Artificial Intelligence 209 (2014) 11–28

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Concept drift detection via competence models

Ning Lu, Guangquan Zhang*, Jie Lu

Decision Systems & e-Service Intelligence (DeSI) Lab, Centre for Quantum Computation & Intelligent Systems (QCIS), Faculty of Engineering and Information Technology, University of Technology, Sydney, PO Box 123, Broadway, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 1 November 2012
Received in revised form 16 October 2013
Accepted 9 January 2014
Available online 10 January 2014

Keywords:
Concept drift
Competence model
Case-base maintenance
Incremental supervised learning
Classification

ABSTRACT

Detecting changes of concepts, such as a change of customer preference for telecom services, is very important in terms of prediction and decision applications in dynamic environments. In particular, for case-based reasoning systems, it is important to know when and how concept drift can effectively assist decision makers to perform smarter maintenance operations at an appropriate time. This paper presents a novel method for detecting concept drift in a case-based reasoning system. Rather than measuring the actual case distribution, we introduce a new competence model that detects differences through changes in competence. Our competence-based concept detection method requires no prior knowledge of case distribution and provides statistical guarantees on the reliability of the changes detected, as well as meaningful descriptions and quantification of these changes. This research concludes that changes in data distribution do reflect upon competence. Eight sets of experiments under three categories demonstrate that our method effectively detects concept drift and highlights drifting competence areas accurately. These results directly contribute to the research that tackles concept drift in case-based reasoning, and to competence model studies.

1. Introduction

Learning under concept drift poses an additional challenge to existing learning algorithms. Instead of considering all the past training data, or making a *stationary distribution assumption* [1–3], an effective learner should be able to track these changes and quickly adapt to them [4]. Otherwise, as concept drifts, the induced pattern may not be relevant to the new data [5,6], which may result in an increasing number of errors [7].

The issue of concept drift refers to the change of distribution underlying the data [4,8]. More formally, the problem can be framed as follows. If we denote the feature vector as x and the class label as y, then the data stream will be an infinite sequence of (x, y). If the concept drifts, it means the distribution of $p(x, y)$ is changing between the current data chunk and the yet-to-come data. If we decompose $p(x, y)$ into the following two parts as $p(x, y) = p(x) \times p(y|x)$, we could say there are two sources of concept drift: one is $p(x)$, which evolves with time t, and can also be written as $p(x|t)$, and the other is $p(y|x)$, the conditional probability of feature x [2].

Concept drift can be categorized into two basic types: virtual concept drift (or drift in data distribution), and real concept drift (or drift in decision concepts) [8]. Other kinds of concept drift have also been defined and discussed; for example, based on the extent of drift, Stanley [9] mentioned three kinds of drift: sudden drift, moderate drift and slow drift. Based on class distribution, Forman [10] classified concept drift into three categories: shifting class distribution – shifting the distribution among categories but remaining stable within a given class; shifting sub-class distribution – shifting distribution sub-classes

* Corresponding author.
E-mail addresses: Ning.Lu@uts.edu.au (N. Lu), Guangquan.Zhang@uts.edu.au (G. Zhang), Jie.Lu@uts.edu.au (J. Lu).

0004-3702/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.artint.2014.01.001

within a category, but remaining stable within a given sub-class; and fickle concept drift – individual cases may take on different ground truth labels at different times. Zhang et al. [2] defined and analyzed two kinds of concept drift in their studies: loose concept drifting, in which the genuine concepts remain relatively stable, whereas the vision of the drift is mainly caused by the biased observation of instances; and rigorous concept drifting, in which the genuine concepts undergo continuous change, although such changes can worsen in the face of biased observation. Tsymbal et al. [11] discussed a special scenario of concept drift called local concept drift, by which they mean that the change of concept or data distribution occurs only in some regions of the instance space.

Based on the literature, many learning algorithms have been used as base models to handle concept drift. These include rule-based learning [4,8,12], decision trees and their incremental versions [5,13,14], info-fuzzy networks [15], clustering [16], support vector machines [17], and case-based reasoning (CBR) [11,18–20]. Among them, the CBR method has three reported advantages for handling the concept drift problem [21]. First, CBR performs well with disjointed concepts. Second, CBR, as a lazy learner, is easy to update. Third, CBR allows easy sharing of knowledge for particular types of problems, making it easier to maintain multiple distributed case-bases. Therefore, this study focuses on concept drift detection for CBR.

According to a literature review [22], the first attempt to handle concept drift with the case-based technique was IB3 [20], which discards noisy and outdated cases by monitoring each case's accuracy and retrieval frequency. IB3 has been criticized for being suitable only for gradual concept drift, and for its costly adaptation process [4]. The Locally Weighted Forgetting (LWF) algorithm [23], which reduces the weights of the k-nearest neighbors of a new case and discards a case if its weight falls below a threshold θ, was believed to be one of the best adaptive learning algorithms of its time. Klinkenberg [17] later showed in his experiments that instance weighting techniques tend to overfit the data and perform more poorly than analogous instance selection techniques. Elwell and Polikar [24] presented an ensemble learning algorithm for non-stationary environments (Learn++.NSE) that assumes data are incrementally acquired in batches. For each incoming dataset D^t, their algorithm trains an independent base classifier h^t which is forced to emphasize misclassified data by adjusted data weighting. Then, a weight is assigned to each base classifier based on its performance on the latest dataset. The final classification result is determined by weighted majority voting of all base classifiers. Recent research and development in case-base maintenance (CBM) provides a number of methods for updating all knowledge containers [25] of a CBR system. Among them, the competence-based CBM methods [19,26–30] and case-base mining technologies [31] have been empirically shown to be capable of preserving the competency of a CBR system while removing noisy and redundant cases. Few studies, however, discussed when to trigger maintenance operations, which is also an important consideration, according to Wilson and Leake's CBM framework [32]. In addition, current methods are incapable of distinguishing between noisy cases and cases representing a new concept. Knowing whether concept drift happens could help to recognize obsolete cases that conflict with current concepts and distinguish noise cases from novel cases. Moreover, developing a detection method that is able to explain where and how concept drifts could facilitate further decision capabilities and be suitable for handling local concept drift problems [11].

Motivated by these issues, we propose a new method of concept drift detection for CBR systems, which compares the case distributions of existing cases with newly available cases. The proposed method requires no prior knowledge about the case distribution, but estimates the probability distribution and detects change via a competence model. Besides determining whether there is a concept drift, our method also quantifies and describes the detected change in terms of the competence model. To the best of our knowledge, no literature has reported any research that uses a competence model for concept drift detection purposes.

Compared with other famous non-parametric methods, our detection method demonstrates the following advantages: 1) it can be easily adopted in multi-dimensional data while maintaining similar results in one-dimensional data; 2) it is more stable and achieves better results as shown in experiments, especially for small samples, because data can share distribution contributions among related competence areas, rather than splitting strictly by cutting edges, which makes it more tolerable to sample bias; 3) it is able to describe the detected changes by highlighting some competence areas, which is testified by a real world application.

The novelty and main contribution of this paper lie in the endeavor to discover the difference between the inner nature of the competence group model and our proposed competence closure model. The detailed definitions and theorems afford other researchers an inside view of case-base competence, which has never been discussed in the literature. The theoretical study provided in this paper also reveals the essential differences between the two competence models, and identifies three important aspects of the competence closure model which the competence group model does not possess. Compared with our previous work on competence-based concept drift detection [33], which aims to investigate the impact of concept drift on case-base competence and assert to the possibility of detecting change via competence models, this paper additionally conducts a tremendous number of experiments to thoroughly evaluate our proposed concept drift detection approach.

This paper is organized as follows. Section 2 discusses related works. Section 3 proposes the new competence model and discusses the relationship between our model and current competence models. Section 4 presents the competence-based change detection method. Section 5 determines the critical region and provides a statistical guarantee for the proposed detection method. Section 6 outlines the results of the experimental evaluation. Section 7 concludes this study, with a discussion of future work.

2. Related work

This section formally presents the problem of concept drift detection (Section 2.1), and analyzes the pros and cons of established literature with regard to concept drift detection (Section 2.2).

2.1. Problem description: Concept drift detection

Concept drift detection can be formulated as follows. Suppose there is a CBR system listening to a data stream where each new observation is represented by $c_i = (x_i, y_i)$, where $x_i = (x_{i_1}, x_{i_2}, \dots, x_{i_n}) \in X$ is the feature vector, $y_i \in Y$ is the target label. As it is unrealistic to store the full history of the stream, we base our concept drift detection algorithm on a two-sliding-window paradigm. Both windows contain a number of successive data points. We assume data points within each window are independent random samples taken from two unknown, multi-dimensional, non-parametric distributions F and F', respectively. We then define the null hypothesis H_0, which asserts that F and F' are identical. The goal is to design a proper statistical test that is able to not only refuse H_0, if it is not true, but also highlight some local regions of the problem space where H_0 does not hold and quantify the difference between F and F'. When H_0 is true, the probability of making an error (where the test says that F and F' are different when in fact they are not) should be, at most, α, where α is a user-supplied parameter.

A real world scenario for the application of our method would be spam filtering. As is well-known, one of the challenges in the spam filtering domain is to handle concept drift problems. In a case-based spam filtering system where emails are continuously classified, the problem arises of how we can benefit from the available feedback (new cases) and improve the accuracy of the system. Treating the newest emails as an independent training set, e.g., emails received during the last month, we can detect whether there is a concept drift between our existing case-base that is assumed to follow an unknown distribution F', and the most recent emails, which are assumed to follow an unknown distribution F'. The correct maintenance can accordingly be carried out when a drift has been reported.

2.2. Change detection methods

The most popular trigger technique for learner adaptivity is change detection, which is often implicitly related to a sudden drift [34]. This is usually conducted by a statistical test that monitors the raw data distribution [35–38], the outputs (error) of learners [39–42], or the parameters of the learners [43].

2.2.1. Detecting concept drift by data distribution

When comparing two samples and determining whether these samples are drawn from the same distribution, the Wilcoxon test [44] and the Kolmogorov–Smirnov test [45,46] are the most famous non-parametric methods. We do not assume that the data follows any particular parametric distribution (Section 2.1), since in real world applications, the data that one typically encounters may not arise from any standard distribution, which makes non-parametric tests more practical. However, the Wilcoxon test and the Kolmogorov–Smirnov test are initially designed for data with only one dimension and cannot be easily extended to multi-dimensional data, which limits their scalability [36]. In this sense, we intentionally omit methods that are designed for one-dimensional data [37,38].

Kifer, Ben-David and Gehrke [35] proposed a modification of the Kolmogorov–Smirnov test that, in principle, compares the cumulative distribution functions of two samples with all possible orderings and takes the largest resulting test statistics. They employed a notation of $\mathcal{A}$-distance as their test statistic, which in fact is a relaxation of the *total variation* distance. Their method reported several advantages, including being able to control the rate of false alarm (*false positive*) and missed detection (*false negative*), and to describe and quantify the detected change. Some technical challenges remain, however, which need to be overcome before putting their work into practice, such as how to determine an interesting class of sets $\mathcal{A}$ in higher dimensions.

