ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Winter Conference in Statistics 2013

Compressed Sensing

LECTURE #12

Nonparametric function learning

lions@epfl

Prof. Dr. Volkan Cevher

volkan.cevher@epfl.ch

LIONS/Laboratory for Information and Inference Systems

Function learning

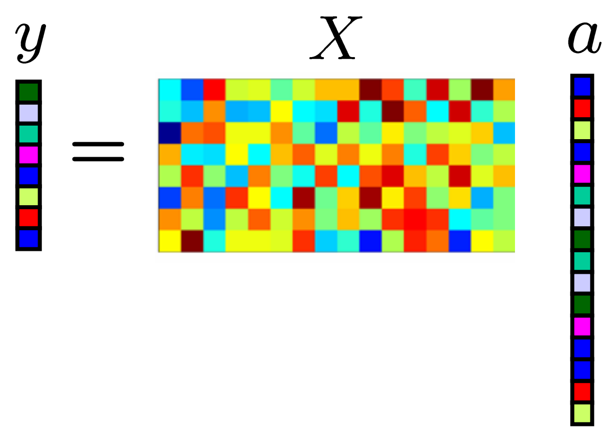
- A fundamental problem:

  given $(y_i, x_i)\colon\ \mathbb{R} \times \mathbb{R}^d, i = 1, \ldots, m,$ learn a mapping $f\colon\ x \to y$

  – some call it “regression”

- Oft-times ***f*** < > parametric form

  e.g., linear regression



learning the model
=
learning the parameters

$$f(x) = a^t x$$

Function learning

- A fundamental problem:

  given (y_i, x_i): $\mathbb{R} \times \mathbb{R}^d, i = 1, \ldots, m,$ learn a mapping $f\colon x \to y$

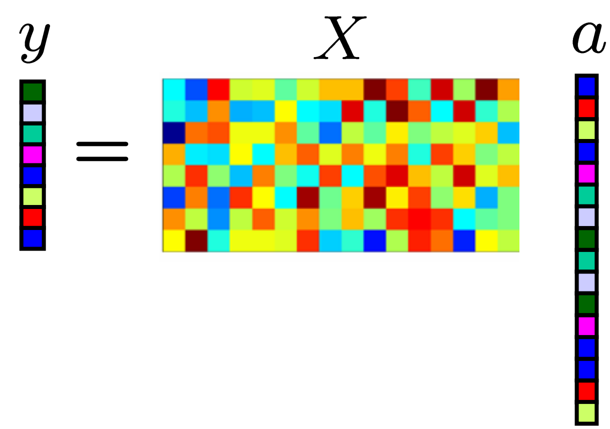
  – some call it "regression"

- Oft-times ***f*** <> parametric form

  e.g., linear regression

learning the model
=
learning the parameters

$$f(x) = a^t x$$



familiar challenge: ***learning via dimensionality reduction***

Function learning

- A fundamental problem:

  given (y_i, x_i): $\mathbb{R} \times \mathbb{R}^d, i = 1, \ldots, m,$ learn a mapping $f\colon x \to y$

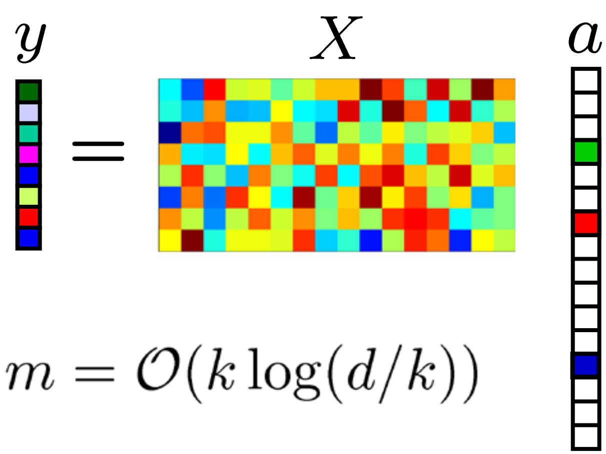
  – some call it "regression"

- Oft-times ***f*** <> parametric form

  e.g., linear regression

learning a
low-dimensional model
=
successful learning the
parameters

$$f(x) = a^t x$$



$$\mathcal{S} = \{i : x_i \neq 0\}$$

$$\|\alpha\|_0 = |\mathcal{S}| = k$$

$$m = \mathcal{O}(k \log(d/k))$$

familiar challenge: ***learning via dimensionality reduction***

Function learning

- A fundamental problem:

  given (y_i, x_i): $\mathbb{R} \times \mathbb{R}^d, i = 1, \ldots, m,$ learn a mapping $f\colon x \to y$

  – some call it "regression"

- Oft-times ***f*** <> parametric form

  e.g., linear regression

  low-dim models >> successful learning

  sparse,

  low-rank*...*

- Any parametric form <> at best an approximation

  emerging alternative: ***non-parametric models***

  learn ***f*** from data!

Function learning

- A fundamental problem:

  given $(y_i, x_i)\colon \mathbb{R} \times \mathbb{R}^d, i = 1, \ldots, m,$ learn a mapping $f\colon x \to y$

  – some call it "regression"

- Oft-times ***f*** <> parametric form

  e.g., linear regression

  low-dim models >> successful learning

  sparse,

  low-rank...

- Any parametric form <> at best an approximation

  emerging alternative: ***non-parametric models***

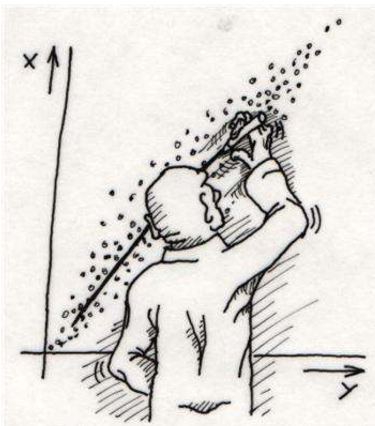
this lecture→ learn **low-dim** ***f*** from data!

Nonparametric model learning

Two distinct camps:

1. Regression < > use given samples

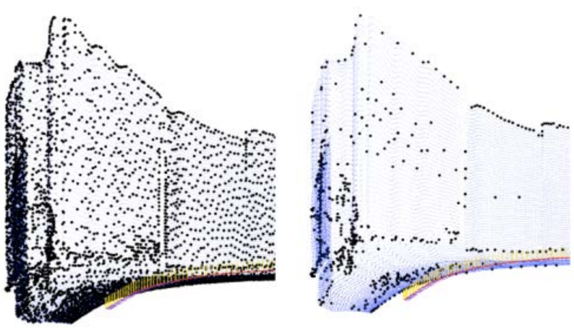
approximation of f



[Friedman and Stuetzle 1981; Li 1991, 1992; Lin and Zhang 2006; Xia 2008; Ravikumar et al., 2009; Raskutti et al., 2010]

2. Active learning (experimental design) < > design a sampling scheme

approximation of f



[Cohen et al., 2010; Fornasier, Schnass, Vybiral, 2011; VC and Tyagi 2012; Tyagi and VC 2012]

maximization/optimization of f

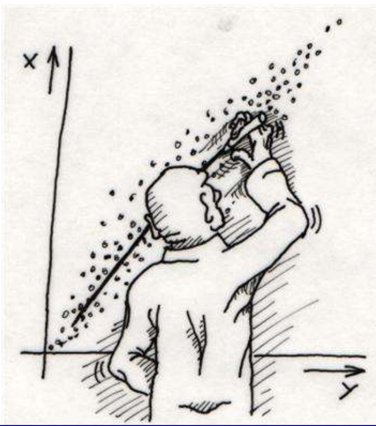
[Srinivas, Krause, Kakade, Seeger, 2012]

Nonparametric model learning—our contributions

Two distinct camps:

1. Regression < > use given samples

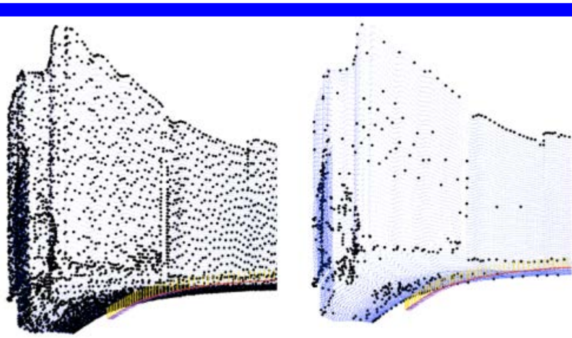
approximation of f



[Friedman and Stuetzle 1981; Li 1991, 1992; Lin and Zhang 2006; Xia 2008; Ravikumar et al., 2009; Raskutti et al., 2010]

2. Active learning (experimental design) < > design a sampling scheme

approximation of f



[Cohen et al., 2010; Fornasier, Schnass, Vybiral, 2011; VC and Tyagi 2012; Tyagi and VC 2012]

maximization/optimization of f

[Srinivas, Krause, Kakade, Seeger, 2012]

