UniGe | MaLGa

Large Scale Kernel Methods for Fun and Profit

Giacomo Meanti

Supervised by
Lorenzo Rosasco

30/05/2023

Kernel Methods for Large Scale Learning

Kernel Methods

Less is More: Nyström Computational Regularization

Fast Randomized Kernel Ridge Regression with Statistical Guarantees*

Sharp analysis of low-rank kernel matrix approximations

FALKON: An Optimal Large Scale Kernel Method

- Strong Theory

- Do not Scale

$$K = \boxed{O(n^3)}$$

UniGe | MaLGa

Kernel Methods for Large Scale Learning

Kernel Methods

- Strong Theory

Less is More: Nyström Computational Regularization

Fast Randomized Kernel Ridge Regression with Statistical Guarantees*

Sharp analysis of low-rank kernel matrix approximations

FALKON: An Optimal Large Scale Kernel Method

- Do not Scale

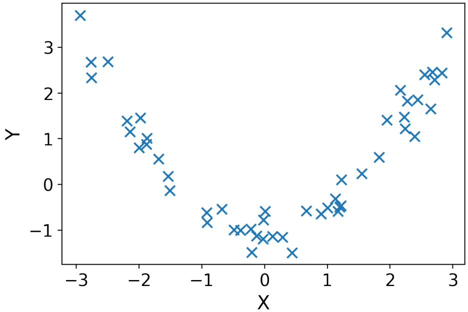
$$K = \boxed{O(n^3)}$$

Do They?

Outline

Supervised learning

Noisy data $\{(x_i, y_i)\}_{i=1}^n$, such that $y_i = \underbrace{f^*(x_i)}_{\text{true function}} + \underbrace{\epsilon_i}_{\text{random noise}}$

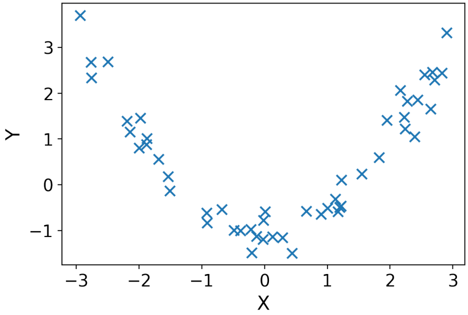


UniGe | MaLGa

Supervised learning

Noisy data $\{(x_i, y_i)\}_{i=1}^n$, such that $y_i = \underbrace{f^*(x_i)}_{\text{true function}} + \underbrace{\epsilon_i}_{\text{random noise}}$

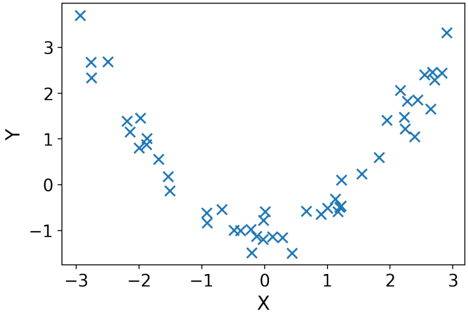
learn $\hat{f} \approx f^*$



Supervised learning

Noisy data $\{(x_i, y_i)\}_{i=1}^n$, such that $y_i = \underbrace{f^*(x_i)}_{\text{true function}} + \underbrace{\epsilon_i}_{\text{random noise}}$

learn $\hat{f} \approx f^*$

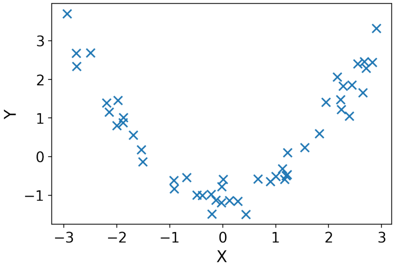


What we need

- Hypothesis space
- Error measure

Ridge Regression

- Hypothesis space: $\mathrm{LIN} = \{f | f(x) = x^\top w, w \in \mathbb{R}^d\}$, $\|f\|^2 = w^\top w$ *linear!*

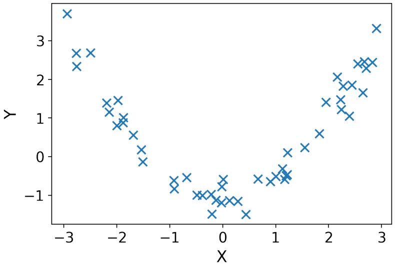


UniGe | MaLGa

Ridge Regression

- Hypothesis space: $\mathrm{LIN} = \{f | f(x) = x^\top w, w \in \mathbb{R}^d\}$, $\|f\|^2 = w^\top w$ *linear!*
- Error measure (regularized):

$$\hat{f} = \arg\min_{f \in \mathrm{LIN}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2$$

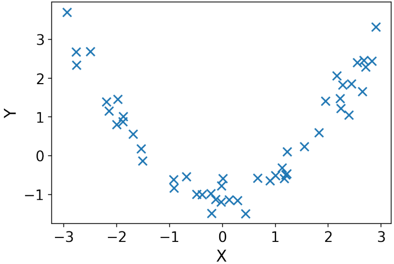


UniGe | MaLGa

Ridge Regression

- Hypothesis space: $\mathrm{LIN} = \{f | f(x) = x^\top w, w \in \mathbb{R}^d\}$, $\|f\|^2 = w^\top w$ *linear!*
- Error measure (regularized):

$$\hat{f} = \operatorname*{arg\,min}_{f \in \mathrm{LIN}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2$$



Solution

$$\hat{f}(x) = x^\top \hat{w} = x^\top (X^\top X + \lambda I)^{-1} X^\top Y$$
$$X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times d}$$

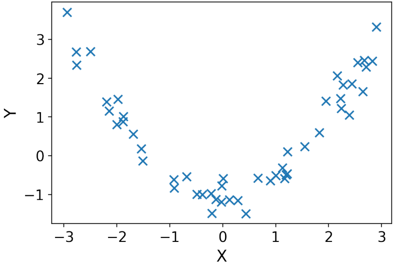
Computations

Time: **Space:**

Ridge Regression

- Hypothesis space: $\mathrm{LIN} = \{f | f(x) = x^\top w, w \in \mathbb{R}^d\}$, $\|f\|^2 = w^\top w$ *linear!*
- Error measure (regularized):

$$\hat{f} = \arg\min_{f \in \mathrm{LIN}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2$$



Solution

$$\hat{f}(x) = x^\top \hat{w} = x^\top \overbrace{(X^\top X + \lambda I)}^{d \times d}{}^{-1} X^\top Y$$

$$X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times d}$$

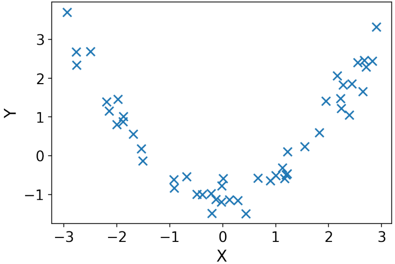
Computations

Time: $\mathcal{O}(nd^2 + d^3)$ **Space:**

Ridge Regression

- Hypothesis space: $\mathrm{LIN} = \{f | f(x) = x^\top w, w \in \mathbb{R}^d\}$, $\|f\|^2 = w^\top w$ *linear!*
- Error measure (regularized):

$$\hat{f} = \operatorname*{arg\,min}_{f \in \mathrm{LIN}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2$$



Solution

$$\hat{f}(x) = x^\top \hat{w} = x^\top (\overbrace{X^\top X}^{d \times d} + \lambda I)^{-1} X^\top Y$$

$$X = [x_1, \dots, x_n]^\top \in \mathbb{R}^{n \times d}$$

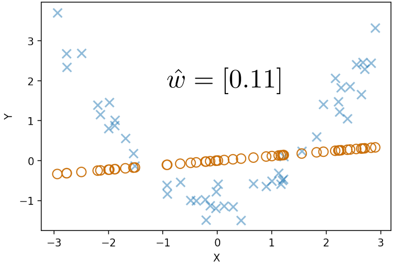
Computations

Time: $\mathcal{O}(nd^2 + d^3)$ **Space:** $\mathcal{O}(nd + d^2)$

Ridge Regression

- Hypothesis space: $\mathrm{LIN} = \{f | f(x) = x^\top w, w \in \mathbb{R}^d\}$, $\|f\|^2 = w^\top w$ *linear!*
- Error measure (regularized):

$$\hat{f} = \operatorname*{arg\,min}_{f \in \mathrm{LIN}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2$$



Solution

$$\hat{f}(x) = x^\top \hat{w} = x^\top (\overbrace{X^\top X}^{d \times d} + \lambda I)^{-1} X^\top Y$$

$$X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times d}$$

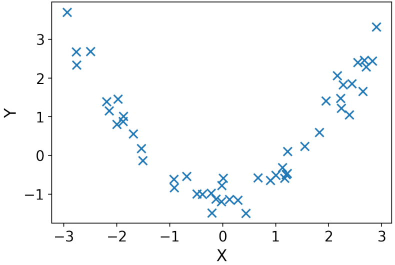
Computations

Time: $\mathcal{O}\left(nd^2 + d^3\right)$ **Space:** $\mathcal{O}\left(nd + d^2\right)$

UniGe | MaLGa

Non-linear Ridge Regression

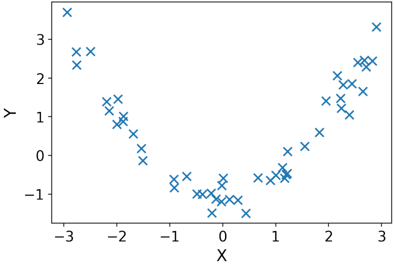
Non-linear transformation $\phi(x) \mapsto [x^2, x, 1] \in \mathbb{R}^3$



