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 (MaLGa),
Alessandro Rudi (INRIA Paris), Bharath Sriperumbudur, Nick Sterge (UPenn),
Alessandro Lazaric (Facebook), Michal Valko (DeepMind)

Data

$\Longleftrightarrow$

Computations

The quest for provably efficient ML algorithms

Outline

Data size matters

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

In many modern applications, space is the real constraint.

Think n, d large!

Dimensionality reduction

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

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

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

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

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

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

UniGe | MaLGa

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

Nyström projections illustrated: least squares

From

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

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

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

that is least squares projected on a random subspace.

[Engl, Hanke, Neubauer '96]

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.

This matters for kernel methods

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

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

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

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

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

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

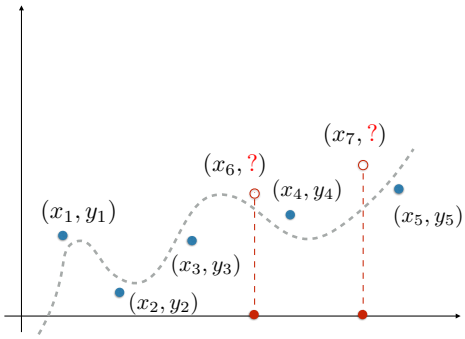
This matters for kernel methods

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

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

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

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

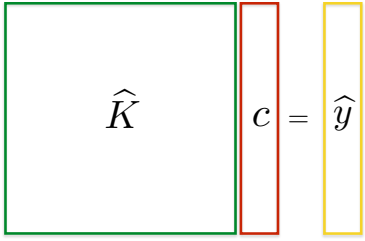


Nyström projections with kernels, aka column subsampling

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

$$\downarrow$$

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



Nyström projections with kernels, aka column subsampling

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

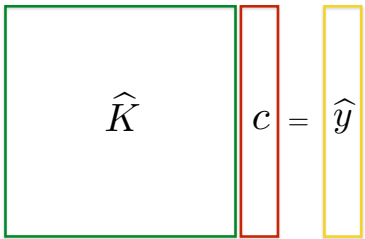
$$\downarrow$$

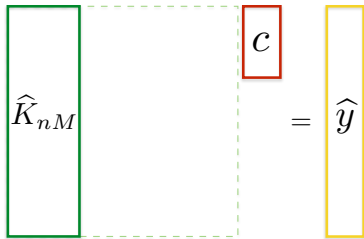
$$f(x) = \sum_{i=1}^{n} k(x, x_i) c_i \qquad \boxed{\widehat{K}c = \widehat{y}}$$

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

$$\downarrow$$

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





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

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

Why Nyström?

Nyström approximation for integral equations

For all x

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

From operators to matrices

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

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

[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

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

Dimensionality reduction improves efficiency, but what about learning accuracy?

Outline

Supervised statistical Learning

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

Solve

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

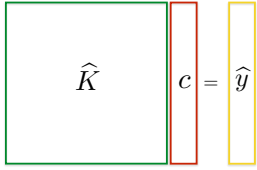
given $(x_i, y_i)_{i=1}^n \sim \rho^n$.

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)

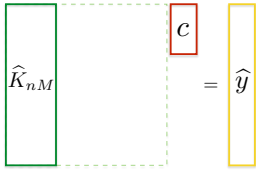
Let $\mathcal{H} = \text{span}\{k(x, \cdot) \mid x \in X\}$*, if* $\lambda = 1/\sqrt{n}$ *then*

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

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*

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

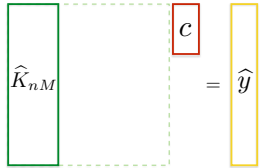
Iterative solvers and preconditoning

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

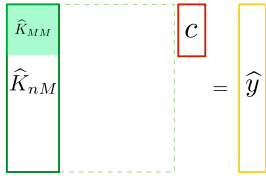
$$PP^\top = (\widehat{K}_{nM}^\top \widehat{K}_{nM} + \lambda n \widehat{K}_{MM})^{-1}$$



