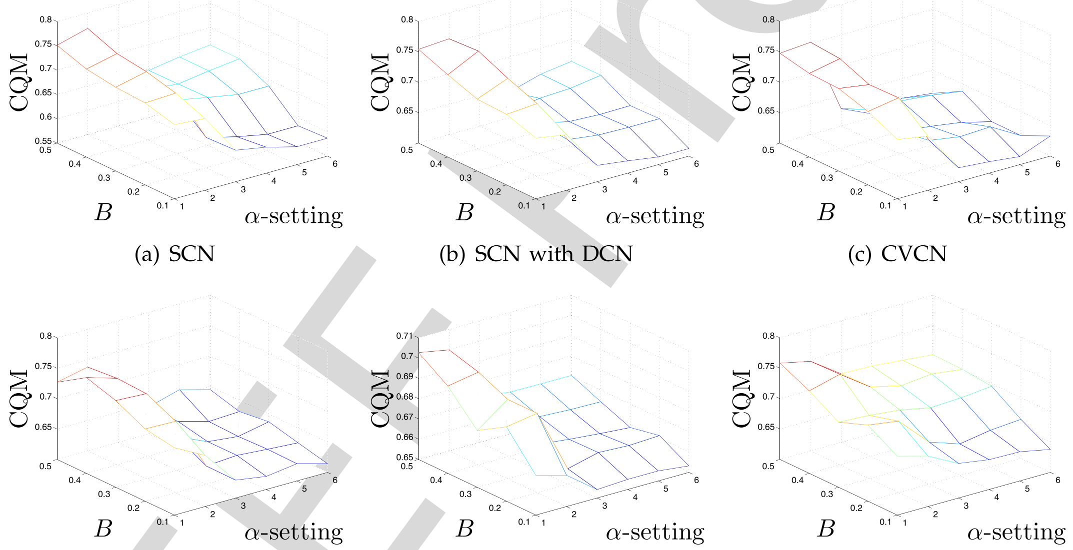
Considering Figs. 7 and 6, we can conclude a fixed learning rate of α and “with DCN” active learning strategies produce good performance for both CF and AB, especially, for low values of B.



(a) SCN (b) SCN with DCN (c) CVCN

(d) CVCN with DCN (e) R (f) RCN

Fig. 6. Results of the different active learning methods using the Colorado Floods dataset (CF) and the CQM measure.



(a) SCN (b) SCN with DCN (c) CVCN

(d) CVCN with DCN (e) R (f) RCN

Fig. 7. Results of the different active learning methods using the Australia Bushfires dataset (AB) and the CQM measure.

4.5 Comparative Studies: AOMPC versus Others

Beside the experiments with different datasets and parameters, we compare AOMPC against the unsupervised k-means algorithm that operates without labels and against a set of supervised online algorithms that require full labeling. This choice should help assess AOMPC against the extreme ends of the labeling spectrum:

- k-means: Given the online setting, the algorithm is run on batches of the data, setting the number of clusters to 10. For the real-world datasets (CF and AB) k-means has been initialized with 5 clusters, because there are fewer items per batch compared to the other datasets. For each batch $bt_i \in Bt$ of the data stream, the final centers obtained from the previous batch serve to initialize the centers of the current batch.
- Discriminative Online (Good?) Matlab Algorithms (DOGMA) [43]: The following algorithms are considered: PA-I [16], RBP and Perceptron [14], Projectron [45], Projectron++ [45], Forgetron (Kernel-Based Perceptron) [18], and Online Independent Support Vector Machines (OISVM) [44]. Because these algorithms are fully supervised, they are trained on all labeled data that is allowed by the budget B.

Running *k-means* on the different datasets produces the results shown in Table 3. CQM is calculated considering that k-means requires no queries ($Q = 0$). Items of a cluster are assigned the label of the majority. This assignment is performed after each batch and it is the base for computing the quality measures. It can be seen that for SSMD, k-means produces lower CQM compared to those of GD. This is also true in the case of AOMPC. Considering Figs. 4 and 5, it can be

