

Context-Aware Collaborative Data Stream Mining in Ubiquitous Devices

João Bártolo Gomes¹, Mohamed Medhat Gaber²,
Pedro A.C. Sousa³, and Ernestina Menasalvas^{1,*}

¹ Facultad de Informatica - Universidad Politecnica Madrid, Spain

`joao.bartolo.gomes@alumnos.upm.es`

`emenasalvas@fi.upm.es`

² School of Computing, University of Portsmouth, England

`mohamed.gaber@port.ac.uk`

³ Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal

`pas@fct.unl.pt`

Abstract. Recent advances in ubiquitous devices open an opportunity to apply new data stream mining techniques to support intelligent decision making in the next generation of ubiquitous applications. This paper motivates and describes a novel Context-aware Collaborative data stream mining system *CC-Stream* that allows intelligent mining and classification of time-changing data streams on-board ubiquitous devices. *CC-Stream* explores the knowledge available in other ubiquitous devices to improve local classification accuracy. Such knowledge is associated with context information that captures the system state for a particular underlying concept. *CC-Stream* uses an ensemble method where the classifiers are selected and weighted based on their local accuracy for different partitions of the instance space and their context similarity in relation to the current context.

Keywords: Collaborative Data Stream Mining, Context-awareness, Concept Drift, Ubiquitous Knowledge Discovery.

1 Introduction and Motivation

The increasing technological advances and popularity of ubiquitous devices, such as smart phones, PDAs (Personal Digital Assistants) and wireless sensor networks, open an opportunity to perform intelligent data analysis in such ubiquitous computing environments [12,7,10].

This work is focused on collaborative data stream mining on-board ubiquitous devices in complex ubiquitous environments. The goal is to learn an anytime, anywhere classification model that represents the underlying concept from a stream of labelled records [6,7]. Such model is used to predict the label of the incoming unlabelled records. However, in real-world ubiquitous applications, it

* The research is partially financed by project TIN2008-05924 of Spanish Ministry of Education.

is common for the underlying concept of interest to change over time [20]. The problem of learning from time-changing data streams is generally known in the literature as concept drift [25,11,19,20,8]. An effective data stream mining system must recognise and adapt to changes by continuously learning the different time-changing concepts [20,8]. Therefore, learning systems should be able to adapt to concept changes without explicitly being informed about such changes. For example, using available contextual features [24,9] or the performance of the base learner [8] as a technique to adapt to concept drift.

Context awareness is an important part of ubiquitous computing [17,5,1]. In most ubiquitous applications concepts are associated with context, this means that they may reappear when a similar context occurs. For example, a weather prediction model usually changes according to the seasons. The same applies with product recommendations or text filtering models where the user interest and behaviour might change over time due to fashion, economy, spatial-temporal situation or any other context [25,9,20]. This has motivated some works [24,9] to analyse how to use context to track concept changes.

In this work, we propose to use the knowledge available in other devices and context information to collaboratively improve the accuracy of local predictions. The data mining problem is assumed to be the same in all the devices, however the feature space and the data distributions are not static, as assumed in traditional data mining approaches [13,20]. We are interested in understanding how context and the knowledge available in other devices can be integrated to improve local predictive accuracy in a ubiquitous data stream mining scenario [7].

As an illustrative example, collaborative spam filtering [2] is one of the possible applications for the proposed collaborative learning approach. Each ubiquitous device learns and maintains a local filter that is incrementally learnt from a local data stream based on features extracted from the incoming mails. In addition, user usage patterns and feedback are used to supervise the filter that represents the target concept (i.e., the distinction between spam and ham). The ubiquitous devices collaborate by sharing their knowledge with others, which can improve local predictive accuracy. Furthermore, the dissemination of knowledge is faster, as devices new to the mining task, or that have access to fewer labelled records, can anticipate spam patterns that were observed in the community, but not yet locally. Moreover, the privacy and computational issues that would result from sharing the original mail are minimised, as only the filters are shared. Consequently, this increases the efficiency of the collaborative learning process.

However, many challenges arise from this scenario, the major ones are:

- how the knowledge from the community can be exploited to improve local predictiveness, while resolving possible conflicts, as ultimately the local knowledge should have priority over the community (i.e., user interests can be unique);
- how to integrate context information in the collaborative learning process; and
- how to adapt to changes in the underlying concept and provide anytime, anywhere data stream classification in ubiquitous devices.

