Explainable AI & Leverage Score Sampling

R. Teal Witter

New York University

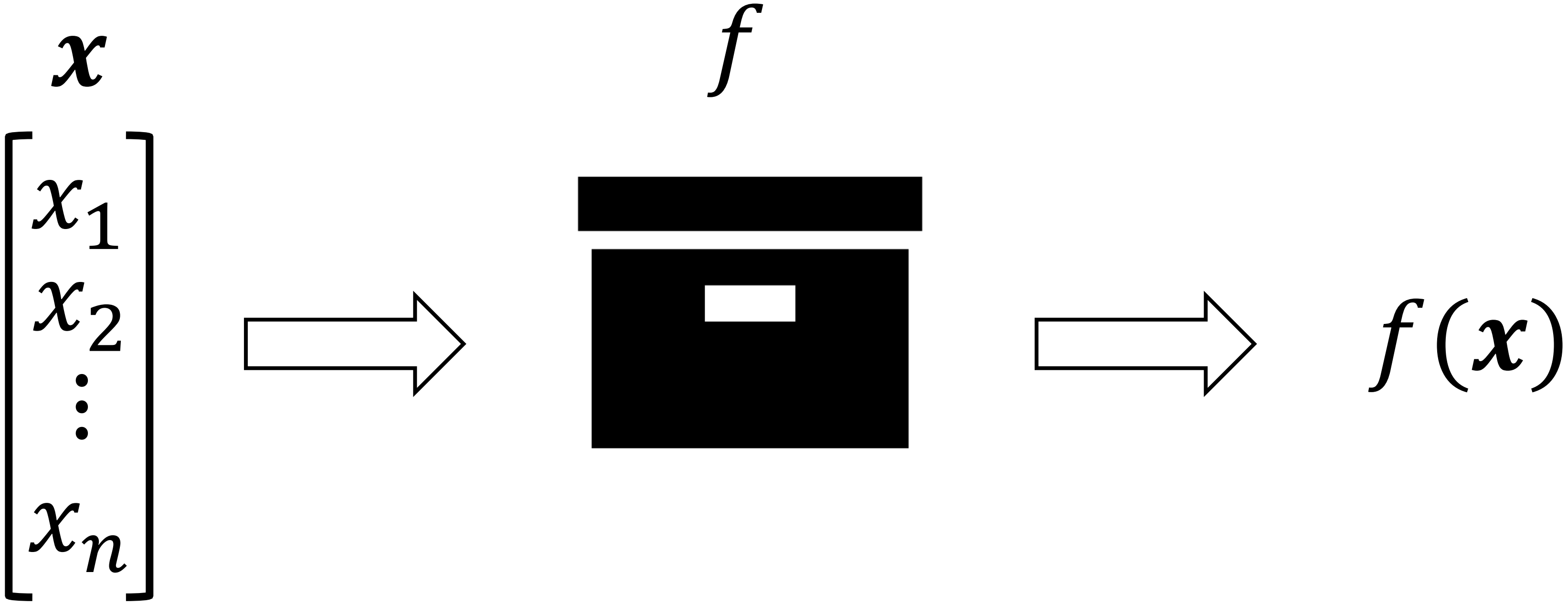
Shapley Values & Leverage Score Sampling

Joint work with

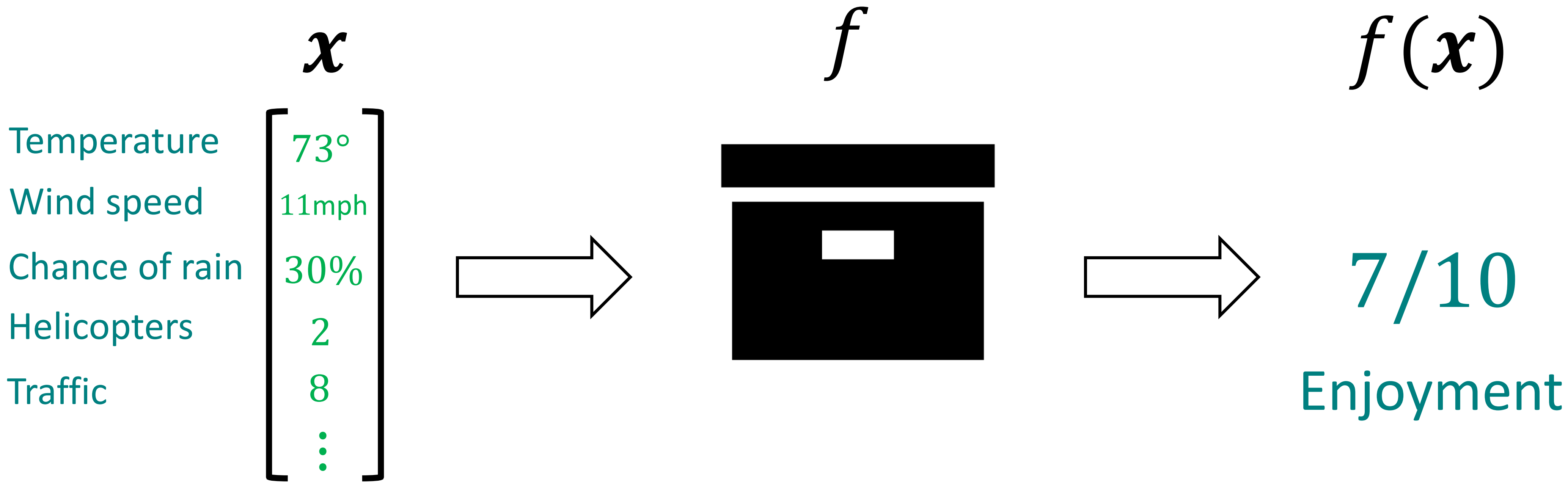
Christopher Musco

New York University

AI Prediction



Example:



$\boldsymbol{x}$
Temperature
Wind speed
Chance of rain
Helicopters
Traffic
73°
11mph
30%
2
8
f
$f(\boldsymbol{x})$
7/10
Enjoyment

Explaining Predictions

Attribute the prediction to features relative to a baseline

"Since the traffic is 8 instead of 3, the ride is 1.7 less enjoyable."

Attribution value!

Explaining Predictions

Attribute the prediction to features relative to a baseline

“Since the traffic is 8 instead of 3, the ride is 1.7 less enjoyable.”

$$
\begin{array}{l}
\text{Temperature} \\
\text{Wind speed} \\
\text{Chance of rain} \\
\text{Helicopters} \\
\text{Traffic} \\
\\
\end{array}
\begin{bmatrix} 73^\circ \\ 11\text{mph} \\ 30\% \\ 2 \\ 8 \\ \vdots \end{bmatrix}
\qquad
\begin{bmatrix} 89^\circ \\ 1\text{mph} \\ 0\% \\ 5 \\ 3 \\ \vdots \end{bmatrix}
$$

Explicand Baseline

					$f(x)$
89°	11mph	30%	5	3	5/10
89°	11mph	30%	5	8	4/10
73°	1mph	0%	5	3	6/10
73°	1mph	0%	5	8	8/10

Attribution Values

What is the effect of the feature in different settings?

Consider subsets $S \subseteq [n]$ and define $v(S) = f(\boldsymbol{x}^S)$ where

S						$f(\boldsymbol{x}^S)$
$\{2,3\}$	$89°$	11mph	30%	5	3	$5/10$
$\{2,3,5\}$	$89°$	11mph	30%	5	8	$4/10$

Attribution Values

What is the effect of the feature in different settings?

Consider subsets $S \subseteq [n]$ and define $v(S) = f(\boldsymbol{x}^S)$ where

S						$f(\boldsymbol{x}^S)$
$\{2,3\}$	89°	11mph	30%	5	3	$5/10$
$\{2,3,5\}$	89°	11mph	30%	5	8	$4/10$

Next: Define attribution value ϕ_i for every feature $i \in [n]$

Desirable Properties

Null Player: *If a feature never changes the prediction, then its attribution value is 0*

Symmetry: *If two features always induce the same change, then their attribution values are the same*

Additivity: *For two predictive functions, the attribution value of a feature in the combined function is the sum of the attribution values for each function*

Efficiency: *The attribution values sum to the difference between the predictions on the explicand and baseline*

$\Leftrightarrow$ Shapley values!

