ParK: Sound and Efficient Kernel Ridge Regression by Feature Space Partitions

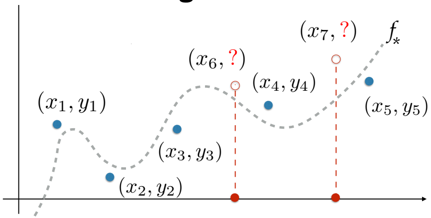
Luigi Carratino[1] Stefano Vigogna[1] Daniele Calandriello[2] Lorenzo Rosasco[13]

[1]MaLGa - DIBRIS, University of Genova [2]DeepMind Paris [3]CBMM, MIT & IIT

NeurIPS 2021

Kernel ridge regression

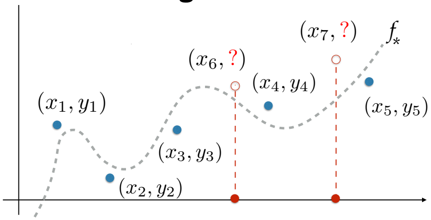
Regression



$$y_i = f_*(x_i) + \epsilon_i \qquad i = 1, \ldots, n$$

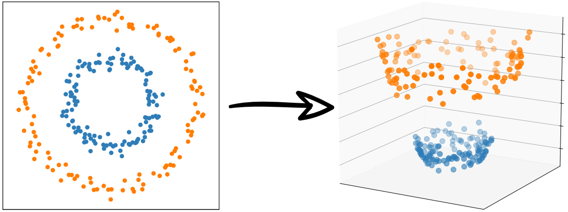
Kernel ridge regression

Regression



$$y_i = f_*(x_i) + \epsilon_i \qquad i = 1, \ldots, n$$

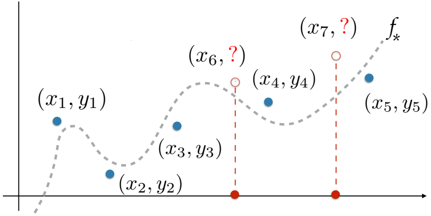
KRR



$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|^2 + \lambda \|f\|_{\mathcal{H}}^2$$

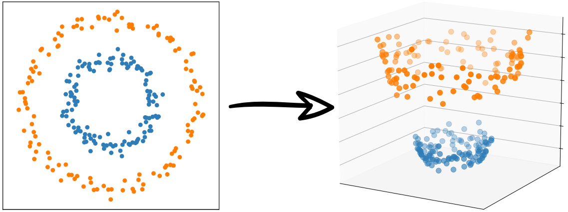
Kernel ridge regression

Regression



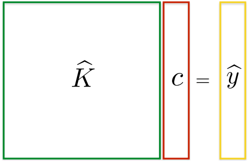
$$y_i = f_*(x_i) + \epsilon_i \qquad i = 1, \dots, n$$

KRR



$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|^2 + \lambda \|f\|_{\mathcal{H}}^2$$

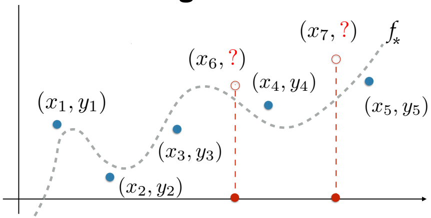
Solution



$$\widehat{f}_\lambda(x) = \sum_{i=1}^{n} c_i K(x_i, x) \qquad c = (\widehat{K} + \lambda n I)^{-1} \widehat{y}$$

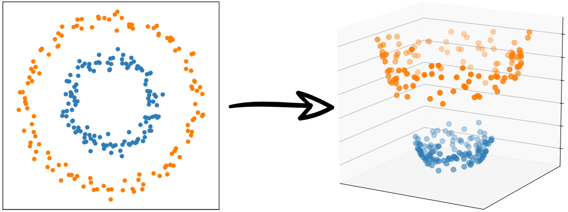
Kernel ridge regression

Regression



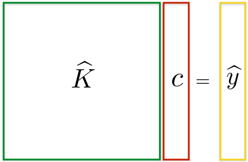
$$y_i = f_*(x_i) + \epsilon_i \qquad i = 1, \dots, n$$

