Learning with Feature Evolvable Streams

Bo-Jian Hou, Lijun Zhang, Zhi-Hua Zhou

LAMDA Group, Nanjing University

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

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

1. Contribution

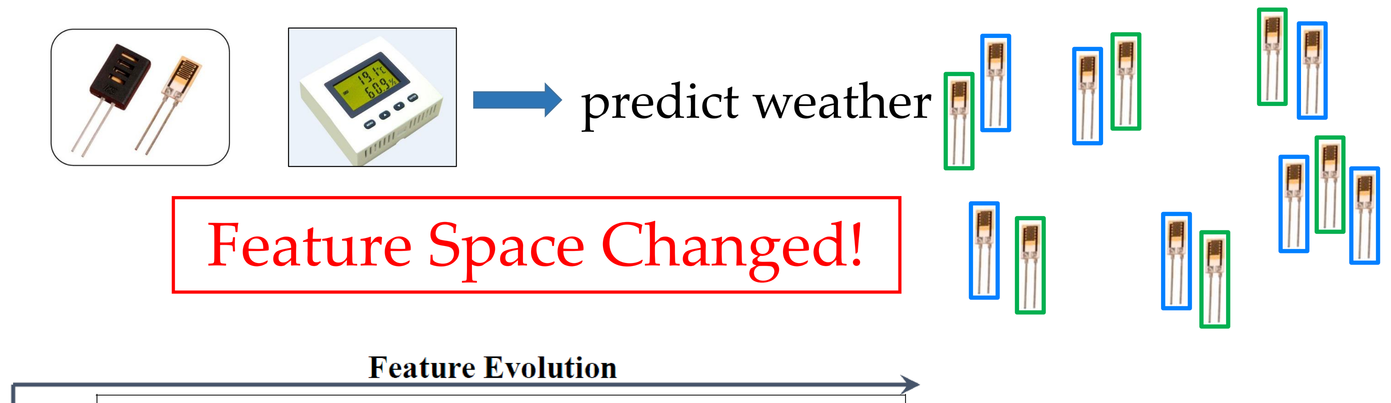
The target of this paper is to address a real-world problem of feature evolvable streams.

The contributions of this paper include:

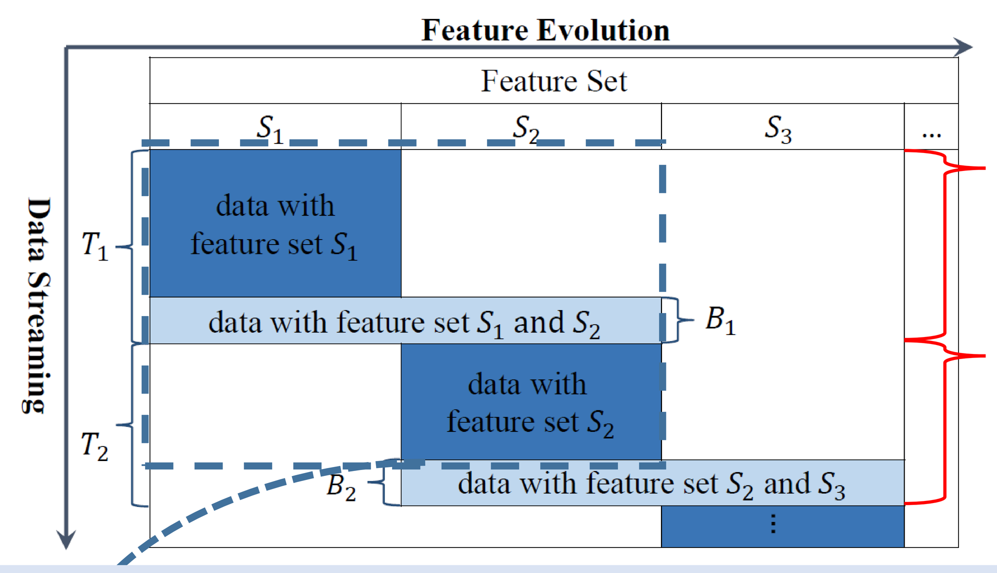
1. A novel learning paradigm, named Feature Evolvable Streaming Learning (FESL), to model our problem;
2. A novel application of online learning techniques to solve FESL;
3. Empirical evaluations on both synthetic and real data sets.

2. Motivation

Common algorithms learned from streaming data assume an unchanged feature space. Unfortunately, this assumption does not hold in many streaming tasks.



