Stochastic optimization in Hilbert spaces

Aymeric Dieuleveut

Outline

Learning vs Statistics

Outline

Tradeoffs of large scale learning

Algorithm complexity. ERM ?

Learning vs Statistics

Outline

Tradeoffs of large scale learning

Algorithm complexity. ERM ?

Stochastic optimization

Why is SGD so useful in learning ?

Learning vs Statistics

Outline

Tradeoffs of large scale learning

Algorithm complexity. ERM ?

Learning vs Statistics

Stochastic optimization

Why is SGD so useful in learning ?

A simple case

Least mean squares, finite dimension

Outline

Tradeoffs of large scale learning

Algorithm complexity. ERM ?

Stochastic optimization

Why is SGD so useful in learning ?

Learning vs Statistics

Higher dimension ?

RKHS, non parametric learning

A simple case

Least mean squares, finite dimension

Outline

Tradeoffs of large scale learning

Algorithm complexity. ERM ?

Lower complexity ?

Column sampling, feature selection

Learning vs Statistics

Stochastic optimization

Why is SGD so useful in learning ?

Higher dimension ?

RKHS, non parametric learning

A simple case

Least mean squares, finite dimension

Statistics vs Machine Learning

1. taken from www.quora.com/What-is-the-difference-between-statistics-and-machine-learning

Statistics vs Machine Learning

Statistics	Machine Learning
Estimation	Learning
Classifier	Hypothesis
Data point	Example/Instance
Regression	Supervised Learning
Classification	Supervised Learning
Covariate	Feature
Response	Label [1]

1. taken from www.quora.com/What-is-the-difference-between-statistics-and-machine-learning

Statistics vs Machine Learning

Statistics	Machine Learning
Estimation	Learning
Classifier	Hypothesis
Data point	Example/Instance
Regression	Supervised Learning
Classification	Supervised Learning
Covariate	Feature
Response	Label [1]

Essentially AI vs math guys doing same kind of stuff. However main differences :

1. taken from www.quora.com/What-is-the-difference-between-statistics-and-machine-learning

Statistics vs Machine Learning

Statistics	Machine Learning
Estimation	Learning
Classifier	Hypothesis
Data point	Example/Instance
Regression	Supervised Learning
Classification	Supervised Learning
Covariate	Feature
Response	Label [1]

Essentially AI vs math guys doing same kind of stuff. However main differences :

- Statisticians are more interested in the model and drawing conclusions about it.

1. taken from www.quora.com/What-is-the-difference-between-statistics-and-machine-learning

Statistics vs Machine Learning

Statistics	Machine Learning
Estimation	Learning
Classifier	Hypothesis
Data point	Example/Instance
Regression	Supervised Learning
Classification	Supervised Learning
Covariate	Feature
Response	Label [1]

Essentially AI vs math guys doing same kind of stuff. However main differences :

- Statisticians are more interested in the model and drawing conclusions about it.
- ML are more interested about prediction with a concern on algorithms for high dim. data.

1. taken from www.quora.com/What-is-the-difference-between-statistics-and-machine-learning

Framework

We consider the classical risk minimization problem. Given :

- a space of input output pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, with probability distribution $P(x, y)$.
- a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, a class of function $\mathcal{F}$.
- the risk of a function $f : \mathcal{X} \to \mathcal{Y}$ is $R(f) := \mathbb{E}_P[\ell(f(x), y)]$.

Our aim is

$$\min_{f \in \mathcal{F}} R(f)$$

Framework

We consider the classical risk minimization problem. Given :

- a space of input output pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, with probability distribution $P(x, y)$.
- a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, a class of function $\mathcal{F}$.
- the risk of a function $f : \mathcal{X} \to \mathcal{Y}$ is $R(f) := \mathbb{E}_P[\ell(f(x), y)]$.

Our aim is

$$\min_{f \in \mathcal{F}} R(f)$$

- R is unknown.

Framework

We consider the classical risk minimization problem. Given :

- a space of input output pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, with probability distribution $P(x, y)$.
- a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, a class of function $\mathcal{F}$.
- the risk of a function $f : \mathcal{X} \to \mathcal{Y}$ is $R(f) := \mathbb{E}_P[\ell(f(x), y)]$.

Our aim is

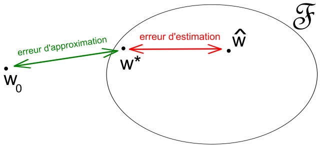
$$\min_{f \in \mathcal{F}} R(f)$$

- R is unknown.
- given a sequence of i.i.d. data points distributed $(x_i, y_i)_{i=1..n} \sim P^{\otimes n}$, we can define the empirical risk

$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

The bias-variance tradeoffs

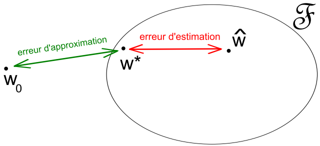
a.k.a. estimation approximation error.



