Regularization Methods for Large Scale Machine Learning

Alessandro Rudi
University of Genova - Istituto Italiano di Tecnologia
Massachusetts Institute of Technology
lcsl.mit.edu

In collaboration with: Raffaello Camoriano, Lorenzo Rosasco

May 6th, 2017 - RegML, Oslo

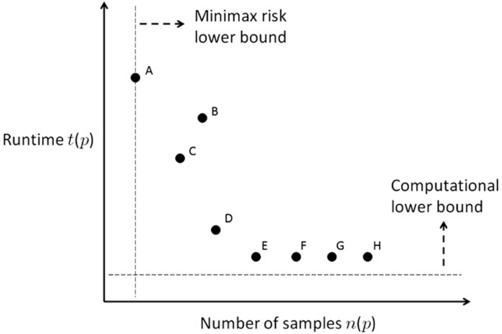
Laboratory for Computational and Statistical Learning

Machine learning Algorithms

Desiderata

- flexible non-linear / non-parametric models
- scalable computation
- statistical guarantees

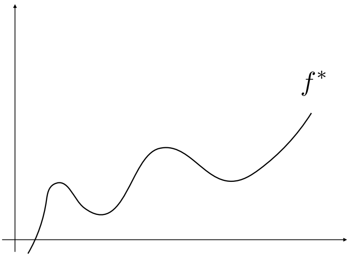
Statistics and computations



(Chandrasekaran, Jordan ’12)

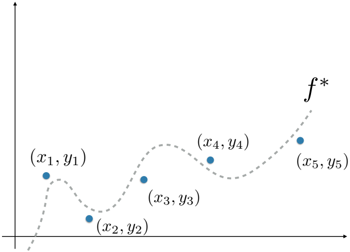
Supervised Learning

Problem: Estimate f_*



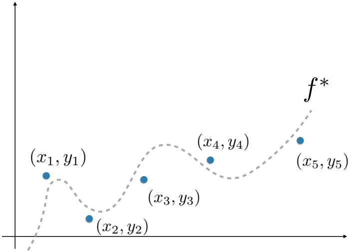
Supervised Learning

Problem: Estimate f_* given $S_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$



Supervised Learning

Problem: Estimate f_* given $S_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$



Setting

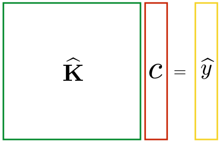
$$y_i = f^*(x_i) + \varepsilon_i \qquad i \in \{1, \ldots, n\}$$

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

Recall Kernel Ridge Regression

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

- $\Phi(x)^{\top}\Phi(x') = K(x, x')$
  kernel, e.g. Gaussian
- $\widehat{K}$ n by n data matrix
- $\widehat{y}$ outputs vector



Computational complexity

Time: $O(n^3)$ **Memory:** $O(n^2)$

Statistical Guarantees

Let $\mathcal{E}(f) = \mathbb{E}\ (y - f(x))^2$.

Theorem (Caponetto, DeVito '05)

Assume $\exists w$ *such that* $f_*(x) = w^\top \Phi(x)$*, subexp noise, and bounded kernel. Then w.h.p.*

$$\mathcal{E}(\widehat{f}^{\lambda}) - \mathcal{E}(f_*) \lesssim \frac{1}{\lambda n} + \lambda,$$

so that for $\lambda_n = 1/\sqrt{n}$*, w.h.p.*

$$\mathcal{E}(\widehat{f}^{\lambda_n}) - \mathcal{E}(f_*) \lesssim \frac{1}{\sqrt{n}}.$$

Remarks

- Bound is minimax **optimal** (Caponetto, DeVito '05)
- Adaptivity via cross validation or Lepskii method (Caponetto, Yao '07, De Vito Pereverzev R. '07)
- Refined results, e.g. for Sobolev classes rate is $n^{-\frac{2s}{2s+d}}$ (Caponetto, DeVito '05)

Computational complexity kills the method for large problems

Computational complexity

Time: $O(n^3)$ **Memory:** $O(n^2)$

BIG DATA

Where it is possible to run Kernel Ridge regression

- $n \approx 10\ 000$ Laptop (~ 1 Gigabyte memory),
- $n \approx 100\ 000$ Desktop (~ 100 Gigabyte memory),
- $n \approx 1000\ 000$ Cluster (~ 10 Terabyte memory),
- $n \approx 10\ 000\ 000$ Supercomputer (TOP10) (~ 1 Petabyte memory)

High energy physics experiments: $n \approx 10^7$ per second...