Shapley Values for Feature Attribution

For a set function $v: 2^{[n]} \to \mathbb{R}$, the ith Shapley value is

$$\phi_i = \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \frac{v(S \cup \{i\}) - v(S)}{\binom{n-1}{|S|}}$$

$$\phi_i = \underbrace{\frac{1}{n} \sum_{k \in [n-1]} \underbrace{\frac{1}{\binom{n-1}{k}} \sum_{S \subseteq [n] \setminus \{i\} : |S| = k} v(S \cup \{i\}) - v(S)}_{\text{Average over sets of size } k}}_{\text{Average over all sizes } k}$$

Estimating Shapley values

$$\phi_i = \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \frac{v(S \cup \{i\}) - v(S)}{\binom{n-1}{|S|}}$$

Monte Carlo Sampling: Sample $S, S \cup \{i\}$ to use $v(S \cup \{i\}) - v(S)$

… but samples only used for one Shapley value

Estimating Shapley values

$$\phi_i = \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \frac{v(S \cup \{i\}) - v(S)}{\binom{n-1}{|S|}} = \frac{1}{n} \sum_{S : i \in S} \frac{v(S)}{\binom{n-1}{|S|}} - \frac{1}{n} \sum_{S : i \notin S} \frac{v(S)}{\binom{n-1}{|S|}}$$

Monte Carlo Sampling: Sample $S, S \cup \{i\}$ to use $v(S \cup \{i\}) - v(S)$

… but samples only used for one Shapley value

Maximum Reuse Sampling: Sample S to either add/subtract $v(S)$ for all i

… but magnitude of $v(S)$ is much larger than magnitude of $v(S \cup \{i\}) - v(S)$

Estimating Shapley values

$$\phi_i = \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \frac{v(S \cup \{i\}) - v(S)}{\binom{n-1}{|S|}}$$

Monte Carlo Sampling: Sample $S, S \cup \{i\}$ to use $v(S \cup \{i\}) - v(S)$

… but samples only used for one Shapley value

Maximum Reuse Sampling: Sample S to either add/subtract $v(S)$ for all i

… but magnitude of $v(S)$ is much larger than magnitude of $v(S \cup \{i\}) - v(S)$

Permutation Sampling: Sample $S_1 \subset S_2 \subset \cdots \subset S_n$ to use $v(S_{\ell+1}) - v(S_\ell)$

… but only 2x reuse

Regression Formulation

$$\boldsymbol{\phi} = \underset{\boldsymbol{\beta}:\langle\boldsymbol{\beta},\mathbf{1}\rangle = v([n]) - v(\emptyset)}{\arg\min} \sum_{\boldsymbol{z}\in\{0,1\}^n:\mathbf{0}<||\boldsymbol{z}||_1<n} w\left(||\boldsymbol{z}||_1\right)\left(\langle\boldsymbol{\beta},\boldsymbol{z}\rangle - v(\boldsymbol{z})\right)^2$$

Best linear fit to set function under weighting

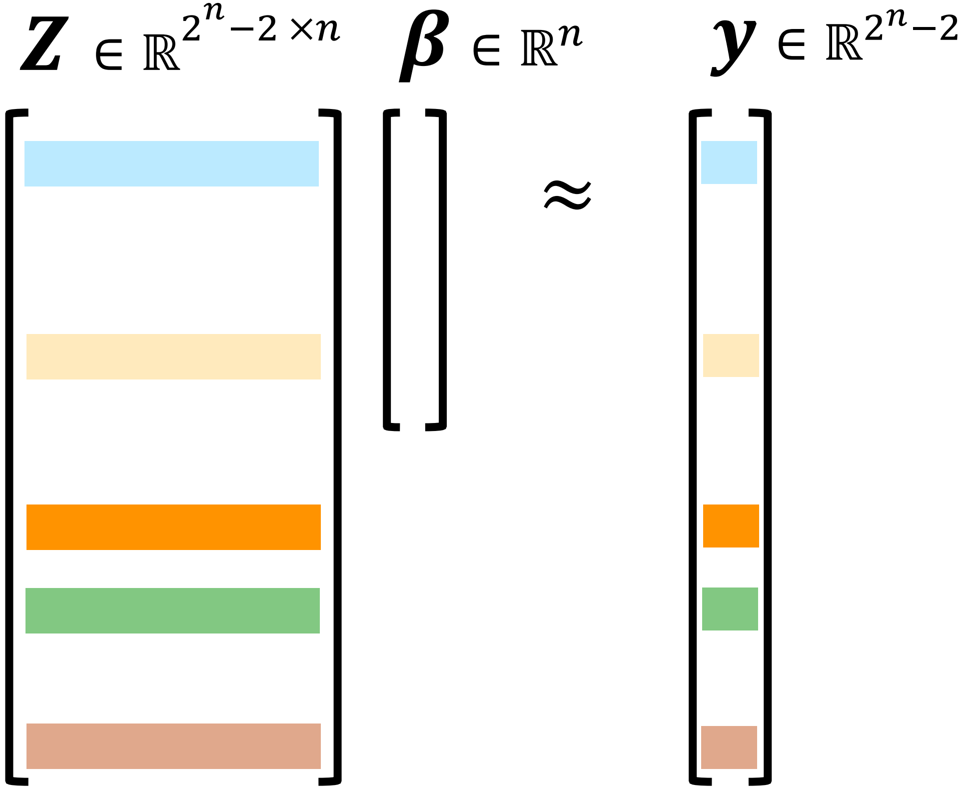
$$w(||\boldsymbol{z}||_1) = \frac{1}{\binom{n}{||\boldsymbol{z}||_1}(n - ||\boldsymbol{z}||_1)||\boldsymbol{z}||_1}$$

Known since 80's [CGKR 1988]

Regression Formulation

$$\boldsymbol{\phi} = \underset{\boldsymbol{\beta}:\langle\boldsymbol{\beta},\mathbf{1}\rangle = v([n]) - v(\emptyset)}{\arg\min} \sum_{\boldsymbol{z}\in\{0,1\}^n : \mathbf{0} < \|\boldsymbol{z}\|_1 < n} w\left(\|\boldsymbol{z}\|_1\right)\left(\langle\boldsymbol{\beta},\boldsymbol{z}\rangle - v(\boldsymbol{z})\right)^2$$

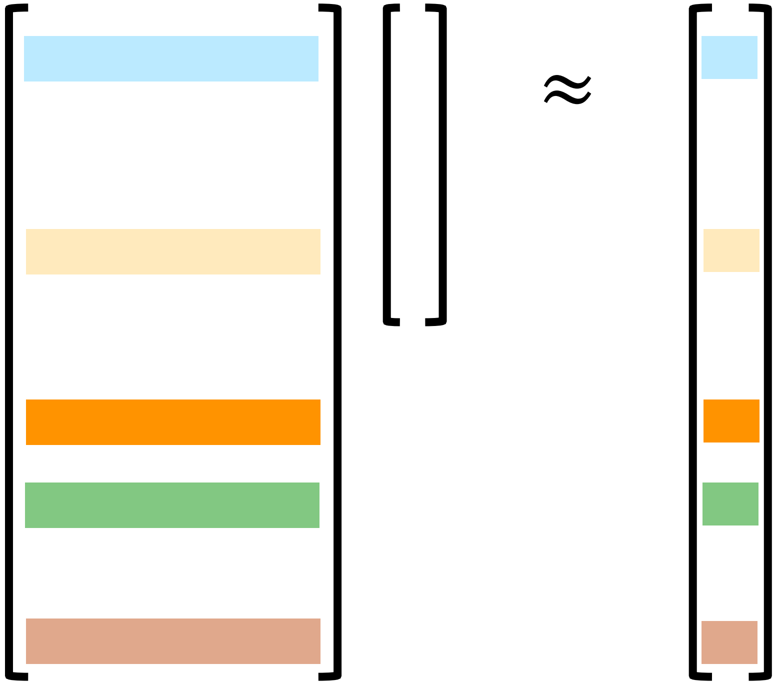
$$= \underset{\boldsymbol{\beta}:\langle\boldsymbol{\beta},\mathbf{1}\rangle = v([n]) - v(\emptyset)}{\arg\min} \|\boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{y}\|_2$$



Regression Formulation

$$\boldsymbol{\phi} = \underset{\boldsymbol{\beta}:\langle \boldsymbol{\beta}, \mathbf{1} \rangle = v([n]) - v(\emptyset)}{\arg\min} ||\boldsymbol{Z}\boldsymbol{\beta} - \boldsymbol{y}||_2$$

$\boldsymbol{Z} \in \mathbb{R}^{2^n - 2 \times n}$ $\boldsymbol{\beta} \in \mathbb{R}^n$ $\boldsymbol{y} \in \mathbb{R}^{2^n-2}$

