Detecting Concept Drift in Medical Triage

Hamish Huggard
University of Auckland
hhug934@aucklanduni.ac.nz

Yun Sing Koh
University of Auckland
ykoh@cs.auckland.ac.nz

Gillian Dobbie
University of Auckland
g.dobbie@auckland.ac.nz

Edmond Zhang
Orion Health
edmond.zhang@orionhealth.com

ABSTRACT

Clinicians triage patients who are referred to a medical facility based on the details provided in their accompanying referral documents, which contain a mix of free text and structured data. By training a model to predict triage decisions from these referral documents, we can partially automate the triage process, resulting in more efficient and systematic triage decisions. One of the difficulties of this task is maintaining robustness against changes in triage priorities due to changes in policy, funding, staff, or other factors. This is reflected as changes in relationship between document features and triage labels, also known as *concept drift*. These changes must be detected so that the model can be retrained to reflect the new environment. We introduce a new concept drift detection algorithm for this domain called calibrated drift detection method (CDDM). We evaluated CDDM on benchmark and synthetic medical triage datasets, and find it competitive with state-of-the-art detectors, while also being less prone to false positives from feature drift.

KEYWORDS

Concept Drift, Medical Triage, Evolving Data

ACM Reference Format:
Hamish Huggard, Yun Sing Koh, Gillian Dobbie, and Edmond Zhang. 2020. Detecting Concept Drift in Medical Triage. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401228

1 INTRODUCTION

When a patient is referred to a medical facility, the referral is documented with free text and structured data, containing such information as condition, comorbidities, and demographics. From this data a clinician will make a decision about how urgently the patient needs to be addressed, and assign the patient a triage priority label. This documentation presents an opportunity: using supervised learning we can train a model to predict triage labels for referral documents, and incorporate it into a decision support system to help clinicians make more efficient and systematic triage decisions. For example, clinicians may review referrals in the priority order recommended by the system.

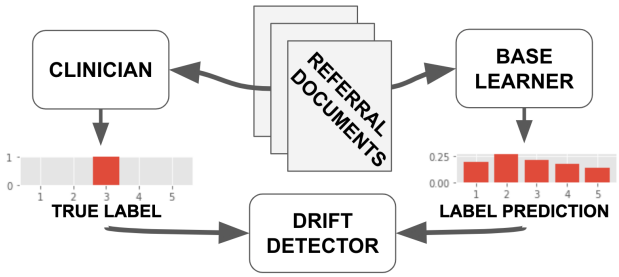


Figure 1: Concept drift detection in medical triage.

Staff, resources, policy, and best practices evolve over time in medical environments, so it is important that a model is able to detect and adapt to these changes. For example, if one ethnic group is discovered to be particularly susceptible to some condition, then clinicians may begin to give this group higher priority. A decision support system must be able to detect and adapt to these changes. This kind of change in a data stream is known as *concept drift*.

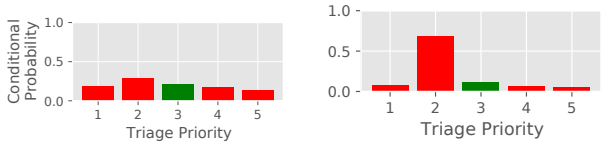
Concept drift can be detected by monitoring a model's accuracy over time. If there exists a time t such that the model is more accurate before t than after t, then this is evidence that the nature of the data stream has changed. If this is the case, then the model should be retrained on data which occurred *after* the change at t. This approach, or variants of it, are the most common strategy for detecting concept drift [1].

This paper explores concept drift detection for a medical triage support system. This setting is illustrated in Figure 1. A base learner is trained on historical examples of triage decisions. When a new referral document arrives, the base learner predicts the triage label. This label provides a first pass triage of the patient, and is eventually reviewed by a clinician, at which point the model's prediction and the true label are fed into the drift detector, which predicts if and when concept drift has occurred. If concept drift is detected, the model is retrained on the post-drift data.

There are several special requirements for drift detection in this application. In a typical concept drift setting we may expect a high rate of incoming data to be processed by a simple model online. By contrast, medical referrals are relatively low volume. Computational economy (low runtime and memory consumption) are not essential requirements. Instead, the emphasis is on precision:

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401228



(a) Weak evidence of drift. **(b) Strong evidence of drift.**

Figure 2: Illustration of the benefit of using probabilistic predictions for concept drift detection.

not causing costly retraining unnecessarily, and recall: promptly detecting concept drift to avoid clinical errors.