Dasu et al. [36] suggested an information-theoretic approach for change detection in data streams, which resorts to the *Kullback–Leibler* divergence to measure the difference between two given distributions. They further estimated whether their measurement is statistically significant through the *Percentile Bootstrap* method [47]. By partitioning the problem space using a *kdq-tree*, their method also exhibited the capability of identifying the regions of greatest difference. However, the *kdq-tree* does not guarantee that a partition will coincide with the real interesting concepts. This means that the detected regions may not be easily explained and understood.

2.2.2. Detecting concept drift by learner outputs

Gama et al. [39] presented a Drift Detection Method (DDM) that traces and controls the online error-rate of the learning algorithm. Treating the error of a set of examples as a random variable from Bernoulli trails, the probability for the number of errors in a sample of n examples can be generalized as Binomial distribution. A significant increase in the error of the algorithm suggests that the class distribution is changing. Their method declares a new concept if the error reaches the warning level, and a new model is learnt when the drift level is exceeded. Although being independent of the learning algorithm, their method is more suitable for rebuilding models rather than updating an existing model, since it assesses a learner through its overall error rate. In addition, their method is criticized for having difficulties when the change is slowly gradual [40].

Baena-García et al. [40] proposed the Early Drift Detection Method (EDDM) to improve the detection in the presence of gradual concept drift. Their EDDM, which is different to DDM, considers the distance between two consecutive erroneous classifications instead of the error rate. They assume that a significant decrease in the distance suggests that the concept is changing. With the calculated average distance between two errors (p'_i) and its standard deviation (s'_i), they defined two thresholds α for a warning level and β for a drift level. When $\beta \leqslant \frac{(p'_i+2\times s'_i)}{(p'_{\max}+2\times s'_{\max})} < \alpha$, where $p'_{\max}$ and $s'_{\max}$ are stored value when $p'_i + 2 \times s'_i$ reaches its maximum, the examples will be stored in case of a possible change of context; when $\frac{(p'_i+2\times s'_i)}{(p'_{\max}+2\times s'_{\max})} < \beta$, a new model is learnt using the examples stored since the warning level was triggered. The values for $p'_{\max}$ and $s'_{\max}$ are reset. 'EDDM performs well for gradual changes; however, it is not good at detecting drift in noisy examples' [48].

Yasumura, Kitani and Uehara [41] developed a sensitive detection method for concept drift that measures the number of instances classified differently by two successive ensemble classifiers and determines whether the difference is significant. To suppress the influence caused by noise, they weighted the instances by taking the inverse weights generated by the AdaBoost algorithm [49], but their method requires building a new classifier each time a new chunk arrives. This is apparently not suitable for all non-ensemble based algorithms to handle concept drift.

Li et al. [42] also suggested tracking the error rate of classification in a chunked data stream. Considering the observed error rate of the latest data chunk $\bar{e}_f$ as a historical classification result, they fitted the estimated error rate of current chunk $\bar{e}_s$ into Hoeffding's inequality to provide a statistical guarantee for the detection. Nevertheless, they still experienced the same problem as occurred in Gama's work [39].

2.2.3. Detecting concept drift by parameters

To the best of our knowledge, the only attempt to model concept drift as a change of parameters was made by Su, Shen and Xu [43]. In their framework, a dynamic probabilistic model is framed as $p(C_k|x_t) = f(w_t) + v$, where w_t is an optimal parameter vector inferred continuously by the extended Kalman filter [50]; and v is a random variable that represents the uncertainty in the posterior distribution $p(C_k|x_t)$. Assuming the expectation of parameter w_{t-1} will be the same as w_t when there is no concept drift, they model the concept drift as a change of parameter vector $w_t = w_{t-1} + s$, where s is the uncertainty caused by concept drift. For simplicity, they assume that s and v follow zero mean normal distributions with isotropic covariance, $s \sim N(0, aI)$, $v \sim N(0, r)$, where I is the identity matrix and a is a single value which controls the variance of the parameter vector w; r is the noise variance. In the implementation of their model, the degree of concept drift aI and the noise variance r need to be estimated from data.

Their framework for modelling concept drift is creative and can be easily applied to many learning models, such as Support Vector Machines (SVM), Regression or Artificial Neural Networks (ANN), however, it is not suitable for a knowledge-based learner like CBR. In addition, an optimal parameter vector may not be organized in a user-understandable way, which prohibits concept drift interpretation.

In this section, we have summarized existing concept drift detection methods into three categories. As shown in later sections, our proposed method will belong to the first category. There is also research on learning methods in outlier and anomaly detection [51,52]. The difference here is that outlier detection is concerned with finding exceptional observations, whereas concept drift detection deals with identifying a shift in the underlying data distribution [53].

3. A new competence model

This study aims to provide an innovative solution for concept drift detection, which compares the data distribution through competence measurement instead of the feature space.

Competence is a measurement of how well a CBR system fulfills its goals. As CBR is a problem-solving methodology, competence is usually taken to be the proportion of problems at hand that can be solved successfully [54]. Because the competence measures the problem-solving capabilities of a CBR system, the probability distribution change of its cases should also reflect upon its competence. This inspired our research to detect concept drift through a competence model. The key idea is to measure the distribution change of cases with regard to their competencies instead of their real distributions.

Smyth and McKenna [26,28,55] proposed a series of models to measure the competence of a CBR system. The definitions are shown as follows.

Definition 1. (See [26].) For a case base $CB = \{c_1, c_2, \ldots, c_n\}$, given a case $c \in CB$, $\mathit{CoverageSet}(c) = \{c' \in CB\colon\ \mathit{Solves}(c, c')\}$, where $\mathit{Solves}(c, c')$ means that c can be retrieved and adapted to solve c'.

Definition 2. (See [26].) $\mathit{ReachibilitySet}(c) = \{c' \in CB\colon\ \mathit{Solves}(c', c)\}$.

Definition 3. (See [55].) $\mathit{RelatedSet}(c) = \mathit{CoverageSet}(c) \cup \mathit{ReachibilitySet}(c)$.

Definition 4. (See [55].) For $c_1, c_2 \in CB$, $\mathit{SharedCoverage}(c_1, c_2)$ iff

$$\bigl[\mathit{RelatedSet}(c_1) \cap \mathit{RelatedSet}(c_2)\bigr] \neq \phi. \tag{1}$$

Definition 5. (See [55].) Let $G = \{c_1, c_2, \ldots, c_m\} \subseteq CB$, we say that G has the property $\mathit{CompetenceGroup}(G)$ iff for any $c_i \in G$, there exists $c_j \in G - \{c_i\}$ so that $\mathit{SharedCoverage}(c_i, c_j)$ holds; and for any $c_k \in CB - G$, there does not exist $c_l \in G$, so that $\mathit{SharedCoverage}(c_k, c_l)$ holds.

In their competence model, *coverage* of a case is the set of problems that this case can solve; conversely, *reachability* is the set of all cases that can solve this case. A cluster of cases, called a competence group, is formed using their *reachability* and *coverage* sets. Based on these definitions, we put forward the following propositions:

Proposition 1. *For any $G \subseteq CB$ if G has property $\mathit{CompetenceGroup}(G)$ then $|G| > 1$.*

Proof. Obvious. □

This makes it difficult for the competence group model to measure and detect noise cases that are distinguished from their close neighbors; thus noise cases tend to solve and be solved only by themselves.

Proposition 2. *Given $G_1, G_2 \subseteq CB$, if G_1 and G_2 are with property $\mathit{CompetenceGroup}(G_1)$ and $\mathit{CompetenceGroup}(G_2)$ respectively, then $G_1 \cup G_2$ has property $\mathit{CompetenceGroup}(G_1 \cup G_2)$.*

Proof. For any $c_i \in G_1 \cup G_2$, $c_i \in G_1$ or $c_i \in G_2$. Without loss of generality, we suppose $c_i \in G_1$. Since G_1 has property $\mathit{CompetenceGroup}(G_1)$, there must exist $c_k \in G_1 \subseteq G_1 \cup G_2$ and $c_k \neq c_i$, so that $\mathit{SharedCoverage}(c_k, c_i)$ holds. And for any $c_j \in CB - \{G_1 \cup G_2\} \subseteq CB - G_1$, there does not exist $c_m \in G_1$, so that $\mathit{SharedCoverage}(c_j, c_m)$ holds. If there exists $c_m \in G_2$ and $c_j \in CB - \{G_1 \cup G_2\} \subseteq CB - G_2$, we have a contradiction where G_2 has property $\mathit{CompetenceGroup}(G_2)$. Therefore $G_1 \cup G_2$ has property $\mathit{CompetenceGroup}(G_1 \cup G_2)$. □

Proposition 3. *Given $G_1, G_2 \subseteq CB$, $G_1 \cap G_2 \neq \phi$, if G_1 and G_2 are with property $\mathit{CompetenceGroup}(G_1)$ and $\mathit{CompetenceGroup}(G_2)$ respectively, then $G_1 \cap G_2$ has property $\mathit{CompetenceGroup}(G_1 \cap G_2)$.*

Proof. For any $c_i \in G_1 \cap G_2$, $c_i \in G_1$ and $c_i \in G_2$. Since G_1 has property $\mathit{CompetenceGroup}(G_1)$, then there must exist $c_k \in G_1$ and $c_k \neq c_i$, so that $\mathit{SharedCoverage}(c_k, c_i)$ holds. If $c_k \notin G_2$, as $c_i \in G_2$, we have a contradiction where G_2 has property $\mathit{CompetenceGroup}(G_2)$. For any $c_j \in CB - G_1 \cap G_2$, $c_j \in CB - G_1$ or $c_j \in CB - G_2$. Without loss of generality, we suppose $c_j \in CB - G_1$. Since G_1 has property $\mathit{CompetenceGroup}(G_1)$, then does not exist $c_m \in G_1 \cap G_2 \subseteq G_1$, so that $\mathit{SharedCoverage}(c_j, c_m)$ holds. Therefore $G_1 \cap G_2$ has property $\mathit{CompetenceGroup}(G_1 \cap G_2)$. □

Through Proposition 2 and Proposition 3, we prove that the *competence group* model cannot fulfill the claim that "*competence groups* individually make an independent contribution to global competence" [56]. To be precise, the competence group model provides a dichotomous partition of the case-base, G and $CB - G$, so that cases of each partition together make a collectively independent contribution to overall case-base competence. The competence group model is useful for analyzing a competence independent sub-set of the case-base, but it is still inadequate and deficient. First, the competence group model does not guarantee a complete splitting of the case-base into several competence groups (inadequacy); in other words, $CB - G$ is not necessarily a competence group, e.g., it contains a disjointed case that is not related to any case. Second, although each competence group may be further split to find more independent partitions, which may finally lead to several smaller independent competence groups, the competence group model does not guarantee that any two competence groups are mutually independent (deficiency). As a result, the competence group model is inappropriate for analyzing the competence of the case-base, which may be composed of several disjointed partitions.

We therefore propose two new competence models – *Competence Closure* and *Related Closure* [33], because current competence models are not sufficient for concept drift detection purposes, although, with existing competence models, one can transfer the infinite case domain into a finite domain of *related sets*, which solves one difficulty of measuring the statistical distance between two case samples. We will demonstrate how our models are superior to existing models by the comparisons shown later in this section. We intentionally omit other proposed competence models such as the *LiabilitySet* [19] and the *Complexity* model [29,30], since they do not provide a measurement of what is solved in the problem space, but only how well or how surely the problems are solved.

