Approximating functions of few variables in high dimension

Based on joint work with Ron DeVore and Guergana Petrova

P. Wojtaszczyk

Instytut Matematyki Stosowanej i Mechaniki
Uniwersytet Warszawski

Paris, June 22, 2010

General problem

Many real life or scientific phenomena are modeled by functions that depend on many variables e.g.

- bank's decision to give credit,
- classification of medical images e.g. mammograms,
- digital picture is a function of milions of variables (brightness at each pixel)

We want to understand (approximate) such a function. This is very difficult to handle – curse of dimensionality.

Light in the tunnel

Very often in practice number of significant variables is much smaller – sparsity

There are many mathematical approaches to deal with this

1. Weighted norms in information based complexity
2. Juntas in theoretical computer science
3. Dimensionality reduction of data sets.
4. Compressed sensing

When we deal with functions one common assumption is that we can compute values of our function but it costs us.

First two are related to our work so let me explain a bit

Light in the tunnel

Very often in practice number of significant variables is much smaller – sparsity

There are many mathematical approaches to deal with this

1. Weighted norms in information based complexity
2. Juntas in theoretical computer science
3. Dimensionality reduction of data sets.
4. Compressed sensing

When we deal with functions one common assumption is that we can compute values of our function but it costs us.

First two are related to our work so let me explain a bit

Light in the tunnel

Very often in practice number of significant variables is much smaller – sparsity

There are many mathematical approaches to deal with this

1. Weighted norms in information based complexity
2. Juntas in theoretical computer science
3. Dimensionality reduction of data sets.
4. Compressed sensing

When we deal with functions one common assumption is that we can compute values of our function but it costs us.

First two are related to our work so let me explain a bit

Information based complexity sampler

Let H be a Hilbert space of smooth functions on $[0,1]^N$. The norm is a weighted norm like

$$\|f\|_w^2 = \sum_{\Lambda \subset \{1,2,\ldots,N\}} a_\Lambda \|\partial^\Lambda f\|_2^2$$

for some positive a_Λ's. We assume that $H \subset C([0,1]^N)$. If a_Λ big then $\|\partial^\Lambda f\|_2$ must be small; various sets of coordinates have different impact on the function.

They want to approximate f with $\|f\|_w \leq 1$ in $L_2([0,1]^N)$.

Question Given accuracy ϵ how many point values are needed to construct $\tilde{f}$ such that $\|f - \tilde{f}\|_2 \leq \epsilon$

Information based complexity sampler

Let H be a Hilbert space of smooth functions on $[0,1]^N$. The norm is a weighted norm like

$$\|f\|_w^2 = \sum_{\Lambda \subset \{1,2,\ldots,N\}} a_\Lambda \|\partial^\Lambda f\|_2^2$$

for some positive a_Λ's. We assume that $H \subset C([0,1]^N)$. If a_Λ big then $\|\partial^\Lambda f\|_2$ must be small; various sets of coordinates have different impact on the function.

They want to approximate f with $\|f\|_w \leq 1$ in $L_2([0,1]^N)$.

Question Given accuracy ϵ how many point values are needed to construct $\tilde{f}$ such that $\|f - \tilde{f}\|_2 \leq \epsilon$

Juntas

A function $f : \{0,1\}^N \to \{0,1\}$ which depends only on ℓ variables is an ℓ-junta.

Question How many point values we need to learn an ℓ–junta i.e. to find all essential variables.

Points may be either random (then we learn with high probability) or deterministic.

Our setup

We have a funtion f defined on $[0,1]^N$ where N is big. We assume that there exists a g on $[0,1]^\ell$ with ℓ small and integers $j_1 < j_2 < \cdots < j_\ell \leq N$ such that

$$f(x_1,\ldots,x_N) = g(x_{j_1},\ldots,x_{j_\ell}) \quad \text{exact case}$$

or

$$\|f(x_1,\ldots,x_N) - g(x_{j_1},\ldots,x_{j_\ell})\|_\infty \leq \epsilon. \quad \text{approximate case}$$

We know ℓ (or an estimate for ℓ). We do not know g, $j_1, j_2, \ldots, j_\ell$ nor ϵ. We assume g is smooth.

We want to find a subset $\mathcal{C} \subset [0,1]^N$ of small cardinality such that values $f(c)$ with $c \in \mathcal{C}$ will allow us to find $\tilde{g}(x_{j'_1},\ldots,x_{j'_\ell})$ which is a good approximation to f in sup norm.