To address these challenges, in this paper, we propose an incremental ensemble approach (*CC-Stream*) where the models from the community are selected and weighted based on their local accuracy for different partitions of the instance space and their context similarity to the current context.

The rest of the paper is organised as follows. The following Section reviews the related work. Section 3 provides the preliminaries of the approach and the problem definition, which is followed by the description of the *CC-Stream* system in Section 4. Finally, in Section 5, our conclusions and future work are presented.

2 Related Work

2.1 Collaborative Data Stream Mining

In collaborative and distributed data mining, the data is partitioned and the goal is to apply data mining to different, usually very small and overlapping, subsets of the entire data [3,18]. In this work, our goal is not to learn a global concept, but to learn from other devices their concepts, while maintaining a local or subjective point of view. Wurst and Morik [26] explore this idea by investigating how communication among peers can enhance the individual local models without aiming at a common global model. The motivation is similar to what is proposed in domain adaptation [4] or transfer learning [16]. However, these assume a batch scenario. When the mining task is performed in a ubiquitous environment [7,10], an incremental learning approach is required.

In ubiquitous data stream mining, the feature space of the records that occur in the data stream may change over time [13] or be different among devices [26]. For example, in a stream of documents where each word is a feature, it is impossible to know in advance which words will appear over time, and thus what is the best feature space to represent the documents with. Using a very large vocabulary of words results inefficient, as most of the words will likely be redundant and only a small subset of words is finally useful for classification. Over time, it is also likely that new important features appear and that previously selected features become less important, which brings change to the subset of relevant features. Such change in the feature space is related to the problem of concept drift, as the target concept may change due to changes in the predictiveness of the available features.

2.2 Context-Aware Data Mining

Context representation in information systems is a problem studied by many researchers as they attempt to formally define the notion of context. Schmidt et al. [17] defined context as the knowledge about the users and device state. Moreover, Dey [5] defines context as ‘Context is any information that can be used to characterise the situation of an entity’. In contrast, Brezillon and Pomerol [1] argue that there is no particular knowledge that can be objectively called context, as context is in the eye of the beholder. They state that ‘knowledge that can be qualified as ‘contextual’ depends on the context!’. In addition, Padovitz

et al. [15] proposed a general approach that models context using geometrical spaces called Context Spaces, which allows the inference of situations in context-aware environments. The context spaces representation is used for the context representation in the system proposed in this paper.

Context dependence has been recognised as a problem in several real world domains [22,24,9]. Turney [22] was among the first ones to introduce the problem of context in machine learning. Widmer [24] exploits what is referred as contextual clues (based on the Turney [22] definition of primary/contextual features) and proposes a meta-learning method to identify such clues. Contextual clues are context-defining attributes or combinations of attributes whose values are characteristic of the underlying concept. When more or less systematic changes in their values are observed this might indicate a change in the target concept. The method automatically detects contextual clues on-line, and when a potential context change is signalled, the knowledge about the recognised context clues is used to adapt the learning process in some appropriate way. However, if the *hidden context* is not represented in the contextual clues it is not possible to detect and adapt to change.

2.3 Ensemble Approaches for Data Stream Mining

Ensemble approaches have been applied successfully to improve classification accuracy in data mining problems and particularly in data streams, where the underlying concept changes [19,23,14]. *CC-Stream* is a context-aware ensemble approach to exploit the community knowledge in a ubiquitous data stream mining scenario [7]. The proposed system is most related in terms of the learning algorithm to what has been proposed in [27] and [21] as both approaches consider concept drift, select the best classifier for each instance based on its position in the instance space, and able to learn from data streams. However, their base classifiers are learnt from chunks of a stream of trained records in a sequential method, while we explore how the knowledge available in the ubiquitous environment can be used and adapted to represent the current underlying concept for a given device. Moreover, these approaches do not consider the integration of context.

3 Preliminaries

Predictive and Context Feature Space definitions. Let X be the space of all attributes and its possible values and Y the set of possible (discrete) class labels.

According to the general idea of Turney [22] and to Widmer [24] definition:

Definition 1 (Predictive features). A feature (attribute-value combination) $a_i : v_{ij}$ is predictive if $p(c_k | a_i = v_{ij})$ is significantly different from $p(c_k)$ for some class $c_k \in Y$.

Definition 2 (Predictive attributes). An attribute a_i is predictive if one of its values v_{ij} (i.e., some feature $a_i : v_{ij}$) is predictive.