TABLE 3
K-means: Avg. Results for GD, SSMD, CF, and AB

	Q	VM	ER	CQM
GD	0	0.8270	2.8750	0.9337
SSMD	0	0.8143	4.7216	0.9207
CF	0	0.9608	0.9235	0.9836
AB	0	0.9477	1.3056	0.9778

seen that AOMPC performs well. Comparing the results of k-means in Table 3 with the results of AOMPC in Table 5, the AOMPC values represent a good performance: AOMPC processes each data point only once and then discards it, whereas k-means uses all data points for computation. Clearly, the CQM values in Table 3 for CF and AB are very high, caused by low values of ER. For CF and AB, we used the same batch size (i.e., every 30 minutes) as for the generated SSMD dataset. More often, only a handful items are contained in the individual batches. Due to the small number of items per batch, it is not possible that relevant and irrelevant items are highly mixed within the created clusters of each batch. Hence, assignments are clear/unambigious.

The results of *DOGMA* algorithms related to the datasets are displayed in Table 4 for the best and worst cases. Details on the remaining algorithms can be found in Appendix C, available in the online supplemental material. Note that the DOGMA algorithms operate with the maximum amount of labels given by the budget. Hence, the training data is as large as the maximum number of items allowed by the budget. The CQM value is calculated such that $Q = B \cdot \#items$. The evaluation measures are computed based on each batch for comparison. DOGMA algorithms are trained based on randomly selected items from the dataset in advance. To ensure a fair comparison of DOGMA algorithms against AOMPC, we applied a 10-cross-validation strategy. The results in Table 4 show that in the case of GD, most of the DOGMA algorithms produce lower CQM compared to AOMPC results, which are illustrated in Fig. 4. It is an indication that the DOGMA algorithms are inefficient when dealing with changes in data, like the one artificially introduced in batch-4 of GD (see Fig. 2 of Section 4.1). In case of SSMD, CQM values obtained by most of the DOGMA algorithms (see Table 4) look similar to those values corresponding to the best active learning method of AOMPC (see Fig. 5 “with DCN” active learning methods). OISVM and PA-I produce the best performance on SSMD. In all, AOMPC performs well for on-the-fly querying. The DOGMA results related to CF and AB are also given in Table 4. Considering CQM as representative measure, DOGMA produced similar results to those produced by AOMPC shown in Figs. 6 and 7.

In a nutshell, AOMPC shows good performance compared to DOGMA, although the selection of items to query is performed on-the-fly. In addition, DOGMA algorithms use fully labeled data, while AOMPC uses only a subset of labeled data whose size is upper bounded by the budget.

TABLE 4
Best and *worst* CQM of DOGMA Algorithms (GD, SSMD, CF, AB)

		Q	B	VM	ER	CQM
GD	Forgetron	80	0.1	0.3029	32.5500	***0.6081***
	OISVM	80	0.1	0.8084	3.2625	**0.9062**
	RBP	160	0.2	0.3188	31.9500	***0.5959***
	OISVM	160	0.2	0.8217	2.9000	**0.8920**
	Forgetron	240	0.3	0.4100	25.3625	***0.6362***
	OISVM	240	0.3	0.8153	3.0250	**0.8695**
	RBP	320	0.4	0.2099	38.6750	***0.4896***
	OISVM	320	0.4	0.8180	2.9750	**0.8505**
	RBP	400	0.5	0.4811	20.9000	***0.6398***
	OISVM	400	0.5	0.8157	3.0250	**0.8296**
SSMD	PA-I	123	0.1	0.7228	5.4406	**0.8696**
	Projectron++	123	0.1	0.4202	11.5303	***0.7484***
	Projectron++	246	0.2	0.4105	10.5367	***0.7305***
	OISVM	246	0.2	0.8427	10.1921	**0.8619**
	PA-I	369	0.3	0.7636	2.2302	**0.8579**
	Forgetron	369	0.3	0.5593	9.7172	***0.7592***
	RBP	492	0.4	0.5025	9.0046	***0.7257***
	OISVM	492	0.4	0.8834	5.0767	**0.8596**
	PA-I	615	0.5	0.8647	1.2505	**0.8532**
	RBP	615	0.5	0.6244	5.3916	***0.7604***
CF	PA-I	98	0.1	0.7631	17.5100	**0.8214**
	Projectron++	98	0.1	0.7137	28.4213	***0.7520***
	PA-I	196	0.2	0.7728	15.9354	**0.8122**
	RBP	196	0.2	0.7141	23.7132	***0.7557***
	PA-I	294	0.3	0.8039	13.8672	**0.8118**
	Forgetron	294	0.3	0.7180	29.8722	***0.7060***
	PA-I	392	0.4	0.8222	12.7396	**0.8030**
	Forgetron	392	0.4	0.7117	28.5864	***0.6906***
	PA-I	490	0.5	0.8405	11.3371	**0.7955**
	Forgetron	490	0.5	0.7353	24.1613	***0.6998***
AB	PA-I	106	0.1	0.6791	22.9801	**0.7688**
	Projectron++	106	0.1	0.6440	32.6142	***0.7101***
	PA-I	212	0.2	0.7094	20.9924	**0.7678**
	Forgetron	212	0.2	0.6643	29.6821	***0.7109***
	PA-I	318	0.3	0.7428	17.6217	**0.7747**
	RBP	318	0.3	0.6707	27.3168	***0.7046***
	PA-I	424	0.4	0.7751	16.0927	**0.7721**
	Forgetron	424	0.4	0.6870	24.4803	***0.7037***
	Forgetron	530	0.5	0.7086	22.5930	***0.6996***
	OISVM	530	0.5	0.8087	13.6702	**0.7743**