2 DETECTING DRIFT WITH CALIBRATION

We propose a novel approach to concept drift detection for this application, called Calibrated Drift Detection Method (CDDM). Existing approaches to concept drift detection monitor model accuracy, and predict drift when accuracy declines. However, accuracy can be quite a coarse metric. For example, if the rate of a condition whose priority is hard to predict increases, then accuracy will decline. Existing drift detectors would likely recommend retraining the model, even though the decision boundary has not changed, so this expensive operation will not recover accuracy. Changes in data stream characteristics, which can be attributed to changes in the feature statistics are called *virtual drift*, whereas those due to changes in decision boundaries are called *real drift*.

Virtual drift will often be a problem in medical triage, as the characteristics of documents will change with demographics, epidemiology, medical facilities, and other factors. It is therefore important that a drift detector is able to distinguish virtual drift from real drift. CDDM achieves this by detecting changes in the calibration [5] of the base learner, rather than changes in accuracy. This requires the learner to predict probability distributions over labels, rather than make point predictions. In the case of a neural network, the probability distribution is the activations of the softmax layer. In the case of a Naïve Bayes, the probability distribution corresponds to the posterior probability of each label.

Figure 2 illustrates how probability distributions can help make concept drift detection more precise. Figure 2a shows a model's conditional probabilities for each triage priority label for a referral document. Previous similar documents have tended to be assigned a priority of 2, so this label receives the highest conditional probability and is the model's prediction. However, the actual priority label assigned by a clinician is 3, which may indicate concept drift. Figure 2b is a similar situation, except that previous similar documents have *more consistently* been assigned a priority label of 2, so more probability mass is assigned to this label. The fact that the model is more "confident" in this incorrect prediction than in Figure 2a is stronger evidence of drift. However, because existing drift detection algorithms only consider point predictions from models, this information will be missed.

Let us say that a model assigns probabilities to each of 5 labels, which in this case are triage priorities. For example, the model assigns probabilities $(0.1, 0.4, 0.3, 0.1, 0.1)$ to priority levels of $(1, 2, 3, 4, 5)$ respectively. Let q be the maximum probability assigned to any of the labels; in this case $q = 0.4$. Let y indicate whether a prediction was correct; in this case if the correct priority level is 2 then $y = 1$ otherwise $y = 0$.

A predictor is said to be "calibrated" if events assigned probability q do in fact obtain at a rate of q [6]. For example, if a calibrated predictor assigns a probability 90% to 10 distinct events, then in expectation 9 of these events should turn out to be true and 1 should not. Our null hypothesis will be that the base models *are* calibrated. See [6] for discussion on adjustments which can be made to model outputs to improve their calibration. When models are not calibrated, we will take this as a sign of concept drift.

When a model is calibrated, we have

$$\mathbb{E}\left[q - y\right] = q \cdot (q - 1) + (1 - q) \cdot (q - 0) = 0. \tag{1}$$

CDDM stores the most recent N values of q, y, and their corresponding referral documents x, in arrays $(q_1, q_2, \ldots, q_N), (y_1, y_2, \ldots, y_N)$, $(x_1, x_2, \ldots, x_N)$. q_i is the i-th most recent q value. At each time step, CDDM calculates the mean value of $q - y$ for the most recent T instances, for each $T = 1, 2, \ldots, N$, which we denote k_T. Using Equation 1 and Hoeffding's inequality, we can bound the probability of observing a mean as extreme as k_T under non-drift conditions:

$$P\left(\sum_{t=1}^{T} \frac{q_t - y_t}{T} \geq k_T\right) \leq \exp\left(-\frac{2T^2 k_T^2}{\sum_{t=1}^{T}(b_t - a_t)^2}\right) \tag{2}$$

where a_t and b_t are lower and upper bounds on $q_t - y_t$. Because $0 \leq q_t, y_t \leq 1$, we have that $a_t = -1$ and $b_t = 1$, so the right side of the equation is equal to

$$\exp\left(-\frac{2T^2 k_T^2}{\sum_{t=1}^{T} 4}\right) = \exp\left(-\frac{Tk_T^2}{2}\right). \tag{3}$$

When the P-value given in Equation 2 falls below a critical threshold, we conclude that $\mathbb{E}[q - y] > 0$ for the most recent T instances, and the model is making incorrect predictions at a higher rate than expected.

