Approximate Modularity Revisited

Uriel Feige, Michal Feldman, Inbal Talgam-Cohen
YoungEC Workshop, Tel-Aviv, January 2017
Part of this work was done at Microsoft Research Herzliya

*Many slides from Uri’s talk at NY Theory Day
Motivation: Robustness

Mechanism and market design as a “data-driven enterprise”.

- "Nice" valuations -> Mechanism -> Good outcome
- Small errors
- Data on "nice" valuations

- "Close" to "nice" valuations -> Mechanism
- Good outcome???
Motivation: Robustness

Mechanism and market design as a “data-driven enterprise”.

“Nice” set functions → Optimization algorithm → Good outcome

Data on “nice” set functions → Small errors → “Close” to “nice” set functions

Optimization algorithm → Good outcome???
Motivation: Robustness

Mechanism and market design as a “data-driven enterprise”.

“Nice” set functions → Learning algorithm → Good outcome

Data on “nice” set functions → Small errors

“Close” to “nice” set functions → Learning algorithm → Good outcome???
Research agenda

What happens when set functions are only “close” to being “nice”?

Many sub-questions:

- Notions of closeness.
- Optimization – can good approximation ratios still be achieved?
- Mechanism design – do good economic properties continue to hold?
- Learning – can nice set functions be recovered?

- One take-away: Basic questions still open; interesting math involved
Set functions

Universe of $n$ items.
Function $f: \{0,1\}^n \to \mathbb{R}$ assigns values $f(S)$ to sets of items.

Examples:

• Items for sale in a *combinatorial auction*. Value of set (bundle) for a bidder – arbitrary valuation over bundles.

• Items can be *vertices* in an edge-weighted graph. Value of set: sum of weights of *cut* edges.
Representation of set functions

Explicit: $2^n$ entries.

Value queries: upon query $S$ learn $f(S)$.

- In a combinatorial auctions, one may ask a bidder how much she is willing to pay for the bundle.
- Given a set $S$ of vertices, one can compute in polynomial time the total weight of edges in the cut $(S, \overline{S})$.

Other types of queries (such as demand queries) have been studied.
Classes of “nice” set functions

Some classes of set functions have polynomial representations:

Additive \( f(S) = \sum_{i \in S} f(i) \), and \( f(\emptyset) = 0 \).

Linear \( f(S) = f(\emptyset) + \sum_{i \in S} f(i) \).

Capped additive \( f(S) = \min[1, g(S)] \) where \( g \) is additive.

Useful properties but not necessarily a polynomial representation:

Submodular \( f(S) + f(T) \geq f(S \cap T) + f(S \cup T) \).

Subadditive \( f(S) + f(T) \geq f(S \cup T) \), and \( f(\emptyset) = 0 \).
Optimization problems for “nice” set functions

**Linear** set functions:

- Many optimization problems are easy.

**Submodular** set functions (cuts, valuations with diminishing returns):

- **Maximization** subject to constraints (e.g., a cardinality constraint) can be solved approximately.

**General** set functions:

- Many optimization problems are computationally hard to approximate.
Motivations for “close to nice” set functions

• Answers to value queries might not be exact (due to noise in measurements).
• Rounding errors.
• Computing approximate values may be cheaper than computing exact values.
• The functions might not be exactly nice (e.g., valuation functions of bidders need not be exactly submodular).
Some related work

Functions on continuous domains (rather than discrete hypercube):
  • Hyers [1941], ...

Data-driven optimization:
  • Bertsimas and Thiele [2014], Singer and Vondrak [2015], Hassidim and Singer [2016], Balkanski, Rubinstein, and Singer [2016], ...

Approximate submodularity, convexity, substitutes:
  • Das and Kempe [2011], Belloni, Liang, Narayanan, and Rakhlin [2015], Roughgarden, T.C., and Vondrak [2016], ...

Learning submodular functions:
  • Balcan and Harvey [2011], ...

...
Results
Suggested to start with an “easy” case: a function $f$ close to linear.

Considered 2 questions:

- To what extent are different measures of closeness to linearity related to each other? Specifically, compare between:
  - being pointwise close to a linear function,
  - nearly satisfying properties of linear functions.

- How to learn a linear function $h$ that is close to $f$?
Δ-linear set functions

Linear set function: $f(S) = f(\emptyset) + \sum_{i \in S} f(i)$.

Given $\Delta \geq 0$, a set function $f$ is $\Delta$-linear if there is a linear set function $g$ such that $|f(S) - g(S)| \leq \Delta$ for every set $S$. 
\( \varepsilon \)-modular set functions

A set function is linear if and only if it is modular:
\[
f(S) + f(T) = f(S \cap T) + f(S \cup T)
\]
for all sets \( S \) and \( T \).

