One-Pass Feature Evolvable Learning with Theoretical Guarantees

Cun-Yuan Xing, M.-Z. Qian, W.-Y. Chen, W. Gao and Z.-H. Zhou

National Key Laboratory for Novel Software Technology, Nanjing University

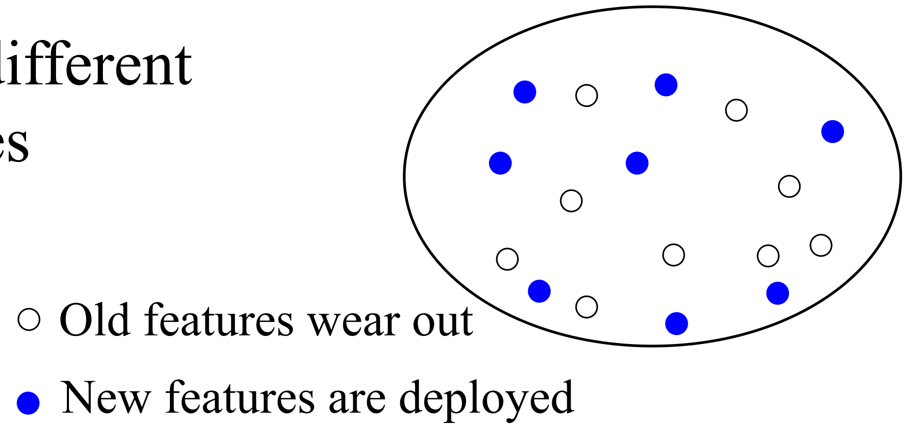
Outline

- **Introduction**
- Our Work
  - Kernel Ortho-Mapping discrepancy
  - Our OPFES approach
- Experiments
- Conclusion

Feature evolvable learning

Feature evolvable learning: old features will vanish and new features will emerge when learning with data streams [Hou et al., 2021; Zhang et al., 2021]

In environmental monitoring, different sensors collect different features



Previous feature evolvable methods consider different relationship

- FESL [Hou et al., 2017] considers linear relationship for feature space
- SF^2EL [Hou et al., 2021] takes kernel relationship for feature space
- OCDS [He et al., 2023] leverage linear relationship with graphical model
- …

About this work

Fundamental problems

- How to characterize the relationships between two different feature spaces
- How to exploit those relationships for feature evolvable learning

In this work, we propose

- **Kernel Ortho-Mapping (KOM) discrepancy** to characterize the relationships between two feature spaces via kernel functions
- **OPFES**: one-pass algorithm which incorporates feature and label relationships via KOM discrepancy

Outline

- Introduction
- **Our Work**
  - **Kernel Ortho-Mapping discrepancy**
  - Our OPFES approach
- Experiments
- Conclusion

Our characterization for feature space

- Kernels are introduced to represent feature spaces

$$\mathcal{K}^{[1]} \longrightarrow \text{old feat. space } \mathcal{X}^{[1]} \quad \text{and} \quad \mathcal{K}^{[2]} \longrightarrow \text{new feat. space } \mathcal{X}^{[2]}$$

- For sample $S_n = \left\{\left(x_i^{[1]}, x_i^{[2]}\right)\right\}_{i=1}^{n} \in \left(\mathcal{X}^{[1]} \times \mathcal{X}^{[2]}\right)^n$, define Gram matrices

$$\mathbf{K}^{[1]} = \left[\mathcal{K}^{[1]}\left(\boldsymbol{x}_i^{[1]}, \boldsymbol{x}_j^{[1]}\right)\right]_{n \times n} \quad \text{and} \quad \mathbf{K}^{[2]} = \left[\mathcal{K}^{[2]}\left(\boldsymbol{x}_i^{[2]}, \boldsymbol{x}_j^{[2]}\right)\right]_{n \times n}$$