KRR



$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} |f(x_i) - y_i|^2 + \lambda \|f\|_{\mathcal{H}}^2$$

Solution



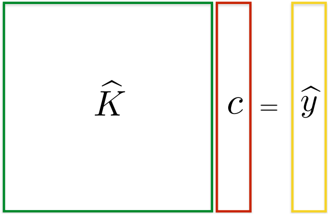
$$\widehat{f}_\lambda(x) = \sum_{i=1}^{n} c_i K(x_i, x) \qquad c = (\widehat{K} + \lambda n I)^{-1} \widehat{y}$$

Complexity

- Statistics: optimal
- Computations: $\mathcal{O}(n^3)$

Accelerated KRR

Naive $\mathcal{O}(n^3)$



Accelerated KRR

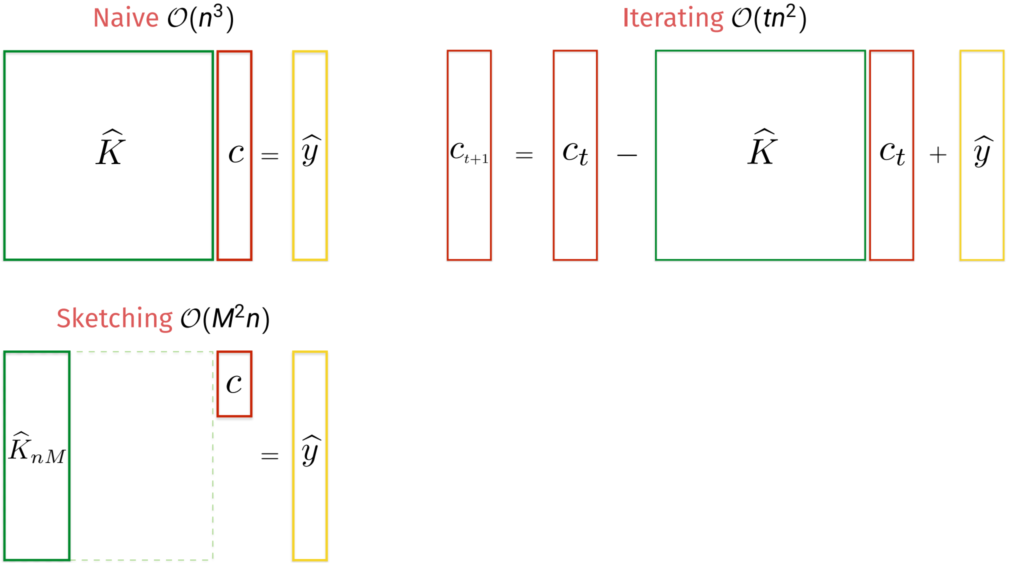
Naive $\mathcal{O}(n^3)$

$$\widehat{K} \, c = \widehat{y}$$

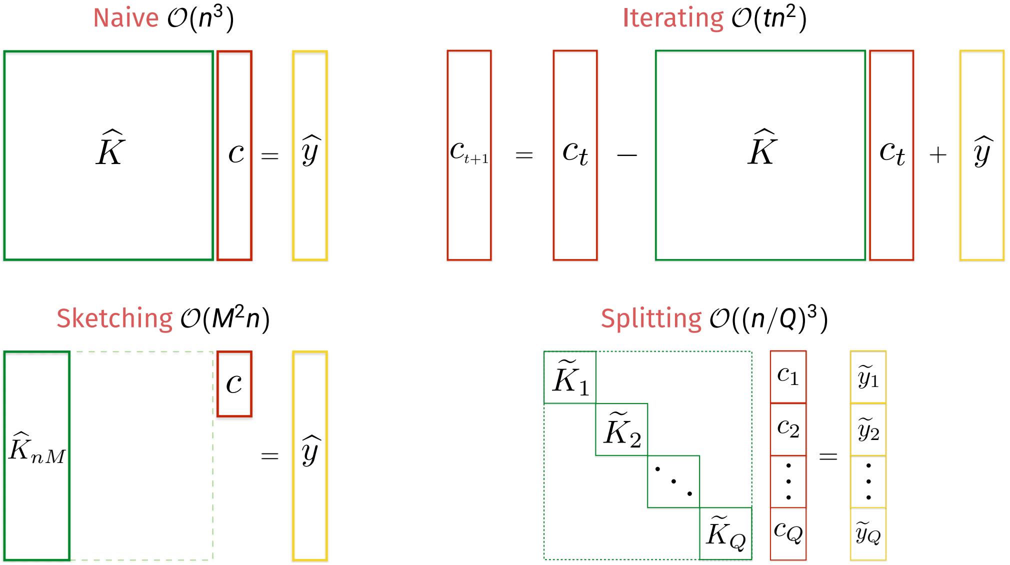
Iterating $\mathcal{O}(tn^2)$

$$c_{t+1} = c_t - \widehat{K} \, c_t + \widehat{y}$$

Accelerated KRR



Accelerated KRR



Combined methods

GD, CG	**Iterating**	$\mathcal{O}(tn^2)$
Nyström[a], RF[b]	**Sketching**	$\mathcal{O}(M^2 n)$
D&C[c], DC-KRR[d]	**Splitting**	$\mathcal{O}((n/Q)^3)$

a. [Williams, Seeger, ’00]
b. [Rahimi, Recht, ’09]

c. [Zhang, Duchi, Wainwright, ’15]
d. [Tandon, Si, Ravikumar, Dhillon, ’16]

Combined methods

FALKON[e]

GD, CG	**Iterating**	$\mathcal{O}(tn^2)$	$\mathcal{O}(tMn)$ (combines Iterating and Sketching)
Nyström[a], RF[b]	**Sketching**	$\mathcal{O}(M^2 n)$	
D&C[c], DC-KRR[d]	**Splitting**	$\mathcal{O}((n/Q)^3)$	

a. [Williams, Seeger, '00]
b. [Rahimi, Recht, '09]

c. [Zhang, Duchi, Wainwright, '15]
d. [Tandon, Si, Ravikumar, Dhillon, '16]