Active function learning

- A motivation by Albert Cohen

Numerical solution of parametric PDE's

$$x \in \mathbb{R}^d$$
$$f \in \Omega$$

$\mathrm{PDE}(f, x) = 0 \longmapsto f(x)$ **: the (implicit) solution**

query of the solution < > running an expensive simulation

Active **function learning**

- A motivation by Albert Cohen

Numerical solution of parametric PDE's

$$x \in \mathbb{R}^d$$

$$f \in \Omega$$

$\mathrm{PDE}(f, x) = 0 \longmapsto f(x)$ **: the (implicit) solution**

query of the solution < > running an expensive simulation

learn an explicit approximation of *f* via multiple queries

Active **function learning**

- A motivation by Albert Cohen

Numerical solution of parametric PDE's

$$x \in \mathbb{R}^d$$

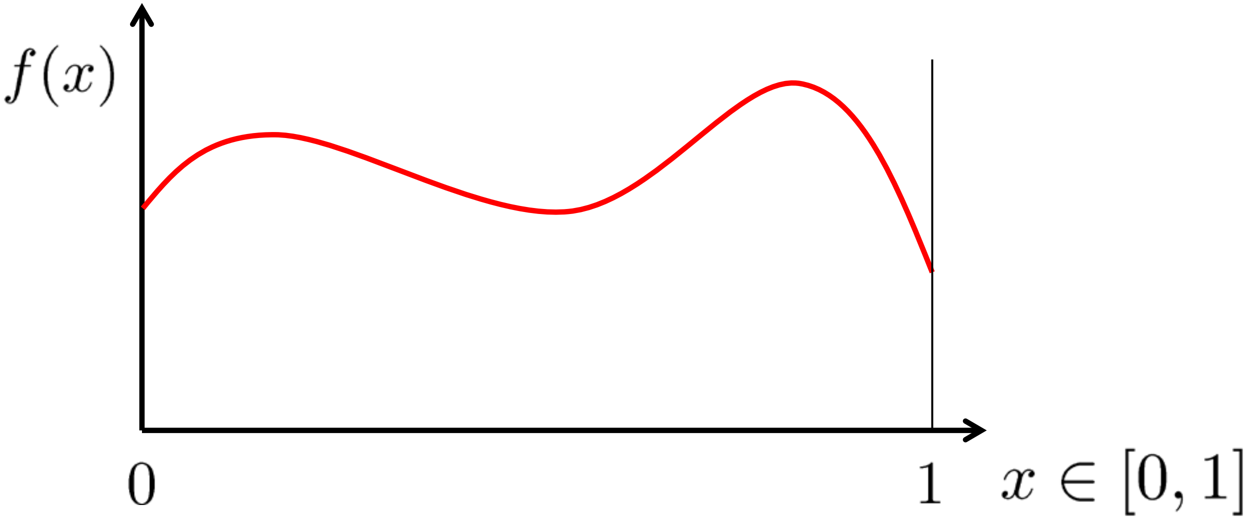
$$f \in \Omega$$

$$\mathrm{PDE}(f, x) = 0 \longmapsto f(x)$$ **: the (implicit) solution**

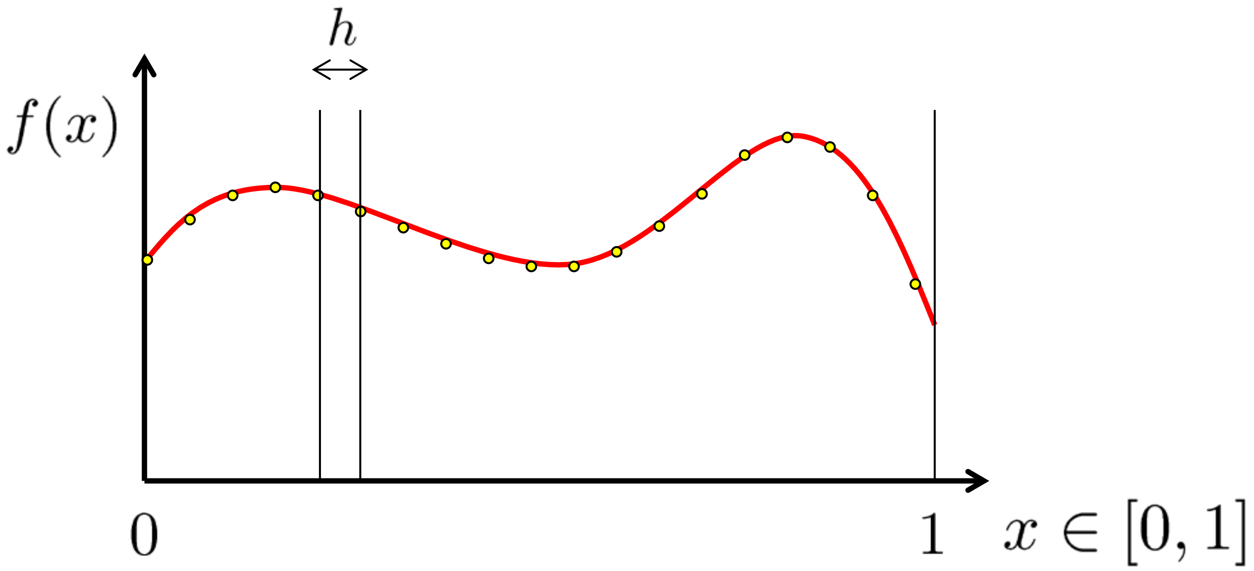
query of the solution < > running an expensive simulation

ability to choose the samples **< >** ***<u>active learning</u>***

Learning via interpolation

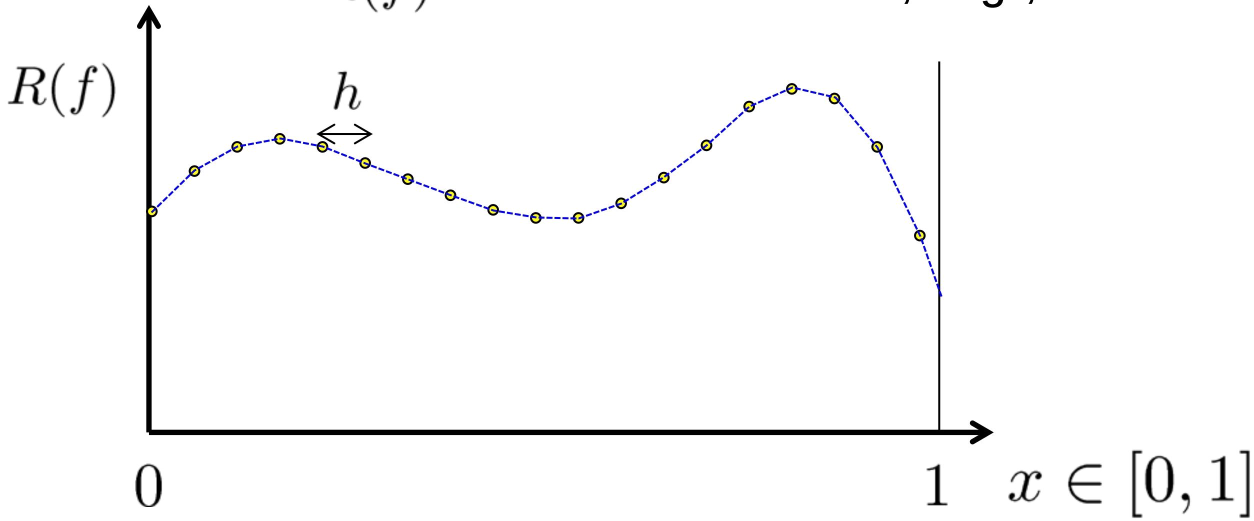


Learning via interpolation



Learning via interpolation

$R(f)$: reconstruction via, e.g., linear interpolation

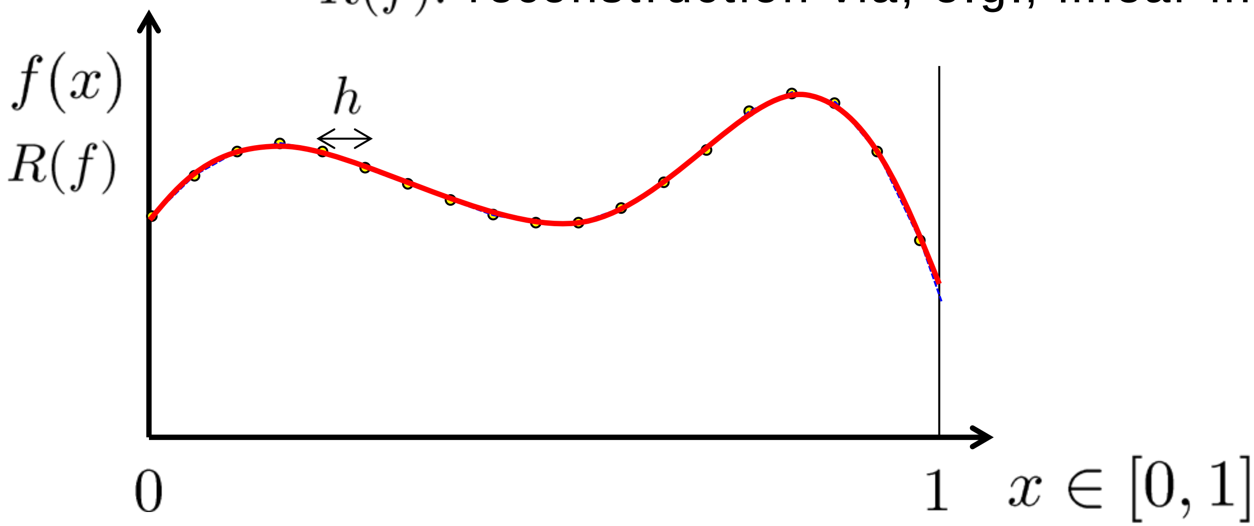


- Error characterization for smooth $f \in \mathcal{C}^s$

$$\|f - R(f)\|_\infty \leq C \|D^s f\|_\infty h^s$$

Learning via interpolation

$R(f)$: reconstruction via, e.g., linear interpolation



- Error characterization for smooth $f \in \mathcal{C}^s$

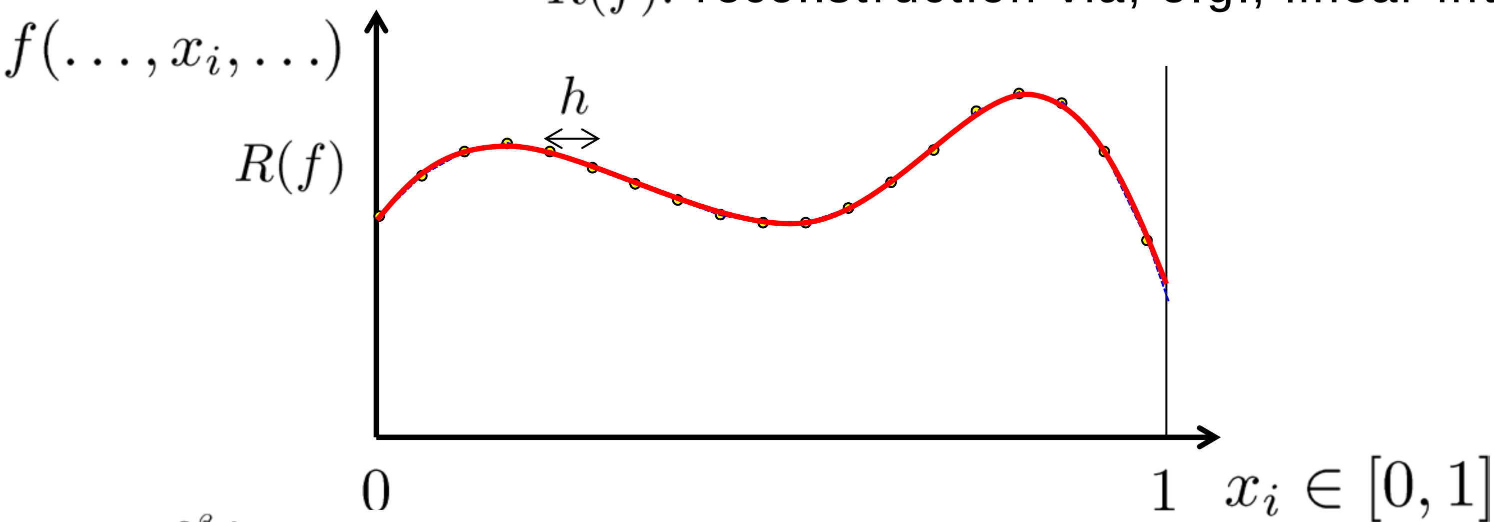
$$\|f - R(f)\|_\infty \leq C \|D^s f\|_\infty h^s$$

number of samples $N = \mathcal{O}(h^{-1})$ <> $\|f - R(f)\|_\infty = \mathcal{O}(N^{-s})$

