Sequential Dependence Decomposition & Feature Learning

Xiangxiang Xu
Joint work with Prof. Lizhong Zheng

Allerton Conference 2023

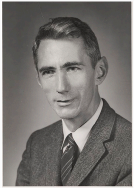
Sequences

- 

- Videos
- `exampe of dependnce in plai tetx`

Guessing Next Letter

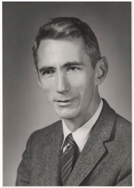
1. E $\Longrightarrow$ solve E ?



Claude E Shannon. A Mathematical Theory of Communication. 1948.

Guessing Next Letter

1. E ⟹ solve E ?

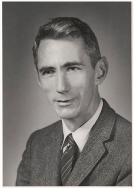


- count the transition: DEPENDENCE
  - "E" followed by N: 0.5, P: 0.25, 'SPC': 0.25

Claude E Shannon. A Mathematical Theory of Communication. 1948.

Shannon's Experiments [1948]

Guessing Next Letter



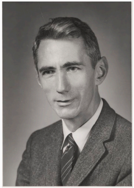
1. E $\Longrightarrow$ solve E ?
2. E N $\Longrightarrow$ solve N ?

- count the transition: DEPENDENCE
  - "E" followed by N: 0.5, P: 0.25, 'SPC': 0.25
- predict based on the last letter and digram frequency

Claude E Shannon. A Mathematical Theory of Communication. 1948.

Shannon's Experiments [1948]

Guessing Next Letter



1. E $\Longrightarrow$ solve E ?
2. E N $\Longrightarrow$ solve N ?

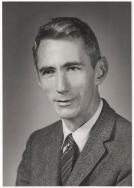
- count the transition: DEPENDENCE
  - "E" followed by N: 0.5, P: 0.25, 'SPC': 0.25
- predict based on the last letter and digram frequency
- generate a Markov chain of letters

Claude E Shannon. A Mathematical Theory of Communication. 1948.

Shannon's Experiments [1948]

Guessing Next Letter

1. E $\Longrightarrow$ solve E ?
2. E N $\Longrightarrow$ solve N ?



- count the transition: DEPENDENCE
  - "E" followed by N: 0.5, P: 0.25, 'SPC': 0.25
- predict based on the last letter and digram frequency
- generate a Markov chain of letters
  - word-level construction
  - trigram, *n*-gram (higher order Markov chains)

Claude E Shannon. A Mathematical Theory of Communication. 1948.

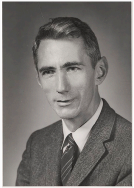
Guessing Next Letter

Second-order approximation (digram structure as in English).

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

Third-order approximation (trigram structure as in English).

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE



Claude E Shannon. A Mathematical Theory of Communication. 1948.

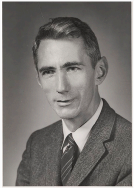
Guessing Next Letter

Second-order approximation (digram structure as in English).

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

Third-order approximation (trigram structure as in English).

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE



Operations

- frequency tables

264 SECRET AND URGENT

TABLE XII

English Trigrams

Figures represent approximate frequencies for 20,000 words, but are more important in their relation to one another.

A		agn	2	ame	53	aro	7
Aam	1	ago	6	amh	1	arp	1
		agr	8	ami	6	arr	19
Aba	6	agu	2	aml	1	ars	46
abd	1			amn	1	art	48
abe	1	Ahe	2	amo	8	arv	3
abi	3	aho	1	amp	10	ary	34
abl	39	ahu	1	ams	6		
abo	28	ahy	1	amy	1		
abr	1					Asa	1
abs	1	Aid	24	Ana	6	asc	3
		aig	3	anc	39	ase	20

Fletcher Pratt. Secret and Urgent. 1939.

Claude E Shannon. A Mathematical Theory of Communication. 1948.

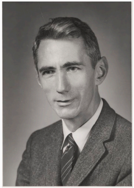
Guessing Next Letter

Second-order approximation (digram structure as in English).

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

Third-order approximation (trigram structure as in English).

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE



Operations

- frequency tables
- *opens a book...*
  select a letter at random

264 SECRET AND URGENT

TABLE XII

ENGLISH TRIGRAMS

Figures represent approximate frequencies for 20,000 words, but are more important in their relation to one another.

A		agn	2	ame	53	aro	7
Aam	1	ago	6	amh	1	arp	1
		agr	8	ami	6	arr	19
Aba	6	agu	2	aml	1	ars	46
abd	1			amn	1	art	48
abe	1	Ahe	2	amo	8	arv	3
abi	3	aho	1	amp	10	ary	34
abl	39	ahu	1	ams	6		
abo	28	ahy	1	amy	1		
abr	1					Asa	1
abs	1	Aid	24	Ana	6	asc	3
		aig	3	anc	39	ase	20

Fletcher Pratt. Secret and Urgent. 1939.