e. [Rudi, Carratino, Rosasco, '17]

Combined methods

				FALKON[e]	LocalNysation[f]
GD, CG	**Iterating**	$\mathcal{O}(tn^2)$	]	$\mathcal{O}(tMn)$	
Nyström[a], RF[b]	**Sketching**	$\mathcal{O}(M^2 n)$	]		] $\mathcal{O}(M^2(n/Q))$
D&C[c], DC-KRR[d]	**Splitting**	$\mathcal{O}((n/Q)^3)$			]

a. [Williams, Seeger, ’00]
b. [Rahimi, Recht, ’09]

c. [Zhang, Duchi, Wainwright, ’15]
d. [Tandon, Si, Ravikumar, Dhillon, ’16]

e. [Rudi, Carratino, Rosasco, ’17]
f. [Mücke, ’19]

Combined methods

			FALKON[e]	LocalNysation[f]	ParK
GD, CG	**Iterating**	$\mathcal{O}(tn^2)$	$\mathcal{O}(tMn)$		$\mathcal{O}(tM(n/Q)^2)$
Nyström[a], RF[b]	**Sketching**	$\mathcal{O}(M^2n)$		$\mathcal{O}(M^2(n/Q))$	
D&C[c], DC-KRR[d]	**Splitting**	$\mathcal{O}((n/Q)^3)$			

a. [Williams, Seeger, ’00]
b. [Rahimi, Recht, ’09]

c. [Zhang, Duchi, Wainwright, ’15]
d. [Tandon, Si, Ravikumar, Dhillon, ’16]

e. [Rudi, Carratino, Rosasco, ’17]
f. [Mücke, ’19]

Our contribution

ParK, a new large-scale KRR solver that

Our contribution

ParK, a new large-scale KRR solver that

- combines the computational benefits of iterations, sketching and splitting

Our contribution

ParK, a new large-scale KRR solver that

- combines the computational benefits of iterations, sketching and splitting
- preserves the generalization power under suitable partitions