There are both similarities and differences between our setup and IBC or juntas

Our setup

We have a funtion f defined on $[0,1]^N$ where N is big. We assume that there exists a g on $[0,1]^\ell$ with ℓ small and integers $j_1 < j_2 < \cdots < j_\ell \leq N$ such that

$$f(x_1,\ldots,x_N) = g(x_{j_1},\ldots,x_{j_\ell}) \quad \text{exact case}$$

or

$$\|f(x_1,\ldots,x_N) - g(x_{j_1},\ldots,x_{j_\ell})\|_\infty \leq \epsilon. \quad \text{approximate case}$$

We know ℓ (or an estimate for ℓ). We do not know g, $j_1, j_2, \ldots, j_\ell$ nor ϵ.
We assume g is smooth.
We want to find a subset $\mathcal{C} \subset [0,1]^N$ of small cardinality such that values $f(c)$ with $c \in \mathcal{C}$ will allow us to find $\tilde{g}(x_{j'_1},\ldots,x_{j'_\ell})$ which is a good approximation to f in sup norm.
There are both similarities and differences between our setup and IBC or juntas

Our setup

We have a funtion f defined on $[0,1]^N$ where N is big. We assume that there exists a g on $[0,1]^\ell$ with ℓ small and integers $j_1 < j_2 < \cdots < j_\ell \leq N$ such that

$$f(x_1,\ldots,x_N) = g(x_{j_1},\ldots,x_{j_\ell}) \quad \text{exact case}$$

or

$$\|f(x_1,\ldots,x_N) - g(x_{j_1},\ldots,x_{j_\ell})\|_\infty \leq \epsilon. \quad \text{approximate case}$$

We know ℓ (or an estimate for ℓ). We do not know g, $j_1, j_2, \ldots, j_\ell$ nor ϵ.
We assume g is smooth.
We want to find a subset $\mathcal{C} \subset [0,1]^N$ of small cardinality such that values $f(c)$ with $c \in \mathcal{C}$ will allow us to find $\tilde{g}(x_{j'_1},\ldots,x_{j'_\ell})$ which is a good approximation to f in sup norm.
There are both similarities and differences between our setup and IBC or juntas

Our setup–comments

- I will talk mostly about non–adaptive algorithm.
- Even in the exact case we cannot expect to find $j_1, \ldots, j_\ell$. There may be coordinates of g that do not show when we take our point values.
- Obviously there is no hope to recover g exactly.
- Our algorithms are rather theoretical–we are interested in the number of points needed.

Smoothness and approximation

Let $h = 1/L$ with L an integer. Lattice of equally spaced points $\mathcal{L} := h\mathcal{L}_\ell := \{h(i_1, \ldots, i_\ell), 0 \le i_1, \ldots, i_\ell \le L\} \subset [0,1]^\ell$. For each h, we have a nice linear operator $A_h : C(\mathcal{L}) \to C([0,1]^\ell)$ such that:

1. $\|A_h\| \le C_0$, for all $L = 1, 2, \ldots$.
2. If g depends on $k \le \ell$ variables then $A_h(g)$ depends only on these variables. So, $A_h(g) = g$, for any $g \equiv const$.

We consider the following approximation class:

$$\mathcal{A}^s := \mathcal{A}^s((A_h)) = \{g \in C([0,1]^\ell) : \ \|g - A_h(g|\mathcal{L})\|_\infty \le Ch^s\},$$

with semi-norm $|g|_{\mathcal{A}^s} := \sup_h \{h^{-s}\|g - A_h(g|\mathcal{L})\|_\infty\}$.
For natural A_h's those are real smoothness spaces.

Exact case I

We need two things

1. Active variables $j_1, j_2, \ldots, j_\ell$.
2. Values of g in points from $\mathcal{L}$.

This will give us approximation up to Ch^s. Even when we know $j_1, \ldots, j_\ell$ for 2. we need at least $(L+1)^\ell$ points.
How much do we have to pay to find appropriate coordinates?

Exact case–First attempt

Let $\mathcal{A}$ be a collection of partitions $\mathbf{A}$ of $\{1, 2, \ldots, N\}$. $\mathcal{A}$ is ν–separating if

1. each $\mathbf{A}$ consists of ν disjoint sets $A_1, \ldots, A_\nu$
2. given any ν distinct integers $i_1, \ldots, i_\nu \in \{1, \ldots, N\}$, there is a partition $\mathbf{A}$ in $\mathcal{A}$ such that each set in $\mathbf{A}$ contains precisely one of the integers $i_1, \ldots, i_\nu$.

