Detecting Change via Competence Model

Ning Lu, Guangquan Zhang and 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
P.O. Box 123, Broadway, NSW 2007, Australia
{philiplu, jielu, zhangg}@it.uts.edu.au

Abstract. In real world applications, interested concepts are more likely to change rather than remain stable, which is known as *concept drift*. This situation causes problems on predictions for many learning algorithms including case-base reasoning (CBR). When learning under concept drift, a critical issue is to identify and determine "when" and "how" the concept changes. In this paper, we developed a *competence-based empirical distance* between case chunks and then proposed a change detection method based on it. As a main contribution of our work, the change detection method provides an approach to measure the distribution change of cases of an infinite domain through finite samples and requires no prior knowledge about the case distribution, which makes it more practical in real world applications. Also, different from many other change detection methods, we not only detect the change of concepts but also quantify and describe this change.

Keywords: Case-based Reasoning, Competence Model, Concept Drift.

1 Introduction

In recent years, with the rapid development of information, modern organizations are accumulating data at unprecedented rates. Examples of such data streams include customer purchase logs, telephone calling records, credit card transactional flows. While these data may contain valuable knowledge, the distribution or pattern underlying the data is more likely to change over time rather than remain stable, which is also known as concept drift [1, 2]. As a result, when a certain learning algorithm considers all the past training data or makes assumption that the training data is a random sample drawn from a stationary distribution, the induced pattern may not relevant to the new data. In practical terms, this means an increasing error in classifying new records with existing models [3, 4].

Generally there are three approaches for handling concept drift: 1) instance selection (window-based); 2) instance weighting (weight-based); 3) ensemble learning [5, 6]. In instance selection, the key idea is to select the most relevant instances to the current concept. The typical technique of this category is to pick up the training dataset within a fixed or dynamic window that moves over recently arrived instances to construct a model [2, 3, 7]. Many case-base editing strategies in case-based reasoning (CBR) that delete noisy, irrelevant and redundant cases are also a form of instance selection [8]. In Instance weighting, each instance is assigned a weight to represent the decreasing relevance of existing training examples. And learning algorithms are adopted to process these weighted instances, such as Support Vector Machines (SVMs) [9]. Instances can be weighted according to their age, and their competence with regard to the current concept [5]. Ensemble learning deals with concept drift by utilizing multiple models and by voting or selecting the most relevant one to construct a proper predictive model [10-12]. Generally, there are two ensemble frameworks: 1) horizontal ensemble, which builds on a set of buffered data chunks; 2) vertical ensemble, which builds on the most recent data chunk only. More recently, an aggregate ensemble framework, which could been seen as a hybrid approach of the two, has been proposed [13].

All these proposed methods reported great improvement for learning under concept drift. However, most of current solutions implicitly assume that concept drift is ubiquitous and global. This causes problem when change in the concept or data distribution occur in some regions of instance space only, which is known as local concept drift [14]. So instead of directly assigning a weight to each classifier or chunk of training set, Tsymbal, Pechenizkiy, Cunningham and Puuronen [14] gave a weighted strategy from instance level, which estimated the local performance of each base classifier for each instance of the coming instance set. However, their method is not able to determine whether there is a concept drift happened. On one hand, when concept remains, clearly old training examples can help to achieve a more robust model. But on the other hand, when concept drift occurs, old training data do not always help produce a more accurate hypothesis than using the most recent data only [15]. As a result, further information about when and where the change has occurred is needed, so that a learner can distinguish whether there is a concept drift and make better use of existing training data. Addressing to this issue, we propose a new change detection method for CBR system, which compares the distribution of existing case base and newly available cases. Our method not only decides whether concept drift occurs, but also provides a meaningful explanation about where and how the underlying distribution change is.

The remaining of this paper is organized as follows. Section 2 reviews the related works concerning change detection for data stream and a competence model for CBR. In Section 3, a competence-based empirical distance between case chunks is introduced with a simple example. Then we present our change detection method in more details. The results of experimental evaluation are shown in Section 4. Finally, conclusions and future works come in Section 5.

2 Related Work

In this section, we first introduce a change detection method for data streams. Following that, a competence model for CBR systems will be discussed.

2.1 A Change Detection Method

A natural approach of detecting concept drift is to compare the distribution of the data. However, in real world applications, the data that one typically encounters may not arise from any standard distribution, which makes non-parametric tests more practical. Moreover, the data may contain several dimensions. As a result, traditional non-parametric tests like the Wilcoxon and Kolmogorov-Smirnov cannot be easily adopted. Kifer, Ben-David and Gehrke [16] proposed a change detection method by employing a notation of distance which could be seen a generalization of Kolmogorov-Smirnov statistic (Def. 1). Two probability distributions are considered as ε-close if their distance is no greater than ε.

Definition 1. [16] Fix a measure space and let $\mathcal{A}$ be a collection of measurable sets. Let P and P' be probability distributions over this space.