Claude E Shannon. A Mathematical Theory of Communication. 1948.

- n-gram table on $\mathcal{X} \implies |\mathcal{X}|^n$ entries
- “book”: may not contain all combinations

Markov Approximation

- n-gram table on $\mathcal{X} \implies |\mathcal{X}|^n$ entries
- "book": may not contain all combinations

Deep Learning Solution

- entries are structured

- n-gram table on $\mathcal{X} \implies |\mathcal{X}|^n$ entries
- "book": may not contain all combinations

Deep Learning Solution

- entries are structured
- have "table" parameterized
  - deep neural nets: LSTM, Transformer
- learn parameters from "books"

- n-gram table on $\mathcal{X} \implies |\mathcal{X}|^n$ entries
- "book": may not contain all combinations

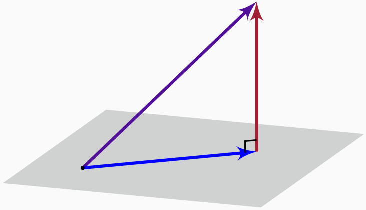
Deep Learning Solution

- entries are structured
- have "table" parameterized
  - deep neural nets: LSTM, Transformer
- learn parameters from "books"

Hidden Parts?

- black-box features
- dependence structure, e.g., the "order"

Feature Geometry



$(X, Y) \sim P_{X,Y}$ variables of interest, e.g., input and output pair

Feature Space

- Features of X: $\mathcal{F}_{\mathcal{X}} \triangleq \{\mathcal{X} \to \mathbb{R}\}$

$(X, Y) \sim P_{X,Y}$ variables of interest, e.g., input and output pair

Feature Space

- Features of X: $\mathcal{F}_{\mathcal{X}} \triangleq \{\mathcal{X} \to \mathbb{R}\}$
- Inner Product $\langle f_1, f_2 \rangle \triangleq \mathbb{E}_{P_X}[f_1(X) f_2(X)]$

Feature Geometry

$(X, Y) \sim P_{X,Y}$ variables of interest, e.g., input and output pair

Feature Space

- Features of X: $\mathcal{F}_{\mathcal{X}} \triangleq \{\mathcal{X} \to \mathbb{R}\}$
- Inner Product $\langle f_1, f_2 \rangle \triangleq \mathbb{E}_{P_X}[f_1(X) f_2(X)]$
  - Induced geometry: norm, orthogonality, projection, etc.

$(X, Y) \sim P_{X,Y}$ variables of interest, e.g., input and output pair

Feature Space

- Features of X: $\mathcal{F}_{\mathcal{X}} \triangleq \{\mathcal{X} \to \mathbb{R}\}$
- Inner Product $\langle f_1, f_2 \rangle \triangleq \mathbb{E}_{P_X}[f_1(X) f_2(X)]$
  - Induced geometry: norm, orthogonality, projection, etc.
  - P_X: Metric (distribution)

Feature Geometry

$(X, Y) \sim P_{X,Y}$ variables of interest, e.g., input and output pair

Feature Space

- Features of X: $\mathcal{F}_{\mathcal{X}} \triangleq \{\mathcal{X} \to \mathbb{R}\}$
- Inner Product $\langle f_1, f_2 \rangle \triangleq \mathbb{E}_{P_X}[f_1(X) f_2(X)]$
  - Induced geometry: norm, orthogonality, projection, etc.
  - P_X: Metric (distribution)
- Inner product spaces $\mathcal{F}_{\mathcal{X}}[P_X], \mathcal{F}_{\mathcal{Y}}[P_Y]$

Feature Geometry

$(X, Y) \sim P_{X,Y}$ variables of interest, e.g., input and output pair

Feature Space

- Features of X: $\mathcal{F}_{\mathcal{X}} \triangleq \{\mathcal{X} \to \mathbb{R}\}$
- Inner Product $\langle f_1, f_2 \rangle \triangleq \mathbb{E}_{P_X}[f_1(X) f_2(X)]$
  - Induced geometry: norm, orthogonality, projection, etc.
  - P_X: Metric (distribution)
- Inner product spaces $\mathcal{F}_{\mathcal{X}}[P_X], \mathcal{F}_{\mathcal{Y}}[P_Y]$

Joint functions $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}}[P_X P_Y]$