Definition 3 (Contextual features). A feature $a_i : v_{ij}$ is contextual if it is predictive of predictive features, i.e., if $p(a_k : v_{kl} | a_i = v_{ij})$ is significantly different from $p(a_k : v_{kl})$ for some feature $a_k : v_{kl}$ that is predictive.

Definition 4 (Contextual attributes). An attribute a_i is contextual if one of its values v_{ij} is contextual.

Such notions are based on a probability distribution for the observed classes given the features. However, when the probability distribution is unknown it is often possible to use background knowledge to distinguish between predictive and contextual features [22].

3.1 Problem Definition

This work is focused on collaborative data stream mining between ubiquitous devices. Each ubiquitous device d aims to learn the underlying concept from a local stream DS_d of labelled records where the set of class labels Y is fixed. However, the feature space X is not static. Let $X_d^i \subseteq X$ be the feature space for DS_d and its i^{th} record $X_i = (\mathbf{x}_i, y_i)$ with $x_i \in X_d^i$ and $y_i \in Y$. We assume that the underlying concept is a function f_d that assigns each record x_i to the true class label y_i . This function f_d can be approximated using a data stream mining algorithm to train a model m_d at device d from the DS_d labelled records. The model m_d returns the class label of an unlabelled record \mathbf{x} , such that $m_d(\mathbf{x}) = y \in Y$. The aim is to minimise the error of m_d (i.e., the number of predictions different from f_d). However, the underlying concept of interest f_d may change over time and the number of labelled records available for that concept are sometimes limited. To address such situations, we propose to exploit models from the community and use the available labelled records from DS_d to obtain a model m_d . We expect m_d to be more accurate than using the local labelled records alone when building the model. The incremental learning of m_d should adapt to changes in the underlying concept. Moreover, we assume context information is available in the ubiquitous devices.

3.2 Context Integration

In situations where contextual information is related to the underlying concepts, such knowledge could be exploited to improve the adaptation to the underlying concept. Nevertheless, these relations are not known a priori, and it is even possible that given the available context information it is not possible to find such relations. Still, in many real-world problems we find examples where the available context information does not explain all global concept changes, but partially explains some of these. For example, user preferences often change with time or location. Imagine a user that has different interests during the weekends, weekdays, when at home or at work. In general, different concepts can recur due

to periodic context (e.g., seasons, locations, week days) or non-periodic context (e.g., rare events, fashion, economic trends).

When exploiting context relations to improve local predictive accuracy, one could argue that context information should be simply added as additional attributes in the base learner. However, that would increase the problem dimensional complexity and introduce noise to the learning process, since concepts may change due to factors that may not be expressed as context attributes. Therefore, we believe that such context information should be integrated carefully in a meta-learning level (as discussed in [22,24,9]).

3.3 Context Representation

The context representation and similarity we propose to integrate in our approach is inspired on the Context Spaces model [15], where a context state is represented as an object in a multidimensional Euclidean space. A context state c_i in context space C is defined as a tuple of N context attribute-values,

$$c_i = (a_1^i, \dots, a_n^i)$$

where a_n^i represents the value of context attribute a_n for the i^{th} context state $c_i \in C$.

The available context information depends on the learning environment and data mining problem. Context information can represent simple sensors (e.g., time, temperature, humidity) or a more complex context (e.g., season, location, gait) defined by domain experts or inferred by other means beyond the scope of the problem discussed in this work. We assume that at anytime a context provider can be asked the current context state.

Context Similarity. Context similarity is not a trivial problem [15], because while it could be more immediate to measure the (dis)similarity between two values in a continuous attribute, the same is not that easy when we consider categorical ones and to a greater extent when integrating the heterogeneous attributes similarity into a (dis)similarity measure between context states. For the purposes of this work the degree of similarity between context states c_i and c_j , using the Euclidean distance is defined as:

$$|c_i - c_j| = \sqrt{\sum_{K=1}^N dist(a_k^i - a_k^j)}$$

where a_k^i represents the k^{th} attribute-value in context state c_i . For numerical attributes distance is defined as:

$$dist(a_k^i, a_k^j) = \frac{(a_k^i - a_k^j)^2}{s^2}$$

where s is the estimated standard deviation for a_k . For nominal attributes distance is defined as:

$$dist(a_k^i, a_k^j) = \begin{cases} 0 & \text{if } a_k^i = a_k^j \\ 1 & \text{otherwise} \end{cases}$$

4 CC-Stream

In this work, a context-aware collaborative learning system *CC-Stream* for ubiquitous data stream mining is proposed. *CC-Stream* uses an ensemble to combine the knowledge of different models, in order to represent the current underlying concept.

