Unsupervised Change Analysis Using Supervised Learning

Shohei Hido, Tsuyoshi Idé, Hisashi Kashima,
Harunobu Kubo, and Hirofumi Matsuzawa

IBM Research, Tokyo Research Laboratory,
1623-14 Shimo-tsuruma, Yamato-shi, Kanagawa, 242-8502 Japan
{hido,goodidea,hkashima,kuboh,matuzawa}@jp.ibm.com

Abstract. We propose a formulation of a new problem, which we call *change analysis*, and a novel method for solving the problem. In contrast to the existing methods of change (or outlier) detection, the goal of change analysis goes beyond detecting whether or not any changes exist. Its ultimate goal is to find the explanation of the changes. While change analysis falls in the category of unsupervised learning in nature, we propose a novel approach based on *supervised* learning to achieve the goal. The key idea is to use a supervised classifier for interpreting the changes. A classifier should be able to discriminate between the two data sets if they actually come from two different data sources. In other words, we use a hypothetical label to train the supervised learner, and exploit the learner for interpreting the change. Experimental results using real data show the proposed approach is promising in change analysis as well as concept drift analysis.

Keywords: change analysis, two-sample test, concept drift.

1 Introduction

Outlier (or novelty) detection is one of the typical unsupervised learning tasks. It aims at deciding on whether or not an observed sample is "strange" based on some distance metric to the rest of the data. Change detection is similar to outlier detection, which is typically formulated as a statistical test for the probability distribution of a data set under some online settings.

In many practical data analysis problems, however, the problem of change detection appears with a slightly different motivation. For example, a marketing researcher may be interested in comparing the current list of customers' profile with a past list to get information about changes. Here, detecting the changes itself is not of particular interest. What the researcher really wants is the detailed information about *how* they changed.

In this paper, we formulate this practically important problem, which we call *change analysis*. In contrast to change detection, we focus on developing a general framework of how to describe a change between two data sets. Clearly, the change analysis problem is an unsupervised learning task in nature. We assume that we are given two data sets, each of which contains a set of unlabeled vectors. Our goal is to find some diagnosis information based on the comparisons between the two data sets, without using side information about the internal structure of the system. The main contribution of this

T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 148–159, 2008.

paper is to show that this essentially unsupervised problem can be solved with *supervised learners*.

To date, the problem of comparing two data sets has been addressed in various areas. For example, the two-sample test [1,2,3], which is essentially to tell whether or not two (unlabeled) data sets are distinct, has a long history in statistics [4]. Another example is concept drift analysis [5,6,7], which basically addresses changes in supervised learners when the (labeled) training data set changes over time. However, most of the existing approaches have a serious drawback in practice in that they focus almost only on whether or not any change exists. As mentioned before, in most of the practical problems, what we really want to know is which variables could explain the change and how.

The layout of this paper is as follows. In Section 2, we describe the definition of the change analysis problem, and give an overview of our approach. Unexpectedly, this unsupervised problem can be solved using supervised learners, as explained in Section 3 in detail. Based on these sections, in Section 4, we present experimental results using real data to show the proposed approach is promising. Finally, we give a brief review of related work in Section 5, and conclude the paper in Section 6.

2 Problem Setting and Overview

In this section, we define a task of change analysis somewhat formally, and give an overview of our approach.

2.1 The Change Analysis Problem

Suppose that we are given two sets of unlabeled data, $\mathrm{X}_\mathrm{A} \equiv \{\boldsymbol{x}_\mathrm{A}^{(1)}, \boldsymbol{x}_\mathrm{A}^{(2)}, \ldots, \boldsymbol{x}_\mathrm{A}^{(N_\mathrm{A})}\}$ and $\mathrm{X}_\mathrm{B} \equiv \{\boldsymbol{x}_\mathrm{B}^{(1)}, \boldsymbol{x}_\mathrm{B}^{(2)}, \ldots, \boldsymbol{x}_\mathrm{B}^{(N_\mathrm{B})}\}$, where N_A and N_B are the numbers of data items in X_A and X_B, respectively. Each of $\boldsymbol{x}_\mathrm{A}^{(i)}$ and $\boldsymbol{x}_\mathrm{B}^{(i)}$ is an i.i.d. sample in a d-dimensional feature space.