Learning via interpolation

Curse-of-dimensionality

$R(f)$: reconstruction via, e.g., linear interpolation



$$D^\beta f = \frac{\partial^\beta f}{\partial y_1^{\beta_1} \ldots \partial y_k^{\beta_k}} \; ; \quad \beta = \beta_1 + \cdots + \beta_k, \{\beta_i\}_{i=1}^k \in \mathbb{Z}_+$$

- Error characterization for smooth $f \in \mathcal{C}^s$ and $x \in \mathbb{R}^d$

$$\|f - R(f)\|_\infty \leq C \|D^s f\|_\infty h^s$$

number of samples $N = \mathcal{O}(h^{-d})$ <> $\|f - R(f)\|_\infty = \mathcal{O}(N^{-s/d})$

Learning via interpolation

Curse-of-dimensionality

- The nonlinear N-width

$$d_N(\Omega) := \inf_{E,R} \max_{f \in \Omega} \| f - R(E(f)) \|_\infty$$

E: encoder $\Omega \to \mathbb{R}^N$
R: reconstructor $\mathbb{R}^N \to \Omega$
Ω: compact set

infimum is taken over all continuous maps (E,R)

$$\Omega = \mathcal{C}^s([0,1]^d) \Rightarrow cN^{-s/d} \le d_N(\Omega) \le CN^{-s/d}$$

[Traub et al., 1988; Devore, Howard, and Micchelli 1989]

Learning via interpolation

Curse-of-dimensionality

- The nonlinear N-width

$$d_N(\Omega) := \inf_{E,R} \max_{f\in\Omega} \|f - R(E(f))\|_\infty$$

E: encoder $\Omega \to \mathbb{R}^N$
R: reconstructor $\mathbb{R}^N \to \Omega$
Ω: compact set

infimum is taken over all continuous maps (E,R)

$$\Omega = \mathcal{C}^s([0,1]^d) \Rightarrow \min\{N : d_N(\Omega) \leq \epsilon\} \geq c\,(1/\epsilon)^{d/s}$$

[Traub et al., 1988; Devore, Howard, and Micchelli 1989]

Learning via interpolation

Curse-of-dimensionality

- The nonlinear N-width

$$d_N(\Omega) := \inf_{E,R} \max_{f \in \Omega} \|f - R(E(f))\|_\infty$$

E: encoder $\Omega \to \mathbb{R}^N$
R: reconstructor $\mathbb{R}^N \to \Omega$
Ω: compact set

infimum is taken over all continuous maps (E,R)

$$\Omega = \mathcal{C}^s([0,1]^d) \Rightarrow \min\{N : d_N(\Omega) \le \epsilon\} \ge c\,(1/\epsilon)^{d/s}$$

$$\Omega = \mathcal{C}^\infty([0,1]^d) \Rightarrow \min\{N : d_N(\Omega) \le 0.5\} \ge c2^{d/2}$$

- Take home message

smoothness-only >> intractability in sample complexity

need additional assumptions on the problem structure!!!

[Traub et al., 1988; Devore, Howard, and Micchelli 1989; Nowak and Wosniakowski 2009]

Learning multi-ridge functions

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$ $\quad k < d$

Model 2: $f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$

$$f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R} \qquad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

other names:
multi-index models
partially linear single/multi index models
generalized additive model
sparse additive models...

[Friedman and Stuetzle 1981; Li 1991, 1992; Lin and Zhang 2006; Xia 2008; Ravikumar et al., 2009; Raskutti et al., 2010; Cohen et al., 2010; Fornasier, Schnass, Vybiral, 2011; VC and Tyagi 2012; Tyagi and VC 2012]

Prior Art

Prior work—Regression camp

- ***local smoothing*** <> *first order* low-rank model
  a common approach in nonparametric regression (kernel, nearest neighbor, splines)

  [Friedman and Stuetzle 1981; Li 1991, 1992; Fan and Gijbels 1996; Lin and Zhang 2006; Xia 2008]

Prior work—Regression camp

- ***local smoothing*** < > *first order* low-rank model

  a common approach in nonparametric regression (kernel, nearest neighbor, splines)

  [Friedman and Stuetzle 1981; Li 1991, 1992; Fan and Gijbels 1996; Lin and Zhang 2006; Xia 2008]

  $$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$$

  1. assume orthogonality

  $$\mathbf{A}\mathbf{A}^T = \mathbf{I}_k$$

  $$f(\mathbf{x}) = g(\overbrace{\mathbf{U}\Sigma\mathbf{V}^T}^{\text{SVD of } \mathbf{A}}\mathbf{x}) = \bar{g}(\mathbf{V}^T\mathbf{x}),$$
  where $\bar{g}(\mathbf{y}) = g(\mathbf{U}\Sigma\mathbf{y})$

Prior work—Regression camp

- ***local smoothing*** < > *first order* low-rank model

  a common approach in nonparametric regression (kernel, nearest neighbor, splines)

  [Friedman and Stuetzle 1981; Li 1991, 1992; Fan and Gijbels 1996; Lin and Zhang 2006; Xia 2008]

$$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$$

1. assume orthogonality

$$\mathbf{A}\mathbf{A}^T = \mathbf{I}_k$$

2. note the differentiability of ***f***

$$\nabla f(\mathbf{x}) = \mathbf{A}^T \nabla g(\mathbf{A}\mathbf{x})$$

$$f(\mathbf{x}) = g(\underbrace{\mathbf{U}\Sigma\mathbf{V}^T}_{\text{SVD of } \mathbf{A}}\mathbf{x}) = \bar{g}(\mathbf{V}^T\mathbf{x}),$$

where $\bar{g}(\mathbf{y}) = g(\mathbf{U}\Sigma\mathbf{y})$

Key observation #1:
gradients live in at most k-dim. subspaces

Prior work—Regression camp

- ***local smoothing*** < > *first order* low-rank model

a common approach in nonparametric regression (**kernel, nearest neighbor, splines**)

[Friedman and Stuetzle 1981; Li 1991, 1992; Fan and Gijbels 1996; Lin and Zhang 2006; Xia 2008]

$$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$$

1. assume orthogonality

$$\mathbf{A}\mathbf{A}^T = \mathbf{I}_k$$

SVD of $\mathbf{A}$

$$f(\mathbf{x}) \;=\; g(\underbrace{\mathbf{U}\Sigma\mathbf{V}^T}_{\text{SVD of } \mathbf{A}}\mathbf{x}) \;=\; \bar{g}(\mathbf{V}^T\mathbf{x}),$$
$$\text{where } \bar{g}(\mathbf{y}) = g(\mathbf{U}\Sigma\mathbf{y})$$

2. note the differentiability of ***f***

$$\nabla f(\mathbf{x}) = \mathbf{A}^T \nabla g(\mathbf{A}\mathbf{x})$$

Key observation #1:
gradients live in at most k-dim. subspaces

3. leverage samples to obtain the hessian via local **K / N-N / S**...

$$H^f := \mathbf{A}^T H^g \mathbf{A}$$

required: rank-k H^g

$$H^f := E\left\{\left[\nabla f(\mathbf{x}) - E(\nabla f(\mathbf{x}))\right]\left[\nabla f(\mathbf{x}) - E(\nabla f(\mathbf{x}))\right]^T\right\}$$

Key observation #2:
k- principal components of H^f leads to A

Prior work—Regression camp

- local smoothing < > *first order* low-rank model
  a common approach in nonparametric regression (kernel, nearest neighbor, splines)

  [Friedman and Stuetzle 1981; Li 1991, 1992; Fan and Gijbels 1996; Lin and Zhang 2006; Xia 2008]

- Recent trends < > ***additive sparse models***

  [Stone 1985; Tibshirani and Hastie 1990; Lin Zhang 2006; Ravikumar et al., 2009; Raskutti et al., 2010; Meier et al. 2007; Koltchinski and Yuan, 2008, 2010]

$$f(x_1, \ldots, x_d) = \sum_{j : j \in \mathcal{S}, |S| \leq k} g_j(x_j)$$

$$f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$$

  - encode ***smoothness*** via the kernel
  - leverage sparse greedy/convex optimization
  - establish consistency rates: $\|f - \widehat{f}\|_{L_2} \leq \mathcal{O}\left(k\delta^2 + \frac{k \log(d)}{m}\right)$

Prior work—Regression camp

- local smoothing < > *first order* low-rank model

  a common approach in nonparametric regression (kernel, nearest neighbor, splines)

  [Friedman and Stuetzle 1981; Li 1991, 1992; Fan and Gijbels 1996; Lin and Zhang 2006; Xia 2008]

- Recent trends < > ***additive sparse models***

  [Stone 1985; Tibshirani and Hastie 1990; Lin Zhang 2006; Ravikumar et al., 2009; Raskutti et al., 2010; Meier et al. 2007; Koltchinski and Yuan, 2008, 2010]

$$f(x_1, \ldots, x_d) = \sum_{j : j \in \mathcal{S}, |S| \leq k} g_j(x_j)$$

g belongs to reproducing kernel Hilbert space

- encode ***smoothness*** via the kernel
- leverage sparse greedy/convex optimization
- establish consistency rates:

difficulty of estimating the kernel	difficulty of subset selection

$$\|f - \widehat{f}\|_{L_2} \leq \mathcal{O}\left(k\delta^2 + \frac{k \log(d)}{m}\right)$$

Prior work—Active learning camp

- Progress thus far < > ***the sparse way***

  highlights:

  1. Cohen, Daubechies, DeVore, Kerkyacharian, and Picard (2010)

$$f(\mathbf{x}) = g(\mathbf{a}^T\mathbf{x})$$

$$g : [0, 1] \to \mathbb{R} \text{ is a } \mathcal{C}^s \text{ function for } s > 1$$

$$\mathbf{a} \succeq 0, \mathbf{1}^T\mathbf{a} = 1 \quad \mathbf{a} \in w\ell_q \quad q < 1$$

(i.e., compressible)

Prior work—Active learning camp

- Progress thus far < > ***the sparse way***

highlights:

1. Cohen, Daubechies, DeVore, Kerkyacharian, and Picard (2010) $f(\mathbf{x}) = g(\mathbf{a}^T\mathbf{x})$

$$g : [0,1] \to \mathbb{R} \text{ is a } \mathcal{C}^s \text{ function for } s > 1$$

$$\mathbf{a} \succeq 0, \mathbf{1}^T\mathbf{a} = 1 \quad \mathbf{a} \in w\ell_q \quad q < 1$$ (i.e., compressible)

2. Fornassier, Schnass, and Vybiral (2011) $f(\mathbf{x}) = g(A\mathbf{x})$

$$g : B_{\mathbb{R}^d}(1+\bar{\epsilon}) \to \mathbb{R} \text{ is } \mathcal{C}^s \quad \mathbf{a}_i \in w\ell_q, q < 2 \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

extends on the same ***local observation model*** in regression

Taylor series

$$f(\mathbf{x} + \epsilon\phi) = f(\mathbf{x}) + \epsilon\left\langle \phi, \nabla f(\mathbf{x})\right\rangle + \epsilon E(\mathbf{x}, \epsilon, \phi) \qquad \epsilon \ll 1$$

$$\Rightarrow \left\langle \phi, A^T \nabla g(A\mathbf{x})\right\rangle = \frac{1}{\epsilon}\left(f(\mathbf{x} + \epsilon\phi) - f(\mathbf{x})\right) - E(\mathbf{x}, \epsilon, \phi)$$