1. Green sensors are spread first;
2. Blue sensors are spread before the green ones expire;
3. Green sensors expire, left blue ones alone.

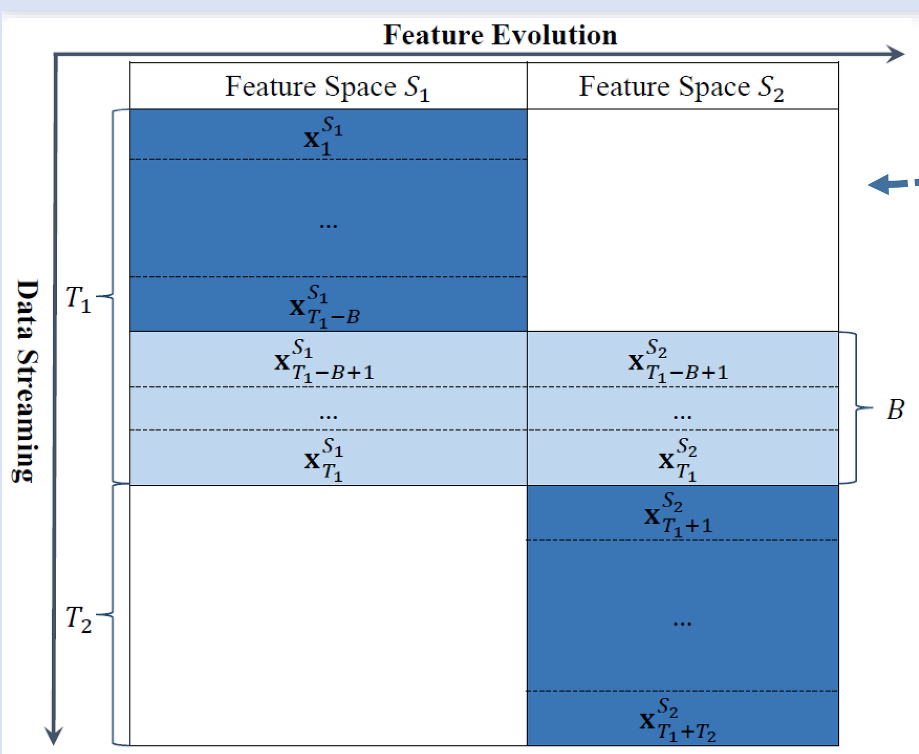


In period T_1, S_1 is valid. At the end of T_1, period B_1 appears, where S_1 is still accessible, but S_2 is included;

Then in T_2, S_1 vanishes, only S_2 is valid but at the end of T_2, period B_2 appears where S_3 comes.

This process will repeat again and again.

3. Preliminaries



We only need to focus on one cycle and it is easy to extend to the case with multiple cycles.

- For $t = 1, \dots, T_1 - B$, the learner observes $x_t^{S_1} \in \mathbb{R}^{d_1}$ sampled from S_1;
- For $t = T_1 - B + 1, \dots, T_1$, the learner observes both $x_t^{S_1} \in \mathbb{R}^{d_1}$ and $x_t^{S_2} \in \mathbb{R}^{d_2}$ from S_1 and S_2, respectively;
- For $t = T_1 + 1, \dots, T_1 + T_2$, the learner observes $x_t^{S_2} \in \mathbb{R}^{d_2}$ sampled from S_2;
- Note that B is small, so we can omit the streaming data from S_2 on rounds $T_1 - B + 1, \dots, T_1$ since they have minor effect on training the model in S_2.

Algorithm 1 Baseline

```
Initialize w_{1,1} ∈ Ω_1 randomly;
for t = 1, 2, ..., T_1 do
    Receive x_t^{S_1} ∈ R^{d_1};
    Predict f_t = w_{1,t}^T x_t^{S_1} ∈ R;
    Receive the target y_t ∈ R, and suffer loss ℓ(f_t, y_t);
    Update model using (1) where τ_t = 1/√t;
end for
Initialize w_{2,T_1+1} ∈ Ω_2 randomly;
for t = T_1 + 1, T_1 + 2, ..., T_1 + T_2 do
    Receive x_t^{S_2} ∈ R^{d_2};
    Predict f_t = w_{2,t}^T x_t^{S_2} ∈ R;
    Receive the target y_t ∈ R, and suffer loss ℓ(f_t, y_t);
    Update model using (1) where τ_t = 1/√(t − T_1);
end for
```

The baseline is to apply online gradient descent on every feature space.

