Winter Conference in Statistics 2013

Compressed Sensing

LECTURE #12
Nonparametric function learning

Prof. Dr. Volkan Cevher
volkan.cevher@epfl.ch
LIONS/Laboratory for Information and Inference Systems
Function learning

- A fundamental problem:
  given $(y_i, x_i) : \mathbb{R} \times \mathbb{R}^d, i = 1, \ldots, m$, learn a mapping $f: x \rightarrow y$
  - some call it “regression”

- Oft-times $f \leftrightarrow$ parametric form
e.g., linear regression

  learning the model
  $= \quad$ 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 \(f: x \rightarrow y\)
  
  - some call it “regression”

- Oft-times \(f\) \(<>\) parametric form
  
  e.g., linear regression

\[
\begin{align*}
    &\text{learning the model} \\
    &\text{= learning the parameters}
\end{align*}
\]

\[
f(x) = a^t x
\]

familiar challenge: \textit{learning via dimensionality reduction}
Function learning

• A fundamental problem:

\[ \text{given } (y_i, x_i) : \mathbb{R} \times \mathbb{R}^d, i = 1, \ldots, m, \text{ learn a mapping } f: x \rightarrow y \]

— some call it "regression"

• Oft-times \( f \leftrightarrow \) parametric form
e.g., linear regression

learning a \textit{low-dimensional} model
\( = \)
\textit{successful} learning the parameters

\[ f(x) = a^t x \]

familiar challenge: \textit{learning via dimensionality reduction}
A fundamental problem:

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

— some call it “regression”

Oft-times \(f\) <> parametric form
e.g., linear regression

low-dim models \(\gg\) successful learning

\(\text{sparse, low-rank...}\)

Any parametric form <> at best an approximation

emerging alternative: \(\text{non-parametric models}\)

learn \(f\) from data!
Function learning

• A fundamental problem:
  
  given \((y_i, x_i): \mathbb{R} \times \mathbb{R}^d, i = 1, \ldots, m,\) learn a mapping \(f: x \to y\)
  
  — some call it “regression”

• Oft-times \(f\) <> parametric form
  
  e.g., linear regression
  
  low-dim models \(\gg\) successful learning

  *sparse, low-rank*

• Any parametric form \(<>\) at best an approximation
  
  emerging alternative: \textit{non-parametric models}

\textit{this lecture} \(\Rightarrow\)

learn \textbf{low-dim} \(f\) from data!
Nonparametric model learning

Two distinct camps:

1. Regression

   <> use given samples

   \emph{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

   \emph{approximation of f}

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

   \emph{maximization/optimization of f}

   [Srinivas, Krause, Kakade, Seeger, 2012]
Nonparametric model learning—our contributions

Two distinct camps:

1. Regression
   <> use given samples
   \( \text{approximation of } f \)
   
   [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
   \( \text{approximation of } f \)
   
   [Cohen et al., 2010; Fornasier, Schnass, Vybiral, 2011; VC and Tyagi 2012; Tyagi and VC 2012]

\[ \text{maximization/optimization of } f \]

[Srinivas, Krause, Kakade, Seeger, 2012]
Active function learning

- A motivation by Albert Cohen

Numerical solution of parametric PDE’s

\[ \text{PDE}(f, x) = 0 \rightarrow f(x): \text{the (implicit) solution} \]

query of the solution \(<>\) running an expensive simulation
Active function learning

- A motivation by Albert Cohen

Numerical solution of parametric PDE’s

\[ \text{PDE}(f, x) = 0 \quad \rightarrow \quad f(x) : \text{the (implicit) solution} \]

query of the solution $\leftrightarrow$ 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

\[ \text{PDE}(f, x) = 0 \quad \longrightarrow \quad f(x) : \text{the (implicit) solution} \]

query of the solution \( <> \) running an expensive simulation

ability to choose the samples \( <> \) active learning
Learning via interpolation
Learning via interpolation

\[ f(x) \]

\[ h \leftrightarrow \]

\[ x \in [0, 1] \]
Learning via interpolation

\[ R(f) : \text{reconstruction via, e.g., linear interpolation} \]

\[ f \in C^s \]

\[ \| f - R(f) \|_\infty \leq C \| D^s f \|_\infty h^s \]
Learning via interpolation

Error characterization for smooth $f \in C^s$

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

number of samples $N = \mathcal{O}(h^{-1})$ \iff $\| f - R(f) \|_\infty = \mathcal{O}(N^{-s})$
Learning via interpolation

Curse-of-dimensionality

\[ f(\ldots, x_i, \ldots) \]

\[ R(f) \]

\[ 0 \leq h \leq 1 \]

\[ x_i \in [0, 1] \]

