Algorithmic Regularization for Fast and Optimal Large-Scale Machine Learning

Luigi Carratino
University of Genova

joint work with Alessandro Rudi (INRIA), Lorenzo Rosasco (UniGe, MIT, IIT)

Jul, 9th 2019 – SWSL 2019

Learning problem

Let $(x, y) \sim \rho, \quad x \in X \subseteq \mathbb{R}^d, \quad y \in Y \subseteq \mathbb{R}$.

Learn

$$f_{\mathcal{H}} = \operatorname*{argmin}_{f \in \mathcal{H}} \mathcal{E}(f), \qquad \mathcal{E}(f) = \int d\rho(x, y)(y - f(x))^2$$

with ρ **unknown** but given $(x_i, y_i)_{i=1}^n$ i.i.d. samples.

Remarks:

- $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ RKHS with bounded kernel K (e.g. $K(x, x') = e^{-\gamma \|x - x'\|^2}$)
- $\mathcal{H} = \overline{\operatorname{span}\{K(x, \cdot) | x \in X\}}$
- Let $\phi : \mathbb{R}^d \to \mathcal{H}$, then $K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$

Statistics

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

Theorem[Smale, Zhou '05, Caponnetto, De Vito '05]

For $\|\phi(x)\|, |y| \leq 1$,

$$\mathbb{E}\, \underbrace{\mathcal{E}(\widehat{f}_\lambda) - \mathcal{E}(f_{\mathcal{H}})}_{\textit{excess risk}} \lesssim \frac{1}{\lambda n} + \lambda.$$

By selecting $\lambda_n = \frac{1}{\sqrt{n}}$

$$\mathbb{E}\mathcal{E}(\widehat{f}_{\lambda_n}) - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{1}{\sqrt{n}}$$

- Minmax bound.
- Faster rate under refined assumptions

Optimization

$$\widehat{f}_{t+1} = \widehat{f}_t - \gamma_t \nabla \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \widehat{f}_t(x_i))^2 + \lambda \|f_t\|^2 \right)$$

Theorem

If $\gamma_t \leq 1$, then

$$\|\widehat{f}_t - \widehat{f}_\lambda\| \lesssim e^{-t\lambda}$$

Computational tricks = (implicit) regularization?

- **iterations**
- acceleration
- **stochastic gradients**
- **step-size**
- **mini-batch**
- averaging
- **sketching**
- subsampling
- preconditioning
- ...

Random features

Let $f(x)$ be

$$f(x) = \langle w, \phi_M(x) \rangle$$

where $\phi_M : \mathbb{R}^d \to \mathbb{R}^M$

$$\phi_M(x) := \left(\underbrace{\sigma(\langle x, s_1 \rangle)}_{\text{random feature}}, \dots, \sigma(\langle x, s_M \rangle) \right)$$

- $s_1, \dots, s_M \in \mathbb{R}^d$ i.i.d random vectors
- $\sigma : \mathbb{R} \to \mathbb{R}$ nonlinear function (e.g. $\sigma(a) = cos(a)$, $\sigma(a) = |a|_+$, $\dots$)

[Rahimi, Recht '06'08'08]

Link with kernels

Recall

$$f(x) = \langle w, \phi_M(x) \rangle = \sum_{j=1}^{M} w^j \sigma(\langle s_j, x \rangle)$$

with $s_1, \ldots, s_M \sim \pi$, then

$$\lim_{M \to \infty} \frac{1}{M} \sum_{j=1}^{M} w^j \sigma(\langle s_j, x \rangle) \in \mathcal{H}$$

and

$$K(x, x') = \int \sigma(\langle s, x \rangle) \sigma(\langle s, x' \rangle) d\pi(s)$$

[Neal '95; Rahimi, Recht '07; Cho, Saul '09]

Multi-pass SGD-RF with mini-batching

For $t = 1, \ldots, T$

$$\widehat{w}_{t+1} = \widehat{w}_t - \gamma_t \frac{1}{b} \sum_{i=b(t-1)+1}^{bt} \nabla \left(\left(y_{j_i} - \langle \widehat{w}_t, \phi_M(x_{j_i}) \rangle \right)^2 \right)$$

with $J = j_1, \ldots, j_{bT}$ sampling strategy.