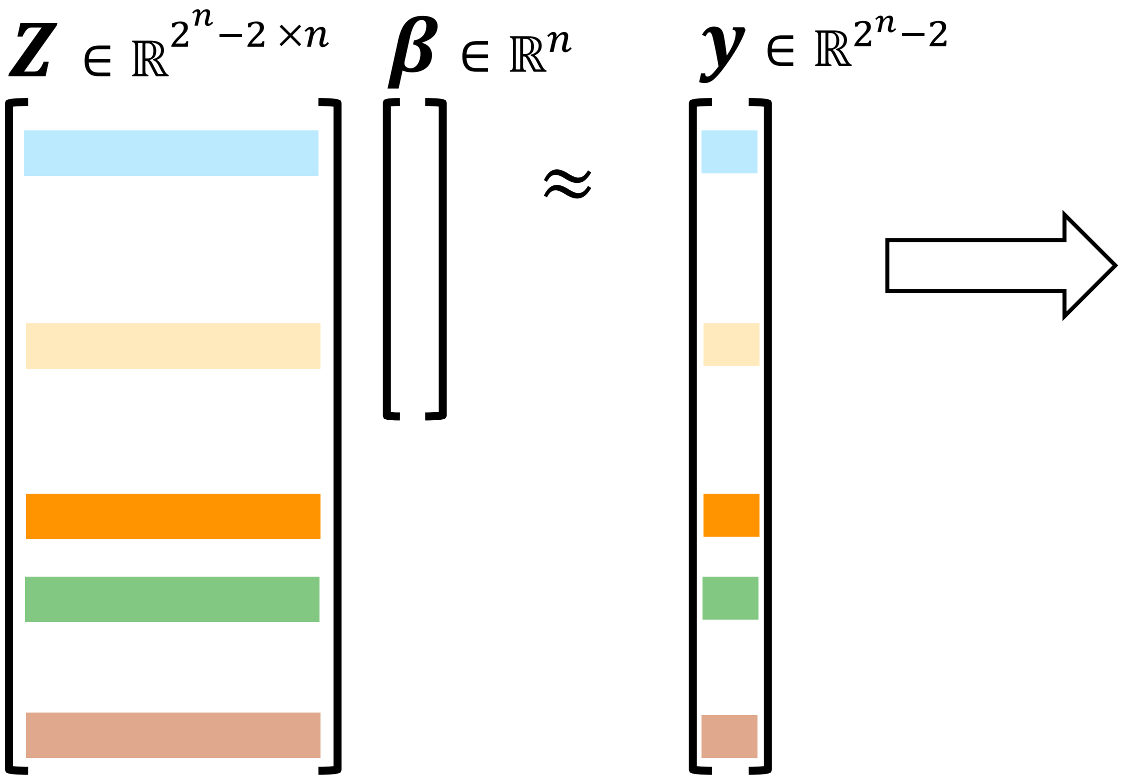


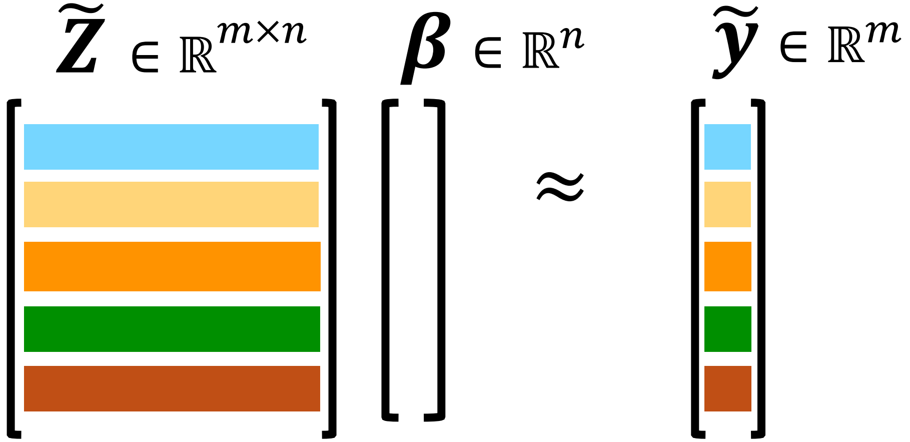
Very cool...

but still exponential time to solve!

Kernel SHAP

$$\tilde{\boldsymbol{\phi}} = \underset{\boldsymbol{\beta}:\langle \boldsymbol{\beta}, \mathbf{1} \rangle = v([n]) - v(\emptyset)}{\arg\min} ||\tilde{\boldsymbol{Z}}\boldsymbol{\beta} - \tilde{\boldsymbol{y}}||_2$$

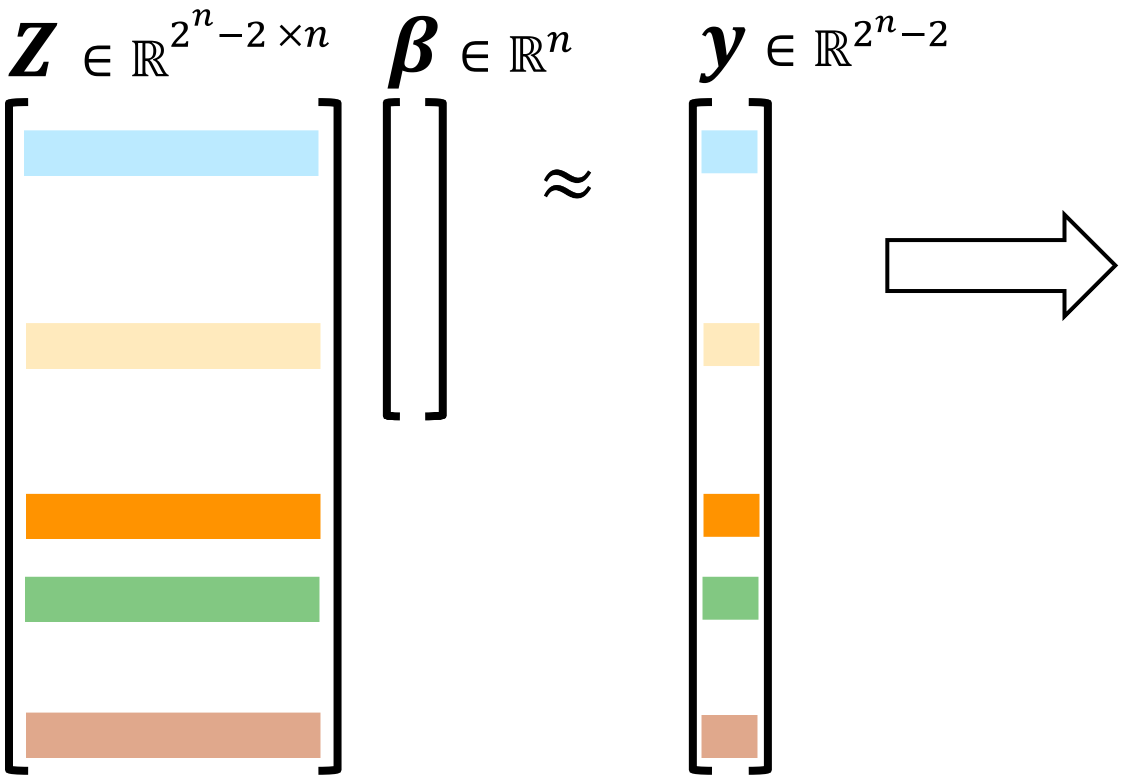


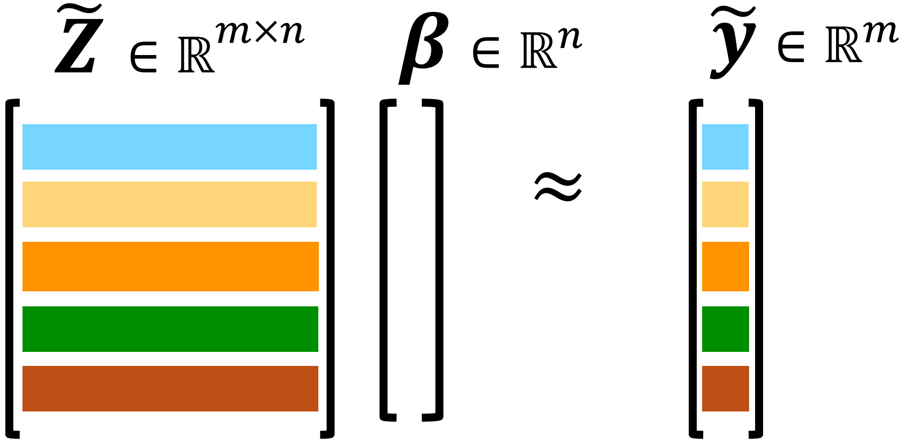


1. Subsample to manageable problem
2. Solve subsampled problem exactly

Kernel SHAP

$$\tilde{\boldsymbol{\phi}} = \underset{\boldsymbol{\beta}:\langle \boldsymbol{\beta}, \mathbf{1} \rangle = v([n]) - v(\emptyset)}{\arg\min} ||\tilde{\boldsymbol{Z}}\boldsymbol{\beta} - \tilde{\boldsymbol{y}}||_2$$





How should we subsample?

Kernel SHAP selects row $\boldsymbol{z}$ with probability proportional to $w(||\boldsymbol{z}||)$

Constrained to Unconstrained Regression

$$\boldsymbol{\phi} = \underset{\boldsymbol{\beta}:\langle\boldsymbol{\beta},\mathbf{1}\rangle=v([n])-v(\emptyset)}{\arg\min} ||\boldsymbol{Z\beta}-\boldsymbol{y}||_2$$

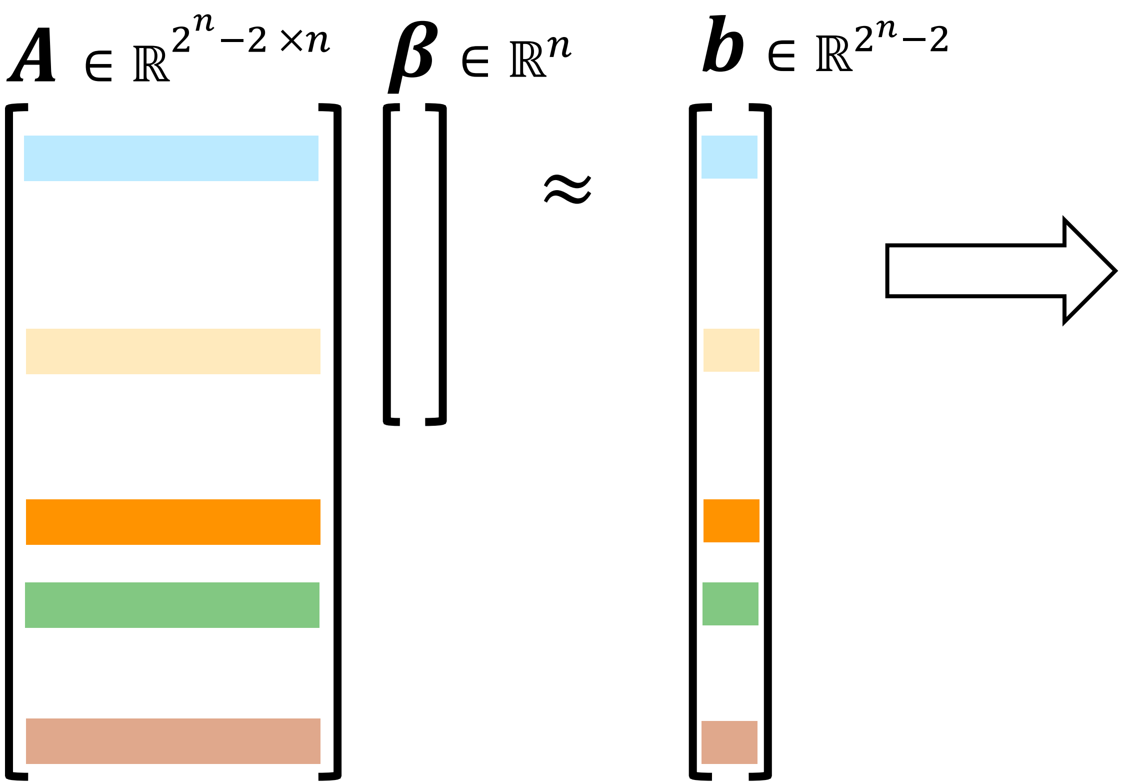
$$= \underset{\boldsymbol{\beta}}{\arg\min} ||\boldsymbol{A\beta}-\boldsymbol{b}||_2 + \mathbf{1}\frac{v([n])-v(\emptyset)}{n}$$

