Concept Drift Adaptation for Acoustic Scene Classifier Based on Gaussian Mixture Model

Ibnu Daqiqil Id
岡山大学 阿部研究室
Abe Laboratory, Okayama University

Abstract. In the real-world, data are non-stationary, which means that the probability distribution that generates data samples is time-varying. This phenomenon is known as concept drift. The ability to detect and adapt to concept drift in order to prevent degradation in accuracy is essential. In this research, we propose two algorithms to detect and adapt the concept drift to detect environment scene, namely Kernel Density Drift Detection (KD3) and Combine-Merge Gaussian Mixture Model (CMGMM). KD3 is based on a variational comparison of two kernel density estimation of current and previous windows, and CMGMM is an algorithm for adapt to concept drift is based on combining and merging the existing and new components Gaussian mixture model. In the experiment, our methods show better performance and can detect concept drift then adapt to the change, so the model accuracy is relatively stable.

1 Introduction

Most machine learning implementation focuses on static building models, i.e., a model that is trained in the training dataset and then is applied to test the data. This works well when the underlying data distribution that is modeled remains stationary. However, in many real-world applications, the process is not stationary and continues to change over time. For example, scene sound from the park is changing over time depends on the season where in summer, we can hear the insect sound as background noises.

An acoustic scene is defined as a concept that humans commonly use to identify an acoustic environment that contains an ensemble of background noise and sound events in a particular scenario[1]. In the real world, each scene class might contain many types of event sounds and background noises, depending on the location and time. Moreover, the sound can be corrupted by non-stationary noise, diverse sound events, overlapping audio events in time or frequency domain, might have echoes or reverberant operating conditions. The above situation might exist simultaneously, which increases the complexity of the task in a combinatorial way [7] and change the data distribution. Besides, it is difficult to collect all those possible sound as the dataset for model training.

One solution to maintain models with the non-stationary environment is to retrain and redeploy the models. However, this process can be time-consuming, difficult to select the frequency of updates, and expensive to retrain models. Another promising approach is using an incremental or evolving learning paradigm [2][3]. Under incremental or evolving learning, model iteratively updated upon newly arrived data [4]. Each iteration is denoted as an incremental clustering step to update the current model, and every step only processes the newly incoming subset and obtains data concepts based upon these data.

In this paper, we propose a Combine-Merge Gaussian Mixture Model (CMGMM). The motivation of the proposed algorithm is to develop and implement an incremental audio scene classifier that works in a real-life scenario where the event sounds and background noises in the scene might change. There have been some previous research works on the evolving GMM algorithm. In [5], propose an IGMM (Incremental Gaussian Mixture Model) clustering algorithm by incremental density approximation from the observed samples, [6] propose IGMM algorithm based on Robbins–Monro stochastic approximation. Also, [7] apply IGMM by the expectation-maximization algorithm and a cluster merging strategy using multivariate statistical tests for equality of covariance and mean. Most of these works use a passive approach where adaptation is cyclically performed to handle non-stationary data.

On the contrary, CMGMM employs an active approach where model adaptation is performed only when concept drift is detected. Another feature of the CMGMM is to take into account distributions of pre-trained scene classifier as well because basic scene classes are not drastically changed over time. For example, beach scenes contain typical sounds of beaches, and the sound is hardly changed, but only beach-related sounds are added.

2 Proposed Method

2.1 Kernel Density Drift Detection

Kernel Density Drift Detection (KD3) works based on the kernel density estimation (KDE) method [8] to estimate the probability density function with one random variable. By comparing two density functions, it is possible to establish the degree of variation in values between the corresponding variables—the greater the variation, the more evidence for the concept drifts. In addition to detecting concept drifts, KD3 also acts as a data collector for the adaptation process by marking a warning zone where data begin to show symptoms of concept drift.