Similar to several other drift detectors [1], we use two critical thresholds: α_{warn}, a warning threshold, and α_{drift}, a drift threshold. When P falls below α_{warn} for any T value, CDDM issues a warning that drift may be occurring. When P falls below α_{warn} for any T, CDDM signals that drift has been detected, and the referral documents since the earliest T for which $P < \alpha_{warn}$ can be retrieved to retrain the model. We use the values $\alpha_{warn} = 0.05$ and $\alpha_{drift} = 0.01$, but to correct for multiple comparisons across N values of T, a Bonferroni correction is applied to these thresholds. The pseudocode for CDDM is given in Algorithm 1.

3 EXPERIMENTS

We present three sets of experiment results to evaluate CDDM.[1] The first is intended to validate that CDDM is useful for differentiating real and virtual drift, which will be useful for triage drift detection. The second set of experiments explores the wider applicability of CDDM by testing it on several benchmark datasets. The third set of experiments evaluates several drift detectors on synthetic medical triage data.

Two common base learners in concept drift detection experiments are Naïve Bayes and perceptron learners. However, Naïve

[1] https://github.com/lajesticvantrashell/CDDM_SIGIR2020.git.

Algorithm 1 CDDM

Require: Warning threshold α_{warn}
Require: Drift threshold α_{drift}
Require: Window size N
  Window $\leftarrow$ []
  for $t = 0, 1, 2, \ldots$ **do**
    $(q_1, q_2, \ldots, q_n) \leftarrow$model.predict(referral$_t$)
    $q \leftarrow \max_i q_i$
    label$_{pred} \leftarrow \arg\max_i q_i$
    $y \leftarrow \mathbb{1}$[label$_t$ =label$_{pred}$]
    Window $\leftarrow$ Window $\cup (y - q)$
    if Window.length $> N$ **then**
      Window $\leftarrow$ Window[1:]
    end if
    for $T = 1, 2, \ldots, N$ **do**
      $k_T \leftarrow \sum_{i=1}^{T}$ Window$[N - i]$
      $p \leftarrow \exp\left(\frac{Tk_T^2}{2}\right)$
      if $p \leq \alpha_{drift}/N$ **then**
        `status` $\leftarrow$ `drift`
      else if $p \leq \alpha_{warn}/N$ **then**
        `status` $\leftarrow$ `warning`
      end if
    end for
  end for

Bayes' are known to be poorly calibrated [6] and so do not work well with CDDM. We therefore report on perceptron experiments, but Naïve Bayes experiments are also provided in the supplementary material. Due to space constraints we only compare CDDM with two other drift detectors. In a large scale evaluation of drift detectors [1], the HDDM-A [3] and RDDM [2] algorithms obtained the best average performance across all metrics and datasets, so we use these as our state of the art comparisons. We use the Tornado framework for implementations of detection algorithms and benchmark dataset generators [7].

3.0.1 Real and Virtual Drift. We now present a data stream intended to illustrate how concept drift is detected using calibrated probability estimates for medical referrals triage. This data stream embodies several kinds of drifts which may occur and for which an error-rate based drift detector will be ill-equipped to handle. In particular, it involves feature drift and noise which is unevenly distributed in feature space. These factors can result in false positives and false negatives for a error-rate based drift detector.

The data stream consists of a single Bernoulli variable x, and a binary label y. Initially, the value of x is distributed as $P(x = 1) = p$, and the value of y is given by $y = x$. However, noise is present in the $x = 1$ region of feature space, so that with probability ϵ the label will be negated. That is,

$$P(y = 1|x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 - \epsilon & \text{if } x = 1. \end{cases} \quad (4)$$

The irreducible error rate is thus $p\epsilon$. At time τ, the data stream will drift in one of two ways. The first is a feature drift, in which $P(x = 1)$ becomes $1 - p$. If $\epsilon(2 - p) > 0$, then the irreducible error rate will increase, thereby triggering a false positive in an error-rate based drift detector.

