

# Less is More: Nyström Computational Regularization

**Alessandro Rudi**, Raffaello Camoriano, Lorenzo Rosasco  
University of Genova - Istituto Italiano di Tecnologia  
Massachusetts Institute of Technology  
ale\_rudi@mit.edu

Dec 10th  
NIPS 2015



Laboratory for Computational  
and Statistical Learning



## A Starting Point

### Classically:

Statistics and optimization **distinct steps** in algorithm design

# A Starting Point

## Classically:

Statistics and optimization **distinct steps** in algorithm design

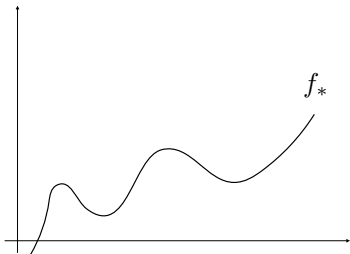
## Large Scale:

Consider **interplay** between statistics and optimization!

(Bottou, Bousquet '08)

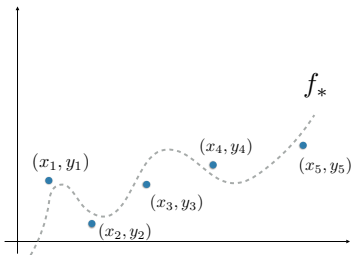
# Supervised Learning

**Problem:** Estimate  $f^*$



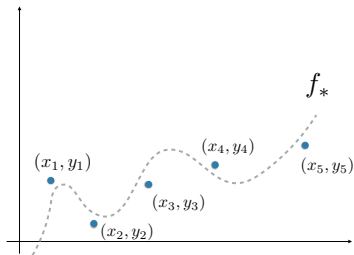
## Supervised Learning

**Problem:** Estimate  $f^*$  given  $S_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$



# Supervised Learning

**Problem:** Estimate  $f^*$  given  $S_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$



## The Setting

$$y_i = f^*(x_i) + \varepsilon_i \quad i \in \{1, \dots, n\}$$

- ▶  $\varepsilon_i \in \mathbb{R}, x_i \in \mathbb{R}^d$  **random** (with unknown distribution)
- ▶  $f^*$  **unknown**

# Outline

Learning with kernels

Data Dependent Subsampling

## Non-linear/non-parametric learning

$$\hat{f}(x) = \sum_{i=1}^M c_i q(x, w_i)$$



## Non-linear/non-parametric learning

$$\hat{f}(x) = \sum_{i=1}^M c_i q(x, w_i)$$

- ▶  $q$  non linear function

## Non-linear/non-parametric learning

$$\hat{f}(x) = \sum_{i=1}^M c_i q(x, w_i)$$

- ▶  $q$  non linear function
- ▶  $w_i \in \mathbb{R}^d$  centers

## Non-linear/non-parametric learning

$$\hat{f}(x) = \sum_{i=1}^M c_i q(x, w_i)$$

- ▶  $q$  non linear function
- ▶  $w_i \in \mathbb{R}^d$  centers
- ▶  $c_i \in \mathbb{R}$  coefficients

## Non-linear/non-parametric learning

$$\hat{f}(x) = \sum_{i=1}^M c_i q(x, w_i)$$

- ▶  $q$  non linear function
- ▶  $w_i \in \mathbb{R}^d$  centers
- ▶  $c_i \in \mathbb{R}$  coefficients
- ▶  $M = M_n$  could/should *grow* with  $n$

## Non-linear/non-parametric learning

$$\hat{f}(x) = \sum_{i=1}^M c_i q(x, w_i)$$

- ▶  $q$  non linear function
- ▶  $w_i \in \mathbb{R}^d$  centers
- ▶  $c_i \in \mathbb{R}$  coefficients
- ▶  $M = M_n$  could/should *grow* with  $n$

**Question:** How to choose  $w_i$ ,  $c_i$  and  $M$  given  $S_n$  ?

## Learning with Positive Definite Kernels

There is an *elegant* answer if:

- ▶  $q$  is **symmetric**
- ▶ *all* the matrices  $\hat{Q}_{ij} = q(x_i, x_j)$  are **positive semi-definite**<sup>1</sup>

---

<sup>1</sup>They have non-negative eigenvalues

# Learning with Positive Definite Kernels

There is an *elegant* answer if:

- ▶  $q$  is **symmetric**
- ▶ all the matrices  $\hat{Q}_{ij} = q(x_i, x_j)$  are **positive semi-definite**<sup>1</sup>

**Representer Theorem** (Kimeldorf, Wahba '70; Schölkopf et al. '01)

- ▶  $M = n$ ,
- ▶  $w_i = x_i$ ,
- ▶  $c_i$  by **convex** optimization!

