UniGe | MaLGa

Efficient learning with Nyström projections

Lorenzo Rosasco
MaLGa, Università degli Studi di Genova, MIT, IIT

Joint with:
Daniele Calandriello, Raffaello Camoriano, Luigi Carratino, Giacomo Meanti, Francesca Odone, Paolo Alfano, Vito Paolo Pastore (MaLGa), Alessandro Rudi (INRIA Paris), Bharath Sriperumbudur, Nick Sterge (UPenn), Alessandro Lazaric (Facebook), Michal Valko (DeepMind)

Data

⟺

Computations

The quest for provably efficient ML algorithms

Outline

Different views on Nyström projections

Supervised statistical learning

Bandit Optimization

Unsupervised statistical learning

Data size matters

$$\underbrace{\hat{X}}_{n \times d}$$

In many modern applications, space is the real constraint.

Think n, d large!

UniGe | MaLGa

Dimensionality reduction

$$\underbrace{\widehat{X}_M}_{n \times M} = \underbrace{\widehat{X}}_{n \times d} \underbrace{S}_{d \times M}$$

UniGe | MaLGa

Dimensionality reduction

$$\underbrace{\widehat{X}_M}_{n \times M} = \underbrace{\widehat{X}}_{n \times d} \underbrace{S}_{d \times M}$$

Classic example (not efficient): PCA

$$\widehat{X} = \widehat{U}\widehat{\Lambda}\widehat{V}^T \quad \Rightarrow S = \widehat{V}_M.$$

Dimensionality reduction

$$\underbrace{\widehat{X}_M}_{n \times M} = \underbrace{\widehat{X}}_{n \times d} \underbrace{S}_{d \times M}$$

Classic example (not efficient): PCA

$$\widehat{X} = \widehat{U}\widehat{\Lambda}\widehat{V}^T \quad \Rightarrow S = \widehat{V}_M.$$

Other example (efficient): random sketches

$$S_{ij} \sim \mathcal{N}(0, 1).$$

Nyström projections

$$\widehat{X}_M = \widehat{X}\overline{X}_M^\top$$

Random subsampling[1] (efficient)

$\overline{X}_M = \{\overline{x}_1, \ldots, \overline{x}_M\} \subset \{x_1, \ldots, x_n\} = \widehat{X}$.

[1] It's data dependent sketching NOT throwing data away!!!

Nyström projections

$$\widehat{X}_M = \widehat{X}\overline{X}_M^\top$$

Random subsampling[1] (efficient)
$\overline{X}_M = \{\overline{x}_1, \dots, \overline{x}_M\} \subset \{x_1, \dots, x_n\} = \widehat{X}$.

Computing $\widehat{X}_M$ is efficient!

[Williams, Seeger, Smola, Schölkopf, Bach, Muscos, Clarkson, Mahoney, Woodruff, Avron, Drineas, Tropp, ...]

[1] It's data dependent sketching NOT throwing data away!!!

Nyström projections illustrated: least squares

From

$$\min_{w \in \mathbb{R}^d} \|\widehat{X}w - \widehat{y}\|^2, \qquad \widehat{X} \in \mathbb{R}^{n,d}$$

UniGe | MaLGa

Nyström projections illustrated: least squares

From

$$\min_{w \in \mathbb{R}^d} \|\widehat{X}w - \widehat{y}\|^2, \qquad \widehat{X} \in \mathbb{R}^{n,d}$$

to

$$\min_{c \in \mathbb{R}^M} \|\widehat{X}_M c - \widehat{y}\|^2, \qquad \widehat{X}_M \in \mathbb{R}^{n,M}$$

Nyström projections illustrated: least squares

From

$$\min_{w \in \mathbb{R}^d} \|\widehat{X} w - \widehat{y}\|^2, \qquad \widehat{X} \in \mathbb{R}^{n,d}$$

to

$$\min_{c \in \mathbb{R}^M} \|\widehat{X}_M c - \widehat{y}\|^2, \qquad \widehat{X}_M \in \mathbb{R}^{n,M}$$

The latter problem is equivalent to

$$\min_{w = \overline{X}_M^\top c, \quad c \in \mathbb{R}^M} \|\widehat{X} w - y\|^2,$$

that is least squares projected on a random subspace.

[Engl, Hanke, Neubauer '96]

UniGe | MaLGa

Nyström least squares: computations

(think n huge d ginormous)

From

$$w = \widehat{X}^\top c, \qquad c = (\underbrace{\widehat{X}\widehat{X}^\top}_{\widehat{K} \in \mathbb{R}^{n,n}})^{-1} \widehat{y} \in \mathbb{R}^{n}$$

Nyström least squares: computations

(think n huge d ginormous)

From

$$w = \widehat{X}^\top c, \qquad c = (\underbrace{\widehat{X}\widehat{X}^\top}_{\widehat{K} \in \mathbb{R}^{n,n}})^{-1}\widehat{y} \in \mathbb{R}^{n}$$

to

$$c = (\widehat{X}_M^\top \underbrace{\widehat{X}_M}_{\widehat{K}_{nM} \in \mathbb{R}^{n,M}})^{-1}\widehat{X}_M^\top \widehat{y} \in \mathbb{R}^{M}$$

Nyström least squares: computations

(think n huge d ginormous)

From

$$w = \widehat{X}^\top c, \qquad c = (\underbrace{\widehat{X}\widehat{X}^\top}_{\widehat{K} \in \mathbb{R}^{n,n}})^{-1}\widehat{y} \in \mathbb{R}^{n}$$

to

$$c = (\widehat{X}_M^\top \underbrace{\widehat{X}_M}_{\widehat{K}_{nM} \in \mathbb{R}^{n,M}})^{-1}\widehat{X}_M^\top\widehat{y} \in \mathbb{R}^{M}$$

From $O(n^2 d + n^3)$ time/$O(nd + n^2)$ space to $O(nM^2 + M^3)$ time/$O(nM + M^2)$ space.

UniGe | MaLGa

This matters for kernel methods

$$x^\top x' \quad \mapsto \quad k(x, x'), \qquad \text{e.g.} \quad k(x, x') = e^{-\|x - x'\|^2 \gamma}$$

UniGe | MaLGa

This matters for kernel methods

$$x^\top x' \quad \mapsto \quad k(x, x'), \qquad \text{e.g.} \quad k(x, x') = e^{-\|x - x'\|^2 \gamma}$$

$$x^\top w = x^\top \widehat{X}^\top c \quad \mapsto \quad f(x) = \sum_{i=1}^{n} k(x, x_i) c_i$$