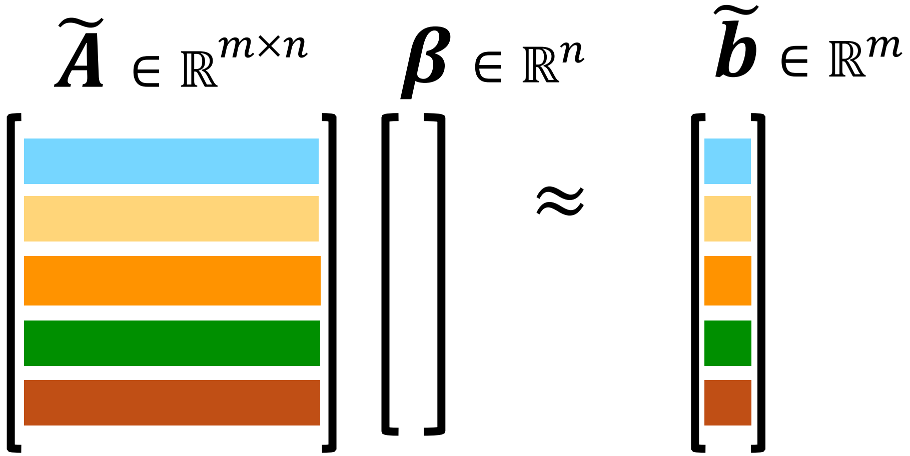
- By constraint, we know the component in the $\mathbf{1}$ direction
- Only optimize to residual target in space orthogonal to $\mathbf{1}$

Formulate as unconstrained problem so we can apply our favorite tools!

Regression Subsampling

$$\phi = \arg\min_{\boldsymbol{\beta}} ||\boldsymbol{A}\boldsymbol{\beta} - \boldsymbol{b}||_2 + \mathbf{1}\frac{v([n]) - v(\emptyset)}{n}$$

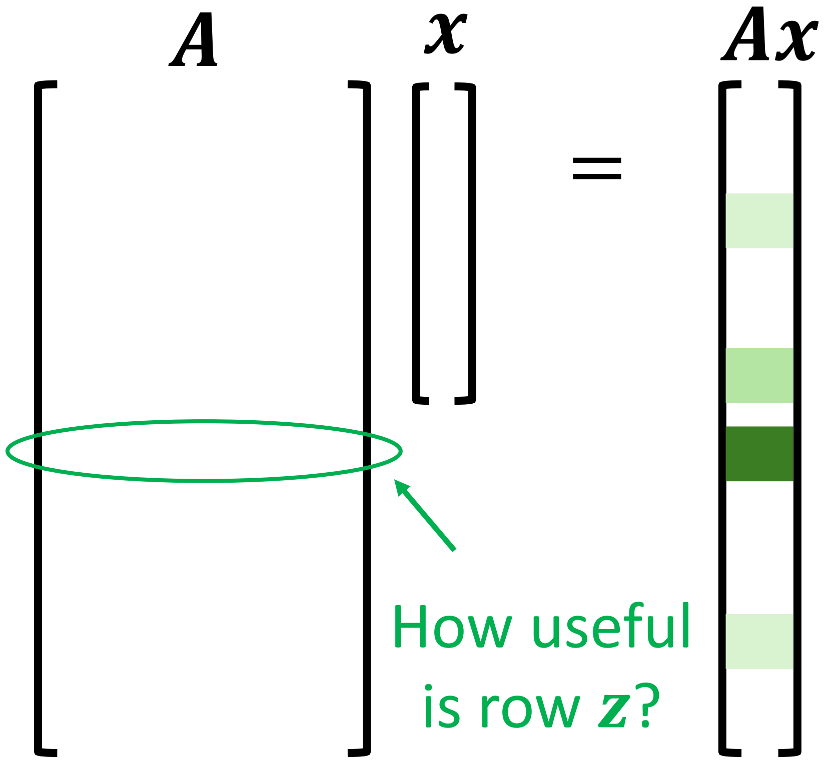




How should we subsample?

Select rows that will help us!

Leverage Scores



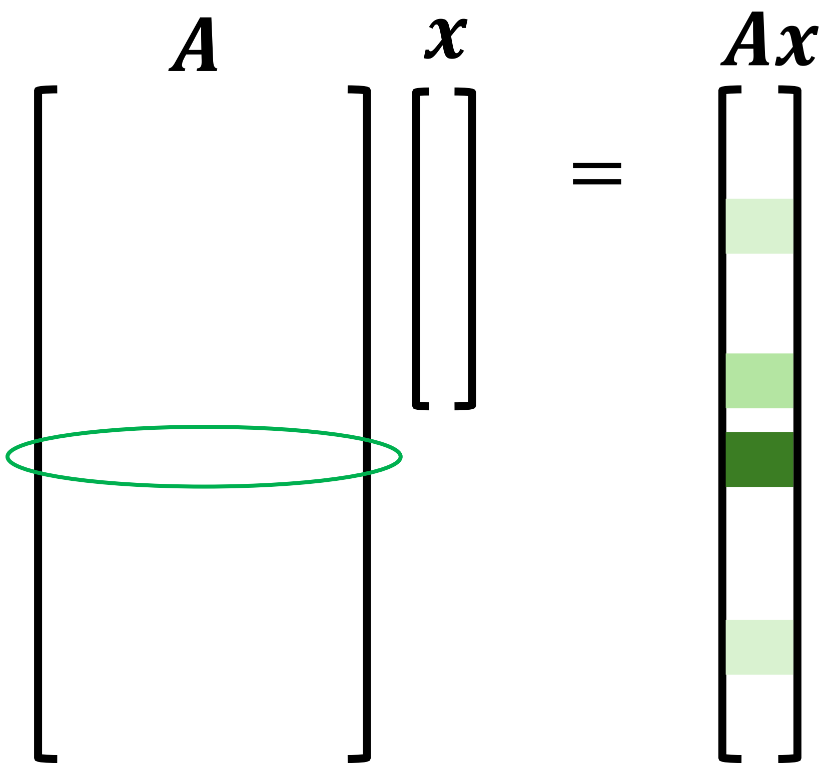
How useful is row z?

If there is an Ax like this, then we *need* row z.

Row z has "leverage":

$$\ell_z = \max_x \frac{(Ax)^2{}_z}{||Ax||_2^2}$$

Leverage Scores and Shapley Values



Row $\boldsymbol{z}$ has "leverage":

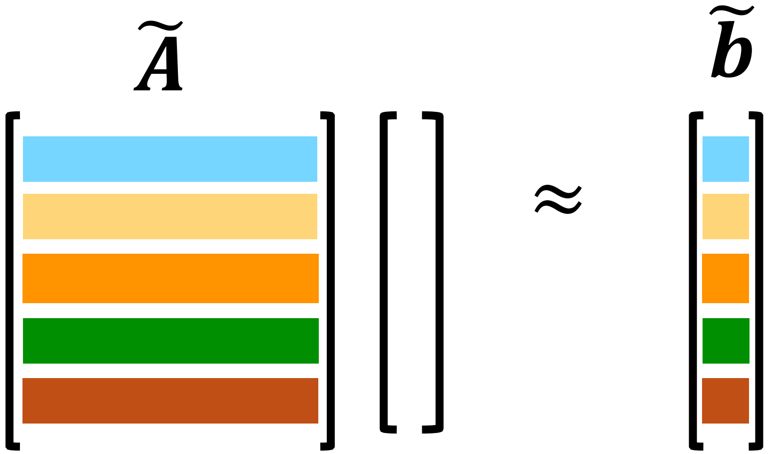
$$\ell_{\boldsymbol{z}} = \max_{\boldsymbol{x}} \frac{(\boldsymbol{A}\boldsymbol{x})^2{}_{\boldsymbol{z}}}{||\boldsymbol{A}\boldsymbol{x}\,||_2^2}$$

$$\ell_{\boldsymbol{z}} = \binom{n}{||\boldsymbol{z}||}^{-1}$$

Very similar to weighting in Shapley value definition!