Definition 6. (See [33].) For $G = \{c_1, c_2, \ldots, c_m\} \subseteq CB$, $G \neq \phi$, we say that G has the property $\mathit{CompetenceClosure}(G)$ if, and only if, for any $c_i, c_j \in G$, $c_i \neq c_j$, there exist $\{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\} \subseteq G$ so that $\mathit{SharedCoverage}(c_{i_j}, c_{i_{j+1}})$ $(j = 0, \ldots, k)$ holds, where $c_i = c_{i_0}$, $c_j = c_{i_{k+1}}$, and for any $c_k \in CB - G$, there does not exist $c_l \in G$, so that $\mathit{SharedCoverage}(c_k, c_l)$ holds.

In other words a competence closure is a group of cases that can connect to one another through a series of *SharedCoverage* relations, while a case not in this competence closure cannot have a *SharedCoverage* relation with cases in this competence closure.

Proposition 4. *For $G \in CB$, $|G| > 1$, if G has property CompetenceClosure(G) then G has property CompetenceGroup(G).*

Proof. Obvious. □

Remark. In Proposition 4, the condition of $|G| > 1$ is essential. When $|G| = 1$, G is a single case set $G = \{c_1\}$, which has property *CompetenceClosure*(G) if there does not exist $c_k \in CB - G$, so that *SharedCoverage*(c_1, c_k) holds. However, G cannot have property *CompetenceGroup*(G), since we cannot find an element other than c_1 in G.

Proposition 4 shows that the competence closure model is fully compatible with the competence group model because any competence closure with at least two cases also has the *CompetenceGroup* property, and is also a competence group. A single case point can never be modeled by the competence group model. Proposition 4 ensures that any method that adopts the competence group model can also be fitted with the competence closure model.

Theorem 1. *For $G_1, G_2, \ldots, G_n \subseteq CB$, $|G_i| > 1$, if G_i has property CompetenceClosure(G_i) then $\bigcup_{i=1}^{n} G_i$ has property CompetenceGroup$(\bigcup_{i=1}^{n} G_i)$.*

Proof. As for Proposition 2 and Proposition 4. □

Theorem 2. *For $G_1, G_2 \subseteq CB$, $G_1 \neq G_2$, if G_1 and G_2 have properties CompetenceClosure(G_1) and CompetenceClosure(G_2) then $G_1 \cap G_2 = \phi$.*

Proof. As $G_1 \neq G_2$, without loss of generality, we suppose there exists $c_1 \in G_1$ and $c_1 \notin G_2$. If $G_1 \cap G_2 \neq \phi$, then let $c_0 \in G_1 \cap G_2 \subset G_1$. Since G_1 has property *CompetenceClosure*(G_1), then there exists $\{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\} \subseteq G_1$, so that *SharedCoverage*$(c_{i_j}, c_{i_{j+1}})$ $(j = 0, \ldots, k)$ holds, where $c_0 = c_{i_0}$, $c_1 = c_{i_{k+1}}$. As $c_1 \in G_1$ and $c_1 \notin G_2$ and $c_0 \in G_2$, there must exist c_{i_m}, $c_{i_{m+1}}$ $(0 \leqslant m \leqslant k)$, $c_{i_m} \in G_2$, $c_{i_{m+1}} \notin G_2$, $c_{i_{m+1}} \in G_1$, so that *SharedCoverage*$(c_{i_m}, c_{i_{m+1}})$ holds. However, this conflicts with the condition of G_2, which has the property *CompetenceClosure*(G_2). □

Compared with the *competence group* model, our *competence closure* model does not permit sharing competence coverage between *competence closures* and truly makes an independent contribution to global competence. We claim that this is an important characteristic because partitioning the space independently assists in the estimation of empirical probability, e.g., $P(A) + P(B) = P(A \cup B)$, when A and B are mutually exclusive.

Corollary 1. *For $G_1 \subseteq CB$, if G_1 has property CompetenceClosure(G_1) then there does not exist $G_2 \subseteq CB$, $G_1 \subset G_2$ so that G_2 has property CompetenceClosure(G_2).*

Proof. As for Theorem 2. □

Corollary 1 proves the later claim that 'a competence closure is a maximal set of cases, which are competence related'.

Proposition 5. *For $G \subseteq CB$, $G_1 \subseteq G$, G has property CompetenceGroup(G) and G_1 has property CompetenceClosure(G_1). Let $G_2 = G - G_1$, if $G_2 \neq \phi$, then G_2 has property CompetenceGroup(G_2).*

Proof. For any $c_i \in G_2 = G - G_1$, as G has property *CompetenceGroup*(G), there exists $c_j \in G$, so that *SharedCoverage*(c_i, c_j) holds. As $G = G_1 \cup G_2$ and $G_1 \cap G_2 = \phi$, we have $c_j \in G_1$ or $c_j \in G_2$. If $c_j \in G_1$, this conflicts with G_1, which has property *CompetenceClosure*(G_1). Therefore, $c_j \in G_2$, that is for any $c_i \in G_2$, there exists $c_j \in G_2$, so that *SharedCoverage*(c_i, c_j) holds.

For any $c_j \in CB - G_2 = (CB - G) \cup G_1$, $c_j \in CB - G$ or $c_j \in G_1$. If $c_j \in CB - G$, since G has property *CompetenceGroup*(G), there does not exist $c_m \in G_2 \subset G$, so that *SharedCoverage*(c_j, c_m) holds. Or, if $c_j \in G_1$, there exists $c_m \in G_2 = G - G_1$, so that *SharedCoverage*(c_j, c_m) holds. This will conflict with G_1, which has property *CompetenceClosure*(G_1). □

Theorem 3. *For $G \subseteq CB$, if G has property CompetenceGroup(G) and G does not have property CompetenceClosure(G), then there exists $\bigcup_{i=1}^{n} G_i = G$, so that G_i has property CompetenceClosure(G_i).*

Proof. For any $c_i \in G$, since G has property *CompetenceGroup*(G), but does not have property *CompetenceClosure*(G), then there must exist $c_j \in G$, so that we cannot find $\{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\} \subseteq G$ so that *SharedCoverage*$(c_{i_j}, c_{i_{j+1}})$ $(j = 0, \ldots, k)$ holds, where $c_i = c_{i_0}$, $c_j = c_{i_{k+1}}$. We continuously remove c_j from G until we cannot find a c_j, to construct a $G_1 \subset G$. Since G has

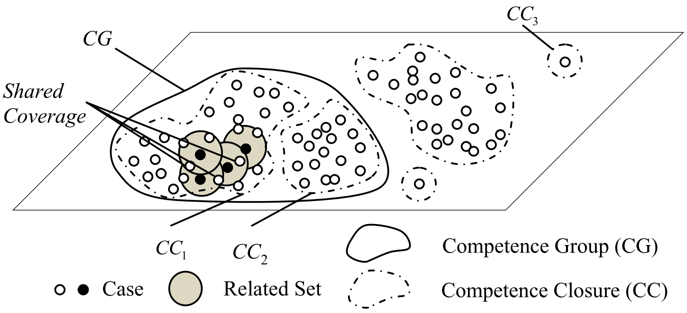


Fig. 1. Competence closure vs. competence group.

property *CompetenceGroup*(G), there at least exist $c_i, c_k \in G_1$, so that *SharedCoverage*(c_i, c_k) holds, and we have $G_1 \neq \phi$. Then, for any $c_m \notin G_1$, if there exists $c_l \in G_1$, so that *SharedCoverage*(c_m, c_l) holds, then for any $c_i \in G_1$ we can find $\{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\} \subseteq G_1$ so that *SharedCoverage*$(c_{i_j}, c_{i_{j+1}})$ $(j = 0, \ldots, k)$ holds where $c_m = c_{i_0}$, $c_l = c_{i_1}, c_i = c_{i_{k+1}}$. Therefore, $c_m \in G_1$, which conflicts with $c_m \notin G_1$. So, for any $c_m \notin G_1$, there does not exist $c_l \in G_1$, so that *SharedCoverage*(c_m, c_l) holds. We then conclude $G_1 \subset G$, and G_1 has property *CompetenceClosure*(G_1). Let $G' = G - G_1$, according to Proposition 5, G' has property *CompetenceGroup*(G') if $G' \neq \phi$. We can construct G_{m+1} continuously from $G' = G - \bigcup_{i=1}^{m} G_i$, $m = 1, 2, \ldots, n-1$, until $G_{m+1} = G - \bigcup_{i=1}^{m} G_i$, and G_{m+1} has property *CompetenceClosure*(G_{m+1}). □

As the entire case-base can be viewed as a competence group, Theorem 3 certifies that any case-base can be split into a set of competence closures.

To sum up, the *competence closure* model outweighs the *competence group* model because of its features, as follows:

First, the *competence closure* model is able to uniquely model the entire case-base. Considering the whole case-base as a *competence group*, according to Theorem 3, the case-base can be modeled into a set of *competence closures*. A suspicious case, which differs from its neighbors and solves nothing and can only be solved by itself, cannot be modeled by the *competence group* model. This is important because suspicious cases are more likely to behave like noisy or novel cases [57].

Second, the competence closure model provides a smaller granularity than the competence group model (Theorem 1 and Theorem 3). This is essential for competence-guided case discovery [56], in which the competence holes are believed to exist between neighbor competence groups.

Third, competence closure is the maximum set concerning a series of related problems (Corollary 1) and two competence closures are said to be independent in terms of the related set (Theorem 2). This facilitates the narrowing of the analysis within any interesting sub-problem space represented by one or more competence closures.

We also show the differences between our competence closure model and the existing competence group model in Fig. 1. It can be seen that the *competence closure* is defined as the maximal set of cases linked through their *related sets*, where a *competence group* is the union of *competence closures* or any *competence closure* with more than one case.

In Smyth and McKenna's [26,28,55] competence model, the competence contributed by a certain case is modeled by its *coverage set* (Definition 1). Meanwhile, the existence of a case could also be a support for those cases that solve it: in other words, its *reachability set* (Definition 2). The *related set* provides a measurement of related competence for each single case. Intuitively, the *related set* highlights a set of interesting target problems related to a case. However, since the *related set* overlaps, the *related set* of a certain case may not be the only measurement of problem space that relates to this case. In this sense, we propose another competence model – *related closure* (Definition 7) to represent the problem space with relation to a certain case or a group of cases, in terms of *related sets*.

Definition 7. (See [33].) For $c \in CB$, denote the *RelatedSet*(c) with regard to CB as $R^{CB}(c)$, and we define the *Related Closure* of c with regard to CB as

$$\Re^{CB}(c) = \{R^{CB}(c_i) \colon \forall c_i \in CB, \exists R^{CB}(c_i) \text{ s.t. } c \in R^{CB}(c_i)\} \tag{2}$$

For a group of cases $S \subseteq CB$, we define the *Related Closure* of S with regard to CB as

$$\Re^{CB}(S) = \bigcup_{c \in S} \Re^{CB}(c) \tag{3}$$

Example 1. Let $CB = \{c_1, c_2, c_3, c_4\}$, $R^{CB}(c_1) = \{c_1, c_2\}$, $R^{CB}(c_2) = \{c_1, c_2, c_3, c_4\}$, $R^{CB}(c_3) = R^{CB}(c_4) = \{c_2, c_3, c_4\}$. The *related closure* of c_3 is the set of all *related sets*, with regard to CB, which contain the case c_3. That is $\Re^{CB}(c_3) = \{\{c_1, c_2, c_3, c_4\}, \{c_2, c_3, c_4\}\}$.