FALKON

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

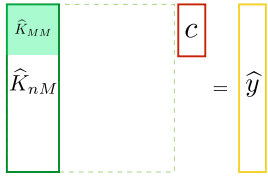
$$PP^{\top} = \left(\frac{n}{M}\widehat{K}_{MM}^{2} + \lambda n \widehat{K}_{MM}\right)^{-1}$$



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)

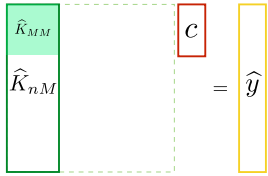
Let $(\widetilde{x}_i)_{i=1}^M \subseteq (x_i)_{i=1}^n$ *uniformly at random, then if* $\lambda = 1/\sqrt{n}$, $M \geqslant \sqrt{n}$ *and* $t \geqslant \log(n)$

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

FALKON

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

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



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* $t \geqslant \log(n)$

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

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

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: $\dagger$ = 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.

Outline

Bandit Optimization

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

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

Bandit Optimization

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

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

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

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

Bandit Optimization

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

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

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

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

(2) Select x_{t+1}

Bandit Optimization

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

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

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

(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) $\mathcal{A} = \{x_1, \dots, x_A\} \subset \mathbb{R}^d$, let $f : \mathcal{A} \to \mathbb{R}$ unknown, $(\eta_t)_t$ random, and

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

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

(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

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

Gaussian processes

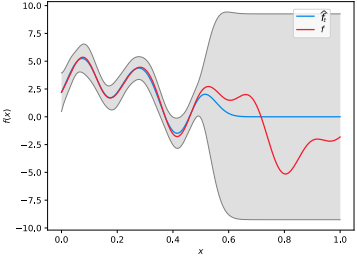
aka kernel ridge regression

$\widehat{K}_t \in \mathbb{R}^{t,t}$ **s.t.** $(\widehat{K})_{i,j} = k(x_i, x_j)$**,** $i, j = 1, \dots, t$
$\widehat{k}_t(x) = (k(x_1, x), \dots, k(x_t, x)) \in \mathbb{R}^t$

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

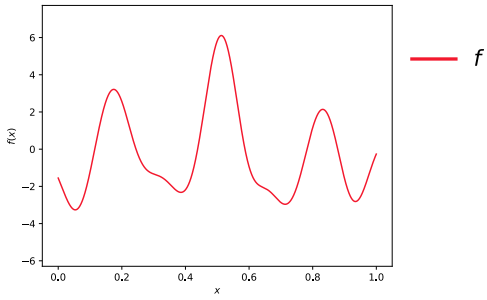
$$\widehat{f}_t(x) = \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{y}_t$$

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



GP-UCB

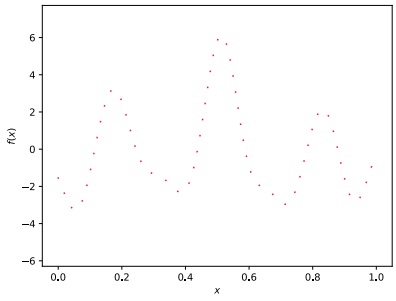
$f : \mathcal{A} \to \mathbb{R}$ unknown



(Srinivas, Krause, Kakade, Seeger '10)

GP-UCB

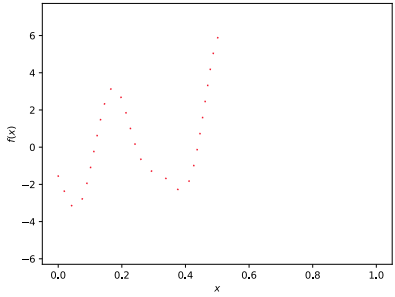
Arms $\mathcal{A} = \{x_i\}_{i=1}^{A}$



(Srinivas, Krause, Kakade, Seeger '10)