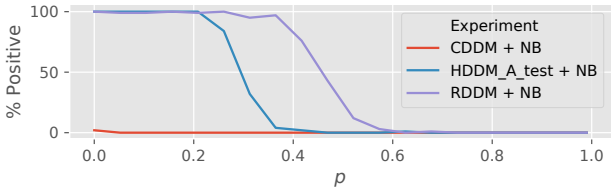


Figure 3: Percentage of (false) positive drift detections for the Bernoulli data stream with virtual drift.

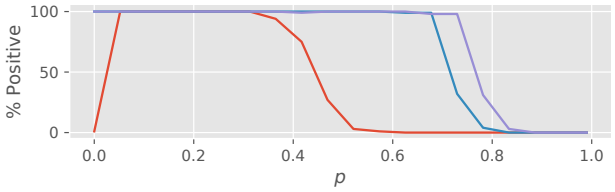


Figure 4: Percentage of (true) positive drift detection for the Bernoulli data stream with real drift.

The second is a feature drift (with the same details as above) in addition to a real drift so that $x = 1$, the correct label is now $y = 0$ (still with a noise rate of ϵ). In this case, if $1 < \epsilon + p$ then the change in irreducible error rate will *decrease*, resulting in a false negative for a drift detector.

We evaluated how well drift detectors can differentiate between these two scenarios by fixing the noise level at $\epsilon = 0.2$, and for given p values between 0 and 1, running 1000 time steps in the initial concept, and then 1000 time steps in the drifted concept. This is repeated 100 times for each value of p for both the real and virtual drift conditions. The percentage of trials which result in a positive drift detection for each detector are shown in Figures 3 and 4. We see that CDDM can reliably avoid false positives from virtual drift, unlike HDDM-A and RDDM, although this does come at the cost of lower sensitivity to true positives.

3.0.2 Benchmark Datasets. We compare CDDM with state of the art drift detection on standard benchmarks from the concept drift literature. The details for each data stream can be found at [1]. We used default parameters from Tornado [7]. For each combination of data stream, drift detector, and base learner, 10 trials were run. Each trial contained 80,000 time steps, with concept drifts every 20,000 timesteps. For each trial, the precision, recall, memory consumption, runtime, and total detection delay are recorded, although due to space constraints not all of these metrics are reported and may be found in the supplementary material. The mean results across trials and their standard deviations are given in Table 1.

We ran a Nemenyi post-hoc test to compare the detection methods. CDDM had a substantially longer runtime than HDDM-A and RDDM (p <0.001). It consumed more memory than HDDM-A, but less than RDDM (p <0.01). However, CDDM had higher precision

Table 1: Drift detector results on standard benchmarks.

	Memory (bytes)	Runtime (ms)	Precision	Recall	Mean Delay (time steps)
MIXED+CDDM	15.76 (0.01)	13234.50 (163.42)	1.00 (0.00)	1.00 (0.00)	73.12 (6.13)
MIXED+RDDM	73.10 (0.00)	252.06 (14.05)	0.91 (0.12)	1.00 (0.00)	121.50 (16.82)
MIXED+HDDM$_A$	10.49 (0.00)	383.95 (11.66)	0.91 (0.12)	1.00 (0.00)	87.58 (17.91)
SINE+CDDM	13.96 (0.00)	13216.29 (84.73)	1.00 (0.00)	1.00 (0.00)	65.25 (4.06)
SINE+RDDM	71.29 (0.00)	227.60 (0.72)	1.00 (0.00)	1.00 (0.00)	99.58 (4.10)
SINE+HDDM$_A$	8.68 (0.00)	347.98 (13.02)	0.94 (0.09)	1.00 (0.00)	59.02 (11.98)
LED+CDDM	77.54 (0.00)	47030.06 (12875.64)	0.00 (0.00)	0.00 (0.00)	250.00 (0.00)
LED+RDDM	136.76 (0.04)	2329.73 (120.75)	0.00 (0.00)	0.00 (0.00)	250.00 (0.00)
LED+HDDM$_A$	74.19 (0.05)	2489.91 (103.31)	0.33 (0.25)	0.37 (0.31)	226.80 (28.74)
CIRCLES+CDDM	13.93 (0.00)	57066.05 (2814.82)	0.00 (0.00)	0.00 (0.00)	250.00 (0.00)
CIRCLES+RDDM	71.26 (0.00)	424.91 (175.00)	0.04 (0.09)	0.07 (0.13)	246.03 (7.98)
CIRCLES+HDDM$_A$	8.68 (0.00)	360.27 (87.74)	0.24 (0.07)	0.43 (0.15)	208.37 (22.72)

Table 2: Drift detection experiments on synthetic medical triage data.