KD3 requires four input parameters: z, α, β, and h. z denotes a set of likelihood windows, $z = \{z_1, z_2, z_3, \dots, z_c\}$, where z_c is the current likelihood window that contains a sequence of log-likelihood ℓ from the model prediction, $z_c = \{\ell_1, \ell_2, \ell_3, \dots, \ell_h\}$. h denotes the window length, α denotes the margin for detecting a concept drift, and β denotes the margin for accumulating the density distance.

KD3 works by comparing two windows, namely, the current window z_c and the previous window z_{c-1}. Let ℓ_n be the latest generated ℓ, and z_c contain h latest ℓ from ℓ_n, $z_c \in [\ell_{n-h}, \ell_n]$ and z_{c-1} contain $h$latest ℓ from ℓ_{n-h}, $z_{c-1} \in [\ell_{n-2h}, \ell_{n-h}]$. The first step of the algorithm is to compute the Gaussian kernel density function (f_{kde}) of z_c and z_{c-1}. To detect a concept drift, the distance $\dot{d}_t$ between f_{kde} of z_c and z_{c-1} is computed using Eq. 1 within the bounds of $b1$ and $b2$. The bounds are computed based on the maximum and minimum values of the joined ℓ of z_c and

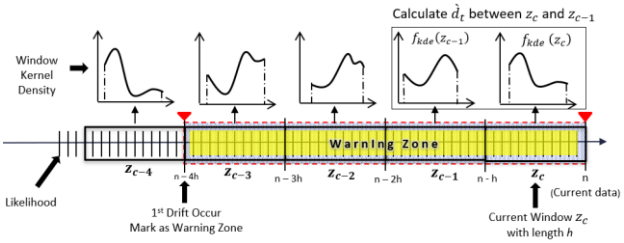

Figure 1. Kernel density drift detection

z_{c-1}.

$$\dot{d}_t = \frac{1}{2} \int_{b1}^{b2} |f_{kde}(z_1) - f_{kde}(z_2)| \, dz \tag{1}$$

2.2 Combine-Merge Gaussian Mixture Model

CMGMM is developed based on a GMM defined by a set of parameters π_j, μ_j, and Σ_j denoting the prior probability, distribution means, and covariance matrix, respectively.

The model adaptation process begins by creating a new model $\mathcal{M}_{\text{drift}}$ from data drifts D_{drift} using Algorithm 1, and then combining the existing model $\mathcal{M}_{\text{curr}}$ to form a new model $\mathcal{M}_{\text{all}}$. $\mathcal{M}_{\text{all}}$ contains all components from both $\mathcal{M}_{\text{curr}}$ and $\mathcal{M}_{\text{drift}}$. The next step is to calculate the pairwise distance between components in $\mathcal{M}_{\text{all}}$ using Kullback–Leibler (KL) discrimination presented in [9]. The KL discrimination formula (Eq. 5) enables us to put an upper bound on the discrimination of the mixture before and after the merging process. According to this formula the following components are selected for merging: components with low weights, components with means being close to their variances, and components with similar covariance matrices. When two components are merged, the moment-preserving merging method [10] is used to preserve the mean and covariance of the overall mixture (Figure 1).

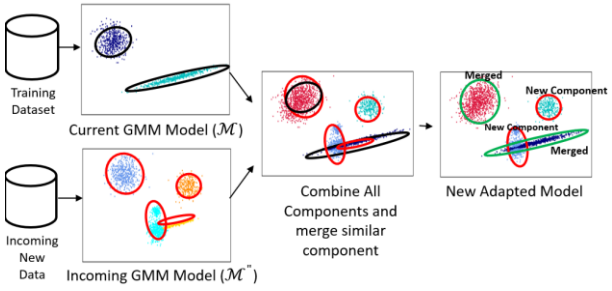

Figure 2. Process of the proposed CMGMM-based algorithm

Finally, the algorithm iteratively merges the components in $\mathcal{M}_{all}$ ranging from n_{CompMin} to n_{CompMax} to obtain the best model $\mathcal{M}_{\text{best}}$, where n_{CompMin} and n_{CompMax} are the smallest and largest numbers of components derived from $\mathcal{M}_{\text{curr}}$ and $\mathcal{M}_{\text{all}}$, respectively, to be included into $\mathcal{M}_{\text{best}}$. For each iteration, the algorithm finds the least similar components. The merging process follows the rules set by Eqs. 2-4