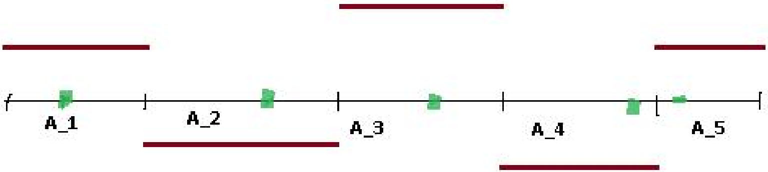
There exists such collections $\mathcal{A}$ of cardinality $\leq 2\nu e^\nu \ln N$
We fix a ℓ–separating collection $\mathcal{A}$.
For any $\mathbf{A} \in \mathcal{A}$ and $s = 1, \ldots, \ell$ we construct base points subordinated to the partition $\mathbf{A}$

$$\mathcal{P}_{\mathbf{A}} = \{P = \sum_{i=1}^{\ell} \alpha_i \chi_{A_i}, \ : \ \ \alpha_i \in \{0, 1/L, \ldots, 1\}$$

i.e. P has coordinate value α_i at each coordinate index in A_i. We consider base points i.e. $\bigcup_{\mathbf{A} \in \mathcal{A}} \mathcal{P}_{\mathbf{A}}$. There are $(L+1)^\ell \#(\mathcal{A})$ base points.

Base points

$\ell = 5$ so $\mathbf{A} = (A_1, A_2, \ldots, A_5)$.



There exists $\mathbf{A}$ and s such that $f(\mathcal{P}^s_{\mathbf{A}}) = g(\mathcal{L})$.

How to locate active coordinates?

For $\mathbf{A} \in \mathcal{A}$, $\mathbf{A} = (A_1, \ldots, A_\ell)$, we examine the base points $\mathcal{P}_{\mathbf{A}}$ subordinate to this $\mathbf{A}$. We say the set A_i is a change set if there is a pair $P, P' \in \mathcal{P}_{\mathbf{A}}$, for which P_i and P'_i only differ on A_i and $f(P_i) \neq f(P'_i)$.
A change set must contain an active variable. But how to locate it?

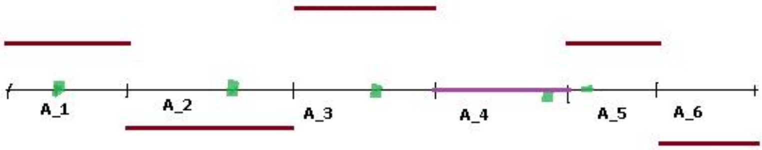
1. Add dummy coordinate – I use it for exact case.
2. Hunt for active variables in change sets I use it for approximate case.

Exact case–Final algorithm I

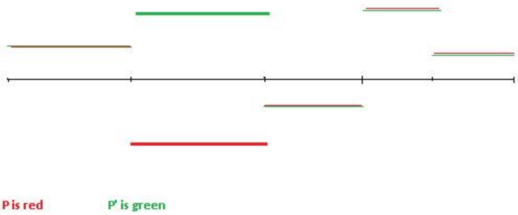
- We fix a $(\ell+1)$–separating collection $\mathcal{A}$.
- For any $\mathbf{A} \in \mathcal{A}$ and $s = 1, \ldots, \ell+1$ we construct base points subordinated to the partition $\mathbf{A}$

$$\mathcal{P}^s_{\mathbf{A}} = \{P = \sum_{i=1}^{\ell+1} \alpha_i \chi_{A_i}, \ : \quad \alpha_i \in \{0, 1/L, \ldots, 1\} \ \text{ and } \ \alpha_s = 0\},$$

i.e. P has coordinate value α_i at each coordinate index in A_i. We consider base points i.e. $\bigcup_{\mathbf{A}\in\mathcal{A}} \bigcup_{s=1}^{\ell+1} \mathcal{P}^s_{\mathbf{A}}$. There are $\leq (L+1)^\ell (\ell+1)\#(\mathcal{A})$ base points.

