MS-26: Large-Scale Optimization and Applications

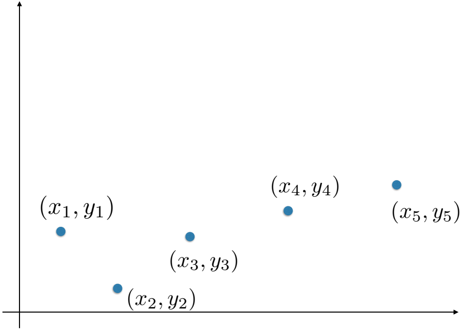
Stochastic Preconditioning for Optimal Large-Scale Machine Learning

Luigi Carratino
University of Genova

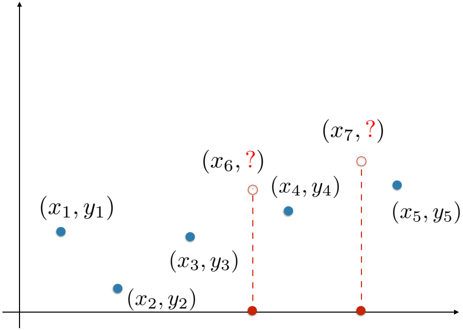
joint work with Alessandro Rudi (INRIA), Lorenzo Rosasco (UniGe, MIT, IIT)

Jul, 6th 2018 – SIMAI 2018

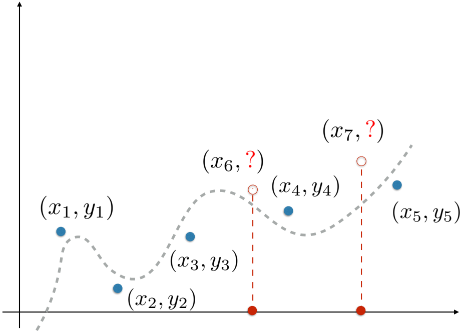
Learning



Learning



Learning



Problem: **given** $\{(x_1, y_1), \dots, (x_n, y_n)\}$ **find** $f(x_{\text{new}}) \sim y_{\text{new}}$

Learning problem

The problem $\mathcal{P}$

Find

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

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

Remarks:

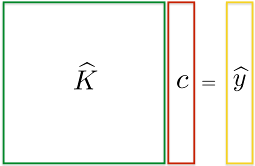
- $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ RKHS with bounded kernel K (e.g. $K(x,x') = e^{-\gamma \|x - x'\|^2}$)
- $\mathcal{H} = \overline{\operatorname{span}\{K(x,\cdot) | x \in X\}}$

Kernel ridge regression

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

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

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



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

Random projections

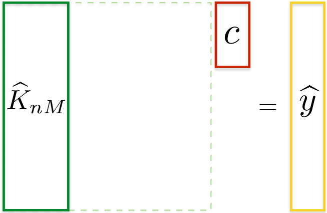
Solve $\widehat{\mathcal{P}}_n$ on $\mathcal{H}_M = \operatorname{span}\{K(\tilde{x}_1,\cdot),\dots,K(\tilde{x}_M,\cdot)\}$

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

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

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

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



$\widehat{K}_{nM}$ c $=$ $\widehat{y}$

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

Nyström KRR: Statistics

Theorem[Rudi, Camoriano, R. '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

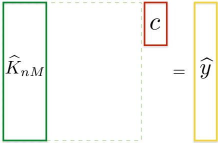
Space:	$O(n)$
Kernel eval.:	$O(n\sqrt{n})$
Time:	$O(n^2)$
Test:	$O(\sqrt{n})$

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

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

Note: Preconditioning KRR (Fasshauer et al '12, Avron et al '16, Cutajat '16, Ma, Belkin '17)

$$H = K + \lambda n I$$

Can we precondition Nystrom-KRR?

Preconditioning Nystom-KRR

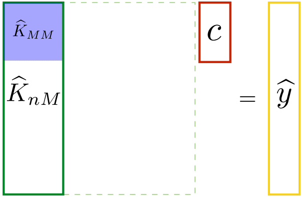
Consider $$H := \widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM}$$

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 ↦ 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

Under (basic), 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 $O(1/\sqrt{n})$:

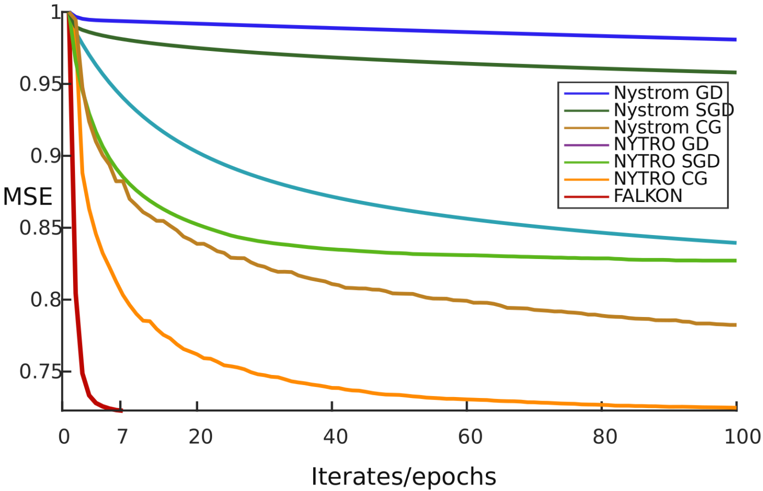
Model:	$O(\sqrt{n})$
Space:	$O(n)$
Kernel eval.:	$O(n\sqrt{n})$
Time:	$\cancel{O(n^2)} \to O(n\sqrt{n})$

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)

In practice

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



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

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

- In the pipeline: adaptive sampling, general projection, SGD
- TBD other loss, other regularizers, other problems, other solvers. . .