We define the **Kernel Ortho-Mapping Discrepancy** over S_n as

$$\hat{\mathcal{E}}(S_n, \mathcal{K}^{[1]}, \mathcal{K}^{[2]}) = \min_{\mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{I}_n} \left\{ \left\| \mathbf{U}\sqrt{\mathbf{K}^{[1]}} - \sqrt{\mathbf{K}^{[2]}} \right\|_F / \sqrt{n} \right\}$$

- ◆ Dimensionality alignment via empirical kernel mapping for different feature space
- ◆ Minimization for the uniqueness of kernel mapping from rotational invariance

Our characterization for feature space

Lemma We have, for sample S_n

$$\hat{\mathcal{E}}\left(S_n, \mathcal{K}^{[1]}, \mathcal{K}^{[2]}\right) = \left(\mathrm{Tr}(\mathbf{K}^{[1]} + \mathbf{K}^{[2]})/n - 2\|\sqrt{\mathbf{K}^{[1]}}\sqrt{\mathbf{K}^{[2]}}\|_*/n\right)^{1/2}$$

KOM discrepancy and optimal classifiers

Old feat. space: optimal classifier $h_*^{[1]} \in \arg\min_{h \in \mathcal{H}^{[1]}} \left\{ \sum_{i=1}^{n} \ell(h, (\boldsymbol{x}_i^{[1]}, y_i))/n + \frac{\lambda}{2} \|h\|_{\mathcal{H}^{[1]}}^2 \right\}$

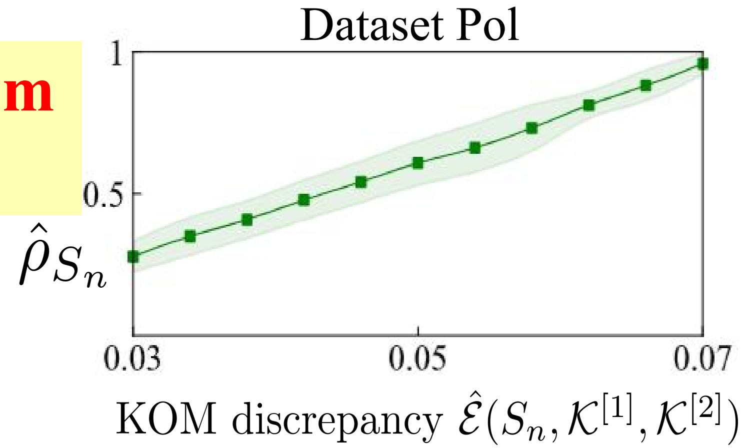
New feat. Space: optimal classifier $h_*^{[2]} \in \arg\min_{h \in \mathcal{H}^{[2]}} \left\{ \sum_{i=1}^{n} \ell(h, (\boldsymbol{x}_i^{[2]}, y_i))/n + \frac{\lambda}{2} \|h\|_{\mathcal{H}^{[2]}}^2 \right\}$

Theorem We have, for sample S_n,

$$\hat{\rho}_{S_n}(h_*^{[1]}, h_*^{[2]}) = \frac{1}{n} \sum_{i=1}^{n} |h_*^{[1]}(\boldsymbol{x}_i^{[1]}) - h_*^{[2]}(\boldsymbol{x}_i^{[2]})| \le \frac{r}{\lambda} \hat{\mathcal{E}}(S_n, \mathcal{K}^{[1]}, \mathcal{K}^{[2]}) + \frac{r}{\lambda} \sqrt{2r \hat{\mathcal{E}}(S_n, \mathcal{K}^{[1]}, \mathcal{K}^{[2]})}$$

Distance between two optimal classifiers **KOM discrepancy** **KOM discrepancy**

New insights: feature evolvable algorithm by optimizing KOM discrepancy



Positive relevance

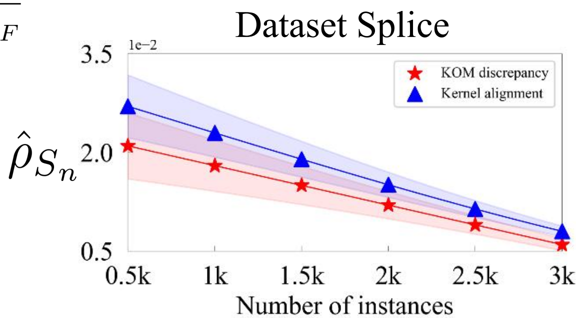
Comparison with previous characterization