The bias-variance tradeoffs

a.k.a. estimation approximation error.
There are many ways of seeing it :

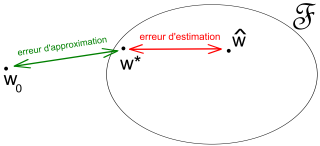
- constraint case
- penalized case
- other regularization



The bias-variance tradeoffs

a.k.a. estimation approximation error.
There are many ways of seeing it :

- constraint case
- penalized case
- other regularization



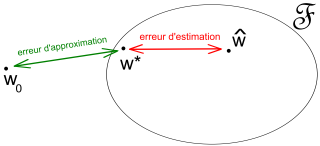
Thus compromise : $\varepsilon_{app} + \varepsilon_{est}$.

	ε_{app}	ε_{est}
$\mathcal{F} \nearrow$		

The bias-variance tradeoffs

a.k.a. estimation approximation error.
There are many ways of seeing it :

- constraint case
- penalized case
- other regularization



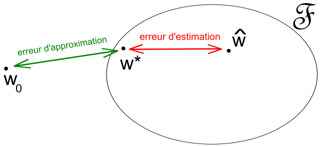
Thus compromise : $\varepsilon_{app} + \varepsilon_{est}$.

	ε_{app}	ε_{est}
$\mathcal{F} \nearrow$	$\searrow$	

The bias-variance tradeoffs

a.k.a. estimation approximation error.
There are many ways of seeing it :

- constraint case
- penalized case
- other regularization



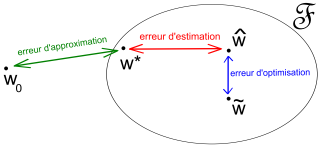
Thus compromise : $\varepsilon_{app} + \varepsilon_{est}$.

	ε_{app}	ε_{est}
$\mathcal{F} \nearrow$	$\searrow$	$\nearrow$

This is the classical setting.

Adding an optimization term

When we face large datasets, it may be uneasy and useless to optimize with high accuracy the estimator. We then question the choice of an algorithm from a fixed budget time point of view. [2]

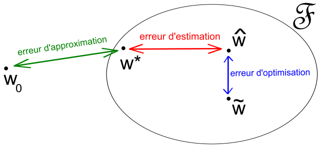


2. Ref :[Shalev-Schwartz and Srebro, 2008, Shalev-Schwartz and K., 2011, Bottou and Bousquet, 2008]

Adding an optimization term

When we face large datasets, it may be uneasy and useless to optimize with high accuracy the estimator. We then question the choice of an algorithm from a fixed budget time point of view. [2]
It questions the following points :

- up to which precision is it necessary to optimize ?

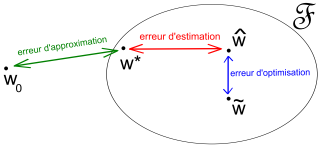


2. Ref :[Shalev-Schwartz and Srebro, 2008, Shalev-Schwartz and K., 2011, Bottou and Bousquet, 2008]

Adding an optimization term

When we face large datasets, it may be uneasy and useless to optimize with high accuracy the estimator. We then question the choice of an algorithm from a fixed budget time point of view. [2]
It questions the following points :

- up to which precision is it necessary to optimize ?
- which is the limiting factor ? (time, data points)



2. Ref :[Shalev-Schwartz and Srebro, 2008, Shalev-Schwartz and K., 2011, Bottou and Bousquet, 2008]

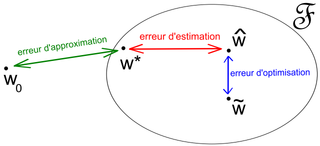
Adding an optimization term

When we face large datasets, it may be uneasy and useless to optimize with high accuracy the estimator. We then question the choice of an algorithm from a fixed budget time point of view. [2]
It questions the following points :

- up to which precision is it necessary to optimize ?
- which is the limiting factor ? (time, data points)

A problem is said to be large scale when time is limiting. For large scale problem :

- which algo ?



2. Ref :[Shalev-Schwartz and Srebro, 2008, Shalev-Schwartz and K., 2011, Bottou and Bousquet, 2008]

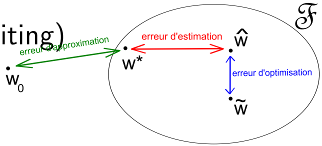
Adding an optimization term

When we face large datasets, it may be uneasy and useless to optimize with high accuracy the estimator. We then question the choice of an algorithm from a fixed budget time point of view. [2]
It questions the following points :

- up to which precision is it necessary to optimize ?
- which is the limiting factor ? (time, data points)

A problem is said to be large scale when time is limiting. For large scale problem :

- which algo ?
- more data less work ? (if time is limiting)



2. Ref :[Shalev-Schwartz and Srebro, 2008, Shalev-Schwartz and K., 2011, Bottou and Bousquet, 2008]

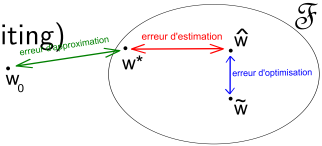
Adding an optimization term

When we face large datasets, it may be uneasy and useless to optimize with high accuracy the estimator. We then question the choice of an algorithm from a fixed budget time point of view. [2]
It questions the following points :

- up to which precision is it necessary to optimize ?
- which is the limiting factor ? (time, data points)

A problem is said to be large scale when time is limiting. For large scale problem :

- which algo ?
- more data less work ? (if time is limiting)