- The *$\mathcal{A}$-distance* between P and P' is defined as

$$\mathrm{d}_{\mathcal{A}}\,(\mathrm{P},\mathrm{P}') = 2\sup_{\mathrm{A}\in\mathcal{A}}|\mathrm{P}(\mathrm{A}) - \mathrm{P}'(\mathrm{A})| \qquad (\mathbf{1})$$

- For a finite domain subset S and a set $\mathrm{A} \in \mathcal{A}$, let the *empirical weight* of A with regard to (*w.r.t.*) S be

$$\mathrm{S}(\mathrm{A}) = \frac{|\mathrm{S} \cap \mathrm{A}|}{|\mathrm{S}|} \qquad (\mathbf{2})$$

- For finite domain subsets, S_1 and S_2, the *empirical distance* is defined as

$$\mathrm{d}_{\mathcal{A}}\,(\mathrm{S}_1,\mathrm{S}_2) = 2\sup_{\mathrm{A}\in\mathcal{A}}|\mathrm{S}_1(\mathrm{A}) - \mathrm{S}_2(\mathrm{A})| \qquad (\mathbf{3})$$

They also provided a variation of notion of the $\mathcal{A}$-distance, called *relativized discrepancy*, which takes the relative magnitude of a change into account. But for this work, we only show how our method works with the *$\mathcal{A}$-distance* in a CBR system and leave the discussion of *relativized discrepancy* for future work. Interested readers please refer to the original work [16] for the details.

Although there exist many other change detection methods [17-20], there is a reported advantage for us to choose Kifer, Ben-David and Gehrke's [16] method. That is being able to quantify and describe the change it detects, which makes it more appropriate for handling local concept drift.

2.2 A Competence Model

Competence is a measurement of how well a CBR system fulfils it goals. As CBR is a problem-solving methodology, competence is usually taken to be the proportion of problems faced that it can solve successfully [21]. According to Smyth and Kenna

[22], the local competence of an individual case is characterized by its coverage and reachability. The coverage of a case is the set of target problems that it can be used to solve. The reachability of a target problem is the set of cases that can be used to provide a solution for the target. Since it is impossible to enumerate all possible future target problems, in practice Smyth and Kenna [22] estimated the *coverage set* of a case by the set of cases that can be solved by its retrieval and adaption. And the *reachability set* of a case is estimated by the set of cases that can bring about its solution. Smyth and McKenna [23] extended this competence model. They defined the *related set* of a case as the union of its *coverage set* and *reachability set*, and said the *shared coverage* of two cases exists if and only if the intersection of the *related sets* of two different cases is not empty. Definition 2 gives a overall view of this competence model based on a survey provided by Smyth and McKenna [24].

Definition 2. [24] For a case base $\mathbb{C} = \{\mathrm{c}_1, \mathrm{c}_2, \cdots, \mathrm{c}_\mathrm{n}\}$, given a case $\mathrm{c} \in \mathbb{C}$

$$\mathrm{CoverageSet(c)} = \{\mathrm{c}' \in \mathbb{C}\colon \mathrm{Solves(c, c')}\} \tag{4}$$

$$\mathrm{ReachibilitySet(c)} = \{\mathrm{c}' \in \mathbb{C}\colon \mathrm{Solves(c', c)}\} \tag{5}$$

$$\mathrm{RelatedSet(c)} = \mathrm{CoverageSet(c)} \cup \mathrm{ReachabilitySet(c)} \tag{6}$$

Further, based on Smyth and McKenna’s competence model [24], we defined a *competence closure* as the maximal set of cases linked together though their *related set* in our previous research (Def. 3).

Definition 3. [25] For $\mathrm{G} = \{\mathrm{c}_1 \cdots \mathrm{c}_\mathrm{m}\} \subseteq \mathbb{C}$,

$$\begin{gathered}\mathrm{CompetenceClosure(G), iff}\ \forall \mathrm{c}_\mathrm{i}, \mathrm{c}_\mathrm{j} \in \mathrm{G}, \mathrm{if}\ \mathrm{c}_\mathrm{i} \neq \mathrm{c}_\mathrm{j}, \exists\left\{\mathrm{c}_{\mathrm{i}_1}, \mathrm{c}_{\mathrm{i}_2}, \cdots, \mathrm{c}_{\mathrm{i}_\mathrm{k}}\right\} \subseteq \mathrm{G}, \\ \mathrm{st.\,SharedCoverage}\left(\mathrm{c}_{\mathrm{i}_\mathrm{p}}, \mathrm{c}_{\mathrm{i}_{\mathrm{p}+1}}\right) \neq \emptyset\ (\mathrm{p} = 0, \cdots, \mathrm{k}) \\ \mathrm{where}\ \mathrm{c}_\mathrm{i} = \mathrm{c}_{\mathrm{i}_0}, \mathrm{c}_\mathrm{j} = \mathrm{c}_{\mathrm{i}_{\mathrm{k}+1}}\ \wedge \\ \forall \mathrm{c}_\mathrm{k} \in \mathbb{C} - \mathrm{G}, \nexists \mathrm{c}_\mathrm{l} \in \mathrm{G}, \mathrm{st.\,SharedCoverage}(\mathrm{c}_\mathrm{k}, \mathrm{c}_\mathrm{l}) \neq \emptyset\end{gathered} \tag{7}$$