	Memory (bytes)	Runtime (ms)	Precision	Recall	Mean Delay (time steps)
REFERRALS+CDDM	3793.13 (0.02)	10462.15 (1766.30)	0.93 (0.14)	0.93 (0.11)	58.88 (23.46)
REFERRALS+RDDM	3850.19 (0.05)	14304.45 (5188.00)	1.00 (0.00)	0.95 (0.15)	55.73 (44.76)
REFERRALS+HDDM$_A$	3788.70 (0.07)	12737.93 (1678.21)	0.96 (0.08)	0.97 (0.07)	18.30 (18.11)

than RDDM (p <0.05). CDDM appears competitive on a range of benchmark datasets, although further optimisation work would be useful.

3.0.3 Referrals Data. For privacy reasons, we evaluated the drift detection methods on simulated referrals data. The synthetic dataset is based on 10,000 radiology medical reports from the MIMIC-III dataset [4]. We parse the document headers to extract structured fields like gender and age, and encode the report body as a bag of words. Stop words and words which occur fewer than 10 times are discarded.

We simulate triage policies using decision trees which have been trained on a randomly labelled sample of 20 documents. For each experimental trial, we shuffle the dataset to obtain a new data stream, and insert drift points at each 2,000 time steps by changing the triage policy. The results are shown in Table 2.

A Nemenyi post-hoc test on these results finds CDDM was competitive in runtime for this experiment, with lower runtime than RDDM ($p < 0.05$). As in the previous experiment, CDDM consumed more memory than HDDM-A, but less than RDDM ($p < 0.05$). CDDM had a significantly longer detection delay than HDDM-A ($p < 0.001$), but was otherwise competitive.

4 CONCLUSION AND FUTURE WORK

In this paper we motivated the task of referrals triage drift detection, and suggested some features of a good solution. We introduced a novel drift detection method for this task. CDDM makes use of calibration to differentiate between real and virtual drift, which will be useful for any domain in which virtual drift is common. Experiments with CDDM on synthetic data indicates it lags in computational efficiency, although this is not essential for medical triage drift detection. On other metrics CDDM appears competitive. It appears to be much less prone to false positives due to feature drift than other detectors.

For future work, we intend to develop a full solution for medical triage concept drift detection. This will include separate detectors for virtual *and* real drift detectors, and a graphical interface to increase interpretability and facilitate a more interactive style of triage machine learning.

ACKNOWLEDGMENTS

This research was supported by the Precision Driven Health Partnership (www.precisiondrivenhealth.com).

REFERENCES

[1] Roberto Barros and Silas Santos. 2018. A Large-scale Comparison of Concept Drift Detectors. *Information Sciences* 451-452 (07 2018). https://doi.org/10.1016/j.ins.2018.04.014

[2] Roberto SM Barros, Danilo RL Cabral, Paulo M Gonçalves Jr, and Silas GTC Santos. 2017. RDDM: Reactive drift detection method. *Expert Systems with Applications* 90 (2017), 344–355.

[3] Isvani Frías-Blanco, José del Campo-Ávila, Gonzalo Ramos-Jimenez, Rafael Morales-Bueno, Agustín Ortiz-Díaz, and Yailé Caballero-Mota. 2014. Online and non-parametric drift detection methods based on Hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2014), 810–823.

[4] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.

[5] Meelis Kull, Telmo M Silva Filho, Peter Flach, et al. 2017. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics* 11, 2 (2017), 5052–5080.

[6] Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 625–632.

[7] Ali Pesaranghader. 2018. *A Reservoir of Adaptive Algorithms for Online Learning from Evolving Data Streams.* Ph.D. Dissertation. University of Ottawa.