2. Ref :[Shalev-Schwartz and Srebro, 2008, Shalev-Schwartz and K., 2011, Bottou and Bousquet, 2008]

Tradeoffs - Large scale learning

	$\mathcal{F} \nearrow$	$n \nearrow$	$\varepsilon \nearrow$

Tradeoffs - Large scale learning

	$\mathcal{F} \nearrow$	$n \nearrow$	$\varepsilon \nearrow$
ε_{app}	$\searrow$		

Tradeoffs - Large scale learning

	$\mathcal{F} \nearrow$	$n \nearrow$	$\varepsilon \nearrow$
$\mathcal{E}_{app}$	$\searrow$		
$\mathcal{E}_{est}$	$\nearrow$	$\searrow$	

Tradeoffs - Large scale learning

	$\mathcal{F} \nearrow$	$n \nearrow$	$\varepsilon \nearrow$
$\mathcal{E}_{app}$	$\searrow$		
$\mathcal{E}_{est}$	$\nearrow$	$\searrow$	
$\mathcal{E}_{opt}$			$\nearrow$

Tradeoffs - Large scale learning

	$\mathcal{F} \nearrow$	$n \nearrow$	$\varepsilon \nearrow$
$\mathcal{E}_{app}$	$\searrow$		
$\mathcal{E}_{est}$	$\nearrow$	$\searrow$	
$\mathcal{E}_{opt}$			$\nearrow$
T	$\nearrow$	$\nearrow$	$\searrow$

Different algorithms

To minimize ERM, a bunch of algorithms may be considered :

- Gradient descent
- Second order gradient descent
- Stochastic gradient descent
- Fast stochastic algorithm (requiring high memory storage)

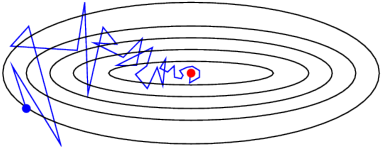
Different algorithms

To minimize ERM, a bunch of algorithms may be considered :

- Gradient descent
- Second order gradient descent
- Stochastic gradient descent
- Fast stochastic algorithm (requiring high memory storage)

Let's compare first order methods : SGD and GD.

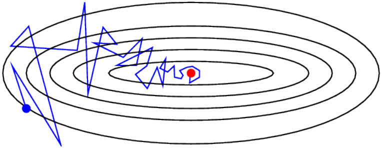
Stochastic gradient algorithms :



Aim : $\min_f R(f)$

- we only access to unbiased estimates of $R(f)$ and $\nabla R(f)$.

Stochastic gradient algorithms :

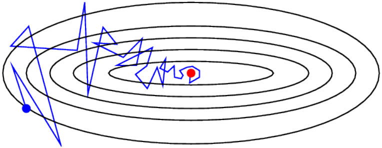


Aim : $\min_f R(f)$

- we only access to unbiased estimates of $R(f)$ and $\nabla R(f)$.

1. Start at some f_0.

Stochastic gradient algorithms :

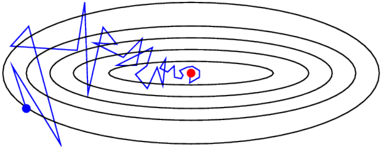


Aim : $\min_f R(f)$

- we only access to unbiased estimates of $R(f)$ and $\nabla R(f)$.

1. Start at some f_0.
2. Iterate :
   - Get unbiased gradient estimate g_k, s.t. $E[g_k] = \nabla R(f_k)$.
   - $f_{k+1} \leftarrow f_k - \gamma_k g_k$.

Stochastic gradient algorithms :



Aim : $\min_f R(f)$

- we only access to unbiased estimates of $R(f)$ and $\nabla R(f)$.

1. Start at some f_0.
2. Iterate :
   - Get unbiased gradient estimate g_k, s.t. $E[g_k] = \nabla R(f_k)$.
   - $f_{k+1} \leftarrow f_k - \gamma_k g_k$.
3. Output f_m or $\bar{f}_m := \frac{1}{m} \sum_{k=1}^{m} f_k$ (averaged SGD).

Gradient descent : same but with "true" gradient.

ERM

SGD in ERM

$\min_{f \in \mathcal{F}} R_n(f)$

ERM

SGD in ERM
$\min_{f\in\mathcal{F}} R_n(f)$
Pick any (x_i, y_i) from empirical sample
$g_k = \nabla_f \ell(f_k, (x_i, y_i))$.
$f_{k+1} \leftarrow (f_k - \gamma_k g_k)$
Output $\bar{f}_m$
$R_n(\bar{f}_m) - R_n(f_n^*) \leqslant O\left(1/\sqrt{m}\right)$
$sup_{f\in\mathcal{F}}\lvert R - R_n\rvert(f) \leqslant O(1/\sqrt{n})$
Cost of one iteration $O(d)$.