This paper addresses two problems about these data sets. The first one is the change detection problem, which is basically the same as the two-sample problem:

Definition 1 (change detection problem). *Given nonidentical data sets X_A and X_B, tell whether or not the difference is significant, and compute the degree of discrepancy between X_A and X_B.*

Note that, unlike concept drift studies, we focus on unlabeled data in this problem. The second problem we address is the *change analysis problem*, which is stated as follows:

Definition 2 (change analysis problem). *Given nonidentical data sets X_A and X_B, output a set of decision rules that explain the difference in terms of individual features*[1].

Since no supervised information is given in getting the decision rules, this is an unsupervised learning task.

[1] The term of "difference analysis" could be more appropriate here, since we do not necessarily confine ourselves within online settings. However, to highlight the contrast to change detection, which is a well-known technical term, we will call the concept change analysis.

To understand the difference between these two problems, let us think about limitations of the two-sample test, which has been thought of as a standard approach to the change detection problem. The two-sample test is a statistical test which attempts to detect the difference between two data sets. Formally, it attempts to decide whether $P_{\mathrm{A}} = P_{\mathrm{B}}$ or $P_{\mathrm{A}} \neq P_{\mathrm{B}}$, where P_{A} and P_{B} are probability distributions for X_{A} and X_{B}, respectively. In statistics, two-sample tests are classified into two categories [4]. The first category is the parametric method, where a parametric functional form is explicitly assumed to model the distribution. In practice, however, such density modeling is generally difficult, since the distribution of real-world data does not have a simple functional form. In addition, even if a good parametric model such as Gaussian had been obtained, explaining the origin of the difference in terms of individual variables is generally a tough task, unless the variables are independent.

The second category is the nonparametric method, which allows us to conduct a statistical test without density modeling. If our interest were to detect only the discrepancy between two data sets, distance-like metrics such as the maximum mean discrepancy [3], the Kolmogorov-Smirnov statistic [1], energy-based metrics [8], and nearest neighbor statistics [2] are available for solving the change detection problem. However, these methods are not capable of handling the change analysis problem. While some of the two-sample tests offer asymptotic distributions for the data in such limit as large number of samples, it is generally very hard to answer the change analysis problem in practice. This is because, first, such a distribution is an asymptotic distribution, so it generally cannot be a good model for real-world data, where, e.g., the number of samples is finite. Second, since the nonparametric approach avoids density modeling, little information is obtained about the internal structure of the data.

2.2 Overview of Our Approach

Considering the limitations of the two-sample test, we propose a simple approach to these two problems. Our key idea is just as follows: Attach a hypothetical label $+1$ to each sample of X_{A}, and -1 to each sample of X_{B}. Then train a classifier in a supervised fashion. We call this classifier the *virtual classifier* (VC) hereafter.

Figure 1 shows a high-level overview of our approach, where ◯ and □ indicate labels of $+1$ and -1, respectively. In our approach, if two data sets actually have the differences, they should be correctly classified by the classifier. Thus a high classification accuracy p indicates a difference between X_{A} and X_{B}. For example, if $P_{\mathrm{A}} = P_{\mathrm{B}}$, the classification accuracy will be about 0.5 when $N_{\mathrm{A}} = N_{\mathrm{B}}$. However, if p is significantly larger than 0.5, we infer that the labels make difference, so $P_{\mathrm{A}} \neq P_{\mathrm{B}}$.

In addition, to solve the change analysis problem, we take advantage of the interpretability of classification algorithms. For example, the logistic regression algorithm gives the weight of each feature representing the importance. For another example, if the decision tree is employed, variables appearing in such nodes that are close to the root should have a major impact. In this way, we can get decision rules about the changes from the VC.

The advantages of this VC approach are as follows: First, it can solve the change detection and analysis problem at the same time. The classifier readily gives the degree of change as the classification accuracy, and also provides diagnosis information about

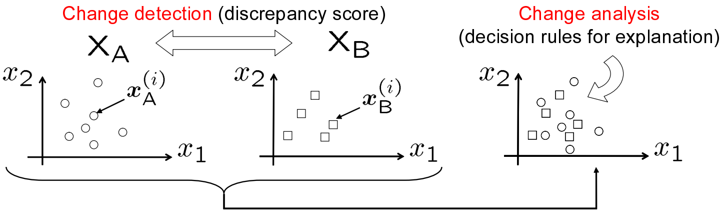

Change detection (discrepancy score)
Change analysis
(decision rules for explanation)
X_A
X_B
x_2
x_1
$\boldsymbol{x}_\mathrm{A}^{(i)}$
$\boldsymbol{x}_\mathrm{B}^{(i)}$

Fig. 1. High-level overview of the virtual classifier approach

changes through its feature selection functions. Second, the VC approach does not need density estimation, which can be hard especially for high dimensional data. Finally, the VC approach allows us to evaluate the significance of changes simply by a binomial test. This is an advantage over traditional nonparametric two-sample tests, which have focused on asymptotic distributions that hold only in some limit.