There is a large number of ensemble methods to combine models, which can be roughly divided into: i) voting methods, where the class that gets most votes is chosen [19,23,14]; and ii) selection methods, where the "best" model for a particular instance is used to predict the class label [27,21].

Algorithm 1. *CC-Stream* Training

Require: Data stream DS of labelled records, Context Provider $context$, window w of records.

```

1: repeat
2:    $c_i = context.getCurrentContext();$ 
3:   Add next record  $DS_i$  from  $DS$  to  $w$ ;
4:   if  $w \rightarrow numRecords > wMaxSize$  then
5:      $forget(w \rightarrow oldestRecord);$ 
6:   end if
7:    $r = getRegion(DS_i);$ 
8:   for all  $Model \rightarrow m_j$  do
9:      $prediction := m_j.classify(DS_i);$ 
10:    if  $prediction = DS_i \rightarrow class$  then
11:       $updateRegionCorrect(r, m_j);$ 
12:       $addContext(c_i, m_j);$ 
13:    end if
14:     $updateRegionTotal(r, m_j);$ 
15:  end for
16: until END OF STREAM

```

The *CC-Stream* system uses a selection method that partitions the instance space X into a set of regions R . For each region, an estimate of the models' accuracy is maintained over a sliding window. This estimated value is updated incrementally as new labelled records are observed in the data stream or new models are available. This process (in Algorithm 1), can be considered a meta-learning task where we try to learn for each model from the community how it best represents the local underlying concept for a particular region $r_i \in R$. Furthermore, for each context state obtained from the context provider the models that are accurate for that context are associated with it. Figure 1 shows the regions that result from the partition of the instance space. Moreover, it shows an illustrative context space with some possible context states that are associated with particular models (i.e., different colours).

For the classification of a new unlabelled record x_i , *CC-Stream* uses the best model prediction. The best model is considered to be the one that is more accurate for the partition r_i that contains the new record and that is associated

Algorithm 2. *CC-Stream* Classification**Require:** Data stream DS of unlabelled records, Context Provider $context$.

```

1: repeat
2:    $c_i = context.getCurrentContext()$ ;
3:   Get  $DS_i$  from  $DS$ ;
4:    $r := getRegion(DS_i)$ ;
5:   for all  $Model \rightarrow m_j$  do
6:      $model := argmax_j(getAccuracy(m_j, r), getContext(m_j, c_i))$ ;
7:   end for
8:   return  $model.classify(DS_i)$ ;
9: until END OF STREAM

```

with a context similar to the current one, as detailed in Algorithm 2. The accuracy for a region r_i is the average accuracy for each value of its attributes. The next sections explain how the regions are created using the attribute values, how the base learner deals with the dynamic feature space in each device and how context is integrated.

4.1 Creating Regions

An important part of *CC-Stream* is to learn for each region of the instance space X which model m_j performs better. This way m_j predictions can be used with confidence to classify incoming unlabelled records that belong to that particular region.

The instance space can be partitioned in several ways, here we follow the method used by Zhu et al. [27], where the partitions are created using the different values of each attribute. For example, if an attribute has two values, two

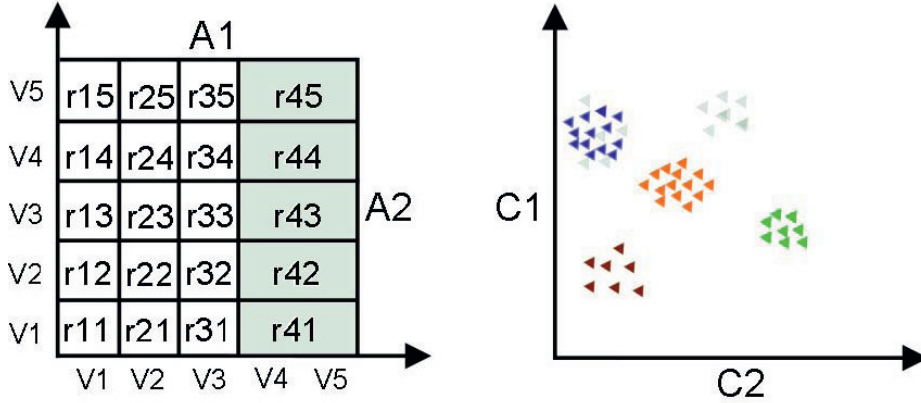


Fig. 1. Partition the instance space into regions and classifiers associated context states