3 Competence-based Change Detection Method

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 considerable close distributions [26]. In CBR, this means the newly available cases represent the concept that we may interested in the future. Obviously, cases in existing case base and the newly available cases could be considered as two samples drawn from two probability distributions. Thus by detecting possible distribution change between existing case base and newly available case chunk, we are able to identify whether there is a concept drift. However, there are two difficulties that prevent us from applying Kifer, Ben-David & Gehrke’s detecting algorithm [16] directly. First, we have no prior knowledge about the probability distributions of either the existing case base or the new case chunk. Second, the cases may come from an infinite domain. As a result, we cannot estimate the distance through the cases directly.

As the competence measures the problem solving capabilities of a CBR system, the probability distribution change of its cases should also reflects upon its competence. This inspired our research of detecting change via competence model. The key idea is to measure the distribution change of cases with regarding to their competence instead of their real distribution. This section will illustrate how to detect change via competence model for CBR systems.

3.1 Competence-based Empirical Distance

Similar as Smyth and McKenna's work [23], we refer the *related set* of a case to represent a local area of target problems. A visible benefit of adopting their competence model is that it transfers the infinite case domain into a finite domain of *related sets*. This solves our difficulties of measuring the statistic distance between two case samples.

Definition 4. Given a case $c \in \mathbb{C}$, denote the *related set* of c with regard to $\mathbb{C}$ as $R^{\mathbb{C}}(c)$

- We define the *related closure* of c *w.r.t.* $\mathbb{C}$ as

$$\mathcal{R}^{\mathbb{C}}(c) = \{R^{\mathbb{C}}(c_i): \forall c_i \in \mathbb{C}, \exists R^{\mathbb{C}}(c_i)\ \mathrm{st.}\, c \in R^{\mathbb{C}}(c_i)\} \tag{8}$$

- For a case sample set $\mathbb{S} \subseteq \mathbb{C}$, we define the *related closure* of $\mathbb{S}$ *w.r.t.* $\mathbb{C}$ as

$$\mathcal{R}^{\mathbb{C}}(\mathbb{S}) = \bigcup_{c \in \mathbb{S}} \mathcal{R}^{\mathbb{C}}(c) \tag{9}$$

To be more clear, $\mathcal{R}^{\mathbb{C}}(c)$ is the set of all *related sets,* with regard to $\mathbb{C}$, which contain the case c. Since the *related set* measures the local competence of a case, the intuitive meaning of the *related closure* is the maximum set of local competence that a case or a group of cases could stand for.

Theorem 1. For a case base of finite size $\mathbb{C}$, and a case sample set $\mathbb{S} \subseteq \mathbb{C}$, $\mathcal{R}^{\mathbb{C}}(\mathbb{S})$ is a finite set and we have:

$$|\mathcal{R}^{\mathbb{C}}(\mathbb{S})| \leq |\mathcal{R}^{\mathbb{C}}(\mathbb{C})| \leq |\mathbb{C}| \tag{10}$$

Since each case in $\mathbb{C}$ corresponding to a *related set,* the proof of Theorem 1 is obvious. Therefore, over a case base of finite size $\mathbb{C}$, for two case samples of $\mathbb{S}_1$, $\mathbb{S}_2 \subseteq \mathbb{C}$, we obtain two finite *related closures,* $\mathcal{R}^{\mathbb{C}}(\mathbb{S}_1)$ and $\mathcal{R}^{\mathbb{C}}(\mathbb{S}_2)$. Intuitively we could measure distance between $\mathbb{S}_1$ and $\mathbb{S}_2$ as the *empirical distance* between

$\mathcal{R}^{\mathbb{C}}(\mathbb{S}_1)$ and $\mathcal{R}^{\mathbb{C}}(\mathbb{S}_2)$. However, it will only represent the distance between the competences covered by these two samples. The relative distribution discrepancy within the competence is missing. This introduces problem when we are comparing two samples of similar *related closures*, but with dramatic different distribution. To address this problem, we assign a weight for each element in $\mathcal{R}^{\mathbb{C}}(\mathbb{S}_1)$ and $\mathcal{R}^{\mathbb{C}}(\mathbb{S}_2)$ to represent the relative density of the cases distributed over their *related closures*.