$$w_{ij} = w_i + w_j \tag{2}$$

$$\mu_{ij} = \frac{w_i}{w_{ij}}\mu_i + \frac{w_j}{w_{ij}}\mu_j \tag{3}$$

$$\Sigma_{ij} = \frac{w_i}{w_{ij}}\Sigma_i + \frac{w_j}{w_{ij}}\Sigma_j + \frac{w_i w_j}{{w_{ij}}^2}(\mu_i - \mu_j)(\mu_i - \mu_j)^T \tag{4}$$

$$d_{kl}\left((w_i, \mu_i, \Sigma_i), (w_j, \mu_j, \Sigma_j)\right) = \frac{1}{2}\left[w_{ij} \log(\det(\Sigma_{ij})) - w_i \log(\det(\Sigma_i)) - w_j \log(\det(\Sigma_j))\right] \tag{5}$$

3 Experiment

For training and evaluations, we generated dataset by selecting 15 kinds of scenes from TUT Acoustic scenes 2017 dataset [11] and TAU Urban Acoustic Scenes 2019 dataset [12] dataset. The training dataset consists of 12000 audio segments of ten seconds, equally distributed between 15 different scenes, and of the annotated ground truth. Then we split randomly the dataset into a training dataset (9000 segments) dan test dataset (3000 segments). The 15 scenes are beach, bus, cafe/restaurant, car, city center, forest path, grocery store, home, library, metro station, office, park, residential area, train, and tram.

To confirm the performance, an artificially generated drifted dataset that contains concept drift scenarios, namely T1, T2, and T3, are used to test how good the proposed method detects and adapts. The number of generated sounds for each scenario is 12000 segments. Furthermore, we also add the test dataset into the drifted dataset, so in total, the number of scene segments is 15000 for each scenario.

The concept drift is simulated by overlaying novel event sounds to scene sound with each scenario, as shown in Figure 3. The novel event sounds are taken from event sound dataset from BBC Sound Class Library, UrbanSound, and UrbanSound8K datasets randomly, on random position and random gain. There are 46 classes and 371 new event sound that will add to the scene based on three scenarios, namely :

- **T1 Dataset**. In T1, several types of unique event sounds related to each scene were overlaid several times with random timing and gains. For example, for the beach scene, sounds representing a dog barking, people talking, people swimming, and raining were overlaid with random timing and gain. None of these sounds were included in the initial training dataset. As a result, the prior distribution of the test data changed, representing the added concept drift.

- **T2 Dataset**. In T2, several sounds were randomly selected from a particular group of event sounds, making the concept drift of this scenario to be more complicated than that of scenario T1. For example, several types of dog-barking sounds from the dog group were added to the beach scene. Similar to T1, the included sounds were related to the respective scene and were overlaid using random timing and gain; none of the added sounds were included in the initial training dataset.
- **T3 Dataset**. T3 was quite similar to T2. The difference was that a group of event sounds could exist in several scenes (co-existing event classes between scenes). For example, the group of dog-barking sounds could exist in three scenes, e.g., the beach, forest path, and city center scenes. As a result, the prior and posterior distribution of the test data changed.

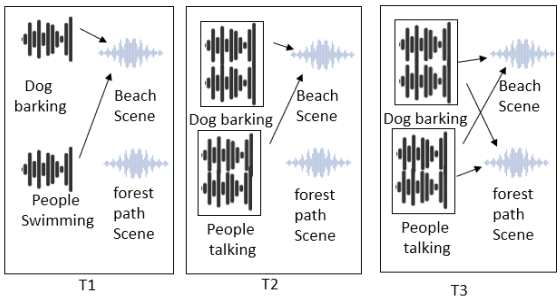

Figure 3. Illustration of the three test scenarios

The effectiveness of the proposed CMGMM was demonstrated by comparing it to the IGMM [13]. The models were tested under the earlier introduced three scenarios (T1, T2, and T3) and the following two approaches, namely, active and passive.