3 Virtual Classifier Approach to Change Analysis

This section presents details of our supervised learning approach to change analysis. For notations, we use bars to denote data sets including the hypothetical labels, such as $\bar{\mathrm{X}}_\mathrm{A} \equiv \{(\boldsymbol{x}_\mathrm{A}^{(i)}, +1) \mid i = 1, ..., N_\mathrm{A}\}$ and $\bar{\mathrm{X}}_\mathrm{B} \equiv \{(\boldsymbol{x}_\mathrm{B}^{(i)}, -1) \mid i = 1, ..., N_\mathrm{B}\}$. The prediction accuracy of VCs is represented by p.

3.1 Condition of No Change

Suppose that we are given the combined data set $\bar{\mathrm{X}} \equiv \bar{\mathrm{X}}_\mathrm{A} \cup \bar{\mathrm{X}}_\mathrm{B}$, and a binary classification algorithm L. We train L using $\bar{\mathrm{X}}$, and evaluate the classification accuracy p, making use of k-fold cross validation (CV). In particular, randomly divide $\bar{\mathrm{X}}$ into k equi-sized portions, leave out one portion for test, and use the remaining $(k-1)$ portions for training. The overall prediction accuracy p is computed as the average of those of the k classifiers.

If $P_\mathrm{A} = P_\mathrm{B}$, classification of each of the samples in $\bar{\mathrm{X}}$ by L can be viewed as a Bernoulli trial. Thus the log-likelihood of $N_\mathrm{A} + N_\mathrm{B}$ trials over all the members of $\bar{\mathrm{X}}$ will be

$$\ln \left[q^{N_\mathrm{A}} (1-q)^{N_\mathrm{B}} \right]$$

under the assumption of i.i.d. samples, where q is the probability of the class A. By differentiating this w.r.t. q, and setting the derivative zero, we have the maximum likelihood solution of this binomial process as $q = N_\mathrm{A}/(N_\mathrm{A} + N_\mathrm{B})$. Since the classification accuracy p should be $\max\{q, 1-q\}$, we see that p is given by

$$p_\mathrm{bin} \equiv \frac{\max\{N_\mathrm{A}, N_\mathrm{B}\}}{N_\mathrm{A} + N_\mathrm{B}}, \tag{1}$$

where the subscript represents binomial.

If $P_\mathrm{A} \neq P_\mathrm{B}$, so the information of the class labels is important, the classification accuracy will be considerably higher than p_bin. Specifically, the larger the differences they have, the higher the prediction accuracy will become. One of the major features of our VC approach is that it enables us to evaluate the significance of p via a binomial test. Consider a null hypothesis that the prediction accuracy is given by $p_{\rm bin}$, and assume $N_{\rm A} > N_{\rm B}$ for simplicity. For a value of the confidence level $\alpha > 0$, we reject the null hypothesis if

$$\sum_{n_{\rm A}=Np}^{N} \frac{N!}{n_{\rm A}!(N-n_{\rm A})!} p_{\rm bin}^{n_{\rm A}} (1-p_{\rm bin})^{N-n_{\rm A}} \leq \alpha, \tag{2}$$

where $N = N_{\rm A} + N_{\rm B}$. This means that the class labels are so informative that p is sufficiently higher than $p_{\rm bin}$. If we parameterize the critical probability as $(1+\gamma_\alpha)p_{\rm bin}$, the condition of no change is represented as

$$p < (1+\gamma_\alpha)p_{\rm bin}. \tag{3}$$

For a numerical example, if $N = 1000$ and $p_{\rm bin} = 0.5$, the 5% and 1% confidence levels correspond to $\gamma_{0.05} = 0.054$ and $\gamma_{0.01} = 0.076$, respectively. For relatively large N, Gaussian approximation can be used for computing γ_α [4].

3.2 Change Analysis Algorithm

Once the binomial test identifies that the difference between $\rm X_{\it A}$ and $\rm X_{\it B}$ is significant, we re-train L (or another type of classification algorithm) using all the samples in $\bar{\rm X}$. If some features play a dominant role in the classifier, then they are the ones that characterize the difference. As an example, imagine that we have employed the C4.5 decision trees [9] as L. The algorithm iteratively identifies the most important feature in terms of information gain, so such features that appear closest to the root will be most important. Thus focusing on such nodes amounts to feature selection, and the selected features are the ones that explain the difference. In this way, feature selection and weighting functions of L are utilized in change analysis.