Definition 5. Denote the i^th^ element in $\mathcal{R}^{\mathbb{C}}(\mathbb{S})$ as $\mathrm{r}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S})$, let $\mathcal{R}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S}) = \{\mathrm{r}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S})\}$, we defined the *density* of $\mathrm{r}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S})$ *w.r.t* $\mathbb{S}$ be

$$\mathrm{w}^*\left(\mathrm{r}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S})\right) = \frac{1}{|\mathbb{S}|} * \sum_{\substack{\mathrm{j}=1 \\ \mathrm{c_j} \in \mathbb{S}}}^{\mathrm{n}=|\mathbb{S}|} \frac{\left|\mathcal{R}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S}) \cap \mathcal{R}^{\mathbb{C}}(\mathrm{c_j})\right|}{\left|\mathcal{R}^{\mathbb{C}}(\mathrm{c_j})\right|} \quad \textbf{(11)}$$

The *density* weights each *related set* in a *related closure* by the degree to which the sample cases distributed.

Theorem 2. For a case base of finite size $\mathbb{C}$, and a case sample set $\mathbb{S} \subseteq \mathbb{C}$, the sum of the *densities* of all elements in $\mathcal{R}^{\mathbb{C}}(\mathbb{S})$ equals to 1.

$$\sum_{\mathrm{i}=1}^{\mathrm{n}=|\mathcal{R}^{\mathbb{C}}(\mathbb{S})|} \mathrm{w}^*\left(\mathrm{r}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S})\right) = 1 \quad \textbf{(12)}$$

Proof. Substitute Equation 11 into Equation 12, we have the left side as

$$\frac{1}{|\mathbb{S}|} * \sum_{\mathrm{i}=1}^{|\mathcal{R}^{\mathbb{C}}(\mathbb{S})|} \sum_{\substack{\mathrm{j}=1 \\ \mathrm{c_j} \in \mathbb{S}}}^{|\mathbb{S}|} \frac{\left|\mathcal{R}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S}) \cap \mathcal{R}^{\mathbb{C}}(\mathrm{c_j})\right|}{\left|\mathcal{R}^{\mathbb{C}}(\mathrm{c_j})\right|} = \frac{1}{|\mathbb{S}|} * \sum_{\substack{\mathrm{j}=1 \\ \mathrm{c_j} \in \mathbb{S}}}^{|\mathbb{S}|} \sum_{\mathrm{i}=1}^{|\mathcal{R}^{\mathbb{C}}(\mathbb{S})|} \frac{\left|\mathcal{R}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S}) \cap \mathcal{R}^{\mathbb{C}}(\mathrm{c_j})\right|}{\left|\mathcal{R}^{\mathbb{C}}(\mathrm{c_j})\right|} \quad \textbf{(13)}$$

According to definition of *related closure* (Def. 4), $\mathcal{R}^{\mathbb{C}}(\mathrm{c_j}) \subseteq \mathcal{R}^{\mathbb{C}}(\mathbb{S})$, therefore we have:

$$\sum_{\mathrm{i}=1}^{|\mathcal{R}^{\mathbb{C}}(\mathbb{S})|} \frac{\left|\mathcal{R}_{\mathrm{i}}^{\mathbb{C}}(\mathbb{S}) \cap \mathcal{R}^{\mathbb{C}}(\mathrm{c_j})\right|}{\left|\mathcal{R}^{\mathbb{C}}(\mathrm{c_j})\right|} = 1 \quad \textbf{(14)}$$

From practical point of view, this means, for all cases in sample $\mathbb{S}$, they are equally important with regarding to the contribution of the total *density* of elements in $\mathcal{R}^{\mathbb{C}}(\mathbb{S})$, no matter what their *related sets* are.

Finally, we define the distance with regard to the competence of two case samples.

Definition 6. Given a case base of finite size $\mathbb{C}$, and a case sample set $\mathbb{S} \subseteq \mathbb{C}$, let $\mathcal{R}^{\mathbb{C}}(\mathbb{C})$ be the measure space and a set $A \subseteq \mathcal{R}^{\mathbb{C}}(\mathbb{C})$

- We define the *competence-based empirical weight* of $\mathbb{S}$ *w.r.t.* A over $\mathbb{C}$ as

$$S^{\mathbb{C}}(A) = \sum_{\substack{i=1 \\ r_i^{\mathbb{C}}(\mathbb{S}) \in A \cap \mathcal{R}^{\mathbb{C}}(\mathbb{S})}}^{i=\left|A \cap \mathcal{R}^{\mathbb{C}}(\mathbb{S})\right|} w^*\left(r_i^{\mathbb{C}}(\mathbb{S})\right) \tag{15}$$

- For two case sample sets $\mathbb{S}_1, \mathbb{S}_2 \subseteq \mathbb{C}$, we define the *competence-based empirical distance* as

$$d^{\mathbb{C}}\,(\mathbb{S}_1, \mathbb{S}_2) = 2 \sup_{A \subseteq \mathcal{R}^{\mathbb{C}}(\mathbb{C})} \left|S_1^{\mathbb{C}}(A) - S_2^{\mathbb{C}}(A)\right| \tag{16}$$

For all sets A which cause the *competence-based empirical distance* to be greater than ε, we say that there is a concept drift. Similar to Kifer, Ben-David & Gehrke's work [16], the set A also depicts a local area where the largest distribution discrepancy between two samples lies.