---

<sup>1</sup>They have non-negative eigenvalues

## Kernel Ridge Regression (KRR)

a.k.a. Penalized Least Squares

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



# Kernel Ridge Regression (KRR)

a.k.a. Penalized Least Squares

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

where

$$\mathcal{H} = \left\{ f \mid f(x) = \sum_{i=1}^M c_i q(x, w_i), c_i \in \mathbb{R}, \underbrace{w_i \in \mathbb{R}^d}_{\text{any center!}}, \underbrace{M \in \mathbb{N}}_{\text{any length!}} \right\}$$

# Kernel Ridge Regression (KRR)

a.k.a. Penalized Least Squares

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

where

$$\mathcal{H} = \left\{ f \mid f(x) = \sum_{i=1}^M c_i q(x, w_i), c_i \in \mathbb{R}, \underbrace{w_i \in \mathbb{R}^d}_{\text{any center!}}, \underbrace{M \in \mathbb{N}}_{\text{any length!}} \right\}$$

Solution

$$\hat{f}_\lambda = \sum_{i=1}^n c_i q(x, x_i) \quad \text{with} \quad c = (\hat{Q} + \lambda n I)^{-1} \hat{y}$$

## KRR: Statistics

## KRR: Statistics

**Well understood** statistical properties:

### Classical Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}} \quad \mathbb{E} (\hat{f}_{\lambda_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

## KRR: Statistics

**Well understood** statistical properties:

### Classical Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}} \quad \mathbb{E}(\hat{f}_{\lambda_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

Remarks

## KRR: Statistics

**Well understood** statistical properties:

### Classical Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}} \quad \mathbb{E} (\hat{f}_{\lambda_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

### Remarks

1. **Optimal nonparametric bound**

## KRR: Statistics

**Well understood** statistical properties:

### Classical Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}} \quad \mathbb{E}(\widehat{f}_{\lambda_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

### Remarks

1. **Optimal nonparametric bound**
2. Results for **general** kernels (e.g. splines/Sobolev etc.)

$$\lambda_* = n^{-\frac{1}{2s+1}}, \quad \mathbb{E}(\widehat{f}_{\lambda_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

## KRR: Statistics

**Well understood** statistical properties:

### Classical Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}} \quad \mathbb{E}(\hat{f}_{\lambda_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

### Remarks

1. **Optimal nonparametric bound**
2. Results for **general** kernels (e.g. splines/Sobolev etc.)

$$\lambda_* = n^{-\frac{1}{2s+1}}, \quad \mathbb{E}(\hat{f}_{\lambda_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

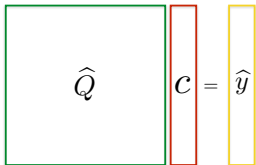
3. **Adaptive** tuning via cross validation



## KRR: Optimization

$$\hat{f}_\lambda = \sum_{i=1}^n c_i q(x, x_i) \quad \text{with} \quad c = (\hat{Q} + \lambda n I)^{-1} \hat{y}$$

### Linear System



A diagram illustrating the linear system  $\hat{Q}c = \hat{y}$ . It consists of three vertical elements: a large green square on the left containing the symbol  $\hat{Q}$ , a smaller red vertical rectangle in the middle containing the symbol  $c$ , and a yellow vertical rectangle on the right containing the symbol  $\hat{y}$ . The symbols are arranged as  $\hat{Q}c = \hat{y}$ .