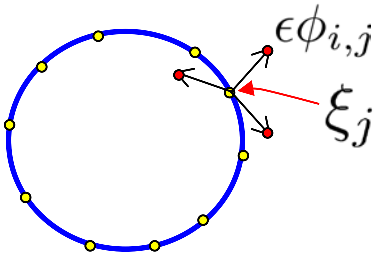
Prior work—Active learning camp (FSV'11)

- A *sparse* observation model $f(\mathbf{x}) = g(A\mathbf{x})$

$$\Rightarrow \langle \phi_{i,j}, A^T \nabla g(A\xi_j) \rangle = \frac{1}{\epsilon}\left(f(\xi_j + \epsilon\phi_{i,j}) - f(\xi_j)\right) - E(\xi_j, \epsilon, \phi_{i,j})$$

curvature effect $E(\mathbf{x}, \epsilon, \phi) = \frac{\epsilon}{2}\phi^T \nabla^2 f(\zeta(\mathbf{x}, \phi))\phi$

$\zeta(\mathbf{x}, \phi) \in [\mathbf{x}, \mathbf{x} + \epsilon\phi]$



with two ingredients

sampling centers $\mathcal{X} = \{\xi_j \in \mathbb{S}^{d-1}; j = 1, \ldots, m_{\mathcal{X}}\}$

sampling directions at each center $\Phi_j = [\phi_{1,j} | \ldots | \phi_{m_\Phi, j}]^T$

leads to

$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \mathbf{\Phi})$$

$\mathbf{X}_i := \mathbf{A}^T \mathbf{G}_i$ — approximately sparse

$y_i = \sum_{j=1}^{m_{\mathcal{X}}} \left[\frac{f(\xi_j + \epsilon\phi_{i,j}) - f(\xi_j)}{\epsilon}\right]$ $\quad \mathbf{G} := [\nabla g(\mathbf{A}\xi_1) | \nabla g(\mathbf{A}\xi_2) | \cdots | \nabla g(\mathbf{A}\xi_{m_{\mathcal{X}}})]_{k \times m_{\mathcal{X}}}$

Prior work—Active learning camp (FSV'11)

$$f(\mathbf{x}) = g(A\mathbf{x})$$

- A sparse observation model

$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \mathbf{\Phi})$$

$$\mathbf{X}_i := \mathbf{A}^T \mathbf{G}_i$$

approximately sparse

- Key contribution: restricted "Hessian" property

$$H^f := \int_{\mathbb{S}^{d-1}} \nabla f(\mathbf{x}) \nabla f(\mathbf{x})^T d\mu_{\mathbb{S}^{d-1}}(\mathbf{x})$$

μ: uniform measure

$$\sigma_1(H^f) \geq \sigma_2(H^f) \geq \ldots \geq \sigma_k(H^f) \geq \alpha > 0 \text{ for some } \alpha$$

recall **G** needs to span a k-dim subspace for identifiability of ***A***

$$\mathbf{G} := [\nabla g(\mathbf{A}\xi_1) | \nabla g(\mathbf{A}\xi_2) | \cdots | \nabla g(\mathbf{A}\xi_{m_{\mathcal{X}}})]_{k \times m_{\mathcal{X}}}$$

with a restricted study of radial functions $f(\mathbf{x}) = g_0(\|A\mathbf{x}\|_2)$

- Analysis < > leverage compressive sensing results

Prior work—Active learning camp (FSV'11)

$$f(\mathbf{x}) = g(A\mathbf{x})$$

- A sparse observation model

$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \mathbf{\Phi})$$

$$\underbrace{\mathbf{X}_i := \mathbf{A}^T \mathbf{G}_i}_{\text{approximately sparse}}$$

- Analysis < > leverage compressive sensing results

- Key contribution: restricted Hessian property for radial functions $f(\mathbf{x}) = g_0(\|A\mathbf{x}\|_2)$

- Two major issues remains to be addressed over FSV'11

1. validity of orthogonal sparse/compressible directions

   need a basis independent model

$$f(\mathbf{x}) = g(\mathbf{A}\Psi^T\Psi\mathbf{x}) = g(A\Psi\mathbf{x}) \quad \text{one } \Psi \text{ for all orthogonal directions?}$$

2. analysis of ***H^f*** for anything other than radial functions

   need a new analysis tool

$$H^f := \int_{\mathbb{S}^{d-1}} \nabla f(\mathbf{x}) \nabla f(\mathbf{x})^T d\mu_{\mathbb{S}^{d-1}}(\mathbf{x})$$

Learning **multi-ridge** functions

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$$ $k < d$

Model 2: $$f(x_1, \ldots, x_d) \;=\; \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$$

$$f : B_{\mathbb{R}^d}(1+\bar{\epsilon}) \rightarrow \mathbb{R} \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

- ***Results:***

w.l.o.g. $g, g_i \in \mathcal{C}^2$

A: compressible

(Model 1): $$m = \mathcal{O}\left(\left(\tfrac{1}{\varepsilon}\right)^{k/2} + \frac{k^{\frac{4-q}{2-q}} d^{\frac{q}{2-q}} \log(k)}{\alpha}\right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$$

[Fornasier, Schnass, Vybiral, 2011]

*if *g* has k-restricted Hessian property...

Learning multi-ridge functions

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$$ $k < d$

Model 2: $$f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$$

$$f : B_{\mathbb{R}^d}(1+\bar{\epsilon}) \to \mathbb{R} \qquad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

- ***Results:***

cost of learning ***g***

w.l.o.g. $g, g_i \in \mathcal{C}^2$

A: compressible

(Model 1): $$m = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{k/2} + \frac{k^{\frac{4-q}{2-q}} d^{\frac{q}{2-q}} \log(k)}{\alpha} \right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$$

*cost of learning ***A***

[Fornasier, Schnass, Vybiral, 2011]

*if *f* has k-restricted Hessian property...

Learning multi-ridge functions

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$ $k < d$

Model 2: $f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$

$$f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R} \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

- ***Results:***

cost of learning ***g***

w.l.o.g. $g, g_i \in \mathcal{C}^2$

A: compressible

(Model 1): $m = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{k/2} + k^{\frac{4-q}{2-q}} d^{\frac{2}{2-q}} \log(k)\right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$

only for radial basis functions

$$f(\mathbf{x}) = g_0(\|A\mathbf{x}\|_2)$$

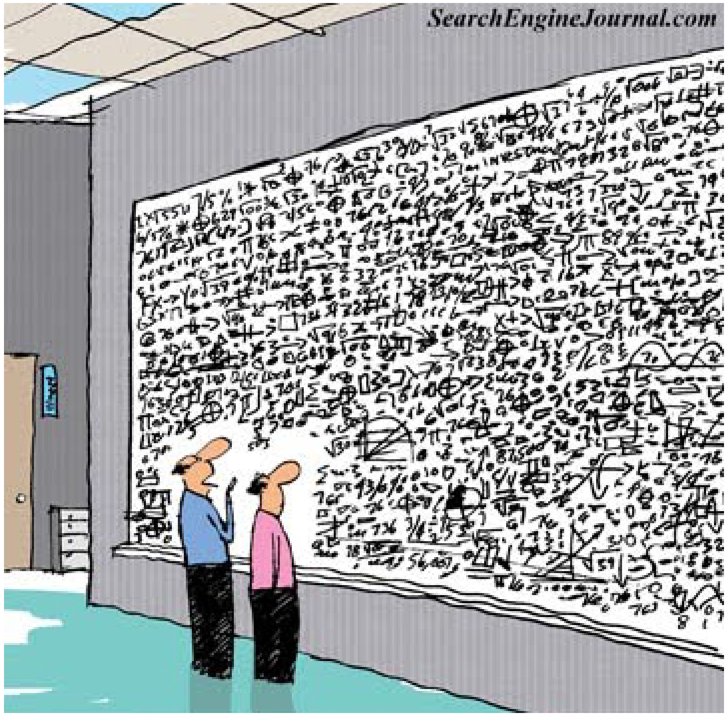
[Fornasier, Schnass, Vybiral, 2011]

*cost of learning ***A***

$\alpha = \Theta(\frac{1}{d})$

*if *f* has k-restricted Hessian property...

Learning Multi-Ridge Functions



...And, this is how you learn non-parametric basis independent models from point-queries via low-rank methods

Learning **multi-ridge** functions

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$$ $k < d$

Model 2: $$f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$$

$$f : B_{\mathbb{R}^d}(1+\bar{\epsilon}) \to \mathbb{R} \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

- ***Results:***

cost of learning ***g***

w.l.o.g. $g, g_i \in \mathcal{C}^2$

A: compressible

(Model 1&2): $$m = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{-k/2} + \frac{k^{\frac{4-q}{2-q}} d^{\frac{q}{2-q}} \log(k)}{\alpha} \right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$$

Our 1st contribution:
a simple verifiable characterization of alpha for a broad set of functions

*cost of learning ***A***

$$\alpha = \Theta\left(\frac{1}{d}\right)$$

*with the L-Lipschitz property...

Learning **multi-ridge** functions: the low-rank way

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$ $k < d$

Model 2: $f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$

$$f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R} \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

- ***Results:***

cost of learning ***g***

w.l.o.g. $g, g_i \in \mathcal{C}^2$

(Model 1): $m = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{-k/2} + \frac{k \log(k)}{\alpha} \times kd\right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$

*cost of learning ***A***

our 2nd contribution:
extension to the general A

*if *f* has k-restricted Hessian property...

Learning **multi-ridge** functions: the low-rank way

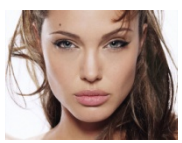
- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$ $k < d$

Model 2: $f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$

$f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R}$ $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$

- ***Results:***



cost of learning ***g_i's***

w.l.o.g. $g, g_i \in \mathcal{C}^2$

(Model 2): $m = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{1/2} k + \frac{k \log(k)}{\alpha} \times kd\right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$

*cost of learning ***A***

our 2nd contribution:
extension to the general A

*with the L-Lipschitz property...

Learning **multi-ridge** functions: the low-rank way

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$ $\quad k < d$

Model 2: $f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$

$(\mathbf{A}R)(\mathbf{A}R)^T = \mathbf{I}_k$

just kidding.

$f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \rightarrow \mathbb{R} \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$

- ***Results:***

w.l.o.g. $g, g_i \in \mathcal{C}^2$

cost of learning ***g_i's***

(Model 2): $m = \mathcal{O}\left((?)^{k/?} + \frac{k \log(k)}{\alpha} \times kd \right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$

*cost of learning ***A***

our 2nd contribution:
extension to the general A

*with the L-Lipschitz property...