GD in ERM
$\min_{f\in\mathcal{F}} R_n(f)$

ERM

SGD in ERM	GD in ERM
$\min_{f\in\mathcal{F}} R_n(f)$	$\min_{f\in\mathcal{F}} R_n(f)$
Pick any (x_i, y_i) from empirical sample $g_k = \nabla_f \ell(f_k,(x_i,y_i))$. $f_{k+1} \leftarrow (f_k - \gamma_k g_k)$	$g_k = \nabla_f \sum_{i=1}^n \ell(f_k,(x_i,y_i)) = \nabla_f R(f_k)$ $f_{k+1} \leftarrow (f_k - \gamma_k g_k)$
Output $\bar{f}_m$ $R_n(\bar{f}_m) - R_n(f_n^*) \leqslant O\left(1/\sqrt{m}\right)$ $sup_{f\in\mathcal{F}}\lvert R - R_n\rvert(f) \leqslant O(1/\sqrt{n})$	Output f_m $R_n(f_m) - R_n(f_n^*) \leqslant O\left((1-\kappa)^m\right)$ $sup_{f\in\mathcal{F}}\lvert R - R_n\rvert(f) \leqslant O(1/\sqrt{n})$
Cost of one iteration $O(d)$.	Cost of one iteration $O(nd)$.

$$R(\bar{f}_m) - R(f^*) \leqslant O\left(1/\sqrt{m}\right) + O(1/\sqrt{n})$$

With step-size γ_k proportional to $\frac{1}{\sqrt{k}}$.

Conclusion

In the large scale setting, it is beneficial to use SGD !

Conclusion

In the large scale setting, it is beneficial to use SGD !

Does more data help ?

- With global estimation error fixed, it seems $T \simeq \frac{1}{R(f_m)-R(f_*)-\frac{1}{\sqrt{n}}}$ is decreasing with n.

Conclusion

In the large scale setting, it is beneficial to use SGD !

Does more data help ?

- With global estimation error fixed, it seems $T \simeq \frac{1}{R(f_m) - R(f_*) - \frac{1}{\sqrt{n}}}$ is decreasing with n.

Upper bounding $R_n - R$ uniformly is dangerous. Indeed, we have to also compare to one pass SGD, which minimizes the true risk R.

Expectation minimization

Stochastic gradient descent may be used to minimize $R(f)$:

SGD in ERM

$\min_{f\in\mathcal{F}} R_n(f)$

Expectation minimization

Stochastic gradient descent may be used to minimize $R(f)$:

SGD in ERM	SGD one pass
$\min_{f\in\mathcal{F}} R_n(f)$	$\min_{f\in\mathcal{F}} R(f)$
Pick any (x_i, y_i) from empirical sample	
$g_k = \nabla_f \ell(f_k, (x_i, y_i))$.	
$f_{k+1} \leftarrow (f_k - \gamma_k g_k)$	
Output $\bar{f}_m$	
$R_n(\bar{f}_m) - R_n(f_n^*) \leqslant O\left(1/\sqrt{m}\right)$	
$sup_{f\in\mathcal{F}} \lvert R - R_n\rvert(f) \leqslant O(1/\sqrt{n})$	
Cost of one iteration $O(d)$.	

Expectation minimization

Stochastic gradient descent may be used to minimize $R(f)$:

SGD in ERM
$\min_{f\in\mathcal{F}} R_n(f)$
Pick any (x_i, y_i) from empirical sample
$g_k = \nabla_f \ell(f_k, (x_i, y_i))$.
$f_{k+1} \leftarrow (f_k - \gamma_k g_k)$
Output $\bar{f}_m$
$R_n(\bar{f}_m) - R_n(f_n^*) \leqslant O\left(1/\sqrt{m}\right)$
$sup_{f\in\mathcal{F}}\lvert R - R_n\rvert(f) \leqslant O(1/\sqrt{n})$
Cost of one iteration $O(d)$.

SGD one pass
$\min_{f\in\mathcal{F}} R(f)$
Pick an **independent** (x, y)
$g_k = \nabla_f \ell(f_k, (x, y))$.
$f_{k+1} \leftarrow (f_k - \gamma_k g_k)$
Output $\bar{f}_k, k \leqslant n$
$R(\bar{f}_k) - R(f^*) \leqslant O\left(1/\sqrt{k}\right)$
Cost of one iteration $O(d)$.

Expectation minimization

Stochastic gradient descent may be used to minimize $R(f)$:

SGD in ERM
$\min_{f\in\mathcal{F}} R_n(f)$
Pick any (x_i, y_i) from empirical sample
$g_k = \nabla_f \ell(f_k, (x_i, y_i))$.
$f_{k+1} \leftarrow (f_k - \gamma_k g_k)$
Output $\bar{f}_m$
$R_n(\bar{f}_m) - R_n(f_n^*) \leqslant O\left(1/\sqrt{m}\right)$
$sup_{f\in\mathcal{F}} \lvert R - R_n \rvert (f) \leqslant O(1/\sqrt{n})$
Cost of one iteration $O(d)$.

SGD one pass
$\min_{f\in\mathcal{F}} R(f)$
Pick an **independent** (x, y)
$g_k = \nabla_f \ell(f_k, (x, y))$.
$f_{k+1} \leftarrow (f_k - \gamma_k g_k)$
Output $\bar{f}_k, k \leqslant n$
$R(\bar{f}_k) - R(f^*) \leqslant O\left(1/\sqrt{k}\right)$
Cost of one iteration $O(d)$.