Free parameters:

- Step-size γ_t
- Mini-batch size b
- Number of random features M
- Number of iterations T

Computational complexity:

- Time: $O(MbT)$
- Space: $O(M)$

Related works

- One pass SGD: from Robbins-Munro '50's... Dieuleveut, Bach '15...
- Multipass SGD: Hardt Recht Singer '16, Rosasco et al. '16
- SGD with averaging: Dieuleveut, Bach '15, Neu, Rosasco '18, Mücke, Neu, Rosasco 19'
- Sketching for Tikhonov regularization: Rudi, Rosasco '17.
- Multipass SGD + Mini-Batching + Sketching: This work!

SGD with Random Features: Statistics

Theorem (C., Rudi, Rosasco '18)

For $\|x\|, |y| \leq 1$ a.s. and $t > 1$

$$\mathbb{E}_J \mathcal{E}(\widehat{w}_{t+1}) - \inf_{w \in \mathcal{H}} \mathcal{E}(w) \lesssim \frac{\gamma}{b} + \left(\frac{\gamma t}{M} + 1\right) \frac{\gamma t}{n} + \frac{1}{M} + \frac{1}{\gamma t}.$$

SGD with Random Features: Statistics

Theorem (C., Rudi, Rosasco '18)

If

1. $b = 1$, $\gamma_t \simeq \frac{1}{\sqrt{n}}$, and $T = n$ iterations (1 pass over the data);
2. $b = \sqrt{n}$, $\gamma_t \simeq 1$, and $T = \sqrt{n}$ iterations (1 pass over the data);
3. $b = n$, $\gamma_t \simeq 1$, and $T = \sqrt{n}$ iterations ($\sqrt{n}$ passes over the data);

and

$$M = \sqrt{n}$$

then with high probability

$$\mathbb{E}_J \mathcal{E}(\widehat{w}_T) - \inf_{w \in \mathcal{H}} \mathcal{E}(w) \lesssim \frac{1}{\sqrt{n}}$$

- Minmax bound.
- Faster rate under refined assumptions

Computational requirements

For $t = 1, \dots, T$

$$\widehat{w}_{t+1} = \widehat{w}_t - \gamma_t \frac{1}{b} \sum_{i=b(t-1)+1}^{bt} \nabla \left(\left(y_{j_i} - \langle \widehat{w}_t, \phi_M(x_{j_i}) \rangle \right)^2 \right)$$

Complexity:

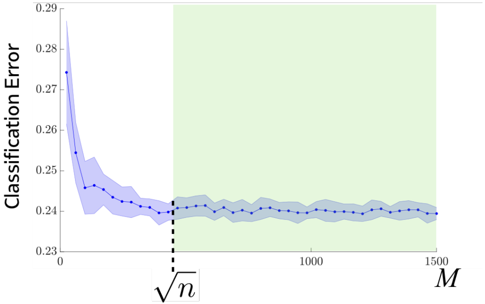
- Time: $O(MbT)$
- Space: $O(M)$

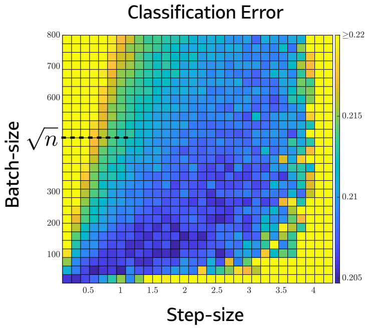
Complexity for $O(1/\sqrt{n})$ rate:

- Time: $O(n\sqrt{n})$
- Space: $O(\sqrt{n})$

Empirical results

SUSY dataset, $n = 6 \times 10^6$





- Same accuracy for $M \geq \sqrt{n}$
- $b = \sqrt{n}$ is the "magic" MB-size

Summing up

- number of passes, step-size mini-batch size and sketching dimension....
  all control the test error!
- They introduces an implicit bias hence regularize the solution