Outline

- **Data independent subsampling**
- Data dependent subsampling
- Adaptive subsampling

An idea

If $K(x, x') = \Phi_M(x)^\top \Phi_M(x')$ with $\Phi_M(x) \in \mathbb{R}^M$

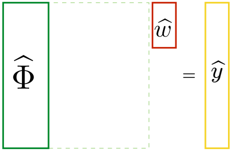
$$\widehat{f}^{\lambda,M}(x) = \Phi_M(x)^\top \widehat{w}^{\lambda,M} \qquad \widehat{w}^{\lambda,M} = (\widehat{\Phi}^\top \widehat{\Phi} + \lambda n I)^{-1} \widehat{\Phi}^\top \widehat{y}$$

An idea

If $K(x, x') = \Phi_M(x)^\top \Phi_M(x')$ with $\Phi_M(x) \in \mathbb{R}^M$

$$\widehat{f}^{\lambda,M}(x) = \Phi_M(x)^\top \widehat{w}^{\lambda,M} \qquad \widehat{w}^{\lambda,M} = (\widehat{\Phi}^\top \widehat{\Phi} + \lambda n I)^{-1} \widehat{\Phi}^\top \widehat{y}$$

- $\widehat{\Phi} = (\Phi_M(x_1), \dots \Phi_M(x_n))^\top \ \in \mathbb{R}^{n \times M}$
- $\widehat{w}^{\lambda,M}$ vector in $\mathbb{R}^M$
- $\widehat{y}$ outputs vector in $\mathbb{R}^n$



Computational complexity

Time: ~~$O(n^3)$~~ $\rightarrow O(nM^2)$ **Memory:** ~~$O(n^2)$~~ $\rightarrow O(nM)$

Kernel Approximation

Find $\Phi_M(x) \in \mathbb{R}^M$ such that

$$K(x, x') \approx \Phi_M(x)^\top \Phi_M(x')$$

Apply previous algorithm.

Random Fourier Features

Gaussian Kernel

$$e^{-\gamma \|x-x'\|^2} = \mathbb{E}_\omega \left[e^{i\omega^\top x} e^{-i\omega^\top x'} \right], \quad \omega \sim \mathcal{N}(0, \gamma)$$

Random Fourier Features

Gaussian Kernel

$$e^{-\gamma\|x-x'\|^2} = \mathbb{E}_\omega\ [e^{i\omega^\top x} e^{-i\omega^\top x'}], \quad \omega \sim \mathcal{N}(0,\gamma)$$

$$\approx \ \frac{1}{M}\sum_{j=1}^{M} e^{i\omega_j^\top x} e^{-i\omega_j^\top x}, \quad \omega_j \sim \mathcal{N}(0,\gamma),$$

Random Fourier Features

Gaussian Kernel

$$e^{-\gamma\|x-x'\|^2} = \mathbb{E}_\omega\left[e^{i\omega^\top x} e^{-i\omega^\top x'}\right], \quad \omega \sim \mathcal{N}(0,\gamma)$$

$$\approx \frac{1}{M}\sum_{j=1}^{M} e^{i\omega_j^\top x} e^{-i\omega_j^\top x}, \quad \omega_j \sim \mathcal{N}(0,\gamma),$$

$$\Phi_M(x) := \frac{1}{\sqrt{M}}(e^{i\omega_1^\top x}, \ldots, e^{i\omega_M^\top x}).$$

Random Features Expansion

Find q such that

$$K(x, x') = \mathbb{E}_\omega \, [q(x, \omega) q(x', \omega)],$$

Random Features Expansion

Find q such that

$$K(x, x') = \mathbb{E}_\omega \; [q(x, \omega) q(x', \omega)],$$

sample $w_1, \ldots w_M$ and consider

$$K(x, x') \;\approx\; \Phi_M(x)^\top \Phi_M(x') := \frac{1}{M} \sum_{j=1}^{M} q(x, \omega_j) q(x, \omega_j).$$

$$\Phi_M(x) := \frac{1}{\sqrt{M}} (q(x, \omega_1), \ldots, q(x, \omega_M)).$$

Examples of Random Features

- **translation invariant** kernels,
- **dot product** kernels,
- **group invariant** kernels,
- infinite **neural nets** kernels,
- homogeneous **additive** kernels,
- **sum, products, composition** of kernels,
- ...