SGD with one pass (**early stopping as a regularization**) achieves a nearly optimal bias variance tradeoff with low complexity.

Rate of convergence

We are interested in prediction.

- Strongly convex objective : $\frac{1}{\mu n}$.
- Non strongly : $\frac{1}{\sqrt{n}}$.

LMS [Bach and Moulines, 2013]

We now consider the simple case where $\mathcal{X} = \mathbb{R}^d$, and the loss ℓ is quadratic. We are interested in linear predictors :

$$\min_{\theta \in R^d} \mathbb{E}_P[(\theta^T x - y)^2].$$

If we assume that the data points are generated according to

$$y_i = \theta_*^T x_i + \varepsilon_i.$$

We consider stochastic gradient algorithm :

$$\begin{aligned} \theta_0 &= 0 \\ \theta_{n+1} &= \theta_n - \gamma_n(\langle x_n, \theta_n \rangle x_n - y_n x_n) \end{aligned}$$

This system may be rewritten :

$$\theta_{n+1} - \theta_* = (I - \gamma x_n x_n^T)(\theta_n - \theta_*) - \gamma_n \xi_n. \tag{1}$$

Rate of convergence, back again !

We are interested in prediction.

- Strongly convex objective : $\frac{1}{\mu n}$.
- Non strongly : $\frac{1}{\sqrt{n}}$.

We define $H = \mathbb{E}[xx^T]$.

Rate of convergence, back again !

We are interested in prediction.

- Strongly convex objective : $\frac{1}{\mu n}$.
- Non strongly : $\frac{1}{\sqrt{n}}$.

We define $H = \mathbb{E}[xx^T]$. We have $\mu = \min \mathrm{Sp}(H)$.

Rate of convergence, back again !

We are interested in prediction.

- Strongly convex objective : $\frac{1}{\mu n}$.
- Non strongly : $\frac{1}{\sqrt{n}}$.

We define $H = \mathbb{E}[xx^T]$. We have $\mu = \min \mathrm{Sp}(H)$.

For least min squares, statistical rate with ordinary LMS estimator is

$$\frac{\sigma^2 d}{n}$$

Rate of convergence, back again !

We are interested in prediction.

- Strongly convex objective : $\frac{1}{\mu n}$.
- Non strongly : $\frac{1}{\sqrt{n}}$.

We define $H = \mathbb{E}[xx^T]$. We have $\mu = \min \mathrm{Sp}(H)$.

For least min squares, statistical rate with ordinary LMS estimator is

$$\frac{\sigma^2 d}{n}$$

there is still a gap to be bridged !

A few assumptions

We define $H = \mathbb{E}[xx^T]$, and $C = \mathbb{E}[\xi\xi^T]$.

A few assumptions

We define $H = \mathbb{E}[xx^T]$, and $C = \mathbb{E}[\xi\xi^T]$.

Bounded noise variance : we assume $C \leqslant \sigma^2 H$.

A few assumptions

We define $H = \mathbb{E}[xx^T]$, and $C = \mathbb{E}[\xi\xi^T]$.

Bounded noise variance : we assume $C \leqslant \sigma^2 H$.

Covariance operator :

- no assumption on minimal eigenvalue,
- $\mathbb{E}[\|x\|^2] \leqslant R^2$.

Result

Theorem

$$\mathbb{E}[R(\bar{\theta}_n) - R(\theta_*)] \leqslant \frac{4}{n}(\sigma^2 d + R^2 \|\theta_0 - \theta^*\|^2)$$

- optimal statistical rate
- $1/n$ without strong convexity.

Outline

What if $d >> n$?

Outline

What if $d >> n$?

Carry analyse in a Hilbert space

using reproducing kernel Hilbert spaces

Outline

What if $d >> n$?

Non parametric regression in RKHS

An interesting problem itself

Carry analyse in a Hilbert space

using reproducing kernel Hilbert spaces

Outline

What if $d >> n$?

Non parametric regression in RKHS

An interesting problem itself

Carry analyse in a Hilbert space

using reproducing kernel Hilbert spaces

Behaviour in FD

Adaptativity, tradeoffs.

Optimal statistical rates in RKHS

Choice of γ

Reproducing kernel Hilbert space [Dieuleveut and Bach, 2014]

We denote $\mathcal{H}_K$ a Hilbert space of function. $\mathcal{H}_K \subset \mathbb{R}^{\mathcal{X}}$.
Which is characterized by the kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$:

- for any x, $K_x : \mathcal{X} \to \mathbb{R}$ defined by $K_x(x') = K(x, x')$ is in $\mathcal{H}_K$.

Reproducing kernel Hilbert space [Dieuleveut and Bach, 2014]

We denote $\mathcal{H}_K$ a Hilbert space of function. $\mathcal{H}_K \subset \mathbb{R}^{\mathcal{X}}$.
Which is characterized by the kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$:

- for any x, $K_x : \mathcal{X} \to \mathbb{R}$ defined by $K_x(x') = K(x, x')$ is in $\mathcal{H}_K$.
- reproducing property : for all $g \in \mathcal{H}_K$ and $x \in \mathcal{X}$, $g(x) = \langle g, K_x \rangle_K$.