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

- 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 \( N = \mathcal{O}(h^{-d}) \) \( \iff \) \( \| f - R(f) \|_\infty = \mathcal{O}(N^{-s/d}) \)
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 \( \mathbb{R}^N \to \Omega \)
\( \Omega: \) compact set

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

\[ \Omega = C^s([0, 1]^d) \Rightarrow cN^{-s/d} \leq d_N(\Omega) \leq C N^{-s/d} \]

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

Curse-of-dimensionality

• The nonlinear N-width

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

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

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

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

Curse-of-dimensionality

- The nonlinear N-width

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

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

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

\[ \Omega = C^\infty([0, 1]^d) \Rightarrow \min \{ N : d_N(\Omega) \leq 0.5 \} \geq c 2^{d/2} \]

- Take home message

smoothness-only >> intractability in sample complexity

need additional assumptions on the problem structure!!!
Learning **multi-ridge functions**

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

Model 1:
\[ f(x) = g(\mathbf{Ax}) \quad k < d \]

Model 2:
\[ f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x) \]

\[ f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R} \quad \mathbf{A} = [a_1, \ldots, a_k]^T \]

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* $\leftrightarrow$ *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**  $\leftrightarrow$ *first order* low-rank model

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

  $$f(x) = g(Ax)$$

  1. assume orthogonality

  $$AA^T = I_k$$

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

  SVD of $A$

  $$f(x) = g(U\Sigma V^T x) = \tilde{g}(V^T x),$$

  where $\tilde{g}(y) = g(U\Sigma y)$
Prior work—Regression camp

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

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

\[ f(x) = g(Ax) \]

1. assume orthogonality
\[ AA^T = I_k \]

2. note the differentiability of \( f \)
\[ \nabla f(x) = A^T \nabla g(Ax) \]

[\text{SVD of } A]

\[ f(x) = g(U\Sigma V^T x) = \bar{g}(V^T x), \text{ where } \bar{g}(y) = g(U\Sigma y) \]

**Key observation #1:**

gradients live in at most \( k \)-dim. subspaces

[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)

  \[ f(x) = g(Ax) \]

  1. assume orthogonality

  \[ AA^T = I_k \]

  2. note the differentiability of \( f \)

  \[ \nabla f(x) = A^T \nabla g(Ax) \]

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

  \[ Hf := A^T Hg A \]

  required: rank-k \( Hg \)

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

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

  SVD of \( A \)

  \[ f(x) = g(U\Sigma V^T x) = \bar{g}(V^T x), \]

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

  **Key observation #1:**
  gradients live in at most \( k \)-dim. subspaces

  **Key observation #2:**
  \( k \)- principal components of \( Hf \) leads to \( A \)
Prior work—Regression camp

- **local smoothing** \(\leftrightarrow\) **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** \(\leftrightarrow\) **additive sparse models**

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

  \[
  f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x)
  \]

  - encode **smoothness** via the kernel
  
  - leverage sparse greedy/convex optimization
  
  - establish consistency rates:
    \[
    \| f - \hat{f} \|_{L_2} \leq O \left( k\delta^2 + \frac{k \log(d)}{m} \right)
    \]
Prior work—Regression camp

- local smoothing \(\rightarrow\) 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 \(\rightarrow\) additive sparse models

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

  \(g\) belongs to reproducing kernel Hilbert space

  - encode smoothness via the kernel

  - leverage sparse greedy/convex optimization

  - establish consistency rates: \(\| f - \hat{f} \|_{L_2} \leq O\left(k\delta^2 + \frac{k \log(d)}{m}\right)\)

  difficulty of estimating the kernel

  difficulty of subset selection

[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]
Prior work—Active learning camp

Progress thus far \[ \leftrightarrow \] \textit{the sparse way}

\textit{highlights:}


\[ f(x) = g(a^T x) \]

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

\[ a \succeq 0, \mathbf{1}^T a = 1 \quad a \in \ell_q \quad q < 1 \quad \text{(i.e., compressible)} \]
Prior work—Active learning camp

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

*highlights:*


\[ f(x) = g(a^T x) \]

\( g : [0, 1] \rightarrow \mathbb{R} \) is a \( C^s \) function for \( s > 1 \)

\( a \geq 0, 1^T a = 1 \quad a \in w\ell_q \quad q < 1 \)  
(i.e., compressible)

2. Fornassier, Schnass, and Vybiral (2011)

\[ f(x) = g(Ax) \]

\( g : B_{\mathbb{R}^d}(1 + \epsilon) \rightarrow \mathbb{R} \) is \( C^s \)  
\( a_i \in w\ell_q, q < 2 \quad A = [a_1, \ldots, a_k]^T \)