- Metric $P_X P_Y$ helps decouple the dependence

Feature Geometry

$(X, Y) \sim P_{X,Y}$ variables of interest, e.g., input and output pair

Feature Space

- Features of X: $\mathcal{F}_{\mathcal{X}} \triangleq \{\mathcal{X} \to \mathbb{R}\}$
- Inner Product $\langle f_1, f_2 \rangle \triangleq \mathbb{E}_{P_X}[f_1(X) f_2(X)]$
  - Induced geometry: norm, orthogonality, projection, etc.
  - P_X: Metric (distribution)
- Inner product spaces $\mathcal{F}_{\mathcal{X}}[P_X], \mathcal{F}_{\mathcal{Y}}[P_Y]$

Joint functions $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}}[P_X P_Y]$

- Metric $P_X P_Y$ helps decouple the dependence
- Product of $f \in \mathcal{F}_{\mathcal{X}}, g \in \mathcal{F}_{\mathcal{Y}}$: $f \otimes g \colon (x, y) \mapsto f(x) g(y)$

$(X, Y) \sim P_{X,Y}$ variables of interest, e.g., input and output pair

Feature Space

- Features of X: $\mathcal{F}_{\mathcal{X}} \triangleq \{\mathcal{X} \to \mathbb{R}\}$
- Inner Product $\langle f_1, f_2 \rangle \triangleq \mathbb{E}_{P_X}[f_1(X)f_2(X)]$
  - Induced geometry: norm, orthogonality, projection, etc.
  - P_X: Metric (distribution)
- Inner product spaces $\mathcal{F}_{\mathcal{X}}[P_X], \mathcal{F}_{\mathcal{Y}}[P_Y]$

Joint functions $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}}[P_X P_Y]$

- Metric $P_X P_Y$ helps decouple the dependence
- Product of $f \in \mathcal{F}_{\mathcal{X}}, g \in \mathcal{F}_{\mathcal{Y}}$: $f \otimes g \colon (x, y) \mapsto f(x)g(y)$
- $f \otimes g \triangleq \sum_{i=1}^{k} f_i \otimes g_i$ for k-dim features

$f = (f_1, \ldots, f_k)$
$g = (g_1, \ldots, g_k)$

Canonical Dependence Kernel (CDK)

$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

Canonical Dependence Kernel (CDK)

$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

- ▷ $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X}\times\mathcal{Y}}$
- ▷ $\|\mathfrak{i}_{X;Y}\| = 0$ iff $X \perp\!\!\!\perp Y$

Canonical Dependence Kernel (CDK)

$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

- $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X} \times \mathcal{Y}}$
- $\|\mathfrak{i}_{X;Y}\| = 0$ iff $X \perp\!\!\!\perp Y$

Learning CDK from Features

$$\mathscr{H}(f, g) \triangleq \frac{1}{2}\left(\|\mathfrak{i}_{X;Y}\|^2 - \|\mathfrak{i}_{X;Y} - f \otimes g\|^2\right)$$

Canonical Dependence Kernel (CDK)

$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

- ▷ $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X} \times \mathcal{Y}}$
- ▷ $\|\mathfrak{i}_{X;Y}\| = 0$ iff $X \perp\!\!\!\perp Y$

Learning CDK from Features

$$\mathscr{H}(f, g) \triangleq \frac{1}{2}\left(\|\mathfrak{i}_{X;Y}\|^2 - \|\mathfrak{i}_{X;Y} - f \otimes g\|^2\right)$$

- learning $\mathfrak{i}_{X;Y}$ by maximizing the H-score

Canonical Dependence Kernel (CDK)

$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

- ▷ $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X} \times \mathcal{Y}}$
- ▷ $\|\mathfrak{i}_{X;Y}\| = 0$ iff $X \perp\!\!\!\perp Y$

Learning CDK from Features

$$\mathscr{H}(f, g) \triangleq \frac{1}{2}\left(\|\mathfrak{i}_{X;Y}\|^2 - \|\mathfrak{i}_{X;Y} - f \otimes g\|^2\right)$$

- learning $\mathfrak{i}_{X;Y}$ by maximizing the H-score
- efficiently computable from data samples $\{(x_i, y_i)\}_{i=1}^n$

$$\mathscr{H} \overset{k=1}{=} \mathrm{cov}(f(X), g(Y)) - \frac{1}{2} \cdot \mathbb{E}\left[f^2(X)\right] \cdot \mathbb{E}\left[g^2(Y)\right]$$

Canonical Dependence Kernel (CDK)

