A Logiford structure of measure spaces

Inkee Jung [1] Siu-Cheong Lau [2]

[1] Mathematics PhD student
Boston University

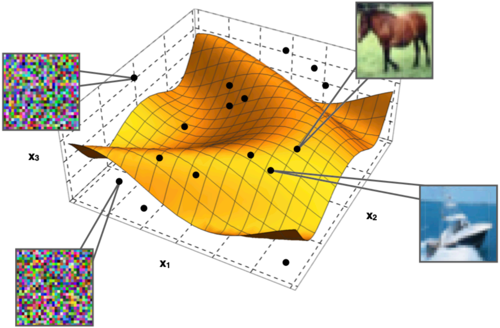
[2] Faculty of Mathematics Department
Boston University

December 18

2024 IMS International Conference on Statistics and Data Science
16 - 19 December 2024, Nice, France

“Manifold” in Data Science

High-dimensional analogue of 2 dimensional surface in $\mathbb{R}^N$

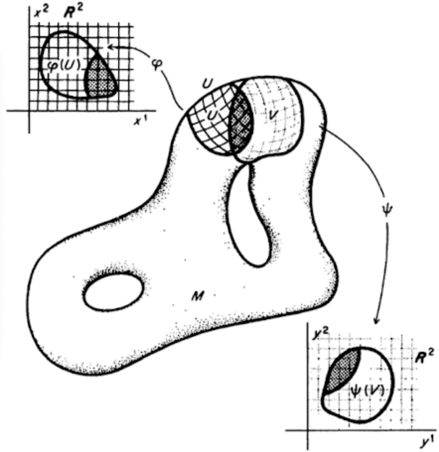


(Image from Sebastian Goldt, Marc Mézard, Florent Krzakala, and Lenka Zdeborová)

Manifold : Local-to-Global Principle

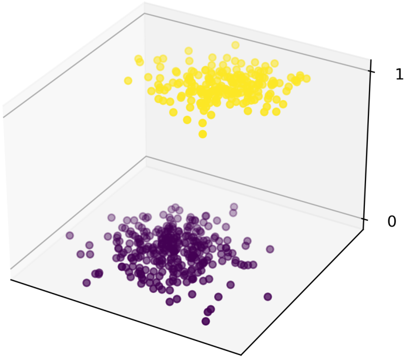
Locally Euclidean Space $(M, \mathcal{U})$ with collection of local data $\mathcal{U} = \{(U_\alpha, \Phi_\alpha)\}$

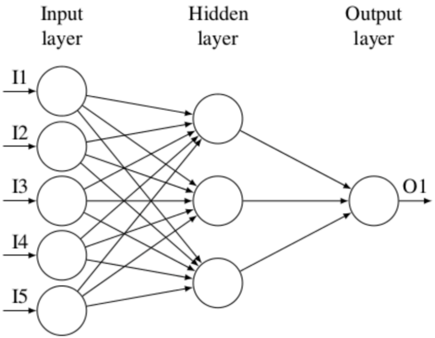
- Modeling Spacetime by Einstein's theory of relativity
- Local-to-Global principle



(Figure based on W. M. Boothby)

Dataset and Neural Network





$$f = \sigma_2 \circ L_2 \circ \sigma_1 \circ L_1$$

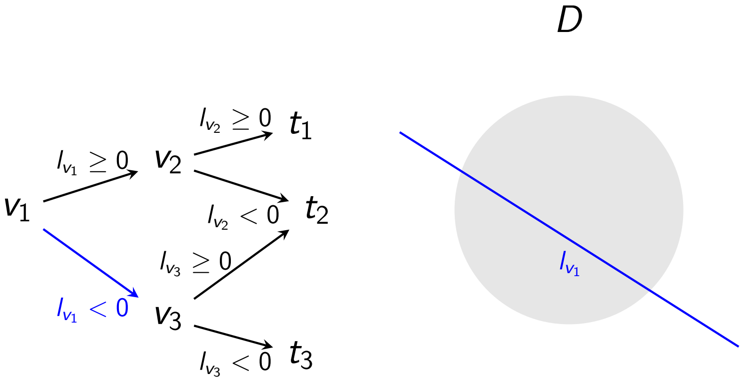
Classification with two classes

- Network models gain tremendous success in describing datasets

Linear Logical Function

Motivated from Neural Network.