Computations

If $M \ll n$

- TIME:

$$O(nM^2) \ll O(n^3)$$

- SPACE:

$$O(nM) \ll O(n^2)$$

Any loss in ACCURACY?

Previous Results

- ***Many*** different random features for different kernels
  (Rahimi, Recht '07, Vedaldi, Zisserman, ... 10+)

- Theoretical guarantees: mainly **kernel approximation**
  (Rahimi, Recht '07, ..., Sriperumbudur and Szabo '15)

$$|K(x,x') - \Phi_M(x)^\top \Phi_M(x')| \lesssim \frac{1}{\sqrt{M}},$$

- Theoretical guarantees: generalization bounds (Rahimi, Recht '09, Bach, '15)

$$\boldsymbol{M = n} \quad \Rightarrow \quad \mathcal{E}(\widehat{f}^{\lambda,M}) - \mathcal{E}(f^*) \leq \frac{1}{\sqrt{n}}$$

Statistical Guarantees

Let $\mathcal{E}(f) = \mathbb{E}(y - f(x))^2$.

Theorem (R., Camoriano, Rosasco '16)

Assume $\exists w$ *such that* $f_*(x) = w^\top \Phi(x)$, *subexp noise, and bounded kernel.*
Then w.h.p.

$$\mathcal{E}(\widehat{f}^{\lambda,M}) - \mathcal{E}(f_*) \lesssim \frac{1}{\lambda n} + \frac{1}{M} + \lambda,$$

so that for

$$\lambda_n = \frac{1}{\sqrt{n}}, \quad \boldsymbol{M_n = \frac{1}{\lambda_n}}$$

the following hold w.h.p.

$$\mathcal{E}(\widehat{f}^{\lambda_n,M_n}) - \mathcal{E}(f_*) \lesssim \frac{1}{\sqrt{n}}.$$

Remarks

- Bound is minimax **optimal**, same as Tikhonov
- Adaptivity via cross validation or Lepskii method
- Refined results, e.g. for Sobolev classes rate is $n^{-\frac{2s}{2s+d}}$ (R., Camoriano, Rosasco '16)

Remarks

- Bound is minimax **optimal**, same as Tikhonov
- Adaptivity via cross validation or Lepskii method
- Refined results, e.g. for Sobolev classes rate is $n^{-\frac{2s}{2s+d}}$ (R., Camoriano, Rosasco '16)

$$M = \sqrt{n} \text{ guarantees NO loss in accuracy}$$

Computational complexity

Time: $\cancel{O(n^3)} \to O(n^2)$ **Memory:** $\cancel{O(n^2)} \to O(n\sqrt{n})$

M controls space, time and Statistics

Corollary: Same result if we set

$$M_n = \sqrt{n}, \quad \lambda_n = \frac{1}{M_n}.$$

- M can be seen as a *regularization parameter*

M controls space, time and Statistics

Corollary: Same result if we set

$$M_n = \sqrt{n}, \quad \lambda_n = \frac{1}{M_n}.$$

- M can be seen as a *regularization parameter*

New **incremental** algorithm

M controls space, time and Statistics

Corollary: Same result if we set

$$M_n = \sqrt{n}, \quad \lambda_n = \frac{1}{M_n}.$$

- M can be seen as a *regularization parameter*

New **incremental** algorithm

- 1. *Pick a random feature*
  $+$ *compute solution*

M controls space, time and Statistics

Corollary: Same result if we set

$$M_n = \sqrt{n}, \quad \lambda_n = \frac{1}{M_n}.$$

- M can be seen as a *regularization parameter*

New **incremental** algorithm

- 1. *Pick a random feature*
     \+ *compute solution*
  2. *Pick another random features*
     \+ **rank one update**