$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

- $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X}\times\mathcal{Y}}$
- $\|\mathfrak{i}_{X;Y}\| = 0$ iff $X \perp\!\!\!\perp Y$

Learning CDK from Features

$$\mathscr{H}(f,g) \triangleq \frac{1}{2}\left(\|\mathfrak{i}_{X;Y}\|^2 - \|\mathfrak{i}_{X;Y} - f \otimes g\|^2\right)$$

- learning $\mathfrak{i}_{X;Y}$ by maximizing the H-score
- efficiently computable from data samples $\{(x_i, y_i)\}_{i=1}^n$

$$\mathscr{H} \overset{k=1}{=} \underbrace{\mathrm{cov}(f(X), g(Y))}_{\frac{1}{n}\sum_i f(x_i)g(x_i)} - \frac{1}{2} \cdot \underbrace{\mathbb{E}\left[f^2(X)\right]}_{\frac{1}{n}\sum_i f^2(x_i)} \cdot \mathbb{E}\left[g^2(Y)\right]$$

Canonical Dependence Kernel (CDK)

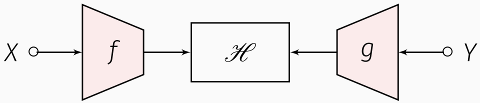
$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

- $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X}\times\mathcal{Y}}$
- $\|\mathfrak{i}_{X;Y}\| = 0$ iff $X \perp\!\!\!\perp Y$

Learning CDK from Features

$$\mathscr{H}(f,g) \triangleq \frac{1}{2}\left(\|\mathfrak{i}_{X;Y}\|^2 - \|\mathfrak{i}_{X;Y} - f\otimes g\|^2\right)$$

- apply neural feature extractors to process data



Canonical Dependence Kernel (CDK)

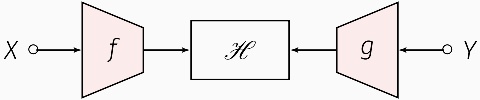
$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

- $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X}\times\mathcal{Y}}$
- $\|\mathfrak{i}_{X;Y}\| = 0$ iff $X \perp\!\!\!\perp Y$

Learning CDK from Features

$$\mathscr{H}(f,g) \triangleq \frac{1}{2}\left(\|\mathfrak{i}_{X;Y}\|^2 - \|\mathfrak{i}_{X;Y} - f \otimes g\|^2\right)$$

- apply neural feature extractors to process data



- maximize $\mathscr{H}(f,g) \implies f \otimes g = \mathfrak{i}_{X;Y}$

Canonical Dependence Kernel (CDK)

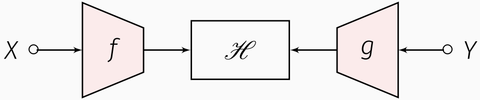
$$\mathfrak{i}_{X;Y} = \frac{P_{X,Y}}{P_X P_Y} - 1$$

- $\mathfrak{i}_{X;Y} \in \mathcal{F}_{\mathcal{X}\times\mathcal{Y}}$
- $\|\mathfrak{i}_{X;Y}\| = 0$ iff $X \perp\!\!\!\perp Y$

Learning CDK from Features

$$\mathscr{H}(f,g) \triangleq \frac{1}{2}\left(\|\mathfrak{i}_{X;Y}\|^2 - \|\mathfrak{i}_{X;Y} - f \otimes g\|^2\right)$$

- apply neural feature extractors to process data



- maximize $\mathscr{H}(f,g) \quad \Longrightarrow \quad f \otimes g = \mathfrak{i}_{X;Y}$
  - compute $\|\mathfrak{i}_{X;Y}\|$ from features
  - prediction/estimation
    $P_{X|Y}(x|y) = P_X(x) \cdot [1 + f(x) \cdot g(y)]$

