Shape Constraints for Set Functions

Andrew Cotter, Maya R. Gupta, Heinrich Jiang, Erez Louidor, James Muller, Taman Narayan, Serena Wang, Tao Zhu

Google Research

Motivation

- **Problem**: Learn a **set function** to predict a label given a *variable-size* set of feature vectors.

$$f : \{x_m \in \mathbb{R}^D\} \to y \in \mathbb{R}$$

Motivation

- **Problem**: Learn a **set function** to predict a label given a *variable-size* set of feature vectors.

$$f : \{x_m \in \mathbb{R}^D\} \to y \in \mathbb{R}$$

- **Use Case:** Classify if a recipe is French given its set of ingredients.

Motivation

- **Problem**: Learn a **set function** to predict a label given a *variable-size* set of feature vectors.

$$f : \{x_m \in \mathbb{R}^D\} \rightarrow y \in \mathbb{R}$$

- **Use Case:** Classify if a recipe is French given its set of ingredients.
- **Use Case:** Estimate label given *compound sparse categorical features*.
    - Predict if a KickStarter campaign will succeed given its **name** "*Superhero Teddy Bear*".

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero Teddy Bear")

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero Teddy Bear")

E(Y | "Superhero") = 0.3

E(Y | "Teddy Bear") = 0.9

Mean ({0.3, 0.9}) = 0.6
Min ({0.3, 0.9}) = 0.3
Max ({0.3, 0.9}) = 0.9
Median ({0.3, 0.9}) = 0.6

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero") = 0.3
Count("Superhero") = 100

E(Y | "Superhero Teddy Bear")

E(Y | "Teddy Bear") = 0.9
Count("Teddy Bear") = 50

$$\frac{\mathbf{0.3}*100 + \mathbf{0.9}*50}{100 + 50}$$

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero Teddy Bear")

E(Y | "Superhero") = 0.3
Count("Superhero") = 100
Size("Superhero") = 1

E(Y | "Teddy Bear") = 0.9
Count("Teddy Bear") = 50
Size("Teddy Bear") = 2

$$\frac{\mathbf{0.3}*100*1 + \mathbf{0.9}*50*2}{100*1 + 50*2}$$

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero Teddy Bear")

E(Y | "Superhero") = 0.3
Count("Superhero") = 100
Size("Superhero") = 1

E(Y | "Teddy Bear") = 0.9
Count("Teddy Bear") = 50
Size("Teddy Bear") = 2

$$\frac{\mathbf{0.3}*100*1 + \mathbf{0.9}*50*2}{100*1 + 50*2}$$

Not flexible enough!

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero Teddy Bear")

E(Y | "Superhero") = 0.3
Count("Superhero") = 100
Size("Superhero") = 1

E(Y | "Teddy Bear") = 0.9
Count("Teddy Bear") = 50
Size("Teddy Bear") = 2

Learned Set Function ({
[0.3, 100, 1],
[0.9, 50, 2]})

$$f(x) = \rho\left(\frac{1}{M(x)}\sum_{m=1}^{M(x)}\phi(x_m)\right)$$

[Deep Sets, Zaheer et al. 2017]

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero Teddy Bear")

E(Y | "Superhero") = 0.3
Count("Superhero") = 100
Size("Superhero") = 1

E(Y | "Teddy Bear") = 0.9
Count("Teddy Bear") = 50
Size("Teddy Bear") = 2

Learned Set Function ({
[0.3, 100, 1],
[0.9, 50, 2]})

Too flexible "over-fit"

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero Teddy Bear")

E(Y | "Superhero") = 0.3
Count("Superhero") = 100
Size("Superhero") = 1

E(Y | "Teddy Bear") = 0.9
Count("Teddy Bear") = 50
Size("Teddy Bear") = 2

Learned Set Function ({
[0.3, 100, 1],
[0.9, 50, 2]})

Set function properties for more regularization and better interpretability

- **Monotonicity**: output does not decrease as E(Y | "Superhero") or E(Y | "Teddy Bear") increases.
- **Conditioning**: *conditioning* feature (count/size) tells how much to trust *primary* feature.

Motivation

How likely a campaign succeeds given its **name** "*Superhero Teddy Bear*"?

E(Y | "Superhero Teddy Bear")

E(Y | "Superhero") = 0.3
Count("Superhero") = 100
Size("Superhero") = 1

E(Y | "Teddy Bear") = 0.9
Count("Teddy Bear") = 50
Size("Teddy Bear") = 2

Learned Set Function ({
[0.3, 100, 1],
[0.9, 50, 2]})

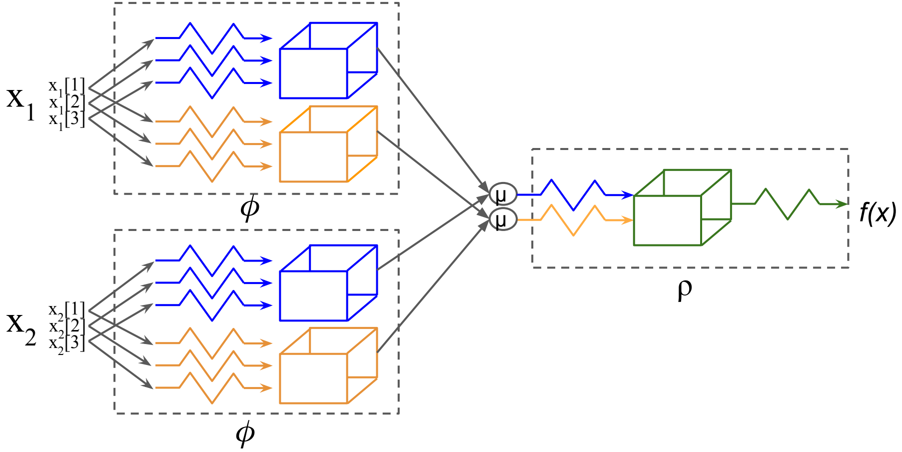
Set function properties for more regularization and better interpretability

- **Monotonicity**: output does not decrease as E(Y | "Superhero") or E(Y | "Teddy Bear") increases.
- **Conditioning**: *conditioning* feature (count/size) tells how much to trust *primary* feature.

