Optimizing Kernel Ridge Regression for remote sensing problems

Gonzalo Mateo-García, Valero Laparra
Luis Gómez-Chova
Image Processing Laboratory (IPL), UVEG, Spain

gonzalo.mateo-garcia@uv.es | http://isp.uv.es
Remote sensing $\rightarrow$ Biophysical parameter estimation
Nitrogen Dioxide

[Source https://www.esa.int/spaceinimages/Images/2017/12/Sentinel-5P_sees_nitrogen_dioxide_over_Europe]
Tree cover

(Hansen et al 2013)
Surface Temperature

ERA Interim ECMWF model
We need functions that map satellite measures to biophysical parameters that are **accurate** and **fast**.
We need functions that map satellite measures to biophysical parameters that are *accurate* and *fast*.

- Physical models
We need functions that map satellite measures to biophysical parameters that are **accurate** and **fast**

- Physical models
- Simultaneous measures $\rightarrow$ **Regression**
We need functions that map satellite measures to biophysical parameters that are **accurate** and **fast**

- Physical models
- Simultaneous measures $\rightarrow$ **Regression**
- Physical model slow $\rightarrow$ **Regression to build emulator**
We need functions that map satellite measures to biophysical parameters that are **accurate** and **fast**

- Physical models
- Simultaneous measures $\rightarrow$ **Regression**
- Physical model slow $\rightarrow$ **Regression to build emulator**
- Physical model forward direction $\rightarrow$ **Regression for model inversion**
Statistical Regression methods: \( \mathcal{D} = \{x_i, y_i\}_{i=1}^N \)

Among regression methods: **kernel methods** and kernel ridge regression:
Introduction

Statistical Regression methods: \( \mathcal{D} = \{x_i, y_i\}_{i=1}^N \)
Among regression methods: **kernel methods** and kernel ridge regression:

- KRR provides **non-linear** solutions for regression problems.
Introduction

Statistical Regression methods: $D = \{x_i, y_i\}_{i=1}^{N}$
Among regression methods: **kernel methods** and kernel ridge regression:

- KRR provides **non-linear** solutions for regression problems.
- **Kernel version** of regularized **linear regression** (ridge regression).
Introduction

Statistical Regression methods: \( D = \{x_i, y_i\}_{i=1}^{N} \)
Among regression methods: **kernel methods** and kernel ridge regression:

- **KRR** provides **non-linear** solutions for regression problems.
- **Kernel version** of regularized **linear regression** (ridge regression).
- **Computationally demanding** when the number of training instances increases.
Introduction

Statistical Regression methods: \( \mathcal{D} = \{x_i, y_i\}_{i=1}^{N} \)

Among regression methods: **kernel methods** and kernel ridge regression:

- KRR provides **non-linear** solutions for regression problems.
- **Kernel version** of regularized **linear regression** (ridge regression).
- **Computationally demanding** when the number of training instances increases.
- Due to the **growth of datasets**, kernel methods have been overshadowed by Neural Networks and tree methods (e.g. random forest, gradient boosting).
Introduction

Statistical Regression methods: \( \mathcal{D} = \{x_i, y_i\}_{i=1}^{N} \)
Among regression methods: kernel methods and kernel ridge regression:

- KRR provides non-linear solutions for regression problems.
- Kernel version of regularized linear regression (ridge regression).
- Computationally demanding when the number of training instances increases.
- Due to the growth of datasets, kernel methods have been overshadowed by Neural Networks and tree methods (e.g. random forest, gradient boosting).

Research question: Can we modify kernel methods to take advantage of large remote sensing datasets?
\begin{itemize}
\item \( \mathcal{D} = \{ \mathbf{x}_i, y_i \}_{i=1}^N \), \( \mathbf{x}_i \in \mathbb{R}^D, y_i \in \mathbb{R} \). We want \( f \in \mathcal{F} \) minimizes:
\end{itemize}

\[
J(f) = \sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i)^2 + \lambda \|f\|_{\mathcal{F}}^2
\]
KRR

\[ \mathcal{D} = \{x_i, y_i\}_{i=1}^N, \ x_i \in \mathbb{R}^D, \ y_i \in \mathbb{R}. \]  
We want \( f \in \mathcal{F} \) minimizes:

\[ J(f) = \sum_{i=1}^{N} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{F}}^2 \]