Learning multi-ridge functions: the low-rank way

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$ $k < d$

Model 2: $f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$

in general $f : B_{\mathbb{R}^d}(1+\bar{\epsilon}) \to \mathbb{R}$ $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$

- ***Results:***

cost of learning ***g / g_i's*** w.l.o.g. $g, g_i \in \mathcal{C}^2$

(Model 1&2): $m = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{k/2} + k^2 d^2 \log(k)\right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$

*cost of learning ***A***

Given 1st and 2nd contribution:
full characterization of Model 1 & 2 with minimal assumptions

*with the L-Lipschitz property...

Learning **multi-ridge** functions

- *Objective:* *approximate* multi-ridge functions via point queries

Model 1: $$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$$ $k < d$

Model 2: $$f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(\mathbf{a}_i^T \mathbf{x})$$

$$f : B_{\mathbb{R}^d}(1+\bar{\epsilon}) \to \mathbb{R} \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

- ***Results:***

cost of learning ***g / g_i's*** $\quad$ w.l.o.g. $g, g_i \in \mathcal{C}^2$

(Model 1&2): $$m = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{k/2} + k^2 d^{4.5} \log(k)\right) \Rightarrow \|f - \widehat{f}\|_{L_\infty} \leq \varepsilon$$

*cost of learning ***A***

Our 3th contribution:
impact of iid noise *f+Z*

*with the L-Lipschitz property...

Non-sparse directions *A*

- A ***low-rank*** observation model

$$\langle \phi, A^T \nabla g(A\mathbf{x}) \rangle = \tfrac{1}{\epsilon} \left(f(\mathbf{x} + \epsilon\phi) - f(\mathbf{x}) \right) - E(\mathbf{x}, \epsilon, \phi)$$

along with two ingredients

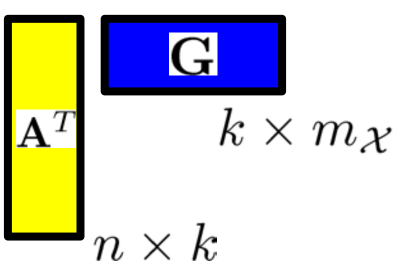
– sampling centers $\mathcal{X} = \{\xi_j \in \mathbb{S}^{d-1}; j = 1, \ldots, m_{\mathcal{X}}\}$

– sampling directions at each center $\Phi_j = [\phi_{1,j} | \ldots | \phi_{m_\Phi,j}]^T$

leads to

$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \mathbf{\Phi})$$

$\mathbf{X} :=$



$$y_i = \sum_{j=1}^{m_{\mathcal{X}}} \left[\frac{f(\xi_j + \epsilon\phi_{i,j}) - f(\xi_j)}{\epsilon} \right]$$

$$\mathbf{G} := [\nabla g(\mathbf{A}\xi_1) | \nabla g(\mathbf{A}\xi_2) | \cdots | \nabla g(\mathbf{A}\xi_{m_{\mathcal{X}}})]_{k \times m_{\mathcal{X}}}$$

Detour #2: low-rank recovery

$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \mathbf{\Phi})$$

$$\Phi : \mathbb{R}^{d \times m_{\mathcal{X}}} \to \mathbb{R}^{m_{\Phi}}$$

- Stable recovery <> measurements commensurate with degrees of freedom
  - stable recovery: $\|\mathbf{X} - \widehat{\mathbf{X}}\|_{\mathrm{F}} \leq C_1 \|\mathbf{X} - \mathbf{X}_k\|_{\mathrm{F}} + C_2 \|E\|_{\mathrm{F}}$
  - measurements: $m_{\Phi} = \mathcal{O}\left(k(d + m_{\mathcal{X}} - k)\right)$

$$\widehat{\mathbf{X}} = \Delta(\mathbf{y}, \Phi): \text{ decoder}$$

$$\mathbf{X}_k = \arg \min_{\mathbf{Z}:\mathrm{rank}(\mathbf{Z}) \leq k} \|\mathbf{X} - \mathbf{Z}\|_{\mathrm{F}}$$

Detour #2: low-rank recovery

$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \boldsymbol{\Phi})$$

$$\Phi : \mathbb{R}^{d \times m_{\mathcal{X}}} \rightarrow \mathbb{R}^{m_\Phi}$$

Matrix ALPS

http://lions.epfl.ch/MALPS

- Stable recovery <> measurements commensurate with degrees of freedom
  - stable recovery: $\|\mathbf{X} - \widehat{\mathbf{X}}\|_{\mathrm{F}} \leq C_1 \|\mathbf{X} - \mathbf{X}_k\|_{\mathrm{F}} + C_2 \|E\|_{\mathrm{F}}$
  - measurements: $m_\Phi = \mathcal{O}\left(k(d + m_{\mathcal{X}} - k)\right)$
- Convex/non-convex decoders <> sampling/noise type
  - affine rank minimization
  - matrix completion
  - robust principal component analysis

[Recht et al. (2010); Meka et al. (2009); Candes and Recht (2009); Candes and Tao (2010); Lee and Bresler (2010); Waters et al. (2011); Kyrillidis and Cevher (2012)]

Detour #2: low-rank recovery

$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \mathbf{\Phi})$$

$$\Phi : \mathbb{R}^{d \times m_{\mathcal{X}}} \to \mathbb{R}^{m_{\Phi}}$$

Matrix ALPS

http://lions.epfl.ch/MALPS

- Stable recovery <> measurements commensurate with degrees of freedom

Matrix restricted isometry property (RIP):

$$(1-\kappa_k) \leq \frac{\|\Phi \mathbf{X}\|_{\mathrm{F}}^2}{\|\mathbf{X}\|_{\mathrm{F}}^2} \leq (1+\kappa_k), \ \forall \mathbf{X} : \mathrm{rank}(\mathbf{X}) \leq k$$

[Plan 2011]

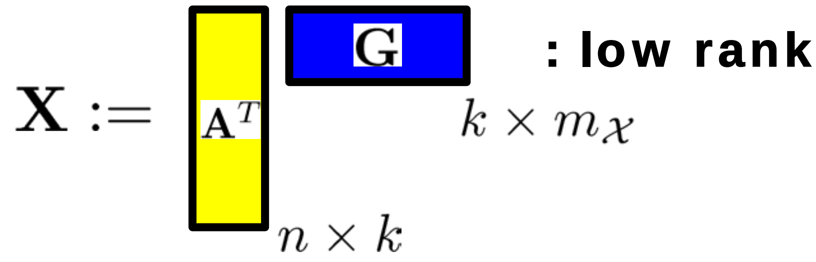
– affine rank minimization

– matrix completion

– robust principal component analysis

[Recht et al. (2010); Meka et al. (2009); Candes and Recht (2009); Candes and Tao (2010); Lee and Bresler (2010); Waters et al. (2011); Kyrillidis and Cevher (2012)]

Active sampling for RIP



$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \mathbf{\Phi})$$

- Recall the two ingredients $\Phi : \mathbb{R}^{d \times m_{\mathcal{X}}} \to \mathbb{R}^{m_\Phi}$
  - sampling centers $\mathcal{X} = \{\xi_j \in \mathbb{S}^{d-1}; j = 1, \ldots, m_{\mathcal{X}}\}$
  - sampling directions at each center $\Phi_j = [\phi_{1,j}|\ldots|\phi_{m_\Phi,j}]^T$

- Matrix RIP <> uniform sampling on the sphere

$$\mathbf{\Phi} = \left\{\phi_{i,j} \in B_{\mathbb{R}^d}\left(\sqrt{d/m_\Phi}\right) : [\phi_{i,j}]_l = \pm\frac{1}{\sqrt{m_\Phi}} \text{with probability } 1/2\right\}$$

$$\Rightarrow 0 < \kappa_r < \kappa < 1 \text{ with probability}$$

$$1-2e^{-m_\Phi q(\kappa)+r(d+m_{\mathcal{X}}+1)u(\kappa)}, \text{ where } q(\kappa) = \frac{1}{144}\left(\kappa^2 - \frac{\kappa^3}{9}\right) \text{ and } u(\kappa) = \log\left(\frac{36\sqrt{2}}{\kappa}\right)$$

[Candes and Plan (2010)]

Here it is… our low-rank approach

Algorithm 1 Estimating $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$

1: Choose m_Φ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\mathbf{\Phi}$.

2: Choose ϵ and construct $\mathbf{y}$ using $y_i = \sum_{j=1}^{m_\mathcal{X}} \left[\frac{f(\xi_j + \epsilon\phi_{i,j}) - f(\xi_j)}{\epsilon} \right]$.

3: Obtain $\widehat{\mathbf{X}}$ via a stable low-rank recovery algorithm.

4: Compute $\mathrm{SVD}(\widehat{\mathbf{X}}) = \widehat{\mathbf{U}}\widehat{\Sigma}\widehat{\mathbf{V}}^T$ and set $\widehat{\mathbf{A}}^T = \widehat{\mathbf{U}}^{(k)}$, corresponding to k largest singular values.

5: Obtain $\widehat{f}(\mathbf{x}) := \widehat{g}(\widehat{\mathbf{A}}\mathbf{x})$ via quasi interpolants where $\widehat{g}(\mathbf{y}) := f(\widehat{\mathbf{A}}^T\mathbf{y})$.

- ***achieve / balance three objectives simultaneously***

1. guarantee RIP on $\mathbf{\Phi}$ with m_Φ
2. ensure rank(***G***)=k with $m_\mathcal{X}$
3. contain ***E***'s impact with ϵ

$$\mathbf{y} = \Phi(\mathbf{X}) + E(\mathcal{X}, \epsilon, \mathbf{\Phi})$$

$$\mathbf{X} := \mathbf{A}^T\mathbf{G}$$

Here it is... our low-rank approach

Algorithm 1 Estimating $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$

1: Choose m_Φ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\mathbf{\Phi}$.
2: Choose ϵ and construct $\mathbf{y}$ using $y_i = \sum_{j=1}^{m_\mathcal{X}} \left[\frac{f(\xi_j + \epsilon\phi_{i,j}) - f(\xi_j)}{\epsilon}\right]$.
3: Obtain $\widehat{\mathbf{X}}$ via a stable low-rank recovery algorithm.
4: Compute $\mathrm{SVD}(\widehat{\mathbf{X}}) = \widehat{\mathbf{U}}\widehat{\Sigma}\widehat{\mathbf{V}}^T$ and set $\widehat{\mathbf{A}}^T = \widehat{\mathbf{U}}^{(k)}$, corresponding to k largest singular values.
5: Obtain $\widehat{f}(\mathbf{x}) := \widehat{g}(\widehat{\mathbf{A}}\mathbf{x})$ via quasi interpolants where $\widehat{g}(\mathbf{y}) := f(\widehat{\mathbf{A}}^T\mathbf{y})$.