GP-UCB

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



— f

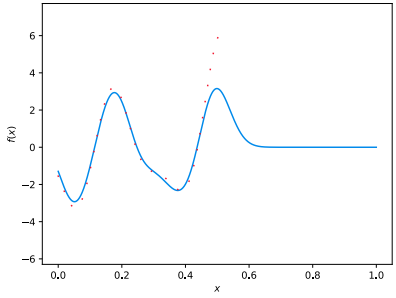
for $t = \{1, \ldots, T-1\}$ **do**

end

(Srinivas, Krause, Kakade, Seeger '10)

GP-UCB

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



$\widehat{f}_t$

f

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

end

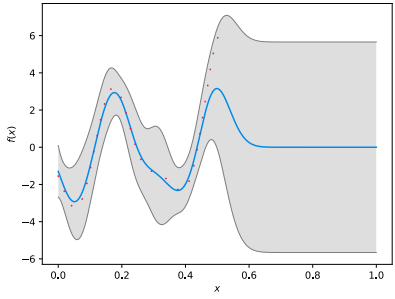
$$\widehat{f}_t(x) = \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{y}_t$$

(Srinivas, Krause, Kakade, Seeger '10)

UniGe | MaLGa

GP-UCB

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



$\widehat{f}_t$

f

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

end

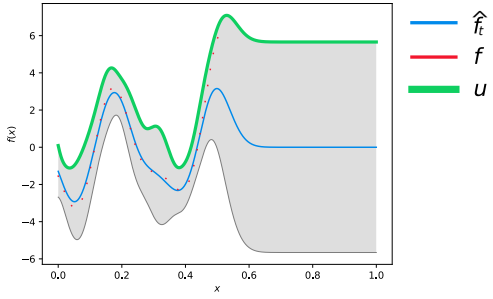
$$\widehat{f}_t(x) = \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{y}_t \qquad \sigma_t^2(x) = k(x,x) - \widehat{k}_t(x)^\top (\widehat{K}_t + \lambda I)^{-1} \widehat{k}_t(x)$$

(Srinivas, Krause, Kakade, Seeger '10)

UniGe | MaLGa

GP-UCB

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



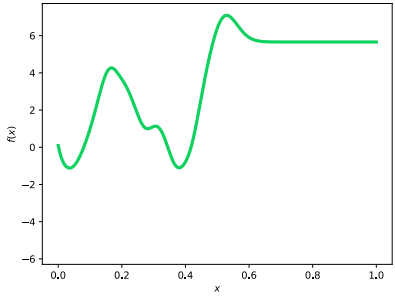
for $t = \{1, \ldots, T-1\}$ **do**
  for $i = \{1, \ldots, A\}$ **do**
    $u_t(x_i) = \widehat{f}_t(x_i) + \beta_t \sigma_t^2(x_i)$;
  end
end

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

(Srinivas, Krause, Kakade, Seeger '10)

GP-UCB

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



u

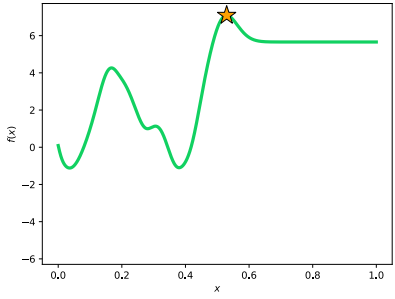
for $t = \{1, \ldots, T-1\}$ **do**
  for $i = \{1, \ldots, A\}$ **do**
    $u_t(x_i) = \widehat{f}_t(x_i) + \beta_t \sigma_t^2(x_i)$;
  end
end

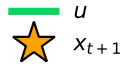
$$u_t(x) = \widehat{f}_t(x) + \beta_t \sigma_t(x)$$

(Srinivas, Krause, Kakade, Seeger '10)

GP-UCB

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



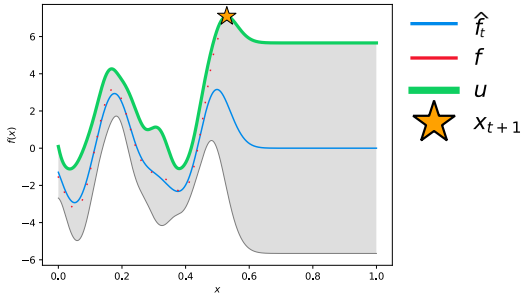


for $t = \{1, \dots, T-1\}$ **do**
  for $i = \{1, \dots, A\}$ **do**
    $u_t(x_i) = \widehat{f}_t(x_i) + \beta_t \sigma_t^2(x_i)$;
  end
  Select $x_{t+1} \leftarrow \text{argmax}_{x_i \in \mathcal{A}}\, u_t(x_i)$;
end