We have now proposed a new competence model. We have not only discussed in more detail the properties of Smyth and McKenna's [26,28,55] competence model, but we have also shown, theoretically, how our competence model is more suitable for change detection purposes. In the following section, we introduce a metric that measures the distance between case chunks, and based on this metric, we develop our change detection method.

4. Competence-base empirical distance

When mining concept drifting data, a common assumption is that the up-to-date data chunk and the yet-to-come data chunk share identical, or considerably close distributions [1]. This means that the newly available cases represent the concept that we may be interested in, in the future. In CBR, considering cases in the existing case base and the newly available cases as two samples drawn from two probability distributions, we are able to identify whether there is a concept drift by detecting a possible distribution change between the existing case base and the newly available case chunk. This section proposes a weighted approach to measure the difference between case chunks that weight the *related sets* differently, according to the distribution of cases. The competence-based empirical distance between two case chunks is defined through those weights.

Given a case base CB, and two case sample sets $S_1, S_2 \subseteq CB$, we obtain two *related closures*, $\mathfrak{R}^{CB}(S_1)$ and $\mathfrak{R}^{CB}(S_2)$. Intuitively, we measure the difference between $\mathfrak{R}^{CB}(S_1)$ and $\mathfrak{R}^{CB}(S_2)$ as the distance between S_1 and S_2. However, it will only represent the distance between the competencies covered by these two samples. The relative distribution discrepancy within the competence is missing. This introduces a problem when we compare two case samples that solve similar problems, but with dramatically different distributions. To address this problem, we assign a weight for each element in $\mathfrak{R}^{CB}(S_1)$ and $\mathfrak{R}^{CB}(S_2)$ to represent the relative density of the cases distributed over their *related closures*.

Definition 8. Let $\mathfrak{R}^{CB}(S) = \{r_1^{CB}(S), r_2^{CB}(S), \ldots, r_n^{CB}(S)\}$, $\mathfrak{R}_i^{CB}(S) = \{r_i^{CB}(S)\}$, we define the *density* of $r_i^{CB}(S)$ with regard to S as

$$w^*\bigl(r_i^{CB}(S)\bigr) = \frac{1}{|S|} \times \sum_{c_j \in S} \frac{|\mathfrak{R}_i^{CB}(S) \cap \mathfrak{R}^{CB}(c_j)|}{|\mathfrak{R}^{CB}(c_j)|} \tag{4}$$

Example 2. Let $S = \{c_1, c_4\}$ be a case sample set taken from the case base in Example 1. The *related closure* of S is the set of all *related sets* that contain at least one element in S. We have

$$\mathfrak{R}^{CB}(c_1) = \bigl\{\{c_1, c_2\}, \{c_1, c_2, c_3, c_4\}\bigr\} \quad \text{and} \quad \mathfrak{R}^{CB}(c_4) = \bigl\{\{c_2, c_3, c_4\}, \{c_1, c_2, c_3, c_4\}\bigr\}$$

thus $\mathfrak{R}^{CB}(S) = \{\{c_1, c_2\}, \{c_2, c_3, c_4\}, \{c_1, c_2, c_3, c_4\}\}$. Let $\mathfrak{R}_1^{CB}(S) = \{c_1, c_2\}$, the weight of $\{c_1, c_2\}$ with regard to S is

$$w^*\bigl(\{c_1, c_2\}\bigr) = 1/2 \times \bigl(\bigl|\mathfrak{R}_1^{CB}(S) \cap \mathfrak{R}^{CB}(c_1)\bigr| / \bigl|\mathfrak{R}^{CB}(c_1)\bigr| + \bigl|\mathfrak{R}_1^{CB}(S) \cap \mathfrak{R}^{CB}(c_4)\bigr| / \bigl|\mathfrak{R}^{CB}(c_4)\bigr|\bigr) = 1/4$$

Theorem 4. *The sum of the densities of all elements in* $\mathfrak{R}^{CB}(S)$ *equals* 1.

$$\sum_{i=1}^{|\mathfrak{R}^{CB}(S)|} w^*\bigl(r_i^{CB}(S)\bigr) = 1 \tag{5}$$

Proof. Substitute Eq. (4) into Eq. (5), we have the left side as:

$$\frac{1}{|S|} \times \sum_{i=1}^{|\mathfrak{R}^{CB}(S)|} \sum_{c_j \in S} \frac{|\mathfrak{R}_i^{CB}(S) \cap \mathfrak{R}^{CB}(c_j)|}{|\mathfrak{R}^{CB}(c_j)|} = \frac{1}{|S|} \times \sum_{c_j \in S} \sum_{i=1}^{|\mathfrak{R}^{CB}(S)|} \frac{|\mathfrak{R}_i^{CB}(S) \cap \mathfrak{R}^{CB}(c_j)|}{\mathfrak{R}^{CB}(c_j)} \tag{6}$$

As $\mathfrak{R}_i^{CB}(S) \cap \mathfrak{R}_j^{CB}(S) = \phi$ $(i \neq j)$, $\mathfrak{R}^{CB}(S) = \bigcup_{i=1}^{|\mathfrak{R}^{CB}(S)|} \mathfrak{R}_i^{CB}(S)$ and according to the definition of *related closure* (Definition 4), $\mathfrak{R}^{CB}(c_j) \subseteq \mathfrak{R}^{CB}(S)$, therefore we have:

$$\sum_{i=1}^{|\mathfrak{R}^{CB}(S)|} \frac{|\mathfrak{R}_i^{CB}(S) \cap \mathfrak{R}^{CB}(c_j)|}{|\mathfrak{R}^{CB}(c_j)|} = \frac{|\mathfrak{R}^{CB}(S) \cap \mathfrak{R}^{CB}(c_j)|}{|\mathfrak{R}^{CB}(c_j)|} = 1 \tag{7}$$

Substitute Eq. (7) into Eq. (6); the left side equals the right side. □

From a practical point of view, this means that all cases in sample S are equally important with regard to the contribution to the *density* of elements in $\mathfrak{R}^{CB}(S)$, no matter what their *related sets* are.

The *density* weights each *related set* in a *related closure* by the degree to which the sample cases are distributed. We then define the *competence-based empirical weight* of a case sample with regard to a certain interesting sub-problem space (Definition 9).

Definition 9. Given a case base CB, and a case sample set $S \subseteq CB$, denote the power set of $\Re^{CB}(CB)$ as $\wp(\Re^{CB}(CB))$. Considering $\wp(\Re^{CB}(CB))$ as the measurable space $\mathcal{A}$, for $A \in \mathcal{A}$, we define the *competence-based empirical weight* of S with regard to A over CB as

$$S^{CB}(A) = \frac{\sum_{i=1,\ r_i^{CB}(S)\in A\cap\Re^{CB}(S)}^{|A\cap\Re^{CB}(S)|} w^*(r_i^{CB}(S))}{\sum_{i=1}^{|\Re^{CB}(S)|} w^*(r_i^{CB}(S))} = \sum_{\substack{i=1 \\ r_i^{CB}(S)\in A\cap\Re^{CB}(S)}}^{|A\cap\Re^{CB}(S)|} w^*\big(r_i^{CB}(S)\big) \tag{8}$$

For any case sample set S, the competence-based empirical weight provides a reference to the degree of case distribution on a competence area represented by A – a sub-set of $\Re^{CB}(CB)$. The higher the weight is, the larger is the proportion of cases in S that support the selected competence area.

Definition 10. For two case sample sets $S_1, S_2 \subseteq CB$, we define the *competence-based empirical distance* between S_1 and S_2 as

$$d^{CB}(S_1, S_2) = 2 \times \sup_{A\in\mathcal{A}} \big|S_1^{CB}(A) - S_2^{CB}(A)\big| \tag{9}$$

Proposition 6. *Given a case base of finite size CB and a case sample set $S \subseteq CB$, $\Re^{CB}(S)$ is a finite set.*

Proof. Since each case c in CB corresponds to a *related set* – $R^{CB}(c)$ (whereas several cases may correspond to the same *related set*) the size of $\Re^{CB}(CB)$ must be less, or equal to, the size of the case base. Again, according to the definition of *related closure* (Definition 7), for any case sample set $S \subseteq CB$, $\Re^{CB}(S) \subseteq \Re^{CB}(CB)$, we have $|\Re^{CB}(S)| \leqslant |\Re^{CB}(CB)| \leqslant |CB|$. As the size of the case base is finite, $\Re^{CB}(S)$ must be a finite set. □

Remark. This ensures that a solution $A \in \mathcal{A}$ exists in the defined *competence-based empirical distance*, since $\mathcal{A}$ is also a finite a set.

Theorem 5. *Given a case base of finite size CB, the competence-based empirical distance in the case sample $S \subseteq CB$, is a normalized quasi-distance function $d^{CB}: S \times S \to [0, 1]$, which satisfies the following conditions for all $S_1, S_2, S_3 \subseteq CB$.*

1. $0 \leqslant d^{CB}(S_1, S_2) \leqslant 1$
2. $d^{CB}(S_1, S_2) = 0$ *if* $S_1 = S_2$
3. $d^{CB}(S_1, S_2) = d^{CB}(S_2, S_1)$
4. $d^{CB}(S_1, S_3) \leqslant d^{CB}(S_1, S_2) + d^{CB}(S_2, S_3)$.

Proof. Condition 1, 2 and 3 are obvious. For condition 4, assume there exists $A_1 \in \mathcal{A}$ where, for any $A_i \in \mathcal{A}$, we have

$$\big|S_1^{CB}(A_1) - S_3^{CB}(A_1)\big| \geqslant \big|S_1^{CB}(A_i) - S_3^{CB}(A_i)\big| \tag{10}$$

We say $d^{CB}(S_1, S_3) = 2 \times |S_1^{CB}(A_1) - S_3^{CB}(A_1)|$ (the existence of A_1 can be proven through Proposition 6). Again, we have $A_2, A_3 \in \mathcal{A}$, that for any $A_i \in \mathcal{A}$

$$d^{CB}(S_1, S_2) = 2 \times \big|S_1^{CB}(A_2) - S_2^{CB}(A_2)\big| \geqslant 2 \times \big|S_1^{CB}(A_i) - S_2^{CB}(A_i)\big| \tag{11}$$

$$d^{CB}(S_2, S_3) = 2 \times \big|S_2^{CB}(A_3) - S_3^{CB}(A_3)\big| \geqslant 2 \times \big|S_2^{CB}(A_i) - S_3^{CB}(A_i)\big| \tag{12}$$

Clearly we have $d^{CB}(S_1, S_2) \geqslant 2 \times |S_1^{CB}(A_1) - S_2^{CB}(A_1)|$ and $d^{CB}(S_2, S_3) \geqslant 2 \times |S_2^{CB}(A_1) - S_3^{CB}(A_1)|$, so we have

$$\begin{aligned} & d^{CB}(S_1, S_2) + d^{CB}(S_2, S_3) \\ & \quad \geqslant 2 \times \big(\big|S_1^{CB}(A_1) - S_2^{CB}(A_1)\big| + \big|S_2^{CB}(A_1) - S_3^{CB}(A_1)\big|\big) \\ & \quad \geqslant 2 \times \big|S_1^{CB}(A_1) - S_3^{CB}(A_1)\big| = d^{CB}(S_1, S_3) \qquad \square \end{aligned} \tag{13}$$

Note that $S_1 = S_2$ is only a sufficient condition for $d^{CB}(S_1, S_2) = 0$, since the *competence-based empirical distance* compares the distance between two case sets through their competencies, rather than their real distribution. Any pair of case sets that exhibit identical distribution with regard to the competence will result in a distance of zero. This can be easily proven by constructing sample sets with paired cases, which are different but of the same *related set*.