Non-linear Ridge Regression

Non-linear transformation $\phi(x) \mapsto [x^2, x, 1] \in \mathbb{R}^3$

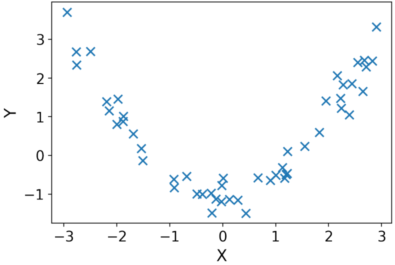
- Hypothesis space: $\mathrm{LIN}_\phi = \{f | f(x) = \phi(x)^\top w, w \in \mathbb{R}^p\}$, $\|f\|^2 = w^\top w$
- Error measure: $\hat{f} = \arg\min_{f \in \mathrm{LIN}_\phi} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Non-linear Ridge Regression

Non-linear transformation $\phi(x) \mapsto [x^2, x, 1] \in \mathbb{R}^3$

- Hypothesis space: $\mathrm{LIN}_\phi = \{f | f(x) = \phi(x)^\top w, w \in \mathbb{R}^p\}$, $\|f\|^2 = w^\top w$
- Error measure: $\hat{f} = \arg\min_{f \in \mathrm{LIN}_\phi} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Solution

$$\hat{f}(x) = \phi(x)^\top \hat{w} = \phi(x)^\top (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top Y$$

$$\Phi = [\phi(x_1), \ldots, \phi(x_n)]^\top \in \mathbb{R}^{n \times p}$$

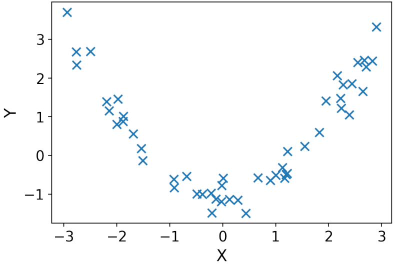
Computations

Time: **Space:**

Non-linear Ridge Regression

Non-linear transformation $\phi(x) \mapsto [x^2, x, 1] \in \mathbb{R}^3$

- Hypothesis space: $\mathrm{LIN}_\phi = \{f | f(x) = \phi(x)^\top w, w \in \mathbb{R}^p\}$, $\|f\|^2 = w^\top w$
- Error measure: $\hat{f} = \arg\min_{f \in \mathrm{LIN}_\phi} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Solution

$$\hat{f}(x) = \phi(x)^\top \hat{w} = \phi(x)^\top (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top Y$$

$$\Phi = [\phi(x_1), \ldots, \phi(x_n)]^\top \in \mathbb{R}^{n \times p}$$

Computations

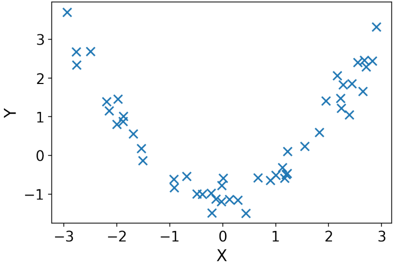
Time: $\mathcal{O}\left(np^2 + p^3\right)$ **Space:**

UniGe | MaLGa

Non-linear Ridge Regression

Non-linear transformation $\phi(x) \mapsto [x^2, x, 1] \in \mathbb{R}^3$

- Hypothesis space: $\mathrm{LIN}_\phi = \{f | f(x) = \phi(x)^\top w, w \in \mathbb{R}^p\}$, $\|f\|^2 = w^\top w$
- Error measure: $\hat{f} = \arg\min_{f \in \mathrm{LIN}_\phi} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Solution

$$\hat{f}(x) = \phi(x)^\top \hat{w} = \phi(x)^\top (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top Y$$

$$\Phi = [\phi(x_1), \ldots, \phi(x_n)]^\top \in \mathbb{R}^{n \times p}$$

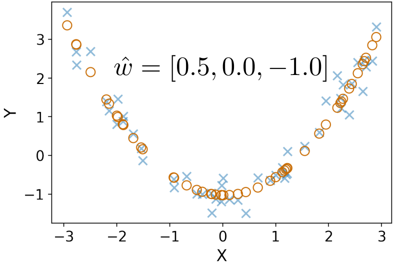
Computations

Time: $\mathcal{O}\left(np^2 + p^3\right)$ **Space:** $\mathcal{O}\left(np + p^2\right)$

Non-linear Ridge Regression

Non-linear transformation $\phi(x) \mapsto [x^2, x, 1] \in \mathbb{R}^3$

- Hypothesis space: $\mathrm{LIN}_\phi = \{f | f(x) = \phi(x)^\top w, w \in \mathbb{R}^p\}$, $\|f\|^2 = w^\top w$
- Error measure: $\hat{f} = \arg\min_{f \in \mathrm{LIN}_\phi} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Solution

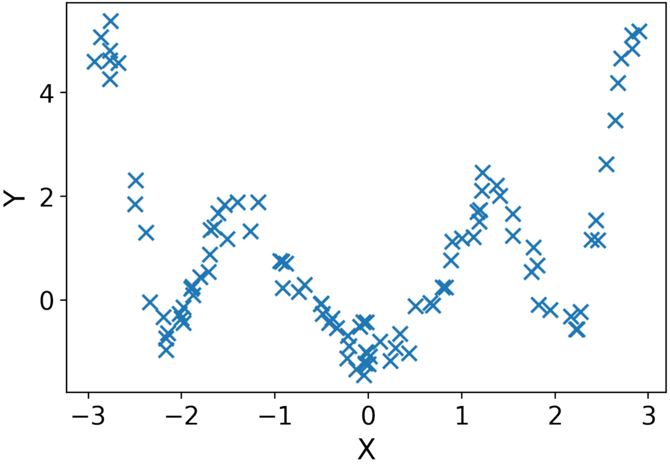
$$\hat{f}(x) = \phi(x)^\top \hat{w} = \phi(x)^\top (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top Y$$

$$\Phi = [\phi(x_1), \ldots, \phi(x_n)]^\top \in \mathbb{R}^{n \times p}$$

Computations

Time: $\mathcal{O}\left(np^2 + p^3\right)$ **Space:** $\mathcal{O}\left(np + p^2\right)$

Another Non-linear Dataset?



Kernel Ridge Regression

Infinite dimensional, non-linear $\phi(x) \in \mathcal{H}$ such that $\underbrace{\phi(x)^\top \phi(x') =: k(x, x') \in \mathbb{R}}_{\text{kernel function}}$

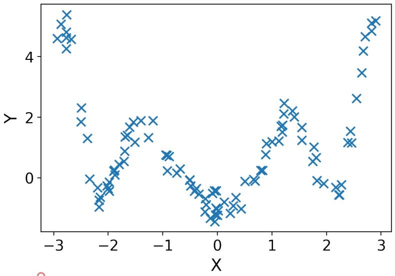
1. k symmetric ($k(x_i, x_j) = k(x_j, x_i)$)
2. k must be a positive semi-definite kernel

- RBF Kernel $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$

Kernel Ridge Regression

Infinite dimensional, non-linear $\phi(x) \in \mathcal{H}$ such that $\underbrace{\phi(x)^\top \phi(x') =: k(x, x') \in \mathbb{R}}_{\text{kernel function}}$

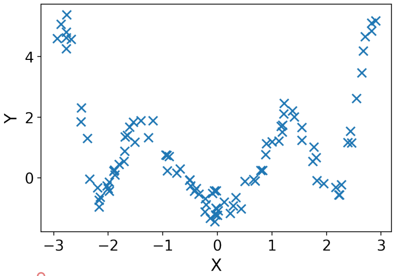
- Hypothesis space:
  $\mathcal{H} = \{f | f(x) = \sum_{i=1}^n \alpha_i k(x, x_i), \alpha \in \mathbb{R}^n\}, \quad \|f\|^2 = \sum_{i,j=1}^n \alpha_i k(x_i, x_j) \alpha_j$
- Error measure: $\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Kernel Ridge Regression

Infinite dimensional, non-linear $\phi(x) \in \mathcal{H}$ such that $\underbrace{\phi(x)^\top \phi(x') =: k(x, x') \in \mathbb{R}}_{\text{kernel function}}$

- Hypothesis space:
  $\mathcal{H} = \{f | f(x) = \sum_{i=1}^n \alpha_i k(x, x_i), \alpha \in \mathbb{R}^n\}, \quad \|f\|^2 = \sum_{i,j=1}^n \alpha_i k(x_i, x_j) \alpha_j$
- Error measure: $\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Solution (Wahba 1990)

$$\hat{f}(x) = k(x, X)\hat{\alpha} = k(x, X)(K + \lambda I)^{-1} Y$$

$$K \in \mathbb{R}^{n \times n}, \quad K_{ij} = k(x_i, x_j), \quad k(x, X) \in \mathbb{R}^n$$

Computations

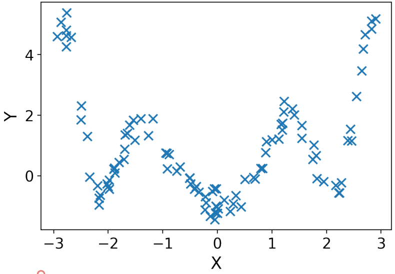
Time: **Space:**

UniGe | MaLGa

Kernel Ridge Regression

Infinite dimensional, non-linear $\phi(x) \in \mathcal{H}$ such that $\underbrace{\phi(x)^\top \phi(x') =: k(x, x') \in \mathbb{R}}_{\text{kernel function}}$

- Hypothesis space:
  $\mathcal{H} = \{f | f(x) = \sum_{i=1}^n \alpha_i k(x, x_i), \alpha \in \mathbb{R}^n\}, \quad \|f\|^2 = \sum_{i,j=1}^n \alpha_i k(x_i, x_j) \alpha_j$
- Error measure: $\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Solution (Wahba 1990)

$$\hat{f}(x) = k(x, X)\hat{\alpha} = k(x, X)(K + \lambda I)^{-1} Y$$

$$K \in \mathbb{R}^{n \times n}, \quad K_{ij} = k(x_i, x_j), \quad k(x, X) \in \mathbb{R}^n$$