Leverage SHAP

$$\tilde{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\beta}} ||\tilde{\boldsymbol{A}}\boldsymbol{\beta} - \tilde{\boldsymbol{b}}||_2 + \mathbf{1}\frac{v([n]) - v(\emptyset)}{n}$$

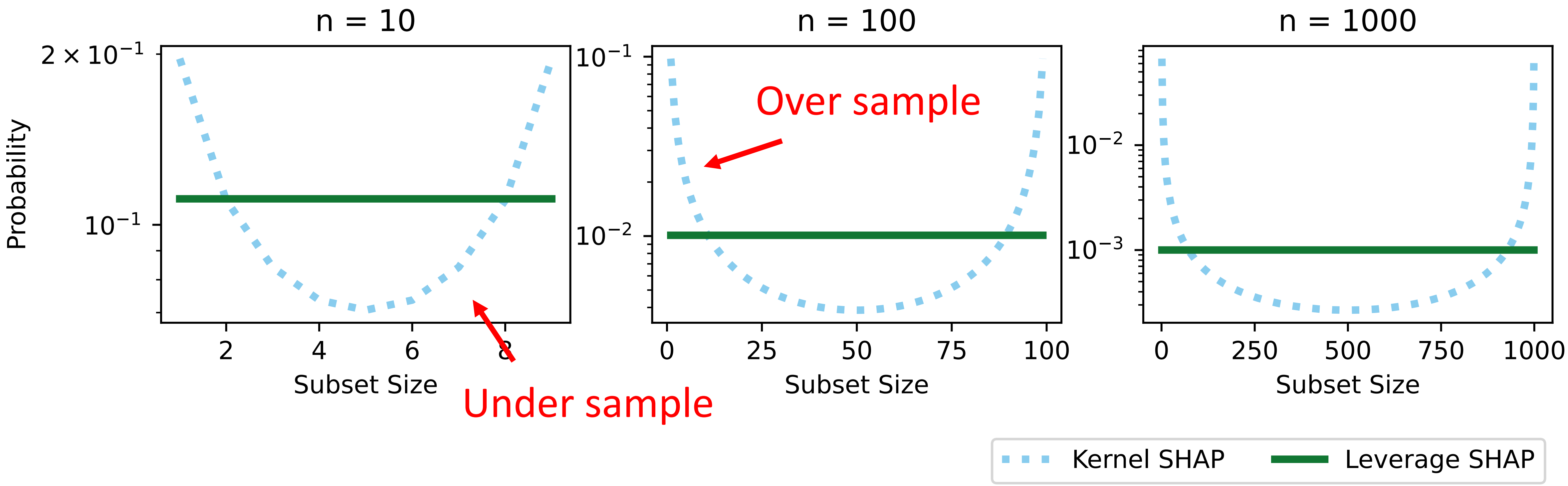


Leverage SHAP selects row $\boldsymbol{z}$ with probability proportional to leverage score!

+ Paired Sampling

+ Bernoulli Sampling

Leverage SHAP vs Kernel SHAP Probabilities

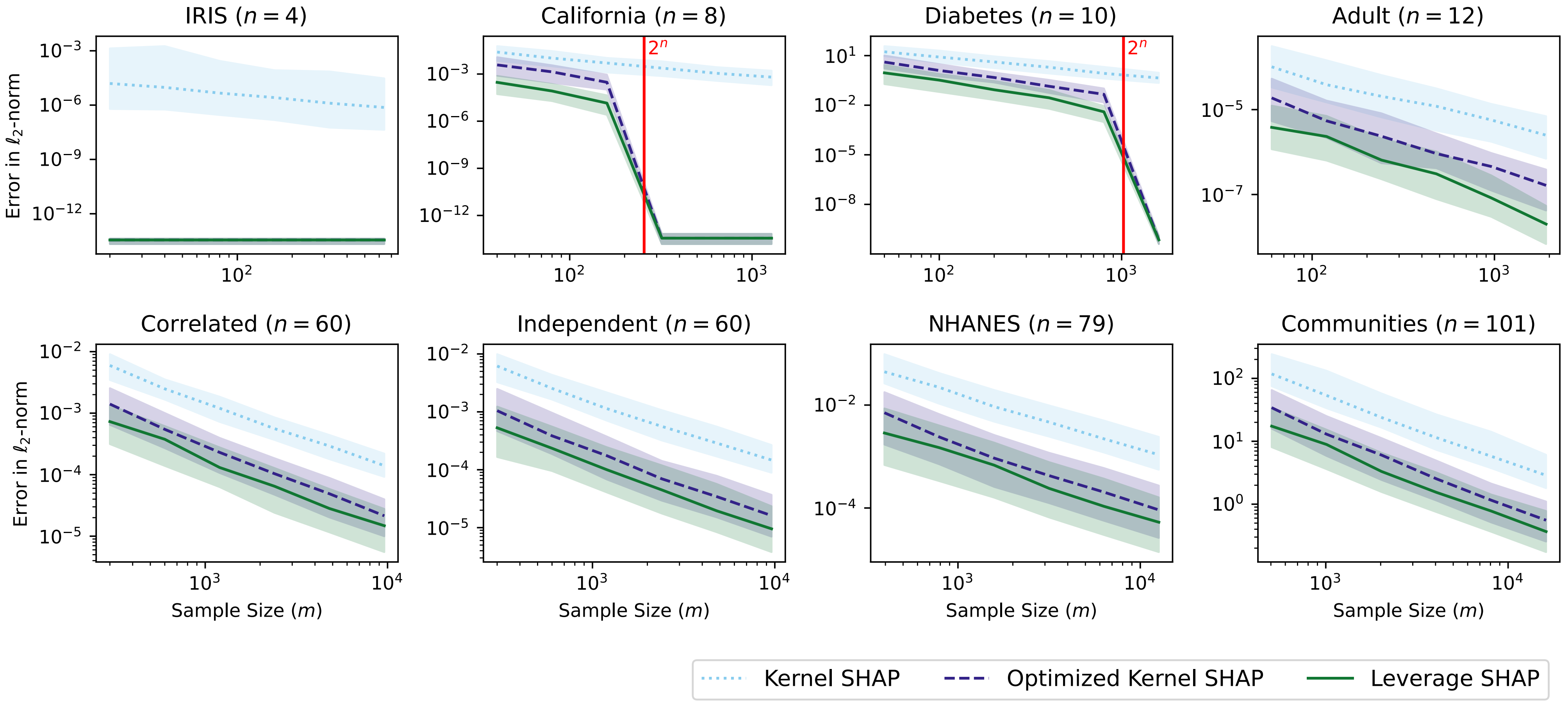


Leverage SHAP vs Kernel SHAP

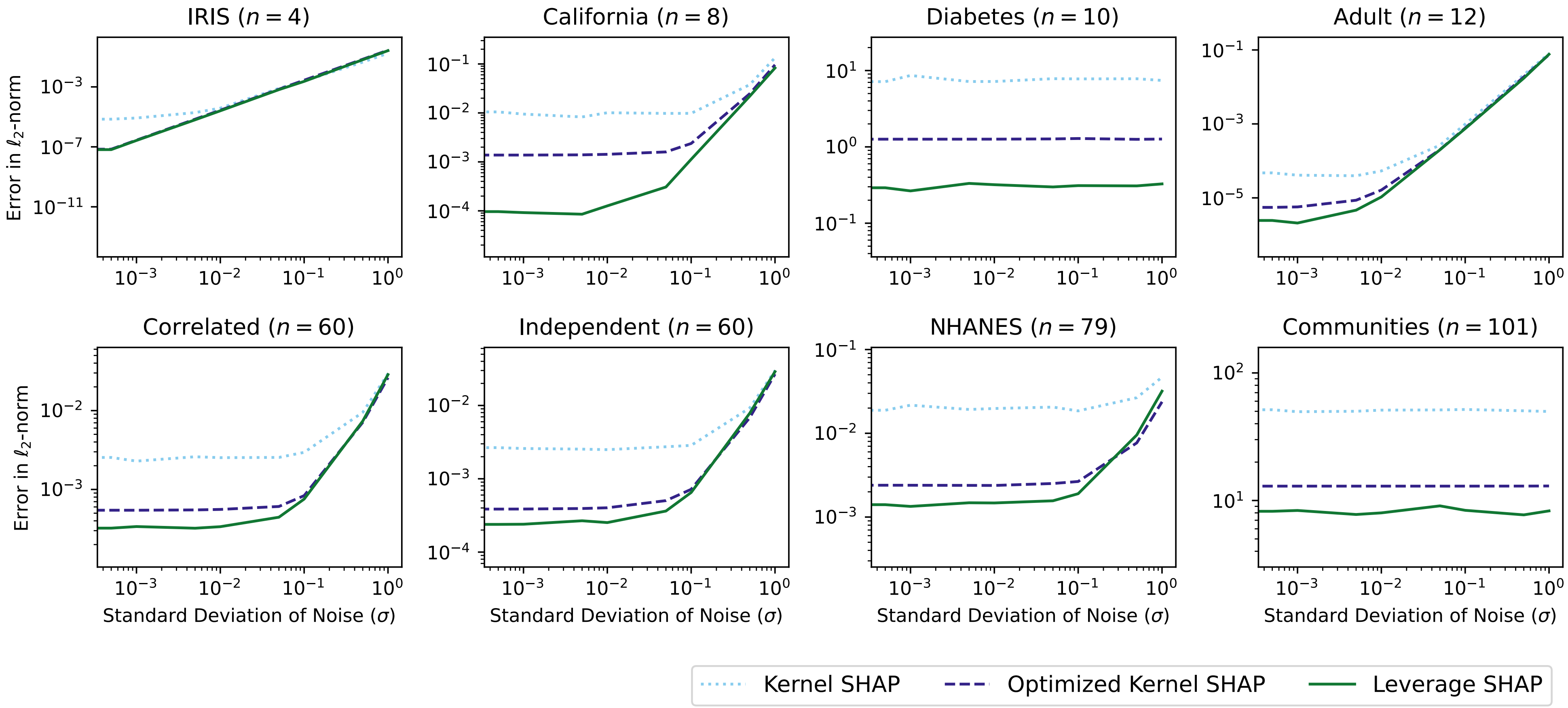
	IRIS	California	Diabetes	Adult	Correlated	Independent	NHANES	Communities
Kernel SHAP								
Mean	0.00261	0.0208	15.4	0.000139	0.00298	0.00324	0.0358	130.0
1st Quartile	5.69e-07	0.0031	3.71	1.48e-05	0.00166	0.00163	0.0106	33.5
2nd Quartile	9.52e-06	0.0103	8.19	3.86e-05	0.00249	0.00254	0.0221	53.6
3rd Quartile	0.00181	0.029	20.1	0.000145	0.00354	0.00436	0.0418	132.0
Optimized Kernel SHAP								
Mean	3.28e-14	0.00248	2.33	1.81e-05	0.000739	0.000649	0.00551	21.8
1st Quartile	2.12e-14	0.000279	0.549	2.16e-06	0.00027	0.000187	0.000707	5.85
2nd Quartile	3.55e-14	0.00138	1.26	5.43e-06	0.000546	0.000385	0.0024	13.0
3rd Quartile	4.22e-14	0.0036	3.03	1.63e-05	0.00101	0.000964	0.00665	25.1
Leverage SHAP								
Mean	3.28e-14	0.000186	0.63	5.21e-06	0.000458	0.000359	0.00385	14.7
1st Quartile	2.12e-14	1.91e-05	0.0631	6.3e-07	0.000139	9.51e-05	0.000333	3.6
2nd Quartile	3.55e-14	8.31e-05	0.328	2.33e-06	0.000376	0.000235	0.00149	8.9
3rd Quartile	4.22e-14	0.000231	0.769	7.09e-06	0.000617	0.000556	0.00401	15.3