This matters for kernel methods

$$x^\top x' \quad \mapsto \quad k(x, x'), \qquad \text{e.g.} \quad k(x, x') = e^{-\|x - x'\|^2 \gamma}$$

$$x^\top w = x^\top \widehat{X}^\top c \quad \mapsto \quad f(x) = \sum_{i=1}^{n} k(x, x_i) c_i$$

$$\widehat{X} w = \widehat{X}\widehat{X}^\top c = \widehat{y} \quad \mapsto \quad \widehat{K} c = \widehat{y}$$

$$\widehat{K}_{i,j} = k(x_i, x_j)$$

UniGe | MaLGa

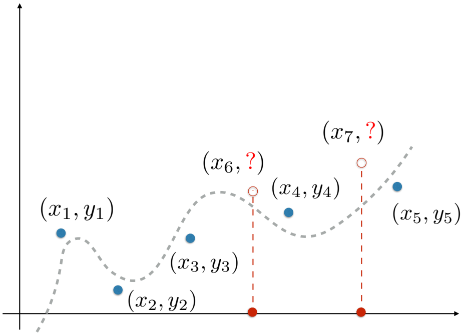
This matters for kernel methods

$$x^\top x' \quad \mapsto \quad k(x, x'), \qquad \text{e.g.} \quad k(x, x') = e^{-\|x-x'\|^2 \gamma}$$

$$x^\top w = x^\top \widehat{X}^\top c \quad \mapsto \quad f(x) = \sum_{i=1}^{n} k(x, x_i) c_i$$

$$\widehat{X} w = \widehat{X}\widehat{X}^\top c = \widehat{y} \quad \mapsto \quad \widehat{K} c = \widehat{y}$$

$$\widehat{K}_{i,j} = k(x_i, x_j)$$



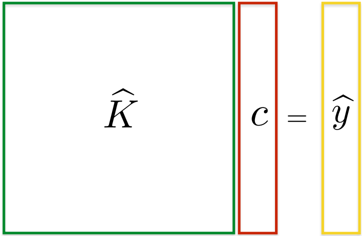
UniGe | MaLGa

Nyström projections with kernels, aka column subsampling

$$\widehat{X}w = \widehat{X}\widehat{X}^\top c = \widehat{y}$$

$$\downarrow$$

$$f(x) = \sum_{i=1}^{n} k(x, x_i)c_i \qquad \boxed{\widehat{K}c = \widehat{y}}$$



Nyström projections with kernels, aka column subsampling

$$\widehat{X}w = \widehat{X}\widehat{X}^\top c = \widehat{y}$$

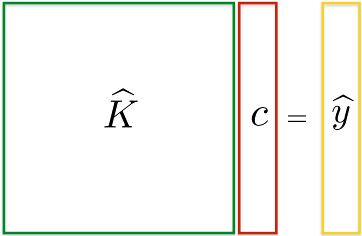
$$\downarrow$$

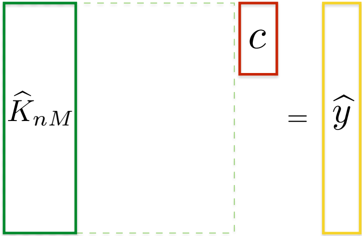
$$f(x) = \sum_{i=1}^{n} k(x, x_i)c_i \qquad \boxed{\widehat{K}c = \widehat{y}}$$

$$\widehat{X}_M c = \widehat{y}$$

$$\downarrow$$

$$f(x) = \sum_{i=1}^{n} k(x, \overline{x}_i)c_i \qquad \boxed{\widehat{K}_{nM}c = \widehat{y}}$$





From $O(n^3)$ time/$O(n^2)$ space to $O(nM^2 + M^3)$ time/$O(nM)$ space.

UniGe | MaLGa

[Williams, Seeger, Smola Scholkopf, ...Mahoney, Drineas, ...]

Why Nyström?

Nyström approximation for integral equations

For all x

$$\int k(x,x')c(x')dx' = y(x) \qquad \mapsto \qquad \sum_{j=1}^{M} k(x,\bar{x}_j)c(\bar{x}_j) = y(x).$$

From operators to matrices

For all $i = 1, \ldots, n$

$$\sum_{j=1}^{n} k(x_i,x_j)c_j = y_j \qquad \mapsto \qquad \sum_{j=1}^{M} k(x_i,\bar{x}_j)c_i = y_j.$$

[Kress '89]

Nyström approximation and subsampling

For all $i = 1, \dots, n$

$$\sum_{j=1}^{n} k(x_i, x_j)c_j = y_j \qquad \mapsto \qquad \sum_{j=1}^{M} k(x_i, \overline{x}_j)c_i = y_j.$$

[Williams, Seeger '00]

The above formulation highlights the connection to columns sampling,

$$\widehat{K}c = \widehat{y} \qquad \mapsto \qquad \widehat{K}_{nM}c = \widehat{y}.$$

So far

Nyström projections and connection to

- Sketching
- Projected least squares
- Column subsampling
- Nyström approximation

$$\widehat{X}_M = \widehat{X X_M^\top}$$

Dimensionality reduction improves efficiency, but what about learning accuracy?

Outline

Different views on Nyström projections

Supervised statistical learning

Bandit Optimization

Unsupervised statistical learning

Supervised statistical Learning

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

Solve

$$\min_{f \in \mathcal{H}} L(f), \qquad\qquad L(f) = \mathbb{E}_{x,y}(y - f(x))^2$$

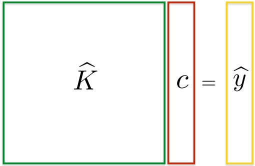
given $(x_i, y_i)_{i=1}^n \sim \rho^n$.

Kernel Ridge Regression (KRR)

aka Gaussian Process (GP) regression

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

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



Theorem (Caponetto, De Vito ’05)

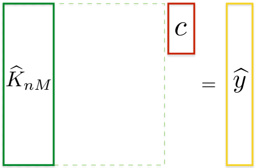
Let $\mathcal{H} = \mathrm{span}\{k(x,\cdot) \mid x \in X\}$, *if* $\lambda = 1/\sqrt{n}$ *then*

$$\mathbb{E}L(\widehat{f}_\lambda) - \min_{f \in \mathcal{H}} L(f) \lesssim \frac{1}{\sqrt{n}}$$

Nyström KRR

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

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



Theorem (Rudi, Camoriano, R. '15)

Let $(\widetilde{x}_i)_{i=1}^M \subseteq (x_i)_{i=1}^n$ *picked uniformly at random, if* $\lambda = 1/\sqrt{n}$ *and* $M \geqslant \sqrt{n}$ *then*

$$\mathbb{E} L(\widehat{f}_{\lambda,M}) - \min_{f \in \mathcal{H}} L(f) \lesssim \frac{1}{\sqrt{n}}$$

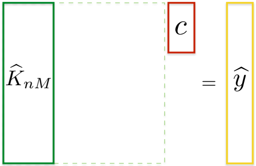
Iterative solvers and preconditoning

Consider an iterative solver, e.g. conjugate gradient (CG), on a preconditioned system

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

...ideally

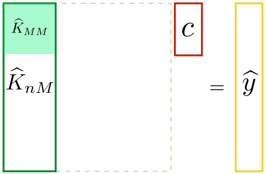
$$PP^\top = (\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM})^{-1}$$



FALKON

$\widehat{f}_{\lambda,M,t}$ CG iteration with preconditioner

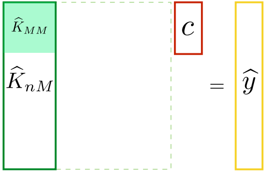
$$PP^{\top} = \left(\frac{n}{M}\widehat{K}^2_{MM} + \lambda n \widehat{K}_{MM}\right)^{-1}$$



FALKON

$\widehat{f}_{\lambda,M,t}$ CG iteration with preconditioner

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



Theorem (Rudi, Carratino, Rosasco '17)

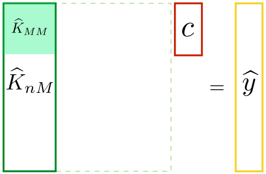
Let $(\widetilde{x}_i)_{i=1}^M \subseteq (x_i)_{i=1}^n$ *uniformly at random*, *then if* $\lambda = 1/\sqrt{n}$, $M \geqslant \sqrt{n}$ *and* $t \geqslant \log(n)$

$$\mathbb{E}L(\widehat{f}_{\lambda,M,t}) - \min_{f \in \mathcal{H}} L(f) \lesssim \frac{1}{\sqrt{n}}$$

FALKON

$\widehat{f}_{\lambda,M,t}$ CG iteration with preconditioner

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



Theorem (Rudi, Carratino, Rosasco '17)

Let $(\widetilde{x}_i)_{i=1}^M \subseteq (x_i)_{i=1}^n$ *uniformly at random, then if* $\lambda = 1/\sqrt{n}$, $M \geqslant \sqrt{n}$ *and* $t \geqslant \log(n)$

$$\mathbb{E}L(\widehat{f}_{\lambda,M,t}) - \min_{f\in\mathcal{H}} L(f) \lesssim \frac{1}{\sqrt{n}}$$

KRR: Space $O(n^2)$ **/ Time** $O(n^3)$ vs **FALKON: Space** $O(n)$ **/ Time** $O(n\sqrt{n}\log(n))$

UniGe | MaLGa

Falkon 1.0: 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.