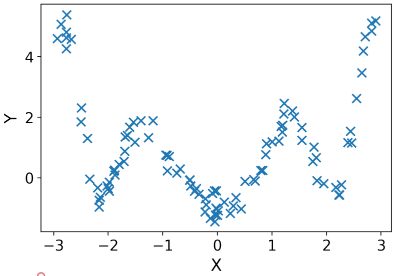
Computations

Time: $\mathcal{O}(n^3)$ **Space:**

Kernel Ridge Regression

Infinite dimensional, non-linear $\phi(x) \in \mathcal{H}$ such that $\underbrace{\phi(x)^\top \phi(x') =: k(x, x') \in \mathbb{R}}_{\text{kernel function}}$

- Hypothesis space:
  $\mathcal{H} = \{f | f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i), \alpha \in \mathbb{R}^n\}, \quad \|f\|^2 = \sum_{i,j=1}^{n} \alpha_i k(x_i, x_j) \alpha_j$
- Error measure: $\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2$



Solution (Wahba 1990)

$$\hat{f}(x) = k(x, X)\hat{\alpha} = k(x, X)(K + \lambda I)^{-1} Y$$

$$K \in \mathbb{R}^{n \times n}, \quad K_{ij} = k(x_i, x_j), \quad k(x, X) \in \mathbb{R}^n$$

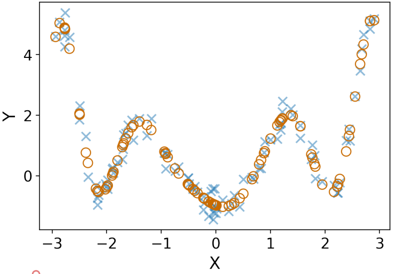
Computations

Time: $\mathcal{O}(n^3)$ **Space:** $\mathcal{O}(n^2)$

Kernel Ridge Regression

Infinite dimensional, non-linear $\phi(x) \in \mathcal{H}$ such that $\underbrace{\phi(x)^\top \phi(x') =: k(x, x') \in \mathbb{R}}_{\text{kernel function}}$

- Hypothesis space:
  $\mathcal{H} = \{f | f(x) = \sum_{i=1}^n \alpha_i k(x, x_i), \alpha \in \mathbb{R}^n\}, \quad \|f\|^2 = \sum_{i,j=1}^n \alpha_i k(x_i, x_j) \alpha_j$
- Error measure: $\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|^2$



Solution (Wahba 1990)

$$\hat{f}(x) = k(x, X)\hat{\alpha} = k(x, X)(K + \lambda I)^{-1} Y$$

$$K \in \mathbb{R}^{n \times n}, \quad K_{ij} = k(x_i, x_j), \quad k(x, X) \in \mathbb{R}^n$$

Computations

Time: $\mathcal{O}(n^3)$ **Space:** $\mathcal{O}(n^2)$

KRR: Summary

$$\text{KRR:} \quad \hat{\alpha} = (K + \lambda I)^{-1} Y, \qquad \hat{f}_\lambda = \sum_{i=1}^{n} \hat{\alpha}_i k(x, x_i)$$

$$\textbf{kernel matrix } K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

Linear system

$$\boxed{K + \lambda I} \; \boxed{\hat{\alpha}} = \boxed{Y}$$

KRR: Summary

KRR: $\hat{\alpha} = (K + \lambda I)^{-1} Y, \qquad \hat{f}_\lambda = \sum_{i=1}^{n} \hat{\alpha}_i k(x, x_i)$

kernel matrix $K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \dots & k(x_n,x_n) \end{bmatrix}$

Linear system

$$\boxed{K + \lambda I}\ \boxed{\hat{\alpha}} = \boxed{Y}$$

✔ Can learn many non-linear functions easily

KRR: Summary

$$\text{KRR:} \quad \hat{\alpha} = (K + \lambda I)^{-1} Y, \qquad \hat{f}_\lambda = \sum_{i=1}^{n} \hat{\alpha}_i k(x, x_i)$$

kernel matrix $K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$

Linear system

$$\boxed{K + \lambda I} \; \boxed{\hat{\alpha}} = \boxed{Y}$$

- ✔ Can learn many non-linear functions easily
- ✔ It is *similar* to linear models, so easy to prove stuff!

KRR: Summary

$$\text{KRR:} \quad \hat{\alpha} = (K + \lambda I)^{-1} Y, \qquad \hat{f}_\lambda = \sum_{i=1}^{n} \hat{\alpha}_i k(x, x_i)$$

kernel matrix $K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \dots & k(x_n,x_n) \end{bmatrix}$

Linear system

$$\boxed{K + \lambda I}\ \boxed{\hat{\alpha}} = \boxed{Y}$$

- ✔ Can learn many non-linear functions easily
- ✔ It is *similar* to linear models, so easy to prove stuff!
- ✘ Finding $\hat{\alpha}$ scales poorly with big data (large n)

Outline

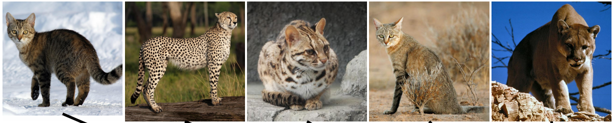
Background

- Introduction to Kernel Methods
- Falkon 1.0

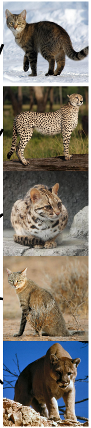
Contributions

- Falkon 2.0 – Large Scale KRR
- Hyperparameter Tuning for Falkon 2.0
- Falkon Applications

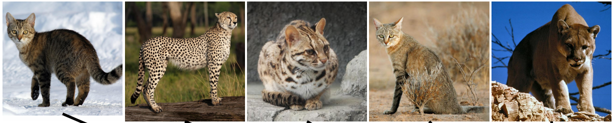
The Nyström Approximation

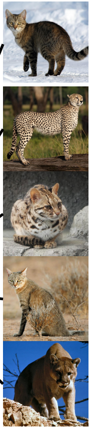


$$K = \begin{bmatrix} 1.0 & 0.1 & 0.6 & 0.8 & 0.1 \\ 0.1 & 1.0 & 0.1 & 0.1 & 0.7 \\ 0.6 & 0.1 & 1.0 & 0.6 & 0.1 \\ 0.8 & 0.1 & 0.6 & 1.0 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$



The Nyström Approximation

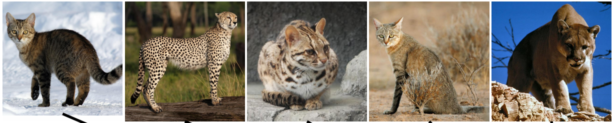




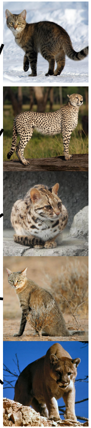
$$K = \begin{bmatrix} 1.0 & 0.1 & 0.6 & 0.8 & 0.1 \\ 0.1 & 1.0 & 0.1 & 0.1 & 0.7 \\ 0.6 & 0.1 & ?? & ?? & ?? \\ 0.8 & 0.1 & ?? & ?? & ?? \\ 0.1 & 0.7 & ?? & ?? & ?? \end{bmatrix}$$

UniGe | MaLGa

The Nyström Approximation

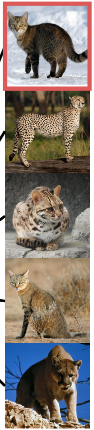


$$K = \begin{bmatrix} 1.0 & 0.1 & 0.6 & 0.8 & 0.1 \\ 0.1 & 1.0 & 0.1 & 0.1 & 0.7 \\ 0.6 & 0.1 & ?? & ?? & {\color{red}??} \\ 0.8 & 0.1 & ?? & ?? & ?? \\ 0.1 & 0.7 & ?? & ?? & ?? \end{bmatrix}$$



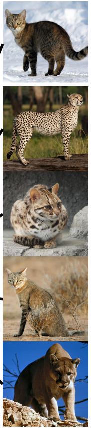
The Nyström Approximation

$$K = \begin{bmatrix} 1.0 & 0.1 & 0.6 & 0.8 & 0.1 \\ 0.1 & 1.0 & 0.1 & 0.1 & 0.7 \\ 0.6 & 0.1 & ?? & ?? & ?? \\ 0.8 & 0.1 & ?? & ?? & ?? \\ 0.1 & 0.7 & ?? & ?? & ?? \end{bmatrix}$$



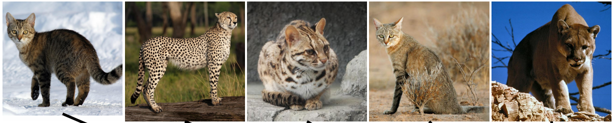
The Nyström Approximation

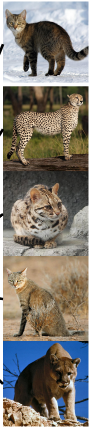
$$K = \begin{bmatrix} 1.0 & 0.1 & 0.6 & 0.8 & 0.1 \\ 0.1 & 1.0 & 0.1 & 0.1 & 0.7 \\ 0.6 & 0.1 & ?? & ?? & ?? \\ 0.8 & 0.1 & ?? & ?? & ?? \\ 0.1 & 0.7 & ?? & ?? & ?? \end{bmatrix}$$



$$K = \begin{bmatrix} A & S \\ S^\top & Q \end{bmatrix}$$

The Nyström Approximation





$$K = \begin{bmatrix} 1.0 & 0.1 & 0.6 & 0.8 & 0.1 \\ 0.1 & 1.0 & 0.1 & 0.1 & 0.7 \\ 0.6 & 0.1 & ?? & ?? & ?? \\ 0.8 & 0.1 & ?? & ?? & ?? \\ 0.1 & 0.7 & ?? & ?? & ?? \end{bmatrix}$$

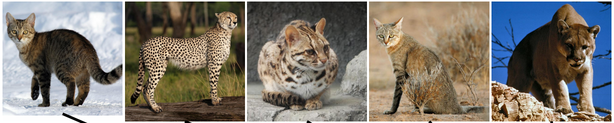
$$K = \begin{bmatrix} A & S \\ S^\top & Q \end{bmatrix}$$