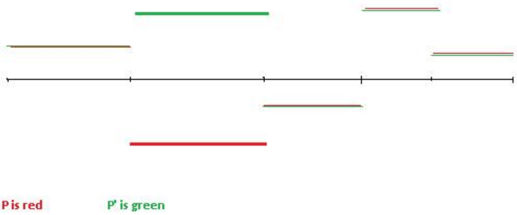


- We take the values of f at all base points. There exists $\mathbf{A}$ and s such that $f(\mathcal{P}^s_{\mathbf{A}}) = g(\mathcal{L})$.
- For $\mathbf{A} \in \mathcal{A}$, $\mathbf{A} = (A_1, \ldots, A_{\ell+1})$, we examine the base points $\mathcal{P}_{\mathbf{A}} =: \bigcup_{s=1}^{\ell+1} \mathcal{P}^s_{\mathbf{A}}$ subordinate to this $\mathbf{A}$. We say the set A_i is a change set if there is a pair $P, P' \in \mathcal{P}_{\mathbf{A}}$, for which P_i and P'_i only differ on A_i and $f(P_i) \neq f(P'_i)$.



- We take all partitions $\mathbf{A}_1, \ldots, \mathbf{A}_s$ with maximal number of change sets; the maximal number of change sets of a partition is $\mu \leq \ell$

- We take the values of f at all base points. There exists $\mathbf{A}$ and s such that $f(\mathcal{P}^s_{\mathbf{A}}) = g(\mathcal{L})$.
- For $\mathbf{A} \in \mathcal{A}$, $\mathbf{A} = (A_1, \ldots, A_{\ell+1})$, we examine the base points $\mathcal{P}_{\mathbf{A}} =: \bigcup_{s=1}^{\ell+1} \mathcal{P}^s_{\mathbf{A}}$ subordinate to this $\mathbf{A}$. We say the set A_i is a change set if there is a pair $P, P' \in \mathcal{P}_{\mathbf{A}}$, for which P_i and P'_i only differ on A_i and $f(P_i) \neq f(P'_i)$.



- We take all partitions $\mathbf{A}_1, \ldots, \mathbf{A}_s$ with maximal number of change sets; the maximal number of change sets of a partition is $\mu \leq \ell$

Exact case, Final algorithm III

- For each such partition $\mathbf{A}_j$ let U_j be the union of non change sets of this partition.
- We take $\{1, \ldots, N\} \setminus \bigcup_{j=1}^{s} U_j$. This equals $\{j_1, \ldots, j_\mu\}$. Any $\mathbf{A}_j$ with maximal number of change sets gives us $g(\mathcal{L})$ and $\{j_1, \ldots, j_\mu\}$ are right coordinates.
- Fix $\mathbf{A}$ with maximal number of change sets and assume $j_k \in A_k$ for $k = 1, \ldots, \mu$. $f|\mathcal{P}_{\mathbf{A}}$ gives a function $\hat{g}$ on $\ell \geq \mu$ dimensional lattice $\mathcal{L}$ so we get a function

$$\hat{f}(x_1, \ldots, x_N) = A_h(\hat{g})(x_{j_1}, \ldots, x_{j_\mu}) \in C([0,1]^N)$$

such that $\|f - \hat{f}\|_{C[0,1]^N} \leq |g|_{\mathcal{A}^s} h^s$.
- This algorithm is nonadaptive and uses $2(L+1)^\ell(\ell+1)^2 e^{\ell+1} \log N$ points

Exact case, Final algorithm III

- For each such partition $\mathbf{A}_j$ let U_j be the union of non change sets of this partition.
- We take $\{1,\ldots,N\} \setminus \bigcup_{j=1}^{s} U_j$. This equals $\{j_1,\ldots,j_\mu\}$. Any $\mathbf{A}_j$ with maximal number of change sets gives us $g(\mathcal{L})$ and $\{j_1,\ldots,j_\mu\}$ are right coordinates.
- Fix $\mathbf{A}$ with maximal number of change sets and assume $j_k \in A_k$ for $k = 1,\ldots,\mu$. $f|\mathcal{P}_{\mathbf{A}}$ gives a function $\hat{g}$ on $\ell \geq \mu$ dimensional lattice $\mathcal{L}$ so we get a function

$$\hat{f}(x_1,\ldots,x_N) = A_h(\hat{g})(x_{j_1},\ldots,x_{j_\mu}) \in C([0,1]^N)$$

  such that $\|f - \hat{f}\|_{C[0,1]^N} \leq |g|_{\mathcal{A}^s} h^s$.
- This algorithm is nonadaptive and uses $2(L+1)^\ell(\ell+1)^2 e^{\ell+1} \log N$ points