UniGe | MaLGa

Falkon 1.0: 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%**	-

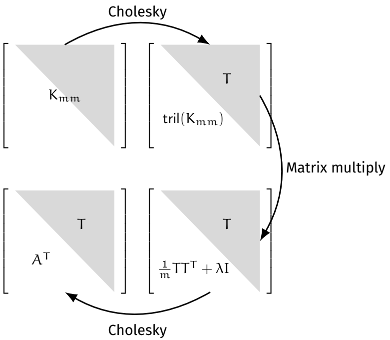
Table: Architectures: $\dagger$ = cluster with IBM POWER8 12-core cpu, 512 GB RAM, $\wr$ = single machine with two Intel Xeon E5-2620, one Nvidia GTX Titan X GPU, 128GB RAM, $\ddagger$ = single machine.

Implementing Falkon 2.0

[Meanti, Carratino, R., Rudi '20]

Function $\texttt{Falkon}(X \in \mathbb{R}^{n\times d}, y \in \mathbb{R}^n, \lambda, m, t)$:

- $X_m \leftarrow \texttt{RamdomSubsample}(X, m)$;
- $T, A \leftarrow \texttt{Preconditioner}(X_m, \lambda)$;
- **Function** $\texttt{LinOp}(\beta)$:
  - $v \leftarrow A^{-1}\beta$;
  - $c \leftarrow k(X_m, X)k(X, X_m)T^{-1}v$;
  - **return** $A^{-\top}T^{-\top}c + \lambda n v$;
- $\text{rhs} \leftarrow A^{-\top}T^{-\top}k(X, X_m)y$;
- $\beta \leftarrow \texttt{ConjugateGradient}(\texttt{LinOp}, \text{rhs}, t)$;
- **return** $T^{-1}A^{-1}\beta$;



Falkon2.0

- Least squares and logisitic loss [Marteau Ferey, Bach, Rudi ’18,’19]

- Multi-GPU
- Mixed precision
- Optimized matrix-vector product
- Optimized kernel computation
- Out of core modules

Falkon2.0

Table: Relative performance improvement wrt Falkon 1.0

Experiment	Preconditioner		Iterations	
	Time	Improvement	Time	Improvement
Falkon1.0	2337 s	—	4565 s	—
Float32 precision	1306 s	1.8×	1496 s	3×
GPU preconditioner	179 s	7.3×	1344 s	1.1×
2 GPUs	118 s	1.5×	693 s	1.9×
KeOps	119 s	1×	232 s	3×
Overall improvement		19.7×		18.8×

UniGe | MaLGa

Falkon2.0: thousands of points in seconds

	MNIST $n = 6 \cdot 10^4, d = 780$	CIFAR10 $n = 6 \cdot 10^4, d = 1024$	SVHN $n = 7 \cdot 10^4, d = 1024$
Falkon	10.9 s	13.7 s	17.2 s
ThunderSVM	19.6 s	82.9 s	166.4 s

Table: Comparing running times of FALKON and ThunderSVM. Parameters were tuned to have approximately the same accuracy.

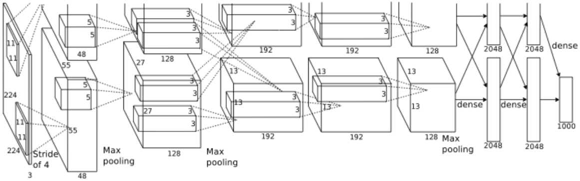
Falkon2.0: millions/billions (!) of points in minutes

	TAXI $n \approx 10^9$		HIGGS $n \approx 10^7$		YELP $n \approx 10^6$, $d \approx 10^7$		TIMIT $n \approx 10^6$	
	RMSE	time(h)	AUC	time(m)	RMSE	time(m)	c-err	time(m)
FALKON	**311.7**	**1**	0.8196	**7.4**	**0.810**	**16.8**	32.27%	**4.8**
LogFALKON	–	–	**0.8213**	37.8	–	–	–	–
EigenPro2	FAIL		FAIL		FAIL		**31.91%**	29
GPyTorch	322.5	10.8	0.8005	52.9	FAIL		–	–
GPflow	313.2	8.5	0.8042	24.3	FAIL		33.78%	44.5

	AIRLINE-CLS $n \approx 10^6$		AIRLINE $n \approx 10^6$		MSD $n \approx 10^5$		SUSY $n \approx 10^6$	
	c-err	time(m)	MSE	time(m)	relative error	time(m)	c-err	time(m)
FALKON	31.5%	**3.1**	**0.758**	**4.1**	4.4834×10^{-3}	**1**	19.67%	**0.4**
LogFALKON	**31.3%**	21.5	–	–	–	–	**19.58%**	1.4
EigenPro2	32.5%	27.2	0.785	24.5	4.4778×10^{-3}	6.3	20.08%	1.5
GPyTorch	33.0%	24.2	0.803	31	4.5344×10^{-3}	15.5	19.71%	16.5
GPflow	32.6%	10.5	0.790	28.7	4.4986×10^{-3}	8.8	19.65%	9.3

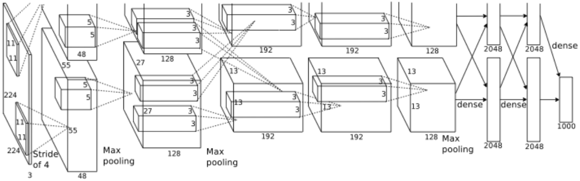
Deep neural networks (DNN)

$$f(x) = \langle w, \Phi(x) \rangle\,, \qquad \underbrace{x \mapsto \Phi_L \circ \cdots \circ \Phi_1(x)}_{\text{compositional representation}}$$



Convolutional and fully connected DNN

$$f(x) = \langle w, \Phi(x) \rangle \,, \qquad x \mapsto \underbrace{\Phi_L \circ \ldots \Phi_K}_{\text{Fully connected}} \circ \underbrace{\Phi_{K-1} \cdots \circ \Phi_1(x)}_{\text{Convolutional}}$$



- Convolutional layers, thousands parameters.
- Fully connected layers, million parameters.

$\rightarrow$ End to end learning.

(LeCun et al. '98)

UniGe | MaLGa

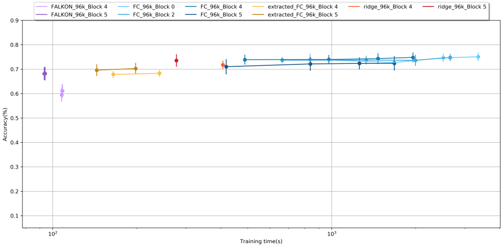
Trading DNN for kernel methods

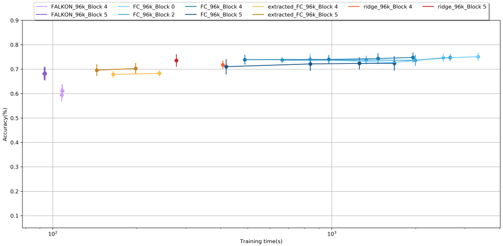
$$f(x) = \langle w, \Phi(x) \rangle\,, \quad x \mapsto \underbrace{\Phi_L}_{\text{Kernel representation}} \circ \underbrace{\Phi_{L-1} \cdots \circ \Phi_1(x)}_{\text{Convolutional}}$$

- not about data representation...
- ...but scaling kernel methods to millions of points.