estimators of how the classifiers perform for each value are kept. If the attribute is numeric, it is discretised and the regions use the values that result from the discretisation process. This method has shown good results and it represents a natural way of partitioning the instance space. However, there is an increased memory cost associated with a larger number of regions. To minimise this cost the regions can be partitioned into higher granularity ones, aggregating attribute values into a larger partition. This is illustrated in Figure 1, where the values $V4$ and $V5$ of attribute $A1$ are grouped into regions $r41$ to $r45$.

4.2 Base Learner

In *CC-Stream* the knowledge base of each device uses local knowledge and knowledge received from the community. The local model is incrementally built using the data stream incoming records. This model is treated similarly to the models received from the community. Its global performance is monitored and it is expected to increase with the number of training records [6]. Once its performance is stable the model can be shared with other devices.

The base learner algorithm is used to learn a model that represents the data stream underlying concept. Any classification algorithm able to learn incrementally and to handle a dynamic feature space (i.e., each feature is treated independently) can be used for this task. The decision of the base learner algorithm can be made according to the nature of data to be mined, choosing the algorithm that best suits it (e.g., high accuracy, handles noise, memory consumed, faster processing time). Some popular algorithms that can be used as base learner are the Naive Bayes and Nearest Neighbour Algorithms [24,13].

4.3 Integration of Context

To exploit the context information available in ubiquitous devices, we propose to learn which models are able to represent the current underlying concept given a certain context state. When a new labelled records is processed in the data stream the context provider is asked for the current context state. For each context state we keep estimators of model performance (described in Algorithm 1). Such information is used when the system is asked to classify a new unlabelled record. This procedure is described in Algorithm 2. In line 6 we are interested in maximising the following function

$$\begin{aligned} \operatorname{argmax}_j(\operatorname{getAccuracy}(m_j, r), \operatorname{getContext}(m_j, c_i)) = \\ w_r \times \operatorname{getAccuracy}(m_j, r) + w_c \times \operatorname{getContext}(m_j, c_i) \end{aligned}$$

where the $\operatorname{getAccuracy}(m_j, r)$ is the accuracy of model m_j for region r and $\operatorname{getContext}(m_j, c_i)$ is the context performance of model m_j in situations of a context similar to the occurring context state c_i . This is calculated using the context state c_j that is associated with model m_j and that is the nearest (i.e., smallest distance, using the distance function defined in Section 3.3) to c_i . The w_r and w_c are weights given to the different factors of the selection function.

In contrast with the information about models accuracy for the different regions of the instance space that is kept over a sliding window (because of concept drift), context information associated with the models is kept over the entire learning process as it will capture higher level patterns (i.e., the relation between context and models/concepts) of the long-term process [24].

Resource Awareness. Resource-awareness is an important issue in ubiquitous data stream mining [7]. In such a dynamic ubiquitous usage scenario, it is possible for *CC-Stream* to receive more knowledge from the community over time that it can keep in memory. In such situations we propose to discard past models that have the lowest performance to allow the integration of new, promising models.

5 Conclusions and Future Work

This paper discusses collaborative data stream mining in ubiquitous environments to support intelligent decision making in the next generation of ubiquitous applications. We propose the *CC-Stream* system, an ensemble approach combines the knowledge of different models available in the ubiquitous environment. It incrementally learns which models are more accurate for certain regions of the feature space and context space. *CC-Stream* is able to locally adapt to changes in the underlying concept using a sliding window of the models estimates for each region. Moreover, as base learner *CC-Stream* uses incremental algorithms that consider each individual feature independently. This way it is able to handle the heterogeneity across the devices models and able to deal with the data stream dynamic feature space.

As future work we would like to evaluate *CC-Stream* with synthetic and real data sets in terms of accuracy and resource consumption. The context similarity function is something we also plan to evaluate and improve. Moreover, we plan to develop a news recommender application that uses *CC-Stream* to deliver personalised content to a smart phone user while reducing information overload.