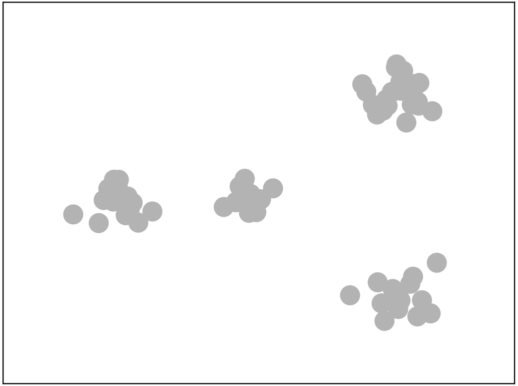
Our contribution

ParK, a new large-scale KRR solver that

- combines the computational benefits of iterations, sketching and splitting
- preserves the generalization power under suitable partitions
- introduces a new principled partition scheme for kernel methods

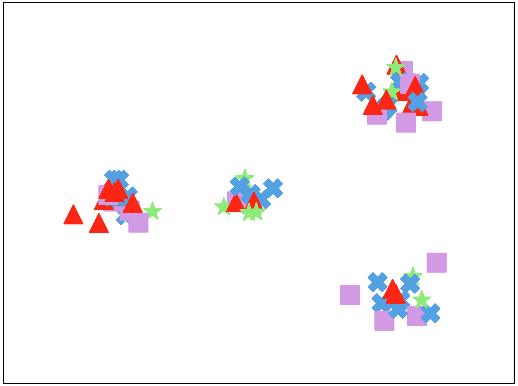
Data splitting vs space partitons

Splitting



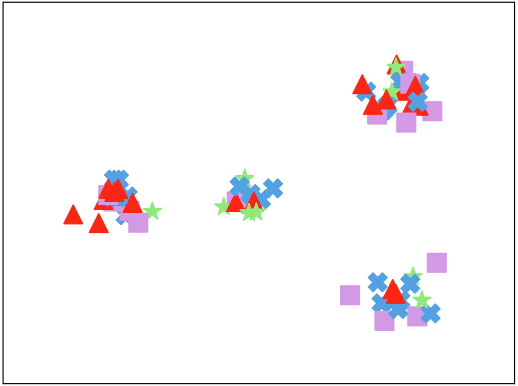
Data splitting vs space partitons

Splitting

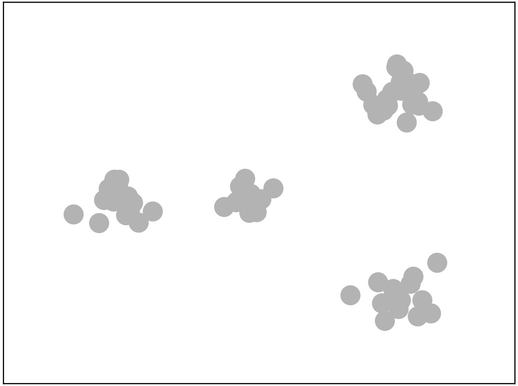


Data splitting vs space partitons

Splitting

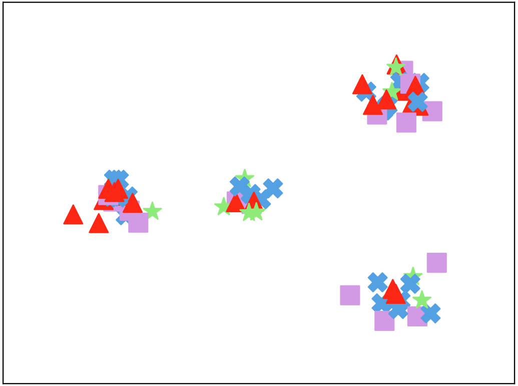


Partitioning

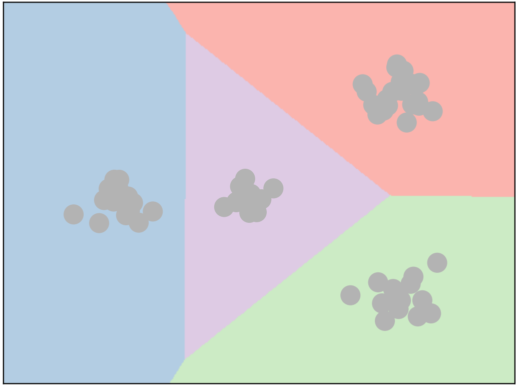


Data splitting vs space partitons

Splitting