Now, we consider a simple example to illustrate how to measure the *competence-based empirical distance* between two case sample sets. Suppose there is a case base $\mathbb{C} = \{c_1, c_2, c_3, c_4, c_5, c_6\}$ aims to determine whether a cup of milk turns bad after some hours taken out of a fridge. Assume we have c_1 represents the milk is still good after 4 hours, and c_2 (7hs, good), c_3 (12hs, bad), c_4 (16hs, bad), c_5 (19hs, bad), c_6 (21hs, bad). Also we assume two cases can be used to retrieve each other if they were both taken out within 5 hours. Constructing the competence model over $\mathbb{C}$, we have the *related set* for each case in $\mathbb{C}$ is as follows: $R^{\mathbb{C}}(c_1) = R^{\mathbb{C}}(c_2) = \{c_1, c_2\}$, $R^{\mathbb{C}}(c_3) = \{c_3, c_4\}$, $R^{\mathbb{C}}(c_4) = \{c_3, c_4, c_5, c_6\}$, $R^{\mathbb{C}}(c_5) = R^{\mathbb{C}}(c_6) = \{c_4, c_5, c_6\}$. Our goal is to measure the *competence-based empirical distance* between two sample sets drawn from the case base $S_1 = \{c_1, c_4, c_5\}$ and $S_2 = \{c_2, c_3, c_6\}$. The *related closure* of c_3 with regard to $\mathbb{C}$ is the set of all *related sets* which contain c_3. Thus we have $\mathcal{R}^{\mathbb{C}}(c_3) = \{\{c_3, c_4\}, \{c_3, c_4, c_5, c_6\}\}$. Also we have the *related closure* of S_1 and S_2 as $\mathcal{R}^{\mathbb{C}}(S_1) = \mathcal{R}^{\mathbb{C}}(S_2) = \{\{c_1, c_2\}, \{c_3, c_4\}, \{c_3, c_4, c_5, c_6\}, \{c_4, c_5, c_6\}\}$. The *density* of $\{c_3, c_4\}$ with regard to S_1 is calculated as 1/3*(0/1+1/3+0/2) = 1/9. And the *density* of $\{c_1, c_2\}$, $\{c_3, c_4, c_5, c_6\}$, $\{c_4, c_5, c_6\}$ with regard to S_1 is 1/3, 5/18, 5/18 respectively. Accordingly, the *density* of $\{c_1, c_2\}$, $\{c_3, c_4\}$, $\{c_3, c_4, c_5, c_6\}$, $\{c_4, c_5, c_6\}$ with regard to S_2 is 1/3, 1/6, 1/3, 1/6. In this case, when $A = \{\{c_3, c_4\}, \{c_3, c_4, c_5, c_6\}\}$ or $\{c_4, c_5, c_6\}$, we get 1/9 as the *competence-based empirical distance* between S_1 and S_2.

3.2 A Detection Algorithm

In this section we discuss how to determine whether a concept drift occurs through competence-based empirical distance. Assume we are running a CBR system, and a new case chunk becomes available. Before retaining these new cases, we could like to detect whether there is a concept drift. We measure the *competence-based empirical distance* between current case base and the new case chunk and say a concept drift exists if the distance is large enough (> ε).

If we deem all cases of current case base follow a certain distribution, we may say that there is no significant distribution change between two case samples drawn randomly from the case base. Therefore, the distance between these samples provides a reference for determining ε.

To minimize the error inferred due to sample bias, we incorporate two mechanisms. First, we do multiple experiments and choose ε as the maximum distance rather than rely on a single test. Actually, the number of experiments plays the role of tradeoff between miss detection and false detection error. Second, we split the whole case base into several *competence closures* and draw samples within each *competence closure* respectively. A major concern for us to use *competence closure* but not other methods, such as dividing the case base according to feature values, is that a *competence closure* represents a group of local competence measurements that related to each other. Sampling based on *competence closures* consists with the sense of our competence-based change detection method, also facilitates the work of doing experiments within a certain local competence area when desired. In additional we determine the sample size according to the size of each *competence closure*. That is to draw a larger sample for larger competence closures, and vice versa. By doing this, we expect the case sample set represents the distribution of the original case base to the greatest extend.

The overall process of our change detection method is shown in Figure 1. When new cases become available, we merge these new cases with existing case base. We use this merged case base to represent the whole case domain, and construct competence model on it. Then we draw samples randomly from the original case base and measure the *competence-based empirical distance* between samples. The maximum distance is selected as the bound (ε) for determining whether a concept drift occurs, after a certain number of experiments (*n*).