4.6 Discussion and Future Work

The advantage of AOMPC compared to the other algorithms is the continuous processing of data streams and incremental update of knowledge, where the existing prototypes act as memory for the future. Here forgetting of outdated knowledge is controlled by α, which also depends on the budget. Learning serves to adapt and/or create clusters in a continuous way. The algorithm queries labels on-the-fly for continuously updating the classification model. In summary, it can be said that budget B and threshold UT are related to each other. Increasing their values increases the quality of the algorithm. B has also an influence on the number of clusters that are created (i.e., the more often the user is asked, the more hints for new clusters are given).

The advantage of our algorithm compared to the others is the transferred knowledge from one batch to the next creating a continuous view on the arriving data. The already known prototypes act as memory (i.e., forgetting is based on α and learning is based on the new creation of clusters, see Algorithm 1).

In terms of performance, Table 5 shows the best results of AOMPC for different budget values using the *CQM* measure. For GD, the variable learning rate α and the fixed α rate

TABLE 5
Best Results of AOMPC based on Budget B

	B	Query strategies	α (β for var. α)	Q (Q/#items)	VM	ER	CQM
GD	0.1	SCN	0.03	79.0 (0.10)	0.8460	2.3750	0.9222
	0.2	SCN	1/2 (4)	113.0 (0.14)	0.9180	1.2500	0.9409
	0.3	SCN	1/2 (4)	114.0 (0.14)	0.9180	1.2500	0.9406
	0.4	SCN	1/2 (4)	114.0 (0.14)	0.9180	1.2500	0.9406
	0.5	SCN	1/2 (4)	114.0 (0.14)	0.9180	1.2500	0.9406
SSMD	0.1	CVCN with DCN	0.03	113.0 (0.09)	0.7080	12.2120	0.8329
	0.2	SCN	1/3(1)	140.0 (0.11)	0.8440	12.2762	0.8690
	0.3	SCN	0.03	300.0 (0.24)	0.9161	8.8391	0.8817
	0.4	CVCN with DCN	0.01	256.0 (0.21)	0.8640	5.8791	0.8881
	0.5	CVCN with DCN	0.03	238.0 (0.19)	0.8876	9.4269	0.8804
CF	0.1	SCN	1/2 (2)	27.0 (0.03)	0.7451	18.0411	0.8278
	0.2	CVCN	1/2 (2)	32.0 (0.03)	0.7463	18.0141	0.8273
	0.3	**RCN**	2/3 (2)	223.0 (0.23)	0.8050	13.4949	0.8283
	0.4	SCN	0.03	297.0 (0.30)	0.8261	11.6488	0.8287
	0.5	SCN	0.03	297.0 (0.30)	0.8261	11.6488	0.8287
AB	0.1	CVCN with DCN	0.01	117.0 (0.11)	0.6669	31.4934	0.7204
	0.2	CVCN with DCN	0.03	215.0 (0.20)	0.7325	27.7243	0.7403
	0.3	SCN	0.01	304.0 (0.29)	0.7383	22.7398	0.7501
	0.4	CVCN with DCN	0.01	343.0 (0.33)	0.7607	18.8053	0.7690
	0.5	CVCN	0.03	380.0 (0.36)	0.7728	17.4619	0.7723

in the case of SSMD show good performance. For CF, the variable learning rate seems to be more suitable considering the number of queries. AOMPC produces good results on AB using a fixed learning rate. The reason is that the data items are very similar and that changes within the textual data happen slowly and near the boundary. Finally, comparing the active learning strategies (*"DCN"* options), we can notice that very good performance is achieved especially for SSMD and CF. The quality of clustering increases even for low values of B.