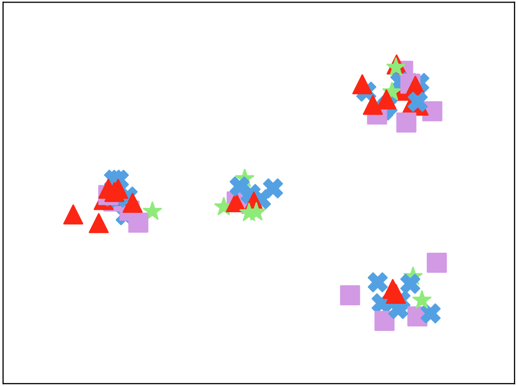


Partitioning

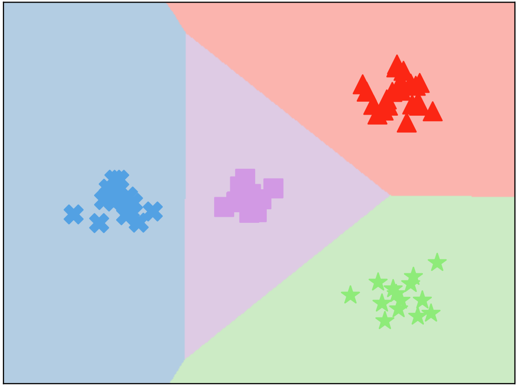


Data splitting vs space partitons

Splitting

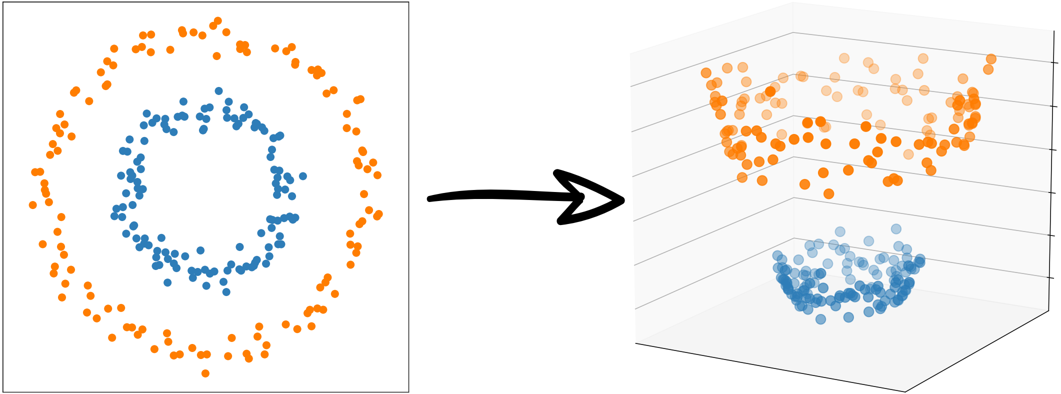


Partitioning



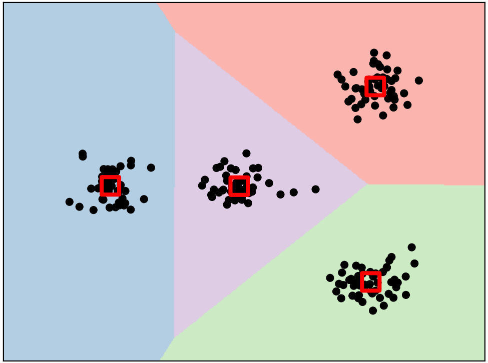
Input vs feature space partitions

$$\mathcal{X} \xrightarrow{\phi} \mathcal{H}$$



Input vs feature space partitions

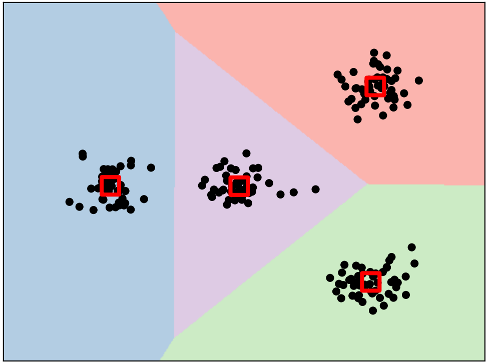
$$\mathcal{X} \xrightarrow{\phi} \mathcal{H}$$



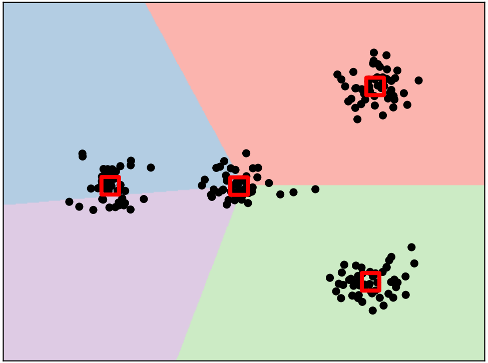
$$\|\mathbf{x}_1 - \mathbf{x}_2\|^2 = \|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2 - 2\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$$

Input vs feature space partitions

$$\mathcal{X} \xrightarrow{\phi} \mathcal{H}$$