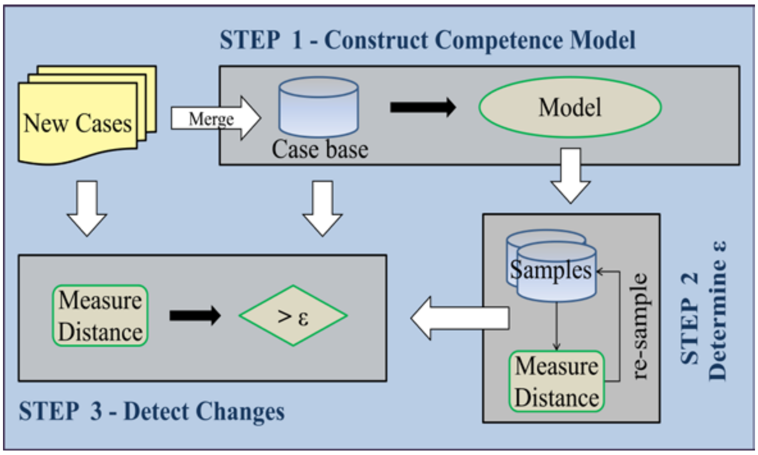

Fig. 1. Competence-based change detection flow chart.

Take our previous example again, but this time we change the environment by a little, thus the milk can last longer. And we get some new cases c_7 (8hs, good), c_8 (15hs good) and c_9 (17hs bad). Merging these new cases into the case base $\mathbb{C}$, we reconstruct the competence model over the merged case base $\mathbb{C}'$. We have $R^{\mathbb{C}'}(c_1) = R^{\mathbb{C}'}(c_2) = R^{\mathbb{C}'}(c_7) = \{c_1, c_2, c_7\}$, $R^{\mathbb{C}'}(c_3) = \{c_3, c_4, c_9\}$, $R^{\mathbb{C}'}(c_4) = R^{\mathbb{C}'}(c_9) = \{c_3, c_4, c_5, c_6, c_9\}$, $R^{\mathbb{C}'}(c_5) = R^{\mathbb{C}'}(c_6) = \{c_4, c_5, c_6, c_9\}$, $R^{\mathbb{C}'}(c_8) = \{c_8\}$. We measure the *competence-based empirical distance* between the new case set $S_{new} = \{c_7, c_8, c_9\}$ and a sample set S_1 drawn from the original case base $\mathbb{C}$. By comparing this distance with distance between two sample sets, S_1 and S_2 for example, drawn from the original case base $\mathbb{C}$, we find there is a concept drift happens, with an decreasing trends for bad cases, and increasing trends for good cases especially around 15 hours (c_8).

4 Experimental Evaluation

In order to evaluate the proposed competence-based change detection method, it is necessary to use simulated data so that the change in generating distributions is known. We use four synthetic datasets based on sets used in other paper concerning concept drift [18]. All the datasets have two classes.

- STAGGER (1S). sudden, noise free. The dataset has three nominal attributes: size (small, medium, large), color (red, green) and shape (circle, non-circle), and has three concepts: 1) [size = small and color = red], 2) [color = green or shape = circle] and 3) [size = medium or large]. Data was randomly generated within the domain and labeled according to current concept.
- MIXED (2M). sudden, noise. The dataset is a mixture of data generated according to two different but overlapped geometric distributions, $f_p(x) = \begin{cases} p(1-p)^{x-1} & 1 \le x < 20 \\ (1-p)^{x-1} & x = 20 \end{cases}$ and $f_p{}'(x) = f_p(21 - x)$. There are two concepts for this dataset. In both concept the positives are generated by $f_{0.25}(x)$, while the negatives are changed from $f_{0.25}{}'(x)$ to $f_{0.33}{}'(x)$. In addition the proportion of the negatives is changed from 1/3 to 3/7. In both concept, we consider a sample as positive if $x \le 11$. The overlapping can be considered as noise. Although the condition of classifying the samples remains the same, the distribution of the data changes, thus concept drift occurs. Figure 2 shows the data distributions before and after concept drift.
- CIRCLES (3C). sudden, noise free. The examples are label according to the condition: if an example is inside the circle, then its label is positive. The change is achieved by displacing the centre of the circle $\big((0.2,0.5) \rightarrow (0.4,0.5)\big)$ and growing its radius $(0.15 \rightarrow 0.2)$. We assume the problem space is $([0,1],[0,1])$, and two cases are considered as similar if their Euclidean distance is not greater than 0.1. Being different from Nishida and

Yamauchi [18], we still consider this dataset as sudden concept drift, and create another dataset CIRCLES-2 based on Stanley's definition on gradual concept drift [27] to compare the results.