One More Variable

Learn X based on

- Y: $\mathbf{i}_{X;Y}$

One More Variable

Learn X based on

- Y: $\mathfrak{i}_{X;Y}$
- (Y, Z): $\mathfrak{i}_{X;Y,Z}$

Space $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}[P_X P_{Y,Z}]$

One More Variable

Learn X based on

- Y: $\mathfrak{i}_{X;Y}$
- (Y, Z): $\mathfrak{i}_{X;Y,Z}$

Space $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}[P_X P_{Y,Z}]$

Contribution of Z

$$\mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$$

One More Variable

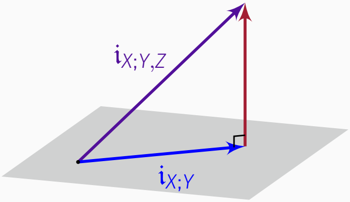
Learn X based on

- Y: $\mathfrak{i}_{X;Y}$
- (Y, Z): $\mathfrak{i}_{X;Y,Z}$

Contribution of Z

$$\mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$$

Space $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}[P_X P_{Y,Z}]$



Markov Plane

- $\mathfrak{i}$: $X - Y - Z$

One More Variable

Learn X based on

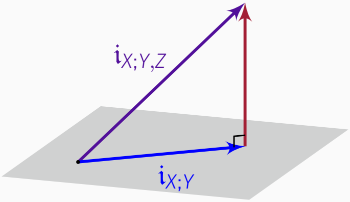
- Y: $\mathfrak{i}_{X;Y}$
- (Y, Z): $\mathfrak{i}_{X;Y,Z}$

Space $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}[P_X P_{Y,Z}]$

Contribution of Z

$$\mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$$

$\mathfrak{i}_{X;Y}$ Markov Component



Markov Plane

- $\mathfrak{i}$: $X - Y - Z$

One More Variable

Learn X based on

- Y: $\mathfrak{i}_{X;Y}$
- (Y, Z): $\mathfrak{i}_{X;Y,Z}$

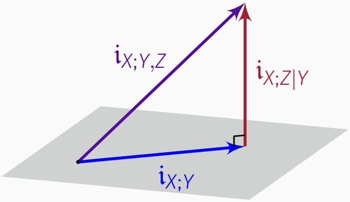
Space $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}[P_X P_{Y,Z}]$

Contribution of Z

$$\mathfrak{i}_{X;Z|Y} \triangleq \mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$$

$\mathfrak{i}_{X;Y}$ Markov Component

$\mathfrak{i}_{X;Z|Y}$ Conditional Dependence



Markov Plane

- $\mathfrak{i}$: $X - Y - Z$

One More Variable

Learn X based on

- Y: $\mathfrak{i}_{X;Y}$
- (Y, Z): $\mathfrak{i}_{X;Y,Z}$

Space $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}[P_X P_{Y,Z}]$

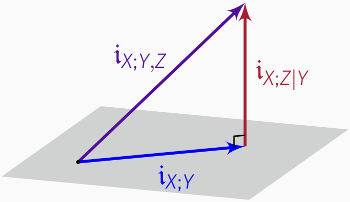
Contribution of Z

$$\mathfrak{i}_{X;Z|Y} \triangleq \mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$$

$\mathfrak{i}_{X;Y}$ Markov Component

$\mathfrak{i}_{X;Z|Y}$ Conditional Dependence

- $\|\mathfrak{i}_{X;Z|Y}\| = 0$ iff $X \perp\!\!\!\perp Z|Y$



Markov Plane

- $\mathfrak{i}$: $X - Y - Z$

Learn X based on

- Y: $\mathfrak{i}_{X;Y}$
- (Y, Z): $\mathfrak{i}_{X;Y,Z}$

Space $\mathcal{F}_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}[P_X P_{Y,Z}]$

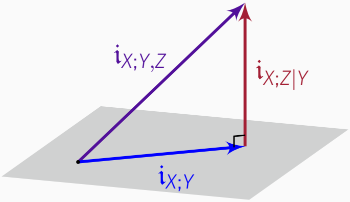
Contribution of Z

$$\mathfrak{i}_{X;Z|Y} \triangleq \mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$$

$\mathfrak{i}_{X;Y}$ Markov Component

$\mathfrak{i}_{X;Z|Y}$ Conditional Dependence

- $\|\mathfrak{i}_{X;Z|Y}\| = 0$ iff $X \perp\!\!\!\perp Z | Y$



Markov Plane

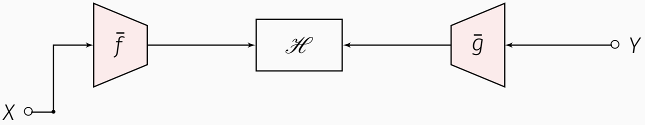
- $\mathfrak{i}$: $X - Y - Z$

$$\|\mathfrak{i}_{X;Y,Z}\|^2 = \|\mathfrak{i}_{X;Y}\|^2 + \|\mathfrak{i}_{X;Z|Y}\|^2$$

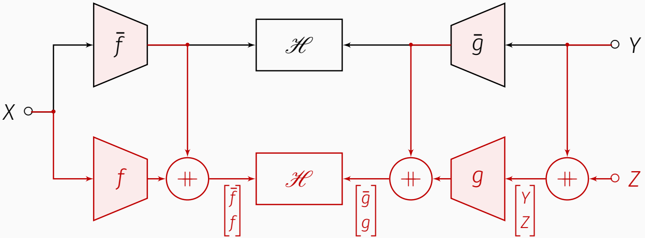
Learning Conditional Dependence

$\mathfrak{i}_{X;Z|Y} = \mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$: dependence Y cannot capture