Table 1: Summary statistics of the ℓ_2-norm error for every dataset. We adopt the Olympic medal convention: gold, silver and bronze cells signify first, second and third best performance, respectively. Except for ties, Leverage SHAP gives the best performance across all settings.

Accuracy by Sample Size



Accuracy by Noise



Robustness is useful, e.g., $v(S) = \mathbb{E}_{\boldsymbol{x}^S}[f(\boldsymbol{x}^S)]$

Theoretical Guarantee

As long as $m = O\left(n \log n + \frac{n}{\epsilon}\right)$, the Leverage SHAP solution $\tilde{\boldsymbol{\phi}}$ satisfies

$$||\boldsymbol{A}\tilde{\boldsymbol{\phi}} - \boldsymbol{b}||_2^2 \leq (1+\epsilon)||\boldsymbol{A}\boldsymbol{\phi} - \boldsymbol{b}||_2^2$$

with probability 9/10.

Guarantee similar to standard leverage analysis but proof requires

- Modifications for paired sampling
- Modifications for sampling without replacement

Interpretable Corollary

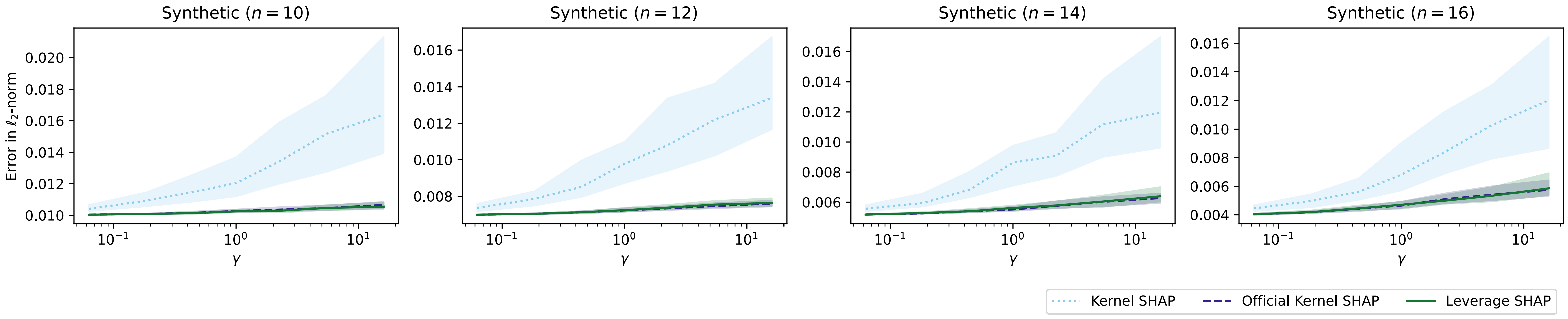
As long as $m = O\left(n \log n + \frac{n}{\epsilon}\right)$, the Leverage SHAP solution $\tilde{\boldsymbol{\phi}}$ satisfies

$$||\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}||_2^2 \leq \epsilon\, \gamma ||\boldsymbol{\phi}||_2^2$$

with probability 9/10 where $\gamma = \frac{||\boldsymbol{A}\,\boldsymbol{\phi} - \boldsymbol{b}||_2^2}{||\boldsymbol{A}\,\boldsymbol{\phi}||_2^2} \in [0, \infty)$

Intuition: We can find $\tilde{\boldsymbol{\phi}}$ close to the optimal in objective value but, when optimal solution is bad, $\tilde{\boldsymbol{\phi}}$ will be far from $\boldsymbol{\phi}$

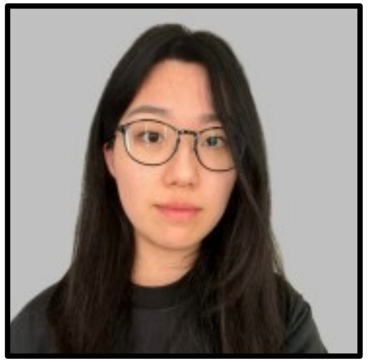
γ in Practice



Takeaway: γ is a parameter of regression (not artifact of analysis)

Banzhaf Values & Leverage Score Sampling

Joint work with





Yurong Liu
NYU

Flip Korn
Google

Tarfah Alrashed
Google

Dimitris Paparas
Google

Juliana Freire
NYU

Another Attribution Value?

What is the effect of the feature in different settings?

Consider subsets $S \subseteq [n]$ and define $v(S) = f(\boldsymbol{x}^S)$ where

S						$f(\boldsymbol{x}^S)$
$\{2,3\}$	89°	11mph	30%	5	3	$5/10$
$\{2,3,5\}$	89°	11mph	30%	5	8	$4/10$

Next: Define *<u>another</u>* attribution value ϕ_i for every feature $i \in [n]$

Desirable Properties

Null Player: *If a feature never changes the prediction, then its attribution value is 0*

Symmetry: *If two features always induce the same change, then their attribution values are the same*

Additivity: *For two predictive functions, the attribution value of a feature in the combined function is the sum of the attribution values for each function*

- Efficiency: *The attribution values sum to the difference between the predictions on the explicand and baseline*

+ 2-Efficiency: *If two features are combined, their combined attribution value is the sum of the features' individual attribution values*

$\Longleftrightarrow$ Banzhaf values!

John Banzhaf is an activist lawyer, he used Banzhaf values to argue a Nassau County voting system was unfair.

Banzhaf Values

For a set function $v: 2^{[n]} \to \mathbb{R}$, the ith Banzhaf value is

$$\phi_i = \frac{1}{2^{n-1}} \sum_{S \subseteq [n] \setminus \{i\}} v(S \cup \{i\}) - v(S)$$

Banzhaf values are

- Simpler
- Empirically easier to approximate

Estimating Banzhaf values

$$\phi_i = \frac{1}{2^{n-1}} \sum_{S \subseteq [n] \setminus \{i\}} v(S \cup \{i\}) - v(S)$$

Monte Carlo (MC): Sample $S, S \cup \{i\}$ to use $v(S \cup \{i\}) - v(S)$

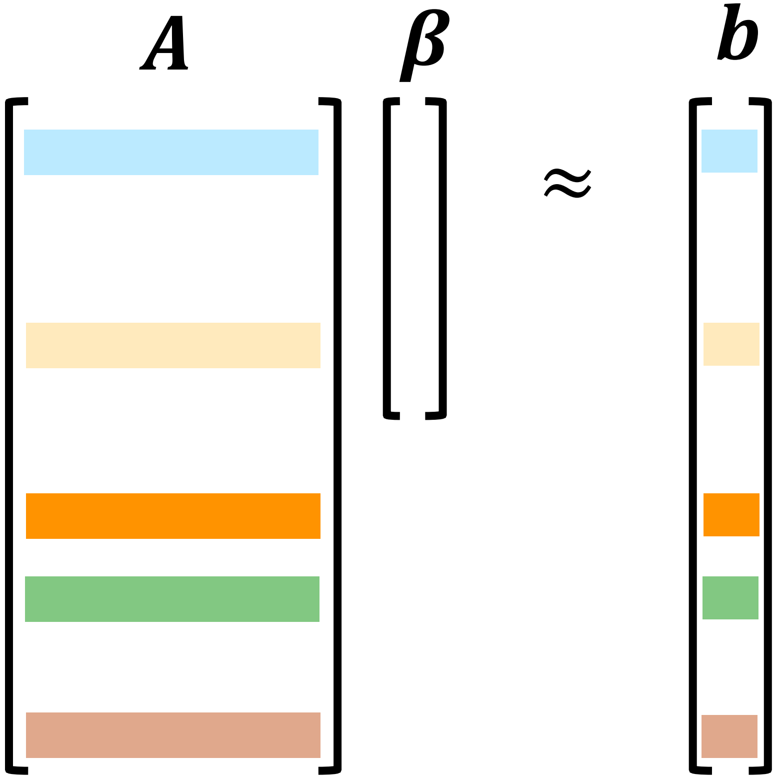
… but samples only used for one Banzhaf value