Don’t fine tune, use kernel methods

[Alfano, Pastore, R., Odone ’20]





UniGe | MaLGa

Outline

Different views on Nyström projections

Supervised statistical learning

Bandit Optimization

Unsupervised statistical learning

Bandit Optimization

Given a set (of arms) $\mathcal{A} = \{x_1, \ldots, x_A\} \subset \mathbb{R}^d$, let $f : \mathcal{A} \to \mathbb{R}$ unknown, $(\eta_t)_t$ random, and

$$x_* = \underset{x \in \mathcal{A}}{\operatorname{argmax}}\, f(x)$$

Bandit Optimization

Given a set (of arms) $\mathcal{A} = \{x_1, \ldots, x_A\} \subset \mathbb{R}^d$, let $f : \mathcal{A} \to \mathbb{R}$ unknown, $(\eta_t)_t$ random, and

$$x_* = \operatorname*{argmax}_{x \in \mathcal{A}} f(x)$$

For $t = 1, \ldots, T$:

(1) Estimate $\widehat{u}_t$ (ideally $\widehat{u}_t \approx f$)

Bandit Optimization

Given a set (of arms) $\mathcal{A} = \{x_1, \ldots, x_A\} \subset \mathbb{R}^d$, let $f : \mathcal{A} \to \mathbb{R}$ unknown, $(\eta_t)_t$ random, and

$$x_* = \operatorname*{argmax}_{x \in \mathcal{A}} f(x)$$

For $t = 1, \ldots, T$:

(1) Estimate $\widehat{u}_t$ (ideally $\widehat{u}_t \approx f$)

(2) Select x_{t+1}

Bandit Optimization

Given a set (of arms) $\mathcal{A} = \{x_1, \ldots, x_A\} \subset \mathbb{R}^d$, let $f : \mathcal{A} \to \mathbb{R}$ unknown, $(\eta_t)_t$ random, and

$$x_* = \operatorname*{argmax}_{x \in \mathcal{A}} f(x)$$

For $t = 1, \ldots, T$:

(1) Estimate $\widehat{u}_t$ (ideally $\widehat{u}_t \approx f$)

(2) Select x_{t+1}

(3) Receive noisy feedback $y_{t+1} = f(x_{t+1}) + \eta_{t+1}$

Bandit Optimization

Given a set (of arms) $\mathcal{A} = \{x_1, \ldots, x_A\} \subset \mathbb{R}^d$, let $f : \mathcal{A} \to \mathbb{R}$ unknown, $(\eta_t)_t$ random, and

$$x_* = \operatorname*{argmax}_{x \in \mathcal{A}} f(x)$$

For $t = 1, \ldots, T$:

(1) Estimate $\widehat{u}_t$ (ideally $\widehat{u}_t \approx f$)

(2) Select x_{t+1}

(3) Receive noisy feedback $y_{t+1} = f(x_{t+1}) + \eta_{t+1}$

Goal: minimize cumulative regret

$$R_T = \sum_{t=1}^{T} f(x_*) - f(x_t)$$

UniGe | MaLGa

Gaussian processes

aka kernel ridge regression

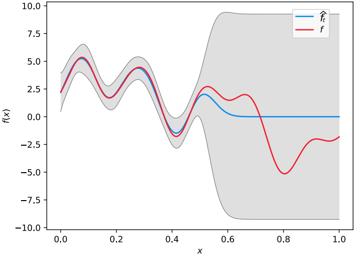
$\widehat{K}_t \in \mathbb{R}^{t,t}$ s.t. $(\widehat{K})_{i,j} = k(x_i, x_j)$, $i, j = 1, \dots, t$

$\widehat{k}_t(x) = (k(x_1, x), \dots, k(x_t, x)) \in \mathbb{R}^t$

$\widehat{y}_t = (y_1, \dots, y_t)$

$$\widehat{f}_t(x) = \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{y}_t$$

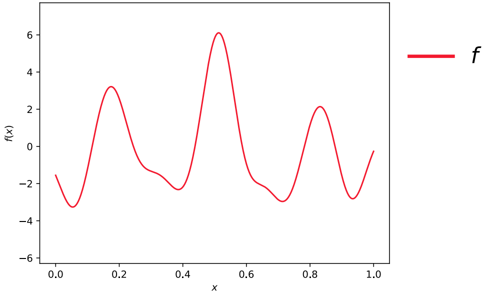
$$\underbrace{\sigma_t^2(x)}_{\text{variance}} = k(x, x) - \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{k}_t(x)$$



UniGe | MaLGa

GP-UCB

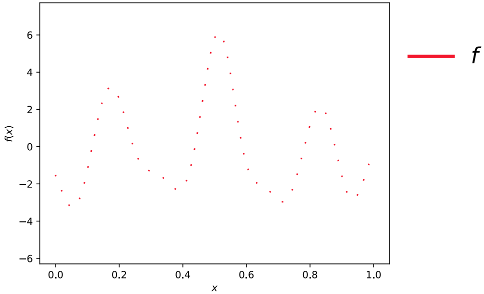
$f : \mathcal{A} \to \mathbb{R}$ unknown



(Srinivas, Krause, Kakade, Seeger ’10)

GP-UCB

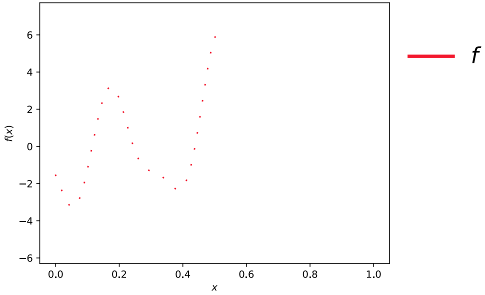
Arms $\mathcal{A} = \{x_i\}_{i=1}^{A}$



(Srinivas, Krause, Kakade, Seeger '10)

GP-UCB

At time t, collected $(x_i, y_i)_{i=1}^{t}$



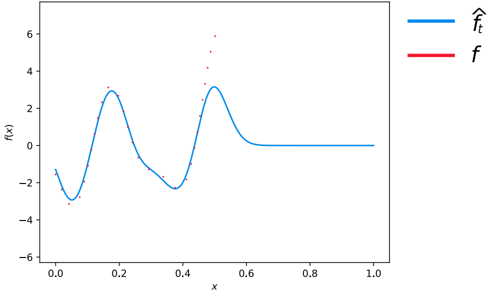
for $t = \{1, \ldots, T-1\}$ **do**

end

(Srinivas, Krause, Kakade, Seeger ’10)

GP-UCB

At time t, collected $(x_i, y_i)_{i=1}^{t}$



for $t = \{1, \ldots, T-1\}$ **do**

end

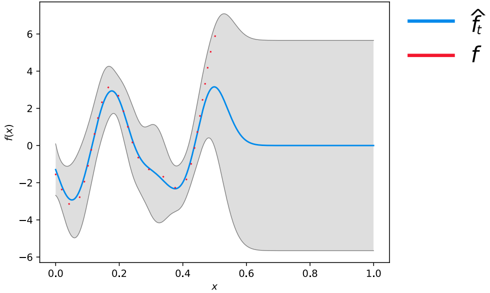
$$\widehat{f}_t(x) = \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{y}_t$$