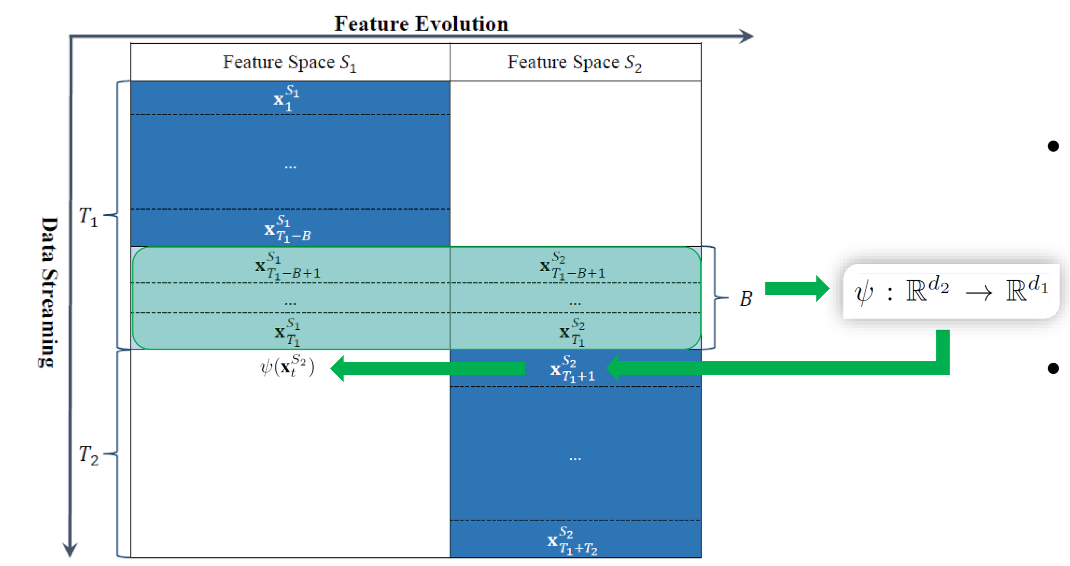
$$\mathbf{w}_{i,t+1} = \Pi_{\Omega_i}\left(\mathbf{w}_{i,t} - \tau_t \nabla \ell(\mathbf{w}_{i,t}^\top \mathbf{x}_t^{S_i}, y_t)\right), \; i = 1, 2, \quad (1)$$

However, the baseline suffers from two main deficiencies.

First, when new features just emerge, there are few data samples and thus, the training samples might be insufficient to train a strong model.
Second, the old model of vanished features is ignored, which is a big waste of our data collection effort.

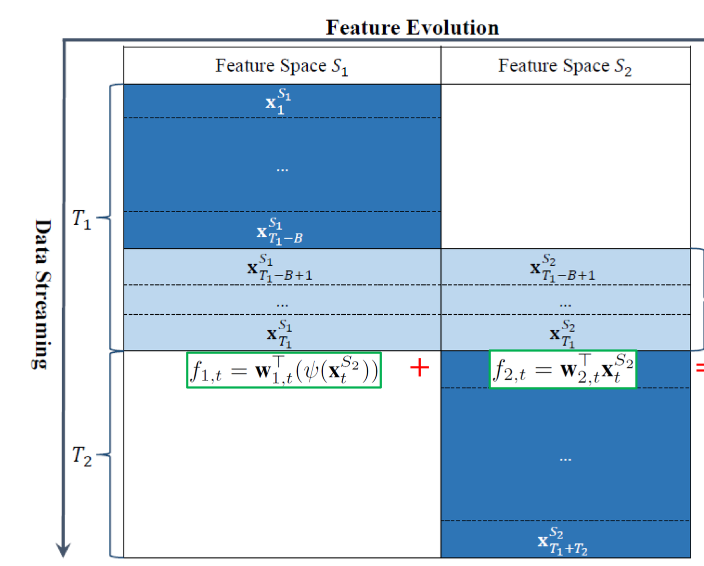
4. The FESL Approach

Frame:



- We assume there is a certain relationship $\psi: \mathbb{R}^{d_2} \to \mathbb{R}^{d_1}$ between the two feature spaces.
- We try to discover ψ in the overlapping period and recover data in S_1 such that the model w_{1,T_1}, which operates in S1, can be used.
- Then we propose two methods:
  1. **FESL-c**, combining predictions with weights.
  2. **FESL-s**, dynamically selecting the best one in each round.