Looking ahead: apply/extend these ideas

- Beyond least squares
- Parallelization
- Non convex problems

From random features to subsampling

Similar results can be obtained considering

$$\overline{x}_1, \dots, \overline{x}_M \subset x_1, \dots, x_n$$

and

$$f(x) = \sum_{j=1}^{M} K(\overline{x}, x) c_j$$

- Nyström method

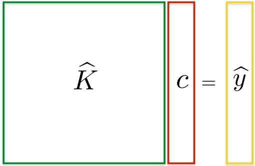
Back to Kernel Ridge Regression

Let K p.d. kernel and $\mathcal{H}$ corresponding RKHS

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

$$\widehat{f}_\lambda(x) = \sum_{i=1}^{n} K(x, x_i) c_i$$

$$(\widehat{K} + \lambda n I)c = \widehat{y}$$



Complexity: **Space** $O(n^2)$ **Kernel eval.** $O(n^2)$ **Time** $O(n^3)$

- Optimal statistical accuracy [Caponnetto, De Vito '05]

Random projections

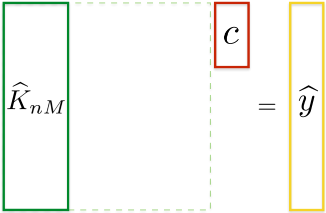
Consider $\mathcal{H}_M = \operatorname{span}\{K(\tilde{x}_1,\cdot),\ldots,K(\tilde{x}_M,\cdot)\} \subseteq \mathcal{H}$

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

- ... that is, pick M columns at random

$$\widehat{f}_{\lambda,M}(x) = \sum_{i=1}^{M} K(x,\tilde{x}_i)c_i$$

$$(\widehat{K}_{nM}^{\top}\widehat{K}_{nM} + \lambda n \widehat{K}_{MM})c = \widehat{K}_{nM}^{\top}\widehat{y}$$



Complexity: **Space** $O(M^2)$ **Kernel eval.** $O(nM)$ **Time** $O(nM^2)$

- **Nyström methods** (Smola, Scholköpf '00)
- Gaussian processes: inducing inputs (Quiñonero-Candela et al '05)
- Galerkin methods and Randomized linear algebra (Halko et al. '11)

Nyström KRR: Statistics

Theorem[Rudi, Camoriano, Rosasco '15] Under basic assumptions and

$$\mathbb{E}\mathcal{E}(\widehat{f}_{\lambda,M}) - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{1}{\lambda n} + \lambda + \frac{1}{M}.$$

By selecting $\lambda_n = \frac{1}{\sqrt{n}}, M_n = \sqrt{n}$

$$\mathbb{E}\mathcal{E}(\widehat{f}_{\lambda_n,M_n}) - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{1}{\sqrt{n}}$$

- Same minmax bound of KRR [Caponnetto, De Vito '05].

Computations required for $O(1/\sqrt{n})$ rate

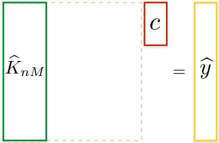
Space:	$O(n)$
Kernel eval.:	$O(n\sqrt{n})$
Time:	$O(n^2)$
Test:	$O(\sqrt{n})$

Possible improvements:

- adaptive sampling
- **optimization**

Optimization to rescue

$$\underbrace{\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM}}_{H} c = \underbrace{\widehat{K}_{nM}^\top \widehat{y}}_{b}.$$



Idea: First order methods

$$c_t = c_{t-1} - \frac{\tau}{n}\left[\widehat{K}_{nM}^\top(\widehat{K}_{nM}c_{t-1} - y_n) \; + \; \lambda n \widehat{K}_{MM} c_{t-1}\right]$$

Pros: requires $O(nMt)$

Cons: $t \propto \kappa(H)$ arbitrarily large- $\kappa(H) = \sigma_{\max}(H)/\sigma_{\min}(H)$ condition number.