We say that there is a concept drift when the *competence-based empirical distance* between the current case base and the newly available case chunk is greater than ε. Similar to Kifer, Ben-David and Gehrke's work [35], the set A depicts a local competence area, in which the largest distribution discrepancy lies between two samples, which helps to explain the detected change. The determination of ε and the statistical guarantee of the detection method will be discussed in Section 5.

5. Statistical guarantee

The choice of distance function used to determine change is only one aspect of the concept drift detection method; another is to provide statistical significance of the detected change. Given an observation of two case samples, we achieve this by answering the question “How likely is it that the observation could have been obtained under the null hypothesis H_0 (in our case, that no concept drift occurs)?”. The smaller this value (the so-called “*p-value*”), the stronger the evidence against H_0.

This work resorts to the *two-sample non-parametric permutation test* method [47] to provide a statistical guarantee for the detected change. The *permutation test* is easy to implement and free of mathematical assumptions, and is commonly used when the theoretical distribution of the test statistic is complicated or unknown, which suits our situation.

5.1. Permutation test

Recall the problem description in Section 2.1, where we have two case sets CB and CB', representing two unknown distributions F_{CB} and $F_{CB'}$, respectively. We would like to perform a hypothesis test and determine whether F_{CB} and $F_{CB'}$ are identical. In Section 4, we defined a notation of distance as *test statistic*, and for simplicity we denote the *test statistic* as $\hat{\theta}$.

Once an observation $\hat{\theta}$ is made (in our case, the calculation of the distance between the current case base and the incoming new case samples), the *achieved significance level* (*ASL*), or the *p-value* of the test, is defined as the probability of observing at least as extreme as the observed $\hat{\theta}$, assuming that the null hypothesis is true,

$$ASL = P_{H_0}\{\hat{\theta}^* \geqslant \hat{\theta}\} \tag{14}$$

where the quantity $\hat{\theta}$ is fixed at its observed value; the random variable $\hat{\theta}^*$ has the null hypothesis distribution, the distribution of $\hat{\theta}$ if H_0 is true [47].

The permutation test is a clever way of approximating an *ASL* for the null hypothesis $F_{CB} = F_{CB'}$, which works as follows: Given a case base of n cases and an incoming new case sample of m cases, under the null hypothesis, any observed case could have come equally well from either of the case sets. We therefore combine all the $m + n$ cases, then take a sample of size n without replacement to represent the case base; the remaining m cases constitute the incoming case sample. We compute the *test statistic* for each permutation and repeat the process a large number of times (N). Finally, we estimate the *ASL* of the permutation test through the Monte Carlo approach [58].

$$ASL_{perm} \approx \hat{ASL}_{perm} = \frac{\#\{\hat{\theta}^* \geqslant \hat{\theta}\}}{N} \tag{15}$$

Once we fix a desired significance level α, we compare α with the permutation *ASL*. We say that there is a concept drift when $ASL_{perm} < \alpha$. For example, when choosing α to be 0.05, we reject the null hypothesis at a 5% level, corresponding respectively to a 5% chance of rejecting the null hypothesis when it is true (*false positive*). In fact, any permutation test that relies on sampling rather than full enumeration will yield an actual significance level larger than α, due to the Monte Carlo error [59,60].

5.2. Permutation sample size

In real world applications, obtaining an exact *ASL* of a permutation test via full enumeration quickly becomes unfeasible as the permutation sample space increases. This raises the question of how many permutation replications (N) are required.

As illustrated by Efron and Tibshirani [47] and Opdyke [60], $N \times \hat{A}$ follows a binomial distribution of $Bi(N, A)$, where $\hat{A} = \hat{ASL}_{perm}$ and $A = ASL_{perm}$. Using the normal approximation to $Bi(N, A)$, the 95% *confidence interval* of $\hat{A}$ is approximated by $A \pm (1.96 \times \sigma)$, where $\sigma = [A(1 - A)/N]^{\frac{1}{2}}$ is the standard deviation of $\hat{A}$. If we do not want the Monte Carlo error to affect our estimation by more than 30% ($\sigma/A \leqslant 0.3$), that gives $N \geqslant 100$ when $A = 0.1$. The precision can be improved with larger N (Fig. 2).

6. Experimental evaluation

Our evaluation of the proposed competence-based concept drift detection method consists of three sections, through eight experiments; all source code can be downloaded from http://decide.it.uts.edu.au/Philip/index.php:

1. We evaluate the competence-based empirical distance and compare it to the test statistics of the two-sample Kolmogorov–Smirnov test (Experiments 1–3).
2. We compare our change detection method to Dasu et al. [36]. We choose synthetic datasets for the first two parts, since we need to know the change in the generated distribution in advance. In addition, with simulated data, we are able to control the change more easily and to see how our detection method performs against different types of changes (Experiments 4–7).

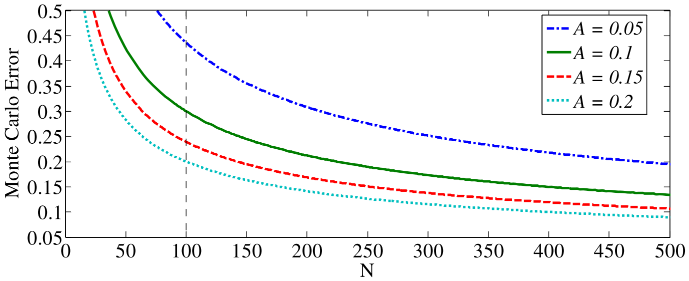


Fig. 2. Monte Carlo error vs. permutation replication size.

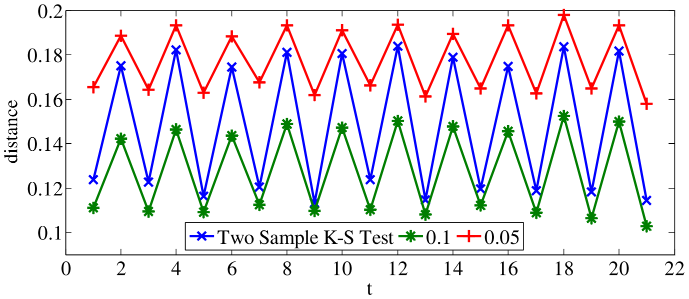


Fig. 3. Competence-based empirical distance between normal distributed data that varies μ.

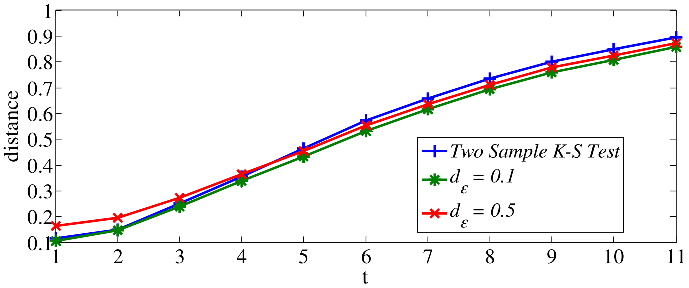


Fig. 4. Cumulated competence-based empirical distance between normal data that varies μ.

3. We perform a case-base editing method based on the results of our detection and compare our results on two real datasets against those from the original author [18], as our detection method aims not only to identify a change, but also to describe it (Experiment 8).

6.1. Evaluating the competence-based empirical distance

In Section 4, we proposed a competence-based empirical distance to measure the difference between one distribution and another. To establish how it varies according to the change between generated distributions, we ran three experiments with generated artificial datasets following 1D normal distributions and compared the results with the test statistics of the two-sample Kolmogorov–Smirnov test. Two cases are considered to be able to mutually solve each other, if their Euclidean distance is smaller than a threshold d_ε. All results are calculated as the mean of 100 independent tests.

Experiment 1 *(Varying the mean μ).* In this experiment, we compared distances between data samples, both of size 100, drawn from 11 normal distributions of fixed standard deviation $\sigma = 0.2$, but with a moving mean value $\mu = 0.2 + 0.06 \times (i - 1)$ for the ith distribution, $i = 1, 2, \ldots, 11$. When $t = 2 \times i - 1$ we compared two samples, both drawn from the ith distribution; when $t = 2 \times i$ we compared two samples drawn from the ith and $(i + 1)$st distribution. Fig. 3 shows how the competence-based empirical distance changes as the distributions vary. Since the extent of the difference depends only on the mean values, we find a similar height on all peaks for each series (Fig. 3). We also tried fixing one data sample while moving the other, in order to show how the distance increases as the difference between the means increases (Fig. 4). It is worth noting that the test statistics of the two sample K-S test seems to be more sensitive to the change, as the margin between the peaks and valleys is larger (Fig. 3), and it also increases faster as the difference accumulates (Fig. 4). This is because our competence-based empirical distance eliminates any change that happened within a problem space where

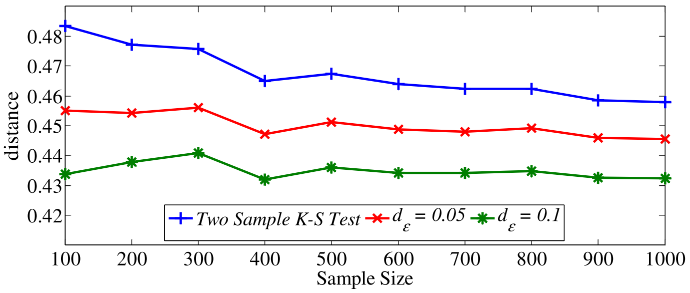


Fig. 5. Competence-based empirical distance between normal data of difference size.

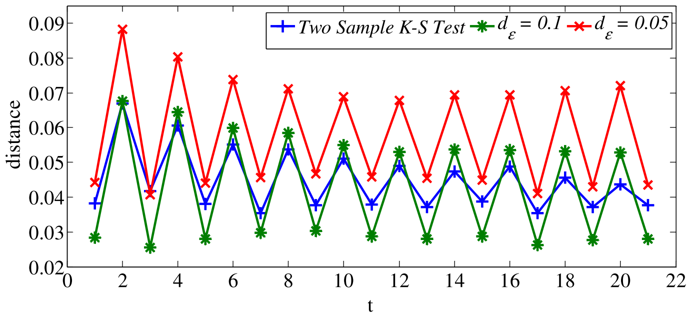


Fig. 6. Competence-based empirical distance between normal distributed data that varies σ.

cases are considered to be similar in CBR. In return, our detection method is more reliable for small samples, as shown in Experiment 2.