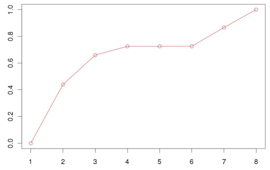
Can we learn flexible set functions while satisfying such properties?

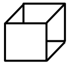
Our approach: DLN with Shape Constraints

Using **Deep Lattice Network (DLN)** (You et al. 2017)

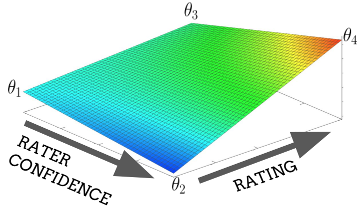


1-D PLF





Multi-D Lattice

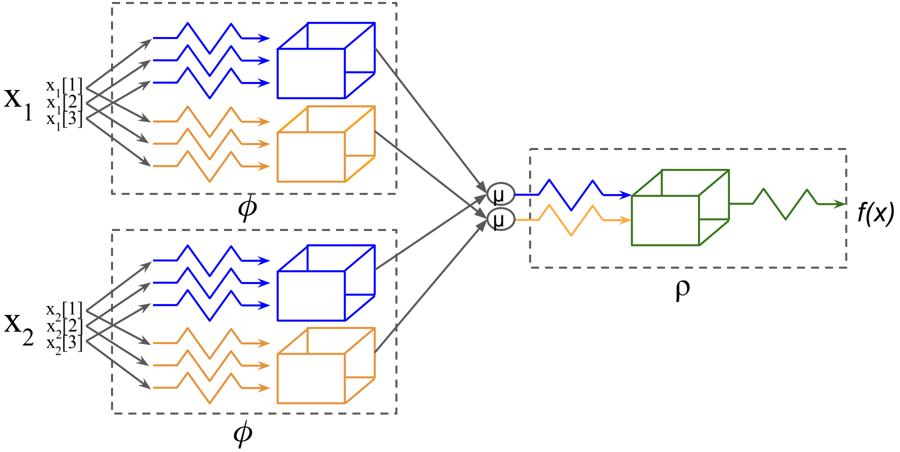


Example lattice function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$

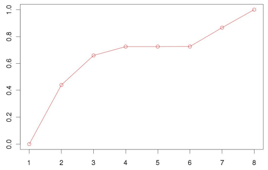
- *Monotonicity*
  $\phi(\theta_3) \geq \phi(\theta_1), \phi(\theta_4) \geq \phi(\theta_2)$
- *Conditioning (Edgeworth)*
  $\phi(\theta_4) - \phi(\theta_2) \geq \phi(\theta_3) - \phi(\theta_1)$
- *Conditioning (Trapezoid)*
  $\phi(\theta_1) \geq \phi(\theta_2), \phi(\theta_4) \geq \phi(\theta_3)$

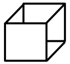
Our approach: DLN with Shape Constraints

Using **Deep Lattice Network (DLN)** (You et al. 2017)

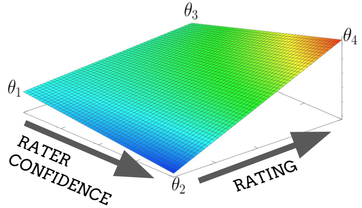


1-D PLF





Multi-D Lattice



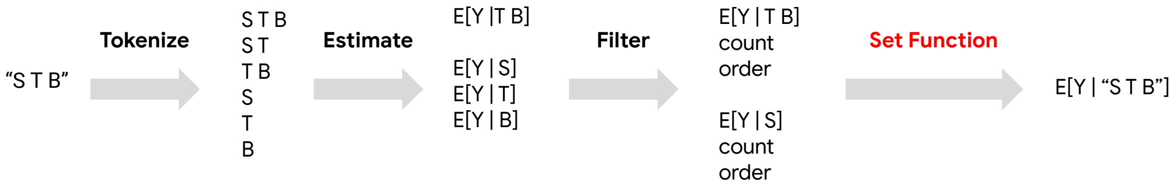
Example lattice function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$

- *Monotonicity*
  $\phi(\theta_3) \geq \phi(\theta_1), \phi(\theta_4) \geq \phi(\theta_2)$
- *Conditioning (Edgeworth)*
  $\phi(\theta_4) - \phi(\theta_2) \geq \phi(\theta_3) - \phi(\theta_1)$
- *Conditioning (Trapezoid)*
  $\phi(\theta_1) \geq \phi(\theta_2), \phi(\theta_4) \geq \phi(\theta_3)$

- Constrained empirical risk minimization based on SGD
- Shapes constraints work for normal functions (set size = 1) using DLN as well

Semantic Feature Engine

- Estimate E(Y | "Superhero Teddy Bear")



- Shape constraints
  - **Monotonicity**: Output monotonically increasing wrt. each ngram estimate.
  - **Conditioning**: Trust more frequent ngrams more...

- **Similar accuracy** as Deep Sets (Zaheer et al. 2017) and DNN, but **with guarantees on model behavior** producing better generalization and more debuggability.

Poster

Tonight 06:30 -- 09:00 PM @ Pacific Ballroom #127