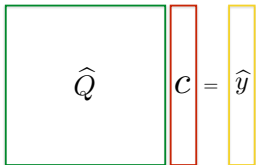
### Complexity

- ▶ **Space**  $O(n^2)$
- ▶ **Time**  $O(n^3)$

## KRR: Optimization

$$\hat{f}_\lambda = \sum_{i=1}^n c_i q(x, x_i) \quad \text{with} \quad c = (\hat{Q} + \lambda n I)^{-1} \hat{y}$$

### Linear System



A diagram illustrating a linear system. It consists of three vertical rectangular boxes. The first box on the left is a large green square containing the symbol  $\hat{Q}$ . To its right is a smaller red vertical rectangle containing the symbol  $c$ . To the right of the red box is an equals sign, followed by a yellow vertical rectangle containing the symbol  $\hat{y}$ .

### Complexity

- ▶ **Space**  $O(n^2)$
- ▶ **Time**  $O(n^3)$

### BIG DATA?

Running out of space before running out of time...

**Can this be fixed?**

# Outline

Learning with kernels

Data Dependent Subsampling

# Subsampling

1. pick  $w_i$  at random...

# Subsampling

1. pick  $w_i$  at random... from training set  
(Smola, Scholköpfung '00)

$$\tilde{w}_1, \dots, \tilde{w}_M \subset x_1, \dots, x_n \quad M \ll n$$



# Subsampling

1. pick  $w_i$  at random... from training set  
(Smola, Scholköpfung '00)

$$\tilde{w}_1, \dots, \tilde{w}_M \subset x_1, \dots, x_n \quad M \ll n$$



2. perform KRR on

$$\mathcal{H}_M = \left\{ f \mid f(x) = \sum_{i=1}^M c_i q(x, \tilde{w}_i), c_i \in \mathbb{R}, \tilde{w}_i \in \mathbb{R}^d, M \in \mathbb{N} \right\}.$$

# Subsampling

1. pick  $w_i$  at random... from training set  
(Smola, Scholköpfung '00)

$$\tilde{w}_1, \dots, \tilde{w}_M \subset x_1, \dots, x_n \quad M \ll n$$



2. perform KRR on

$$\mathcal{H}_M = \left\{ f \mid f(x) = \sum_{i=1}^M c_i q(x, \tilde{w}_i), c_i \in \mathbb{R}, \tilde{w}_i \in \mathbb{R}^d, M \in \mathbb{N} \right\}.$$

## Linear System

$$\hat{Q}_M c = \hat{y}$$

## Complexity

- ▶ **Space**  $O(n^2) \rightarrow O(nM)$
- ▶ **Time**  $O(n^3) \rightarrow O(nM^2)$

# Subsampling

1. pick  $w_i$  at random... from training set  
(Smola, Scholköpfung '00)

$$\tilde{w}_1, \dots, \tilde{w}_M \subset x_1, \dots, x_n \quad M \ll n$$



2. perform KRR on

$$\mathcal{H}_M = \{f \mid f(x) = \sum_{i=1}^M c_i q(x, \tilde{w}_i), c_i \in \mathbb{R}, \tilde{w}_i \in \mathbb{R}^d, M \in \mathbb{N}\}.$$

Linear System

$$\hat{Q}_M c = \hat{y}$$

Complexity

- ▶ **Space**  $O(n^2) \rightarrow O(nM)$
- ▶ **Time**  $O(n^3) \rightarrow O(nM^2)$

What about **statistics**? What's the **price** for efficient computations?



## Putting our Result in Context

- ▶ **\*Many\* different subsampling** schemes  
(Smola, Scholkopf '00; Williams, Seeger '01; ... 20+)

## Putting our Result in Context

- ▶ **\*Many\* different subsampling schemes**  
(Smola, Scholkopf '00; Williams, Seeger '01; ... 20+)
  
- ▶ **Theoretical guarantees** mainly on **matrix approximation**  
(Mahoney and Drineas '09; Cortes et al '10, Kumar et al.'12 ... 10+)

$$\|\hat{Q} - \hat{Q}_M\| \lesssim \frac{1}{\sqrt{M}}$$

## Putting our Result in Context

- ▶ **\*Many\* different subsampling** schemes  
(Smola, Scholkopf '00; Williams, Seeger '01; ... 20+)