Reproducing kernel Hilbert space [Dieuleveut and Bach, 2014]

We denote $\mathcal{H}_K$ a Hilbert space of function. $\mathcal{H}_K \subset \mathbb{R}^{\mathcal{X}}$.
Which is characterized by the kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$:

- for any x, $K_x : \mathcal{X} \to \mathbb{R}$ defined by $K_x(x') = K(x, x')$ is in $\mathcal{H}_K$.
- reproducing property : for all $g \in \mathcal{H}_K$ and $x \in \mathcal{X}$, $g(x) = \langle g, K_x \rangle_K$.

Two usages :

- α) **A hypothesis space for regression.**
- β) **Mapping data points in a linear space.**

α) A hypothesis space for regression.

Classical regression setting :

$$(X_i, Y_i) \sim \rho \quad \text{i.i.d.}$$

$$(X_i, Y_i) \in (\mathcal{X} \times \mathbb{R})$$

Goal : Minimizing prediction error

$$\min_{g \in \mathcal{L}^2} \mathbb{E}[(g(X) - Y)^2].$$

α) A hypothesis space for regression.

Classical regression setting :

$$(X_i, Y_i) \sim \rho \quad \text{i.i.d.}$$
$$(X_i, Y_i) \in (\mathcal{X} \times \mathbb{R})$$

Goal : Minimizing prediction error

$$\min_{g \in \mathcal{L}^2} \mathbb{E}[(g(X) - Y)^2].$$

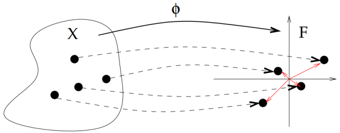
Looking for an estimator $\hat{g}_n$ of $g_\rho(X) = \mathbb{E}[Y|X]$, $g_\rho \in \mathcal{L}^2_{\rho_\mathcal{X}}$. with

$$\mathcal{L}^2_{\rho_\mathcal{X}} = \left\{ f : \mathcal{X} \to \mathbb{R} / \int f^2(t) d\rho_\mathcal{X}(t) < \infty \right\}.$$

β) Mapping data points in a linear space.

Linear regression on data maped into some RKHS.

$$\arg\min_{\theta \in \mathcal{H}} ||Y - X\theta||^2.$$



2 approaches of regression problem :

Link : In general

$$\mathcal{H}_K \subset \mathcal{L}^2_{\rho_{\mathcal{X}}}$$

2 approaches of regression problem :

Link : In general

$$\mathcal{H}_K \subset \mathcal{L}^2_{\rho_\mathcal{X}}$$

And

$$\mathrm{compl}_{||.||_{\mathcal{L}^2_{\rho_\mathcal{X}}}}(\mathrm{RKHS}) = \mathcal{L}^2_{\rho_\mathcal{X}}$$

in some cases. We then look for **an estimator of the regression function in the RKHS.**

2 approaches of regression problem :

Link : In general

$$\mathcal{H}_K \subset \mathcal{L}^2_{\rho_\mathcal{X}}$$

And

$$\mathrm{compl}_{||.||_{\mathcal{L}^2_{\rho_\mathcal{X}}}}(\mathrm{RKHS}) = \mathcal{L}^2_{\rho_\mathcal{X}}$$

in some cases. We then look for **an estimator of the regression function in the RKHS.**

General regression problem
$g_\rho \in \mathcal{L}^2$

2 approaches of regression problem :

Link : In general

$$\mathcal{H}_K \subset \mathcal{L}^2_{\rho_\mathcal{X}}$$

And

$$\text{compl}_{||.||_{\mathcal{L}^2_{\rho_\mathcal{X}}}}(\text{RKHS}) = \mathcal{L}^2_{\rho_\mathcal{X}}$$

in some cases. We then look for **an estimator of the regression function in the RKHS.**

General regression problem $g_\rho \in \mathcal{L}^2$

Linear regression problem in RKHS

2 approaches of regression problem :

Link : In general

$$\mathcal{H}_K \subset \mathcal{L}^2_{\rho_\mathcal{X}}$$

And

$$\text{compl}_{||.||_{\mathcal{L}^2_{\rho_\mathcal{X}}}}(\text{RKHS}) = \mathcal{L}^2_{\rho_\mathcal{X}}$$

in some cases. We then look for **an estimator of the regression function in the RKHS.**

General regression problem $g_\rho \in \mathcal{L}^2$

Linear regression problem in RKHS

looking for an estimator for the first problem using natural algorithms for the second one

Outline

What if $d >> n$?

Non parametric regression in RKHS

An interesting problem itself

Carry analyse in a Hilbert space

using reproducing kernel Hilbert spaces

SGD algorithm in the RKHS

$$\begin{aligned} g_0 &\in \mathcal{H}_K \quad \text{(we often consider } g_0 = 0\text{)}, \\ g_n &= \sum_{i=1}^{n} a_i K_{x_i}, \end{aligned} \tag{2}$$

$(a_n)_n$ such that $a_n := -\gamma_n(g_{n-1}(x_n) - y_n) = -\gamma_n \left(\sum_{i=1}^{n-1} a_i K(x_n, x_i) - y_i \right)$.

$$\begin{aligned} g_n &= g_{n-1} - \gamma_n \left(g_{n-1}(x_n) - y_n\right) K_{x_n} \\ &= \sum_{i=1}^{n} a_i K_{x_i} \quad \text{with } a_n \text{ defined as above.} \end{aligned}$$

$(g_{n-1}(x_n) - y_n) K_{x_n}$ unbiased estimate of $\mathrm{grad}\,\mathbb{E}[(\langle K_x, g_{n-1} \rangle - y)^2]$.