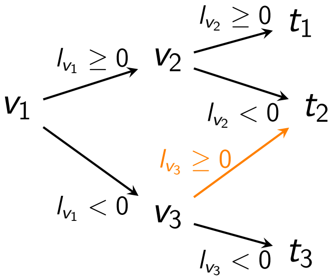
Example: Directed graph G & Set of affine maps $L = \{l_{v_1}, l_{v_2}, l_{v_3}\}$, $D \subset \mathbb{R}^2$

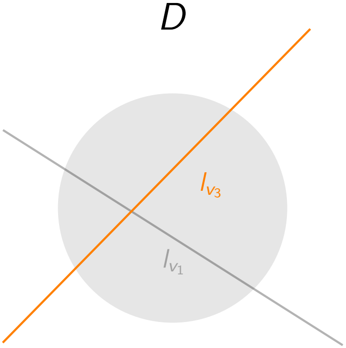


Linear Logical Function

Motivated from Neural Network.

Example: Directed graph G & Set of affine maps $L = \{l_{v_1}, l_{v_2}, l_{v_3}\}$, $D \subset \mathbb{R}^2$

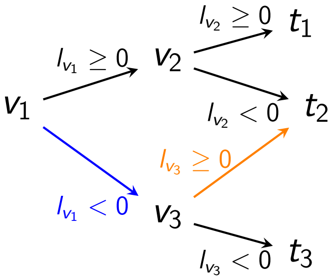


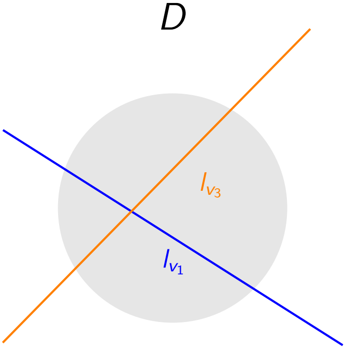


Linear Logical Function

Motivated from Neural Network.

Example: Directed graph G & Set of affine maps $L = \{l_{v_1}, l_{v_2}, l_{v_3}\}$, $D \subset \mathbb{R}^2$

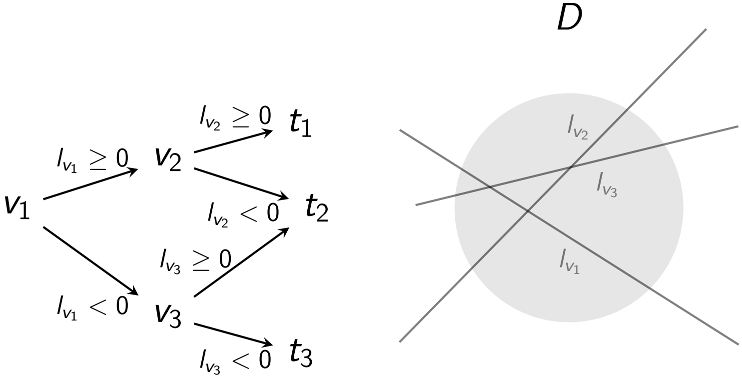




Linear Logical Function

Motivated from Neural Network.

Example: Directed graph G & Set of affine maps $L = \{l_{v_1}, l_{v_2}, l_{v_3}\}$, $D \subset \mathbb{R}^2$



$f : D \to \{t_1, t_2, t_3\}$ is a function defined by G and L.

Definition of Linear Logical Function

- Measurable set $D \subset \mathbb{R}^n$, Finite set T.
- Directed finite graph G without cycle
- Affine maps

$$L = \{l_v : v \text{ is a vertex with more than one outgoing arrows}\}$$

Definition

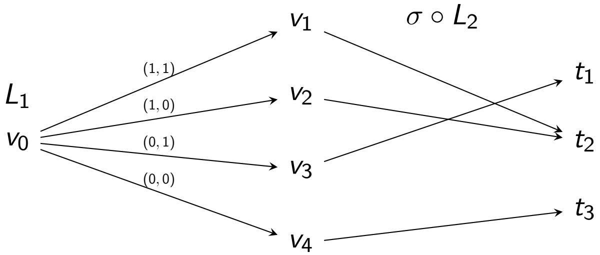
$f_{G,L} : D \to T$ is a linear logical function of (G, L) if $l_v \in L$ are affine linear functions whose chambers in D are one-to-one corresponding to the outgoing arrows of v.
(G, L) is called a linear logical graph.