Previous characterization:

- kernel alignment [Cortes et al., 2012] $\hat{A}(\mathbf{K}^{[1]}, \mathbf{K}^{[2]}) = \frac{\mathrm{Tr}(\mathbf{K}^{[1]}\mathbf{K}^{[2]})}{\|\mathbf{K}^{[1]}\|_F \|\mathbf{K}^{[2]}\|_F}$

Lemma For kernel alignment, we have

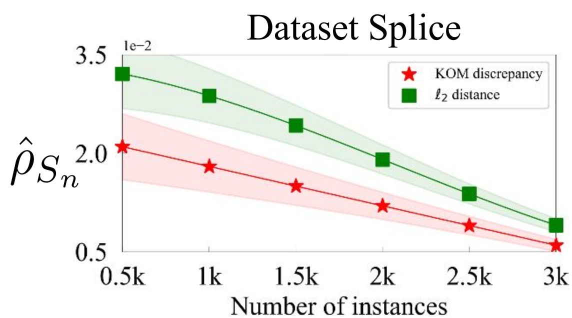
$$\hat{\mathcal{E}}(S_n, \mathcal{K}^{[1]}, \mathcal{K}^{[2]}) \leq r \sqrt[4]{2(1 - \hat{A}(\mathbf{K}^{[1]}, \mathbf{K}^{[2]}))} \; .$$



- ℓ_2 distance [Heo et al., 2019] $L_2(S_n) = \min_{\varphi^{[2]} \in \mathcal{F}} \left\{ \sum_{i=1}^{n} \left\| \varphi^{[1]}(\boldsymbol{x}_i^{[1]}) - \varphi^{[2]}(\boldsymbol{x}_i^{[2]}) \right\|_2^2 / n \right\}$

Lemma We have

$$\hat{\mathcal{E}}(S_n, \mathcal{K}^{[1]}, \mathcal{K}^{[2]}) \leq \sqrt{L_2(S_n)} \; .$$



A smaller difference between two optimal classifiers by optimizing KOM discrepancy

Outline

- Introduction
- Our Work
  - Kernel Ortho-Mapping discrepancy
  - **Our OPFES approach**
- Experiments
- Conclusion

Our OPFES approach

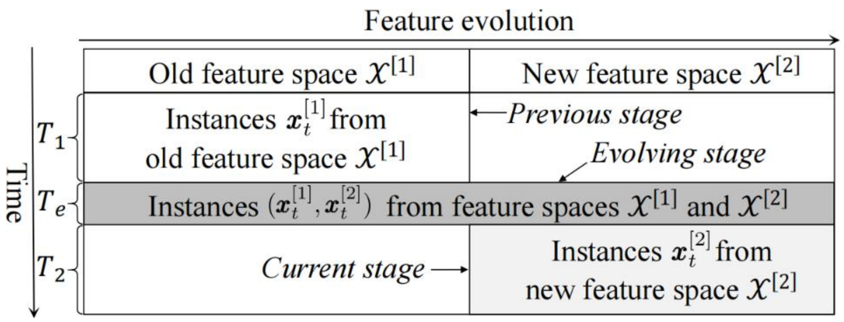
Learning And Mining from DatA

Three stages for feature evolvable learning

① Previous stage: receive instances $x_t^{[1]}$ from the old space $\mathcal{X}^{[1]}$ for $t = 1, \cdots, T_1$;

② Evolving stage: receive instances $x_t^{[1]}$ and $x_t^{[2]}$ from $\mathcal{X}^{[1]}$ and $\mathcal{X}^{[2]}$ respectively for $t = T_1 + 1, \cdots; T_1 + T_e$;

③ Current stage: receive instances $x_t^{[2]}$ from new space $\mathcal{X}^{[2]}$ for $t = T_1 + T_e + 1, \cdots, T_1 + T_e + T_2$.