Experiment 2 *(Varying the sample size).* In this experiment, we compared distances between two data samples drawn from $N(0.2, 0.2)$ and $N(0.44, 0.2)$, respectively. We increased the sample size from 100 to 1000. As shown in Fig. 5, the test statistics of the two sample K-S test shrinks as the sample size increases, while our competence-based empirical distances remain relatively steady. Also, it can be seen that, when the sample size is larger than 800, the distance remains the same for all series.

Experiment 3 *(Varying σ).* In this experiment, we fixed the mean at $\mu = 0.5$, but varied the standard deviation $\sigma = 0.1 + 0.02 \times (i - 1)$ for the ith distribution, $i = 1, 2, \ldots, 11$. Again, when $t = 2 \times i - 1$ we compared two samples drawn from the ith distribution; when $t = 2 \times i$ we compared two samples drawn from the ith and $(i + 1)$st distribution. The sample size was set at 1000. As shown in Fig. 6, the competence-based empirical distances have a larger margin between peaks and valleys, which means that our method is more sensitive to smaller changes than the K-S test, but the amount of change shrinks as σ increases compared with Experiment 1. Intuitively, this is because the distribution becomes less concentrated, so the relative distance is smaller.

6.2. Evaluating the competence-based change detection method

In the above experiments, we have demonstrated how the competence-based empirical distance fluctuates as the underlying distribution changes. In order to determine whether a given measurement is statistically sufficiently significant to qualify as a concept drift, we plug in the permutation test to estimate the *ASL*, as described in Section 5. Given a desired significance level α, we say that there is a concept drift when $ASL_{perm} < \alpha$. In the following experiment, we aim to compare our change detection results with Dasu et al. [36]. There are two main reasons for us to choose their method for comparison. First, we both resort to a similar approach for concept drift detection: that is, adopting a distance metric and then performing a hypothesis test. Second, compared with other error-rate based detection methods, our methods do not depend on a classifier and, therefore, have a broader application scope.

To make a fair comparison with Dasu et al. [36], we set up the same experimental environments, which includes the following three features: strategy, data source and parameter. For strategy, we also adopt the fix-sliding windows model: that is, keeping a fixed reference window if no concept drift is reported, or else moving both windows. Also, we say that a detection is late when it is reported after moving two or more windows, since the actual window contains the change. For the data source, we implement the same artificial datasets used in Dasu et al. [36]. The detailed information of each artificial data source is described in the experiment setup section. We keep all parameters the same for the permutation test when comparing, which includes the desired significance level $\alpha = 1\%$ and the permutation size of $N = 500$. We do not

Table 1
Change detection results on different 2D normal data streams. The detection results of Dasu et al. [36] are shown in brackets.

Stream	Window size (n)	Detected	Late	False	Missed
M(0.05)	10,000	97 (97)	0 (1)	4 (4)	2 (1)
M(0.02)	10,000	86 (70)	6 (20)	5 (4)	7 (9)
C(0.1)	10,000	56 (43)	8 (18)	4 (3)	35 (38)
C(0.15)	10,000	91 (83)	2 (10)	5 (4)	6 (6)
C(0.15)	5000	84 (68)	7 (14)	11 (17)	8 (17)
C(0.2)	5000	96 (93)	1 (5)	10 (15)	2 (1)

Table 2
Change detection results on different 2D Poisson distributed data streams. The detection results of Dasu et al. [36] are shown in brackets.

Step size (Δ)	Window size (n)	Detected	Late	False	Missed
0.1	10,000	88 (67)	2 (17)	4 (1)	9 (15)
0.2	10,000	99 (98)	0 (1)	5 (5)	0 (0)

evaluate changed parameters because their effects have already been shown and proved in Section 5. For all experiments in this section, the parameter d_ε, which is used to construct the competence model, is chosen empirically to have a similar partition size to the *kdq-tree* as in Dasu et al. [36]. A more practical solution for constructing competence models in real world applications is described and used in Section 6.3.

Experiment 4 *(Normal distributions).* This experiment implements two artificial datasets used in Dasu et al. [36]: the $M(\Delta)$ stream – varying the mean μ_1 and μ_2 independently in $[0.2, 0.8]$ with a step size chosen randomly in $[-\Delta, -\Delta/2] \cup [\Delta/2, \Delta]$ and the $C(\Delta)$ stream – varying ρ, which starts at 0 and then randomly walks within $[-1, 1]$, with step size chosen randomly in $[-\Delta, -\Delta/2] \cup [\Delta/2, \Delta]$. Each stream consists of 5,000,000 two-dimensional normal distributed points, which are further divided into groups of 50,000, giving 99 changes in total. To construct the competence model, d_ε is set to 0.05.

From the results shown in Table 1, we can see that our method out-performs Dasu et al. [36]. Both methods achieve similar results for the M(0.05) and C(0.2) stream, however ours performs significantly better for the other streams, which means our method is more sensitive to smaller changes, compared to Dasu et al. [36]. As we expected, the overall detection accuracy increases when the window size increases. Our method results in a dramatic decrease in the number of late detections as the window size increases. This is probably because our proposed *competence-based empirical distance* allows different, but related cases to share their distribution contributions with regard to our competence model, rather than being cut exclusively by regions, thus improving its robustness to smaller samples. The number of false detections is almost the same as Dasu et al. [36]. In fact, the false detection rate largely depends on the desired significance level α due to the nature of the permutation test. Our detection method also depends on the appropriateness of the competence model. In real world scenarios, one possible way of constructing the competence model is to use the leave-one-out strategy (shown in Section 6.3), or resort to other available information such as reasoning history or experts.

Experiment 5 *(Poisson distributions).* In this experiment, we compare the detection results on 2D (discrete) Poisson distributions, with data streams generated according to $(X, Y) \sim Poisson(500(1-\rho), 500(1-\rho), 500\rho)$, where ρ starts at 0.5 and then performs a random walk between 0 and 1 with step size $\Delta = 0.2, 0.1$, in the same manner as in Experiment 4. We generated the bivariate Poisson using Trivariate Reduction [61]. To construct the competence model, d_ε is set to 10.

Experiment 6 *(Higher dimensions).* To test the scalability and performance of our scheme in high dimensions, we also take the C(0.2) stream and extend it to d-dimensional streams by adding dimensions in which the data distributions (also Gaussian with $\sigma = 0.2$) do not change. We keep the same standard deviation of all added dimensions as the C streams in order to maintain the marginal distribution of all dimensions, so that the overall distance between cases contributed by each dimension is equally important. In fact, since our competence model depends on the distance between cases, large marginal stable distributions will dominate calculation of the distance between cases and cause failure of our detection method. However, this issue can be easily solved by adopting a weighted distance calculation. To construct the competence model, d_ε is set to 0.15 for 4-dimensional, 0.3 for 6-dimensional and 0.5 for 10-dimensional; recall that d_ε is chosen to have a similar partition size as the kdq-tree.

The experimental results for different dimensions are listed in Table 3. As we expected, with more stationary dimensions added, it becomes harder to detect the real changes. However, our detection method preserves much of its power as the number of dimensions increase. This again is another proof that our detection method is more sensitive to smaller changes. Also, we have less late detection because of the robustness of the smaller sample of our method.

Table 3
Change detection results on d-dimensional streams. The detection results of Dasu et al. [36] are shown in brackets.

D	Window size (n)	Detected	Late	False	Missed
4	10,000	93 (89)	0 (1)	4 (7)	6 (9)
6	10,000	91 (84)	5 (10)	4(8)	3 (5)
10	10,000	75 (65)	5 (12)	5 (6)	21 (22)

Table 4
Running time with different dimensions and window sizes.

Dimension (d)	Window size (n)	Competence (s)	kdq-tree (s)
6	10,000	0.021	0.021
8	10,000	0.025	0.029
10	10,000	0.025	0.035
10	20,000	0.042	0.037
20	50,000	0.117	0.069

Experiment 7 *(Efficiency).* The nature of our detection method consists of three parts: space partition, calculating test statistics and determining the critical region. We, and Dasu et al. [36] are able to compute the test statistics directly, and we both adopt the permutation test to determine critical region. We only compare the cost of the first part, which is maintenance of the competence model in our case vs. updating the kdq-tree in Dasu's case. We use the standard approach for updating the competence model for case addition [62]. These running times were obtained from our unoptimized C# code on a laptop PC with a 2.3 GHz Intel i5 processor and 4 GB memory.

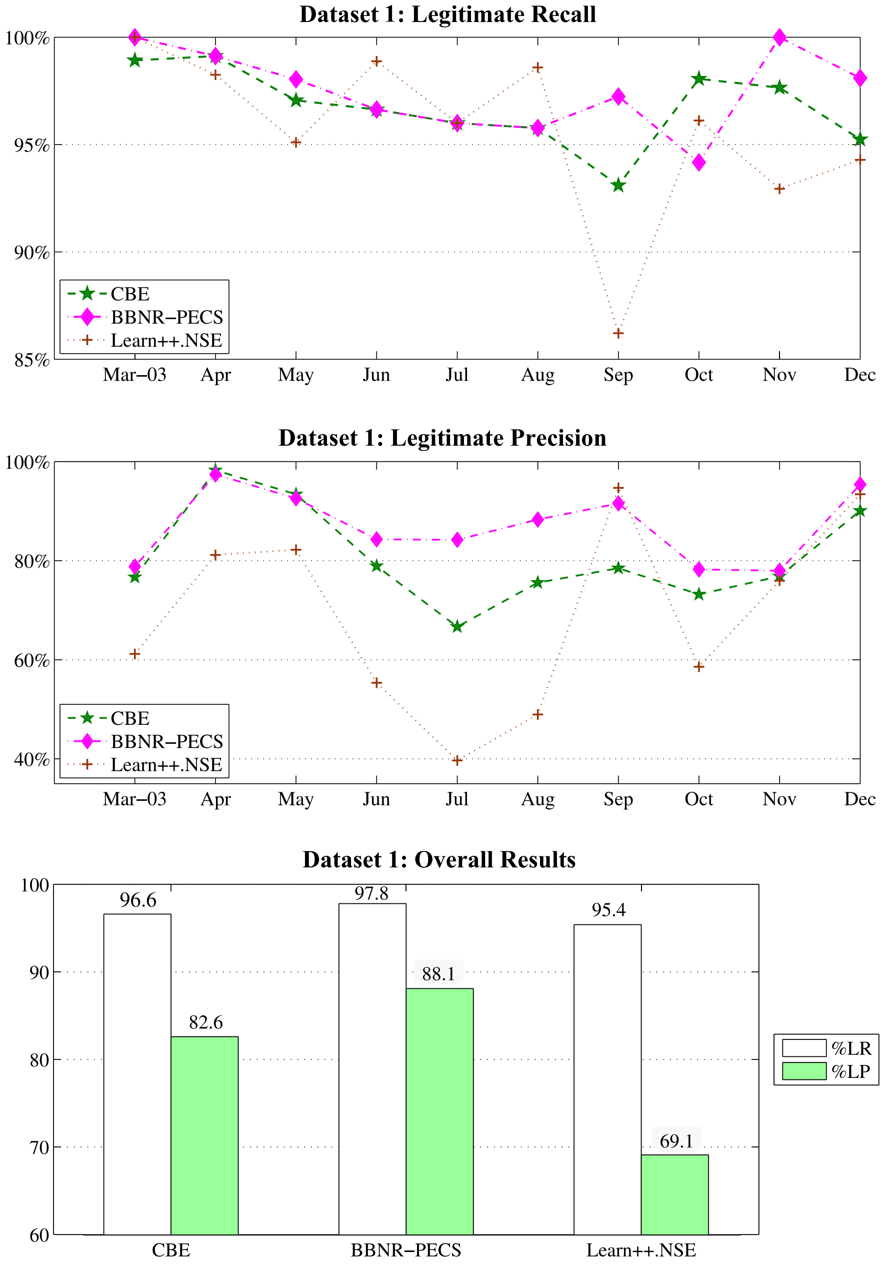
The efficiency of the competence-based concept drift detection algorithm is little affected by the number of dimensions, but directly related to the window size. While the window size increases, the time grows linearly. In fact, the maintenance cost was proven to be estimated by $O(n)$, because each time a new case is added to the current case-base, it must be compared to every case in this case-base to determine which cases can solve it and which cases it can solve [62]. To the contrary, the updating cost of the kdq-tree is little affected by the window size, but exhibits a linear relationship with the dimension. This is because the updating cost of the kdq-tree has been proven to be $O(d \log \frac{1}{\delta})$ [36].