Maximum Sampling Reuse (MSR): Sample S to either add/subtract $v(S)$ for all i

… but magnitude of $v(S)$ is much larger than magnitude of $v(S \cup \{i\}) - v(S)$

Regression Formulation

$$\boldsymbol{\phi} = \arg\min_{\boldsymbol{\beta}} ||\boldsymbol{A\beta} - \boldsymbol{b}||_2$$

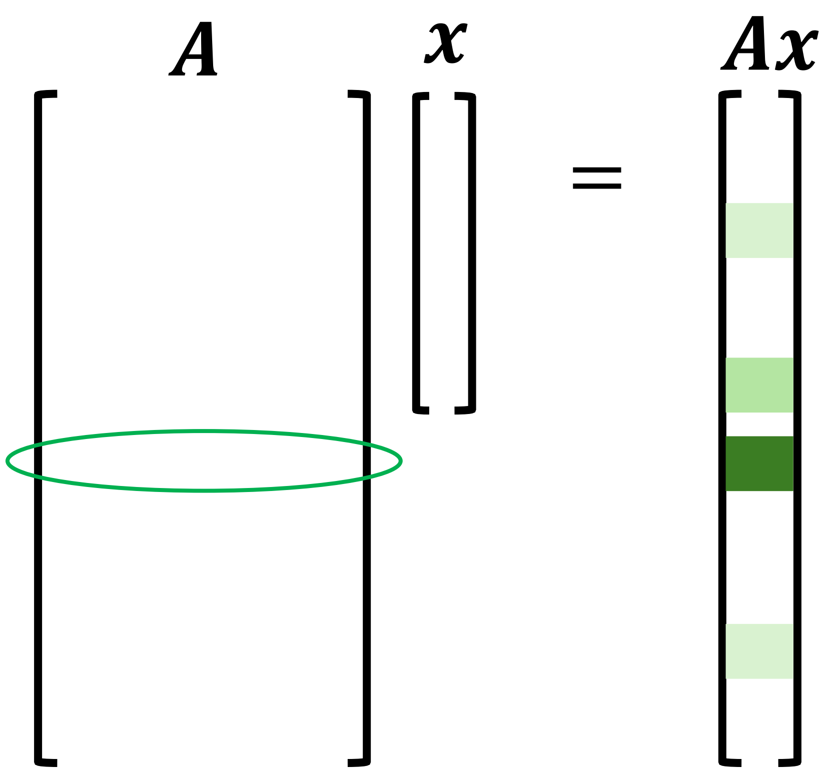


Special case known since 90’s [HH 1992]

Each row/entry corresponds to binary vector

$$\boldsymbol{z} \in \left\{-\frac{1}{2}, \frac{1}{2}\right\}^n$$

Leverage Scores and Banzhaf Values



Row z has "leverage":

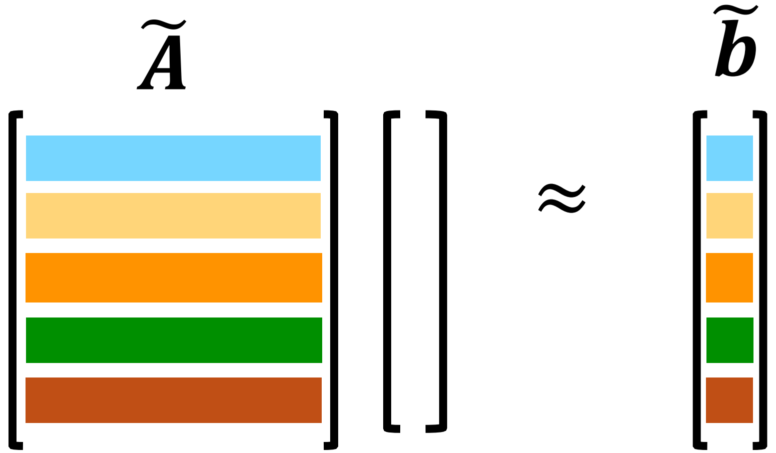
$$\ell_z = \max_{x} \frac{(Ax)^2{}_z}{||Ax||_2^2}$$

$$\ell_z = \frac{n}{2^n}$$

Very similar to weighting in Banzhaf value definition!

Kernel Banzhaf

$$\tilde{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\beta}} ||\tilde{\boldsymbol{A}}\boldsymbol{\beta} - \tilde{\boldsymbol{b}}||_2$$

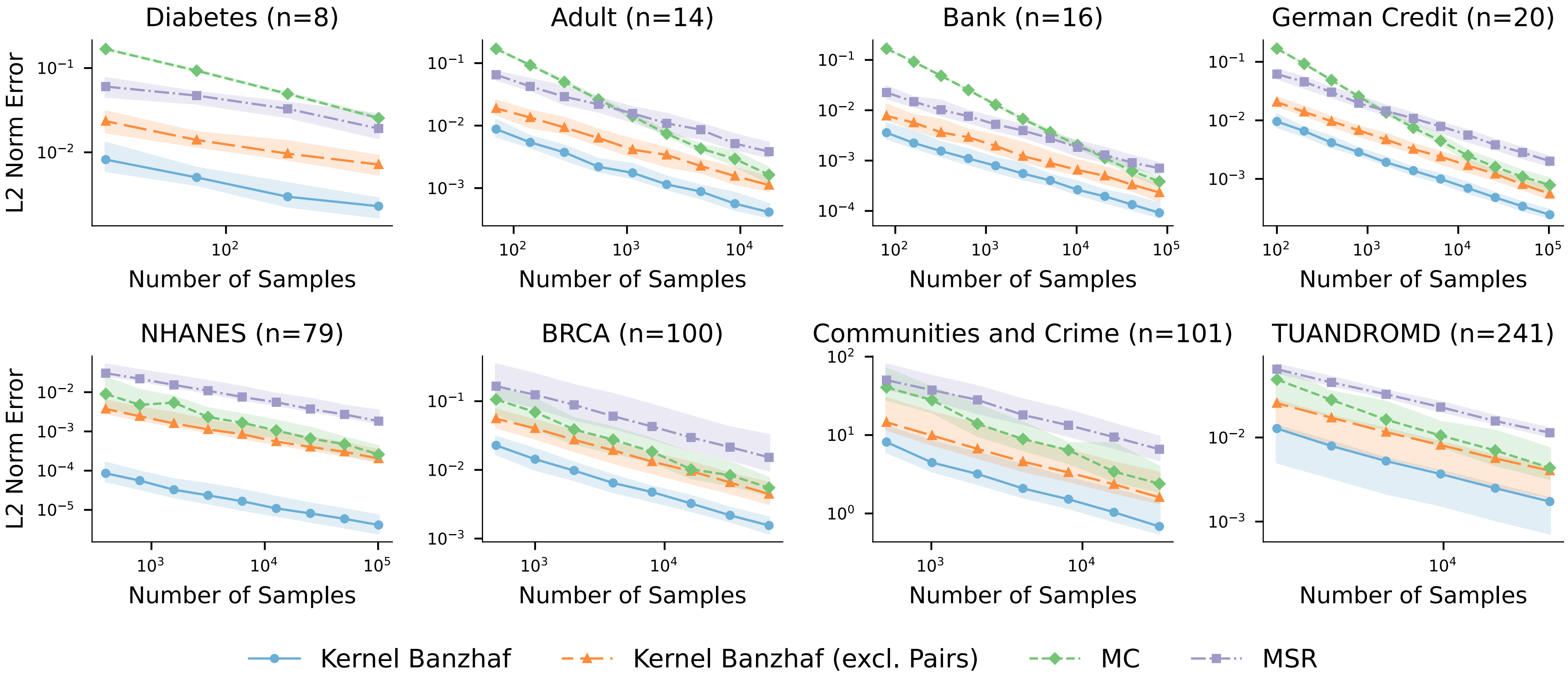


Kernel Banzhaf selects row $\boldsymbol{z}$ with probability proportional to leverage score!