Linear logical function : Example

Activation map : Step function

$f = \sigma \circ L_2 \circ s \circ L_1$ where

- $L_1 : \mathbb{R}^n \to \mathbb{R}^2$ is affine map and s is a component-wise step function.
- $L_2 : \mathbb{R}^2 \to \mathbb{R}^3$ is affine map and σ is the index-max map.

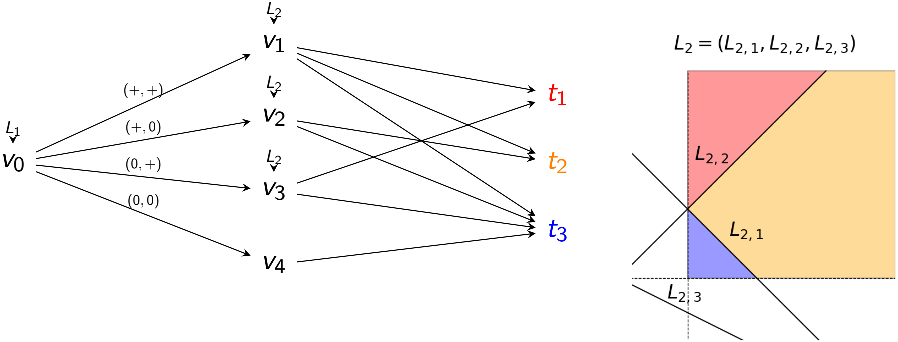


f is a linear logical function with the above graph G and $L = \{L_{v_0}\}$.

Linear logical function : Example

Activation map : ReLu

$f = \sigma \circ L_2 \circ s \circ L_1$, where L_1, L_2 are affine maps and s is a component-wise ReLu function defined as $\mathrm{ReLu}(x) = \max(0, x)$.



f is a linear logical function with the above graph G and $L = \{L_{v_0}, L_{v_1}, L_{v_2}, L_{v_3}\}$.

Fuzzy linear logical function : Example

$f = \sigma \circ L_2 \circ s \circ L_1 : S^n \to S^3$ with SoftMax σ and Sigmoid s, where
$\mathrm{Softmax}(x) = \left(e^{x_k} / \sum_i e^{x_i}\right)_i, \mathrm{Sigmoid}(x) = \left(1 + e^{-x}\right)^{-1}$.

- G is a finite directed graph that has no oriented cycle with exactily one source vertex and target vertex t.

$$v_0 \longrightarrow v_1 \longrightarrow t$$

Fuzzy linear logical function : Example

- Each vertex v of G is equipped with a product of standard simplices P_v, with domain $D = P_{v_0}$.

$$\begin{array}{ccccc} P_{v_0} & & P_{v_1} & & P_t \\ | & & | & & | \\ v_0 & \longrightarrow & v_1 & \longrightarrow & t \end{array}$$

$$P_{v_0} = P_{v_1} = S^n, P_t = S^3.$$

$$S^n = \left\{ (x_0, \ldots, x_n) \in \mathbb{R}^{n+1} : \sum x_i = 1, x_i \geq 0 \right\}.$$

- $p_{a_1} = s \circ L_1 : P_{v_0} \to P_{v_1}, p_{a_2} = \sigma \circ L_2 : P_{v_1} \to P_t$

$$\begin{array}{ccccc} P_{v_0} & & P_{v_1} & & P_t \\ | & & | & & | \\ v_0 & \xrightarrow{a_1} & v_1 & \xrightarrow{a_2} & t \end{array}$$

mapping to the next 'state'.

Fuzzy linear logical function : Definition

- G is a finite directed graph that has no oriented cycle with exactily one source vertex and target vertices $t_1, \ldots, t_K$.
- Each vertex v of G is equipped with a product of standard simplices P_v. Domain D is a subset of P_{v_0}.
- Each arrow a is equipped with a continuous function

$$p_a : P_{s(a)} \to P_{t(a)}$$

  where $s(a), t(a)$ denote the source and target vertices of the arrow a respectively.
- Each vertex v that has more than one outgoing arrows is equipped with affine map l_v whose chambers in P_v are one-to-one corresponding to the outgoing arrows of v.

Given $x \in D$, L and p_a determine a path to a target, and $f_{(G,L,P,p)}(x)$ is defined by the composition of arrow maps along the path.