Exact case, Final algorithm III

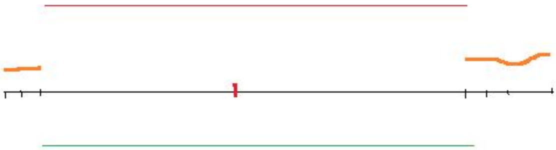
- For each such partition $\mathbf{A}_j$ let U_j be the union of non change sets of this partition.
- We take $\{1, \ldots, N\} \setminus \bigcup_{j=1}^{s} U_j$. This equals $\{j_1, \ldots, j_\mu\}$. Any $\mathbf{A}_j$ with maximal number of change sets gives us $g(\mathcal{L})$ and $\{j_1, \ldots, j_\mu\}$ are right coordinates.
- Fix $\mathbf{A}$ with maximal number of change sets and assume $j_k \in A_k$ for $k = 1, \ldots, \mu$. $f|\mathcal{P}_{\mathbf{A}}$ gives a function $\hat{g}$ on $\ell \geq \mu$ dimensional lattice $\mathcal{L}$ so we get a function

$$\hat{f}(x_1, \ldots, x_N) = A_h(\hat{g})(x_{j_1}, \ldots, x_{j_\mu}) \in C([0,1]^N)$$

  such that $\|f - \hat{f}\|_{C[0,1]^N} \leq |g|_{\mathcal{A}^s} h^s$.
- This algorithm is nonadaptive and uses $2(L+1)^\ell(\ell+1)^2 e^{\ell+1} \log N$ points

Exact case–hunt for change variable

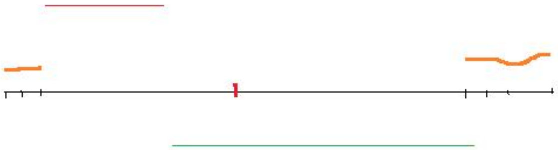
When a change set contains only one change variable?



$\mathcal{P}$ is red and $\mathcal{P}'$ is green and $f(\mathcal{P}) \neq f(\mathcal{P}')$

Exact case–hunt for change variable

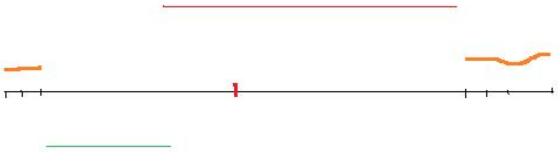
When a change set contains only one change variable?



$\mathcal{P}$ is red and $\mathcal{P}'$ is green and $f(\mathcal{P}) \neq f(\mathcal{P}')$. At this point f equals $f(\mathcal{P}')$.

Exact case–hunt for change variable

When a change set contains only one change variable?



$\mathcal{P}$ is red and $\mathcal{P}'$ is green and $f(\mathcal{P}) \neq f(\mathcal{P}')$. At this point f equals $f(\mathcal{P})$.

Approximate case I

1. We fix an $\ell-$ separating collection $\mathcal{A}$.
2. For any $\mathbf{A} \in \mathcal{A}$ we construct base points subordinated to the partition $\mathbf{A}$

$$\mathcal{P}_{\mathbf{A}} = \{P = \sum_{i=1}^{\ell} \alpha_i \chi_{A_i}, \; : \quad \alpha_i \in \{0, 1/L, \ldots, 1\}\},$$

i.e. P has coordinate value α_i at each coordinate index in A_i. We consider base points i.e. $\bigcup_{\mathbf{A} \in \mathcal{A}} \mathcal{P}_{\mathbf{A}}$. There are $(L+1)^{\ell} \#(\mathcal{A})$ base points.
3. For each $\mathbf{A}$ and each $i = 1, \ldots, \ell$, we choose exactly one pair of base points $P, P' \in \mathcal{P}_{\mathbf{A}}$ such that P and P' agree on all of the sets $A_\mu \neq A_i$ and the oscillation $\operatorname{osc}(P, P') := |f(P) - f(P')|$ is maximal. There are $\ell \#(\mathcal{A})$ such pairs. We call these pairs maximal.

4. We fix a set $\mathcal{B}$ of partitions into two sets such that given $\ell+1$ distinct integers $j, j_1, \ldots, j_\ell$, there is a partition $\mathbf{B} \in \mathcal{B}$ such that one set in $\mathbf{B}$ contains j and the other contains all of $j_1, \ldots, j_\ell$. There are such collections $\mathcal{B}$ of cardinality $\leq 2e(\ell+1)^2 \log N$.
5. For each maximal pair P, P' and each $\mathbf{B} \in \mathcal{B}$, we define two padding points $Q_\nu := [P, P']_{\mathbf{B},\nu}$, $\nu = 0, 1$, as follows. The j-th coordinate of Q_ν for each $j \in A_\mu$, $\mu \neq i$, is the common j-th coordinate of P and P'. For each $j \in A_i$, the j-th coordinate of Q_ν is the same as that of P' if $j \in A_i \cap B_\nu$. Otherwise it is the same as that of P.
6. We call a maximal pair P, P' useful if for each $\mathbf{B} \in \mathcal{B}$, there is exactly one value $\nu(\mathbf{B}) \in \{0, 1\}$ such that