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

(Srinivas, Krause, Kakade, Seeger '10)

GP-UCB: Regret



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

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

For the proper β_t

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

UniGe | MaLGa

Nyström projection again

aka Sparse GP

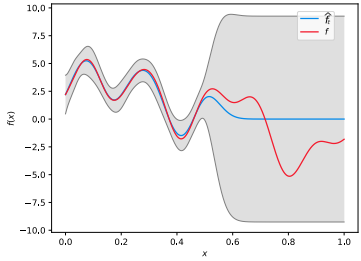
Equivalent formulation:

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 $\widetilde{k}_S(x) = (k(\bar{x}_1, x), \dots, k(\bar{x}_M, x))$

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

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



Nyström projection again

aka Sparse GP

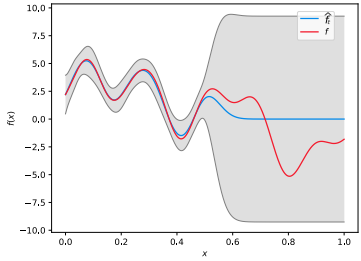
Equivalent formulation:

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 $\widetilde{k}_S(x) = (k(\bar{x}_1, x), \dots, k(\bar{x}_M, x))$

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

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



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 $\rightarrow S_t$ must change with t
- $\sigma_t^2(\cdot)$ captures informative arms $\rightarrow$ include x_i in S_t when $\sigma_t^2(x_i)$ is large

BKB: Regret

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

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

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

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 $S_{t+1} \sim \widetilde{p}_{t+1}$;
end

BKB: Regret

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

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

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

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 $S_{t+1} \sim \widetilde{p}_{t+1}$;
end

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

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

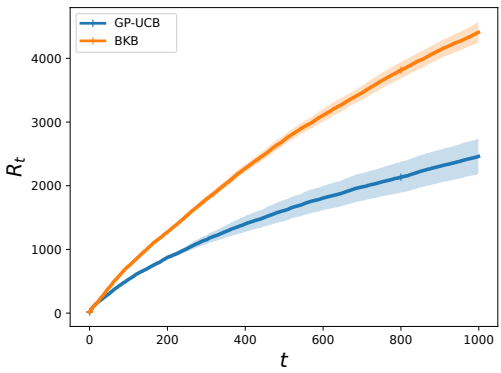
For the proper (and cheap to compute!) $\widetilde{\beta}_t$*, with* $|S_T| \leqslant d_{eff}$ *with* $d_{eff} \ll T$

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

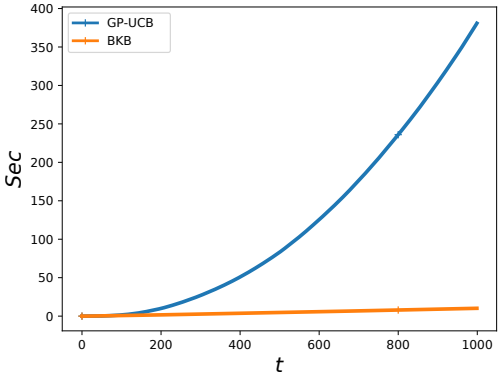
UniGe | MaLGa

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].

Outline

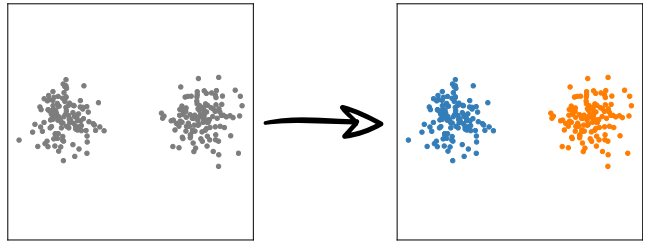
Nyström projection for unsupervised learning?