1. guarantee RIP < > by construction
2. ensure rank(***G***)=k < > by Lipschitz assumption $\alpha = \Theta(\frac{1}{d})$

   rank-1 + diagonal / interval matrices

3. contain ***E***'s impact < > by controlling curvature $\epsilon = \mathcal{O}\left(\frac{\alpha}{d^{0.5}}\right)$
   - collateral damage: additive noise amplification by ϵ^{-1}

   solution: resample the **same** points $d^{3/2+\varepsilon}$-times

[VC and Tyagi 2012; Tyagi and VC, 2012]

L-Lipschitz property

- *New objective: approximate* ***A*** via point queries of ***f***

$$f : B_{\mathbb{R}^d}(1+\bar{\epsilon}) \to \mathbb{R} \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

- *New analysis tool:* L-Lipschitz 2nd order derivative

recall $H^f := \int_{\mathbb{S}^{d-1}} \nabla f(\mathbf{x}) \nabla f(\mathbf{x})^T d\mu_{\mathbb{S}^{d-1}}(\mathbf{x}) \quad \sigma_k(H^f) \geq \alpha > 0$

$$\frac{\left| \frac{\partial^2 g}{\partial y_i \partial y_j}(\mathbf{y}_1) - \frac{\partial^2 g}{\partial y_i \partial y_j}(\mathbf{y}_2) \right|}{\|\mathbf{y}_1 - \mathbf{y}_2\|_{l_2^k}} \leq L_{i,j}$$

Lipschitz constant
$L = \max_{1 \leq i,j \leq k} L_{i,j}$

Proposition: k-th restricted singular value

- *New objective: approximate* ***A*** via point queries of ***f***

$$f : B_{\mathbb{R}^d}(1+\bar{\epsilon}) \to \mathbb{R} \quad \mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]^T$$

- *New analysis tool:* *L*-Lipschitz 2nd order derivative

recall $H^f := \int_{\mathbb{S}^{d-1}} \nabla f(\mathbf{x}) \nabla f(\mathbf{x})^T d\mu_{\mathbb{S}^{d-1}}(\mathbf{x}) \quad \sigma_k(H^f) \geq \alpha > 0$

$$\frac{\left| \frac{\partial^2 g}{\partial y_i \partial y_j}(\mathbf{y}_1) - \frac{\partial^2 g}{\partial y_i \partial y_j}(\mathbf{y}_2) \right|}{\|\mathbf{y}_1 - \mathbf{y}_2\|_{l_2^k}} \leq L_{i,j}$$

Lipschitz constant $L = \max_{1 \leq i,j \leq k} L_{i,j}$

$$\Rightarrow \alpha = \Theta\left(\frac{1}{d}\right)$$

(Model 1): $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$

$+ \nabla^2 g(\mathbf{0})$ is full rank.

(Model 2): $f(\mathbf{x}) = \sum_{i=1}^k g_i(\mathbf{a}_i^T \mathbf{x})$ or $f(\mathbf{x}) = \mathbf{a}_1^T \mathbf{x} + \sum_{i=2}^k g_i(\mathbf{a}_i^T \mathbf{x})$

$+ \nabla^2 g_i(\mathbf{0}) \neq 0, \forall i = 2, \ldots, d$

Theorem: sample complexity

Algorithm 1 Estimating $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$

1: Choose m_Φ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\mathbf{\Phi}$.
2: Choose ϵ and construct $\mathbf{y}$ using $y_i = \sum_{j=1}^{m_\mathcal{X}} \left[\frac{f(\xi_j+\epsilon\phi_{i,j})-f(\xi_j)}{\epsilon}\right]$.
3: Obtain $\widehat{\mathbf{X}}$ via a stable low-rank recovery algorithm.
4: Compute $\text{SVD}(\widehat{\mathbf{X}}) = \widehat{\mathbf{U}}\widehat{\Sigma}\widehat{\mathbf{V}}^T$ and set $\widehat{\mathbf{A}}^T = \widehat{\mathbf{U}}^{(k)}$, corresponding to k largest singular values.
5: Obtain $\widehat{f}(\mathbf{x}) := \widehat{g}(\widehat{\mathbf{A}}\mathbf{x})$ via quasi interpolants where $\widehat{g}(\mathbf{y}) := f(\widehat{\mathbf{A}}^T\mathbf{y})$.

Theorem 1 *[Sample complexity of Algorithm 1] Let $\delta \in \mathbb{R}^+$, $\rho \ll 1$, and $\kappa < \sqrt{2}-1$ be fixed constants. Choose*

$$m_\mathcal{X} \geq \frac{2kC_2^2}{\alpha\rho^2}\log(k/p_1),$$

$$m_\Phi \geq \frac{\log(2/p_2) + 4k(d+m_\mathcal{X}+1)u(\kappa)}{q(\kappa)}, \text{ and}$$

$$\epsilon \leq \frac{\delta}{C_2 k^{5/2} d(\delta + 2C_2\sqrt{2k})}\left(\frac{(1-\rho)m_\Phi\alpha}{(1+\kappa)C_0 m_\mathcal{X}}\right)^{1/2}.$$

Then, given $m = m_\mathcal{X}(m_\Phi+1)$ samples, our function estimator $\widehat{f}$ in step 5 of Algorithm 1 obeys $\left\|f-\widehat{f}\right\|_{L_\infty} \leq \delta$ with probability at least $1-p_1-p_2$.

Theorem: sample complexity

Algorithm 1 Estimating $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$

1: Choose m_Φ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\mathbf{\Phi}$.
2: Choose ϵ and construct $\mathbf{y}$ using $y_i = \sum_{j=1}^{m_\mathcal{X}} \left[\frac{f(\xi_j + \epsilon\phi_{i,j}) - f(\xi_j)}{\epsilon}\right]$.
3: Obtain $\widehat{\mathbf{X}}$ via a stable low-rank recovery algorithm.
4: Compute $\text{SVD}(\widehat{\mathbf{X}}) = \widehat{\mathbf{U}}\widehat{\Sigma}\widehat{\mathbf{V}}^T$ and set $\widehat{\mathbf{A}}^T = \widehat{\mathbf{U}}^{(k)}$, corresponding to k largest singular values.
5: Obtain $\widehat{f}(\mathbf{x}) := \widehat{g}(\widehat{\mathbf{A}}\mathbf{x})$ via quasi interpolants where $\widehat{g}(\mathbf{y}) := f(\widehat{\mathbf{A}}^T\mathbf{y})$.

Theorem 1 *[Sample complexity of Algorithm 1] Let $\delta \in \mathbb{R}^+$, $\rho \ll 1$, and $\kappa < \sqrt{2} - 1$ be fixed constants. Choose*

$$m_\mathcal{X} \geq \frac{2kC_2^2}{\alpha\rho^2}\log(k/p_1),$$

$$m_\Phi \geq \frac{\log(2/p_2) + 4k(d + m_\mathcal{X} + 1)u(\kappa)}{q(\kappa)}, \text{ and}$$

$$\epsilon \leq \frac{\delta}{C_2 k^{5/2} d(\delta + 2C_2\sqrt{2k})}\left(\frac{(1-\rho)m_\Phi\alpha}{(1+\kappa)C_0 m_\mathcal{X}}\right)^{1/2}.$$

Then, given $m = m_\mathcal{X}(m_\Phi + 1)$ samples, our function estimator $\widehat{f}$ in step 5 of Algorithm 1 obeys $\left\|f - \widehat{f}\right\|_{L_\infty} \leq \delta$ with probability at least $1 - p_1 - p_2$.

$$m_\mathcal{X} = \mathcal{O}\left(\frac{k\log k}{\alpha}\right)$$

$$m_\Phi = \mathcal{O}(k(d + m_\mathcal{X}))$$

$$\epsilon = \mathcal{O}\left(\frac{\alpha\delta}{\sqrt{d}}\right)$$

Theorem: proof ingredients

$$m_{\mathcal{X}} = \mathcal{O}\left(\frac{k \log k}{\alpha}\right)$$

$$m_{\Phi} = \mathcal{O}(k(d + m_{\mathcal{X}}))$$

$$\epsilon = \mathcal{O}\left(\frac{\alpha\delta}{\sqrt{d}}\right)$$

- Matrix Danzig selector as running example

$$\widehat{\mathbf{X}}_{DS} = \arg\min_M \|M\|_* \;\; \text{s.t.} \; \|\Phi^*(y - \Phi(M))\| \leq \lambda$$

[Candes and Plan (2010)]

Theorem: proof ingredients

$$m_{\mathcal{X}} = \mathcal{O}\left(\frac{k \log k}{\alpha}\right)$$

$$m_{\Phi} = \mathcal{O}(k(d + m_{\mathcal{X}}))$$

$$\epsilon = \mathcal{O}\left(\frac{\alpha\delta}{\sqrt{d}}\right)$$

- Matrix Danzig selector as running example

$$\widehat{\mathbf{X}}_{DS} = \arg\min_M \|M\|_* \ \text{ s.t. } \|\Phi^*\left(y - \Phi(M)\right)\| \le \lambda$$

- Tuning parameters

Proposition 1 *We have* $\|\varepsilon\|_{\ell_2^{m_\Phi}} \le \frac{C_2 \epsilon d m_{\mathcal{X}} k^2}{2\sqrt{m_\Phi}}$. *Moreover, it holds that* $\|\Phi^*(\varepsilon)\| \le \lambda = \frac{C_2 \epsilon d m_{\mathcal{X}} k^2}{2\sqrt{m_\Phi}}(1+\kappa)^{1/2}$, *with probability at least* $1 - 2e^{-m_\Phi q(\kappa) + (d + m_{\mathcal{X}} + 1)u(\kappa)}$.

Theorem: proof ingredients

$$m_{\mathcal{X}} = \mathcal{O}\left(\frac{k \log k}{\alpha}\right)$$

$$m_{\Phi} = \mathcal{O}(k(d + m_{\mathcal{X}}))$$

$$\epsilon = \mathcal{O}\left(\frac{\alpha\delta}{\sqrt{d}}\right)$$

- Matrix Danzig selector as running example

$$\widehat{\mathbf{X}}_{DS} = \arg\min_M \|M\|_* \;\text{ s.t. } \|\Phi^*(y - \Phi(M))\| \leq \lambda$$

- Tuning parameters

- Recovery guarantees on ***X***

Corollary 1 *Denoting* $\widehat{\mathbf{X}}_{DS}$ *to be the solution of the matrix Danzig selector, if* $\widehat{\mathbf{X}}_{DS}^{(k)}$ *is the best rank-k approximation to* $\widehat{\mathbf{X}}_{DS}$ *in the sense of* $\|\cdot\|_F$*, and if* $\kappa_{4k} < \kappa < \sqrt{2} - 1$*, then we have*

$$\left\|\mathbf{X} - \widehat{\mathbf{X}}_{DS}^{(k)}\right\|_F^2 \leq 4C_0 k\lambda^2 = \frac{C_0 C_2^2 k^5 \epsilon^2 d^2 m_{\mathcal{X}}^2}{m_\Phi}(1+\kappa),$$

with probability at least $1 - 2e^{-m_\Phi q(\kappa) + 4k(d + m_{\mathcal{X}} + 1)u(\kappa)}$.