Acknowledgments. The work of J.P. Bártolo Gomes is supported by a Phd Grant of the Portuguese Foundation for Science and Technology (FCT) and a mobility grant from Consejo Social of UPM that made possible his stay at the University of Portsmouth. This research is partially financed by project TIN2008-05924 of Spanish Ministry of Science and Innovation. Thanks to the FCT project KDUDS (PTDC/EIA-EIA/98355/2008).

References

1. Brezillon, R., Pomerol, J.C.: Contextual knowledge sharing and cooperation in intelligent assistant systems. *Travail Humain* 62, 223–246 (1999)
2. Cortez, P., Lopes, C., Sousa, P., Rocha, M., Rio, M.: Symbiotic Data Mining for Personalized Spam Filtering. In: *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, WI-IAT 2009*, vol. 1, pp. 149–156. IEEE, Los Alamitos (2009)

3. Datta, S., Bhaduri, K., Giannella, C., Wolff, R., Kargupta, H.: Distributed data mining in peer-to-peer networks. *IEEE Internet Computing*, 18–26 (2006)
4. Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research* 26(1), 101–126 (2006)
5. Dey, A.K., Abowd, G.D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction* 16(2), 97–166 (2001)
6. Domingos, P., Hulten, G.: Mining high-speed data streams. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 71–80. ACM, New York (2000)
7. Gaber, M.M., Krishnaswamy, S., Zaslavsky, A.: Ubiquitous data stream mining. In: *Current Research and Future Directions Workshop Proceedings held in conjunction with The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Sydney, Australia*. Citeseer (2004)
8. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) *SBIA 2004*. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
9. Harries, M.B., Sammut, C., Horn, K.: Extracting hidden context. *Machine Learning* 32(2), 101–126 (1998)
10. Hotho, A., Pedersen, R., Wurst, M.: Ubiquitous Data. In: May, M., Saitta, L. (eds.) *Ubiquitous Knowledge Discovery*. LNCS, vol. 6202, pp. 61–74. Springer, Heidelberg (2010)
11. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 97–106. ACM, New York (2001)
12. Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., Dull, J., Sarkar, K., Klein, M., Vasa, M., et al.: VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring. In: *Proceedings of SIAM International Conference on Data Mining*, vol. 334 (2004)
13. Katakis, I., Tsoumakas, G., Vlahavas, I.: On the utility of incremental feature selection for the classification of textual data streams. In: Bozanis, P., Houstis, E.N. (eds.) *PCI 2005*. LNCS, vol. 3746, pp. 338–348. Springer, Heidelberg (2005)
14. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: An ensemble method for drifting concepts. *The Journal of Machine Learning Research* 8, 2755–2790 (2007)
15. Padovitz, A., Loke, S.W., Zaslavsky, A.: Towards a theory of context spaces. In: *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004*, pp. 38–42 (2004)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 1345–1359 (2009)
17. Schmidt, A., Beigl, M., Gellersen, H.W.: There is more to context than location. *Computers & Graphics* 23(6), 893–901 (1999)
18. Stahl, F., Gaber, M.M., Bramer, M., Yu, P.S.: Pocket Data Mining: Towards Collaborative Data Mining in Mobile Computing Environments. In: *2010 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, vol. 2, pp. 323–330. IEEE, Los Alamitos (2010)
19. Street, W.N., Kim, Y.S.: A streaming ensemble algorithm (SEA) for large-scale classification. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 377–382. ACM, New York (2001)
20. Tsymbal, A.: The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin* (2004)

21. Tsymbal, A., Pechenizkiy, M., Cunningham, P., Puuronen, S.: Dynamic integration of classifiers for handling concept drift. *Inf. Fusion* 9, 56–68 (2008)
22. Turney, P.D.: Exploiting context when learning to classify. In: Brazdil, P.B. (ed.) *ECML 1993*. LNCS, vol. 667, pp. 402–407. Springer, Heidelberg (1993)
23. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: *Proceedings of the Ninth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, pp. 226–235. ACM, New York (2003)
24. Widmer, G.: Tracking context changes through meta-learning. *Machine Learning* 27(3), 259–286 (1997)
25. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. *Machine learning* 23(1), 69–101 (1996)
26. Wurst, M., Morik, K.: Distributed feature extraction in a p2p setting—a case study. *Future Generation Computer Systems* 23(1), 69–75 (2007)
27. Zhu, X., Wu, X., Yang, Y.: Effective classification of noisy data streams with attribute-oriented dynamic classifier selection. *Knowl. Inf. Syst.* 9, 339–363 (2006)