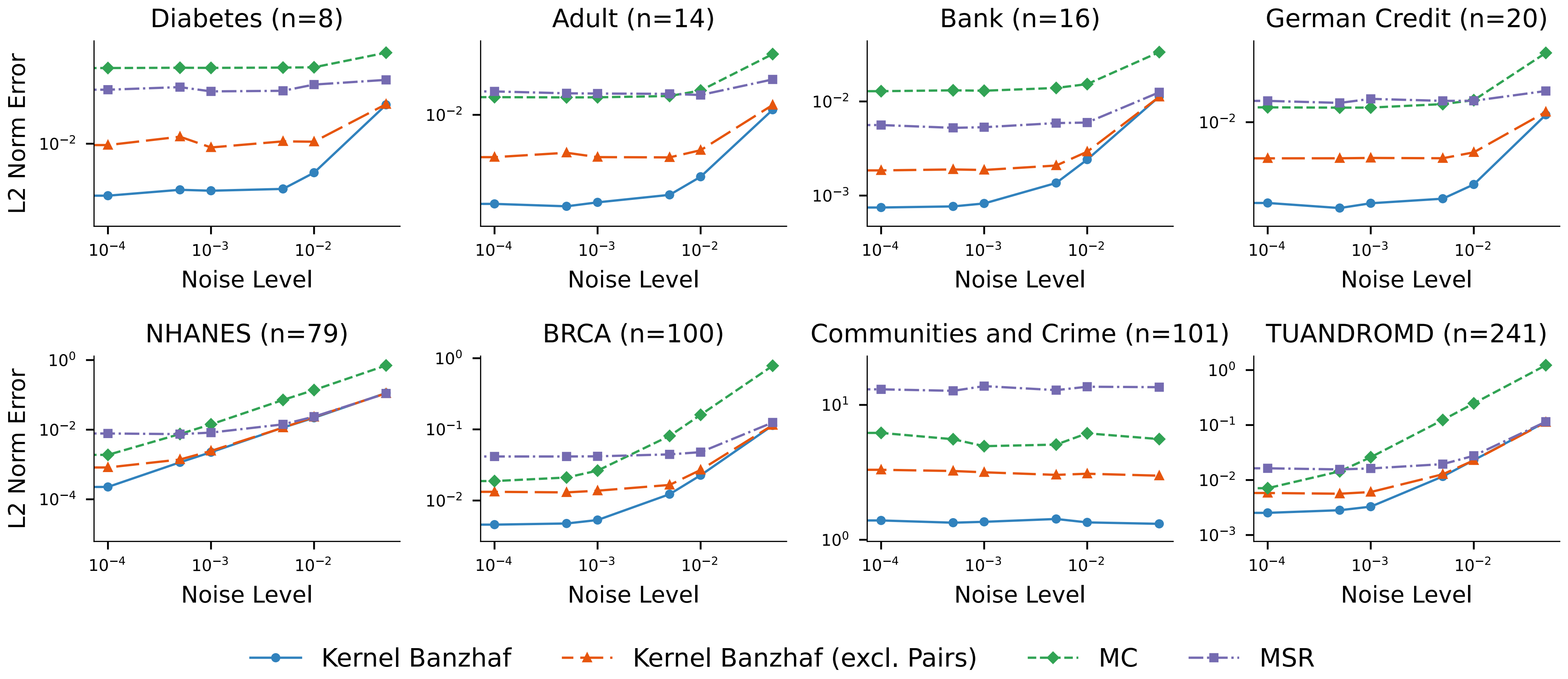
+ Paired Sampling

(But not Bernoulli Sampling)

Accuracy by Number of Samples



Accuracy by Noise



Theoretical Guarantees

As long as $m = O\left(n \log n + \frac{n}{\epsilon}\right)$, the Kernel Banzhaf solution $\tilde{\boldsymbol{\phi}}$ satisfies

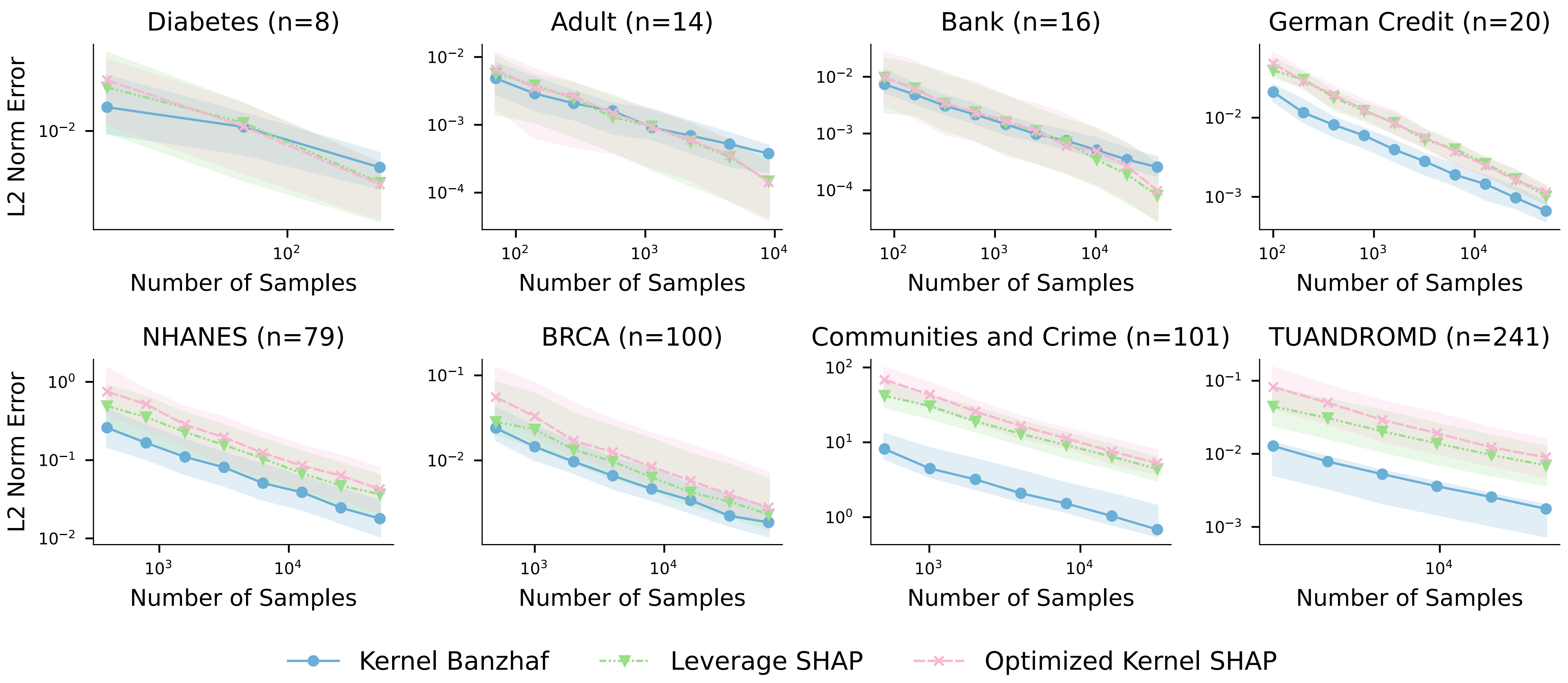
$$||\boldsymbol{A}\tilde{\boldsymbol{\phi}} - \boldsymbol{b}||_2^2 \leq (1+\epsilon)||\boldsymbol{A}\boldsymbol{\phi} - \boldsymbol{b}||_2^2$$

and, for $\gamma = \frac{||\boldsymbol{A}\,\boldsymbol{\phi} - \boldsymbol{b}||_2^2}{||\boldsymbol{A}\,\boldsymbol{\phi}||_2^2} \in [0, \infty)$, $\Updownarrow$ Equivalent for Banzhaf values

$$||\tilde{\boldsymbol{\phi}} - \boldsymbol{\phi}||_2^2 \leq \epsilon\,\gamma||\boldsymbol{\phi}||_2^2$$

with probability 9/10

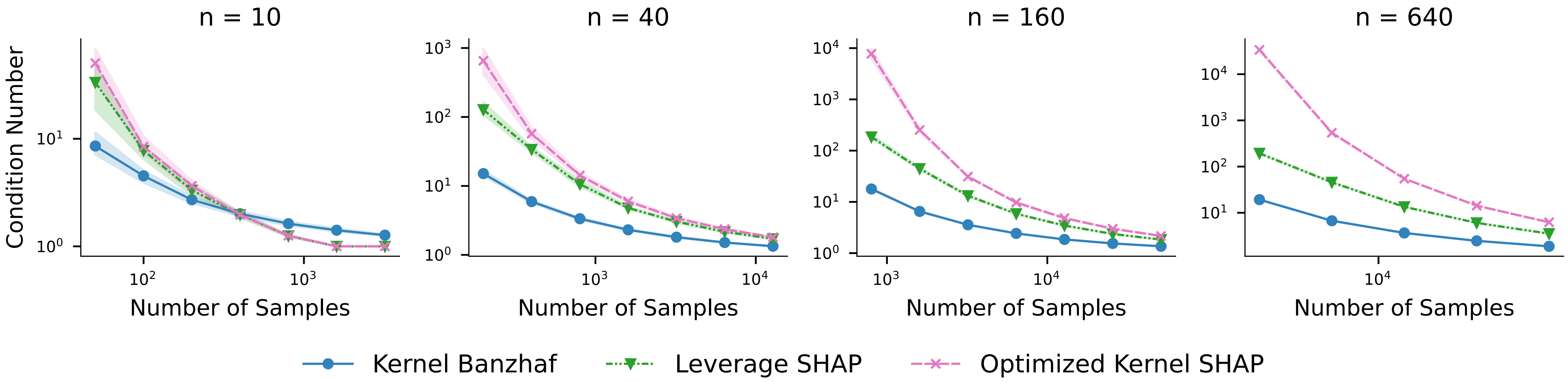
Shapley vs Banzhaf Estimators



Why do Leverage SHAP and Kernel Banzhaf perform differently?

Condition Number of $\widetilde{A}$

Subsampled Banzhaf problem is more well-conditioned.



…an ill-conditioned subsampled problem is not close to the full problem.

Thank you!

Please let me know if you have any questions or comments!

Email: rtealwitter@nyu.edu