$$\|\mathbf{x}_1 - \mathbf{x}_2\|^2 = \|\mathbf{x}_1\|^2 + \|\mathbf{x}_2\|^2 - 2\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$$

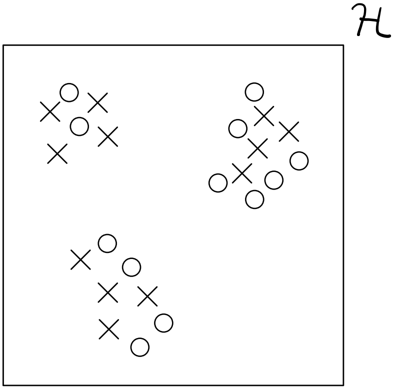


$$\|\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)\|^2_{\mathcal{H}} = 2 - 2\frac{\langle \mathbf{x}_1, \mathbf{x}_2 \rangle}{\|\mathbf{x}_1\|\|\mathbf{x}_2\|}$$

ParK

1. partition the feature space into Q Voronoi cells:

$$\mathcal{H} = \bigcup_{q=1}^{Q} V_q$$

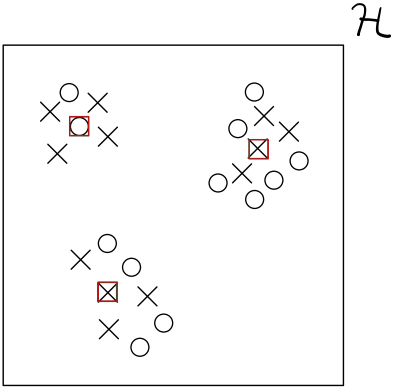


ParK

1. partition the feature space into Q Voronoi cells:

$$\mathcal{H} = \bigcup_{q=1}^{Q} V_q$$

$\phi(c_k)$

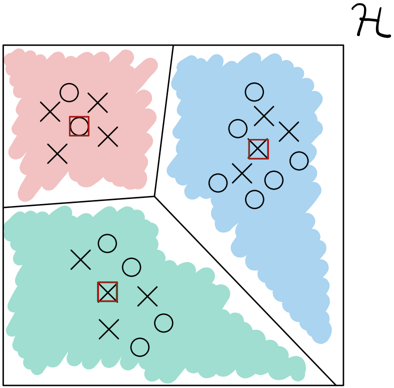


ParK

1. partition the feature space into Q Voronoi cells:

$$\mathcal{H} = \bigcup_{q=1}^{Q} V_q$$

$$V_q = \{\phi(x) : q = \arg\min_k \|\phi(x) - \phi(c_k)\|^2_{\mathcal{H}}\}$$



ParK

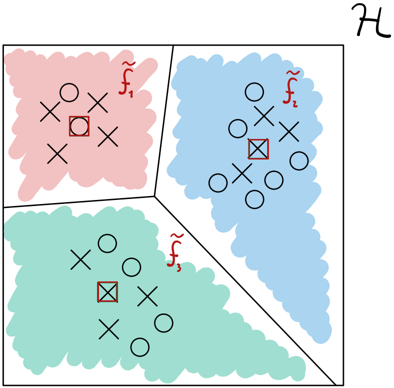
1. partition the feature space into Q Voronoi cells:

$$\mathcal{H} = \bigcup_{q=1}^{Q} V_q$$

$$V_q = \{\phi(x) : q = \arg\min_k \|\phi(x) - \phi(c_k)\|_{\mathcal{H}}^2\}$$

2. solve (iterated, sketched) KRR locally on each cell:

$$\tilde{f}_q \in \mathcal{H}_q = \operatorname{span} V_q$$



ParK

1. partition the feature space into Q Voronoi cells:

$$\mathcal{H} = \bigcup_{q=1}^{Q} V_q$$

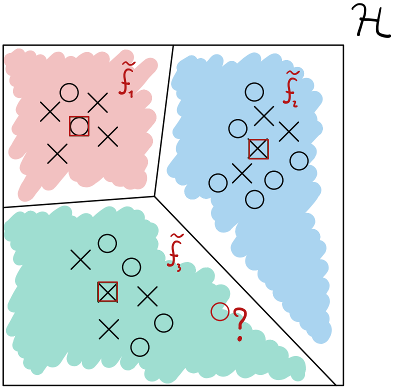
$$V_q = \{\phi(x) : q = \arg\min_k \|\phi(x) - \phi(c_k)\|_{\mathcal{H}}^2\}$$

2. solve (iterated, sketched) KRR locally on each cell:

$$\widetilde{f}_q \in \mathcal{H}_q = \text{span}\, V_q$$

3. predict new samples on the corresponding cells:

$$\widehat{f}(x) = \widetilde{f}_q(x) \qquad \text{if } \phi(x) \in V_q$$



Generalization

KRR generalization without partitioning $\|\widehat{f} - f_*\|^2 \lesssim \lambda + \frac{d_{\text{eff}}(\lambda)}{n}$

Generalization

KRR generalization without partitioning $\|\widehat{f} - f_*\|^2 \lesssim \lambda + \frac{d_{\text{eff}}(\lambda)}{n}$

Theorem (Carratino, Vigogna, Calandriello, Rosasco '21)

Let $\theta = \min_{q \neq k} \angle(\mathcal{H}_q, \mathcal{H}_k)$ and $\lambda_q = \lambda n / \# V_q$. Then w.h.p.

$$\|\widehat{f} - f_*\|^2 \lesssim (1 + Q^2 \cos(\theta))\lambda + \left(1 + \frac{\cos^2(\theta)}{\lambda}\right) \frac{d_{\text{eff}}(\lambda)}{n}$$

Generalization

KRR generalization without partitioning $\|\widehat{f} - f_*\|^2 \lesssim \lambda + \frac{d_{\text{eff}}(\lambda)}{n}$

Theorem (Carratino, Vigogna, Calandriello, Rosasco '21)

Let $\theta = \min_{q \neq k} \angle(\mathcal{H}_q, \mathcal{H}_k)$ and $\lambda_q = \lambda n / \# V_q$. Then w.h.p.

$$\|\widehat{f} - f_*\|^2 \lesssim (1 + Q^2 \cos(\theta))\lambda + \left(1 + \frac{\cos^2(\theta)}{\lambda}\right) \frac{d_{\text{eff}}(\lambda)}{n}$$