We summarize our change analysis algorithm in Fig. 2. The first half (1-3) essentially concerns change detection by evaluating the significance of the changes through the binomial test, while the second half (4-5) addresses change analysis. As shown, there are two input parameters, α and k.

3.3 Application to Labeled Data

While the algorithm in Fig. 2 is for unlabeled data, we can extend the algorithm for labeled data. This extension is practically important since it enables us to do change analysis between classifiers. Suppose that we are given a classification algorithm M, and two labeled data sets $\rm D_A$ and $\rm D_B$, defined as $\{(\boldsymbol{x}_{\rm A}^{(i)}, y_{\rm A}^{(i)})|i = 1, ..., N_{\rm A}\}$ and $\{(\boldsymbol{x}_{\rm B}^{(i)}, y_{\rm B}^{(i)})|i = 1, ..., N_{\rm B}\}$, respectively, where $y_{\rm A}^{(i)}$ and $y_{\rm B}^{(i)}$ represent class labels. We train M based on $\rm D_A$ and $\rm D_B$ to obtain classifiers $\rm M_A$ and $\rm M_B$, respectively. What we wish to solve is a change analysis problem between $\rm M_A$ and $\rm M_B$: Output a set of decision rules that explain the difference between $\rm M_A$ and $\rm M_B$ in terms of individual features.

```
**Algorithm:** Change Analysis
INPUT:
  · Two data sets $\mathrm{X_A}$ and $\mathrm{X_B}$
  · Binary classification algorithm L
  · Number of folds $k$
  · Significance level $\alpha > 0$
1. Give the positive label to each sample of $\mathrm{X_A}$, and the negative label to each sample of $\mathrm{X_B}$.
2. Train L based on $k$-fold cross-validation to obtain the estimated predictive accuracy $p$.
3. If $p < p_{\mathrm{bin}}(1 + \gamma_\alpha)$, then quit. Otherwise, report that $\mathrm{X_A}$ and $\mathrm{X_B}$ have different distributions.
4. Re-train L on all of the data.
5. Investigate the trained classifier to understand the differences between $\mathrm{X_A}$ and $\mathrm{X_B}$.
```

Fig. 2. The virtual classifier algorithm for change analysis

To solve this, we create unlabeled data sets based on the following strategy. For each sample, $\boldsymbol{x}_{\mathrm{A}}^{(i)}$ or $\boldsymbol{x}_{\mathrm{B}}^{(i)}$, we make classification with both $\mathrm{M_A}$ and $\mathrm{M_B}$. If the predictions are inconsistent, then we put the sample into a set $\mathrm{X_A}$, otherwise into $\mathrm{X_B}$. Scanning all the samples, we have two unlabeled data sets $\mathrm{X_A}$ and $\mathrm{X_B}$. By construction, $\mathrm{X_A}$ characterizes the inconsistencies between $\mathrm{M_A}$ and $\mathrm{M_B}$, while $\mathrm{X_B}$ characterizes their commonalities. Thus, by making use of the change analysis algorithm in Fig. 2 for these $\mathrm{X_A}$ and $\mathrm{X_B}$, detailed information about the inconsistencies will be obtained. In our context, the quantity

$$\rho \equiv N_{\mathrm{inc}}/(N_{\mathrm{A}} + N_{\mathrm{B}}) \tag{4}$$

works as the degree of the inconsistencies between $\mathrm{M_A}$ and $\mathrm{M_B}$ (or $\mathrm{D_A}$ and $\mathrm{D_B}$), where N_{inc} represents the number of samples whose predictions are inconsistent.

When the number of possible values for the target variable is small, it is useful to extend the change analysis algorithm to include multi-class classification. As an example, suppose that the given label is binary, i.e. $y_{\mathrm{A}}^{(i)}, y_{\mathrm{B}}^{(i)} \in \{\pm 1\}$. We separate the inconsistent set $\mathrm{X_A}$ into two subsets $\mathrm{X_{A1}}$ and $\mathrm{X_{A2}}$. Here, $\mathrm{X_{A1}}$ consists of the inconsistent samples whose original prediction is $+1$ but cross-classification gives -1. Similarly, $\mathrm{X_{A2}}$ consists of the inconsistent samples that make a transition from -1 to $+1$. Then we apply a three-class classification algorithm L to classify $\mathrm{X_{A1}}$, $\mathrm{X_{A2}}$, and $\mathrm{X_B}$. Finally, we examine the resulting classifier for each type of disagreement.