[Theorem 2.4 from Candes and Plan (2010)]

Theorem: proof ingredients

$$m_{\mathcal{X}} = \mathcal{O}\left(\frac{k \log k}{\alpha}\right)$$

$$m_{\Phi} = \mathcal{O}(k(d + m_{\mathcal{X}}))$$

$$\epsilon = \mathcal{O}\left(\frac{\alpha\delta}{\sqrt{d}}\right)$$

- Matrix Danzig selector as running example

$$\widehat{\mathbf{X}}_{DS} = \arg\min_M \|M\|_* \ \text{ s.t. } \|\Phi^*(y - \Phi(M))\| \le \lambda$$

- Tuning parameters
- Recovery guarantees on ***X***
- Translation of guarantees on ***X*** to guarantees on ***A***

Lemma 1 *For a fixed* $0 < \rho < 1$, $m_{\mathcal{X}} \ge 1$, $m_{\Phi} < m_{\mathcal{X}} d$ *if* $\epsilon < \frac{1}{C_2 k^2 d(\sqrt{k} + \sqrt{2})}\left(\frac{(1-\rho)m_{\Phi}\alpha}{(1+\kappa)C_0 m_{\mathcal{X}}}\right)^{1/2}$, *then with probability at least* $1 - k\exp\left\{-\frac{m_{\mathcal{X}}\alpha\rho^2}{2kC_2^2}\right\} - 2\exp\{-m_{\Phi}q(\kappa) + 4k(d + m_{\mathcal{X}} + 1)u(\kappa)\}$ *we have*

$$\left\|\mathbf{A}\widehat{\mathbf{A}}^T\right\|_F \ge \left(k - \frac{2\tau^2}{(\sqrt{(1-\rho)m_{\mathcal{X}}\alpha} - \tau)^2}\right)^{1/2},$$

where $\tau^2 = \frac{C_0 C_2^2 k^5 \epsilon^2 d^2 m_{\mathcal{X}}^2}{m_{\Phi}}(1+\kappa)$ *is the error bound derived in Corollary 1.*

This is precisely where the restricted Hessian property is used…

Theorem: proof ingredients

$$m_{\mathcal{X}} = \mathcal{O}\left(\frac{k \log k}{\alpha}\right)$$
$$m_{\Phi} = \mathcal{O}(k(d + m_{\mathcal{X}}))$$
$$\epsilon = \mathcal{O}\left(\frac{\alpha\delta}{\sqrt{d}}\right)$$

- Matrix Danzig selector as running example

$$\widehat{\mathbf{X}}_{DS} = \arg\min_M \|M\|_* \quad \text{s.t. } \|\Phi^*\left(y - \Phi(M)\right)\| \leq \lambda$$

- Tuning parameters
- Recovery guarantees on ***X***
- Translation of guarantees on ***X*** to guarantees on ***A***
- Translation of guarantees on ***A*** to guarantees on ***f***

First observe that: $\widehat{f}(\mathbf{x}) = f(\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x}) = g(\mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x})$.

$$\Rightarrow \left|f(\mathbf{x}) - \widehat{f}(\mathbf{x})\right| = \left|g(\mathbf{A}\mathbf{x}) - g(\mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{x})\right| \leq C_2\sqrt{k}\left\|(\mathbf{A} - \mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}})\mathbf{x}\right\|_{\ell_2^k} \leq C_2\sqrt{k}\left\|\mathbf{A} - \mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\right\|_F \|\mathbf{x}\|_{\ell_2^d}.$$

Now it is easy to verify that:

$$\left\|\mathbf{A} - \mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\right\|_F^2 = \mathrm{Tr}((\mathbf{A}^T - \widehat{\mathbf{A}}^T\widehat{\mathbf{A}}\mathbf{A}^T)(\mathbf{A} - \mathbf{A}\widehat{\mathbf{A}}^T\widehat{\mathbf{A}})) = k - \left\|\mathbf{A}\widehat{\mathbf{A}}^T\right\|_F^2.$$

Impact of noisy queries

Algorithm 1 Estimating $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$

1: Choose m_Φ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\mathbf{\Phi}$.
2: Choose ϵ and construct $\mathbf{y}$ using $y_i = \sum_{j=1}^{m_\mathcal{X}} \left[\frac{f(\xi_j + \epsilon \phi_{i,j}) - f(\xi_j)}{\epsilon} \right]$.
3: Obtain $\widehat{\mathbf{X}}$ via a stable low-rank recovery algorithm.
4: Compute $\mathrm{SVD}(\widehat{\mathbf{X}}) = \widehat{\mathbf{U}}\widehat{\Sigma}\widehat{\mathbf{V}}^T$ and set $\widehat{\mathbf{A}}^T = \widehat{\mathbf{U}}^{(k)}$, corresponding to k largest singular values.
5: Obtain $\widehat{f}(\mathbf{x}) := \widehat{g}(\widehat{\mathbf{A}}\mathbf{x})$ via quasi interpolants where $\widehat{g}(\mathbf{y}) := f(\widehat{\mathbf{A}}^T\mathbf{y})$.

- Assume evaluation of ***f*** yields $f(\mathbf{x}) + Z$, where $Z \sim \mathcal{N}(0, \sigma^2)$

Impact of noisy queries

Algorithm 1 Estimating $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$

1: Choose m_Φ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\mathbf{\Phi}$.
2: Choose ϵ and construct $\mathbf{y}$ using $y_i = \sum_{j=1}^{m_\mathcal{X}} \left[\frac{f(\xi_j+\epsilon\phi_{i,j})-f(\xi_j)}{\epsilon}\right]$.
3: Obtain $\widehat{\mathbf{X}}$ via a stable low-rank recovery algorithm.
4: Compute $\text{SVD}(\widehat{\mathbf{X}}) = \widehat{\mathbf{U}}\widehat{\Sigma}\widehat{\mathbf{V}}^T$ and set $\widehat{\mathbf{A}}^T = \widehat{\mathbf{U}}^{(k)}$, corresponding to k largest singular values.
5: Obtain $\widehat{f}(\mathbf{x}) := \widehat{g}(\widehat{\mathbf{A}}\mathbf{x})$ via quasi interpolants where $\widehat{g}(\mathbf{y}) := f(\widehat{\mathbf{A}}^T\mathbf{y})$.

- Assume evaluation of ***f*** yields $f(\mathbf{x}) + Z$, where $Z \sim \mathcal{N}(0, \sigma^2)$

tuning parameter changes:

$$\|\Phi^*(\varepsilon + \mathbf{z})\| \le \frac{2\gamma\sigma}{\epsilon}\sqrt{2(1+\kappa)m_\mathcal{X}m_\Phi} + \frac{C_2\epsilon d m_\mathcal{X}k^2}{2\sqrt{m_\Phi}}(1+\kappa)^{1/2}, \quad (\gamma > 2\sqrt{\log 12}).$$

Impact of noisy queries

Algorithm 1 Estimating $f(\mathbf{x}) = g(\mathbf{A}\mathbf{x})$

1: Choose m_Φ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\mathbf{\Phi}$.
2: Choose ϵ and construct $\mathbf{y}$ using $y_i = \sum_{j=1}^{m_\mathcal{X}} \left[\frac{f(\xi_j+\epsilon\phi_{i,j})-f(\xi_j)}{\epsilon}\right]$.
3: Obtain $\widehat{\mathbf{X}}$ via a stable low-rank recovery algorithm.
4: Compute $\mathrm{SVD}(\widehat{\mathbf{X}}) = \widehat{\mathbf{U}}\widehat{\Sigma}\widehat{\mathbf{V}}^T$ and set $\widehat{\mathbf{A}}^T = \widehat{\mathbf{U}}^{(k)}$, corresponding to k largest singular values.
5: Obtain $\widehat{f}(\mathbf{x}) := \widehat{g}(\widehat{\mathbf{A}}\mathbf{x})$ via quasi interpolants where $\widehat{g}(\mathbf{y}) := f(\widehat{\mathbf{A}}^T\mathbf{y})$.

- Assume evaluation of ***f*** yields $f(\mathbf{x}) + Z$, where $Z \sim \mathcal{N}(0, \sigma^2)$

tuning parameter changes:

$$\|\Phi^*(\varepsilon + \mathbf{z})\| \le \frac{2\gamma\sigma}{\epsilon}\sqrt{2(1+\kappa)m_\mathcal{X}m_\Phi} + \frac{C_2\epsilon d m_\mathcal{X} k^2}{2\sqrt{m_\Phi}}(1+\kappa)^{1/2}, \quad (\gamma > 2\sqrt{\log 12}).$$

$$\Rightarrow m = \mathcal{O}\left(\frac{\sqrt{d}}{\alpha}\right) m_\mathcal{X}(m_\Phi + 1)$$

We resample the same data points $\mathcal{O}(\epsilon^{-1})$-times and average.

Learning a logistic function

$$f(\mathbf{x}) = g(\mathbf{a}^T\mathbf{x}), \text{ where } g(y) = \frac{1}{1+e^{-y}}$$

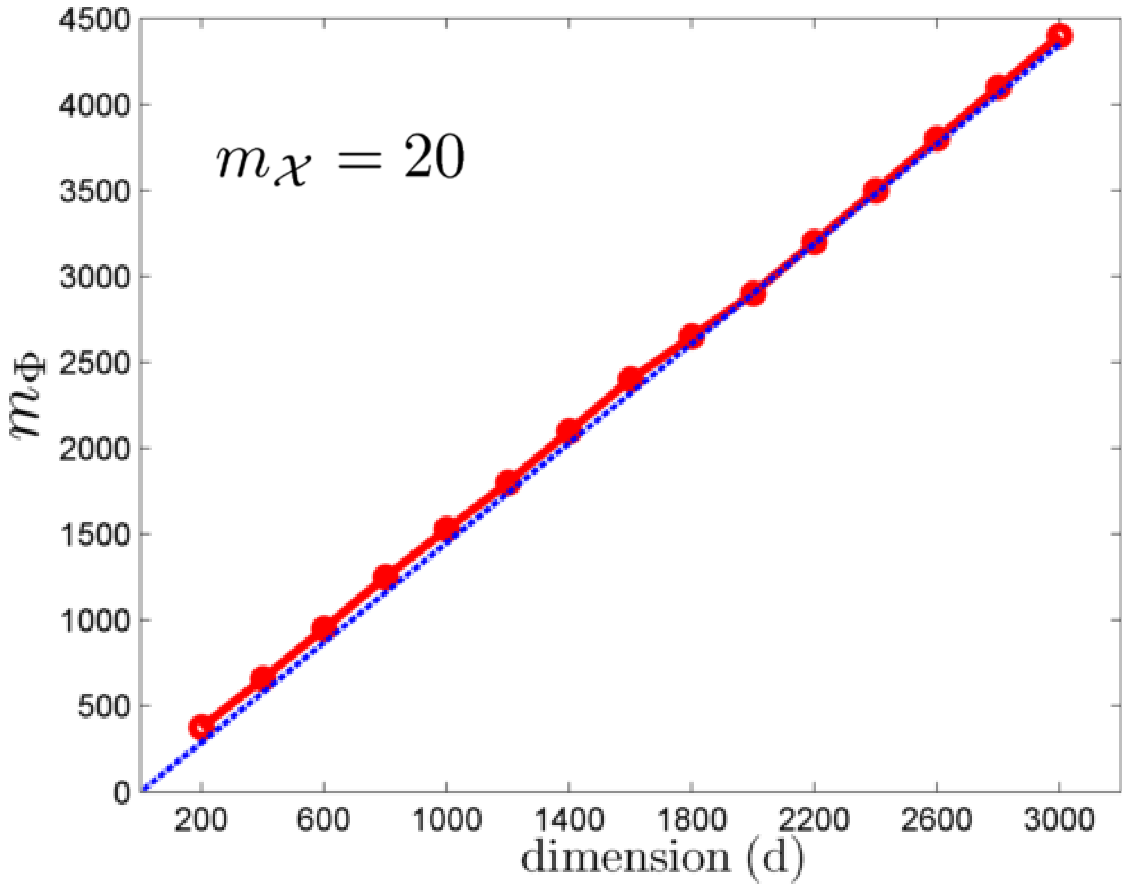
$$\alpha = \int \left|g'(\mathbf{a}^T\mathbf{x})\right|^2 d\mu_{\mathbb{S}^{d-1}} \approx |g'(0)|^2 = (1/16)$$