The Representer Theorem (Schölkopf et. al 2001)

\[ f^*(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x) = \begin{bmatrix} k(x, X) \end{bmatrix} \begin{bmatrix} \alpha \end{bmatrix}_{N \times 1}, \]

Kernel function

\[ k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \] Radial Basis Function (rbf)

\[ \alpha^* = \left( K_{NN} + \lambda I \right)^{-1} y. \]
Computational Cost

\[ \alpha^* = (K_{NN} + \lambda I)^{-1} y. \]

\[ K_{NN} = \begin{pmatrix}
  k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_N) \\
k(x_2, x_1) & k(x_2, x_2) & \ldots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
k(x_N, x_1) & \ldots & \ldots & k(x_N, x_N)
\end{pmatrix} \]

- Store \( N \times N \) matrix. \( \mathcal{O}(N^2) \)
- Invert \( K_{NN} \) matrix. \( \mathcal{O}(N^3) \)
- Predict new point \( \mathcal{O}(D \times N) \)
Computational Cost

- Store $N \times N$ matrix. $\mathcal{O}(N^2)$
- Invert $K_{NN}$ matrix. $\mathcal{O}(N^3)$
- Predict new point $\mathcal{O}(D \times N)$
**Solution:** select $M << N$ points from $\mathcal{D} = (X, y)$ $\mathcal{D}_u = (X_u, y_u)$ and discard rest.

**KRR**

$$f^*(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x) = k(x, X) \alpha$$

$$J(\alpha) = \|K_{NN} \alpha - y\|^2 + \lambda \alpha^\top K_{NN} \alpha$$

$$\Rightarrow \alpha^* = (K_{NN} + \lambda I)^{-1} y$$

Comp. Cost: $\mathcal{O}(N^3)$

**SoD**

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u$$

$$J(\alpha_u) = \|K_{MM} \alpha_u - y_u\|^2 + \lambda \alpha_u^\top K_{MM} \alpha_u$$

$$\alpha_{SoD} = (K_{MM} + \lambda I)^{-1} y_u$$

Comp. Cost: $\mathcal{O}(M^3)$
Solution: select $M << N$ points from $\mathcal{D} = (X, y)$ $\mathcal{D}_u = (X_u, y_u)$ and discard rest.

\begin{align*}
\text{SoD} & \\
\hat{f}(x) &= \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u \\
J(\alpha_u) &= \|K_{MM}\alpha_u - y_u\|^2 + \lambda\alpha_u^\top K_{MM}\alpha_u \\
\alpha_{SoD} &= (K_{MM} + \lambda I)^{-1} y_u \\
\text{Comp. Cost: } &\mathcal{O}(M^3)
\end{align*}
Solution: select $M << N$ points from $\mathcal{D} = (X, y)$ $\mathcal{D}_u = (X_u, y_u)$ and discard rest.

SoD

\[
\begin{aligned}
\alpha^*_{SoD} &= (K_{MM} + \lambda I)^{-1} y_u \\
\mathcal{O}(M^3)
\end{aligned}
\]
Subset of Data (SoD)

**Solution:** select $M << N$ points from $D = (X, y)$ $D_u = (X_u, y_u)$ and discard rest.

\[
\text{SoD} \quad f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u
\]

\[
J(\alpha_u) = \|K_{MM} \alpha_u - y_u\|^2 + \lambda \alpha_u^\top K_{MM} \alpha_u
\]

\[
\alpha_{SoD} = (K_{MM} + \lambda I)^{-1} y_u
\]

Comp. Cost: $O(M^3)$
**Subset of Data (SoD)**

**Solution:** select $M << N$ points from $\mathcal{D} = (X, y) \quad \mathcal{D}_u = (X_u, y_u)$ and discard rest.

Error KRR: 0.009

Error SoD: 1.02
Solution: select $M << N$ points from $\mathcal{D} = (X, y)$ $\mathcal{D}_u = (X_{u}, y_{u})$ and discard rest.