4 Experiment

We evaluated the utility of the VC approach for change analysis using synthetic as well as real-world data. In the following experiments, we used $\alpha = 0.05$ and $k = 10$ unless otherwise noted. For a classification algorithm L, we mainly used the C4.5 decision trees (DT) algorithm implemented as `J48` in Weka [9], which has a parameter named $minNumObj$ meaning the minimum number of instances per leaf. To see the degree of linear separability between the two data sets, we additionally used logistic regression (LR) also implemented as `Logistic` in Weka. Two parameters in `Logistic` (a ridge parameter and the maximum iterations) were left unchanged to the default values (10^{-8} and infinity, respectively).

4.1 Synthetic Data

We conducted two experiments based on synthetic data with $N_{\mathrm{A}} = N_{\mathrm{B}} = 500$. For this number of samples, the critical accuracy is given by 0.527 ($\gamma_{0.05} = 0.054$). In both of the experiments, the ten features were independently generated based on zero-mean Gaussians.

For the *first* experiment, the data sets X_{A} and X_{B} were designed so that P_{A} and P_{B} had a significant difference. In X_{A}, the standard deviations (denoted by σ) were set to be 1.0 except for $Attr1$ (the first feature), where σ was set to be 4.0. On the other hand, in X_{B}, all the σs were 1.0 except for $Attr2$ (the second feature), where σ was set to be 4.0. Figure 3 (a) shows the marginal distribution of this data set in the $Attr1$-$Attr2$ space. Our goal is to pick up $Attr1$ and $Attr2$ as features that are responsible for the difference.

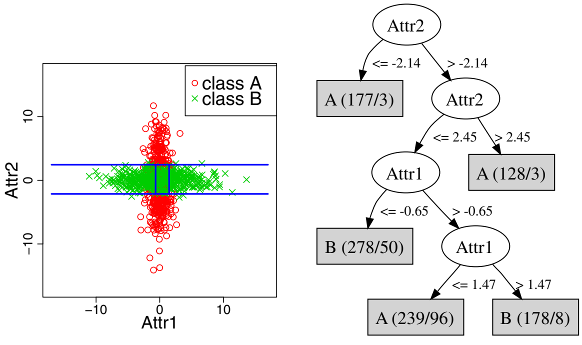
We conducted change analysis for this data set with $minNumObj = 10$ for DT. The estimated prediction accuracy p computed by 10-fold CV was 0.797 (DT), which far exceeds the critical accuracy. This means that the two data sets were correctly judged as being different. Figure 3 (b) represents the DT as the VC, where the labels of the ellipses and the edges show the split variables and the decision rules, respectively. The shaded boxes enclose the class labels as well as (1) the number of instances fallen into the node and (2) the number of misclassified instances in the form of $(1)/(2)$. The latter is to be omitted when zero. The decision boundaries found by the DT are shown by the lines in Fig. 3 (a). Clearly, the model learned the intended nonlinear change between $Attr1$ and $Attr2$. Note that, when LR was used as L, 10-fold CV gave only $p = 0.505$, which is below the critical accuracy. This result clearly shows the crucial role of nonlinear decision boundaries.

For the *second* experiment, P_{A} and P_{B} were designed to be the same. In both X_{A} and X_{B}, all the σs were set to be 1.0 except for $Attr2$, where $\sigma = 4.0$. Figure 4 shows the marginal distribution corresponding to Fig. 3 (a). In contrast to the first experiment, the DT model naturally showed a low p of 0.500, indicating that the differences were not statistically significant. This result shows that our approach using DT generates a valid classifier with statistical significance only when the data set contains a difference between classes.

4.2 Spambase Data

Spambase is a public domain data set in UCI Machine Learning Repository [10]. While the original data contains spam and non-spam email collections, we used only the 1,813 instances belonging to the spam email set. The features consist of fifty-five continuous values of word and symbol statistics. We divided the spam set into halves, X_{A} and X_{B}, keeping the original order of the instances unchanged. In this setting, the critical accuracy is 0.520 ($\gamma_{0.05} = 0.039$). We performed change analysis for X_{A} and X_{B} to see if there was any hidden shift in the data. We used $minNumObj = 2$ for DT.