Our OPFES approach

① In previous stage, we consider random Fourier features [Rahimi & Recht, 2008]

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \approx \sum_{k=1}^{d} p(\boldsymbol{u}_k)\phi(\boldsymbol{x}_i, \boldsymbol{u}_k, b_k)\phi(\boldsymbol{x}_j, \boldsymbol{u}_k, b_k) = \langle \boldsymbol{z}(\boldsymbol{x}_i), \boldsymbol{z}(\boldsymbol{x}_j) \rangle$$

and take one-pass learning algorithm to update model

$$\boldsymbol{w}_t^{[1]} = \boldsymbol{w}_{t-1}^{[1]} - \tau_t^{[1]} \nabla \ell_t^{[1]}(\boldsymbol{w}_{t-1}^{[1]})$$

Our OPFES approach

② In the evolving stage of sample $S_{T_e}^{[e]} = \left\{\left(x_t^{[1]}, x_t^{[2]}\right)\right\}_{t=T_1+1\cdots T_1+T_e}$

- Incorporate feature information by learning $\mathcal{K}^{[2]}$

$$\mathcal{K}^{[2]} \in \arg\min_{\mathcal{K}} \left\{\hat{\mathcal{E}}(S_{T_e}^{[e]}, \mathcal{K}^{[1]}, \mathcal{K})\right\}$$

- Incorporate label information by learning $\mathcal{K}^{l}$

$$\mathcal{K}^{l} \in \arg\min_{\mathcal{K}} \left\{\hat{\mathcal{E}}(S_{T_e}^{[e]}, \mathcal{K}^{*}, \mathcal{K})\right\} \qquad \mathcal{K}^{*}(x, x') = yy'$$

- Reuse previous model

$$\boldsymbol{w}_{T_1+T_e}^{[2]} = \mathbf{U}_*^{\top} \boldsymbol{w}_{T_1}^{[1]}, \quad \mathbf{U}_* \in \arg\min_{\mathbf{U} \in \mathcal{U}_{d_1}} \left\{ \left\| \mathbf{U}\left(\boldsymbol{z}^{[1]}(\boldsymbol{x}_{T_1+i}^{[1]})\right)_{i=1}^{T_e} - \left(\boldsymbol{z}^{[2]}(\boldsymbol{x}_{T_1+i}^{[2]})\right)_{i=1}^{T_e} \right\|_F \right\}$$

Our OPFES approach

③ In the current stage,

- An Online model $h_t^{[2]}(\boldsymbol{x}^{[2]}) \approx \langle \boldsymbol{w}_t^{[2]}, \boldsymbol{z}^{[2]}(\boldsymbol{x}^{[2]}) \rangle$ via random features over $\mathcal{K}^{[2]}$ and update

$$\boldsymbol{w}_t^{[2]} = \boldsymbol{w}_{t-1}^{[2]} - \tau_t^{[2]} \nabla \ell_t^{[2]}(\boldsymbol{w}_{t-1}^{[2]})$$

- Another online model $h_t^l(\boldsymbol{x}^{[2]}) \approx \langle \boldsymbol{w}_t^l, \boldsymbol{z}^l(\boldsymbol{x}^{[2]}) \rangle$ via random features over $\mathcal{K}^l$ and update

$$\boldsymbol{w}_t^l = \boldsymbol{w}_{t-1}^l - \tau_t^l \nabla \ell_t^l(\boldsymbol{w}_{t-1}^l)$$