A set function is \( \varepsilon \)-modular if
\[
|f(S) + f(T) - f(S \cap T) - f(S \cup T)| \leq \varepsilon
\]
for all sets \( S \) and \( T \).

Every \( \Delta \)-linear function is \( \varepsilon \)-modular for \( \varepsilon \leq 4\Delta \).

Is it true that every \( \varepsilon \)-modular function is \( \Delta \)-linear for \( \Delta \leq O(\varepsilon) \)?
Results for $\varepsilon$-modularity

$\varepsilon$-modular is $\Delta$-linear for:

$\Delta \leq O(\varepsilon \log n)$ [Chierichetti, Das, Dasgupta, and Kumar; 2015]

$\Delta \leq 44.5\varepsilon$ [Kalton and Roberts; 1983] (Assaf Naor directed us to this)

$\Delta \leq 35.8\varepsilon$ [Bondarenko, Prymak and Radchenko; 2013]

Our results:

• For every set function $\Delta < 13\varepsilon$.

• Improved bounds for special classes. E.g., for symmetric functions $\Delta \leq \frac{1}{2}\varepsilon$.

• There are set functions (with $n = 70$) for which $\Delta \geq \varepsilon$. 

$\Delta$-linear: pointwise close to linear up to $\pm \Delta$

$\varepsilon$-modular: satisfies modularity eqs. up to $\pm \varepsilon$
Learning $\Delta$-linear set functions

Suppose we are given value query access to a $\Delta$-linear function $f$. Using polynomially many value queries, output a linear function $h$ satisfying $|f(S) - h(S)| \leq \delta$ for every set $S$.

How small can we make $\delta$ as a function of $\Delta$ and $n$?
Results for learning $\Delta$-linear set functions

Results of [Chierichetti et al.]:

• A randomized algorithm making $O(n^2 \log n)$ nonadaptive queries and achieving $\delta \leq O(\Delta \sqrt{n})$ w.h.p.

• Even randomized adaptive algorithms require $\delta \geq \Omega(\Delta \sqrt{\frac{n}{\log n}})$.

Useful when $f$ is very close to linear (e.g. rounding errors).

Our result:

• A deterministic algorithm making $O(n)$ nonadaptive queries and achieving $\delta \leq O(\Delta \sqrt{n})$. 

$\Delta$-linear: pointwise close to linear up to $\pm \Delta$
Sketch of Main Proof
Plan for main proof

W.l.o.g. let $f$ be a 1-modular function whose closest linear function is $g = 0$. Let $M = \max_{S} [f(S)] = -\min_{S} [f(S)]$.

[Chierichetti et al.] characterize such $f$. Show that $M = \Delta$ is bounded, independent of $n$.

(For simplicity suppose $f(\emptyset) = 0$.)

$\Delta$-linear: pointwise close to linear up to $\pm \Delta$

$\epsilon$-modular: satisfies modularity eqs. up to $\pm \epsilon$

$\Delta < 13\epsilon$
Tool: $(r, \theta)$ split-and-merge of $k$ sets

Source sets: $S_1, S_2, \ldots, S_k$.

Split each source set into several intermediate sets (in a clever way).

Intermediate sets: $I_1, I_2, \ldots, I_{rk}$.

Merge together disjoint intermediate sets into target sets.

Target sets: $T_1, T_2, \ldots, T_{\theta k}$ for $\theta < 1$. 
Example: \((r, \theta)\) split-and-merge of \(k\) sets

\(k = 3\) source sets:
\[(1,2,3,4,7)\] \[(1,2,5,6)\] \[(3,4,5,7)\]

\(rk = 6\) intermediate sets (hence \(r = 2\)):
\[(1,2,7) (3,4)\] \[(1,2) (5,6)\] \[(3,4) (5,7)\]

\(\theta k = 2\) target sets (hence \(\theta = \frac{2}{3}\)):
\[(1,2,7,5,6,3,4)\] \[(3,4,1,2,5,7)\]
Implications of 1-modularity

If source set $S$ is the disjoint union of $r$ intermediate sets $I_1, ..., I_r$. Then $\sum_j f(I_j) \geq f(S) - r + 1$.

If target set $T$ is the disjoint union of $r/\theta$ intermediate sets $I_1, ..., I_{r/\theta}$. Then $\sum_j f(I_j) \leq f(T) + r/\theta - 1$.