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

\[ f(x + \epsilon\phi) = f(x) + \epsilon \langle \phi, \nabla f(x) \rangle + \epsilon E(x, \epsilon, \phi) \quad \epsilon \ll 1 \]

\[ \Rightarrow \langle \phi, A^T \nabla g(Ax) \rangle = \frac{1}{\epsilon} (f(x + \epsilon\phi) - f(x)) - E(x, \epsilon, \phi) \]
Prior work—Active learning camp (FSV’11)

- A \textit{sparse} observation model

\[ f(x) = g(Ax) \]

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

\[ E(x, \epsilon, \phi) = \frac{\epsilon}{2} \phi^T \nabla^2 f(\zeta(x, \phi)) \phi \]

\[ \zeta(x, \phi) \in [x, x + \epsilon \phi] \]

\[ \xi_j \in S^{d-1} \]

with two ingredients

\textbf{sampling centers} \quad \mathcal{X} = \{\xi_j \in S^{d-1}; j = 1, \ldots, m_X\}

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

\textbf{leads to}

\[ y = \Phi(X) + E(\mathcal{X}, \epsilon, \Phi) \]

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

\[ G := [\nabla g(A\xi_1) | \nabla g(A\xi_2) | \cdots | \nabla g(A\xi_{m_X}) ]_{k \times m_X} \]

\[ X_i := A^T G_i \]

approximately sparse
Prior work—Active learning camp (FSV’11)

- A sparse observation model

\[ y = \Phi(X) + E(\mathcal{X}, \epsilon, \Phi) \]

- Key contribution: restricted “Hessian” property

\[
H^f := \int_{S^{d-1}} \nabla f(x) \nabla f(x)^T d\mu_{S^{d-1}}(x) \quad \mu: \text{uniform measure}
\]

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

recall \( G \) needs to span a k-dim subspace for identifiability of \( A \)

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

with a restricted study of radial functions \( f(x) = g_0(\|Ax\|_2) \)

- Analysis \( <> \) leverage compressive sensing results
Prior work—Active learning camp (FSV’11)

- A sparse observation model
  \[ f(x) = g(Ax) \]
  \[ y = \Phi(X) + E(X, \epsilon, \Phi) \]
  \[ X_i := A^T G_i \]
  approximately sparse

- Analysis \(<>\) leverage compressive sensing results

- Key contribution: restricted Hessian property for radial functions
  \[ f(x) = g_0(\|Ax\|_2) \]

- Two major issues remains to be addressed over FSV’11
  1. validity of orthogonal sparse/compressible directions
     \textit{need a basis independent model}
     \[ f(x) = g(A\Psi^T \Psi x) = g(A\Psi x) \]
     one \(\Psi\) for all orthogonal directions?
  2. analysis of \(H^f\) for anything other than radial functions
     \textit{need a new analysis tool}
     \[ H^f := \int_{S_{d-1}} \nabla f(x) \nabla f(x)^T \, d\mu_{S_{d-1}}(x) \]
Learning **multi-ridge functions**

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

  - **Model 1:**
    \[
    f(x) = g(Ax)
    \]
    \[k < d\]

  - **Model 2:**
    \[f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x)\]
    \[f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R} \quad A = [a_1, \ldots, a_k]^T\]

- **Results:**
  
  \[\text{w.l.o.g. } g, g_i \in C^2\]
  \[A: \text{ compressible}\]

  \[(\text{Model 1}): \quad m = O\left(\left(\frac{1}{\epsilon}\right)^{k/2} + \frac{\frac{4}{d^2-q}d^{2-q}}{k \alpha \log(k)}\right) \implies \|f - \hat{f}\|_{L_\infty} \leq \epsilon\]

  *if \(g\) has \(k\)-restricted Hessian property...

[Fornasier, Schnass, Vybiral, 2011]
Learning multi-ridge functions

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

   Model 1: \[ f(x) = g(Ax) \quad k < d \]

   Model 2: \[ f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x) \]

\[ f : B_{\mathbb{R}^d}(1 + \bar{\varepsilon}) \rightarrow \mathbb{R} \quad A = [a_1, \ldots, a_k]^T \]

• **Results:**

   cost of learning \( g \)

   \[ m = \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{k/2} + \frac{4-a}{k} \frac{d}{2-a} \frac{a}{\alpha} \log(k)\right) \Rightarrow \| f - \hat{f} \|_{L_\infty} \leq \varepsilon \]

   *cost of learning \( A \)

   w.l.o.g. \( g, g_i \in C^2 \)

   \( A \): compressible

   [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:  
  \[ f(x) = g(Ax) \]
  \[ k < d \]

  Model 2:  
  \[ f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x) \]

  \[ f: B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \rightarrow \mathbb{R} \quad A = [a_1, \ldots, a_k]^T \]

- **Results:**
  - cost of learning \( g \)
    \[ m = \mathcal{O} \left( \left( \frac{1}{\bar{\epsilon}} \right)^{k/2} + k \frac{4-q}{2-q} d^{2-q} \log(k) \right) \Rightarrow \|f - \hat{f}\|_{L_\infty} \leq \epsilon \]
  - w.l.o.g. \( g, g_i \in C^2 \)
  - \( A \): compressible
  - only for radial basis functions
    \[ f(x) = g_0(\|Ax\|_2) \]
  - \( \alpha = \Theta \left( \frac{1}{d} \right) \)
  - *cost of learning \( A \)
  - *if \( f \) has k-restricted Hessian property...