$\mathfrak{i}_{X;Z|Y} = \mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$: dependence Y cannot capture

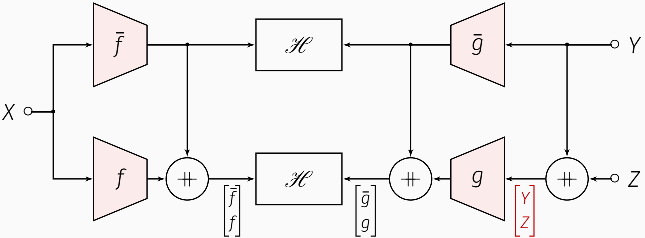


$\mathfrak{i}_{X;Z|Y} = \mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$: dependence Y cannot capture



Learning Conditional Dependence

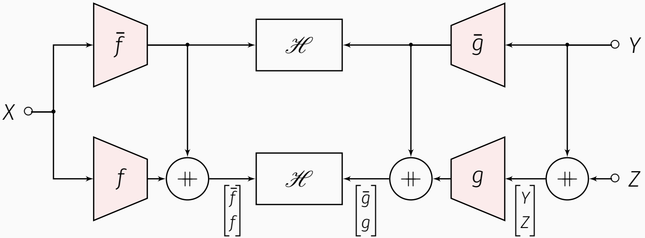
$\mathfrak{i}_{X;Z|Y} = \mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$: dependence Y cannot capture



- nesting: separate conditional dependence from the joint

Learning Conditional Dependence

$\mathfrak{i}_{X;Z|Y} = \mathfrak{i}_{X;Y,Z} - \mathfrak{i}_{X;Y}$: dependence Y cannot capture



- nesting: separate conditional dependence from the joint
- training: maximize the sum of two H-scores
  - optimal solution: $\bar{f} \otimes \bar{g} = \mathfrak{i}_{X;Y}$, $f \otimes g = \mathfrak{i}_{X;Z|Y}$
  - measure the strength of conditional dependence

Sequential Dependence Decomposition

$$\cdots \quad X_{-3} \quad X_{-2} \quad X_{-1} \quad X_0 \quad \cdots$$

$X_{-1} \quad X_0 \quad \cdots$

"Looking Back"

Learn X_0 based on

- previous state: $\mathfrak{i}_{X_0;X_{-1}}$

$X_{-2} \quad X_{-1} \quad X_0 \quad \cdots$

"Looking Back"

Learn X_0 based on

- previous state: $\mathfrak{i}_{X_0;X_{-1}}$
- past 2 states: $\mathfrak{i}_{X_0;(X_{-1},X_{-2})}$

$\cdots \quad X_{-3} \quad X_{-2} \quad X_{-1} \quad X_0 \quad \cdots$

"Looking Back"

Learn X_0 based on

- previous state: $\mathfrak{i}_{X_0;X_{-1}}$
- past 2 states: $\mathfrak{i}_{X_0;(X_{-1},X_{-2})}$

$\vdots$

- past n states: $\mathfrak{i}_{X_0;(X_{-1},\cdots,X_{-n})}$

$$\cdots \quad X_{-3} \quad X_{-2} \quad X_{-1} \quad X_0 \quad \cdots$$

"Looking Back"

Learn X_0 based on

- previous state: $\mathfrak{i}_{X_0;X_{-1}}$
- past 2 states: $\mathfrak{i}_{X_0;(X_{-1},X_{-2})}$

  $\vdots$

- past n states: $\mathfrak{i}_{X_0;(X_{-1},\cdots,X_{-n})}$

- Gain from ℓ-th layer $\quad \mathfrak{i}_\ell \triangleq \mathfrak{i}_{X_0;X_{-\ell}|(X_{-1},\cdots,X_{-l+1})}$
  - conditional dependence at lag ℓ

$$\cdots \quad X_{-3} \quad X_{-2} \quad X_{-1} \quad X_0 \quad \cdots$$

"Looking Back"