(Srinivas, Krause, Kakade, Seeger '10)

UniGe | MaLGa

GP-UCB

At time t, collected $(x_i, y_i)_{i=1}^t$



for $t = \{1, \ldots, T-1\}$ **do**

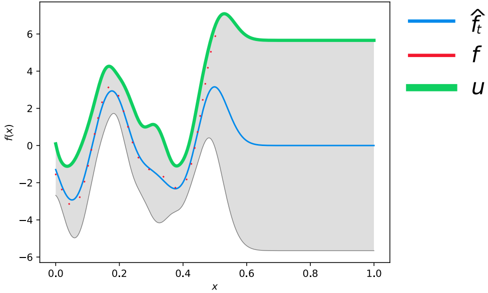
end

$$\widehat{f}_t(x) = \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{y}_t \qquad \sigma_t^2(x) = k(x, x) - \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{k}_t(x)$$

(Srinivas, Krause, Kakade, Seeger ’10)

GP-UCB

At time t, collected $(x_i, y_i)_{i=1}^t$



for $t = \{1, \ldots, T-1\}$ **do**

  for $i = \{1, \ldots, A\}$ **do**

    $u_t(x_i) = \hat{f}_t(x_i) + \beta_t \sigma_t^2(x_i)$;

  end

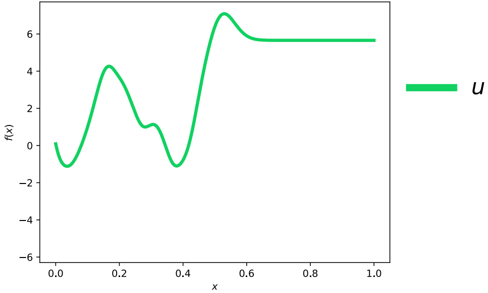
end

$$u_t(x) = \hat{f}_t(x) + \beta_t \sigma_t(x)$$

(Srinivas, Krause, Kakade, Seeger ’10)

GP-UCB

At time t, collected $(x_i, y_i)_{i=1}^{t}$



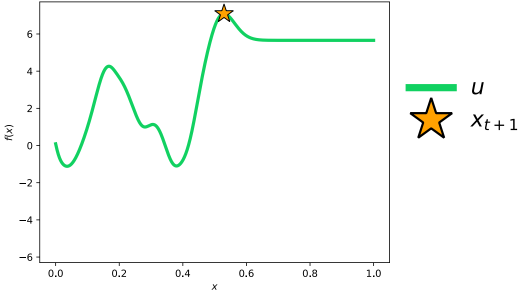
for $t = \{1, \ldots, T-1\}$ **do**
  for $i = \{1, \ldots, A\}$ **do**
    $u_t(x_i) = \widehat{f}_t(x_i) + \beta_t \sigma_t^2(x_i)$;
  end
end

$$u_t(x) = \widehat{f}_t(x) + \beta_t \sigma_t(x)$$

(Srinivas, Krause, Kakade, Seeger '10)

GP-UCB

At time t, collected $(x_i, y_i)_{i=1}^{t}$

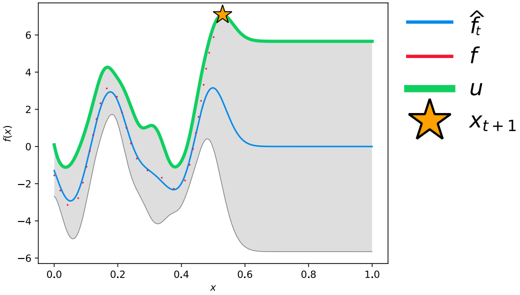


for $t = \{1, \dots, T-1\}$ **do**
  for $i = \{1, \dots, A\}$ **do**
    $u_t(x_i) = \widehat{f}_t(x_i) + \beta_t \sigma_t^2(x_i)$;
  end
  Select $x_{t+1} \leftarrow \text{argmax}_{x_i \in \mathcal{A}}\, u_t(x_i)$;
end

$$u_t(x) = \widehat{f}_t(x) + \beta_t \sigma_t(x) \quad \rightarrow \quad x_{t+1} = \underset{x \in \mathcal{A}}{\text{argmax}}\ u_t(x)$$

(Srinivas, Krause, Kakade, Seeger ’10)

GP-UCB: Regret



Computations: **Time** $O(AT^3)$

Theorem (Srinivas, Krause, Kakade, Seeger ’10)

For the proper β_t

$$R_T \leqslant \sqrt{T}$$

Nyström projection again

aka Sparse GP

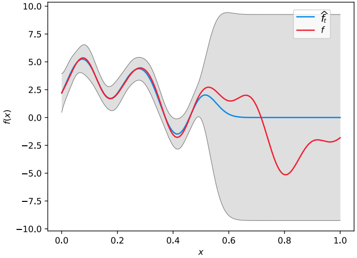
Equivalent formulation:

Given $S = (\bar{x}_i)_{i=1}^{M} \subseteq (x_i)_{i=1}^{t} \to \widetilde{k}(x, x') = \widetilde{k}_S(x)^\top \widehat{K}_S^\dagger \widetilde{k}_S(x')$,

with $\widehat{K}_S \in \mathbb{R}^{M,M}$ s.t. $(\widehat{K}_S)_{i,j} = k(\bar{x}_i, \bar{x}_j)$ and $\widetilde{k}_S(x) = (k(\bar{x}_1, x), \ldots, k(\bar{x}_M, x))$

$$\widetilde{f}_t(x) = \widetilde{k}_t(x)^\top (\widetilde{K}_t + \lambda I)^{-1} \widehat{y}_t$$

$$\widetilde{\sigma}_t^2(x) = \frac{1}{\lambda}\left(k(x, x) - \widetilde{k}_t(x)^\top (\widetilde{K}_t + \lambda I)^{-1} \widetilde{k}_t(x)\right)$$



Nyström projection again

aka Sparse GP

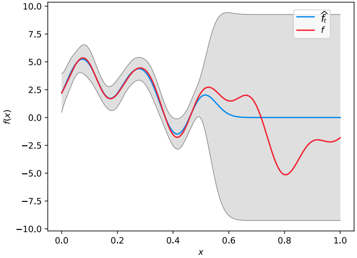
Equivalent formulation:

Given $S = (\bar{x}_i)_{i=1}^M \subseteq (x_i)_{i=1}^t \to \widetilde{k}(x, x') = \widetilde{k}_S(x)^\top \widehat{K}_S^\dagger \widetilde{k}_S(x')$,

with $\widehat{K}_S \in \mathbb{R}^{M,M}$ s.t. $(\widehat{K}_S)_{i,j} = k(\bar{x}_i, \bar{x}_j)$ and $\widetilde{k}_S(x) = (k(\bar{x}_1, x), \ldots, k(\bar{x}_M, x))$

$$\widetilde{f}_t(x) = \widetilde{k}_t(x)^\top (\widetilde{K}_t + \lambda I)^{-1} \widehat{y}_t$$

$$\widetilde{\sigma}_t^2(x) = \frac{1}{\lambda}\left(k(x, x) - \widetilde{k}_t(x)^\top (\widetilde{K}_t + \lambda I)^{-1} \widetilde{k}_t(x)\right)$$



Computations: **Time** $O(AM^2T)$

...but...