1. FESL-c(ombination):



$$L^{S_1} = \sum_{t=T_1+1}^{T_1+T_2} \ell(f_{1,t}, y_t), \; L^{S_2} = \sum_{t=T_1+1}^{T_1+T_2} \ell(f_{2,t}, y_t), \; L^{S_{12}} = \sum_{t=T_1+1}^{T_1+T_2} \ell(\widehat{p}_t, y_t).$$

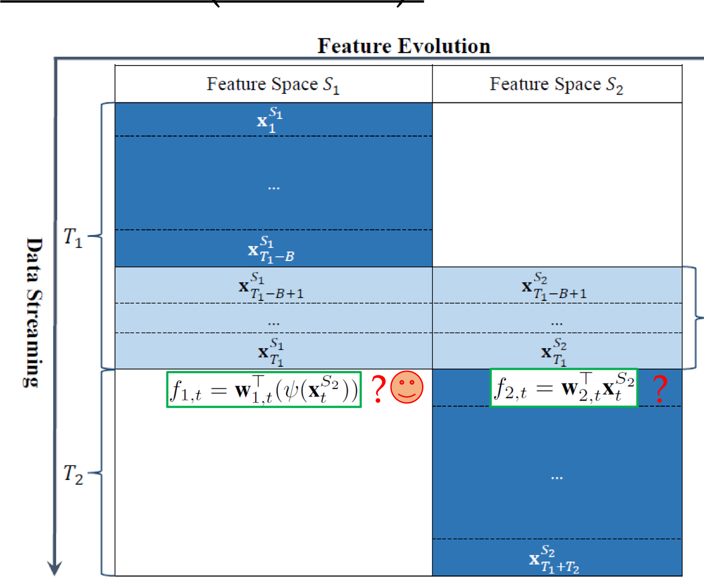
$$\alpha_{i,t+1} = \frac{\alpha_{i,t} e^{-\eta \ell(f_{i,t}, y_t)}}{\sum_{j=1}^{2} \alpha_{j,t} e^{-\eta \ell(f_{j,t}, y_t)}}, \; i = 1, 2 \quad (3)$$

$$\widehat{p}_t = \alpha_{1,t} f_{1,t} + \alpha_{2,t} f_{2,t} \quad (2)$$

Theorem 1. *Assume that the loss function ℓ is convex in its first argument and that it takes value in [0,1]. For all $T_2 > 1$ and for all $y_t \in \mathcal{Y}$ with $t = T_1 + 1, \dots, T_1 + T_2$, $L^{S_{12}}$ with parameter $\eta_t = \sqrt{8(\ln 2)/T_2}$ satisfies*

$$L^{S_{12}} \leq \min(L^{S_1}, L^{S_2}) + \sqrt{(T_2/2)\ln 2}$$

2. FESL-s(election):



Both the two theorems implies that our two methods are comparable to the best predictor/classifier all the time.

$$v_{i,t} = \alpha_{i,t-1} e^{-\eta \ell(f_{i,t}, y_t)}, \; i = 1, 2,$$
$$\alpha_{i,t} = \delta \frac{W_t}{2} + (1-\delta) v_{i,t}, \; i = 1, 2, \quad (8)$$

$$p_{i,t} = \frac{\alpha_{i,t-1}}{\sum_{j=1}^{2} \alpha_{j,t-1}} \; i = 1, 2 \quad (7)$$

$$L^s = \sum_{t=T_1+1}^{s} \ell(f_{1,t}, y_t) + \sum_{t=s+1}^{T_2} \ell(f_{2,t}, y_t)$$

Theorem 2. *For all $T_2 > 1$, if the model is run with parameter $\eta = \sqrt{8/T_2\,(2\ln 2 + (T_2 - 1)H(1/T_2 - 1))}$ and $\delta = 1/(T_2 - 1)$, then*

$$L^{S_{12}} \leq \min_{T_1+1 \leq s \leq T_1+T_2} L^s + \sqrt{\frac{T_2}{2}\left(2\ln 2 + \frac{H(\delta)}{\delta}\right)}$$

where $H(x) = -x \ln x - (1-x)\ln(1-x)$ is the binary entropy function.