Nyström approximation:

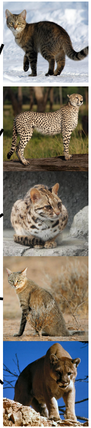
$$Q \approx S^\top A^{-1} S$$

Williams, Seeger (2000)

The Nyström Approximation



$$K = \begin{bmatrix} 1.0 & 0.1 & 0.6 & 0.8 & 0.1 \\ 0.1 & 1.0 & 0.1 & 0.1 & 0.7 \\ 0.6 & 0.1 & 1.0 & 0.6 & 0.1 \\ 0.8 & 0.1 & 0.6 & 1.0 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$



$$K = \begin{bmatrix} A & S \\ S^\top & Q \end{bmatrix}$$

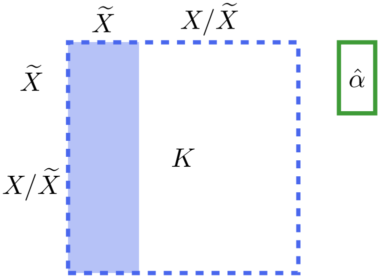
Nyström approximation:

$$Q \approx S^\top A^{-1} S$$

Williams, Seeger (2000)

Nyström KRR

1. Choose $m \ll n$ inducing points $\widetilde{X} \subset X$

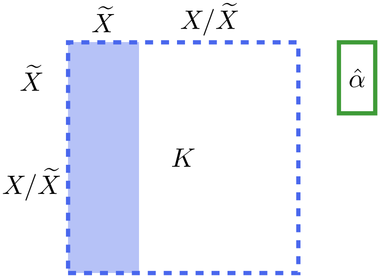


Nyström KRR

1. Choose $m \ll n$ inducing points $\widetilde{X} \subset X$
2. New small hypothesis space:

$$\mathcal{H}_m = \left\{ f | f(x) = \sum_{i=1}^{m} \alpha_i k(x, \widetilde{x}_i), \alpha \in \mathbb{R}^m \right\}$$

$$\|f\|^2 = \sum_{i,j=1}^{m} \alpha_i k(\widetilde{x}_i, \widetilde{x}_j) \alpha_j$$



UniGe | MaLGa

Nyström KRR

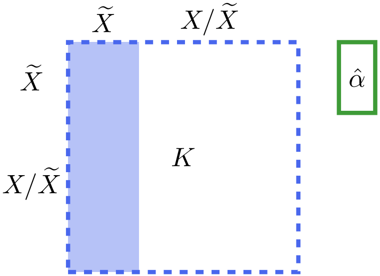
1. Choose $m \ll n$ inducing points $\widetilde{X} \subset X$
2. New small hypothesis space:

$$\mathcal{H}_m = \left\{ f | f(x) = \sum_{i=1}^{m} \alpha_i k(x, \widetilde{x}_i), \alpha \in \mathbb{R}^m \right\}$$

$$\|f\|^2 = \sum_{i,j=1}^{m} \alpha_i k(\widetilde{x}_i, \widetilde{x}_j)\alpha_j$$

3. Same error measure

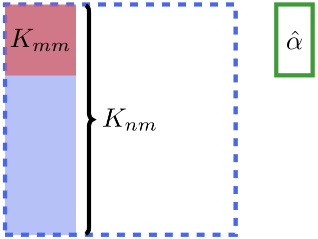
$$\hat{f} = \arg\min_{f \in \mathcal{H}_m} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|^2$$



Nyström KRR

The solution can be shown to be (Rudi et al. 2015)

$$\boxed{\hat{f}(x) = \sum_{i=1}^{m} \hat{\alpha}_i k(\widetilde{x}_i, x), \qquad \hat{\alpha} = (K_{mn}K_{nm} + \lambda K_{mm})^{-1} K_{mn} Y}$$



Complexity

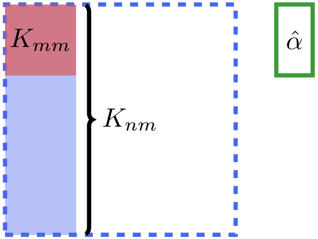
Time:

Space:

Nyström KRR

The solution can be shown to be (Rudi et al. 2015)

$$\hat{f}(x) = \sum_{i=1}^{m} \hat{\alpha}_i k(\widetilde{x}_i, x), \qquad \hat{\alpha} = (K_{mn} K_{nm} + \lambda K_{mm})^{-1} K_{mn} Y$$



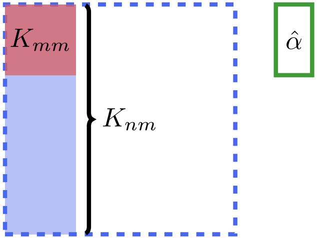
Complexity

Time: $\mathcal{O}\left(nm^2 + \quad\right)$ **Space:**

Nyström KRR

The solution can be shown to be (Rudi et al. 2015)

$$\boxed{\hat{f}(x) = \sum_{i=1}^{m} \hat{\alpha}_i k(\widetilde{x}_i, x), \qquad \hat{\alpha} = (K_{mn}K_{nm} + \lambda K_{mm})^{-1} K_{mn} Y}$$



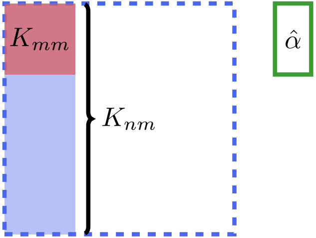
Complexity

Time: $\mathcal{O}\left(nm^2 + m^3\right)$ **Space:**

Nyström KRR

The solution can be shown to be (Rudi et al. 2015)

$$\boxed{\hat{f}(x) = \sum_{i=1}^{m} \hat{\alpha}_i k(\widetilde{x}_i, x), \qquad \hat{\alpha} = (K_{mn} K_{nm} + \lambda K_{mm})^{-1} K_{mn} Y}$$



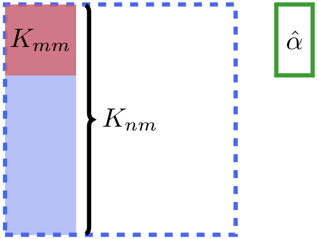
Complexity

Time: $\mathcal{O}\left(nm^2 + m^3\right)$ **Space:** $\mathcal{O}(nm)$

Nyström KRR

The solution can be shown to be (Rudi et al. 2015)

$$\boxed{\hat{f}(x) = \sum_{i=1}^{m} \hat{\alpha}_i k(\widetilde{x}_i, x), \qquad \hat{\alpha} = (K_{mn}K_{nm} + \lambda K_{mm})^{-1} K_{mn} Y}$$



Complexity

Time: $\mathcal{O}\left(nm^2 + m^3\right)$ **Space:** $\mathcal{O}(nm)$

Better solvers?

Falkon: Efficient Preconditioning

Iterative solver (gradient descent)

$$\hat{\alpha}_t = \hat{\alpha}_{t-1} - \gamma\Big[\,(K_{mn}K_{nm} + \lambda K_{mm})\,\hat{\alpha}_{t-1} - K_{nm}Y\Big]$$

Falkon: Efficient Preconditioning

Iterative solver (gradient descent)

$$\hat{\alpha}_t = \hat{\alpha}_{t-1} - \gamma \Big[PP^\top (K_{mn} K_{nm} + \lambda K_{mm})\, \hat{\alpha}_{t-1} - PP^\top K_{nm} Y \Big]$$

Ideal preconditioner

$$PP^\top = (K_{mn} K_{nm} + \lambda K_{mm})^{-1}$$

Falkon: Efficient Preconditioning

Iterative solver (gradient descent)

$$\hat{\alpha}_t = \hat{\alpha}_{t-1} - \gamma \left[\underbrace{PP^\top (K_{mn} K_{nm} + \lambda K_{mm})}_{=I} \hat{\alpha}_{t-1} - PP^\top K_{nm} Y \right]$$

Ideal preconditioner

$$PP^\top = (K_{mn} K_{nm} + \lambda K_{mm})^{-1}$$

Falkon: Efficient Preconditioning

Iterative solver (gradient descent)

$$\hat{\alpha}_t = \hat{\alpha}_{t-1} - \gamma\Big[\underbrace{PP^\top(K_{mn}K_{nm} + \lambda K_{mm})}_{=I}\hat{\alpha}_{t-1} - PP^\top K_{nm}Y\Big]$$

Ideal preconditioner *Expensive*

$$\cancel{PP^\top = (K_{mn}K_{nm} + \lambda K_{mm})^{-1}}$$

Falkon: Efficient Preconditioning

Iterative solver (gradient descent)

$$\hat{\alpha}_t = \hat{\alpha}_{t-1} - \gamma \Big[PP^\top (K_{mn}K_{nm} + \lambda K_{mm})\,\hat{\alpha}_{t-1} - PP^\top K_{nm} Y \Big]$$

Ideal preconditioner *Expensive*

$$\xcancel{PP^\top = (K_{mn}K_{nm} + \lambda K_{mm})^{-1}}$$

Efficient preconditioner $\rightarrow$ Falkon (Rudi et al. 2017)

$$PP^T = (K_{mm}^2 + \lambda K_{mm})^{-1}$$

Statistics vs. Computations

- KRR (Caponnetto, De Vito (2007)): if $f^* \in \mathcal{H}$,

$$\lambda_* = \frac{1}{\sqrt{n}} \qquad \frac{1}{\sqrt{n}} \lesssim \underbrace{\mathbb{E}\left[(\hat{f}(x) - f^*(x))^2\right]}_{\text{generalization error}} \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n^3)$ **Space** $\mathcal{O}(n^2)$