NO guarantees on regret (overconfident when S is "bad")

BKB: Regret

BKB: Regret

- $(x_i)_{i=1}^{t}$ changes with time $\rightarrow S_t$ must change with t

BKB: Regret

- $(x_i)_{i=1}^t$ changes with time $\rightarrow \mathcal{S}_t$ must change with t
- $\sigma_t^2(\cdot)$ captures informative arms $\rightarrow$ include x_i in $\mathcal{S}_t$ when $\sigma_t^2(x_i)$ is large

UniGe | MaLGa

BKB: Regret

- $(x_i)_{i=1}^t$ changes with time $\rightarrow \mathcal{S}_t$ must change with t
- $\sigma_t^2(\cdot)$ captures informative arms $\rightarrow$ include x_i in $\mathcal{S}_t$ when $\sigma_t^2(x_i)$ is large

$$\widetilde{f}_t(x) = \widetilde{k}_t(x)^\top(\widetilde{K}_t + \lambda I)^{-1}\widehat{y}$$

$$\widetilde{\sigma}_t^2(x) = \frac{1}{\lambda}\left(k(x,x) - \widetilde{k}_t(x)^\top(\widetilde{K}_t + \lambda I)^{-1}\widetilde{k}_t(x)\right)$$

for $t = \{1, \dots, T-1\}$ **do**
  for $i = \{1, \dots, A\}$ **do**
    $\widetilde{u}_t(x_i) = \widetilde{f}_t(x_i) + \widetilde{\beta}_t\widetilde{\sigma}_t^2(x_i)$;
  end
  Select $x_{t+1} \leftarrow \text{argmax}_{x_i \in \mathcal{A}}\, \widetilde{u}_t(x_i)$;
  Set $\widetilde{p}_{t+1} \propto [\widetilde{\sigma}_t^2(x_1), \dots, \widetilde{\sigma}_t^2(x_{t+1})]$;
  Sample $\mathcal{S}_{t+1} \sim \widetilde{p}_{t+1}$;
end

UniGe | MaLGa

BKB: Regret

- $(x_i)_{i=1}^t$ changes with time $\rightarrow \mathcal{S}_t$ must change with t
- $\sigma_t^2(\cdot)$ captures informative arms $\rightarrow$ include x_i in $\mathcal{S}_t$ when $\sigma_t^2(x_i)$ is large

$$\widetilde{f}_t(x) = \widetilde{k}_t(x)^\top(\widetilde{K}_t + \lambda I)^{-1}\widehat{y}$$

$$\widetilde{\sigma}_t^2(x) = \frac{1}{\lambda}\left(k(x,x) - \widetilde{k}_t(x)^\top(\widetilde{K}_t + \lambda I)^{-1}\widetilde{k}_t(x)\right)$$

for $t = \{1, \ldots, T-1\}$ **do**
  for $i = \{1, \ldots, A\}$ **do**
    $\widetilde{u}_t(x_i) = \widetilde{f}_t(x_i) + \widetilde{\beta}_t\widetilde{\sigma}_t^2(x_i)$;
  end
  Select $x_{t+1} \leftarrow \mathrm{argmax}_{x_i \in \mathcal{A}}\, \widetilde{u}_t(x_i)$;
  Set $\widetilde{p}_{t+1} \propto [\widetilde{\sigma}_t^2(x_1), \ldots, \widetilde{\sigma}_t^2(x_{t+1})]$;
  Sample $S_{t+1} \sim \widetilde{p}_{t+1}$;
end

Computations: **Time** $O(Ad_{\text{eff}}^2 T)$

Theorem (Calandriello, Carratino, Lazaric, Valko, R. '19)

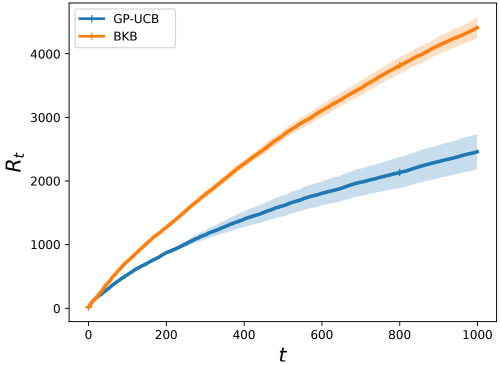
For the proper (and cheap to compute!) $\widetilde{\beta}_t$, *with* $|S_T| \leqslant d_{\mathit{eff}}$ *with* $d_{\mathit{eff}} \ll T$

$$R_T \leqslant \sqrt{T}$$

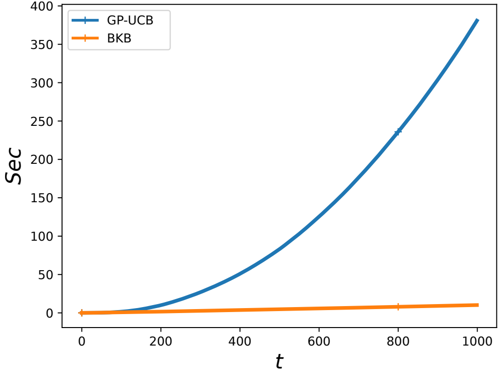
UniGe | MaLGa

In practice

Cumulative regret R_t



Times



Sublinear regret in a fraction of the time

Recent improvement using batching [Calandriello, Carratino, Lazaric, Valko, R. '20].

UniGe | MaLGa

Outline

Different views on Nyström projections

Supervised statistical learning

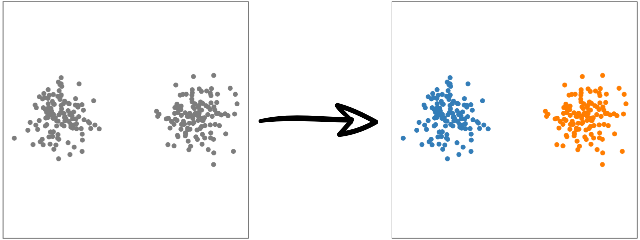
Bandit Optimization

Unsupervised statistical learning

Nyström projection for unsupervised learning?

- **Kernel K-means**
- Kernel PCA

K-means



Partition n points into k clusters.

$$\widehat{C}_K = \min_{[c_1,\dots,c_j]} \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\dots,K} \|x_i - c_j\|^2$$

Only linear separations

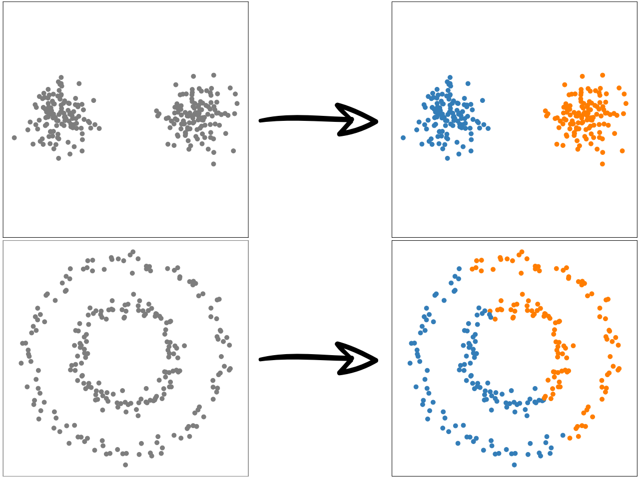
UniGe | MaLGa

K-means

Partition n points into k clusters.

$$\widehat{C}_K = \min_{[c_1,\dots,c_j]} \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\dots,K} \|x_i - c_j\|^2$$

Only linear separations



From K-means to kernel K-means

Partition n points into K clusters.

$$\widehat{C}_K = \min_{[c_1,\ldots,c_j]} \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\ldots,K} \|x_i - c_j\|^2$$

note that: $\|x - \overline{x}\|^2 = x^\top x + \overline{x}^\top \overline{x} - 2x^\top \overline{x}$