Overall, AOMPC shows a quite good performance (see Tables 4, 3, and 5), despite the fact that it operates online and handles labeling just-in-time. Moreover, AOMPC was run on batches just for the sake of feature selection (see Section 3.3). AOMPC can run in purely point-based online mode (i.e., item-by-item) as well. In the future, we plan to extend this algorithm by deleting clusters when they lose their importance. This could also be done for features in order to obtain an evolving feature space. We also plan to implement a variable budget strategy so that, for instance, the number of queries (i.e., budget) is bigger for cold-start and gets reduced afterward, depending on the uncertainty and the performance of the algorithm. Finally, it would be interesting to identify drift, without defining a threshold, but by considering the general case, where classes are non-contiguous.

5 Conclusion

This paper presents a streaming analysis framework for distinguishing between relevant and irrelevant data items. It integrates the user into the learning process by considering the active learning mechanism. We evaluated the framework for different datasets, with different parameters and active learning strategies. We considered synthetic datasets to understand the behavior of the algorithm and real-world social media datasets related to crises. We compared the proposed algorithm, AOMPC, against many existing algorithms to illustrate the good performance under different parameter settings. As explained in Section 4.6, the algorithm can be extended to overcome many issues, for instance by considering: dynamic budget, dynamic deletion of stale clusters, and generalization to handle non-contiguous class distribution.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°261817 and was partly performed in the Lakeside Labs research cluster at Alpen-Adria-Universität Klagenfurt. A. Bouchachia was supported by the European Commission under the Horizon 2020 Grant 687691 related to the Project PROTEUS: Scalable Online Machine Learning for Predictive Analytics and Real-Time Interactive Visualization.

References

[1] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao, "Semantics + Filtering + Search = Twitcident. Exploring Information in Social Web Streams," in *Proc. 23rd ACM Conf. Hypertext Social Media*, 2012, pp. 285–294.

[2] U. Ahmad, A. Zahid, M. Shoaib, and A. AlAmri, "Harvis: An integrated social media content analysis framework for youtube platform," *Inf. Syst.*, vol. 69, pp. 25–39, 2017.

[3] G. Backfried, J. Gollner, G. Qirchmayr, K. Rainer, G. Kienast, G. Thallinger, C. Schmidt, and A. Peer, "Integration of media sources for situation analysis in the different phases of disaster management: The QuOIMA project," in *Proc. Eur. Intell. Security Informat. Conf.*, Aug 2013, pp. 143–146.

[4] BBC News Europe, England Riots: Maps and Timeline, 2012, Aug. [Online]. Available: http://www.bbc.co.uk/news/uk-14436499

[5] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 291–300.

[6] J. Bezdek, T. Reichherzer, G. Lim, and Y. Attikiouzel, "Multiple-prototype classifier design," *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.*, vol. 28, no. 1, pp. 67–79, Feb. 1998.

[7] M. Biehl, B. Hammer, and T. Villmann, "Prototype-based models in machine learning," *Wiley Interdisciplinary Reviews: Cognitive Sci.*, vol. 7, no. 2, pp. 92–111, 2016.

[8] A. Bouchachia, "Learning with incrementality," in *Proc. Int. Conf. Neural Inf. Process.*, 2006, pp. 137–146.

[9] A. Bouchachia, "Incremental learning with multi-level adaptation," *Neurocomputing*, vol. 74, no. 11, pp. 1785–1799, 2011.

[10] A. Bouchachia and C. Vanaret, "Incremental learning based on growing gaussian mixture models," in *Proc. 10th Int. Conf. Mach. Learn. Appl. Workshops*, Dec. 2011, vol. 2, pp. 47–52.

[11] A. Bouchachia and C. Vanaret, "GT2FC: An online growing interval type-2 self-learning fuzzy classifier," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 4, pp. 999–1018, Aug. 2014.