[Forasier, Schnass, Vybiral, 2011]
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:** \( f(x) = g(Ax) \) \( k < d \)
  - **Model 2:** \( f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x) \)

\[ f : B_{\mathbb{R}^d}(1 + \epsilon) \rightarrow \mathbb{R} \quad A = [a_1, \ldots, a_k]^T \]

- **Results:**
  - cost of learning \( g \)
  - w.l.o.g. \( g, g_i \in \mathcal{C}^2 \)
  - \( A \): compressible

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

  - *cost of learning \( A \)
  - \( \alpha = \Theta\left( \frac{1}{d} \right) \)

  - *with the L-Lipschitz property...*

Our 1st contribution:
a simple verifiable characterization of alpha for a broad set of functions
Learning **multi-ridge** functions: the low-rank way

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

Model 1:

\[ f(x) = g(Ax) \quad k < d \]

Model 2:

\[ f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x) \]

\[ f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R} \quad A = [a_1, \ldots, a_k]^T \]

• **Results:**

- cost of learning \( g \)
- w.l.o.g. \( g, g_i \in C^2 \)

(Model 1):

\[ m = O\left( \left(\frac{1}{\bar{\epsilon}}\right)^{-k/2} + \frac{k \log(k)}{\alpha} \times kd \right) \Rightarrow \|f - \hat{f}\|_{L_\infty} \leq \varepsilon \]

our 2\textsuperscript{nd} contribution: extension to the general \( A \)

*cost of learning \( A \)

*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:
  \[ f(x) = g(Ax) \]

  Model 2:
  \[ f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x) \]

  \[ f : B_{\mathbb{R}^d}(1 + \varepsilon) \to \mathbb{R} \quad A = [a_1, \ldots, a_k]^T \]

- **Results:** cost of learning \( g_i \)'s

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

  **our 2\textsuperscript{nd} contribution:** extension to the general \( A \)

  *cost of learning \( A \)

  *with the L-Lipschitz property...*
Learning multi-ridge functions: the low-rank way

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

  \[
  f(x) = g(Ax)
  \quad k < d
  \]

  Model 1:

  \[
  f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x)
  \]

  Model 2:

  \[(AR)(AR)^T = I_k\]

  just kidding.

- **Results:**

  cost of learning \(g_i\)'s

  w.l.o.g. \(g, g_i \in C^2\)

  (Model 2):

  \[
  m = \mathcal{O}\left(\left(\frac{1}{\epsilon}\right)^k \frac{k \log(k)}{\alpha} \times kd\right) \Rightarrow \|f - \hat{f}\|_{L_\infty} \leq \epsilon
  \]

  our 2\textsuperscript{nd} contribution: extension to the general \(A\)

  *cost of learning \(A\)

  *with the L-Lipschitz property...
Learning **multi-ridge** functions: the low-rank way

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

  **Model 1:**
  \[ f(x) = g(Ax) \]
  \[ k < d \]

  **Model 2:**
  \[ f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x) \]

  in general
  \[ f : B_{\mathbb{R}^d}(1 + \epsilon) \rightarrow \mathbb{R} \]
  \[ A = [a_1, \ldots, a_k]^T \]

- **Results:** cost of learning \( g / g_i \)'s w.l.o.g. \( g, g_i \in \mathcal{C}^2 \)

  (Model 1&2):
  \[ m = O \left( \left( \frac{1}{\epsilon} \right)^{k/2} + k^2 d^2 \log(k) \right) \Rightarrow \| f - \hat{f} \|_{L_{\infty}} \leq \epsilon \]

Given 1\textsuperscript{st} and 2\textsuperscript{nd} contribution:

**full characterization of Model 1 & 2 with minimal assumptions**

*cost of learning \( A \)

*with the L-Lipschitz property...
Learning multi-ridge functions

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

  Model 1:
  \[ f(x) = g(Ax) \quad k < d \]

  Model 2:
  \[ f(x_1, \ldots, x_d) = \sum_{i=1}^{k} g_i(a_i^T x) \]

  \[ f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R} \quad A = [a_1, \ldots, a_k]^T \]