Error KRR: 0.009
Error SoD: 1.02

Problem: we ignore most of the data!
Nyström (Williams and Seeger 2000)

Solution: select $M << N$ BUT optimize w.r.t. all the data:

\[
\begin{align*}
\text{KRR} & & \\
\quad f^*(x) &= \sum_{i=1}^{N} \alpha_i k(x_i, x) = \underbrace{k(x, X)}_{1 \times N} \underbrace{\alpha}_{N \times 1} \\
J(\alpha) &= \|K_{NN}\alpha - y\|^2 + \lambda \alpha^\top K_{NN} \alpha \\
\implies \alpha^* &= (K_{NN} + \lambda I)^{-1} y \\
\text{Comp. Cost: } \mathcal{O}(N^3)
\end{align*}
\]

\[
\begin{align*}
\text{Nyström} & & \\
\quad f^*(x) &= \sum_{i=1}^{M} \alpha_i k(x_i, x) = \underbrace{k(x, X_u)}_{M \times 1} \underbrace{\alpha_u}_{1 \times M} \\
J(\alpha_u) &= \|K_{NM}\alpha_u - y\|^2 + \lambda \alpha_u^\top K_{MM} \alpha_u \\
\alpha_{\text{Nyström}} &= (K_{MN}K_{NM} + \lambda K_{MM})^{-1} K_{MN}y \\
\text{Comp. Cost: } \mathcal{O}(NM^2 + M^3)
\end{align*}
\]
Nyström (Williams and Seeger 2000)

Solution: select $M << N$ BUT optimize w.r.t. all the data:

**SoD**

\[
f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, \mathbf{X}_u) \mathbf{\alpha}_u
\]

\[
J(\mathbf{\alpha}_u) = \|K_{MM} \mathbf{\alpha}_u - y_u\|^2 + \lambda \mathbf{\alpha}_u^T K_{MM} \mathbf{\alpha}_u
\]

\[
\mathbf{\alpha}_{SoD} = (K_{MM} + \lambda I)^{-1} y_u
\]

Comp. Cost: $\mathcal{O}(M^3)$

**Nyström**

\[
f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, \mathbf{X}_u) \mathbf{\alpha}_u
\]

\[
J(\mathbf{\alpha}_u) = \|K_{NM} \mathbf{\alpha}_u - y\|^2 + \lambda \mathbf{\alpha}_u^T K_{MM} \mathbf{\alpha}_u
\]

\[
\mathbf{\alpha}_{Nyström} = (K_{MN} K_{NM} + \lambda K_{MM})^{-1} K_{MN} y
\]

Comp. Cost: $\mathcal{O}(NM^2 + M^3)$
Nyström (Williams and Seeger 2000)

**Solution:** select $M << N$ BUT optimize w.r.t. all the data:

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u$$

$$J(\alpha_u) = \|K_{NM} \alpha_u - y\|^2 + \lambda \alpha_u^T K_{MM} \alpha_u$$

$$\alpha_{Nyström} = \left( K_{MN} K_{NM} + \lambda K_{MM} \right)^{-1} K_{MN} y$$

Comp. Cost: $\mathcal{O}(NM^2 + M^3)$
**Nyström (Williams and Seeger 2000)**

**Solution:** select $M << N$ BUT optimize w.r.t. all the data:

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u$$

$$J(\alpha_u) = \|K_{NM} \alpha_u - y\|^2 + \lambda \alpha_u^T K_{MM} \alpha_u$$

$$\alpha_{Nyström} = (K_{MN} K_{NM} + \lambda K_{MM})^{-1} K_{MNY}$$

Comp. Cost: $O(NM^2 + M^3)$

Error KRR: 0.009
Error SoD: 1.02
Error Nyström: 0.73
Nyström (Williams and Seeger 2000)