- ▶ **Theoretical guarantees** mainly on **matrix approximation**  
(Mahoney and Drineas '09; Cortes et al '10, Kumar et al.'12 ... 10+)

$$\|\hat{Q} - \hat{Q}_M\| \lesssim \frac{1}{\sqrt{M}}$$

- ▶ Few prediction guarantees either **suboptimal** or in **restricted setting** (Cortes et al. '10; Jin et al. '11, Bach '13, Alaoui, Mahoney '14)

## Main Result

### Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}}, \quad M_* = \frac{1}{\lambda_*}, \quad \mathbb{E}(\hat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

# Main Result

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}}, \quad M_* = \frac{1}{\lambda_*}, \quad \mathbb{E}(\hat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

## Remarks

# Main Result

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}}, \quad M_* = \frac{1}{\lambda_*}, \quad \mathbb{E}(\hat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

## Remarks

1. Subsampling achieves **optimal** bound...

# Main Result

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}}, \quad M_* = \frac{1}{\lambda_*}, \quad \mathbb{E}(\hat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

## Remarks

1. Subsampling achieves **optimal** bound...
2. ...with  $M_* \sim \sqrt{n}$  !!

# Main Result

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}}, \quad M_* = \frac{1}{\lambda_*}, \quad \mathbb{E}(\hat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

## Remarks

1. Subsampling achieves **optimal** bound...
2. ...with  $M_* \sim \sqrt{n}$  !!
3. **More generally,**

$$\lambda_* = n^{-\frac{1}{2s+1}}, \quad M_* = \frac{1}{\lambda_*}, \quad \mathbb{E}_x(\hat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$



# Main Result

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$\lambda_* = \frac{1}{\sqrt{n}}, \quad M_* = \frac{1}{\lambda_*}, \quad \mathbb{E}(\hat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim \frac{1}{\sqrt{n}}$$

## Remarks

1. Subsampling achieves **optimal** bound...
2. ...with  $M_* \sim \sqrt{n}$  !!
3. **More generally,**

$$\lambda_* = n^{-\frac{1}{2s+1}}, \quad M_* = \frac{1}{\lambda_*}, \quad \mathbb{E}_x(\hat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

**Note:** An interesting insight is obtained rewriting the result...

## Computational Regularization (CoRe)

A simple idea: “*swap*” the role of  $\lambda$  and  $M$ ...

## Computational Regularization (CoRe)

A simple idea: “swap” the role of  $\lambda$  and  $M$ ...

### Theorem

If  $f^* \in \mathcal{H}$ , then

$$M_* = n^{\frac{1}{2s+1}}, \quad \lambda_* = \frac{1}{M_*}, \quad \mathbb{E}_x (\widehat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

## Computational Regularization (CoRe)

A simple idea: “swap” the role of  $\lambda$  and  $M$ ...

### Theorem

If  $f^* \in \mathcal{H}$ , then

$$M_* = n^{\frac{1}{2s+1}}, \quad \lambda_* = \frac{1}{M_*}, \quad \mathbb{E}_x (\widehat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

- ▶  $\lambda$  and  $M$  play the same role...  
...new interpretation: **subsampling regularizes!**

# Computational Regularization (CoRe)

A simple idea: “swap” the role of  $\lambda$  and  $M$ ...

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$M_* = n^{\frac{1}{2s+1}}, \quad \lambda_* = \frac{1}{M_*}, \quad \mathbb{E}_x (\widehat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

- ▶  $\lambda$  and  $M$  play the same role...  
... new interpretation: **subsampling regularizes!**
- ▶ New natural **incremental** algorithm...

## Algorithm

# Computational Regularization (CoRe)

A simple idea: “swap” the role of  $\lambda$  and  $M$ ...

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$M_* = n^{\frac{1}{2s+1}}, \quad \lambda_* = \frac{1}{M_*}, \quad \mathbb{E}_x (\widehat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

- ▶  $\lambda$  and  $M$  play the same role...  
... new interpretation: **subsampling regularizes!**
- ▶ New natural **incremental** algorithm...

## Algorithm

1. *Pick a center + compute solution*

# Computational Regularization (CoRe)

A simple idea: “swap” the role of  $\lambda$  and  $M$ ...