[12] M.-R. Bouguelia, Y. Belaïd, and A. Belaïd, "An adaptive streaming active learning strategy based on instance weighting," *Pattern Recognit. Lett.*, vol. 70, pp. 38–44, 2016.

[13] M. Büscher and M. Liegl, "Connected communities in crises," in H. Hellwagner, D. Pohl and R. Kaiser(ed.), "Social Media Analysis for Crisis Management" IEEE Comput. Soc. Special Technical Community on Social Networking E-Letter, Mar. 2014, vol. 2, no. 1.

[14] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Tracking the best hyperplane with a simple budget perceptron," *Mach. Learn.*, vol. 69, no. 2–3, pp. 143–167, 2007.

[15] L. Chen, K. S. M. Tozammel Hossain, P. Butler, N. Ramakrishnan, and B. A. Prakash, "Syndromic surveillance of flu on Twitter using weakly supervised temporal topic models," *Data Mining Knowl. Discovery*, vol. 30, no. 3, pp. 681–710, May 2016.

[16] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, Dec. 2006.

[17] S. Dashti, L. Palen, M. P. Heris, K. M. Anderson, S. Anderson, and S. Anderson, "Supporting disaster reconnaissance with social media data: A design-oriented case study of the 2013 colorado floods," in *Proc. 11th Int. Conf. Inform. Syst. Crisis Response Manag.*, 2014, pp. 632–641.

[18] O. Dekel, S. Shalev-Shwartz, and Y. Singer, "The forgetron: A kernel-based perceptron on a fixed budget," in *NIPS*. Cambridge, MA, USA: MIT Press, 2005, pp. 259–266.

[19] A. Denecke, H. Wersing, J. Steil, and E. Körner, "Online figure-ground segmentation with adaptive metrics in generalized LVQ," *Neurocomputing*, vol. 72, no. 7–9, pp. 1470–1482, 2009.

[20] S. Denef, P. S. Bayerl, and N. Kaptein, "Social media and the police - Tweeting practices of british police forces during the August 2011 riots," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, May 2013, pp. 3471–3480.

[21] N. Dufty, "Using social media to build community disaster resilience," *Australian J. Emergency Manag.*, vol. 27, no. 1, pp. 40–45, 2012.

[22] M. Freeman and A. Freeman, "Bonding over bushfires: Social networks in action," in *Proc. IEEE Int. Symp. Technol. Soc.*, Jun. 2010, pp. 419–426.

[23] J. A. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 44:1–44:37, 2014.

[24] B. Hammer, D. Hofmann, F.-M. Schleif, and X. Zhu, "Learning vector quantization for (dis-)similarities," *Neurocomputing*, vol. 131, pp. 43–51, 2014.

[25] S. Hao, J. Lu, P. Zhao, C. Zhang, S. C. H. Hoi, and C. Miao, "Second-order online active learning and its applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1338–1351, Jul. 2018.

[26] S. Hao, P. Hu, P. Zhao, S. C. H. Hoi, and C. Miao, "Online active learning with expert advice," *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 5, pp. 58:1–58:22, 2018.

[27] S. R. Hiltz, B. van de Walle, and M. Turoff, "The domain of emergency management information," in *Proc. Inf. Syst. Emergency Manag.*, 2010, vol. 16, pp. 3–19.

[28] D. Ienco, A. Bifet, I. Žliobaitė, and B. Pfahringer, "Clustering based active learning for evolving data streams," in *Discovery Sci.*, J. Fürnkranz, E. Hüllermeier, and T. Higuchi, Eds. Berlin, Germany: Springer, 2013, vol. 8140, pp. 79–93.

[29] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "AIDR: Artificial intelligence for disaster response," in *Proc. Companion Publication 23rd Int. Conf. World Wide Web*, Apr. 2014, pp. 159–162.

[30] Y. Ishikawa, Y. Chen, and H. Kitagawa, "An on-line document clustering method based on forgetting factors," in *Research and Advanced Technology for Digital Libraries*, P. Constantopoulos and I. T. Solvberg, Eds., vol. 2163, Berlin, Germany: Springer, 2001, pp. 325–339.

[31] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464 –1480, Sep. 1990.

[32] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, 1966, Art. no. 707.

[33] R. Li, K. H. Lei, R. Khadiwala, and K.-C. Chang, "TEDAS: A Twitter-based event detection and analysis system," in *Proc. IEEE 28th Int. Conf. Data Eng.*, 2012, pp. 1273–1276.