6.3. Evaluating the description of the competence-based change detection method

In order to show how our method will help in real world scenarios, we apply our detection method to two real world concept drift datasets that are available from http://decide.it.uts.edu.au/Philip/index.php. Each dataset consists of more than 10,000 emails collected over a period of approximately one year by an individual. A training set of 1000 cases, including the last 500 spam emails and 500 legitimate emails, is set up for each dataset. The remaining data is used to test our algorithm over time. We perform real-time change detection for every incoming new email, and trigger maintenance operations in the competence areas that are identified to be undergoing changes. For more detailed information of the concept drift datasets, please refer to Section 5 of Delany et al. [18].

Experiment 8. We compare our results with the original author of this dataset [18] and a recent work on concept drift [24]. In order to compare fairly with Delany et al. [18], we choose the same classifier, that is the k-nearest neighbor with $k = 3$ using unanimous voting. Again, we also weight all features equally and the similarity between two cases is measured by the proportion of matched features. The *related set* is constructed using leave-one-out classification. That is, any case c_i is considered to be solved by the actually retrieved cases that successfully solve c_i bounded by the closest case c_j that fails to solve to c_i. For Learn++.NSE [24], we build a new case-base every month using the same criteria described above as the base classifier, and choose the same parameter values that are suggested in their paper, i.e., $a = 0.5$, $b = 10$. We are aware that it may not be fair to choose the same parameter values for a different dataset; however, a method for determining these parameters was not provided and these values are reported to work well on all the scenarios attempted.

Our spam filtering algorithm starts with the provided training set. After each classification, we perform the competence-based change detection algorithm. A maintenance operation is triggered when a concept drift is detected; therefore, the system can adapt quickly to the new concept and tune for future detections. Our tuning mechanism is a hybrid of PECS [23] and BBNR [19], which is specifically adopted to enhance PECS. On one hand, when there is no concept drift, BBNR prevents a noisy case from inclusion; on the other hand, when there is indeed concept drift, from all the areas that are reported to be changing, which in fact are measured by the set A (Definition 10), we identify and remove cases that conflict with the new case. The idea behind this tuning strategy is that, when there is no concept drift in the data stream, old cases can help to identify noise and improve accuracy; however, when there is indeed concept drift, new instances are more representative of the novel concept, while old cases that conflict may not help.


Dataset 1: Legitimate Recall

Dataset 1: Legitimate Precision

Dataset 1: Overall Results

Fig. 7. Results of concept drift dataset 1.

As the CBE method described in Delany et al. [18] considers both competence enhancement and competence preservation, we also plugin a redundancy removal operation, which removes cases uniformly in the competence space and ensures all removed cases can still be correctly solved. We keep 1500 cases for dataset 1 and 2500 cases for dataset 2. This limit is determined according to Delany et al. [18], which reports that 'the resulting size of the case-base after all the data has been applied (i.e. after 10 months for dataset 1 and 12 months for dataset 2) is 1512 and 2518, respectively'. We do not incorporate any redundancy removal operation for Learn++.NSE, because: 1) we intend to retain their original methods; 2) each base classifier in Learn++.NSE is relatively small, which meets the speed and storage requirements.

Since error rate is not a good metric for skewed datasets, the most common performance metrics [63] for spam filtering are used here to evaluate performance, where the Legitimate Recall (LR) is defined as $LR = \frac{TN}{TN+FP} = \frac{TN}{N}$, and the Legitimate Precision (LP) is defined as $LP = \frac{TN}{TN+FN}$, where FP means legitimate emails that are incorrectly classified as spam; FN means spam emails that are incorrectly classified as legitimate.

The classification results of each month for both datasets are shown, respectively, in Fig. 7 and Fig. 8.

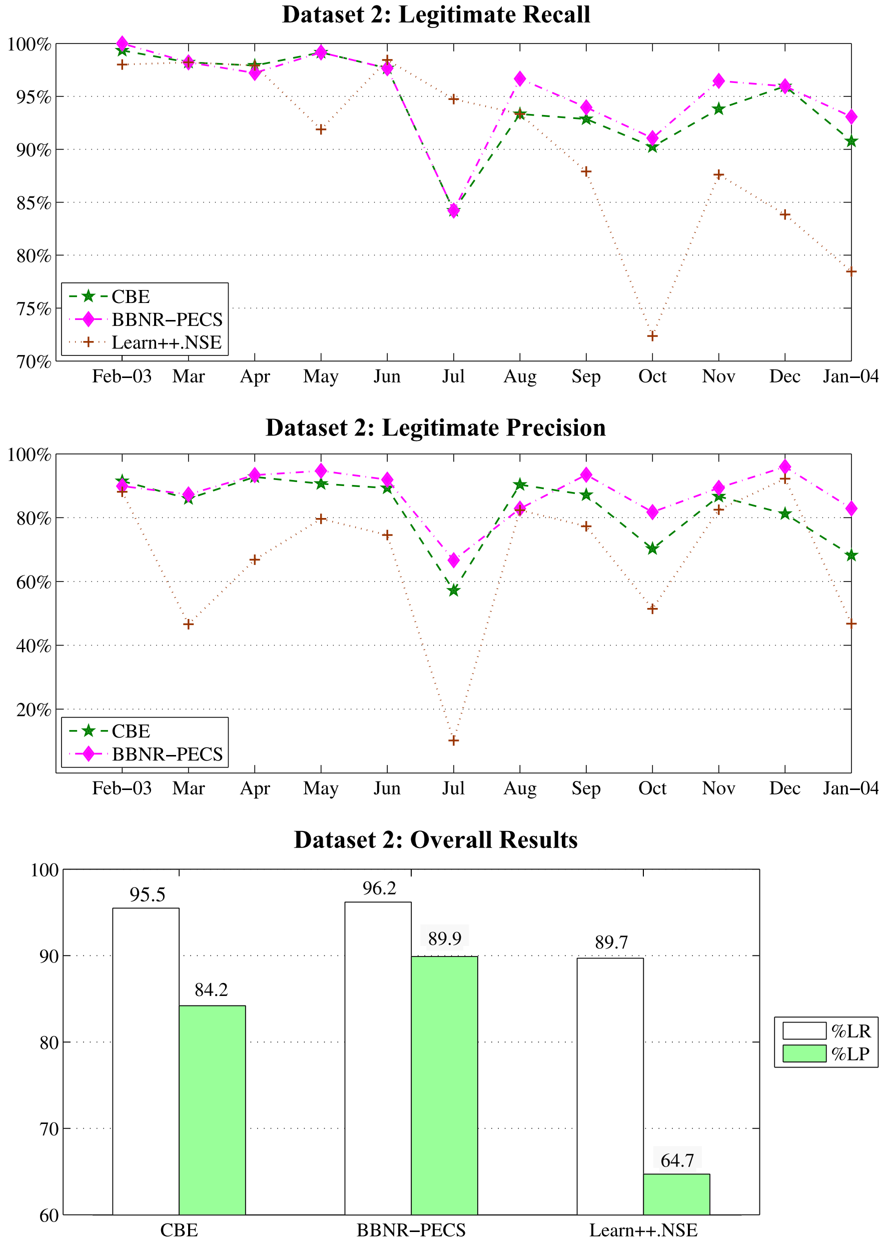


Fig. 8. Results of concept drift dataset 2.

These results clearly demonstrate that our strategy effectively improved the results on both LR and LP for both datasets. Our approach mainly improves LP. This is because we are using k-NN with unanimous voting. As a result, mistakenly retaining one or two noisy spam emails in the same competence area will not affect the classification result for a legitimate email; however, mistakenly retaining a legitimate email will dramatically affect the classification result for a spam email. Compared with CBE, the improvement on LP is statistically significant for both datasets, by paired t-test, at 94% confidence level. This shows that our method can accurately catch the concept drift and perform well on targeted reactions. We also observed that Learn++.NSE was the worst on most months, which was principally for the following two reasons: 1. The ensemble approach adopted in Learn++.NSE performed a postponed updating schema, i.e., a new base classifier was trained monthly. As a result, the system could not take advantage of the up-to-date feedback and make a timely adaptation. It would appear that an online learning schema is more suitable for the spam filtering domain where the feature space is large and sparse, and concept drift changes rapidly. 2. The weight of each base classifier is calculated by a log operator which may generate an unlimited figure. As a result, a single base classifier can dominate the final prediction result. In our experiments at time t, the base classifier h^t trained with the latest dataset D^t always achieved the best performance on D^t. Note that the weight of each base classifier is assigned based on its behavior on D^t; however, when concept drifts, the incoming dataset D^{t+1} may follow a different distribution of D^t. Unfortunately, Learn++. NSE does not incorporate any drift detection mechanism to cope with this situation.

7. Conclusions and further studies

The competence group model is inadequate and deficient. We have consequently proposed and defined a competence closure model. This study conducts a theoretical study on the competence group model, and finds that it does not guarantee a complete splitting of the case-base into several competence groups (inadequacy), nor does it guarantee that any two competence groups are mutually independent (deficiency). This research defines a competence closure model to overcome these problems and reveals the essential differences between these two models. It also identifies three important aspects of the competence closure model which the competence group model does not possess: 1) the competence closure model is able to uniquely model the entire case-base; 2) the competence closure model provides a smaller granularity than the competence group model; 3) competence closure is the maximum set concerning a series of related problems and two competence closures are said to be independent in terms of the related set.

Concept drift can also reflect on competence measurement. We have presented a method to detect concept drift via competence models which requires no prior knowledge of distribution but measures it through a competence model. The competence-based change detection method can be applied to CBR systems, where new cases are available sequentially over time. Empirical experiments report three advantages of our proposed competence-based change detection method: 1) a high achieved detection rate, 2) robustness on small sample size, and 3) ability to quantify and describe the changes it detects, which makes it highly suitable for handling local concept drift problems.

We have found that the *competence-based empirical weight* provides a rough estimation of the competence distribution of the cases. Our next attempt will aim to provide a more reliable competence distribution through fuzzy probability theory. Another improvement may be achieved by drawing a sample that contains no duplicates in the permutation test, as proven by Opdyke [60]. Finally, this paper is part of our work in handling concept drift problems in CBR. Successive case base editing methods and metric learning methods that take advantage of change detection are needed to improve the final learning performance.