$$C_2 = \sup_{|\beta|\leq 2} \left|g^{(\beta)}(y)\right| = 1$$

- Declare success if

$$|\langle \hat{\mathbf{a}}, \mathbf{a}\rangle| \geq 0.99$$

theory: $m_\Phi = \mathcal{O}(d)$
practice: $m_\Phi = 1.45d$



Learning sum of Gaussian functions

$$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x} + \mathbf{b}) = \sum_{i=1}^{k} g_i(a_i^T \mathbf{x} + b_i)$$

$d = 100$
$\epsilon = 10^{-3}$
$m_{\mathcal{X}} = 100$

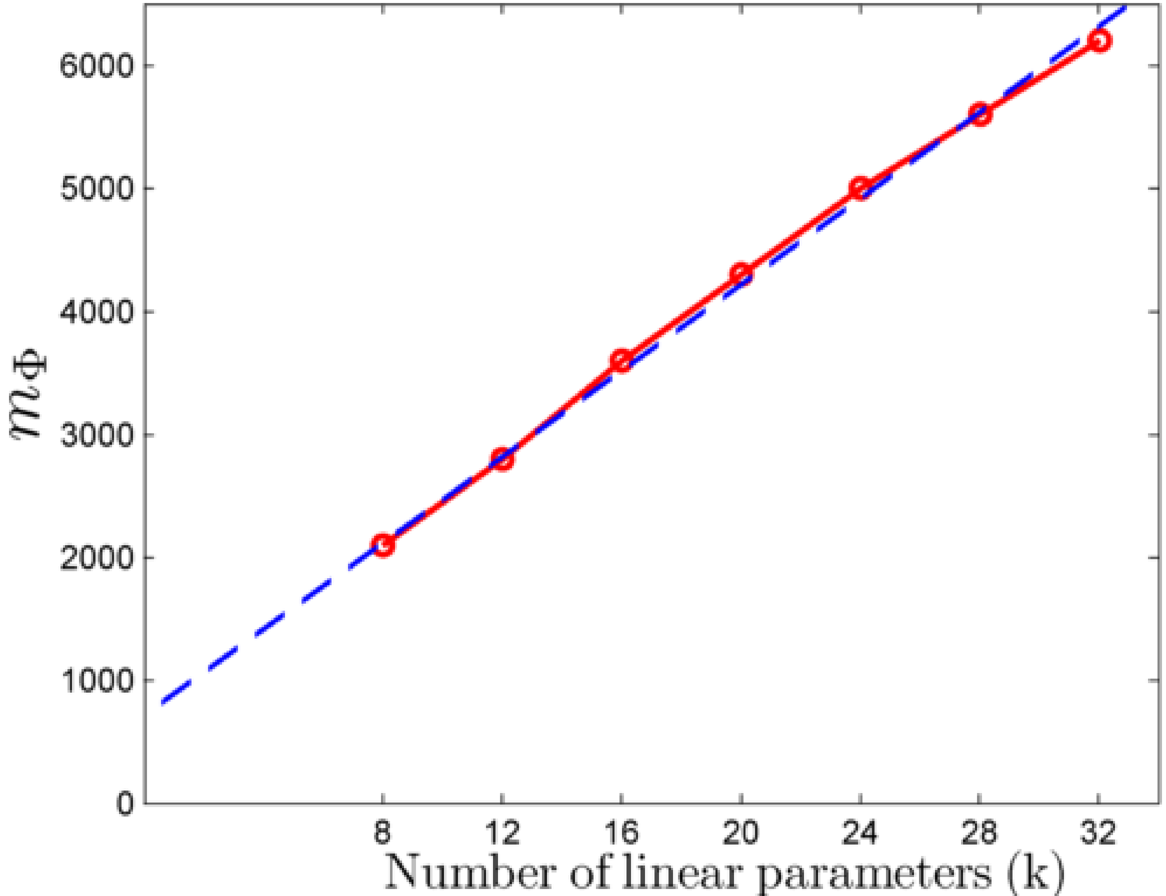
$$g_i(y) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y+b_i)^2}{2\sigma_i^2}\right)$$

- Declare success if

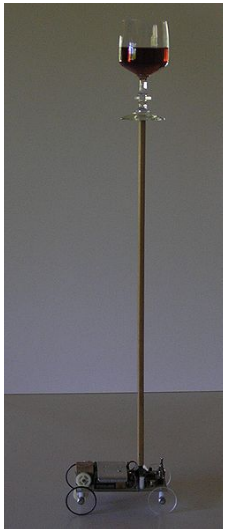
$$\frac{1}{k}\left\|\mathbf{A}\widehat{\mathbf{A}}^T\right\|_F^2 \geq 0.99$$

$\sigma \sim \mathcal{U}[0.1, 0.5]$
$b_i \sim \mathcal{U}(0.2\mathbb{S}^{k-1})$

theory: $m_\Phi = \mathcal{O}(d)$



Stability example with the quadratic



$$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x}) = \|\mathbf{A}\mathbf{x} - b\|^2$$

$$\tilde{f}(\mathbf{x}) = f(\mathbf{x}) + \sigma\mathcal{N}(0,1)$$

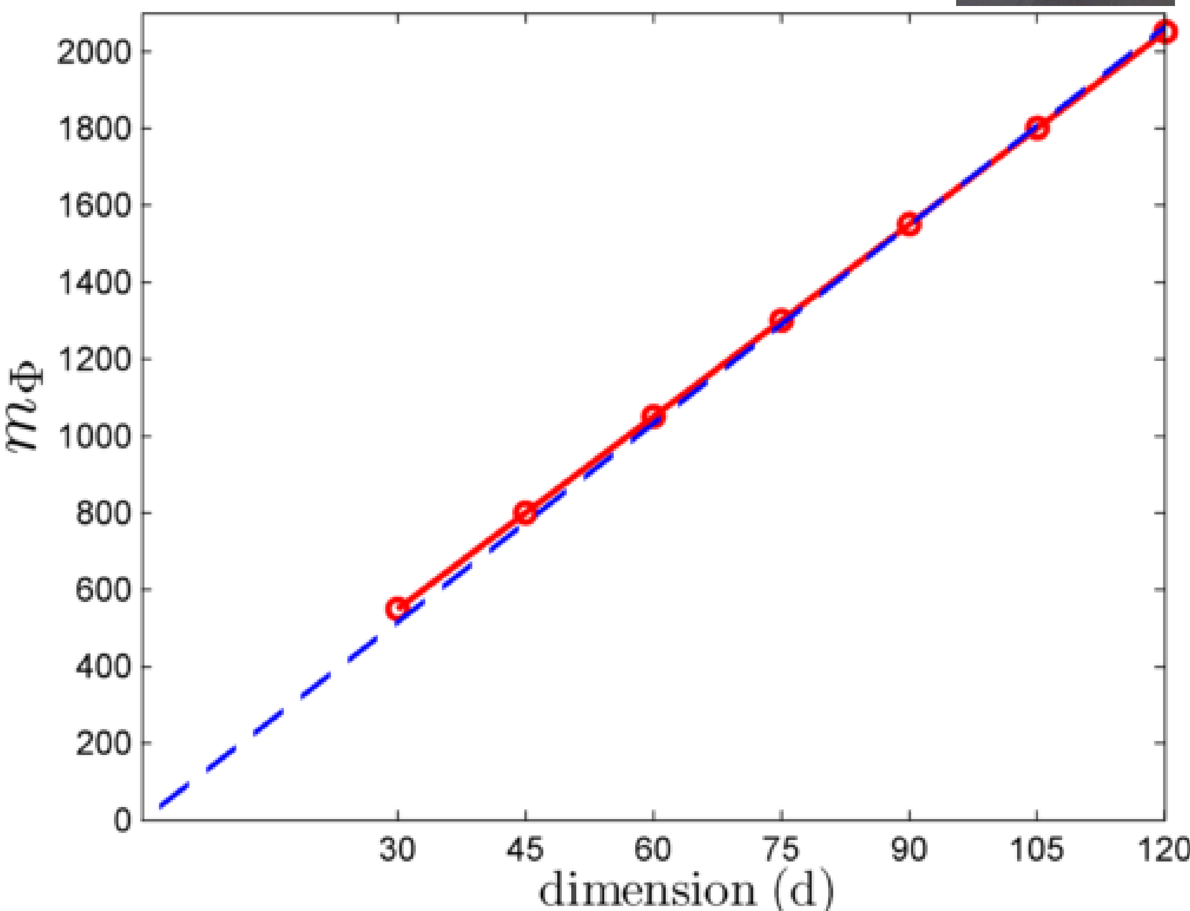
$$k = 5$$
$$\epsilon = 10^{-1}$$
$$m_{\mathcal{X}} = 30$$
$$\sigma = 0.01$$

- Declare success if

$$\frac{1}{k}\left\|\mathbf{A}\widehat{\mathbf{A}}^T\right\|_F^2 \geq 0.99$$

$$\text{theory: } \frac{\tilde{m}_\Phi}{d^{3/2}} = \mathcal{O}(d)$$

$$b_i \sim \mathcal{U}(\mathbb{S}^{k-1})$$



Conclusions

- Main focus <> estimation of low-dim subspace for dimensionality reduction

  *learning/optimizing **f** for later*

  model building, cluster analysis, variable selection...

- Active setting polynomial time samples/scheme

  *a new link between old **low-rank** models with new **low-rank** algorithms*

- New tools <> L-Lipschitz 2nd order derivative matrix ALPS for low-rank recov.

  beyond linear models

  system calibration, PDE models, matrix compression...