- Online ensemble classifier

$$h_t(\boldsymbol{x}_t^{[2]}) = \omega_t \langle \boldsymbol{w}_t^{[2]}, \boldsymbol{z}^{[2]}(\boldsymbol{x}_t^{[2]}) \rangle + (1 - \omega_t) \langle \boldsymbol{w}_t^l, \boldsymbol{z}^l(\boldsymbol{x}_t^{[2]}) \rangle$$

Our OPFES approach

Algorithm 2 The OPFES method

Input: Feature evolvable stream sample $S_{T_1+T_e+T_2}$, kernel $\mathcal{K}^{[1]}$, stepsize $\tau_t^{[1]}$, $\tau_t^{[2]}$ and τ_t^l, sensitivity parameter γ

Initialize: $\boldsymbol{w}_0^{[1]} = \mathbf{0}$

Output: classifier $h_{T_1+T_e+T_2}$

Obtain random Fourier features $(\boldsymbol{u}_k^{[1]}, b_k^{[1]})_{k=1}^{d_1}$ and $(\boldsymbol{u}_k^{[2]}, b_k^{[2]}, \boldsymbol{u}_k^l, b_k^l)_{k=1}^{d_2}$ via Eqn. (1)
for $t = 1, \cdots, T_1$ **do**
  Update $\boldsymbol{w}_t^{[1]}$ by online gradient descent in Eqn. (2) — One-pass online kernel learning
end for
Obtain $\boldsymbol{p}^{[2]}$ and $\boldsymbol{p}^l$ from Algorithm 1 — Feature and label information incorporation
Compute $\boldsymbol{w}_{T_1+T_e}^{[2]}$ by Eqn. (11) — Previous model reuse
for $t = T_1 + T_e + 1, \cdots, T_1 + T_e + T_2$ **do**
  Update $\boldsymbol{w}_t^{[2]}$ and $\boldsymbol{w}_t^l$ by Eqns. (9)-(10), respectively — Learn two online models
  Update the combined classifier h_t by Eqn. (12) — Prediction with online ensemble
end for
return: classifier $h_{T_1+T_e+T_2}$

Convergence guarantee

Theorem The following holds with probability at least $1-\delta\ (0<\delta<1)$,

$$\frac{1}{T_2}\sum_{t=T_1+T_e+1}^{T_1+T_e+T_2}\left(\ell_t^{[2]}(\boldsymbol{w}_t^{[2]})-\ell_t^{[2]}(\boldsymbol{w}_*^{[2]})\right)$$

$$\leq \quad \frac{4r^2}{\lambda\sqrt{T_2}}\left(\frac{\mathcal{E}}{r}+\sqrt{\frac{\mathcal{E}}{r}}\right)^{1/2}+\frac{c_2 r^2}{\lambda\sqrt{T_2}}\left[\left(\frac{1}{\sqrt{T_1}}+\frac{1}{\sqrt{T_e}}+\frac{1}{\sqrt[4]{T_2}}\right)\sqrt{\ln\frac{6}{\delta}}\right]^{1/2}$$

- We obtain a tighter bound as for closer feature relationship
- It is useful to exploit information and model from old feature space

Outline

- Introduction
- Our Work
  - Kernel Ortho-Mapping discrepancy
  - Our OPFES approach
- **Experiments**
- Conclusion

Datasets

Benchmark datasets

Dataset	# Inst.	# Feat.	Dataset	# Inst.	# Feat.
jungle	2351	87	usps	9298	25
splice	3175	60	aileron	13750	40
bioresponse	3751	1776	elevators	16599	18
christine	5418	1636	pol	15000	44
svmguide1	7089	4	magic	19020	10

Large-scale datasets

Dataset	# Inst.	# Feat.	Dataset	# Inst.	# Feat.
letter	20000	16	nomao	34465	118
house	22784	16	adult	48842	108
acoustic	78823	50	runwalk	88588	6
higgs	98049	28	miniboone	130064	50
ijcnn1	141691	22	covtype	581012	54