Interestingly, the 10-fold CV produced a rather high prediction accuracy of $p = 0.548$ (DT), which is higher than the critical accuracy. According to the VC, the major features were the frequencies of the words ‘edu’, ‘85’, and ‘hp’, although space limitation does not permit showing the output DT. Considering the additional fact that LR produced just $p = 0.515$, we conclude that the spam class in Spambase has some



(a) (b)

Fig. 3. (a) Distribution over $Attr1$ and $Attr2$ in the first synthetic data, and (b) the resulting virtual classifier

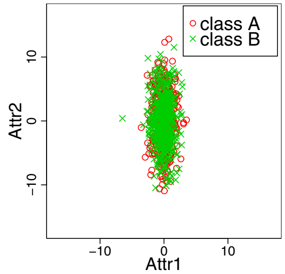


Fig. 4. Distribution over $Attr1$ and $Attr2$ in the second synthetic data

nonlinear changes on the word frequencies, which are difficult to find using a linear model like LR. This result is of particular practical importance, since it suggests that learning algorithms that depend on the order of the training samples might tend to have considerable biases.

4.3 Enron Email Data

The Enron email data set is an archive of real email at the now defunct Enron Corporation, and no class labels are available [11]. We used the year 2001 subset that contains 272,823 email messages in a bag-of-words representation [12]. We separated the data into the first (1H) and the second (2H) halves of this year, and generated feature vectors by including the 100 and 150 most frequent words in each period. Meaningless zero vectors including none of the selected feature words were omitted. Each half was further divided into halves to allow comparison on quarterly basis. We conducted change analysis within either 1H or 2H with $numMinObj = 1,000$. For example, in the analysis of 2H, $\mathrm{X_A}$ and $\mathrm{X_B}$ roughly correspond to the data in the third (3Q) and fourth (4Q) quarters, respectively.

Table 1 shows the estimated prediction accuracies. We see that both LR and DT mark accuracies much higher than the critical accuracies. To explore the details of the differences, we picked the 2H data, and did change analysis to obtain the DT in Fig. 5, where top 5 nodes from the root have been selected, comparing between 100- and 150-word models. The notation of the trees are the same as Section 4.1, although the rank of each feature has been added here ('access' is 44th frequent, etc.). The threshold values represent the occurrence numbers of feature words in each email. Since we followed the simple frequency-based feature generation strategy, the 150-word tree tends to include such words that bear particular meanings.

We see that 'position' is at the root node in the 150-word model in spite of its less frequency (144th rank). Enron went bankrupt at the end of 2001. If we imagine what had been talked about by the employees who were doomed to lose their job position,

Table 1. Prediction accuracies on Enron

Data set		Algorithm	
Period	Words	LR	DT
2001-1H	100	64.3%	67.4%
2001-1H	150	65.4%	68.4%
2001-2H	100	60.9%	62.8%
2001-2H	150	62.3%	64.1%

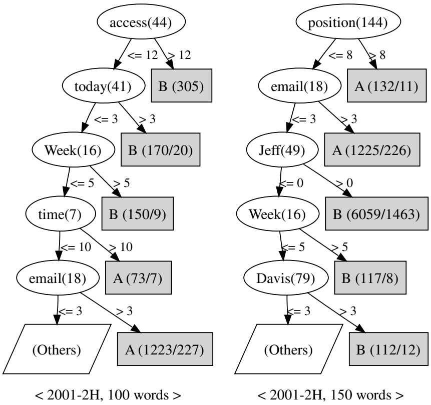


Fig. 5. VCs on the Enron 2001-2H data set

this result is quite suggestive. In addition, we see that ‘Jeff’ and ‘Davis’ are dominant features to characterize the 4Q data. Interestingly, the name of CEO of Enron in 2001 was Jeffrey Skilling, who unexpectedly resigned from this position on August 2001 after selling all his stock options. Many employees must have said something to him at the moment of the collapse. For Davis, there was a key person named Gray Davis, who was California’s Governor in the course of the California electricity crisis in the same year. It may result from his response to the investigation of Enron in 4Q. Note that the VC has discovered these key persons without any newspaper information, demonstrating the utility in studying the dynamics of complex systems such as Enron.

4.4 Academic Activities Data

As an example of application to labeled and categorical data, we performed change analysis for “academic activities” data collected in a research laboratory. This data set consists of 4,683 records over five years in the form of $(\boldsymbol{x}^{(s)}, y^{(s)})$, where s is the time index and $y^{(s)}$ represents a binary label of either ‘Y’ (meaning important) or ‘N’ (unimportant). Each of the vectors $\boldsymbol{x}^{(s)}$ includes three categorical features, *title* , *group*, and *category*, whose values are shown in Table 2.

