Storage Fit Learning with Feature Evolvable Streams

Bo-Jian Hou, Yu-Hu Yan, Peng Zhao, Zhi-Hua Zhou

LAMDA Group, Nanjing University

Contact Information:
{houbj, yanyh, zhaop, zhouzh}@lamda.nju.edu.cn

AAAI 2021

LAMDA Learning And Mining from DatA http://www.lamda.nju.edu.cn

1. Contribution

This paper is to address a real world problem of feature evolvable streams when labels are rarely given. The contributions are as follows:

1. A novel learning paradigm, named Storage Fit Learning with Feature Evolvable Streams (SF^{2}EL), to model our problem.
2. A novel approach combining manifold regularization and reservoir sampling to solve SF^{2}EL.
3. Empirical evaluations on both synthetic and real data sets.
4. Theoretical guarantees on the performance and the storage-fit issues.

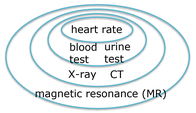
2. Background and Motivation

In open environment, **features often change or evolve** when learning with streaming data.

- In environment monitoring, we need to deploy light, sound, humidity sensors to gather data.
- The data collected by sensors of different locations and types represent different features.

When the old sensors (red circles) expire and are replaced with new sensors (yellow circles), the features will change because the number and location of the old and new sensors are different.

Example 1



Example 2

data used to describe patients

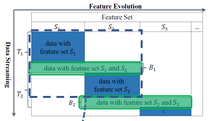
heart rate	blood test	CT			
heart rate	urine test	blood test			
heart rate	blood test	X-ray			
heart rate	urine test	blood test	X-ray	CT	MR
heart rate	urine test	blood test	X-ray	CT	
heart rate	MR				

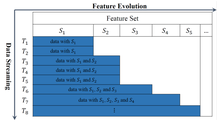
The feature of the data used to describe patients can change.

Challenge 1: The well-learned model in the old feature space is wasted.

Challenge 2: The data collection in the old feature space is wasted.

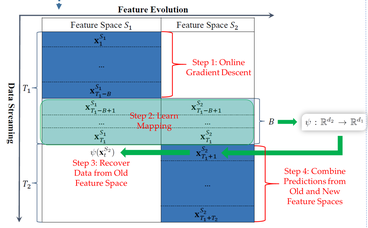
Challenge 3: The model in the new feature space cannot learn well.





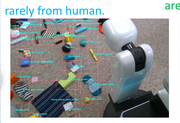
1. FESL [Hou et al., 2017a] assumes there is an instance overlapping period when feature space switches.
2. OPID [Hou and Zhou, 2017] assumes there is a feature overlapping when feature switches.
3. OL$_{SF}$ [Zhang et al., 2016] assumes features of new samples are always equal to or larger than the old samples.
4. In [He et al., 2019], features can vary arbitrarily but with the assumption that there is a universal feature space.

The goal of these methods is that the model in the new feature space is able to perform well at any time step with the help of old feature space.



We focus on the FESL setting, other feature evolvable learning methods can also adapt to our framework.

In object detecting system, the robot receives names rarely from human.



In ecosystem protecting system, the conditions of the environment are rarely provided by human.

Labels Are Rarely Given!

The conventional learning with feature evolvable streams methods cannot apply in the situation where labels are rarely given.

Framework

We first exploit manifold regularization to mitigate the problem where labels are rarely given, and then the online gradient descent can be calculated again;

Based on the modification in the first step, we can derive our learning procedure from FESL naturally, yet with a potential problem of storage and computation;

Finally, we use a buffering strategy to solve the storage and computation problem and subsequently solve the storage-fit problem based on this strategy.

3. Detailed Approach

- FESL adopts linear predictor, whereas to be general, nonlinear predictor is chosen in our paper.
- Let K_i denote a kernel over $\mathbf{x}^{S_i}$ and $\mathcal{H}_{K_i}$ the corresponding Reproducing Kernel Hilbert Space (RKHS).
- We define the projection as $\Pi_{\mathcal{H}_K}(b) = \operatorname{argmin}_{\mathbf{a}\in\mathcal{H}_K} \|a-b\|_K$.
- The predictor learned from the sequence is denoted as $f \in \mathcal{H}_K$.
- $\ell(f(\mathbf{x}), y)$ is convex in its first argument such as hinge loss or logistic loss.
- Then the online gradient descent in each feature space can be calculated as

$$f_{i,t+1} = \Pi_{\mathcal{H}_{K_i}}\left(f_{i,t} - \tau_t \nabla \ell(f_{i,t}(\mathbf{x}_t^{S_i}), y_t)\right),\ i = 1, 2$$

The label is rarely given! We cannot calculate ℓ at every time step.