UniGe | MaLGa

Statistics vs. Computations

- KRR (Caponnetto, De Vito (2007)): if $f^* \in \mathcal{H}$,

$$\lambda_* = \frac{1}{\sqrt{n}} \qquad \frac{1}{\sqrt{n}} \lesssim \underbrace{\mathbb{E}\left[(\hat{f}(x) - f^*(x))^2\right]}_{\text{generalization error}} \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n^3)$ **Space** $\mathcal{O}(n^2)$

Statistics vs. Computations

- KRR (Caponnetto, De Vito (2007)): if $f^* \in \mathcal{H}$,

$$\lambda_* = \frac{1}{\sqrt{n}} \qquad \frac{1}{\sqrt{n}} \lesssim \underbrace{\mathbb{E}\left[(\hat{f}(x) - f^*(x))^2\right]}_{\text{generalization error}} \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n^3)$ **Space** $\mathcal{O}(n^2)$

Statistics vs. Computations

- KRR (Caponnetto, De Vito (2007)): if $f^* \in \mathcal{H}$,

$$\lambda_* = \frac{1}{\sqrt{n}} \qquad \frac{1}{\sqrt{n}} \lesssim \underbrace{\mathbb{E}\Big[(\hat{f}(x) - f^*(x))^2\Big]}_{\text{generalization error}} \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n^3)$ Space $\mathcal{O}(n^2)$

- Falkon (Rudi et al. (2017)): if $f^* \in \mathcal{H}$, $m \geq \mathcal{O}(\sqrt{n})$

$$\lambda_* = \frac{1}{\sqrt{n}}, m_* = \sqrt{n} \qquad \mathbb{E}\Big[(\hat{f}(x) - f^*(x))^2\Big] \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n\sqrt{n})$ Space $\mathcal{O}(n)$

Statistics vs. Computations

- KRR (Caponnetto, De Vito (2007)): if $f^* \in \mathcal{H}$,

$$\lambda_* = \frac{1}{\sqrt{n}} \qquad \frac{1}{\sqrt{n}} \lesssim \underbrace{\mathbb{E}\left[(\hat{f}(x) - f^*(x))^2\right]}_{\text{generalization error}} \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n^3)$ Space $\mathcal{O}(n^2)$

- Falkon (Rudi et al. (2017)): if $f^* \in \mathcal{H}$, $m \geq \mathcal{O}(\sqrt{n})$

$$\lambda_* = \frac{1}{\sqrt{n}}, m_* = \sqrt{n} \qquad \mathbb{E}\left[(\hat{f}(x) - f^*(x))^2\right] \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n\sqrt{n})$ Space $\mathcal{O}(n)$

Statistics vs. Computations

- KRR (Caponnetto, De Vito (2007)): if $f^* \in \mathcal{H}$,

$$\lambda_* = \frac{1}{\sqrt{n}} \qquad \frac{1}{\sqrt{n}} \lesssim \underbrace{\mathbb{E}\left[(\hat{f}(x) - f^*(x))^2\right]}_{\text{generalization error}} \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n^3)$ **Space** $\mathcal{O}(n^2)$

- Falkon (Rudi et al. (2017)): if $f^* \in \mathcal{H}$, $m \geq \mathcal{O}(\sqrt{n})$

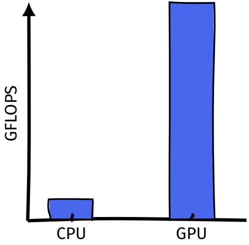
$$\lambda_* = \frac{1}{\sqrt{n}}, m_* = \sqrt{n} \qquad \mathbb{E}\left[(\hat{f}(x) - f^*(x))^2\right] \lesssim \frac{1}{\sqrt{n}}$$

Time $\mathcal{O}(n\sqrt{n})$ **Space** $\mathcal{O}(n)$

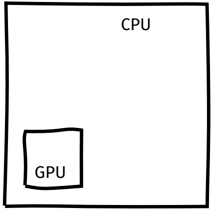
From Theory to Practice

Scalability

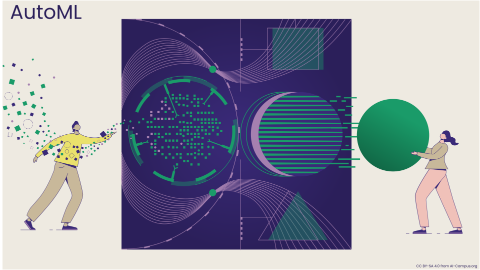
Computations



Memory



Flexibility & User Friendliness

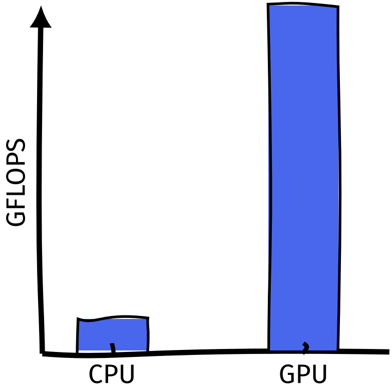


Hutter et al. 2019

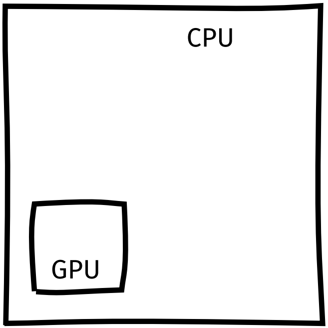
Outline

From Theory to Practice: Scalability

Computations



Memory



Scaling up Falkon

GPU

- out-of-core operations
- multi-GPU support

Memory

- preconditioner
- matrix-vector products
- low precision floats

Communications

- Optimized memory transfers

$\boxed{20\times \text{ Improvement}}$

Scaling up Falkon

GPU

- out-of-core operations
- multi-GPU support

Memory

- preconditioner
- matrix-vector products
- low precision floats

Communications

- Optimized memory transfers

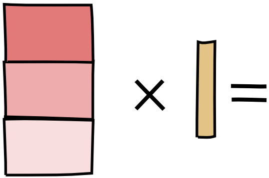
$20\times$ Improvement

Memory: Matrix-Vector Products

K_{nm} $\boldsymbol{v}$

- Kernel-vector products

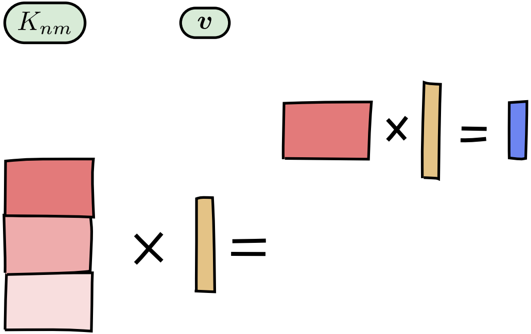
$$\underbrace{K_{nm}}_{\text{Very large}} \boldsymbol{v} = \underbrace{\boldsymbol{c}}_{\text{Small}}$$



Memory: Matrix-Vector Products

- Kernel-vector products

$$\underbrace{K_{nm}}_{\text{Very large}} \boldsymbol{v} = \underbrace{\boldsymbol{c}}_{\text{Small}}$$

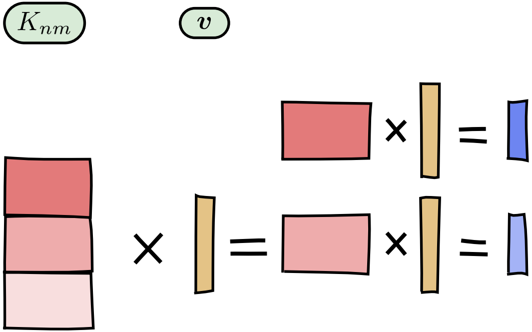


UniGe | MaLGa

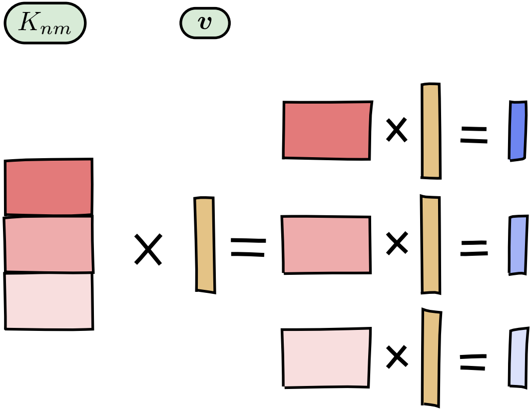
Memory: Matrix-Vector Products

- Kernel-vector products

$$\underbrace{K_{nm}}_{\text{Very large}} \boldsymbol{v} = \underbrace{\boldsymbol{c}}_{\text{Small}}$$



Memory: Matrix-Vector Products



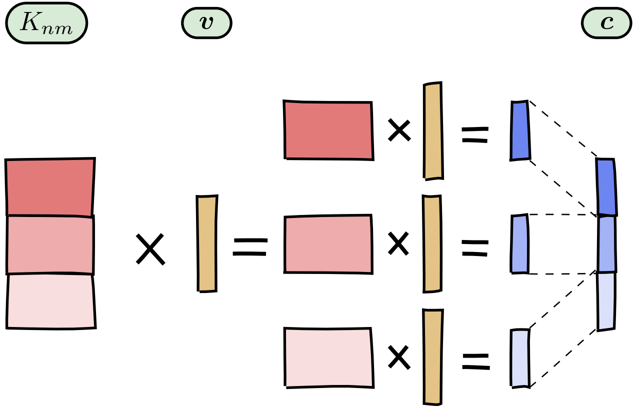
- Kernel-vector products

$$\underbrace{K_{nm}}_{\text{Very large}} \; \boldsymbol{v} = \underbrace{\boldsymbol{c}}_{\text{Small}}$$