- **Active approach**. This approach assumes that the models detect concept drifts using a concept drift detector and make adaptations upon the detected drifts. In this study, we compared the proposed drift detection method KD3 to the adaptive windowing (ADWIN) method [14] and drift detection method based on the Hoeffding's inequality (HDDM) [15].
- **Passive approach**. This approach assumes that the models adapt continuously as soon as new incoming data are received. Several adaption cycles were tested, namely, 50,100, 150, and 200.

4 Result

Experimental results are shown in Table 1, both active and passive approaches. In terms of the active approach, results are shown for all combinations of detector methods and adaptation methods and passive approach in every cycle of the adaptation process.

Judging from overall accuracy and F1 Score, we can say that the active approach is promising because, in all scenarios, the best performance was achieved by the active approach. Moreover, KD3 is showed better performance than ADWIN except for T1. This is mainly because KD3 precisely takes distributions into account, while ADWIN uses the only average of the distribution. A drawback of KD3 is its high computational cost. On the other hand, HDDM showed extremely worse performance than all the others. The reason is that, as shown in Table 1, HDDM detected the drift too much, say more than 350 times. According to all the above results, the detector of the concept drift plays an important role in the concept drift adaptation.

In terms of the overall accuracy, CMGMM shows better performance than IGMM except for T1. Different from CMGMM, IGMM has the parameter of the maximum number of Gaussian components for trained distribution. Therefore, when a concept drift is simple as T1, IGMM showed better performance than

TABLE I. Experiment Result in Active and Passive Scenario

ACTIVE APPROACH											
ADAPTOR	**DETECTOR**	**ACCURACY**			**F1 SCORE**			**EXECUTION TIME**			**Drift**
		T1	**T2**	**T3**	**T1**	**T2**	**T3**	**T1**	**T2**	**T3**	
CMGMM*	KD3*	0.8373	**0.7962**	**0.7409**	0.8432	**0.7993**	**0.7460**	128.06	115.07	110.49	39
	ADWIN	0.8471	0.6415	0.6332	0.8518	0.6379	0.6379	83.07	84.22	85.44	23
	HDDM	0.2762	0.2627	0.2990	0.3184	0.2992	0.3406	84.81	84.53	83.11	373
IGMM	KD3*	0.8283	0.7574	0.6622	0.8173	0.7499	0.6488	120.04	128.75	120.50	35
	ADWIN	**0.8419**	0.5711	0.6057	0.8329	0.5722	0.6063	82.80	84.219	83.08	21
	HDDM	0.2363	0.2507	0.2032	0.2436	0.3055	0.2675	84.37	87.55	84.87	350
PASSIVE APPROACH											
ADAPTOR	**ADAPTATION CYCLE**	**ACCURACY**			**F1 SCORE**			**EXECUTION TIME**			**Drift**
		T1	**T2**	**T3**	**T1**	**T2**	**T3**	**T1**	**T2**	**T3**	
CMGMM*	50	0.5621	0.4451	0.4003	0.5719	0.4615	0.4373	83.178	82.945	83.163	-
	100	0.7424	0.6547	0.6434	0.7451	0.6583	0.6476	83.688	82.265	83.704	-
	150	0.8002	**0.7437**	**0.7301**	0.8043	**0.7482**	**0.7327**	84.527	82.298	89.555	-
	200	0.7602	0.7073	0.6904	0.7663	0.714	0.7001	83.414	85.922	84.594	-
IGMM	50	0.5365	0.4209	0.3687	0.5458	0.4319	0.3888	84.467	82.776	82.655	-
	100	0.7324	0.643	0.6371	0.736	0.6448	0.6388	82.693	82.652	82.199	-
	150	**0.8056**	0.7401	0.7291	**0.8107**	0.7431	0.7223	83.659	82.645	83.453	-
	200	0.7528	0.7149	0.6899	0.7609	0.722	0.6981	84.597	84.228	85.797	-