UniGe | MaLGa

From K-means to kernel K-means

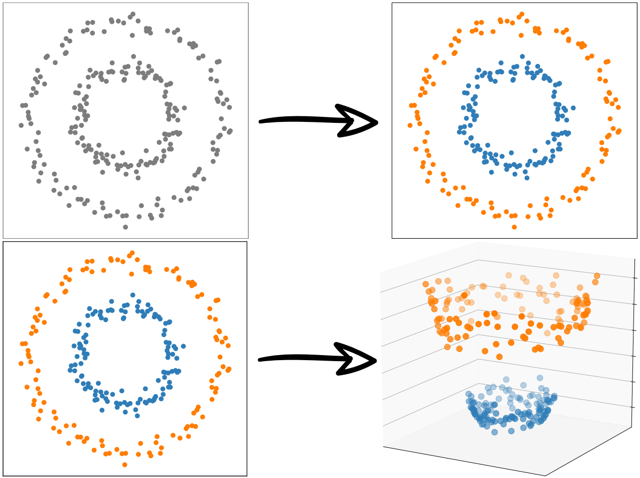
Partition n points into K clusters.

$$\widehat{C}_K = \min_{[c_1,\dots,c_j]} \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\dots,K} \|x_i - c_j\|^2$$

note that: $\|x - \overline{x}\|^2 = x^\top x + \overline{x}^\top \overline{x} - 2x^\top \overline{x}$

Kernel to rescue!

$$x^\top \overline{x} \quad \mapsto \quad k(x, \overline{x})$$



UniGe | MaLGa

From K-means to kernel K-means

Partition n points into K clusters.

$$\widehat{C}_K = \min_{[c_1,\dots,c_j]} \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\dots,K} \|x_i - c_j\|^2$$

note that: $\|x - \overline{x}\|^2 = x^\top x + \overline{x}^\top \overline{x} - 2x^\top \overline{x}$

Kernel to rescue!

$$x^\top \overline{x} \quad \mapsto \quad k(x, \overline{x})$$

$$\widehat{K}$$

From K-means to Nyström K-means

Partition n points into K clusters.

$$\widehat{C}_K = \min_{[c_1,\dots,c_j]} \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\dots,K} \|x_i - c_j\|^2$$

note that: $\|x - \overline{x}\|^2 = x^\top x + \overline{x}^\top \overline{x} - 2x^\top \overline{x}$

From K-means to Nyström K-means

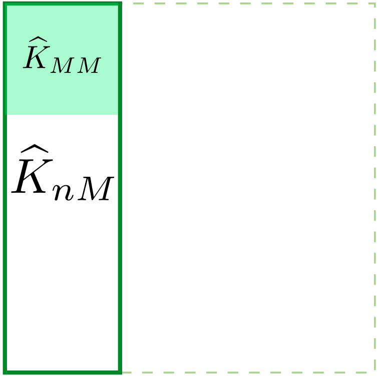
Partition n points into K clusters.

$$\widehat{C}_K = \min_{[c_1,\dots,c_j]} \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\dots,K} \|x_i - c_j\|^2$$

note that: $\|x - \overline{x}\|^2 = x^\top x + \overline{x}^\top \overline{x} - 2x^\top \overline{x}$

Nyström to rescue!

$$k(x,x) \quad \mapsto \quad \widetilde{k}(x,x) = \widetilde{k}_M(x)^\top \widehat{K}_M^\dagger \widetilde{k}_M(x')$$



Guarantees for Nyström K-means

Assume $(x_i)_{i=1}^n \sim \rho^n$, $\widehat{C}_K$ the Nyström K-means solution and [2]

$$L(\widehat{C}_K) = \mathbb{E}_x[\min_{j=1,\dots,K} \|x - c_j\|^2].$$

Theorem (Calandriello, R. '19)

Assume $\|x\| \leqslant 1$, and the Nyström centers chosen uniformly at random, then

$$\mathbb{E}[L(\widehat{C}_K)] \lesssim \frac{K}{\sqrt{n}} + \frac{K}{M},$$

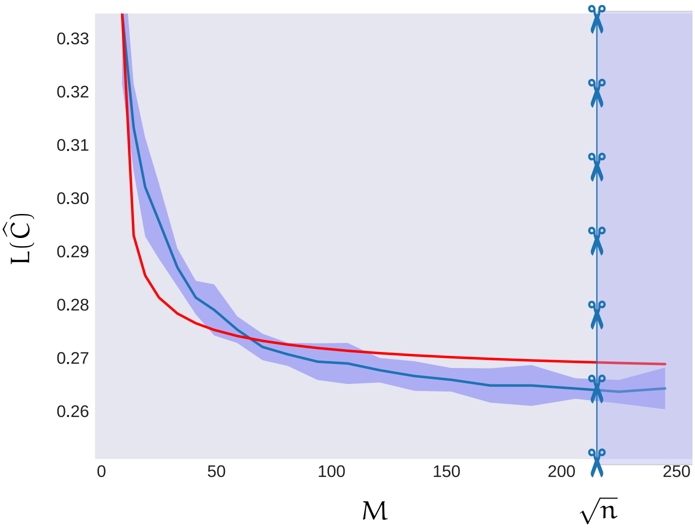
so that if $M = \sqrt{n}$ then

$$\mathbb{E}[L(\widehat{C}_K)] \lesssim \frac{K}{\sqrt{n}}.$$

The above bound matches that of exact kernel k-means.

[2] Naturally extend to kernels $x \mapsto k(x, \cdot)$.

MNIST-60k: expected loss vs projection size M



UniGe | MaLGa

Nyström projections for unsupervised learning?

- Kernel K-means
- **Kernel PCA**

UniGe | MaLGa

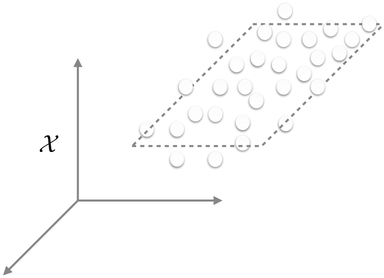
PCA

$$\widehat{\Sigma} = \frac{1}{n}\widehat{X}^\top \widehat{X} = \widehat{V}\widehat{\Lambda}^2\widehat{V}^\top$$

$$\widehat{\Lambda} = \mathrm{diag}(\widehat{\lambda}_1^2, \dots, \widehat{\lambda}_n^2).$$

$$x \quad \mapsto \quad (x^\top v_1, \dots, x^\top v_\ell)$$

Project only on linear subspacess



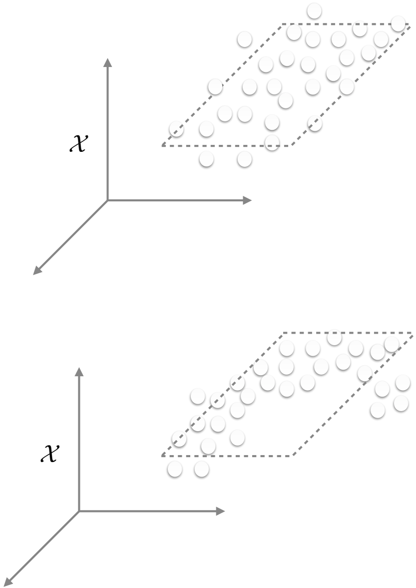
PCA

$$\widehat{\Sigma} = \frac{1}{n}\widehat{X}^\top \widehat{X} = \widehat{V}\widehat{\Lambda}^2\widehat{V}^\top$$

$$\widehat{\Lambda} = \text{diag}(\widehat{\lambda}_1^2, \dots, \widehat{\lambda}_n^2).$$

$$x \quad \mapsto \quad (x^\top v_1, \dots, x^\top v_\ell)$$

Project only on linear subspacess



Kernel PCA

$$\widehat{K} = \frac{1}{n}\widehat{X}\widehat{X}^\top = \widehat{U}\widehat{\Lambda}^2\widehat{U}$$

$$v_1 = \frac{1}{n\widehat{\lambda}_1}\widehat{X}^\top\widehat{u}_1$$

Kernel PCA

$$\widehat{K} = \frac{1}{n}\widehat{X}\widehat{X}^\top = \widehat{U}\widehat{\Lambda}^2\widehat{U}$$

$$v_1 = \frac{1}{n\widehat{\lambda}_1}\widehat{X}^\top \widehat{u}_1 \quad \Rightarrow \quad x^\top \widehat{v}_1 = \frac{1}{n\widehat{\lambda}_1}\sum_{i=1}^{n} x^\top x_i (\widehat{u}_1)_i$$