Compared methods

- **lin-ROGD**: linear model, ℓ_2 distance for feature relationship [Hou et al., 2017]
- **lin-FESL**: lin-ROGD + linear model learned from scratch [Hou et al., 2021]
- **ker-ROGD**: kernel model, ℓ_2 distance for feature relationship [Hou et al., 2021]
- **ker-FESL**: ker-ROGD + kernel model learned from scratch [Hou et al., 2021]
- **OCDS**: linear model, generative graphical model for feature relationship [He et al., 2021]

Additional compared methods

- **rff-ROGD**: random feature model, ℓ_2 distance for feature relationship [Lu et al., 2016]
- **rff-FESL**: rff-ROGD + random feature model learned from scratch [Hou et al., 2021]
- **align-FESL**: random feature model, kernel alignment for feature and label relationship [Sinha & Duchi, 2016]

"+" stands for online ensemble

Results

Dataset	Our OPFES	lin-FESL	lin-ROGD	OCDS
jungle	.0097 ± .0047	.1084 ± .0152●	.1471 ± .0144●	.1106 ± .0138●
splice	.3070 ± .0079	.3447 ± .0097●	.4307 ± .0213●	.3547 ± .0156●
bioresponse	.2763 ± .0117	.2938 ± .0093●	.3684 ± .0102●	.2951 ± .0112●
christine	.3192 ± .0095	.3443 ± .0098●	.3439 ± .0098●	.3663 ± .0116●
svmguide1	.1614 ± .0052	.2399 ± .0062●	.2451 ± .0102●	.2442 ± .0070●
usps	.1684 ± .0044	.2654 ± .0081●	.2839 ± .0073●	.2746 ± .0085●
aileron	.1963 ± .0034	.2466 ± .0066●	.2465 ± .0066●	.3026 ± .0047●
pol	.0654 ± .0036	.1484 ± .0035●	.1655 ± .0041●	.1515 ± .0038●
elevators	.2422 ± .0039	.3073 ± .0043●	.3045 ± .0040●	.3073 ± .0042●
magic	.2154 ± .0039	.2535 ± .0040●	.2988 ± .0073●	.2554 ± .0045●
letter	.1354 ± .0043	.3380 ± .0038●	.3565 ± .0071●	.3390 ± .0034●
house	.1849 ± .0040	.2623 ± .0084●	.2658 ± .0120●	.2853 ± .0037●
nomao	.0646 ± .0026	.0860 ± .0019●	.1107 ± .0039●	.0882 ± .0023●
adult	.1875 ± .0023	.2050 ± .0033●	.2036 ± .0042●	.2303 ± .0026●
acoustic	.3074 ± .0024	.4321 ± .0079●	.4317 ± .0075●	.4668 ± .0022●
runwalk	.2602 ± .0033	.4945 ± .0021●	.4963 ± .0033●	.4972 ± .0027●
higgs	.3946 ± .0045	.4309 ± .0028●	.4481 ± .0139●	.4366 ± .0021●
miniboone	.1602 ± .0036	.2384 ± .0039●	.2384 ± .0039●	.2803 ± .0011●
ijcnn1	.0616 ± .0115	.0951 ± .0007●	.0957 ± .0009●	.0957 ± .0009●
covtype	.3782 ± .0008	.3790 ± .0007●	.3920 ± .0034●	.3792 ± .0007●
Win/Tie/Loss		20/0/0	20/0/0	20/0/0

Our OPFES is significantly better than linear methods with ℓ_2-distance for relationship characterization