Learn X_0 based on

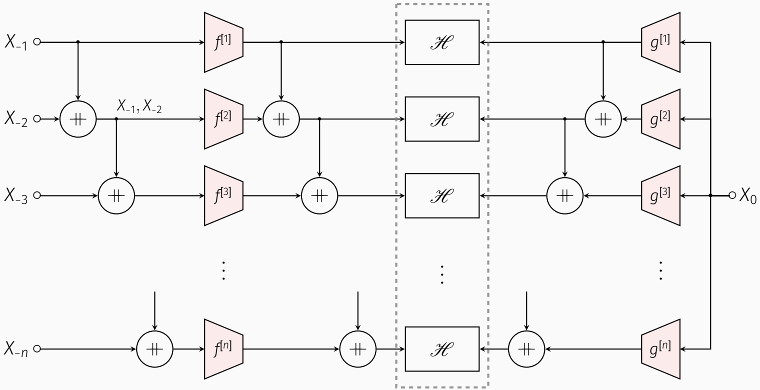
- previous state: $\mathfrak{i}_{X_0;X_{-1}}$
- past 2 states: $\mathfrak{i}_{X_0;(X_{-1},X_{-2})}$

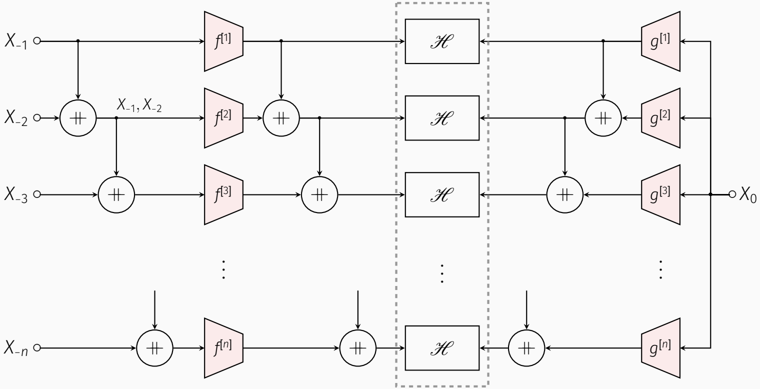
  ⋮

- past n states: $\mathfrak{i}_{X_0;(X_{-1},\cdots,X_{-n})}$

- Gain from ℓ-th layer $\quad \mathfrak{i}_\ell \triangleq \mathfrak{i}_{X_0;X_{-\ell}|(X_{-1},\cdots,X_{-l+1})}$
  - conditional dependence at lag ℓ
- Orthogonal decomposition
  - Dependence between X_0 and past n states $= \sum_{\ell=1}^{n} \mathfrak{i}_\ell$