When cells are orthogonal (*i.e.* $\mathcal{H} = \bigoplus_{q=1}^{Q} \mathcal{H}_q$ *i.e.* $\theta = \pi/2$) we recover $\|\widehat{f} - f_*\|^2 \lesssim \lambda + \frac{d_{\text{eff}}(\lambda)}{n}$

Generalization

KRR generalization without partitioning $\|\widehat{f} - f_*\|^2 \lesssim \lambda + \frac{d_{\text{eff}}(\lambda)}{n}$

Theorem (Carratino, Vigogna, Calandriello, Rosasco '21)

Let $\theta = \min_{q \neq k} \angle(\mathcal{H}_q, \mathcal{H}_k)$ and $\lambda_q = \lambda n / \# V_q$. Then w.h.p.

$$\|\widehat{f} - f_*\|^2 \lesssim (1 + Q^2 \cos(\theta))\lambda + \left(1 + \frac{\cos^2(\theta)}{\lambda}\right) \frac{d_{\text{eff}}(\lambda)}{n}$$

When cells are orthogonal (*i.e.* $\mathcal{H} = \bigoplus_{q=1}^{Q} \mathcal{H}_q$ *i.e.* $\theta = \pi/2$) we recover $\|\widehat{f} - f_*\|^2 \lesssim \lambda + \frac{d_{\text{eff}}(\lambda)}{n}$

When $\cos(\theta) = \mathcal{O}(\min(1/Q^2, \lambda))$ we obtain $\|\widehat{f} - f_*\|^2 \lesssim \mathcal{O}(\lambda + \frac{d_{\text{eff}}(\lambda)}{n})$

Feature space Voronoi partitions

Voronoi centroids:

greedy select

$$c_{q+1} = \operatorname*{arg\,max}_{c \in \{x_i\}_{i=1}^n \setminus \{c_1, \dots, c_q\}} \mathrm{SC}_q(c)$$

where $\mathrm{SC}_q(c)$ is the Schur complement of $[K(c_k, c_h)]_{k,h=1}^q$ in $\begin{bmatrix} K(c,c) & K(c,c_k) \\ K(c,c_k)^\top & K(c_k,c_h) \end{bmatrix}$

Feature space Voronoi partitions

Voronoi centroids:

greedy select

$$c_{q+1} = \underset{c \in \{x_i\}_{i=1}^n \setminus \{c_1, \ldots, c_q\}}{\arg\max} \mathrm{SC}_q(c)$$

where $\mathrm{SC}_q(c)$ is the Schur complement of $[K(c_k, c_h)]_{k,h=1}^q$ in $\begin{bmatrix} K(c,c) & K(c,c_k) \\ K(c,c_k)^\top & K(c_k,c_h) \end{bmatrix}$

ParK complexity: $\mathcal{O}(Q^2 n \log(n) + \max_q t_q M_q n_q)$

Experiments

	TAXI $n \approx 10^9$				HIGGS $n \approx 10^7$			
	ERROR (RMSE)	TIME (MIN.)			ERROR (1−AUC)	TIME (SEC.)		
		INIT	TRAIN	TOTAL		INIT	TRAIN	TOTAL
PARK	312.0±0.2	25±1	39±13	64±13	0.182±0.001	30±2	474±172	504±172
FALKON	311.7±0.1	-	-	120±1	0.180±0.001	-	-	715±6
D&C-FALK	356.2±0.2	-	-	14±1	0.212±0.000	-	-	50±1
D&C	OUT OF MEMORY				OUT OF MEMORY			

	AIRLINE $n \approx 10^6$				AIRLINE-CLS $n \approx 10^6$			
	ERROR (MSE)	TIME (SEC.)			ERROR (C-ERR)	TIME (SEC.)		
		INIT	TRAIN	TOTAL		INIT	TRAIN	TOTAL
PARK	0.760±0.005	6±1	71±9	77±10	31.5±0.2 %	9±1	55±6	64±6
FALKON	0.758±0.005	-	-	334±2	31.5±0.2 %	-	-	391±5
D&C-FALK	0.834±0.005	-	-	27±1	33.2±0.1 %	-	-	20±1
D&C	OUT OF MEMORY				OUT OF MEMORY			

Thank you!