Acknowledgements

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Project DP110103733. We also wish to thank the anonymous reviewers for their helpful comments.

References

[1] J. Gao, W. Fan, J. Han, On appropriate assumptions to mine data streams: analysis and practice, in: Proceedings of the Seventh IEEE International Conference on Data Mining, Omaha, October 28–31, 2007, pp. 143–152.

[2] P. Zhang, X. Zhu, Y. Shi, Categorizing and mining concept drifting data streams, in: Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24–27, 2008, pp. 812–820.

[3] Z. Ouyang, M. Zhou, T. Wang, Q. Wu, Mining concept-drifting and noisy data streams using ensemble classifiers, in: Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence (AICI), China, November 7–8, 2009, pp. 360–364.

[4] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Mach. Learn. 23 (1996) 69–101.

[5] G. Hulten, L. Spencer, P. Domingos, Mining time-changing data streams, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, August 26–29, 2001, pp. 97–106.

[6] L. Cohen, G. Avrahami, M. Last, A. Kandel, Info-fuzzy algorithms for mining dynamic data streams, Appl. Soft Comput. 8 (2008) 1283–1294.

[7] C.-P. Wei, I.T. Chiu, Turning telecommunications call details to churn prediction: a data mining approach, Expert Syst. Appl. 23 (2002) 103–112.

[8] G. Widmer, M. Kubat, Effective learning in dynamic environments by explicit context tracking, in: Proceedings of the European Conference on Machine Learning (ECML-93), Vienna, Austria, April 5–7, 1993, in: Lecture Notes on Artificial Intelligence, Springer, 1993, pp. 227–243.

[9] K.O. Stanley, Learning concept drift with a committee of decision trees, 2003, UT-AI-TR-03-302.

[10] G. Forman, Tackling concept drift by temporal inductive transfer, in: Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6–11, 2006, pp. 252–259.

[11] A. Tsymbal, M. Pechenizkiy, P. Cunningham, S. Puuronen, Dynamic integration of classifiers for handling concept drift, Inf. Fusion 9 (2008) 56–68.

[12] M.A. Maloof, R.S. Michalski, Incremental learning with partial instance memory, Artif. Intell. 154 (2004) 95–126.

[13] W.N. Street, Y. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, August 26–29, 2001, pp. 377–382.

[14] H. Wang, W. Fan, P.S. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, D.C., August 24–27, 2003, pp. 226–235.

[15] M. Last, Online classification of nonstationary data streams, Intell. Data Anal. 6 (2002) 129–147.

[16] W.-F. Hsiao, T.-M. Chang, An incremental cluster-based approach to spam filtering, Expert Syst. Appl. 34 (2008) 1599–1608.

[17] R. Klinkenberg, Learning drifting concepts: Example selection vs. example weighting, Intell. Data Anal. 8 (2004) 281–300.

[18] S.J. Delany, P. Cunningham, A. Tsymbal, L. Coyle, A case-based technique for tracking concept drift in spam filtering, Knowl.-Based Syst. 18 (2005) 187–195.

[19] S.J. Delany, P. Cunningham, An analysis of case-base editing in a spam filtering system, in: Proceedings of the Seventh European Conference in Case-Based Reasoning, Madrid, Spain, August/September, 2004, pp. 128–141.

[20] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1991) 37–66.

[21] P. Cunningham, N. Nowlan, S.J. Delany, M. Haahr, A case-based approach to spam filtering that can track concept drift, in: Proceedings of the ICCBR'03 Workshop on Long-Lived CBR Systems, Trondheim, Norway, June 2003.

[22] A. Tsymbal, The problem of concept drift: Definitions and related work, Department of Computer Science, Trinity College Dublin, Ireland, April 2004, TCD-CS-2004-15.

[23] M. Salganicoff, Tolerating concept and sampling shift in lazy learning using prediction error context switching, Artif. Intell. Rev. 11 (1997) 133–155.

[24] R. Elwell, R. Polikar, Incremental learning of concept drift in nonstationary environments, IEEE Trans. Neural Netw. 22 (2011) 1517–1531.

[25] M.M. Richter, Introduction, in: M. Lenz, et al. (Eds.), Case-Based Reasoning Technology: From Foundations to Applications, Springer, Heidelberg, 1998, pp. 1–15.

[26] B. Smyth, M.T. Keane, Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, San Francisco, August 20–25, Morgan Kaufmann, 1995, pp. 377–382.

[27] J. Zhu, Q. Yang, Remembering to add: Competence-preserving case-addition policies for case-base maintenance, in: Proceedings of the Sixteenth International Joint Conference in Artificial Intelligence (IJCAI), Stockholm, Sweden, July 31–August 6, 1999, pp. 234–241.

[28] B. Smyth, E. McKenna, Competence models and the maintenance problem, Comput. Intell. 17 (2001) 235–249.

[29] S. Massie, S. Craw, N. Wiratunga, Complexity profiling for informed case-base editing, in: Proceedings of the Eighth European Conference (ECCBR), Fethiye, Turkey, September 4–7, 2006, pp. 325–339.

[30] S. Craw, S. Massie, N. Wiratunga, Informed case base maintenance: a complexity profiling approach, in: Proceedings of the Twenty-Second National Conference on Artificial Intelligence, vol. 2, Vancouver, British Columbia, Canada, July 22–26, 2007, pp. 1618–1621.

[31] R. Pan, Q. Yang, S.J. Pan, Mining competent case bases for case-based reasoning, Artif. Intell. 171 (2007) 1039–1068.

[32] D.C. Wilson, D.B. Leake, Maintaining case-based reasoners: Dimensions and directions, Comput. Intell. 17 (2001) 196–213.

[33] N. Lu, G. Zhang, J. Lu, Detecting Change via Competence Model, in: Proceedings of the Eighteenth International Conference on Case-Based Reasoning Research and Development, Alessandria, Italy, July 19–22, 2010, pp. 201–212.

[34] I. Žliobaitė, Adaptive training set formation, PhD thesis, Vilnius University, Vilnius, 2010.

[35] D. Kifer, S. Ben-David, J. Gehrke, Detecting change in data streams, in: Proceedings of the Thirtieth International Conference on Very Large Databases, vol. 30, Toronto, Canada, August 31–September 3, 2004, pp. 180–191.

[36] T. Dasu, S. Krishnan, S. Venkatasubramanian, K. Yi, An information-theoretic approach to detecting changes in multi-dimensional data streams, in: Proceedings of the Thirty-Eighth Symposium on the Interface of Statistics, Computing Science and Applications (Interface '06), Pasadena, CA, May 24–27, 2006, pp. 1–24.

[37] J.P. Patist, Optimal window change detection, in: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops (ICDMW '07), Omaha, NE, USA, Oct 28–31, 2007, pp. 557–562.

[38] A. Bifet, R. Gavaldà, Learning from time-changing data with adaptive windowing, in: Proceedings of the Seventh SIAM International Conference on Data Mining (SDM'07), Minneapolis, MN, USA, Apr. 26–28, 2007, pp. 443–448.

[39] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: Proceedings of the Seventeenth Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29–October 1, 2004, pp. 286–295.

[40] M. Baena-García, J. Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, R. Morales-Bueno, Early drift detection method, in: ECML/PKDD 2006, Workshop on Knowledge Discovery from Data Streams, Berlin, Germany, Sep. 18, 2006, pp. 77–86.

[41] Y. Yasumura, N. Kitani, K. Uehara, Quick adaptation to changing concepts by sensitive detection, in: Proceedings of the Twentieth International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems, Kyoto, June 26–29, 2007, pp. 855–864.

[42] P. Li, X. Hu, Q. Liang, Y. Gao, Concept drifting detection on noisy streaming data in random ensemble decision trees, in: Proceedings of the Sixth International Conference on Machine Learning and Data Mining in Pattern Recognition, Leipzig, July 23–25, 2009, pp. 236–250.

[43] B. Su, Y.D. Shen, W. Xu, Modeling concept drift from the perspective of classifiers, in: IEEE Conference on Cybernetics and Intelligent Systems (CIS '08), Chengdu, China, Sep. 21–24, 2008, pp. 1055–1060.

[44] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (1945) 80–83.

[45] A. Kolmogoroff, Confidence limits for an unknown distribution function, Ann. Math. Stat. 12 (1941) 461–463.

[46] N.V. Smirnov, Approximate laws of distribution of random variables from empirical data, Usp. Mat. Nauk 10 (1944) 179–206.

[47] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, New York, 1993.

[48] K. Nishida, K. Yamauchi, Detecting concept drift using statistical testing, in: Proceedings of the Tenth International Conference on Discovery Science (DS '07), Sendai, Japan, Oct. 1–4, 2007, pp. 264–269.

[49] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (2007) 119–139.

[50] G. Welch, G. Bishop, An introduction to the Kalman filter, Technical Report TR-95-041, University of North Carolina, 1995.

[51] C. Englund, A. Verikas, A hybrid approach to outlier detection in the offset lithographic printing process, Eng. Appl. Artif. Intell. 18 (2005) 759–768.

[52] F. Angiulli, R.B.-E. Zohary, L. Palopoli, Outlier detection using default reasoning, Artif. Intell. 172 (2008) 1837–1872.

[53] A. Dries, U. Rückert, Adaptive concept drift detection, Stat. Anal. Data Min. 2 (2009) 311–327.

[54] S. Massie, S. Craw, N. Wiratunga, What is CBR competence? BCS-SGAI Expert Update 8 (2005) 7–10.

[55] B. Smyth, E. McKenna, Modelling the competence of case-bases, in: Proceedings of the Fourth European Workshop on Advances in Case-Based Reasoning, Dublin, Ireland, September 1998, pp. 208–220.

[56] E. McKenna, B. Smyth, Competence-guided case discovery, in: Proceedings of the Twenty-First SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence, Cambridge UK, December 10–12, 2001, pp. 97–108.

[57] H. Brighton, C. Mellish, Advances in instance selection for instance-based learning algorithms, Data Min. Knowl. Discov. 6 (2002) 153–172.

[58] E. Koehler, E. Brown, S.J.-P.A. Haneuse, On the assessment of Monte Carlo error in simulation-based statistical analyses, Am. Stat. 63 (2009) 155–162.

[59] K.J. Berry, P.W. Mielke, Moment approximations as an alternative to the F test in analysis of variance, Br. J. Math. Stat. Psychol. 36 (1983) 202–206.

[60] J.D. Opdyke, Fast permutation tests that maximize power under conventional Monte Carlo sampling for pairwise and multiple comparisons, J. Mod. Appl. Stat. Methods 2 (2004) 27–49.

[61] K.V. Mardia, Families of Bivariate Distributions, Griffin, London, 1970.

[62] B. Smyth, E. McKenna, An efficient and effective procedure for updating a competence model for case-based reasoners, in: Proceedings of Eleventh European Conference on Machine Learning (ECML '00), Barcelona, Catalonia, Spain, May 31–June 2, 2000, pp. 357–368.

[63] K.R. Gee, Using latent semantic indexing to filter spam, in: Proceedings of Eighteenth Annual ACM Symposium on Applied Computing (SAC '03), Melbourne, FL, USA, Mar. 9–12, 2003, pp. 460–464.