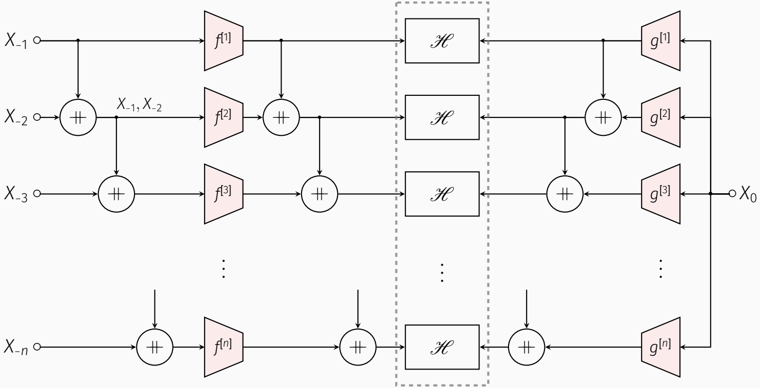
Learning the Decomposition



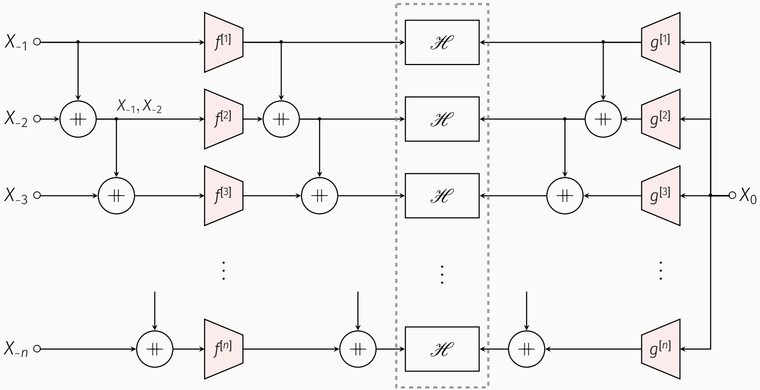


- ℓ-th branch learns $\mathfrak{i}_\ell$: conditional dependence at lag ℓ

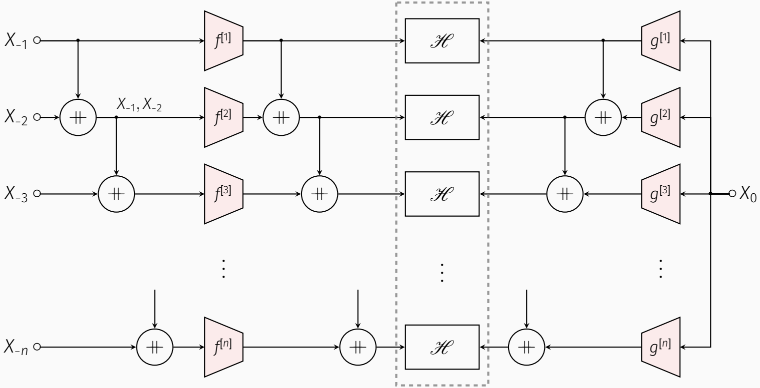
Learning the Decomposition



- ℓ-th branch learns $\mathfrak{i}_\ell$: conditional dependence at lag ℓ
  - top ℓ branches: dependence between X_0 and past ℓ states



- ℓ-th branch learns $\mathbf{i}_\ell$: conditional dependence at lag ℓ
  - top ℓ branches: dependence between X_0 and past ℓ states
- dependence "spectrum" over lags: $\{\|\mathbf{i}_\ell\|^2, \ell \geq 1\}$



- ℓ-th branch learns $\mathfrak{i}_\ell$: conditional dependence at lag ℓ
  - top ℓ branches: dependence between X_0 and past ℓ states
- dependence "spectrum" over lags: $\{\|\mathfrak{i}_\ell\|^2, \ell \geq 1\}$
  - Markov Chain of Order $M \implies$ cutoff at $\ell = M$

Sequence Observations

- Dependence on the History?

Demo

Sequence Observations

- Dependence on the History?
- First/Second/Third-Order?

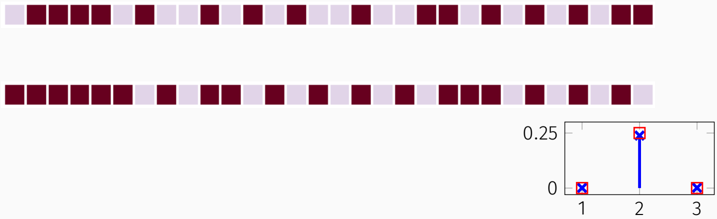
Demo

Sequence Observations

- Dependence on the History?
- First/Second/Third-Order?
- ⋆ Plot Dependence Spectrum $\|\mathbf{i}_\ell\|^2$, $\ell \geq 1$

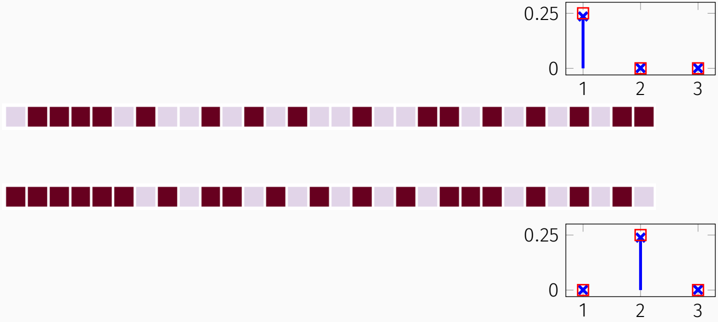
Sequence Observations

- ▶ Dependence on the History?
- ▶ First/Second/Third-Order?
- ⋆ Plot Dependence Spectrum $\|\mathbf{i}_\ell\|^2$, $\ell \geq 1$



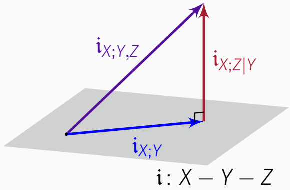
Sequence Observations

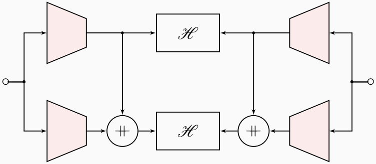
- ▶ Dependence on the History?
- ▶ First/Second/Third-Order?
- ⋆ Plot Dependence Spectrum $\|\mathbf{i}_\ell\|^2$, $\ell \geq 1$



Summary

Summary





- Feature Geometry
  - Feature Learning $\leftrightarrow$ Geometric Operations
  - Nesting Technique
- Case Study: Learning Random Processes
  - Decompose Sequential Dependence

Learn More

- arXiv: 2309.10140