**Solution:** select $M << N$ BUT optimize w.r.t. all the data:

- Error KRR: 0.009
- Error SoD: 1.02
- Error Nyström: 0.73

**Nyström**

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u$$

$$J(\alpha_u) = \|K_{NM} \alpha_u - y\|^2 + \lambda \alpha_u^T K_{MM} \alpha_u$$

$$\alpha_{Nyström} = (K_{MN} K_{NM} + \lambda K_{MM})^{-1} K_{MNY}$$

- Comp. Cost: $\mathcal{O}(NM^2 + M^3)$

**Problem:** very restrictive to sampled $X_u$ observations
Optimized KRR (OKRR)

**Solution:** optimize w.r.t. all the data both $\alpha_u$ and $X_u$.

**KRR**

$$f^*(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x) = k(x, X) \alpha$$

$$J(\alpha) = \|K_{NN}\alpha - y\|^2 + \lambda \alpha^\top K_{NN} \alpha$$

$$\Rightarrow \alpha^* = \left(\frac{1}{N \times N} K_{NN} + \lambda I\right)^{-1} y$$

Comp. Cost: $O(N^3)$

**OKRR**

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u$$

$$J(\alpha_u, X_u) = \|K_{NM}\alpha_u - y\|^2 + \lambda \alpha_u^\top K_{MM} \alpha_u$$

Optimize $J(\alpha_u, X_u)$ using stochastic gradient descent (SGD)

Comp. Cost each SGD step: $O(M^2)$
**Optimized KRR (OKRR)**

**Solution:** optimize w.r.t. all the data both $\alpha_u$ and $X_u$.

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = \underbrace{k(x, X_u)}_{1 \times M} \underbrace{\alpha_u}_{M \times 1}$$

$$J(\alpha_u, X_u) = \|K_{NM}\alpha_u - y\|^2 + \lambda \alpha_u^T K_{MM} \alpha_u$$

- $\alpha_u(0) = \alpha_{SoD}$
- $X_u(0) = X_u$
- $\alpha_u(t+1) = \alpha_u(t) + \nabla_{\alpha_u} J(\alpha_u(t), X_u(t))$
- $X_u(t+1) = X_u(t) + \nabla_{X_u} J(\alpha_u(t), X_u(t))$
Optimized KRR (OKRR)

Solution: optimize w.r.t. all the data both $\alpha_u$ and $X_u$.

Start with SoD solution

OKRR

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u$$

$$J(\alpha_u, X_u) = \|K_{NM}\alpha_u - y\|^2 + \lambda \alpha_u^T K_{MM} \alpha_u$$

$$\alpha_u(0) = \alpha_{SoD}$$

$$X_u(0) = X_u$$

$$\alpha_u(t + 1) = \alpha_u(t) + \nabla_{\alpha_u} J(\alpha_u(t), X_u(t))$$

$$X_u(t + 1) = X_u(t) + \nabla_{X_u} J(\alpha_u(t), X_u(t))$$
Optimized KRR (OKRR)

**Solution:** optimize w.r.t. all the data both $\alpha_u$ and $X_u$.

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u$$

$$J(\alpha_u, X_u) = \|K_{NM}\alpha_u - y\|^2 + \lambda \alpha_u^\top K_{MM} \alpha_u$$

$$\alpha_u(0) = \alpha_{SoD}$$

$$X_u(0) = X_u$$

$$\alpha_u(t + 1) = \alpha_u(t) + \nabla_{\alpha_u} J(\alpha_u(t), X_u(t))$$

$$X_u(t + 1) = X_u(t) + \nabla_{X_u} J(\alpha_u(t), X_u(t))$$
Optimized KRR (OKRR)

**Solution:** optimize w.r.t. all the data both $\alpha_u$ and $X_u$.

$$f^*(x) = \sum_{i=1}^{M} \alpha_i k(x_i, x) = k(x, X_u) \alpha_u$$

$$J(\alpha_u, X_u) = \| K_{NM} \alpha_u - y \|^2 + \lambda \alpha_u^\top K_{MM} \alpha_u$$

Error KRR: 0.009
Error SoD: 1.02
Error Nyström: 0.73
Error OKRR: 0.015
OKRR

- Fast training time. It does not depend on $N$ ($O(M^2)$ each SGD update).
- More flexible functions than Nyström.
- More parameters to fit ($X_u: M \times D$ matrix and $\alpha_u: M$ vector)
- Leverage automatic differentiation tools like tensorflow.
- Fast prediction time ($O(MD)$)
- Related with Sparse Gaussian Processes (SGPs) methods: (Quiñonero-Candela et al. 2005), (Snelson and Ghahramani 2005), (Titsias 2009) and (Hensman et. al 2013). Fourier random features (Rahimi and Recht 2007).
- Available at: https://github.com/gonzmg88/obfkrr
Input Data

- Infrared atmospheric sounding interferometer (IASI) instrument on-board the MetOp polar orbiting satellite.
- IASI measures in the infrared part of the electromagnetic spectrum at a horizontal resolution of 12 km (swath width 2,200 km).
- Spectral range between 645 and 2760 cm$^{-1}$ yielding 8461 spectral channels.