- **Results:**
  - cost of learning \( g / g_i \)'s with l.o.g. \( g, g_i \in \mathcal{C}^2 \)
  - (Model 1\&2):
    \[ m = \mathcal{O}\left( \left(\frac{1}{\bar{\epsilon}}\right)^{k/2} + k^2 d^{4.5} \log(k) \right) \Rightarrow \|f - \hat{f}\|_{L_\infty} \leq \epsilon \]

  **Our 3\textsuperscript{rd} contribution:**
  - impact of iid noise \( f + Z \)
  - *cost of learning \( A \)
  - *with the L-Lipschitz property...
Non-sparse directions $A$

- A **low-rank** observation model

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

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
  $$\Phi_j = [\phi_{1,j} \mid \ldots \mid \phi_{m_\Phi,j}]^T$$

leads to

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

$$X := A^T k \times m_\mathcal{X}$$

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

$$G := [\nabla g(A\xi_1) | \nabla g(A\xi_2) | \cdots | \nabla g(A\xi_{m_\mathcal{X}})]_{k \times m_\mathcal{X}}$$
Detour #2: low-rank recovery

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

\[ \Phi : \mathbb{R}^{d \times m_X} \rightarrow \mathbb{R}^{m_\Phi} \]

- Stable recovery \iff measurements commensurate with degrees of freedom

- stable recovery: \[ \| \mathbf{X} - \hat{\mathbf{X}} \|_F \leq C_1 \| \mathbf{X} - \mathbf{X}_k \|_F + C_2 \| E \|_F \]

- measurements: \[ m_\Phi = \mathcal{O}(k(d + m_X - k)) \]

\[ \hat{\mathbf{X}} = \Delta(\mathbf{y}, \Phi) : \text{decoder} \]

\[ \mathbf{X}_k = \arg \min_{\mathbf{Z}: \text{rank}(\mathbf{Z}) \leq k} \| \mathbf{X} - \mathbf{Z} \|_F \]
Detour #2: low-rank recovery

\[
y = \Phi(X) + E(\lambda, \epsilon, \Phi)
\]

\[\Phi : \mathbb{R}^{d \times m \lambda} \rightarrow \mathbb{R}^{m \Phi}\]

- Stable recovery \(<>\) measurements commensurate with degrees of freedom

- stable recovery:
  \[\|X - \hat{X}\|_F \leq C_1\|X - X_k\|_F + C_2\|E\|_F\]

- measurements:
  \[m_\Phi = \mathcal{O}(k(d + m\lambda - k))\]

- 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)]

Matrix ALPS
http://lions.epfl.ch/MALPS
Detour #2: low-rank recovery

\[ y = \Phi(X) + E(\mathcal{X}, \epsilon, \Phi) \]

- Stable recovery <-> measurements commensurate with degrees of freedom
- Convex/non-convex decoders <-> sampling/noise type
  - affine rank minimization
  - matrix completion
  - robust principal component analysis

Matrix restricted isometry property (RIP):

\[
(1 - \kappa_k) \leq \frac{\|\Phi X\|_F^2}{\|X\|_F^2} \leq (1 + \kappa_k), \quad \forall X: \text{rank}(X) \leq k
\]

- affine rank minimization
- matrix completion
- robust principal component analysis

Matrix ALPS

http://lions.epfl.ch/MALPS

[Plan 2011]

[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

\[ y = \Phi(X) + E(\mathcal{X}, \epsilon, \Phi) \]

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

- Matrix RIP \( \iff \) uniform sampling on the sphere

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

\[ \Rightarrow 0 < \kappa_T < \kappa < 1 \text{ with probability } 1 - 2e^{-m_\Phi q(\kappa) + r(d + m_x + 1)u(\kappa)}, \text{ where } q(\kappa) = \frac{1}{144} \left( \kappa^2 - \frac{\kappa^3}{9} \right) \text{ and } u(\kappa) = \log \left( \frac{36 \sqrt{2}}{\kappa} \right) \]

[Candès and Plan (2010)]
Here it is... our low-rank approach

Algorithm 1 Estimating $f(x) = g(Ax)$

1. Choose $m_\Phi$ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\Phi$.
2. Choose $\epsilon$ and construct $y$ 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 $\hat{X}$ via a stable low-rank recovery algorithm.
4. Compute $\text{SVD}(\hat{X}) = \hat{U}\hat{\Sigma}\hat{V}^T$ and set $\hat{A}^T = \hat{U}^{(k)}$, corresponding to $k$ largest singular values.
5. Obtain $\hat{f}(x) := \hat{g}(\hat{A}x)$ via quasi interpolants where $\hat{g}(y) := f(\hat{A}^Ty)$.