Universality of Linear logical function

- $D \subset \mathbb{R}^N$ with $\mu(D) < \infty$, where μ is the Lebesgue measure.
- T is finite

Theorem (I. Jung and S.C. Lau)

For any (Lebesgue) measurable function $f : D \to T$, we have a linear logical function that approximates to f.

Universality of Linear logical function

Theorem (I. Jung and S.C. Lau)

For any (Lebesgue) measurable function $f : D \to T$, we have a linear logical function that approximates to f.

Proof of idea

Definition (Lou van den Dries)

A structure S on the real line consists of a boolean algebra S_n of subsets of $\mathbb{R}^n$ for each $n = 0, 1, \ldots,$ such that

- $\{x \in \mathbb{R}^n : x_i = x_j\}, 1 \leq i < j \leq n \in S_n$.
- Closed under Cartesian product.
- Closed under projection $(A \in S_{n+1} \to \pi(A) \in S_n)$.
- $\{(x, y) \in \mathbb{R}^2 : x < y\} \in S_2$.

Universality of Linear logical function

Theorem (I. Jung and S.C. Lau)

For any (Lebesgue) measurable function $f : D \to T$, we have a linear logical function that approximates to f.

Proof of idea

For instance, let ϕ and ψ be 1st order logic formulas on $(x, y) \in X \times Y$.

$$\Phi := \{(x, y) \in X \times Y : \phi(x, y)\}, \Psi := \{(x, y) \in X \times Y : \psi(x, y)\}.$$

$\phi \wedge \psi$ and $\phi \vee \psi$ define $\Phi \cap \Psi$ and $\Phi \cup \Psi$.
$\exists x \phi(x, y)$ defines $\pi_Y(\Phi)$.
$\forall x \phi(x, y)$ defines $Y \setminus \pi_Y(X \times Y \setminus \Phi)$.

Universality of Linear logical function

Theorem (I. Jung and S.C. Lau)

For any (Lebesgue) measurable function $f : D \to T$, we have a linear logical function that approximates to f.

Proof of idea

semilinear set of $\mathbb{R}^n$: Finite unions of

$$\{x \in \mathbb{R}^n : f_1(x) = \cdots = f_k(x), g_1(x) > 0, \ldots, g_l(x) > 0\}$$

with affines f_i and g_j.

Semilinear sets form o-minimal structure, in which every definable subset is a finite union of intervals and points.

Universality of Linear logical function

Theorem (I. Jung and S.C. Lau)

For any (Lebesgue) measurable function $f : D \to T$, we have a linear logical function that approximates to f.

Proof of idea

Lemma

A function $f : D \to T$, where $D \subset \mathbb{R}^n$ and T is a finite set, is semilinear if and only if it is a linear logical function.

Using this lemma, we can approximate f with linear logical functions by constructing approximations of semilinear functions.

Universality of Linear logical function

Theorem (I. Jung and S.C. Lau)

For any (Lebesgue) measurable function $f : D \to T$, we have a linear logical function that approximates to f.

Corollary

There exists a family $\mathcal{L}$ of linear logical functions $L_i : D_i \to T$, where $D_i \subset D$ and $L_i \equiv f|_{D_i}$, such that $D \setminus \bigcup_i D_i$ is measure zero set.

Fuzzy linear logifold

Definition

A fuzzy linear logifold is a tuple $(X, \mathcal{P}, \mathcal{U})$, where $(X, \mathcal{U})$ be a logifold and

- $\mathcal{U}$ is a collection of tuples (ρ_i, ϕ_i, f_i)
- $\rho_i : X \to [0,1]$ describe fuzzy subsets of X with $\sum_i \rho_i \leq 1_X$
- $U_i = \{x \in X : \rho_i(x) > 0\}$ be the support of ρ_i

In classification problems,

- $X = \mathbb{R}^n \times T$
- $\mathcal{P} : X \to [0,1]$ describes how likely an element of $\mathbb{R}^n \times T$ is classified as 'yes'
- ρ_i can be 'generalization performance', or 'constant'.

Example of logifold

$f : (0,1] \to \{0,1\}$ be a function defined as

$$f(x) = \sum_{n=0}^{\infty} \left(\frac{(-1)^n + 1}{2} \right) I_{E_n}(x)$$

where $E_n = (1 - 2^{-n}, 1 - 2^{-n-1}]$.
The graph of $f \subset [0,1) \times \{0,1\}$

with countably many 'jumps' or 'discontinuities' near at $x = 0$.