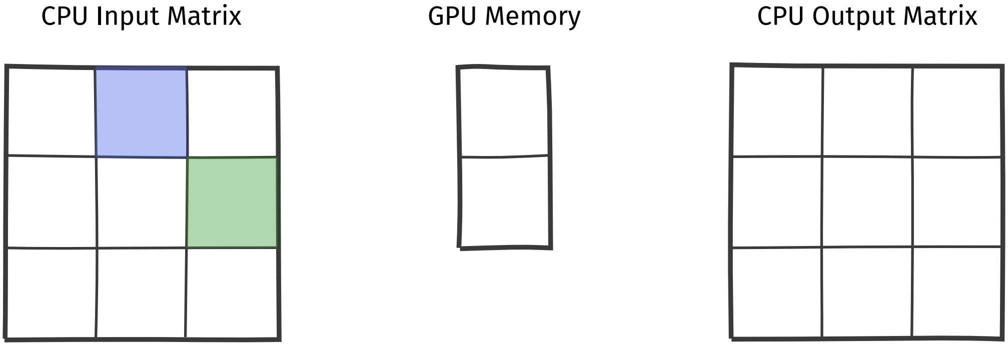
Memory: Matrix-Vector Products

- Kernel-vector products

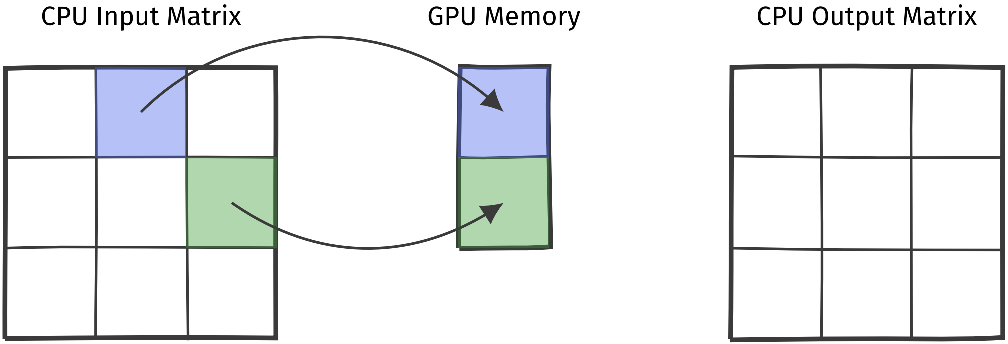
$$\underbrace{K_{nm}}_{\text{Very large}} \boldsymbol{v} = \underbrace{\boldsymbol{c}}_{\text{Small}}$$



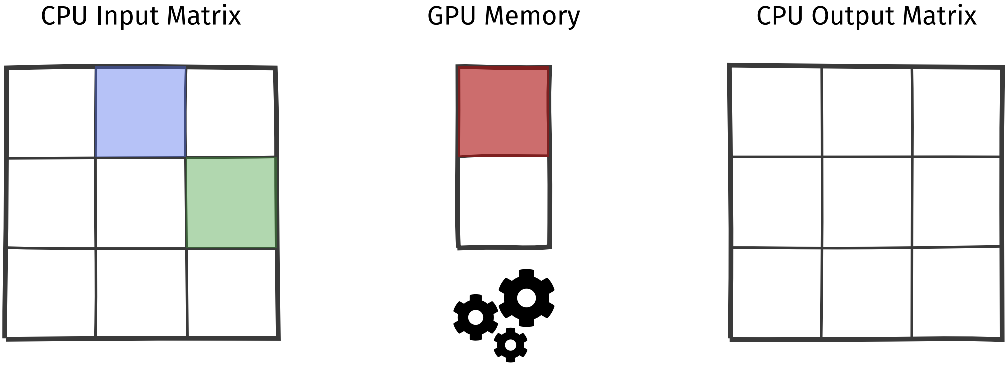
GPU: Out-of-core Operations



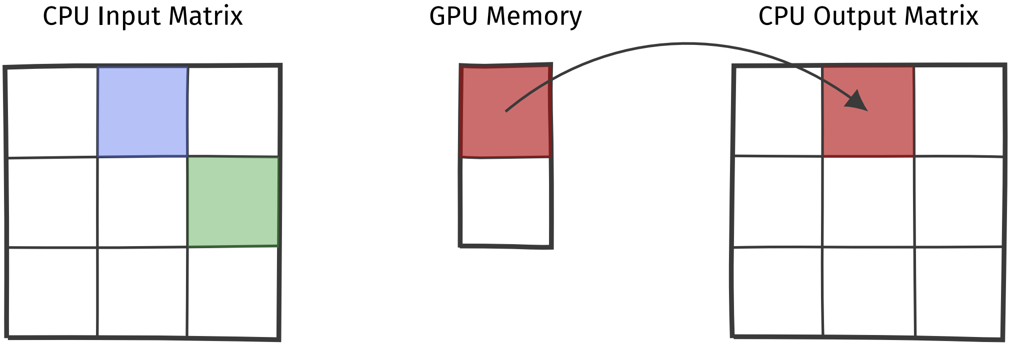
GPU: Out-of-core Operations



GPU: Out-of-core Operations



GPU: Out-of-core Operations



Falkon 1.0 vs 2.0

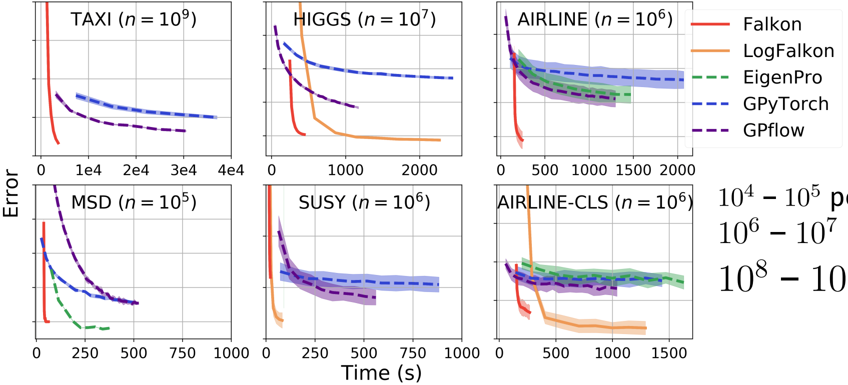
Experiment	Preconditioner		Iterations	
	Time	Improvement	Time	Improvement
Baseline Rudi et al. (2017)	2337 s	–	4565 s	–
Float32 precision	1306 s	1.8×	1496 s	3×
GPU preconditioner	179 s	7.3×	1344 s	1.1×
2 GPUs	**118 s**	1.5×	693 s	1.9×
KeOps Charlier et al. (2020)	**119 s**	1×	**232 s**	3×
Improvement M. et al. (2020)		19.7×		18.8×

UniGe | MaLGa

$10^4 - 10^5$ **Points In Seconds**

	MNIST $n = 6 \cdot 10^4, d = 780$	CIFAR10 $n = 6 \cdot 10^4, d = 1024$	SVHN $n = 7 \cdot 10^4, d = 1024$
`InCoreFalkon 2.0`	**6.5 s**	**7.9 s**	**6.7 s**
`Falkon 2.0`	10.9 s	13.7 s	17.2 s
`ThunderSVM` Wen et al. (2018)	19.6 s	82.9 s	166.4 s

Going Big



$10^4 - 10^5$ points in seconds

$10^6 - 10^7$ points in minutes

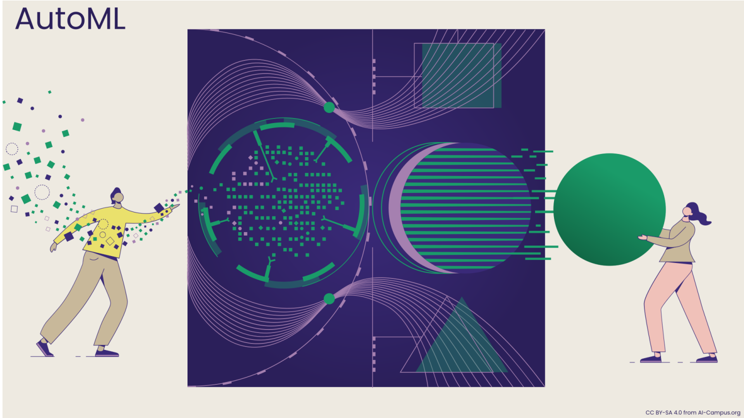
$10^8 - 10^9$ points in hours

UniGe | MaLGa

Outline

UniGe | MaLGa

From Theory to Practice: Flexibility & User Friendliness



Falkon's Hyperparameters

$$\hat{f}(x) = \sum_{i=1}^{m} \hat{\alpha}_i k(\widetilde{x}_i, x), \qquad \hat{\alpha} = (K_{mn}K_{nm} + \lambda K_{mm})^{-1} K_{mn} Y$$

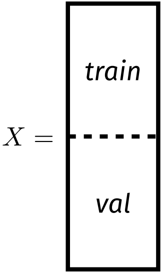
Hyperparameters

- regularization λ
- kernel parameters, e.g. $k(x, x') = \exp -\gamma \|x - x'\|^2$
- inducing points m, or $\{\widetilde{x}_i\}_{i=1}^m$

θ

Cross Validation

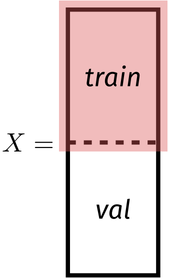
$$\hat{f}_\theta = \underset{f_\theta \in \mathcal{H}_m}{\arg\min} \sum_{i \in \text{train}} (y_i - f_\theta(x_i))^2 + \lambda \|f_\theta\|^2$$



Arlot, Celisse (2018)

Cross Validation

$$\hat{f}_\theta = \arg\min_{f_\theta \in \mathcal{H}_m} \sum_{i \in \text{train}} (y_i - f_\theta(x_i))^2 + \lambda \|f_\theta\|^2$$

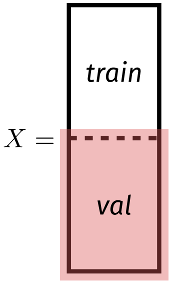


Arlot, Celisse (2018)

Cross Validation

$$\hat{f}_\theta = \arg\min_{f_\theta \in \mathcal{H}_m} \sum_{i \in \text{train}} (y_i - f_\theta(x_i))^2 + \lambda \|f_\theta\|^2$$

$$\hat{\theta} = \arg\min_{\theta} \sum_{i \in \text{val}} (y_i - \hat{f}_\theta(x_i))^2$$