M controls space, time and Statistics

Corollary: Same result if we set

$$M_n = \sqrt{n}, \quad \lambda_n = \frac{1}{M_n}.$$

- M can be seen as a *regularization parameter*

New **incremental** algorithm

- 1. *Pick a random feature*
     $+$ *compute solution*
  2. *Pick another random features*
     $+$ **rank one update**
  3. *Pick another random feature . . .*

M controls space, time and Statistics

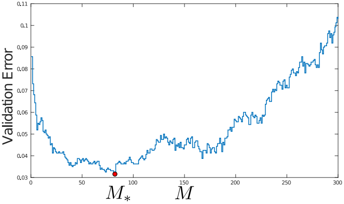
Corollary: Same result if we set

$$M_n = \sqrt{n}, \quad \lambda_n = \frac{1}{M_n}.$$

- M can be seen as a *regularization parameter*

New **incremental** algorithm

- 1. *Pick a random feature*
     $+$ *compute solution*
  2. *Pick another random features*
     $+$ **rank one update**
  3. *Pick another random feature ...*



M controls at the same time: Space, Time, Statistics

Outline

- Data independent subsampling
- **Data dependent subsampling**
- Adaptive subsampling

Data dependent subsampling with Nyström

$$\{\tilde{x}_1, \dots \tilde{x}_M\} \subseteq \{x_1, \dots, x_n\}$$

Data dependent subsampling with Nyström

$$\{\tilde{x}_1, \dots \tilde{x}_M\} \subseteq \{x_1, \dots, x_n\}$$

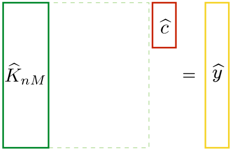
$$\widehat{f}^{\lambda,M}(x) = \sum_{j=1}^{M} K(x, \tilde{x}_j)\widehat{c}_i^{\lambda,M}, \qquad \widehat{c}^{\lambda,M} = (\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM}^\top)^{-1} \widehat{K}_{nM}^\top \widehat{y}$$

Data dependent subsampling with Nyström

$$\{\tilde{x}_1, \dots \tilde{x}_M\} \subseteq \{x_1, \dots, x_n\}$$

$$\widehat{f}^{\lambda,M}(x) = \sum_{j=1}^{M} K(x, \tilde{x}_j)\widehat{c}_i^{\lambda,M}, \qquad \widehat{c}^{\lambda,M} = (\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM}^\top)^{-1} \widehat{K}_{nM}^\top \widehat{y}$$

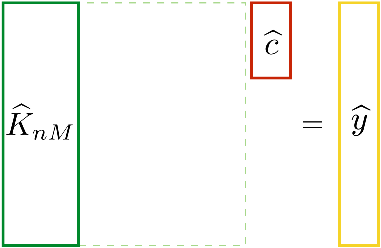
- $(\widehat{K}_{nM})_{ij} = K(x_i, \tilde{x}_j)$ n by M matrix
- $(\widehat{K}_{MM})_{ij} = K(\tilde{x}_i, \tilde{x}_j)$ M by M matrix
- $\widehat{y}$ outputs vector



Computational complexity

Time:$O(nM^2)$ **Memory:** $O(nM)$

Remarks



- Plenty of methods for subsampling . . .
- Connections to Nyström for integral operators
- Previous results: kernel approximation $\|\widehat{K} - \widehat{K}_{nM}^{\top} \widehat{K}_{MM}^{\dagger} \widehat{K}_{nM}\|$.

Any loss in ACCURACY?

Statistical Guarantees

Let $\mathcal{E}(f) = \mathbb{E}(y - f(x))^2$.