Summing up over source, target sets and combining we get $\sum_j f(S_j) - rk + k \leq \sum_j f(I_j) \leq \sum_j f(T_j) + rk - \theta k$. 
(r, \theta) split-and-merge of k sets with value M

From prev. slide: \[ \sum_j f(S_j) - rk + k \leq \sum_j f(T_j) + rk - \theta k. \]

Suppose that \( f(S_j) = M \) for every \( S_j \).

This implies \( kM - rk + k \leq \theta kM + rk - \theta k \), and thus \( M \leq \frac{2r - 1 - \theta}{1 - \theta} \).

Thus if \( r \) is constant and \( \theta < 1 \), we derive that \( M (=\Delta) \) is constant.

What condition ensures that \( k \) sets have an (r, \theta) split-and-merge?
**$\alpha$-sparse collections**

For $\alpha < 1$, a collection of $k$ sets is $\alpha$-sparse if each item appears in at most $\alpha k$ sets.

**Lemma** [Kalton and Roberts, 1983]: For any $\alpha$-sparse collection there is an $(r, \theta)$ split-and-merge as required, where $r$ and $\theta < 1$ depend on parameters of bipartite expander graphs.

**Corollary**: To show $M = O(1)$ it remains to show an $\alpha$-sparse collection of sets with value (nearly) $M$. 
Example: From $\alpha$-sparse to split-and-merge via expanders

\begin{itemize}
  \item $(1,2,3,4,7)$
  \item $(1,2,5,6)$
  \item $(3,4,5,7)$
\end{itemize}

$\frac{2}{3}$-sparse collection

Expander:
Every set with $\leq \frac{2}{3}$ of the top vertices has a matching to the bottom vertices.
Example: From $\alpha$-sparse to split-and-merge via expanders

$\frac{2}{3}$-sparse collection

Expander:
Every set with $\leq \frac{2}{3}$ of the top vertices has a matching to the bottom vertices.
Example: From $\alpha$-sparse to split-and-merge via expanders

\(\frac{2}{3}\)-sparse collection

Expander:
Every set with \(\leq \frac{2}{3}\) of the top vertices has a matching to the bottom vertices.
Example: From $\alpha$-sparse to split-and-merge via expanders

$\frac{2}{3}$-sparse collection

Expander:
Every set with $\leq \frac{2}{3}$ of the top vertices has a matching to the bottom vertices.
Example: From $\alpha$-sparse to split-and-merge via expanders

Source sets ($k$)

Intermediate sets ($rk$)

Target sets ($\theta k$)
Existence of $\alpha$-sparse collection with value $\approx M$

Lemma [only “morally” correct]: for every 1-modular function $f$ whose closest linear function is 0, there is a $\frac{1}{2}$-sparse collection of sets of average value $M - d$, where deficit $d \leq \frac{1}{2}$.

This implies $M \leq \frac{2r+d-1-\theta}{1-\theta} \leq 26.8$, where the last inequality is by existence of expanders such that every set with $\leq \frac{1}{2}$ of the top vertices has a matching, there are $rk = 5.05k$ edges and $\hat{\theta}k = \frac{2}{3}k$ bottom vertices [BPR 2013].
The road to substantial improvements

Key observation: if we require “less expansion” (i.e., that only “smaller” subsets of top vertices have matchings), then there exist expanders with relatively less edges and bottom vertices $\Rightarrow$ smaller $r, \theta$.

We can pair these with sparser collections of sets – as long as their average value is still $M - d'$ for small $d'$ – $\Rightarrow$ better upper bound $\frac{2r + d' - 1 - \theta}{1 - \theta} \geq M$. 
Example of implementation

Goal: lowering $\alpha$ of $\alpha$-sparse collection while controlling the deficit

We already have a $\frac{1}{2}$-sparse collection of sets $S_1, S_2, \ldots$ of average value $M - d$, where $d \leq \frac{1}{2}$.

Consider a collection composed of all pairwise intersections.

Observe it is $\frac{1}{4}$-sparse.

Its average value can be bounded by:

$$f(S_i \cap S_j) \geq f(S_i) + f(S_j) - f(S_i \cup S_j) - 1 \geq M - 2d - 1$$
Summary

Relating two notions of “close to” linear functions:
• Every $\varepsilon$-modular set function is $\Delta$-linear for $\Delta \leq O(\varepsilon)$.
• Current bounds are $\varepsilon \leq \Delta < 13\varepsilon$.
• Proof based on expander graphs.

Learning close to linear functions – only when really close
• A linear function $h$ that is $O(\Delta\sqrt{n})$-close to a $\Delta$-linear function $f$ can be learned by making $O(n)$ value queries non-adaptively.
• Nearly best possible.