- After using manifold regularization (MR), the instantaneous regularized risk J at time t will be

$$J_{i,t}(f_{i,t}) = \frac{T}{l}\delta(y_t)\ell(f_{i,t}(\mathbf{x}_t^{S_i}), y_t) + \frac{\lambda_1}{2}\|f_{i,t}\|_{K_i}^2 + \lambda_2 \sum_{s=1}^{t-1}(f_{i,t}(\mathbf{x}_s^{S_i}) - f_{i,t}(\mathbf{x}_t^{S_i}))^2 w_{st},\ i = 1, 2$$

indicator · loss · 2-norm regularizor · manifold regularizor

- The online gradient descent algorithm applied on the instantaneous regularized risk will derive

$$f_{i,t+1} = \Pi_{\mathcal{H}_{K_i}}\left(f_{i,t} - \tau_t \nabla J_{i,t}(f_{i,t}(\mathbf{x}_t^{S_i}))\right),\ i = 1, 2$$

We can calculate the risk J and update the model even without label!

- Leveraging MR needs to calculate the similarity between the current sample and all the previous samples w_{st}, and this needs to keep all the previous data

$$J_{i,t}(f_{i,t}) = \frac{T}{l}\delta(y_t)\ell(f_{i,t}(\mathbf{x}_t^{S_i}), y_t) + \frac{\lambda_1}{2}\|f_{i,t}\|_{K_i}^2 + \lambda_2 \sum_{s=1}^{t-1}(f_{i,t}(\mathbf{x}_s^{S_i}) - f_{i,t}(\mathbf{x}_t^{S_i}))^2 w_{st},\ i = 1, 2$$

- However, keeping all the previous data is not realistic in this big data era. Thus we need a buffer to store a part of them.

$$J_{i,t}(f_{i,t}(\mathbf{x}_t^{S_i})) = \frac{T}{l}\delta(y_t)\ell(f_{i,t}(\mathbf{x}_t^{S_i}), y_t) + \frac{\lambda_1}{2}\|f_{i,t}\|_{K_i}^2 + \lambda_2 \frac{t-1}{b}\sum_{\mathbf{x}_s^{S_i}\in B}(f_{i,t}(\mathbf{x}_s^{S_i}) - f_{i,t}(\mathbf{x}_t^{S_i}))^2 w_{st},\ i = 1, 2,$$

How to choose the samples in the buffer?

- Different devices may have different storage budgets. Even the same device may have different available storage in different time.

- It is important to fit our method to different storage situations (known as storage-fit problem) [Hou et al., 2017b] and optimize its performance.

We use reservoir sampling technique to choose the samples in the buffer. [Vitter, 1985]

Reservoir Sampling:

- Assume the buffer size is b. When receiving a sample $\mathbf{x}_t^{S_i}$, we will directly add it to the buffer if $b > t$.
- Otherwise, with probability $b = t$, we update the buffer by randomly replacing one sample in the buffer with $\mathbf{x}_t^{S_i}$.

Reservoir sampling technique guarantees that the samples in the buffer are provably sampled from the original dataset uniformly.

- In the beginning of the new feature space, the well-learned model from the old feature space will provide good prediction on the recovered data.
- As time goes on, the new model in the new feature space will become better and better while the model in the old feature space will become worse due to the compounding error brought by the inaccurate mapping.

Theorem 1. *Assume that the risk function J_t takes value in [0,1]. For all $T_2 > 1$ and for all $y_t \in \mathcal{Y}$ with $t = T_1 + 1, \ldots, T_1 + T_2$, $\mathcal{J}^{S_{12}}$ with parameter $\eta = \sqrt{\ln 2/T_2}$ satisfies*

$$\mathcal{J}^{S_{12}} \le \min(\mathcal{J}^{S_1}, \mathcal{J}^{S_2}) + \sqrt{T_2 \ln 2}. \qquad (21)$$

- This theorem demonstrates that our model is always comparable to the best baseline.