- CIRCLES-2 (4C). gradual, noise free. The concept is the same as CIRCLES, but gradually changed over Δx samples. We assume the probability of the coming instance being in the first concept declines linearly as the probability of an instance being in the second concept increases until the first concept is completely replaced by the second concept. That means for the i^{th} new instance c_i, it still has the probability of $\frac{\Delta x - i}{\Delta x}$ to follow the first concept, when $i \geq \Delta x$ the second concept will completely replace the first one. We continuously detect whether there is a concept drift each time a certain number of instances (samples) become available. We assume the size of the case base is ten times the size of the samples. And previous samples containing both concepts are retained and considered as noise for detection.

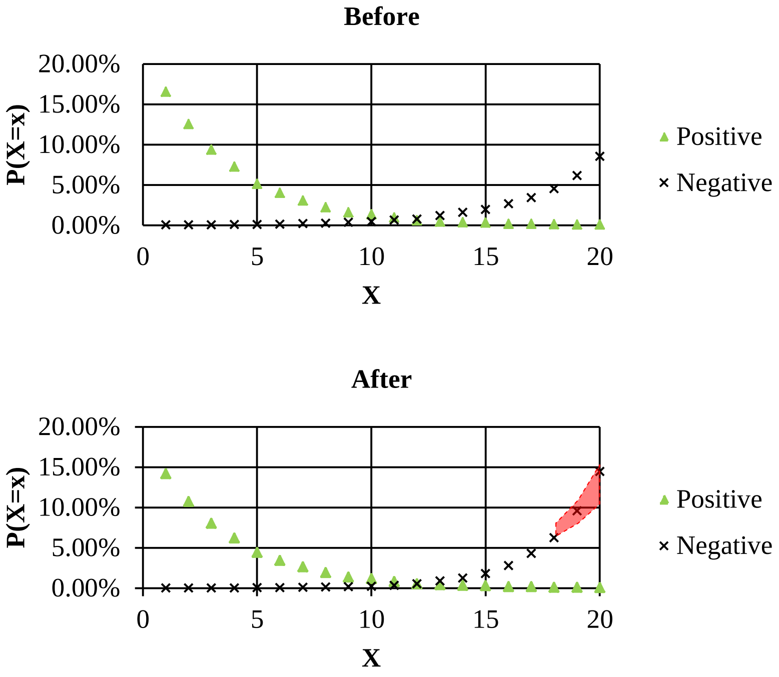

Fig. 2. Distributions of the mixed dataset (2M) before and after concept drift

The experiment results are shown in Table 1. We varied the sample size (N) for each concept and number of experiments (n) used to determine the upper bound (ε) to see how our detection method was affected. We evaluate all results by two types of error rate, false positive (miss detection) and true negative (false detection). All error rates were calculated based on 5K tests.

Table 1. Experiment Results of Competence-based Change Detection

Data Set	Δx	N	n	FP	TN
1S		500	100	0.00%	0.73%
		500	20	0.00%	3.52%
		100	100	0.34%	0.70%
		100	20	0.10%	3.55%
2M		500	100	0.50%	0.80%
		500	20	0.02%	4.36%
		100	100	56.59%	0.97%
		100	20	38.23%	4.50%
3C		100	10	20.05%	10.43%
		200	10	2.1%	9.51%
4C	300	001-100	10	77.32%	10.74%
		101-200	10	62.39%	9.31%
		201-300	10	42.58%	9.13%
		301-400	10	38.43%	9.96%
	600	001-200	10		
		201-400	10		
		401-600	10		
		601-800	10		

It can be seen that for all these dataset, our method can detect change with very low error rates. However, one thing to note is that the false positive error rate increases dramatically with relative a small sample size. This is probably due to the sample bias, considering that this size of samples may not fully represent the distribution of the data. When the sample size is large enough, the error rate remains stable. Second, the number of experiments (n) balances between the two types of error. An increasing of n will lead to an increasing of miss detection error but lower the false detection error and vice versa. Third, by comparing with the results of the third dataset, we found our method achieve a

Finally, being a specialty of our change detection method, it is able to quantify and describe the change detected. For example, we detect a dramatic increase of negative samples when $17 \le x \le 20$ (Fig. 2)

5 Conclusions and Future Works

We present a method for detecting change in the distribution of samples. The method requires no prior knowledge about distribution but measures it through a competence model. The competence-based change detection method can be applied to CBR systems where new case chunks are available sequentially over time. Empirical experiments show that the competence-based change detection method works well for large samples and is not too sensitive to noises. Being special, the competence-based change detection method is able to quantify and describe the change it detects, which makes it more suitable for handling local concept drift.

For future works, the proposed approach for detecting concept drift requires a sample data which is large enough to represent the true data distribution. How to track concept drift with very little data is still a remaining issue. Second, how to construct the competence model and detect change for non-classification problems will be another issue. Finally, detecting change is only the first step towards handling concept drift. Successive methods that takes advantage of change detection are needed to improve the final learning performance.