Arlot, Celisse (2018)

UniGe | MaLGa

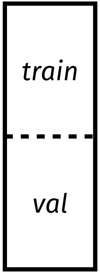
Cross Validation

$$\hat{f}_\theta = \arg\min_{f_\theta \in \mathcal{H}_m} \sum_{i \in \text{train}} (y_i - f_\theta(x_i))^2 + \lambda \|f_\theta\|^2$$

$$\hat{\theta} = \arg\min_{\theta} \sum_{i \in \text{val}} (y_i - \hat{f}_\theta(x_i))^2$$

Bilevel problem

$X =$



Arlot, Celisse (2018)

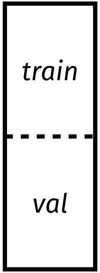
Cross Validation

$$\hat{f}_\theta = \arg\min_{f_\theta \in \mathcal{H}_m} \sum_{i \in \text{train}} (y_i - f_\theta(x_i))^2 + \lambda \|f_\theta\|^2$$

$$\hat{\theta} = \arg\min_{\theta} \sum_{i \in \text{val}} (y_i - \hat{f}_\theta(x_i))^2$$

Bilevel problem

$X =$



- ✔ Easy to implement
- ✔ Unbiased estimator of the generalization error
- ✘ Data-splitting means $\hat{f}_\theta$ may underfit
- ✘ High variance

Arlot, Celisse (2018)

UniGe | MaLGa

Complexity Regularization

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

$$X = \boxed{\textit{train}}$$

Bartlett, Mendelson (02); Efron (04); Arlot, Bach (09)

Complexity Regularization

$$\hat{f}_\theta = \underset{f_\theta \in \mathcal{H}_m}{\arg\min} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda \|f_\theta\|^2$$

$$\hat{\theta} = \underset{\theta}{\arg\min} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \text{penalty}(\theta)$$

$$X = \boxed{\textit{train}}$$

Bartlett, Mendelson (02); Efron (04); Arlot, Bach (09)

Complexity Regularization

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

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \text{penalty}(\theta)$$

Bilevel problem

$$X = \boxed{\textit{train}}$$

Bartlett, Mendelson (02); Efron (04); Arlot, Bach (09)

Complexity Regularization

$$\hat{f}_\theta = \underset{f_\theta \in \mathcal{H}_m}{\arg\min} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda \|f_\theta\|^2$$

$$\hat{\theta} = \underset{\theta}{\arg\min} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \text{penalty}(\theta)$$

Bilevel problem

$$X = \boxed{\textit{train}}$$

✔ Avoids data splitting

? How to choose the penalty

Bartlett, Mendelson (02); Efron (04); Arlot, Bach (09)

Choosing the Penalty

Ideal objective

$$\theta^* = \arg\min_{\theta} \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_\theta(x_i) - f^*(x_i))^2\right] = \arg\min_{\theta} \mathbb{E}\Big[L(\hat{f}_\theta)\Big]$$

Choosing the Penalty

Ideal objective

$$\theta^* = \arg\min_\theta \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_\theta(x_i) - f^*(x_i))^2\right] = \arg\min_\theta \mathbb{E}\Big[L(\hat{f}_\theta)\Big]$$

- expectation wrt all estimators $\hat{f}_\theta$

Choosing the Penalty

Ideal objective

$$\theta^* = \arg\min_\theta \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n (\hat{f}_\theta(x_i) - f^*(x_i))^2\right] = \arg\min_\theta \mathbb{E}\Big[L(\hat{f}_\theta)\Big]$$

- expectation wrt all estimators $\hat{f}_\theta$
- generalization error of one estimator (*fixed design*)

Choosing the Penalty

Ideal objective

$$\theta^* = \arg\min_\theta \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n (\hat{f}_\theta(x_i) - f^*(x_i))^2\right] = \arg\min_\theta \mathbb{E}\left[L(\hat{f}_\theta)\right]$$

- expectation wrt all estimators $\hat{f}_\theta$
- generalization error of one estimator (*fixed design*)

Upper bound $\rightarrow$ Penalty

Given $\mathbb{E}[L(\hat{f}_\theta)] \leq \hat{u}(\theta, X)$: minimize $\hat{u}$

Upper Bound for Falkon

Theorem (Meanti, Carratino, De Vito, Rosasco (2022))

Under fixed-design assumptions,

$$\mathbb{E}\left[L(\hat{f}_\theta^{\mathrm{FLK}})\right] \leq 2\,\underbrace{\mathbb{E}\left[\hat{L}(f_\theta^{\mathrm{KRR}})\right]}_{\textit{data fit}}\underbrace{\left(1 + \tfrac{2}{\lambda}\operatorname{tr}\left(K - \widetilde{K}\right)\right) + \tfrac{2\sigma^2}{n}\operatorname{tr}\left((\widetilde{K} + \lambda I)^{-1}\widetilde{K}\right)}_{\textit{penalty}}$$

Upper Bound for Falkon

Theorem (Meanti, Carratino, De Vito, Rosasco (2022))

Under fixed-design assumptions,

$$\mathbb{E}\left[L(\hat{f}_\theta^{\text{FLK}})\right] \leq 2\,\underbrace{\mathbb{E}\left[\hat{L}(f_\theta^{\text{KRR}})\right]}_{\text{data fit}}\underbrace{\left(1 + \tfrac{2}{\lambda}\,\mathrm{tr}\left(K - \widetilde{K}\right)\right) + \tfrac{2\sigma^2}{n}\,\mathrm{tr}\left((\widetilde{K} + \lambda I)^{-1}\widetilde{K}\right)}_{\text{penalty}}$$

Upper Bound for Falkon

Theorem (Meanti, Carratino, De Vito, Rosasco (2022))
Under fixed-design assumptions,

$$\mathbb{E}\left[L(\hat{f}_\theta^{\mathrm{FLK}})\right] \leq 2\,\underbrace{\mathbb{E}\left[\hat{L}(f_\theta^{\mathrm{KRR}})\right]}_{\textit{data fit}}\,\underbrace{\left(1 + \tfrac{2}{\lambda}\,\mathrm{tr}\left(K - \widetilde{K}\right)\right) + \tfrac{2\sigma^2}{n}\,\mathrm{tr}\left((\widetilde{K} + \lambda I)^{-1}\widetilde{K}\right)}_{\textit{penalty}}$$

Empirical upper bound

$$\hat{u}(\theta, X) = \left[\sum_{i=1}^{n}(y_i - \hat{f}_\theta^{\mathrm{FLK}}(x_i))^2 + \lambda\|\hat{f}_\theta^{\mathrm{FLK}}\|^2\right]\left(1 + \tfrac{1}{\lambda}\,\mathrm{tr}\left(K - \widetilde{K}\right)\right) + \tfrac{1}{n}\,\mathrm{tr}\left((\widetilde{K} + \lambda I)^{-1}\widetilde{K}\right)$$

Upper Bound for Falkon

Theorem (Meanti, Carratino, De Vito, Rosasco (2022))

Under fixed-design assumptions,

$$\mathbb{E}\left[L(\hat{f}_\theta^{\mathrm{FLK}})\right] \leq 2\,\underbrace{\mathbb{E}\left[\hat{L}(f_\theta^{\mathrm{KRR}})\right]}_{\textit{data fit}}\underbrace{\left(1+\tfrac{2}{\lambda}\,\mathrm{tr}\left(K-\widetilde{K}\right)\right)+\tfrac{2\sigma^2}{n}\,\mathrm{tr}\left((\widetilde{K}+\lambda I)^{-1}\widetilde{K}\right)}_{\textit{penalty}}$$

Empirical upper bound

$$\hat{u}(\theta, X) = \left[\sum_{i=1}^{n}(y_i - \hat{f}_\theta^{\mathrm{FLK}}(x_i))^2 + \lambda\|\hat{f}_\theta^{\mathrm{FLK}}\|^2\right]\left(1+\tfrac{1}{\lambda}\,\mathrm{tr}\left(K-\widetilde{K}\right)\right)+\tfrac{1}{n}\,\mathrm{tr}\left((\widetilde{K}+\lambda I)^{-1}\widetilde{K}\right)$$

Nyström kernel, $\widetilde{K} = K_{nm}K_{mm}^{-1}K_{mn}$

Optimization

$$\hat{u}(\theta, X) = \left[\sum_{i=1}^{n}(y_i - \hat{f}_\theta^{\mathrm{FLK}}(x_i))^2 + \lambda\|\hat{f}_\theta^{\mathrm{FLK}}\|^2\right]\left(1 + \tfrac{1}{\lambda}\operatorname{tr}\left(K - \widetilde{K}\right)\right) + \tfrac{1}{n}\operatorname{tr}\left((\widetilde{K} + \lambda I)^{-1}\widetilde{K}\right)$$

UniGe | MaLGa

Optimization

$$\hat{u}(\theta, X) = \left[\sum_{i=1}^{n}(y_i - \hat{f}_\theta^{\mathrm{FLK}}(x_i))^2 + \lambda \|\hat{f}_\theta^{\mathrm{FLK}}\|^2\right]\left(1 + \frac{1}{\lambda}\operatorname{tr}\left(K - \widetilde{K}\right)\right) + \frac{1}{n}\operatorname{tr}\left((\widetilde{K} + \lambda I)^{-1}\widetilde{K}\right)$$

Trace estimation

$\widetilde{K} \in \mathbb{R}^{n \times n}$ is huge. Approximate! (Hutchinson, 1990):

$$\operatorname{tr}\widetilde{K} \approx \sum_{i=1}^{t \ll n} v_i^\top \widetilde{K} v_i = \sum_{i=1}^{t \ll n} v_i^\top K_{nm} K_{mm}^{-1} \underbrace{K_{mn} v_i}_{\text{reuse efficient kernel-vector products}}, \quad v_i \sim \mathcal{N}(0, 1)$$