Preconditioning

Idea: solve an equivalent linear system with better condition number

Preconditioning

$$Hc = b \quad \mapsto \quad P^\top H P \beta = P^\top b, \quad c = P\beta.$$

Ideally $PP^\top = H^{-1}$, so that

$$t = O(\kappa(H)) \quad \mapsto \quad t = O(1)!$$

(Fasshauer et al '12, Avron et al '16, Cutajat '16, Ma, Belkin '17)

Preconditioning Nystom-KRR

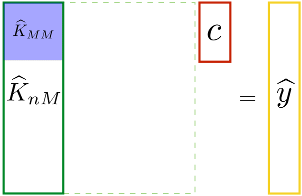
Consider $$H := \widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM}$$

Proposed Preconditioning

$$PP^\top = \left(\frac{n}{M} \widehat{K}_{MM}^2 + \lambda n \widehat{K}_{MM} \right)^{-1}$$

Compare to naive preconditioning

$$PP^\top = \left(\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM} \right)^{-1}.$$



Baby FALKON

Proposed Preconditioning

$$PP^\top = \left(\frac{n}{M}\widehat{K}_{MM}^2 + \lambda n \widehat{K}_{MM}\right)^{-1},$$

Gradient descent

$$\widehat{f}_{\lambda,M,t}(x) = \sum_{i=1}^{M} K(x, \widetilde{x}_i)c_{t,i}, \qquad c_t = P\beta_t$$

$$\beta_t = \beta_{t-1} - \frac{\tau}{n}P^\top \left[\widehat{K}_{nM}^\top(\widehat{K}_{nM}P\beta_{t-1} - y_n) \ + \ \lambda n \widehat{K}_{MM}P\beta_{t-1}\right]$$

FALKON

- Gradient descent $\mapsto$ conjugate gradient
- Computing P

$$P = \frac{1}{\sqrt{n}} T^{-1} A^{-1}, \quad T = \mathrm{chol}(K_{MM}), \quad A = \mathrm{chol}\left(\frac{1}{M}\, TT^\top + \lambda I\right),$$

where $\mathrm{chol}(\cdot)$ is the Cholesky decomposition.

Falkon statistics

Theorem (Rudi, C., Rosasco '17)

For $\|\phi(x)\|, |y| \leq 1$, *when* $M > \frac{\log n}{\lambda}$,

$$\mathbb{E}\mathcal{E}(\widehat{f}_{\lambda_n, M_n, t_n}) - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{1}{\lambda n} + \lambda + \frac{1}{M} + \exp\left[-t\ \left(1 - \frac{\log n}{\lambda M}\right)^{1/2}\right]$$

By selecting

$$\lambda_n = \frac{1}{\sqrt{n}}, \qquad M_n = \frac{2\log n}{\lambda}, \qquad t_n = \log n,$$

then

$$\mathbb{E}\mathcal{E}(\widehat{f}_{\lambda_n, M_n, t_n}) - \mathcal{E}(f_{\mathcal{H}}) \lesssim \frac{1}{\sqrt{n}}$$

Remarks

- Same rates and memory of NKRR, much smaller time complexity, for $O(1/\sqrt{n})$:

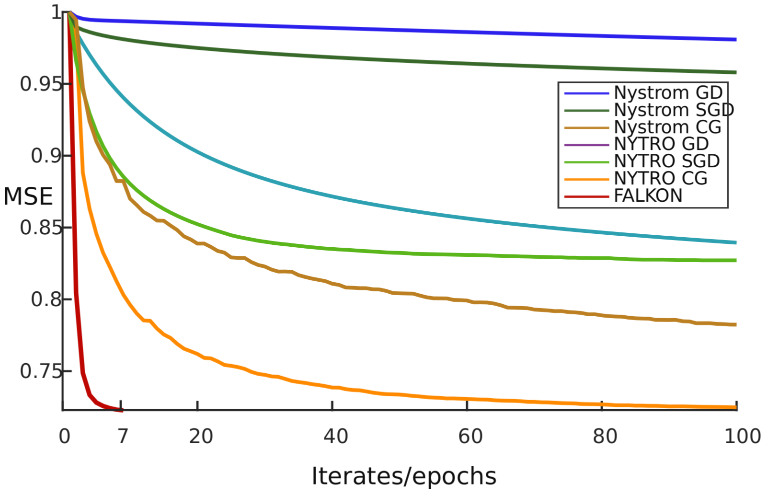
$$\begin{aligned} \text{Model:} \quad & O(\sqrt{n}) \\ \text{Space:} \quad & O(n) \\ \text{Kernel eval.:} \quad & O(n\sqrt{n}) \\ \text{Time:} \quad & \cancel{O(n^2)} \to O(n\sqrt{n}) \end{aligned}$$