5. Experiments

Datasets:

Dataset	n	d_1	d_2	Dataset	n	d_1	d_2	Dataset	n	d_1	d_2
Australian	690	42	29	r.EN-FR	18,758	21,531	24,892	r.GR-IT	29,953	34,279	15,505
Credit-a	653	15	10	r.EN-GR	18,758	21,531	34,215	r.GR-SP	29,953	34,279	11,547
Credit-g	1,000	20	14	r.EN-IT	18,758	21,531	15,506	r.IT-EN	24,039	15,506	21,517
Diabetes	768	8	5	r.EN-SP	18,758	21,531	11,547	r.IT-FR	24,039	15,506	24,892
DNA	940	180	125	r.FR-EN	26,648	24,893	21,531	r.IT-GR	24,039	15,506	34,278
German	1,000	59	41	r.FR-GR	26,648	24,893	34,287	r.IT-SP	24,039	15,506	11,547
Kr-vs-kp	3,196	36	25	r.FR-IT	26,648	24,893	15,503	r.SP-EN	12,342	11,547	21,530
Splice	3,175	60	42	r.FR-SP	26,648	24,893	11,547	r.SP-FR	12,342	11,547	24,892
Svmguide3	1,284	22	15	r.GR-EN	29,953	34,279	21,531	r.SP-GR	12,342	11,547	34,262
RFID	940	78	72	r.GR-FR	29,953	34,279	24,892	r.SP-IT	12,342	11,547	15,500

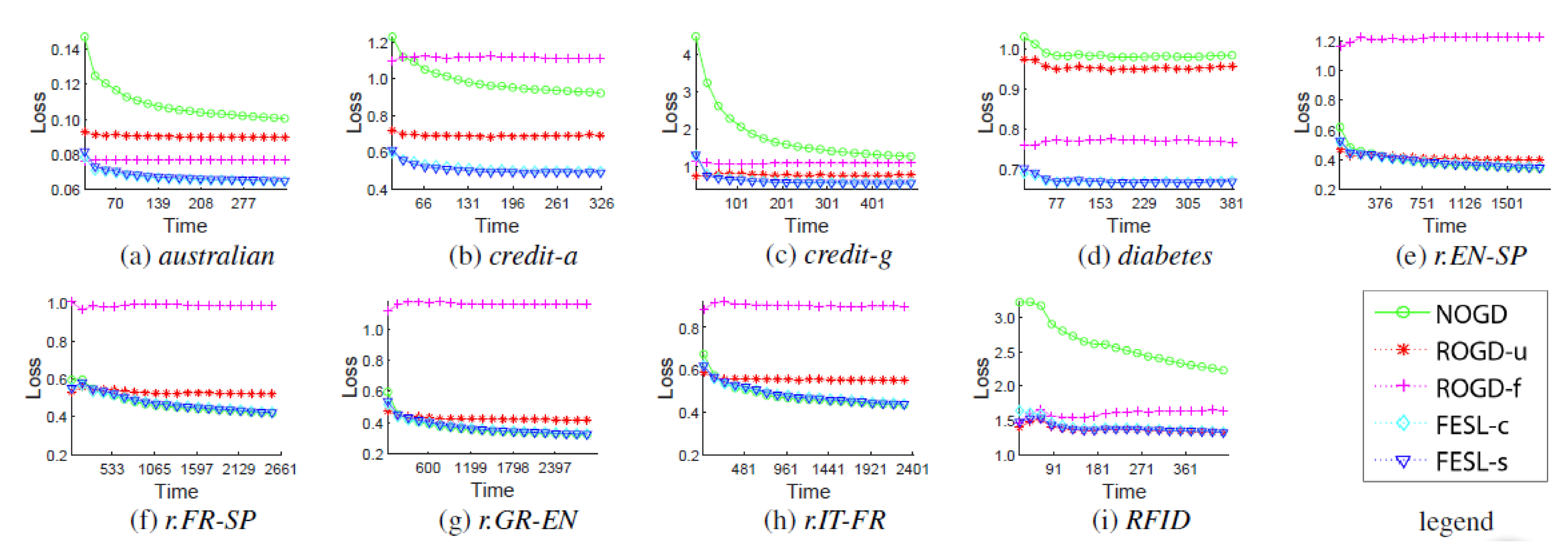
Synthetic
Reuter
Real

We want to emphasize that we collected one real dataset (e.g., RFID) by ourselves since the required datasets are not widely available yet.

Compared Methods:

NOGD: OGD algorithm will be invoked from scratch.
ROGD-u: the algorithm utilizes the classifier learned from feature space S_1 to do predictions on the recovered data, keeps updating.
ROGD-f: resembles ROGD-u, but do not update.