UniGe | MaLGa

Optimization

$$\hat{u}(\theta, X) = \left[\sum_{i=1}^{n}(y_i - \hat{f}_\theta^{\mathrm{FLK}}(x_i))^2 + \lambda \| \hat{f}_\theta^{\mathrm{FLK}} \|^2\right]\left(1 + \tfrac{1}{\lambda}\operatorname{tr}\left(K - \widetilde{K}\right)\right) + \tfrac{1}{n}\operatorname{tr}\left((\widetilde{K} + \lambda I)^{-1}\widetilde{K}\right)$$

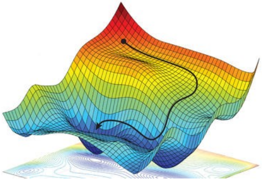
Trace estimation

$\widetilde{K} \in \mathbb{R}^{n \times n}$ is huge. Approximate! (Hutchinson, 1990):

$$\operatorname{tr}\widetilde{K} \approx \sum_{i=1}^{t \ll n} v_i^\top \widetilde{K} v_i = \sum_{i=1}^{t \ll n} v_i^\top K_{nm} K_{mm}^{-1} \underbrace{K_{mn} v_i}_{\text{reuse efficient kernel-vector products}}, \quad v_i \sim \mathcal{N}(0, 1)$$

Gradient descent

$\hat{u}(\theta, X)$ is differentiable wrt all $\theta := \lambda, \gamma, \{\widetilde{x}_i\}_{i=1}^m$



Optimization

$$\hat{u}(\theta, X) = \left[\sum_{i=1}^{n}(y_i - \hat{f}_\theta^{\mathrm{FLK}}(x_i))^2 + \lambda \|\hat{f}_\theta^{\mathrm{FLK}}\|^2\right]\left(1 + \tfrac{1}{\lambda}\operatorname{tr}\left(K - \widetilde{K}\right)\right) + \tfrac{1}{n}\operatorname{tr}\left((\widetilde{K} + \lambda I)^{-1}\widetilde{K}\right)$$

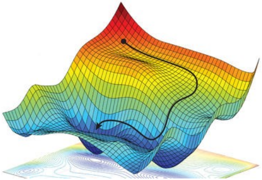
Trace estimation

$\widetilde{K} \in \mathbb{R}^{n \times n}$ is huge. Approximate! (Hutchinson, 1990):

$$\operatorname{tr}\widetilde{K} \approx \sum_{i=1}^{t \ll n} v_i^\top \widetilde{K} v_i = \sum_{i=1}^{t \ll n} v_i^\top K_{nm} K_{mm}^{-1} \underbrace{K_{mn} v_i}_{\text{reuse efficient kernel-vector products}}, \quad v_i \sim \mathcal{N}(0, 1)$$

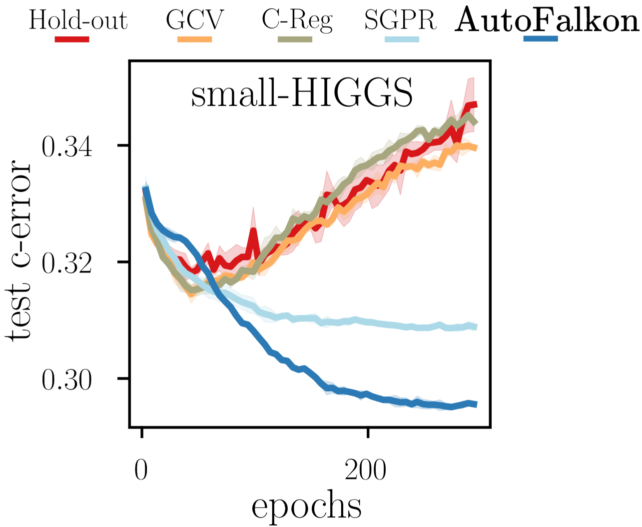
Gradient descent

$\hat{u}(\theta, X)$ is differentiable wrt all $\theta := \lambda, \gamma, \{\widetilde{x}_i\}_{i=1}^{m}$



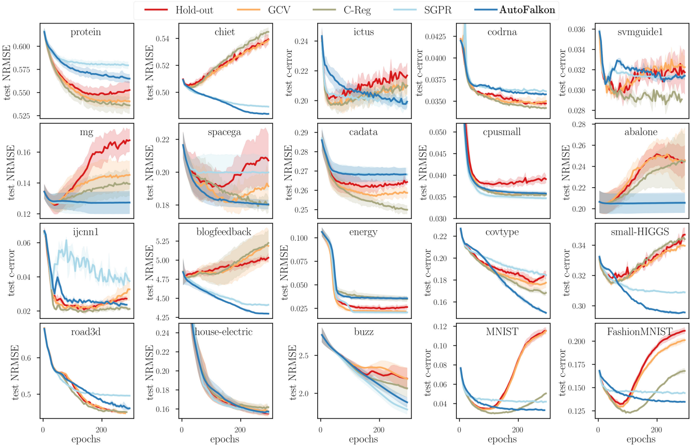
Optimize up to $|\theta| \approx 50\,000$ hyperparameters

A First Comparison



Small Scale Experiments

25 small, tabular+image datasets



- No single winner
- *AutoFalkon* ranks first

Large Scale Experiments

Large, tabular datasets

		AutoFalkon	GPyTorch	GPFlow	Falkon 2.0
Flights $n \approx 10^6$	error	0.794	0.803	0.790	0.758
	time(s)	355	1862	1720	245
	m	5000	1000	2000	10^5
Flights-Cls $n \approx 10^6$	error	32.2	33.0	32.6	31.5
	time(s)	310	1451	627	186
	m	5000	1000	2000	10^5
Higgs $n \approx 10^7$	error	0.191	0.199	0.196	0.180
	time(s)	1244	3171	1457	443
	m	5000	1000	2000	10^5

m big

vs.

$\{\widetilde{x}_i\}_{i=1}^m$ optimized

UniGe | MaLGa

Large Scale Experiments

Large, tabular datasets

		AutoFalkon	GPyTorch	GPFlow	Falkon 2.0
Flights $n \approx 10^6$	error	0.794	0.803	0.790	0.758
	time(s)	355	1862	1720	245
	m	5000	1000	2000	10^5
Flights-Cls $n \approx 10^6$	error	32.2	33.0	32.6	31.5
	time(s)	310	1451	627	186
	m	5000	1000	2000	10^5
Higgs $n \approx 10^7$	error	0.191	0.199	0.196	0.180
	time(s)	1244	3171	1457	443
	m	5000	1000	2000	10^5

m big

vs.

$\{\tilde{x}_i\}_{i=1}^m$ optimized

Outline

Background

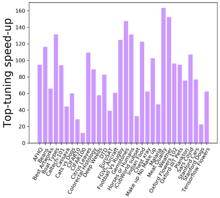
- Introduction to Kernel Methods
- Falkon 1.0

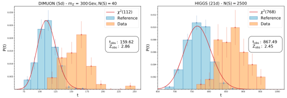
Contributions

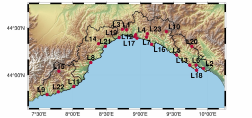
- Falkon 2.0 – Large Scale KRR
- Hyperparameter Tuning for Falkon 2.0
- **Falkon Applications**

Falkon: From Practice to Applications

- Fine Tuning or Top Tuning (Alfano et al. 2022)
- Object Segmentation on iCub (Ceola et al. 2022)
- Physics Discovery (Letizia et al. 2022)
- Wind Speed Forecasting (Lagomarsino Oneto, M., Pagliana, Verri, Mazzino, Rosasco, Seminara 2023)

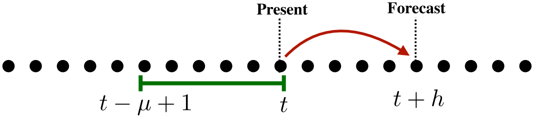




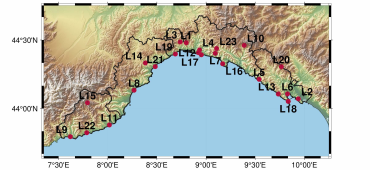


Wind Speed Forecasting

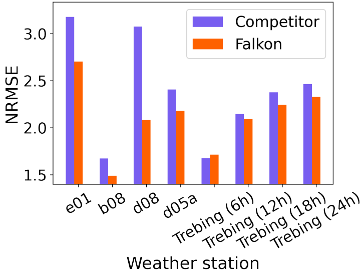
Time series forecasting



Anemometer locations



Compare with LSTMs, CNNs



Trained 6000 models, $n \approx 20\,000$

KRR: 20 h $\rightarrow$ Falkon: 1 h

(Araya et al., 2020, Trebing et al., 2020)

UniGe | MaLGa

Contributions

Falkon 2.0 – Large Scale KRR

Hyperparameter Tuning for Falkon 2.0

Applications: *Wind Forecasting*

Contributions

Falkon 2.0 – Large Scale KRR

Hyperparameter Tuning for Falkon 2.0

Applications: *Wind Forecasting*

Future Directions

- Falkon 3.0:
  - More parallelization
  - More parameters
- Structured kernels
- Dynamical systems & molecular dynamics

UniGe | MaLGa

Summary of Published Articles

Large Scale Kernels: Algorithms & Theory

- Falkon 2.0 M., Carratino, Rosasco, Rudi (2020)
- Hyperparameter Optimization for N-KRR M., Carratino, De Vito, Rosasco (2022)
- Exponential rates for multiclass learning Vigogna, M., De Vito, Rosasco (2022)

Large Scale Kernels: Applications

- Wind speed prediction Lagomarsino, M., Pagliana, Verri, Mazzino, Rosasco, Seminara (2023)
- Fast object segmentation on iCub robot Ceola, Maiettini, Pasquale, M., Rosasco, Natale (2022)

Miscellanea

- Efficient Neural Radiance Fields Fridovich-Keil*, M.*, Warburg, Recht, Kanazawa (2023)

Questions?