Acknowledgment. The work presented in this paper was supported by Australian Research Council (ARC) under Discovery Project DP0880739. We also wish to thank the anonymous reviewers for their helpful comments

References

1. Widmer, G. and Kubat, M.: Effective learning in dynamic environments by explicit context tracking. In: Brazdil, P. (eds.) Machine Learning: ECML 1993. LNCS, vol. 667, pp. 227--243. Springer, Heidelberg (1993)
2. Widmer, G. and Kubat, M.: Learning in the Presence of Concept Drift and Hidden Contexts. Machine Learning. 23(1), pp. 69--101 (1996)
3. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 97--106. ACM Press, San Francisco, California (2001)
4. Cohen, L., Avrahami, G., Last, M., Kandel, A.: Info-fuzzy algorithms for mining dynamic data streams. Applied Soft Computing. 8(4), pp. 1283--1294 (2008)
5. Tsymbal, A.: The Problem of Concept Drift: Definitions and Related Work. Technical Report TCD-CS-2004-15, Department of Computer Science, Trinity College Dublin, Ireland (2004)
6. Tsai, C.-J., Lee, C.-I., Yang, W.-P.: Mining decision rules on data streams in the presence of concept drifts. Expert Syst. Appl. 36(2), pp. 1164--1178 (2009)
7. Maloof, M.A., Michalski, R.S.: Incremental learning with partial instance memory. Artificial Intelligence. 154(1-2), pp. 95--126 (2004)
8. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A case-based technique for tracking concept drift in spam filtering. Knowledge-Based Systems. 18(4-5), pp. 187--195 (2005)
9. Klinkenberg, R.: Learning drifting concepts: Example selection vs. example weighting. Intell. Data Anal. 8(3), pp. 281--300 (2004)
10. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 377--382. ACM Press, San Francisco, California (2001)
11. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 226--235. ACM Press, Washington D.C. (2003)
12. Kolter, J.Z., Maloof, M.A.: Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts. J. Mach. Learn. Res. 8, pp. 2755--2790 (2007)
13. Zhang, P., Zhu, X., Shi, Y., Wu, X.: An Aggregate Ensemble for Mining Concept Drifting Data Streams with Noise. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) Advances in Knowledge Discovery and Data Mining: PAKDD 2009. LNCS, vol. 5476, pp. 1021--1029. Springer, Heidelberg (2009)

14. Tsymbal, A., Pechenizkiy, M., Cunningham, P., Puuronen, S.: Dynamic integration of classifiers for handling concept drift. Information Fusion. 9(1), pp. 56--68 (2008)
15. Fan, W.: Systematic data selection to mine concept-drifting data streams. In: 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 128--137. ACM Press, Seattle, Washington (2004)
16. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: 13th International Conference on Very Large Data Bases, pp. 180--191. VLDB Endowment, Toronto, Canada (2004)
17. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with Drift Detection. In: 17th Brazilian Symposium on Artificial Intelligence, pp. 286--295, Springer, Sao Luis, Maranhao, Brazil (2004)
18. Nishida, K., Yamauchi, K.: Detecting Concept Drift Using Statistical Testing. In: 10th International Conference on Discovery Science, pp. 264--269. Springer, Heidelberg, Sendai, Japan (2007)
19. Song, X., Wu, M., Jermaine, C., Ranka, S.: Statistical change detection for multi-dimensional data. In: 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 667--676. ACM Press, San Jose, California (2007)
20. Dries A., Rückert, U.: Adaptive concept drift detection. Statistical Analysis and Data Mining. 2(5-6), pp. 311--327 (2009)
21. Massie, S., Craw, S., Wiratunga, N.: What is CBR competence? BCS-SGAI Expert Update. 8(1), pp. 7--10 (2005)
22. Smyth, B., Keane, M.T.: Remembering To Forget: A Competence-Preserving Case Deletion Policy for Case-Based Reasoning Systems. In: 14th International Joint Conference on Artificial Intelligence. pp. 377--382. Morgan Kaufmann, Montreal, Quebec, Canada (1995)
23. Smyth, B., McKenna, E.: Footprint-Based Retrieval. In: 3rd International Conference on Case-Based Reasoning and Development, pp. 343--357, Springer, Seeon Monastery, Germany (1999)
24. Smyth B., McKenna, E.: Competence Models and the Maintenance Problem. Computational Intelligence. 17(2), pp. 235--249 (2001)
25. Lu, N., Lu, J., Zhang, G.: Maintaining Footprint-Based Retrieval for Case Deletion. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) Neural Information Processing: ICONIP 2009. LNCS, vol. 5864. pp. 318--325. Springer, Heidelberg (2009)
26. Gao, J., Fan, W., Han, J.: On Appropriate Assumptions to Mine Data Streams: Analysis and Practice. In: 7th IEEE International Conference on Data Mining, pp. 143--152, IEEE Computer Society, Omaha, NE (2007)
27. Stanley, K.O.: Learning concept drift with a committee of decision trees. Technical Report UT-AI-TR-03-302, Department of Computer Science, University of Texas at Austin, USA, (2003)