$$|f([P, P']_{\mathbf{B},\nu(\mathbf{B})}) - f(P')| < \frac{1}{4}\mathrm{osc}(P, P'),$$

and

$$|f([P, P']_{\mathbf{B},\mu}) - f(P)| < \frac{1}{4}\mathrm{osc}(P, P'),$$

for $\mu \neq \nu(\mathbf{B})$.

7. For each maximal and useful pair of points P, P' which differ on A_i, we define

$$J_{P,P'} := \bigcap_{\mathbf{B}\in\mathcal{B}} B_{\nu(\mathbf{B})} \cap A_i.$$

This set is either empty or contains one coordinate.
8. We take all $J_{P,P'}$'s and choose ℓ with biggest $\operatorname{osc}(P, P')$. Those are our coordinates $\{j_1', \ldots, j_\ell'\}$.
9. We take $\mathbf{A} \in \mathcal{A}$ which separates $\{j_1', \ldots, j_\ell'\}$. Values $f(\mathcal{P}_{\mathbf{A}})$ give us function $\hat{g}$ on ℓ dimensional lattice $\mathcal{L}$ so we get a function

$$\hat{f}(x_1, \ldots, x_N) = A_h(\hat{g})(x_{j_1'}, \ldots, x_{j_\mu'}) \in C([0,1]^N)$$

Approximate case–summary

Theorem

Suppose that $f \in C([0,1]^N)$ *and there exists a function* $g \in \mathcal{A}^s$ *and a coordinates* $j_1, \ldots, j_\ell$ *such that* $\|f - \tilde{g}\|_{C([0,1]^N)} \leq \epsilon$ *where* $\tilde{g}(x_1, \ldots, x_N) = g(x_{j_1}, \ldots, x_{j_\ell})$. *Then the above algorithm produces a function* $\hat{f}$ *such that*

$$\|f - \hat{f}\|_{C(\Omega)} \leq |g|_{\mathcal{A}^s} h^s + C(28\ell + 1)\epsilon,$$

where C *is the constant dependent on the approximation process* A_h. *The number of point values used in the algorithm in the adaptive version is*

$$\leq (L+1)^\ell \#(\mathcal{A}) + 2\ell \#(\mathcal{A})\#(\mathcal{B}) \leq (L+1)^\ell 2\ell e^\ell \log N + 8(\ell+1)^4 e^\ell \log^2 N.$$

The number of point values used in the algorithm in the non-adaptive version is

$$\leq (L+1)^{\ell+1} 8(\ell+1)^4 e^\ell \log^2 N.$$

What next?

- Are our estimates for the number of points correct?
- What if the function depends on few variables but they are not the given ones?
- There is a parallel to compressed sensing–what about more analytical algorthms?
- What if points are given to us (i.e. we cannot choose them)?

That's all—Thanks

What next?

- Are our estimates for the number of points correct?
- What if the function depends on few variables but they are not the given ones?
- There is a parallel to compressed sensing–what about more analytical algorthms?
- What if points are given to us (i.e. we cannot choose them)?

That's all—Thanks

What next?

- Are our estimates for the number of points correct?
- What if the function depends on few variables but they are not the given ones?
- There is a parallel to compressed sensing–what about more analytical algorthms?
- What if points are given to us (i.e. we cannot choose them)?

That's all—Thanks

What next?

- Are our estimates for the number of points correct?
- What if the function depends on few variables but they are not the given ones?
- There is a parallel to compressed sensing–what about more analytical algorthms?
- What if points are given to us (i.e. we cannot choose them)?

That's all—Thanks

What next?

- Are our estimates for the number of points correct?
- What if the function depends on few variables but they are not the given ones?
- There is a parallel to compressed sensing–what about more analytical algorthms?
- What if points are given to us (i.e. we cannot choose them)?

That's all—Thanks

What next?

- Are our estimates for the number of points correct?
- What if the function depends on few variables but they are not the given ones?
- There is a parallel to compressed sensing–what about more analytical algorithms?
- What if points are given to us (i.e. we cannot choose them)?

That's all—Thanks