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$M_* = n^{\frac{1}{2s+1}}, \quad \lambda_* = \frac{1}{M_*}, \quad \mathbb{E}_x (\widehat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

- ▶  $\lambda$  and  $M$  play the same role...  
... new interpretation: **subsampling regularizes!**
- ▶ New natural **incremental** algorithm...

## Algorithm

1. Pick a center + compute solution
2. Pick another center + **rank one update**

# Computational Regularization (CoRe)

A simple idea: “swap” the role of  $\lambda$  and  $M$ ...

## Theorem

If  $f^* \in \mathcal{H}$ , then

$$M_* = n^{\frac{1}{2s+1}}, \quad \lambda_* = \frac{1}{M_*}, \quad \mathbb{E}_x (\widehat{f}_{\lambda_*, M_*}(x) - f^*(x))^2 \lesssim n^{-\frac{2s}{2s+1}}$$

- ▶  $\lambda$  and  $M$  play the same role...  
... new interpretation: **subsampling regularizes!**
- ▶ New natural **incremental** algorithm...

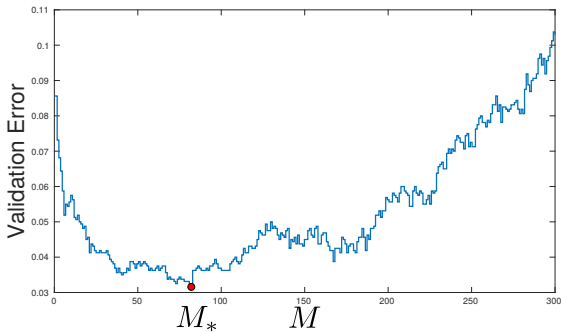
## Algorithm

1. Pick a center + compute solution
2. Pick another center + **rank one update**
3. Pick another center ...



# CoRe Illustrated

$n, \lambda$  are fixed



Computation controls stability!

Time/space requirement tailored to **generalization**

# Experiments

comparable/better w.r.t. the state of the art

<i>Dataset</i>	<i>n<sub>tr</sub></i>	<i>d</i>	<i>Incremental CoRe</i>	<i>Standard KRLS</i>	<i>Standard Nyström</i>	<i>Random Features</i>	<i>Fastfood RF</i>
Ins. Co.	5822	85	$0.23180 \pm 4 \times 10^{-5}$	<b>0.231</b>	0.232	0.266	0.264
CPU	6554	21	<b><math>2.8466 \pm 0.0497</math></b>	7.271	6.758	7.103	7.366
CT slices	42800	384	<b><math>7.1106 \pm 0.0772</math></b>	NA	60.683	49.491	43.858
Year Pred.	463715	90	<b><math>0.10470 \pm 5 \times 10^{-5}</math></b>	NA	0.113	0.123	0.115
Forest	522910	54	$0.9638 \pm 0.0186$	NA	<b>0.837</b>	0.840	0.840

- ▶ Random Features (Rahimi, Recht '07)
- ▶ Fastfood (Le et al. '13)

## Contributions

- ▶ **Optimal** learning with data dependent subsampling
- ▶ **Beyond uniform sampling** – come to the poster!

## Contributions

- ▶ **Optimal** learning with data dependent subsampling
- ▶ **Beyond uniform sampling** – come to the poster!

Some questions:

- ▶ Beyond ridge regression– **SGD and early stopping**
- ▶ Data independent sampling– **random features**
- ▶ Beyond randomization– **non convex optimization?**

## Contributions

- ▶ **Optimal** learning with data dependent subsampling
- ▶ **Beyond uniform sampling** – come to the poster!

### Some questions:

- ▶ Beyond ridge regression– **SGD and early stopping**
- ▶ Data independent sampling– **random features**
- ▶ Beyond randomization– **non convex optimization?**

### Some perspectives:

- ▶ **Computational regularization**: subsampling regularizes!
- ▶ **Algorithm design**: Control statistics with computations

**Thank you!**

Come to poster N.63 for the details!!

CODE: `lcs1.github.io/NystromCoRe`

Alessandro Rudi - `ale_rudi@mit.edu`

Laboratory for Computational and Statistical Learning - `lcs1.mit.edu`