UniGe | MaLGa

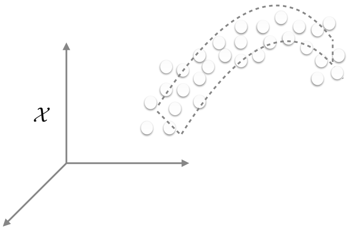
Kernel PCA

$$\widehat{K} = \frac{1}{n}\widehat{X}\widehat{X}^\top = \widehat{U}\widehat{\Lambda}^2\widehat{U}$$

$$v_1 = \frac{1}{n\widehat{\lambda}_1}\widehat{X}^\top\widehat{u}_1 \quad \Rightarrow \quad x^\top\widehat{v}_1 = \frac{1}{n\widehat{\lambda}_1}\sum_{i=1}^n x^\top x_i(\widehat{u}_1)_i$$

$$x^\top\bar{x} \quad \mapsto \quad k(x,\bar{x})$$

Project on non linear subspaces



Kernel PCA

$$\widehat{K} = \frac{1}{n}\widehat{X}\widehat{X}^\top = \widehat{U}\widehat{\Lambda}^2\widehat{U}$$

$$v_1 = \frac{1}{n\widehat{\lambda}_1}\widehat{X}^\top\widehat{u}_1 \quad \Rightarrow \quad x^\top\widehat{v}_1 = \frac{1}{n\widehat{\lambda}_1}\sum_{i=1}^{n} x^\top x_i(\widehat{u}_1)_i$$

$$x^\top\overline{x} \quad \mapsto \quad k(x,\overline{x})$$

Project on non linear subspaces

$$\widehat{K}$$

Nyström PCA

$$v_1 = \operatorname*{argmax}_{\|w\|=1} w^\top \Sigma w \quad \Rightarrow \quad \tilde{v}_1 = \operatorname*{argmax}_{w=\overline{X}_M^\top c \;:\; \|w\|=1} w^\top \Sigma w$$

Nyström PCA

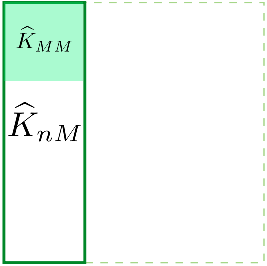
$$v_1 = \operatorname*{argmax}_{\|w\|=1} w^\top \Sigma w \quad \Rightarrow \quad \tilde{v}_1 = \operatorname*{argmax}_{w = \overline{X}_M^\top c \,:\, \|w\|=1} w^\top \Sigma w$$

$$\ldots$$

$$x^\top v_1 \approx x^\top \tilde{v}_1 = \sum_{i=1}^{M} x^\top \overline{x}_i \widehat{K}_M^{-1/2} (\tilde{u}_1)_i$$

with

$$\widehat{K}_M^{-1/2} \widehat{K}_{nM}^\top \widehat{K}_{nM} \widehat{K}_M^{-1/2} = \widetilde{U} \widetilde{\Lambda}^2 \widetilde{U}^\top.$$



Guarantees for Nyström PCA

Assume $(x_i)_{i=1}^n \sim \rho^n$, $\widehat{P}_\ell$ the Nyström PCA projection and [3]

$$L(\widehat{P}_\ell) = \mathbb{E}_x[\|x - P_\ell x\|^2].$$

Theorem (Sterge, Sriperumbudur, R., Rudi '20)

Assume $\|x\| \leqslant 1$*, and Nyström centers chosen uniformly at random. Let* $\Sigma = \mathbb{E}_x[xx^\top]$ *with*

$$\lambda_j^2(\Sigma) \sim j^{-\alpha}, \quad \alpha > 1$$

Then for $\ell = n^{\frac{\theta}{\alpha}}$, $\theta < 1$ *and* $M \leqslant n^\theta \log n$

$$\mathbb{E}[L(\widehat{C})] \lesssim n^{-\theta(1-\frac{1}{\alpha})}.$$

The above bound matches that of exact KPCA.

[3] Naturally extend to kernels $x \mapsto k(x, \cdot)$.

Wrapping up

Contribution

- Nyström projections allow computational savings with no accuracy loss.
- Further results: adaptive sampling,
  - leverage scores [Calandriello, Rudi, Carratino, R. '18],
  - DPP sampling [Dereziński Calandriello, Valko '19]
- Related results: random features [Rudi, R. '16].

We are thinking about:

- More Nyström kernel [`add your favorite`].
- Interpolation regimes $n \ll 2^d$.
- Combine Nyström and multiscale approaches [Chen, Avron, Sindawhani '16].

UniGe | MaLGa

PhD/Postdoc positions available!

erc

UniGe | MaLGa

Relevant stuff

Papers

Less is More: Nyström Computational Regularization
A. Rudi, R. Camoriano and L. Rosasco · NIPS15

FALKON: An Optimal Large Scale Kernel Method
A. Rudi, L. Carratino and L. Rosasco · NIPS17

Gaussian Process Optimization with Adaptive Sketching: Scalable and No Regret
D. Calandriello, L. Carratino, A. Lazaric, M. Valko and L. Rosasco · COLT19

Statistical and computational trade-offs in kernel k-means
D. Calandriello, L. Rosasco · NeurIPS18

Gain with no Pain: Efficient Kernel-PCA by Nyström Sampling
N. Sterge, B. Sriperumbur, L. Rosasco, A. Rudi · AISTATS20

Code

FALKON
G. Meanti,L. Carratino, L. Rosasco and A. Rudi · `http://lcsl.mit.edu`

BKB
D. Calandriello, L. Carratino, A. Lazaric, M. Valko and L. Rosasco · `http://lcsl.mit.edu`

More relevant stuff

Papers

Learning with SGD and Random Features
L. Carratino, A. Rudi and L. Rosasco · NeurIPS18

On Fast Leverage Score Sampling and Optimal Learning
A. Rudi, D. Calandriello, L. Carratino and L. Rosasco · NeurIPS18

Exact sampling of determinantal point processes with sublinear time preprocessing
M. Dereziński , D. Calandriello, M. Valko · NeurIPS19

Near-linear Time GP Optimization with Adaptive Batching and Resparsification
D. Calandriello, L. Carratino, A. Lazaric, M. Valko and L. Rosasco · Preprint 2020

Code

BLESS: leverage score sampling
A. Rudi, D. Calandriello, L. Carratino and L. Rosasco · `http://lcsl.mit.edu`

DPP sampling
M. Dereziński , D. Calandriello, M. Valko · `http://lcsl.mit.edu`