• achieve/balance three objectives simultaneously

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

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

$$X := \Lambda^T \mathbf{G}$$
Here it is... our low-rank approach

Algorithm 1 Estimating $f(x) = g(Ax)$

1. Choose $m_\Phi$ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\Phi$.
2. Choose $\epsilon$ and construct $y$ 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 $\hat{X}$ via a stable low-rank recovery algorithm.
4. Compute $\text{SVD}(\hat{X}) = \hat{U}\Sigma\hat{V}^T$ and set $\hat{A}^T = \hat{U}^{(k)}$, corresponding to $k$ largest singular values.
5. Obtain $\hat{f}(x) := \hat{g}(\hat{A}x)$ via quasi interpolants where $\hat{g}(y) := f(\hat{A}^Ty)$.

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

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

[VC and Tyagi 2012; Tyagi and VC, 2012]
L-Lipschitz property

- **New objective:** approximate $A$ via point queries of $f$
  
  $$f : B_{\mathbb{R}^d}(1 + \bar{e}) \to \mathbb{R} \quad A = [a_1, \ldots, a_k]^T$$

- **New analysis tool:** $L$-Lipschitz 2nd order derivative
  
  Recall:
  
  $$H^f := \int_{S^{d-1}} \nabla f(x) \nabla f(x)^T d\mu_{S^{d-1}}(x) \quad \sigma_k(H^f) \geq \alpha > 0$$

  \[
  \frac{\partial^2 g}{\partial y_i \partial y_j}(y_1) - \frac{\partial^2 g}{\partial y_i \partial y_j}(y_2) \leq L_{i,j} \left\| y_1 - y_2 \right\|_{l_2}^k
  \]

  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 $f$
  \[ f : B_{\mathbb{R}^d}(1 + \bar{\epsilon}) \to \mathbb{R} \quad A = [a_1, \ldots, a_k]^T \]

- **New analysis tool:** $L$-Lipschitz 2$^{\text{nd}}$ order derivative
  \[ H^f := \int_{S^{d-1}} \nabla f(x) \nabla f(x)^T d\mu_{S^{d-1}}(x) \quad \sigma_k(H^f) \geq \alpha > 0 \]

\[
\left| \frac{\partial^2 g}{\partial y_i \partial y_j}(y_1) - \frac{\partial^2 g}{\partial y_i \partial y_j}(y_2) \right| \leq L_{i,j} \quad \frac{||y_1 - y_2||}{||y_1 - y_2||_i^k} \\
\text{Lipschitz constant} \\
L = \max_{1 \leq i,j \leq k} L_{i,j} \quad \Rightarrow \alpha = \Theta \left( \frac{1}{d} \right)
\]

(Model 1): \[ f(x) = g(Ax) \]

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

(Model 2): \[ f(x) = \sum_{i=1}^{k} g_i(a_i^T x) \text{ or } f(x) = a_1^T x + \sum_{i=2}^{k} g_i(a_i^T x) \]

+ $\nabla^2 g_i(0) \neq 0, \forall i = 2, \ldots, d$
**Theorem:** sample complexity

**Algorithm 1** Estimating $f(x) = g(Ax)$

1. Choose $m_\Phi$ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\Phi$.
2. Choose $\epsilon$ and construct $y$ 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 $\hat{\mathbf{X}}$ via a stable low-rank recovery algorithm.
4. Compute $\text{SVD}(\hat{\mathbf{X}}) = \hat{\mathbf{U}} \Sigma \hat{\mathbf{V}}^T$ and set $\hat{A}^T = \hat{\mathbf{U}}^{(k)}$, corresponding to $k$ largest singular values.
5. Obtain $\hat{f}(x) := \hat{g}(\hat{A}x)$ via quasi interpolants where $\hat{g}(y) := f(\hat{A}^T y)$.

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

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

$m_\Phi \geq \frac{\log(2/p_2) + 4k(d + m_\mathcal{X} + 1)u(\kappa)}{q(\kappa)},$ 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 $\hat{f}$ in step 5 of Algorithm 1 obeys $\|f - \hat{f}\|_{L_\infty} \leq \delta$ with probability at least $1 - p_1 - p_2$. 
Theorem: Sample complexity

**Algorithm 1** Estimating $f(x) = g(Ax)$

1. Choose $m_\Phi$ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\Phi$.
2. Choose $\epsilon$ and construct $y$ 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 $\hat{X}$ via a stable low-rank recovery algorithm.
4. Compute $\text{SVD}(\hat{X}) = \hat{U}\hat{\Sigma}\hat{V}^T$ and set $\hat{A}^T = \hat{U}^{(k)}$, corresponding to $k$ largest singular values.
5. Obtain $\hat{f}(x) := \hat{g}(\hat{A}x)$ via quasi interpolants where $\hat{g}(y) := f(\hat{A}^T y)$.