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

K-means

Partition n points into k clusters.

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



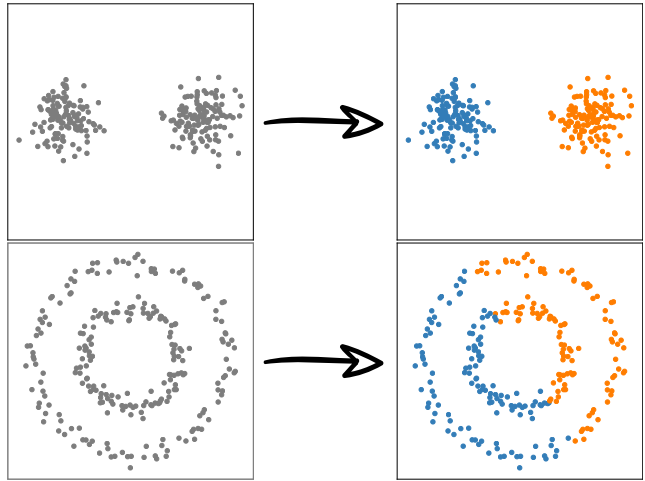
Only linear separations

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.

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

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

From K-means to kernel K-means

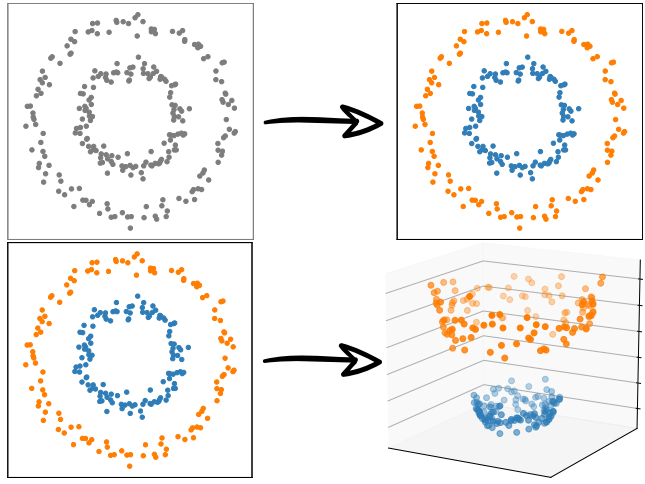
Partition n points into K clusters.

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

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

Kernel to rescue!

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



From K-means to kernel K-means

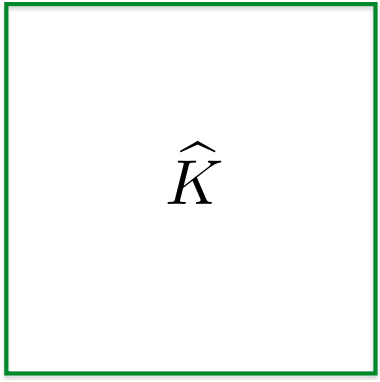
Partition n points into K clusters.

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

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

Kernel to rescue!

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



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

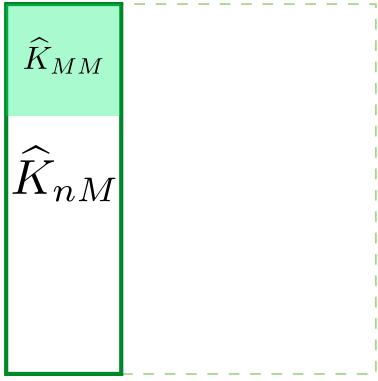
Partition n points into K clusters.