(*) Proposed Method

CMGMM. However, when the concept drift has occurred with a complicated combination of event sounds, CMGMM outperformed IGMM, because CMGMM allows flexible adaptation with combination and merge mechanism and has no limitation for the number of Gaussian components

5 Conclusion

In this paper, we proposed the CMGMM algorithm for adapting the concept drift together with the KD3 algorithm to detect concept drift based on log-likelihood. Judging from the experiment results, we can say that the proposed algorithm works well in detecting and adapting for the three kinds of drift scenarios. Moreover, by comparing the active approach and passive approach, we found that the number of data to adapt to the model is a crucial factor in having better accuracy. Although KD3 can successfully change the number of data to adapt to some extent, as future work, we would like to improve the concept drift detection algorithm to optimally conges the number of data.

6 Reference

[1] M. Valenti, S. Squartini, A. Diment, G. Parascandolo, and T. Virtanen, "A convolutional neural network approach for acoustic scene classification," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2017-May, pp. 1547–1554, 2017, doi: 10.1109/IJCNN.2017.7966035.

[2] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: An overview," *Prog. Artif. Intell.*, vol. 1, no. 1, pp. 89–101, 2012, doi: 10.1007/s13748-011-0008-0.

[3] I. Žliobaitė, "Learning under Concept Drift: an Overview," pp. 1–36, 2010, doi: 10.1002/sam.

[4] C. Chen *et al.*, "Improving Accuracy of Evolving GMM under GPGPU-Friendly Block-Evolutionary Pattern," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 3, pp. 1–34, 2020, doi: 10.1142/S0218001420500068.

[5] M. Kristan, D. Skocaj, and A. Leonardis, "Incremental learning with Gaussian mixture models," in *13th Computer Vision Winter Workshop (Slovenian Pattern Recognition Society, Moravske Toplice, Slovenia, 2008)*, 2008, pp. 25–32.

[6] P. M. Engel and M. R. Heinen, "Incremental learning of multivariate Gaussian mixture models," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6404 LNAI, pp. 82–91, 2010, doi: 10.1007/978-3-642-16138-4_9.

[7] M. Song and H. Wang, "Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering," *Intell. Comput. Theory Appl. III*, vol. 5803, p. 174, 2005, doi: 10.1117/12.601724.

[8] E. Parzen, "On the Estimation of Probability Density Functions and Mode," *Ann. Math. Stat.*, vol. 33, pp. 1065–1076, 1962.

[9] A. R. Runnalls, "Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 989–999, 2007, doi: 10.1109/TAES.2007.4383588.

[10] J. L. Williams and P. S. Maybeck, "Cost-function-based Gaussian mixture reduction for target tracking," *Proc. 6th Int. Conf. Inf. Fusion*, vol. 2, pp. 1047–1054, 2003, doi: 10.1109/ICIF.2003.177354.

[11] A. Mesaros, T. Heittola, and T. Virtanen, "TUT Acoustic scenes 2017, Development dataset," Mar. 2017. doi: 10.5281/ZENODO.400515.

[12] T. Heittola, A. Mesaros, and T. Virtanen, "TAU Urban Acoustic Scenes 2019, Development dataset," Mar. 2019. doi: 10.5281/ZENODO.2589280.

[13] J. M. Acevedo-Valle, K. Trejo, and C. Angulo, "Multivariate regression with incremental learning of Gaussian mixture models," *Front. Artif. Intell. Appl.*, vol. 300, pp. 196–205, 2017, doi: 10.3233/978-1-61499-806-8-196.

[14] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," *Proc. 7th SIAM Int. Conf. Data Min.*, no. April, pp. 443–448, 2007, doi: 10.1137/1.9781611972771.42.

[15] I. Frías-Blanco, J. Del Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, and Y. Caballero-Mota, "Online and non-parametric drift detection methods based on Hoeffding's bounds," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 810–823, 2015, doi: 10.1109/TKDE.2014.2345382.