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

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

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

$$\epsilon \leq \frac{\delta}{C_2k^{5/2}d(\delta + 2C_2\sqrt{2}k)} \left( \frac{(1-\rho)m_\Phi \alpha}{(1+\kappa)C_0m_\mathcal{X}} \right)^{1/2}.$$

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

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

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

$$\epsilon = O\left( \frac{\alpha \delta}{\sqrt{d}} \right)$$
Theorem: proof ingredients

- Matrix Danzig selector as running example

\[ \hat{X}_{DS} = \arg \min_M \| M \|_* \text{ s.t. } \| \Phi^* (y - \Phi(M)) \| \leq \lambda \]

\[
\begin{align*}
    m_\chi &= \mathcal{O} \left( \frac{k \log k}{\alpha} \right) \\
    m_\Phi &= \mathcal{O} (k(d + m_\chi)) \\
    \epsilon &= \mathcal{O} \left( \frac{\alpha \delta}{\sqrt{d}} \right)
\end{align*}
\]

[Candes and Plan (2010)]
Theorem: proof ingredients

- Matrix Danzig selector as running example

\[ \hat{X}_{DS} = \arg \min_M \| M \|_* \quad \text{s.t.} \quad \| \Phi^* (y - \Phi(M)) \| \leq \lambda \]

- Tuning parameters

Proposition 1 We have \( \| \varepsilon \|_{\ell_2^m} \leq \frac{C_2 \varepsilon d m \chi k^2}{2 \sqrt{m}} \). Moreover, it holds that \( \| \Phi^*(\varepsilon) \| \leq \lambda = \frac{C_2 \varepsilon d m \chi k^2}{2 \sqrt{m}} (1 + \kappa)^{1/2} \), with probability at least \( 1 - 2e^{-m \Phi q(\kappa) + (d + m \chi + 1) u(\kappa)} \).
Theorem: proof ingredients

- Matrix Danzig selector as running example
  \[ \hat{X}_{DS} = \arg \min_M \|M\|_* \text{ s.t. } \|\Phi^* (y - \Phi(M))\| \leq \lambda \]

- Tuning parameters

- Recovery guarantees on \( X \)

Corollary 1: Denoting \( \hat{X}_{DS} \) to be the solution of the matrix Danzig selector, if \( \hat{X}^{(k)}_{DS} \) is the best rank-k approximation to \( \hat{X}_{DS} \) in the sense of \( \|\cdot\|_F \), and if \( \kappa_4 k < \kappa < \sqrt{2} - 1 \), then we have

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

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

[Theorem 2.4 from Candes and Plan (2010)]
Theorem: proof ingredients

- Matrix Danzig selector as running example

\[ \hat{X}_{DS} = \arg \min_M \| M \|_* \quad \text{s.t.} \quad \| \Phi^* (y - \Phi(M)) \| \leq \lambda \]

- Tuning parameters

- Recovery guarantees on \( X \)

- Translation of guarantees on \( X \) to guarantees on \( A \)

**Lemma 1** For a fixed \( 0 < \rho < 1 \), \( m_X \geq 1 \), \( m_\Phi < m_X d \) if \( \epsilon < \frac{1}{C_2 k^2 d \sqrt{k + 2}} \left( \frac{(1 - \rho)m_\Phi \alpha}{(1 + \kappa)C_0 m_X} \right)^{1/2} \),

then with probability at least \( 1 - k \exp \left\{ -\frac{m_X \alpha \rho^2}{2kC_2^2} \right\} - 2 \exp \left\{ -m_\Phi q(\kappa) + 4k(d + m_X + 1)u(\kappa) \right\} \)

we have

\[ \| \hat{A}^T \hat{A} \|_F \geq \left( k - \frac{2\tau^2}{(\sqrt{1-\rho})m_X \alpha - \tau^2} \right)^{1/2} \],

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

This is precisely where the restricted Hessian property is used...
Theorem: proof ingredients

- Matrix Danzig selector as running example
  \[ \hat{X}_{DS} = \arg \min_M \| M \|_* \text{ s.t. } \| \Phi^* (y - \Phi(M)) \| \leq \lambda \]

- 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: \( \hat{f}(x) = f(\hat{A}^T \hat{A}x) = g(\hat{A} \hat{A}^T \hat{A}x) \).
\[ \Rightarrow |f(x) - \hat{f}(x)| = |g(Ax) - g(\hat{A} \hat{A}^T \hat{A}x)| \leq C_2 \sqrt{k} \| (A - \hat{A} \hat{A}^T \hat{A})x \|_{\ell^2_k} \leq C_2 \sqrt{k} \| A - \hat{A} \hat{A}^T \hat{A} \|_F \| x \|_{\ell^2} . \]