E1, The Trend of Loss:



(a) *australian* (b) *credit-a* (c) *credit-g* (d) *diabetes* (e) *r.EN-SP* (f) *r.FR-SP* (g) *r.GR-EN* (h) *r.IT-FR* (i) *RFID* legend

We present the trend of average cumulative loss. At each time t', the loss $\bar{l}_{t'}$ of every method is the average of the cumulative loss over $1, \dots, t'$, namely $\bar{l}_{t'} = (1/t')\sum_{t=1}^{t'} l_t$.

The smaller the average cumulative loss, the better. The average cumulative loss of our methods is comparable to the best of baseline methods on all datasets and are smaller than them on 8 datasets.

E2, The Accuracy Results:

Synthetic Datasets

Dataset	NOGD	ROGD-u	ROGD-f	FESL-c	FESL-s
australian	.767±.009	**.849±.009**	.809±.025	**.849±.009**	**.849±.009**
credit-a	.811±.006	.826±.018	.785±.051	.827±.014	**.831±.009**
credit-g	.659±.010	**.733±.006**	.716±.011	**.733±.006**	**.733±.006**
diabetes	.650±.002	**.652±.009**	.651±.006	**.652±.007**	**.652±.009**
dna	.610±.013	.691±.023	.608±.064	.691±.023	**.692±.021**
german	.684±.006	.700±.002	.700±.002	.700±.001	**.703±.004**
kr-vs-kp	.612±.005	.621±.036	.538±.024	.626±.028	**.630±.016**
splice	.568±.005	**.612±.022**	.567±.057	**.612±.022**	**.612±.022**
svmguide3	.680±.010	**.779±.010**	.748±.012	**.779±.010**	.778±.010

Reuter Datasets

r.EN-FR	.902±.004	.849±.003	.769±.069	**.903±.003**	.902±.005
r.EN-GR	.867±.005	.836±.007	.802±.036	**.870±.002**	**.870±.003**
r.EN-IT	.858±.014	.847±.014	.831±.018	.861±.010	**.863±.013**
r.EN-SP	.900±.002	.848±.002	.825±.001	**.901±.001**	.899±.002
r.FR-EN	**.858±.007**	.776±.009	.754±.012	**.858±.007**	**.858±.007**
r.FR-GR	.869±.004	.774±.019	.753±.021	**.870±.004**	.868±.003
r.FR-IT	.874±.005	.780±.022	.744±.040	**.874±.005**	.873±.005
r.FR-SP	**.872±.001**	.778±.022	.735±.013	**.872±.001**	.871±.002
r.GR-EN	**.907±.000**	.850±.007	.801±.035	**.907±.001**	.906±.000
r.GR-FR	**.898±.001**	.827±.009	.802±.023	**.898±.001**	**.898±.000**
r.GR-IT	.847±.011	**.851±.017**	.816±.006	.850±.018	**.851±.017**
r.GR-SP	**.902±.001**	.845±.003	.797±.012	**.902±.001**	**.902±.001**
r.IT-EN	.854±.003	.760±.006	.730±.024	**.856±.002**	.854±.003
r.IT-FR	.863±.002	.753±.012	.730±.020	**.864±.002**	.862±.003
r.IT-GR	.849±.004	.736±.022	.702±.012	**.849±.004**	.846±.004
r.IT-SP	**.839±.006**	.753±.014	.726±.005	**.839±.007**	**.839±.006**
r.SP-EN	**.926±.002**	.860±.005	.814±.021	**.926±.002**	.924±.001
r.SP-FR	.876±.005	.873±.017	.833±.042	.876±.014	**.878±.012**
r.SP-GR	.871±.013	.827±.025	.810±.026	**.873±.013**	**.873±.013**
r.SP-IT	**.928±.002**	.861±.005	.826±.005	**.928±.003**	.927±.002

For synthetic datasets, FESL-s outperforms other methods on 8 datasets, FESL-c gets the best on 5. Our methods can follow the best baseline method or even outperform it.

For Reuter datasets, FESL-c outperforms other methods on 17 datasets, FESL-s gets the best on 9. Our two methods can take the advantage of NOGD and ROGD-f and perform better than them.

6. Conclusion

1. A new setting: feature evolvable streaming learning. **2.** Key observation: in learning with streaming data, old features vanish and new ones occur. **3.** We assume there is an overlapping period that contains samples from both feature spaces.