Results

Dataset	Our OPFES	rff-FESL	rff-ROGD	ker-FESL	ker-ROGD
jungle	.0097 ± .0047	.0161 ± .0035•	.0246 ± .0069•	.0276 ± .0055•	.0329 ± .0061•
splice	.3070 ± .0079	.3234 ± .0087•	.3662 ± .0215•	.4192 ± .0160•	.4240 ± .0188•
bioresponse	.2763 ± .0117	.3051 ± .0137•	.4285 ± .0192•	.3690 ± .0095•	.4454 ± .0116•
christine	.3192 ± .0095	.3316 ± .0108•	.3503 ± .0092•	.3858 ± .0096•	.4506 ± .0117•
svmguide1	.1614 ± .0052	.1632 ± .0054•	.2295 ± .0107•	.1900 ± .0061•	.2316 ± .0050•
usps	.1684 ± .0044	.1658 ± .0051	.2184 ± .0061•	.2267 ± .0084•	.2857 ± .0063•
aileron	.1963 ± .0034	.2139 ± .0059•	.2344 ± .0098•	.2531 ± .0076•	.2523 ± .0076•
pol	.0654 ± .0036	.0686 ± .0023•	.0807 ± .0028•	.0865 ± .0023•	.0956 ± .0034•
elevators	.2422 ± .0039	.2467 ± .0037•	.2619 ± .0045•	.2963 ± .0042•	.3003 ± .0051•
magic	.2154 ± .0039	.2121 ± .0046	.2434 ± .0057•	.2656 ± .0033•	.3119 ± .0074•
letter	.1354 ± .0043	.1557 ± .0037•	.2311 ± .0067•	.3139 ± .0060•	.3373 ± .0076•
house	.1849 ± .0040	.1894 ± .0043•	.2001 ± .0093•	.2598 ± .0112•	.2597 ± .0113•
nomao	.0646 ± .0026	.0778 ± .0017•	.0845 ± .0041•	.1302 ± .0034•	.1355 ± .0032•
adult	.1875 ± .0023	.1906 ± .0021•	.1932 ± .0029•	.2277 ± .0019•	.2218 ± .0031•
acoustic	.3074 ± .0024	.2967 ± .0045◦	.2977 ± .0043◦	.4168 ± .0073•	.4107 ± .0072•
runwalk	.2602 ± .0033	.2578 ± .0016◦	.3496 ± .0130•	.3558 ± .0021•	.4355 ± .0061•
higgs	.3946 ± .0045	.3803 ± .0074◦	.3807 ± .0080◦	.4577 ± .0054•	.4570 ± .0055•
miniboone	.1602 ± .0036	.1729 ± .0029•	.1603 ± .0029	.2488 ± .0047•	.2484 ± .0047•
ijcnn1	.0616 ± .0115	.0673 ± .0028•	.0747 ± .0083•	.0957 ± .0009•	.0957 ± .0009•
covtype	.3782 ± .0008	.3783 ± .0009	.3795 ± .0012•	.4095 ± .0025•	.4093 ± .0025•
Win/Tie/Loss		14/3/3	17/1/2	20/0/0	20/0/0