SGD algorithm in the RKHS takes very simple form

Assumptions

Two important points characterize the difficulty of the problem :

- The regularity of the objective function
- The spectrum of the covariance operator

Covariance operator

We have $\Sigma = \mathbb{E}\left[K_x \otimes K_x\right]$. Where $K_x \otimes K_x : g \mapsto \langle K_x, g \rangle K_x = g(x) K_x$

Covariance operator is a self adjoint operator which contains information on the distribution of K_x

Covariance operator

We have $\Sigma = \mathbb{E}\left[K_x \otimes K_x\right]$. Where $K_x \otimes K_x : g \mapsto \langle K_x, g\rangle K_x = g(x) K_x$

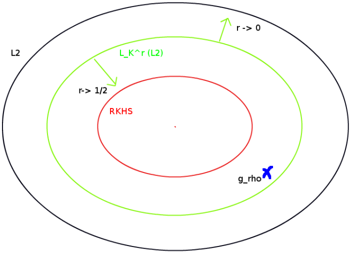
Covariance operator is a self adjoint operator which contains information on the distribution of K_x

Assumption :

- $\mathrm{tr}(\Sigma^\alpha) < \infty$, for $\alpha \in [0;1]$.
- on g_ρ : $g_\rho \in \Sigma^r(\mathcal{L}^2_{\rho(X)})$ with $r \geq 0$.

Interpretation

- Eigenvalues decrease
- Ellipsoid class of function. (we do not assume $g_\rho \in \mathcal{H}_K$)



Result :

Theorem

Under a few hidden assumptions :

$$\mathbb{E}\left[R\left(\bar{g}_n\right) - R(g_\rho)\right] \leqslant O\left(\frac{\sigma^2 \operatorname{tr}(\Sigma^\alpha)\gamma^\alpha}{n^{1-\alpha}}\right) + O\left(\frac{||\Sigma^{-r} g_\rho||_2}{(n\gamma)^{2(r\wedge 1)}}\right)$$

Result :

Theorem

Under a few hidden assumptions :

$$\mathbb{E}\left[R\left(\bar{g}_n\right) - R(g_\rho)\right] \leqslant O\left(\frac{\sigma^2 \operatorname{tr}(\Sigma^\alpha)\gamma^\alpha}{n^{1-\alpha}}\right) + O\left(\frac{||\Sigma^{-r} g_\rho||_2}{(n\gamma)^{2(r\wedge 1)}}\right)$$

- Bias Variance decomposition

Result :

Theorem

Under a few hidden assumptions :

$$\mathbb{E}\left[R\left(\bar{g}_n\right) - R(g_\rho)\right] \leqslant O\left(\frac{\sigma^2 \operatorname{tr}(\Sigma^\alpha)\gamma^\alpha}{n^{1-\alpha}}\right) + O\left(\frac{||\Sigma^{-r} g_\rho||_2}{(n\gamma)^{2(r\wedge 1)}}\right)$$

- Bias Variance decomposition
- O is a known constant (4 or 8)
- Finite horizon result here but extends to online setting.
- Saturation

Corollary

Corollary

Assume **A1-8** *:*
If $\frac{1-\alpha}{2} < r < \frac{2-\alpha}{2}$, *with* $\gamma = n^{-\frac{2r+\alpha-1}{2r+\alpha}}$ *we get the optimal rate :*

$$\mathbb{E}\left[R\left(\bar{g}_n\right) - R(g_\rho)\right] = O\left(n^{-\frac{2r}{2r+\alpha}}\right) \tag{3}$$

Conclusion 1

Optimal statistical rates in RKHS

Choice of γ

Conclusion 1

Optimal statistical rates in RKHS

Choice of γ

- We get statistical optimal rate of convergence for learning in RKHS with SGD with one pass.

Conclusion 1

Optimal statistical rates in RKHS

Choice of γ

- We get statistical optimal rate of convergence for learning in RKHS with SGD with one pass.
- We get insights on how to choose the kernel and the step size.

Conclusion 1

Optimal statistical rates in RKHS

Choice of γ

- We get statistical optimal rate of convergence for learning in RKHS with SGD with one pass.
- We get insights on how to choose the kernel and the step size.
- We compare favorably to [Ying and Pontil, 2008, Caponnetto and De Vito, 2007, Tarrès and Yao, 2011].

Conclusion 2

Behaviour in FD

Adaptativity, tradeoffs.

Conclusion 2

Behaviour in FD

Adaptativity, tradeoffs.

Theorem can be rewritten :

$$\mathbb{E}\left[R\left(\bar{\theta}_n\right) - R(\theta_*)\right] \leqslant O\left(\frac{\sigma^2 \operatorname{tr}(\Sigma^\alpha)\gamma^\alpha}{n^{1-\alpha}}\right) + O\left(\frac{\theta_*^T \Sigma^{2r-1} \theta^T}{(n\gamma)^{2(r\wedge 1)}}\right) \tag{4}$$

where the ellipsoid condition appears more clearly.