Global Earth coverage every 12 hours, 14 orbits. Each orbit 100,000 hyperpixels.
Output Data

**Surface temperature** which is obtained from the ERA Interim model from the European Center for Medium-Range Weather Forecast (ECMWF).
Experimental setup

Preprocessing:
- We took 12 orbits, we used 6 for training and 6 for testing.
- PCA to reduce the spectral dimensionality: $8461 \rightarrow 50,10$.

We compare 4 different methods:
1. Linear Regression
2. Subset of Data
3. Nystrom
4. OKRR

Hyperparameters of the kernel ($\gamma$) and regularization ($\lambda$) optimized by cross validation.
Experimental setup

Preprocessing:
- We took 12 orbits, we used 6 for training and 6 for testing.
- PCA to reduce the spectral dimensionality: $8461 \rightarrow 50,10$.

We compare 4 different methods:
1. Linear Regression
2. Subset of Data
3. Nystrom
4. OKRR

Hyperparameters of the kernel ($\gamma$) and regularization ($\lambda$) optimized by cross validation.
Quantitative results using 500 basis ($M = 500, D = 50$):

<table>
<thead>
<tr>
<th>Method</th>
<th>Root Mean Square Error (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression (LR)</td>
<td>6.22 K</td>
</tr>
<tr>
<td>Subset of Data</td>
<td>5.67 K</td>
</tr>
<tr>
<td>Nystrom</td>
<td>4.95 K</td>
</tr>
<tr>
<td>OKRR</td>
<td>4.95 K</td>
</tr>
</tbody>
</table>

Quantitative results using 50 basis ($M = 50, D = 10$):

<table>
<thead>
<tr>
<th>Method</th>
<th>Root Mean Square Error (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression (LR)</td>
<td>8.52 K</td>
</tr>
<tr>
<td>Subset of Data</td>
<td>9.08 K</td>
</tr>
<tr>
<td>Nystrom</td>
<td>5.78 K</td>
</tr>
<tr>
<td>OKRR</td>
<td>5.18 K</td>
</tr>
</tbody>
</table>
Results

Linear Regression

Subset of Data

OKRR
Results
Results

Subset of Data

degrees (K)
Results
Conclusion and future work

Conclusions:
- Presented an extension of KRR (OKRR) that can be trained with SGD without constraints in the amount of available training data.
- Higher number of trainable parameters $\rightarrow$ higher flexibility.
- Tested on surface temperature retrieval using IASI data.

Future work:
- Theoretical and empirical comparison with Sparse Gaussian Processes (SGPs).
- Comparisons in terms of training time and prediction time.
Conclusion and future work

Conclusions:
- Presented an extension of KRR (OKRR) that can be trained with SGD without constraints in the amount of available training data.
- Higher number of trainable parameters $\rightarrow$ higher flexibility.
- Tested on surface temperature retrieval using IASI data.

Future work:
- Theoretical and empirical comparison with Sparse Gaussian Processes ($SGPs$).
- Comparisons in terms of training time and prediction time.
Conclusions:
- Presented an extension of KRR (OKRR) that can be trained with SGD without constraints in the amount of available training data.
- Higher number of trainable parameters $\rightarrow$ higher flexibility.
- Tested on surface temperature retrieval using IASI data.

Future work:
- Theoretical and empirical comparison with Sparse Gaussian Processes ($SGPs$).
- Comparisons in terms of training time and prediction time.

Thank you!
Optimizing Kernel Ridge Regression for remote sensing problems

Gonzalo Mateo-García, Valero Laparra
Luis Gómez-Chova
Image Processing Laboratory (IPL), UVEG, Spain

Visit http://isp.uv.es/