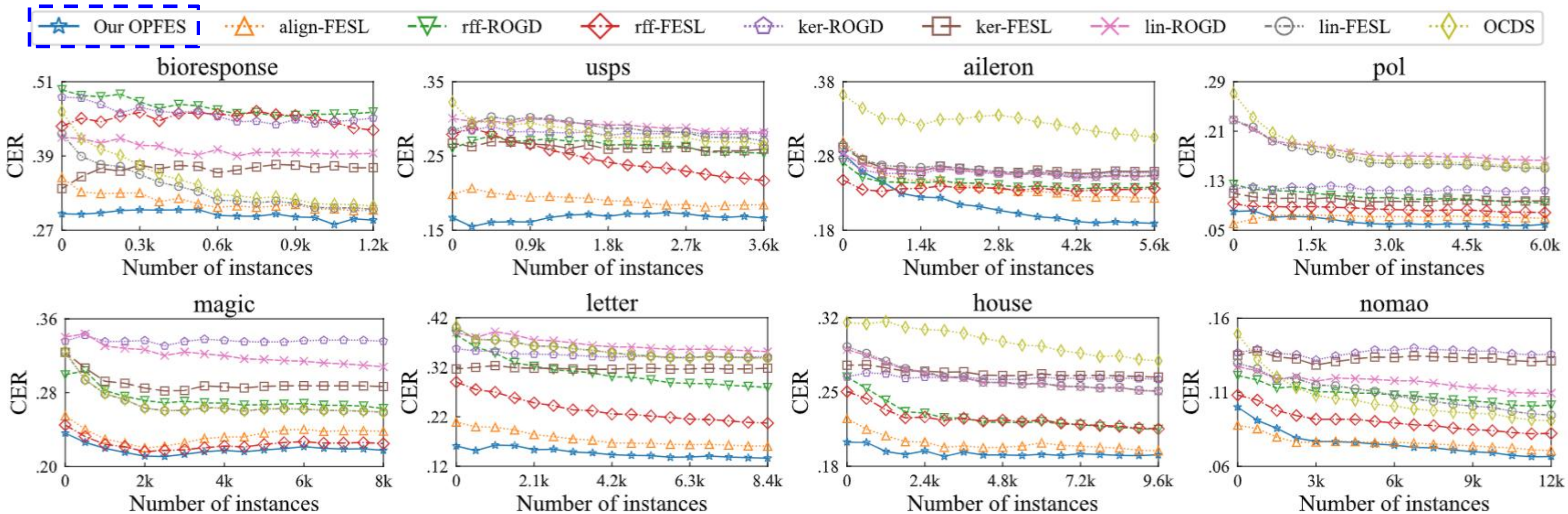
Our OPFES also outperforms kernel and random feature models with ℓ_2-distance for feature relationship

Results

Dataset	Our OPFES	align-FESL
jungle	$.0097 \pm .0047$	$.0099 \pm .0028$
splice	$.3070 \pm .0079$	$.3126 \pm .0136$
bioresponse	$.2763 \pm .0117$	$.2921 \pm .0106\bullet$
christine	$.3192 \pm .0095$	$.3205 \pm .0090$
svmguide1	$.1614 \pm .0052$	$.1617 \pm .0056$
usps	$.1684 \pm .0044$	$.1875 \pm .0073\bullet$
aileron	$.1963 \pm .0034$	$.2144 \pm .0081\bullet$
pol	$.0654 \pm .0036$	$.0692 \pm .0044\bullet$
elevators	$.2422 \pm .0039$	$.2419 \pm .0038$
magic	$.2154 \pm .0039$	$.2206 \pm .0039\bullet$
letter	$.1354 \pm .0043$	$.1568 \pm .0086\bullet$
house	$.1849 \pm .0040$	$.1927 \pm .0030\bullet$
nomao	$.0646 \pm .0026$	$.0680 \pm .0024\bullet$
adult	$.1875 \pm .0023$	$.1942 \pm .0027\bullet$
acoustic	$.3074 \pm .0024$	$.3227 \pm .0036\bullet$
runwalk	$.2602 \pm .0033$	$.2890 \pm .0055\bullet$
higgs	$.3946 \pm .0045$	$.4135 \pm .0061\bullet$
miniboone	$.1602 \pm .0036$	$.2804 \pm .0011\bullet$
ijcnn1	$.0616 \pm .0115$	$.0746 \pm .0038\bullet$
covtype	$.3782 \pm .0008$	$.3813 \pm .0025\bullet$
Win/Tie/Loss		15/5/0

Our OPFES is also better than random feature models with kernel alignment for feature and label relationship

Convergence results



Our OPFES takes a faster convergence from feature and label relationship characterization with KOM discrepancy and model reuse

Outline

- Introduction
- Our Work
  - Kernel Ortho-Mapping discrepancy
  - Our OPFES approach
- Experiments
- **Conclusion**

Conclusion

Learning And Mining from DatA

In this work, we propose

- **Kernel Ortho-Mapping (KOM) discrepancy** to characterize the relationship between two feature spaces via kernel functions
- **OPFES**: one-pass algorithm which incorporates feature and label relationships via KOM discrepancy

Future work: Extension of KOM discrepancy to deep learning.

Paper link Contact

Thanks!