Conclusion 2

Behaviour in FD

Adaptativity, tradeoffs.

Theorem can be rewritten :

$$\mathbb{E}\left[R\left(\bar{\theta}_n\right) - R(\theta_*)\right] \leqslant O\left(\frac{\sigma^2 \operatorname{tr}(\Sigma^\alpha)\gamma^\alpha}{n^{1-\alpha}}\right) + O\left(\frac{\theta_*^T \Sigma^{2r-1} \theta^T}{(n\gamma)^{2(r\wedge 1)}}\right) \tag{4}$$

where the ellipsoid condition appears more clearly.
Thus :

- SGD is adaptative to the regularity of the problem
- bridges the gap between the different regimes and explains behaviour when $d >> n$.

Reducing complexity : sampling methods

However the complexity of such a method remains quadratic with respect of the number of examples : iteration number n costs n kernel calculations.

Reducing complexity : sampling methods

However the complexity of such a method remains quadratic with respect of the number of examples : iteration number n costs n kernel calculations.

	Rate	Complexity
Finite Dimension	$\frac{d}{n}$	$O(dn)$

Reducing complexity : sampling methods

However the complexity of such a method remains quadratic with respect of the number of examples : iteration number n costs n kernel calculations.

	Rate	Complexity
Finite Dimension	$\frac{d}{n}$	$O(dn)$
Infinite dimension	$\frac{d_n}{n}$	$O(n^2)$

2 related methods

- Approximate the kernel matrix
- Approximate the kernel

Results from [Bach, 2012].
Such results have been extended by [Alaoui and Mahoney, 2014, Rudi et al.,
There also exist results in the second situation [Rahimi and Recht, 2008, Dai et al., 2014]

Sharp analysis

We only consider a fixed design setting. Then we have to approximate the kernel matrix : instead of computing the whole matrix, we randomly pick a number d_n of columns.

Sharp analysis

We only consider a fixed design setting. Then we have to approximate the kernel matrix : instead of computing the whole matrix, we randomly pick a number d_n of columns.
Then we still get the same estimation errors.
Leading to :

	Rate	Complexity
Finite Dimension	$\frac{d}{n}$	$O(dn)$
Infinite dimension	$\frac{d_n}{n}$	$O(nd_n^2)$

Random feature selection

Many kernels may be represented, due to Bochner's theorem as

$$K(x, y) = \int_W \phi(w, x)\phi(w, y) d\mu(w).$$

(think of translation invariant kernels and Fourier transform).

Random feature selection

Many kernels may be represented, due to Bochner's theorem as

$$K(x,y) = \int_W \phi(w,x)\phi(w,y)d\mu(w).$$

(think of translation invariant kernels and Fourier transform).
We thus consider the low rank approximation :

$$\tilde{K}(x,y) = \frac{1}{d}\sum_{i=1}^{n} \phi(x,w_i)\phi(y,w_i).$$

where $w_i \sim \mu$.
We use this approximation of the kernel in SGD.

Directions

What I am working on for the moment :

- Random feature selection
- Tuning the sampling to improve accuracy of the approximation
- Acceleration + stochasticity (with Nicolas Flammarion).

Some references I

Alaoui, A. E. and Mahoney, M. W. (2014).
Fast randomized kernel methods with statistical guarantees.
CoRR, abs/1411.0306.

Bach, F. (2012).
Sharp analysis of low-rank kernel matrix approximations.
ArXiv e-prints.

Bach, F. and Moulines, E. (2013).
Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n).
ArXiv e-prints.

Bottou, L. and Bousquet, O. (2008).
The tradeoffs of large scale learning.
In *IN : ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 20*.

Caponnetto, A. and De Vito, E. (2007).
Optimal Rates for the Regularized Least-Squares Algorithm.
Foundations of Computational Mathematics, 7(3) :331–368.

Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M., and Song, L. (2014).
Scalable kernel methods via doubly stochastic gradients.
In *Advances in Neural Information Processing Systems 27 : Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3041–3049.

Dieuleveut, A. and Bach, F. (2014).
Non-parametric Stochastic Approximation with Large Step sizes.
ArXiv e-prints.

Some references II

Rahimi, A. and Recht, B. (2008).
Weighted sums of random kitchen sinks : Replacing minimization with randomization in learning.
In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 1313–1320.

Rudi, A., Camoriano, R., and Rosasco, L. (2015).
Less is more : Nyström computational regularization.
CoRR, abs/1507.04717.

Shalev-Schwartz, S. and K., S. (2011).
Theorical basis for more data less work.

Shalev-Schwartz, S. and Srebro, N. (2008).
SVM optimisation : Inverse dependance on training set size.
Proceedings of the International Conference on Machine Learning (ICML).

Tarrès, P. and Yao, Y. (2011).
Online learning as stochastic approximation of regularization paths.
ArXiv e-prints 1103.5538.

Ying, Y. and Pontil, M. (2008).
Online gradient descent learning algorithms.
Foundations of Computational Mathematics, 5.

Thank you for your attention !