Practical Aspect : Ensemble Machine Learning

In classification problems, $X = \mathbb{R}^n \times T$ and each model $g_i : X \to T$ with $U_i = X$. Define $G_i : X \times T \to [0, 1]$ by g such that $G_i(x, t) = (g_i(x))_t$. Let N be the total number of classifiers.

- If $\rho_i = \frac{1}{N}$ for any i, then $P : X \times T \to [0, 1]$ is defined by

$$P(x, t) = \sum \rho_i(x) 1_{t_{i,0}(x)}(t)$$

, where $t_{i,0}(x) = \arg\max G_i(x, t)$ denoting 'the answer of g_i', and therefore the system employs **majority voting**.

Practical Aspect : Ensemble Machine Learning

In classification problems, $X = \mathbb{R}^n \times T$ and each model $g_i : X \to T$ with $U_i = X$. Define $G_i : X \times T \to [0, 1]$ by g such that $G_i(x, t) = (g_i(x))_t$. Let N be the total number of classifiers.

- If $\rho_i = \frac{1}{N}$ for any i, then $P : X \times T \to [0, 1]$ is defined by

  $$P(x, t) = \sum \rho_i(x) 1_{t_{i,0}(x)}(t)$$

  , where $t_{i,0}(x) = \arg\max G_i(x, t)$ denoting 'the answer of g_i', and therefore the system employs **majority voting**.
- If $\rho_i = \frac{1}{N}$ for any i, then $P : X \times T \to [0, 1]$ is defined by

  $$P(x, t) = \sum G_i(x, t)$$

  , which is **simple average**.

Practical Aspect : Ensemble Machine Learning

In classification problems, $X = \mathbb{R}^n \times T$ and each model $g_i : X \to T$ with $U_i = X$. Define $G_i : X \times T \to [0,1]$ by g such that $G_i(x,t) = (g_i(x))_t$. Let N be the total number of classifiers.

- If $\rho_i = \frac{1}{N}$ for any i, then $P : X \times T \to [0,1]$ is defined by

  $$P(x,t) = \sum \rho_i(x) 1_{t_{i,0}(x)}(t)$$

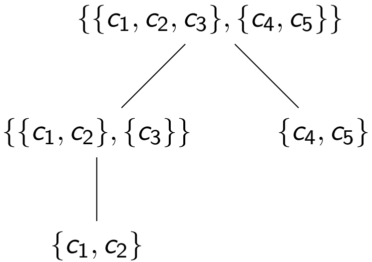
  , where $t_{i,0}(x) = \arg\max G_i(x,t)$ denoting 'the answer of g_i', and therefore the system employs **majority voting**.
- If $\rho_i = \frac{1}{N}$ for any i, then $P : X \times T \to [0,1]$ is defined by

  $$P(x,t) = \sum G_i(x,t)$$

  , which is **simple average**.
- If $\rho_i(x) = \frac{\max g_i(x)}{N}$ then $P(x,t) = \sum \rho_i(x) G_i(x,t)$ be **the weighted average**.

Practical Aspect : Flexible target and Fuzzy Domain

Models	Targets
g_1	$\{\{c_1, c_2, c_3\}, \{c_4, c_5\}\}$
g_2	$\{\{c_1, c_2\}, \{c_3\}\}$
g_3	$\{c_1, c_2\}$
g_4	$\{c_4, c_5\}$



$$\text{Certainty} = \max g(x)$$

$$\text{Certain domain} = \{\text{certainty} > \alpha\}, \quad \alpha = \text{threshold}$$

Then compute the precisions for each target of g on *the Certain domain*, which contribute to $\rho_i(x)$ along with the *target tree*.

Experimental Result 1

Dataset : CIFAR10
Six Simple CNN structure models trained on CIFAR10 (56.45% in average)
ResNet20 structure model trained on CIFAR10 (85.96%)
Simple average : 62.55%
Majority voting provides 58.72%.
Our logifold formulation : 84.86%

Experimental Result 2

dataset : CIFAR10, MNIST, Fashion MNIST (resized to 32*32*3 pixels)

- Filters are models classifying coarse targets. It only classify given data into three classes ; CIFAR10, MNIST, and Fashion MNIST.
- Models only classifying either CIFAR10, MNIST, or Fashion MNIST.

Single model classifying 30 classes : 76.41% in average.
Simple average of models classifying 30 classes : 82.35%
Our logifold formulation : 94.94%.