[34] S. Liu, L. Palen, J. Sutton, A. Hughes, and S. Vieweg, "In search of the bigger picture: The emergent role of on-line photo-sharing in times of disaster," in *Proc. 5th Int. ISCRAM Conf.*, 2008, pp. 140–149.

[35] L. Ma, S. Destercke, and Y. Wang, "Online active learning of decision trees with evidential data," *Pattern Recognit.*, vol. 52, pp. 33–45, 2016.

[36] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[37] S. Mohamad, A. Bouchachia, and M. Sayed-Mouchaweh, "A bi-criteria active learning algorithm for dynamic data streams," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 74–86, Jan. 2018.

[38] S. Mohamad, M. Sayed-Mouchaweh, and A. Bouchachia, "Active learning for classifying data streams with unknown number of classes," *Neural Netw.*, vol. 98, pp. 1–15, 2018.

[39] B. Mokbel, B. Paassen, F.-M. Schleif, and B. Hammer, "Metric learning for sequences in relational LVQ," *Neurocomputing*, vol. 169, pp. 306–322, 2015.

[40] B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden, "Scaling up crowd-sourcing to very large datasets: A case for active learning," *Proc. VLDB Endow.*, vol. 8, no. 2, pp. 125–136, Oct. 2014.

[41] V. K. Neppalli, C. Caragea, A. Squicciarini, A. Tapia, and S. Stehle, "Sentiment analysis during hurricane sandy in emergency response," *Int. J. Disaster Risk Reduction*, vol. 21, pp. 213–222, 2017.

[42] A. Olteanu, S. Vieweg, and C. Castillo, "What to expect when the unexpected happens: Social media communications across crises," in *Proc. ACM Conf. Comput. Supported Cooperative Work Social Comput.*, 2015, pp. 994–1009.

[43] F. Orabona, *DOGMA: A MATLAB Toolbox for Online Learning*, 2009. [Online]. Available: http://dogma.sourceforge.net

[44] F. Orabona, C. Castellini, B. Caputo, L. Jie, and G. Sandini, "On-line independent support vector machines," *Pattern Recognit.*, vol. 43, no. 4, pp. 1402–1412, 2010.

[45] F. Orabona, J. Keshet, and B. Caputo, "Bounded kernel-based online learning," *J. Mach. Learn. Res.*, vol. 10, pp. 2643–2666, Dec. 2009.

[46] S.-Y. Perng, M. Büscher, L. Wood, R. Halvorsrud, M. Stiso, L. Ramirez, and A. Al-Akka, "Peripheral response: Microblogging during the 22/7/2011 norway attacks," *Int. J. Inf. Syst. Crisis Response Manag.*, vol. 5, no. 1, pp. 41–57, 2013.

[47] D. Pohl, A. Bouchachia, and H. Hellwagner, "Online processing of social media data for emergency management," in *Proc. Int. Conf. Mach. Learn. Appl.*, Dec. 2013, vol. 2, pp. 333–338.

[48] D. Pohl, "Social media analysis for crisis management: A brief survey," in H. Hellwagner, D. Pohl, and R. Kaiser, (ed.), Social Media Analysis for Crisis Management, IEEE Comput. Soc. Special Technical Community on Social Networking E-Letter, Mar. 2014, vol. 2, no. 1.

[49] D. Pohl, A. Bouchachia, and H. Hellwagner, "Social media for crisis management: Clustering approaches for sub-event detection," *Multimedia Tools Appl.*, vol. 74, pp. 3901–3932, 2013.

[50] D. Pohl, A. Bouchachia, and H. Hellwagner, "Online indexing and clustering of social media data for emergency management," *Neurocomputing*, vol. 172, pp. 168–179, 2016.

[51] J. R. Ragini, P. R. Anand, and V. Bhaskar, "Mining crisis information: A strategic approach for detection of people at risk through social media analysis," *Int. J. Disaster Risk Reduction*, vol. 27, pp. 556–566, 2018.

[52] C. Reuter and M. Kaufhold, "Fifteen years of social media in emergencies: A retrospective review and future directions for crisis informatics," *J. Contingencies Crisis Manag.*, vol. 26, no. 1, pp. 41–57, 2018.