Now it is easy to verify that:
\[ \| A - \hat{A} \hat{A}^T \hat{A} \|_F^2 = \text{Tr}((A^T - \hat{A}^T \hat{A} \hat{A}^T)(A - \hat{A} \hat{A}^T \hat{A})) = k - \| \hat{A} \hat{A}^T \|_F^2 . \]
Impact of noisy queries

Algorithm 1 Estimating $f(x) = g(Ax)$

1: Choose $m_\Phi$ and $m_\mathcal{X}$ and construct the sets $\mathcal{X}$ and $\Phi$.
2: Choose $\epsilon$ and construct $y$ 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 $\hat{X}$ via a stable low-rank recovery algorithm.
4: Compute $\text{SVD}(\hat{X}) = \hat{U}\hat{\Sigma}\hat{V}^T$ and set $\hat{A}^T = \hat{U}^{(k)}$, corresponding to $k$ largest singular values.
5: Obtain $\hat{f}(x) := \hat{g}(\hat{A}x)$ via quasi interpolants where $\hat{g}(y) := f(\hat{A}^T y)$.

• Assume evaluation of $f$ yields $f(x) + Z$, where $Z \sim \mathcal{N}(0, \sigma^2)$
Impact of noisy queries

Algorithm 1 Estimating $f(x) = g(Ax)$

1: Choose $m_{\Phi}$ and $m_{\mathcal{X}}$ and construct the sets $\mathcal{X}$ and $\Phi$.
2: Choose $\epsilon$ and construct $y$ 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 $\hat{X}$ via a stable low-rank recovery algorithm.
4: Compute $\text{SVD}(\hat{X}) = \hat{U}\hat{\Sigma}\hat{V}^T$ and set $\hat{A}^T = \hat{U}^{(k)}$, corresponding to $k$ largest singular values.
5: Obtain $\hat{f}(x) := \hat{g}(\hat{A}x)$ via quasi interpolants where $\hat{g}(y) := f(\hat{A}^Ty)$.

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

**tuning parameter changes:**

$$\|\Phi^*(\epsilon + z)\| \leq \frac{2\gamma\sigma}{\epsilon} \sqrt{2(1 + \kappa)m_{\mathcal{X}}m_{\Phi}} + \frac{C_2 \epsilon dm_{\mathcal{X}}k^2}{2\sqrt{m_{\Phi}}} (1 + \kappa)^{1/2}, \quad (\gamma > 2\sqrt{\log 12}).$$
Impact of noisy queries

Algorithm 1 Estimating $f(x) = g(Ax)$

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

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

  tuning parameter changes:

  $$\|\Phi^*(\varepsilon + z)\| \leq \frac{2\gamma\sigma}{\varepsilon} \sqrt{2(1 + \kappa)m_\mathcal{X}m_\Phi} + \frac{C_2\varepsilon dm_\mathcal{X}k^2}{2\sqrt{m_\Phi}}(1+\kappa)^{1/2}, \; (\gamma > 2\sqrt{\log 12})$$

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

  We resample the same data points $O(\varepsilon^{-1})$-times and average.
Learning a logistic function

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

\[ \alpha = \int |g'(a^T x)|^2 d\mu_{S_{d-1}} \approx |g'(0)|^2 = (1/16) \]

\[ C_2 = \sup_{|\beta| \leq 2} |g^{(\beta)}(y)| = 1 \]

- Declare success if

\[ |\langle \hat{a}, a \rangle| \geq 0.99 \]

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

\[ m_\chi = 20 \]
Learning sum of Gaussian functions

\[ f(x) = g(Ax + b) = \sum_{i=1}^{k} g_i(a_i^T x + b_i) \]

- \( d = 100 \)
- \( \epsilon = 10^{-3} \)
- \( m_{\chi} = 100 \)

- Declare success if

\[ \frac{1}{k} \| A\widehat{A}^T \|_F^2 \geq 0.99 \]

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

- theory: \( m_\Phi = O(d) \)
Stability example with the quadratic

\[ f(x) = g(Ax) = \|Ax - b\|^2 \]

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

- Declare success if

\[ \frac{1}{k} \left\| A \hat{A}^T \right\|_F^2 \geq 0.99 \]

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

\[ b_i \sim \mathcal{U}(S^{k-1}) \]

\[ k = 5 \]
\[ \epsilon = 10^{-1} \]
\[ m_x = 30 \]
\[ \sigma = 0.01 \]
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 2$^{nd}$ order derivative matrix ALPS for low-rank recov.
  **beyond linear models**
  system calibration, PDE models, matrix compression...