Our model can perform well at any time step in the new feature space regardless of the limitation that only few data emerge in the beginning.

Theorem 2. *With the reservoir sampling mechanism, the approximated objective is an unbiased estimation of objective formed by the original data, namely, $\mathbb{E}[R_t] = \mathbb{E}[\hat{R}_t]$.*

- The variance of the approximated objective will decrease with more observed samples in a larger buffer, leading to a more accurate approximation.
- This suggests us to make the best of the buffer storage to store previous observed samples.

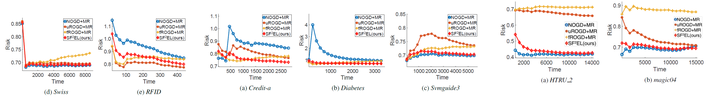
We can fit our method for different storages to optimize the performance by taking full advantage of the budget.

4. Experiments

Compared Methods:

- NOGD: the online gradient descent algorithm is invoked from scratch.
- ROGD-u: utilizes the classifier learned from feature space S_1 by online gradient descent to do predictions on the recovered data, keeps updating.
- ROGD-f: resembles ROGD-u, but do not update with recovered data.
- FESL-variant: FESL [Hou, et al., 2017a] cannot be directly applied in our setting. For fair comparison, we modify the original FESL to a non-linear version which only updates on the rounds when there is a label revealed.

E1, The Trend of Loss:



(d) Swiss (e) RFID (a) Credit-a (b) Diabetes (c) Svmguide3 (a) HTRU_2 (b) magic04

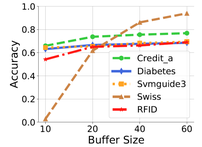
- Risk comparison, the less the better.
- Our methods can always follow the best baseline so as to perform well at any time step, which is the fundamental goal of learning with feature evolvable streams.

E2, The Accuracy Results:

Dataset	Credit-a	Diabetes	Svmguide3	Swiss	RFID	HTRU_2	magic04
NOGD	.690±.051	.643±.029	.657±.036	.711±.044	.687±.042	.885±.022	.580±.120
NOGD+MR	•.706±.049	•.672±.019	•.668±.037	•.807±.026	•.688±.042	•.907±.001	•.616±.058
uROGD	.740±.038	.658±.021	.675±.045	.694±.071	.571±.036	.943±.014	.550±.172
uROGD+MR	•.760±.034	•.678±.015	•.680±.048	•.824±.050	•.572±.036	**•.944±.010**	•.603±.079
fROGD	.672±.087	.633±.056	.648±.035	.702±.073	.560±.045	.757±.179	.550±.172
fROGD+MR	•.697±.079	•.654±.041	•.659±.037	•.811±.067	•.561±.045	•.943±.020	**•.649±.001**
FESL-Variant	.759±.028	.666±.018	.686±.039	.855±.034	.688±.040	.885±.022	.556±.162
SF^{2}EL (ours)	**•.768±.029**	**•.685±.011**	**•.694±.040**	**•.939±.020**	**•.690±.040**	•.912±.008	**•.649±.001**

- +MR means the baseline methods are boosted by manifold regularization.
- The experimental results show that MR can promote the performance.
- Our method yields the best results in 6 datasets out of 7.

E3, The Accuracy Tendency:



Buffer	Credit-a	Diabetes	Svmguide3	Swiss	RFID
10	.659±.052	.631±.066	.644±.103	.290±.070	.542±.056
20	.737±.036	.666±.025	.655±.064	.617±.067	.650±.059
40	.755±.039	.676±.016	.683±.034	.861±.031	.662±.054
60	**.768±.029**	**.685±.011**	**.694±.040**	**.939±.020**	**.690±.040**

- The accuracy tendency with the increasing of the buffer size.
- A larger buffer brings better performance.
- In order to make our model adjust its behavior to fit for different storage budgets, we should fully exploit the storage budget to optimize its performance.

5. Conclusion

- Learning with feature evolvable streams usually assumes label can be revealed immediately in each round. However, in reality this assumption may not hold.
- We introduce manifold regularization into FESL and let FESL can work well in this scenario. Other feature evolvable learning like FESL can also adapt to our framework.
- Both theoretical and experimental results validate that our method can follow the best baselines and thus work well at any time step.
- Besides, we theoretically and empirically demonstrate that a larger buffer can bring better performance and thus our method can fit for different storages by taking full advantage of the budget.