Related

- EigenPro (Belkin et al. '16)
- SGD (Smale, Yao '05, Tarres, Yao '07, Ying, Pontil '08, Bach et al. '14-. . . ,)
- RF-KRR (Rahimi, Recht '07; Bach '15; Rudi, Rosasco '17)
- Divide and conquer (Zhang et al. '13)
- NYTRO (Angles et al '16)
- Nyström SGD (Lin, Rosasco '16)
- SGD-RF (C., Rosasco '18)

In practice

Higgs dataset: $n = 10,000,000, \quad M = 50,000$



Some experiments

	MillionSongs ($n \sim 10^6$)			YELP ($n \sim 10^6$)		TIMIT ($n \sim 10^6$)	
	MSE	Relative error	Time(s)	RMSE	Time(m)	c-err	Time(h)
FALKON	**80.30**	$\mathbf{4.51 \times 10^{-3}}$	**55**	**0.833**	**20**	32.3%	**1.5**
Prec. KRR	-	4.58×10^{-3}	$289^\dagger$	-	-	-	-
Hierarchical	-	4.56×10^{-3}	$293^\star$	-	-	-	-
D&C	80.35	-	737^*	-	-	-	-
Rand. Feat.	80.93	-	772^*	-	-	-	-
Nyström	80.38	-	876^*	-	-	-	-
ADMM R. F.	-	5.01×10^{-3}	$958^\dagger$	-	-	-	-
BCD R. F.	-	-	-	0.949	$42^\ddagger$	34.0%	$1.7^\ddagger$
BCD Nyström	-	-	-	0.861	$60^\ddagger$	33.7%	$1.7^\ddagger$
KRR	-	4.55×10^{-3}	-	0.854	$500^\ddagger$	33.5%	$8.3^\ddagger$
EigenPro	-	-	-	-	-	32.6%	$3.9^\wr$
Deep NN	-	-	-	-	-	32.4%	-
Sparse Kernels	-	-	-	-	-	**30.9%**	-
Ensemble	-	-	-	-	-	33.5%	-

Table: MillionSongs, YELP and TIMIT Datasets. Times obtained on: $\ddagger$ = cluster of 128 EC2 r3.2xlarge machines, $\dagger$ = cluster of 8 EC2 r3.8xlarge machines, $\wr$ = single machine with two Intel Xeon E5-2620, one Nvidia GTX Titan X GPU and 128GB of RAM, $\star$ = cluster with 512 GB of RAM and IBM POWER8 12-core processor, $*$ = unknown platform.

Some more experiments

	SUSY ($n \sim 10^6$)			HIGGS ($n \sim 10^7$)		IMAGENET ($n \sim 10^6$)	
	c-err	AUC	Time(m)	AUC	Time(h)	c-err	Time(h)
FALKON	**19.6%**	0.877	**4**	0.833	**3**	20.7%	**4**
EigenPro	19.8%	-	$6^{\wr}$	-	-	-	-
Hierarchical	20.1%	-	$40^{\dagger}$	-	-	-	-
Boosted Decision Tree	-	0.863	-	0.810	-	-	-
Neural Network	-	0.875	-	0.816	-	-	-
Deep Neural Network	-	**0.879**	$4680^{\ddagger}$	**0.885**	$78^{\ddagger}$	-	-
Inception-V4	-	-	-	-	-	**20.0%**	-

Table: Architectures: † cluster with IBM POWER8 12-core cpu, 512 GB RAM, ≀ single machine with two Intel Xeon E5-2620, one Nvidia GTX Titan X GPU, 128GB RAM, ‡ single machine.

Contributions

- Best computations so far for optimal statistics

$$\boxed{\textbf{Space } O(n) \qquad \textbf{Time } O(n\sqrt{n})}$$

Other flavours:

- SGD, mini-batching, random features [C., Rudi, Rosasco 18']
- adaptive sampling [Rudi, Calandriello, C., Rosasco 18']

- In the pipeline: accelerated stochastic methods, distributed optimization
- TBD: other loss, other regularizers, other problems, other solvers. . .