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

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

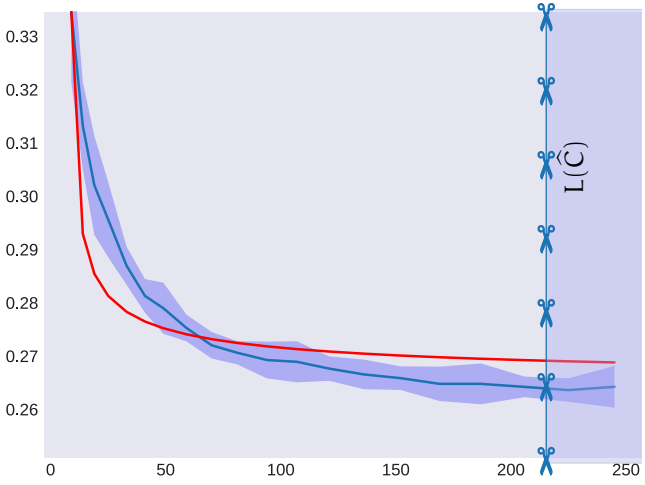
$$\mathbb{E}[L(\widehat{C}_K)] \lesssim \frac{K}{\sqrt{n}}.$$

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

UniGe | MaLGa

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

MNIST-60k: expected loss vs projection size M

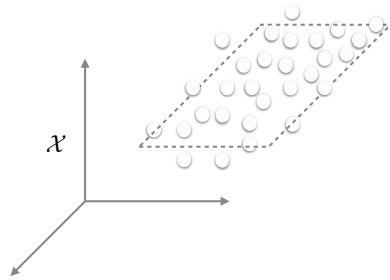


M

Nyström projections for unsupervised learning?

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

PCA



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

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

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

Project only on linear subspacess

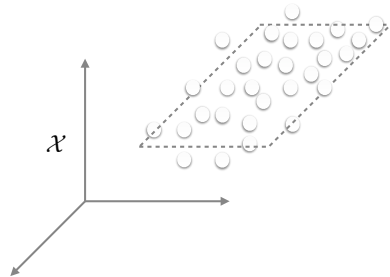
PCA

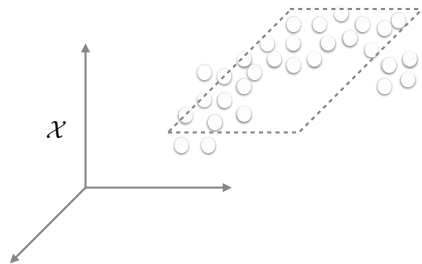
$$\widehat{\Sigma} = \frac{1}{n}\widehat{X}^\top \widehat{X} = \widehat{V}\widehat{\Lambda}^2\widehat{V}^\top$$

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

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

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

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

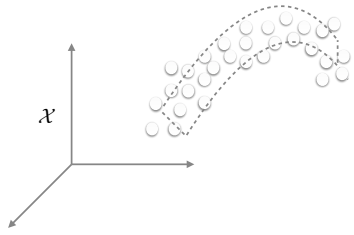
$$v_1 = \frac{1}{n\widehat{\lambda}_1}\widehat{X}^\top\widehat{u}_1 \quad \Rightarrow \quad x^\top\widehat{v}_1 = \frac{1}{n\widehat{\lambda}_1}\sum_{i=1}^{n} x^\top x_i(\widehat{u}_1)_i$$

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

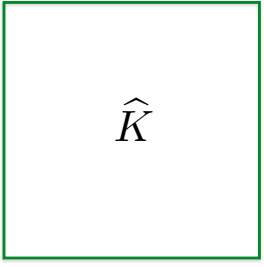
Kernel PCA

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

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

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

Project on non linear subspaces



Nyström PCA

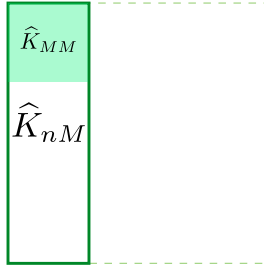
$$v_1 = \underset{\|w\|=1}{\operatorname{argmax}}\, w^\top \Sigma w \quad \Rightarrow \quad \tilde{v}_1 = \underset{w=\overline{X}_M^\top c \,:\, \|w\|=1}{\operatorname{argmax}} \quad w^\top \Sigma w$$

Nyström PCA

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

$$\dots$$

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



with

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

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

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`