Theorem (R., Camoriano, Rosasco '15)

Assume $\exists w$ *such that* $f_*(x) = w^\top \Phi(x)$, *subexp noise, and bounded kernel.*
Then w.h.p.

$$\mathcal{E}(\widehat{f}^{\lambda,M}) - \mathcal{E}(f_*) \lesssim \frac{1}{\lambda n} + \lambda + \frac{1}{M},$$

so that for

$$\lambda_n = \frac{1}{\sqrt{n}}, \quad \boldsymbol{M_n = \frac{1}{\lambda_n}}$$

the following hold w.h.p.

$$\mathcal{E}(\widehat{f}^{\lambda_n,M_n}) - \mathcal{E}(f_*) \lesssim \frac{1}{\sqrt{n}}.$$

Remarks

- Bound is minimax **optimal**, same as Tikhonov
- Adaptivity via cross validation or Lepskii method
- Refined results, e.g. for Sobolev classes rate is $n^{-\frac{2s}{2s+d}}$ (R., Camoriano, Rosasco '15)

$M = \sqrt{n}$ guarantees NO loss in accuracy

Remarks

- Bound is minimax **optimal**, same as Tikhonov
- Adaptivity via cross validation or Lepskii method
- Refined results, e.g. for Sobolev classes rate is $n^{-\frac{2s}{2s+d}}$ (R., Camoriano, Rosasco '15)

$M = \sqrt{n}$ guarantees NO loss in accuracy

Computational complexity

Time: ~~$O(n^3)$~~ $\to O(n^2)$ **Memory:** ~~$O(n^2)$~~ $\to O(n\sqrt{n})$

Outline

- Data independent subsampling
- Data dependent subsampling
- **Adaptive subsampling**

Refined Results

Can we do better than uniform sampling?

Non-uniform subsampling Select the point $\tilde{x} = x_i$ with probability p_i.

Refined Results

Can we do better than uniform sampling?

Non-uniform subsampling Select the point $\tilde{x} = x_i$ with probability p_i.

- Leverage scores

$$p_i := \widehat{K}_i^\top (\widehat{K} + \lambda n I)^{-1} \widehat{K}_i,$$

with $\widehat{K}_i$ the i-th column of the data matrix $\widehat{K}$.

Statistical Guarantees

Let $\mathcal{E}(f) = \mathbb{E}(y - f(x))^2$.

Theorem (R., Camoriano, Rosasco '15)

Assume $\exists w$ *such that* $f_*(x) = w^\top \Phi(x)$, *subexp noise, bounded kernel, and* Φ *induces a Sobolev kernel with smoothness* s.
When

$$\lambda_n = n^{\frac{2s}{2s+d}}, \quad \boldsymbol{M_n = n^{\frac{d}{2s+d}}}$$

the following hold w.h.p.

$$\mathcal{E}(\widehat{f}^{\lambda_n, M_n}) - \mathcal{E}(f_*) \lesssim n^{-\frac{2s}{2s+d}}.$$

Remarks

- Bound is minimax, same as Tikhonov
- Adaptivity via cross validation or Lepskii method

- $M = n^{\frac{d}{2s+d}} \ll \sqrt{n}$ guarantees NO loss in accuracy
- $M = O(1)$ if s is *large*

Remarks

- Bound is minimax, same as Tikhonov
- Adaptivity via cross validation or Lepskii method

- $M = n^{\frac{d}{2s+d}} \ll \sqrt{n}$ guarantees NO loss in accuracy
- $M = O(1)$ if s is *large*

Computational complexity

Time: $\cancel{O(n^3)} \rightarrow O(nn^{\frac{2d}{2s+d}})$ **Memory:** $\cancel{O(n^2)} \rightarrow O(nn^{\frac{d}{2s+d}})$

Contributions & Open Questions

Contributions

- $M = \sqrt{n}$ gives optimal bounds
- $M \ll \sqrt{n}$ for adaptive sampling.
- Fast rates under smoothness conditions.

Open Questions

- Computational lower bounds?
- Efficient adaptive sampling

Back to big data...

- $n \approx 10\,000$ Laptop
- $n \approx 100\,000$ ~~Desktop~~ Laptop
- $n \approx 1000\,000$ ~~Cluster~~ Laptop
- $n \approx 10\,000\,000$ ~~TOP 10 Supercomputer~~ Desktop
- $n \approx 100\,000\,000$ ~~???~~ Desktop