Since the labels are manually attached to $\boldsymbol{x}^{(s)}$s by evaluating each activity, it greatly depends on subjective decision-making of the database administrator. For example, some administrators might think of PAKDD papers as very important, while other might not. Triggered by such events as job rotations of the administrator and revisions of evaluation guidelines, the trend of decision-making is expected to change over time. The purpose on this analysis is to investigate when and what changes have occurred in the decision criteria to select importance labels.

We created 14 data subsets by dividing the data on quarterly basis, denoted by D_1, $D_2, \ldots, D_{14}$. First, to see whether or not distinct concept drifts exist over time, we computed the inconsistency score ρ (see Eq. (4)) between neighboring quarters. Specifically, we think of $\mathrm{D_A}$ and $\mathrm{D_B}$ as D_t and D_{t+1} for $t = 1, 2, ..., 13$. For M, we employed decision trees. Figure 6 shows the inconsistency score ρ for all the pairs. We see that

Table 2. Three features and their values in the academic activity data

category	*title*	*group*
GOVERNANCE, EDITOR, ORGANIZATION, PROFESSIONAL ACTIVITY	COMMITTEE, MEMBER, AWARD, OTHERS	UNIVERSITY, DOMESTIC, STANDARD, PUBLISHER, SOCIETY1, OTHERGROUPS

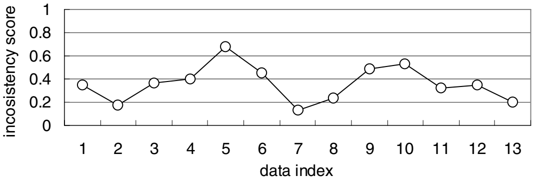


Fig. 6. Inconsistency scores ρ between D_t and D_{t+1}. The largest score can be seen where $t = 5$.

two peaks appear around $t = 5$ and $t = 10$, showing clear concept drifts at those periods. Interestingly, these peaks correspond to when the administrator changed off in reality, suggesting the fact that the handover process did not work well.

Next, to study what happened around $t = 5$, we picked D_5 and D_6 for change analysis. Following the procedure in Section 3.3, we obtained the VC as shown in Fig. 7. Here, we used a three-class DT based on three sets X_{A1}, X_{A2} and X_{B}, where X_{A1} includes samples whose predicted labels make a transition of $\mathrm{Y} \rightarrow \mathrm{N}$. The set X_{A2} includes samples of $\mathrm{N} \rightarrow \mathrm{Y}$, while X_{B} includes consistent samples of $\mathrm{Y} \rightarrow \mathrm{Y}$. If we focus on the leaves of 'NY' in Fig. 7 representing the transition from N to Y, we find interesting changes between D_5 and D_6: The new administrator at $t = 6$ tended to put more importance on such academic activities as program and executive committees as well as journal editors.

One might think that there can be a simpler approach that two decision trees M_5 and M_6 are directly compared, where M_5 and M_6 are decision trees trained only within D_5

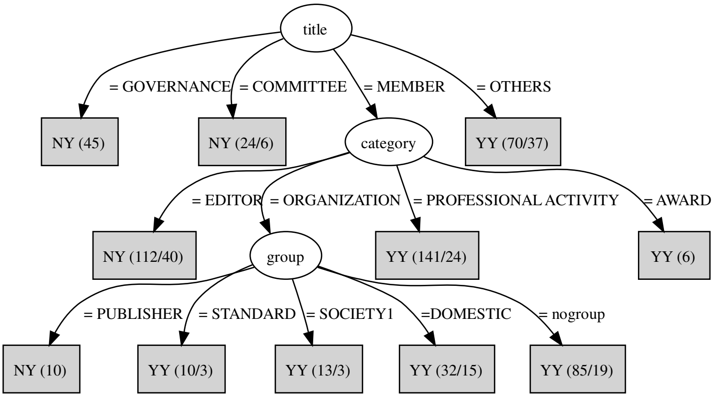


Fig. 7. Virtual classifier for D_5 and D_6

and D_6, respectively. However, considering complex tree structures of decision trees, we see that direct comparison between different decision trees is generally difficult. Our VC approach provides us a direct means of viewing the difference between the classifiers, and is in contrast to such a naive approach.