[53] T. Reuter and P. Cimiano, "Event-based classification of social media streams," in *Proc. 2nd ACM Int. Conf. Multimedia Retrieval*, 2012, pp. 22:1–22:8.

[54] T. Reuter, P. Cimiano, L. Drumond, K. Buza, and L. Schmidt-Thieme, "Scalable event-based clustering of social media via record linkage techniques," in *Proc. 5th Int. Conf. Weblogs Social Media*, 2011, pp. 313–320.

[55] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison. 2009.

[56] E. Shook, K. Leetaru, G. Cao, A. Padmanabhan, and S. Wang, "Happy or not: Generating topic-based emotional heatmaps for culturomics using CyberGIS," in *Proc. IEEE 8th Int. Conf. E-Sci.*, Oct. 2012, pp. 1 –6.

[57] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, "Stream-based active learning for sentiment analysis in the financial domain," *Inf. Sci.*, vol. 285, pp. 181–203, Apr. 2014.

[58] K. Starbird and J. Stamberger, "Tweak the Tweet: Leveraging microblogging proliferation with a prescriptive syntax to support citizen reporting," in *Proc. 7th Int. ISCRAM Conf.*, May 2010, pp. 1–5.

[59] T. Terpstra, A. de Vries, R. Stronkman, and G. L. Paradies, "Towards a realtime Twitter analysis during crises for operational crisis management," in *Proc. 9th Int. ISCRAM Conf.*, Apr. 2012.

[60] M. F. Umer and M. S. H. Khiyal, "Classification of textual documents using learning vector quantization," *Inf. Technol. J.*, vol. 6, no. 1, pp. 154–159, 2007.

[61] S. Vieweg and A. Hodges, "Rethinking context: Leveraging human and machine computation in disaster response," *Comput.*, vol. 47, no. 4, pp. 22–27, Apr. 2014.

[62] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: What Twitter may contribute to situational awareness," in *Proc. Int. Conf. Human Factors Comput. Syst.*, 2010, pp. 1079–1088.

[63] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE Trans. Neural Netw. Learn. Sys.*, vol. 25, no. 1, pp. 27–39, Jan. 2014.

[64] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[65] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power, "Emergency situation awareness from Twitter for crisis management," in *Proc. Int. Workshop Social Web Disaster Manag.*, 2012, Art. no. 1.

[66] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power, "Using social media to enhance emergency situation awareness," *IEEE Intell. Sys.*, vol. 27, no. 6, pp. 52–59, Nov.-Dec. 2012.

Daniela Pohl received the Dipl.-Ing. master's degree in computer science from the Alpen-Adria-Universität Klagenfurt, Austria, in 2008, and the doctoral degree from the Alpen-Adria-Universität Klagenfurt, in 2015. She worked as research assistant with the scope of the EU-funded FP7 project BRIDGE (www.bridgeproject.eu) to develop technical solution to improve crisis management. Her research interests include information retrieval and machine learning.

Abdelhamid Bouchachia is a professor at Bournemouth University, Department of Computing, United Kingdom. His major research interests include machine learning and computational intelligence with a particular focus on scalable online/incremental learning, semi-supervised and active learning, prediction systems, and uncertainty modelling. He published numerous papers in international journals and conferences and edited several special issues and volumes. He founded and served as the general chair of the International Conference on Adaptive and Intelligent Systems (ICAIS) for many years. He currently serves as a program committee member for many conferences and is acting as associate editor of *Evolving Systems* as well as member of the Evolving Intelligent Systems (EIS) Technical Committee (TC) of the IEEE Systems, Man and Cybernetics Society and member of the IEEE Task-Force for Adaptive and Evolving Fuzzy Systems and the IEEE Computational Intelligence Society. He is a senior member of the IEEE.

Hermann Hellwagner is a full professor of informatics with the Institute of Information Technology (ITEC), Klagenfurt University, Austria, leading the Multimedia Communications Group. His current research areas are distributed multimedia systems, multimedia communications, and quality of service. He has received many research grants from national (Austria and Germany) and European funding agencies as well as from industry, is the editor of several books, and has published more than 250 scientific papers on parallel computer architecture, parallel programming, and multimedia communications and adaptation. He is a senior member of the IEEE, and a member of the ACM and OCG (Austrian Computer Society). He was vice president of the Austrian Science Fund (FWF).