5 Related Work

The relationship between supervised classifiers and the change detection problem had been implicitly suggested in the 80's [2], where a nearest-neighbor test was used to solve the two-sample problem. However, it did not address the problem of change analysis. In addition, the nearest-neighbor classifier was not capable of explaining changes, since it did not construct any explicit classification model. FOCUS is another framework for quantifying the deviation between the two data sets [13]. In the case of supervised learning, it constructs two decision trees (dt-models) on each data set, then expands them further until both trees converge to the same structure. The differences between the numbers of the instances which fall into the same region (leaf) indicate the deviation between the original data sets. In high-dimensional settings, however, the models should become ineffective since the size of the converged tree increases exponentially therefore the method requires substantial computational cost and massive instances.

Graphical models such as Bayesian networks [14] are often used in the context of root cause analysis. By adding a variable indicating one of the two data sets, in principle Bayesian networks allow us to handle change analysis. However, a graphical modeling approach inevitably requires a lot of training data and involves extensively time-consuming steps for graph structure learning. Our VC approach allows us to directly explain the data set labels. This is in contrast to graphical model approaches, which basically aim at modeling the joint distribution over all variables.

In stream mining settings, handling concept drift is one of the essential research issues. While much work has been done in this area [5,6,7], little of that addresses the problem of change analysis. One of the exceptions is KBS-stream [15] that quantifies the amount of concept drift, and also provides a difference model. The difference model of KBS-Stream tries to correctly discriminate the positive examples from the negative examples in the misclassified examples under the current hypothesis. On the other hand, our VC tries to correctly discriminate the misclassified examples by the current hypothesis against the correctly classified examples. Both models are of use to analyze concept drift, but the points of view are slightly different.

Other studies such as ensemble averaging [16] and fast decision trees [17] tackle problems which are seemingly similar to but essentially different from change analysis.

6 Conclusion

We have proposed a new framework for the change analysis problem. The key of our approach is to use a virtual classifier, based on the idea that it should be able to tell the two data apart if they came from two different data sources. The resulting classifier is a model explaining the differences between the two data sets, and analyzing this model allows us to obtain insights about the differences. In addition, we showed that the significance of the changes can be statistically evaluated using the binomial test. The experimental results demonstrated that our approach is capable of discovering interesting knowledge about the difference.

For future work, although we have used only decision trees and logistic regression for the virtual classifier, other algorithms also should be examined. Extending our method to allow on-line change analysis and regression models would also be interesting research issues.

References

1. Friedman, J., Rafsky, L.: Multivariate generalizations of the Wolfowitz and Smirnov two-sample tests. Annals of Statistics 7, 697–717 (1979)
2. Henze, Z.: A multivariate two-sample test based on the number of nearest neighbor type coincidences. Annals of Statistics 16, 772–783 (1988)
3. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: Advances in Neural Information Processing Systems 19, pp. 513–520. MIT Press, Cambridge (2007)
4. Stuart, A., Ord, J.K.: Kendall's Advanced Theory of Statistics, 6th edn., vol. 1. Arnold Publishers Inc. (1998)
5. Fan, W.: Streamminer: A classifier ensemble-based engine to mine concept-drifting data streams. In: Proc. the 30th Intl. Conf. Very Large Data Bases, pp. 1257–1260 (2004)
6. Wang, H., Yin, J., Pei, J., Yu, P.S., Yu, J.X.: Suppressing model overfitting in mining concept-drifting data streams. In: Proc. the 12th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining, pp. 20–23 (2006)
7. Yang, Y., Wu, X., Zhu, X.: Combining proactive and reactive predictions for data streams. In: Proc. the 11th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining, pp. 710–715 (2005)
8. Zech, G., Aslan, B.: A multivariate two-sample test based on the concept of minimum energy. In: Proceedings of Statistical Problems in Particle Physics, Astrophysics, and Cosmology, pp. 8–11 (2003)
9. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools. Morgan Kaufmann, San Francisco (2005)
10. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
11. Klimt, B., Yang, Y.: The Enron Corpus: A New Dataset for Email Classification Research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 217–226. Springer, Heidelberg (2004)
12. Other forms of the Enron data:
    `http://www.cs.queensu.ca/~skill/otherforms.html`
13. Ganti, V., Gehrke, J.E., Ramakrishnan, R., Loh, W.: A framework for measuring changes in data characteristics. Journal of Computer and System Sciences 64(3), 542–578 (2002)
14. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco (1988)
15. Scholz, M., Klinkenberg, R.: Boosting classifiers for drifting concepts. Intelligent Data Analysis Journal 11(1), 3–28 (2007)
16. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: Proc. the 7th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining, pp. 377–382 (2001)
17. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proc. the 7th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining, pp. 97–106 (2001)