

# 言語の意味の数学的構造



# はじめに

今回のセミナーの目的は、アメリカ数学会のジャーナル Notice 誌2024年の2月号に掲載された、Tai-Danae Bradleyらの論文“The Structure of Meaning in Language”の概要を紹介することです。

この論文は、現在、主流の生成AIの研究者の言語に対するアプローチを、言語理論と数学的手法の両方の側面から批判したもので、注目すべき論文だと考えています。

論文のIntroductionは、次のように語られています。

# The Structure of Meaning in Language

*Tai-Danae Bradley, Juan Luis Gastaldi, and John Terilla*

<https://www.ams.org/journals/notices/202402/rnoti-p174.pdf>

# Introduction

カテゴリー論が機械学習の分野で支持を集めている一つの理由は、それが学習システムをどのように組み立てることができるかを議論するための形式的な方法を提供することである。

本論文は、それとは異なる、やや狭い範囲にフォーカスを当てている。本論文がフォーカスしているのは、カテゴリー論的思考の助けを借りて、言語のモデリングに使われる基本的なAI技術の個々のピースが、**純粋に構文的な入力から言語の構造的特徴を抽出するプロセスとして、どのように理解できるか**ということについてである。

構造は形式から生じるという考え方は、多くの読者にとって驚きではないかもしれない。

カテゴリー論的な考え方は、三世代にわたって純粋数学に大きな影響を与えてきた。しかし、それだけではない。人工知能に関する現在進行中の議論のいくつかにも関連して、カテゴリー論的な考え方は言語学にとってもいくつかの帰結をもたらす。

我々はこの論文に、次のように主張するセクションを含めた。

この論文の数学は、現代の言語学思想において広く受け入れられているいくつかの考え方を反駁するものであり、言語に対する構造主義的アプローチへの回帰を支持するものであると。

本論文は、線形代数のかなりペダンティックな振り返りからはじまる。それが、線形代数と関連するカテゴリー理論との驚くべきな並行性を設定することになる。

その後、線形代数を使って、現在の大規模言語モデル(LLM)の根底にある語の埋め込みを理解する方法を振り返る。

線形代数をMad Libsスタイルで関連するカテゴリー論に置き換えると、出力は語の埋め込みではなく、Formal Conceptのlatticeになる。

latticeの概念を生み出すカテゴリー論は、特に単純化されたenrich化されたカテゴリー論の一部であり、もう少し簡略化することで、より多くの言語構造が明らかになることを示唆している。

# 「ニューラル言語モデル」批判について

この論文の重要な論点である、「ニューラル言語モデル」批判について、あまり数学的議論の細部に入らずに、彼女の批判を、あらかじめその概略を紹介しようと思います。

というのも、彼女の批判は、言語学的で哲学的なものです。

彼女は、次のように語ります。その議論の意図は数学抜きでも多くの人に理解してもらえるものだと思います。

「意味と形式は切り離せないという考え方は新しいものではないが、現在のAIをめぐる哲学的な議論には浸透していない。」

「構文対象の分析において意味が問題になるとすれば、それはすべて言語の形式に反映された構造的特徴に起因するということである。」

「しかし、現在のニューラル言語モデルが不十分なのは、まさにこの点である。というのも、ニューラル言語モデルは、そのタスクを実行する際に必然的に働く構造的特徴を明らかにしていないからである。」

彼女は、AIによる言語処理技術が大きな飛躍をみせ、社会的にも巨大なインパクトを与えようとしている今、ソーシャルやチャットスキーマが切り開いてきた言語学の基本に立ち返って、言語を捉え返そうとしています。

もちろん、それは大規模言語モデルの成果を否定するものではなく、そこから出発して、新しい発展を目指したものです。

# セミナーの構成

前回、前々回のセミナーのテーマは、  
「大規模言語モデルの数学的構造」  
でしたが、今回のセミナーのテーマは、  
「言語の意味の数学的構造」  
であることに、注意してください。

セミナーは、基本的に論文の叙述に沿って展開します。  
次のような構成を考えています。

# 言語の意味の数学的構造

## Agenda

### Part 1

オブジェクト vs. オブジェクト上の関数

### Part 2

自然言語処理での語の埋め込み

### Part 3

意味の空間から意味の構造へ

### Part 4

ニューラル言語モデル批判

# Part 1

## オブジェクト vs. オブジェクト上の関数

言語の意味の数学的構造

## Part 1 の基本的なアイデア

あまり情報が与えられない、あるいは不完全な構造を持つ数学的対象 $X$ を考えてみよう。

この時、 $X$ を「 $X$ 上の関数」のようなものに置き換えることができる。そこには、通常、自然な埋め込み  $X \rightarrow Fun(X)$  がある。

だから、 $Fun(X)$ を扱うことで、 $X$ のすべてとそれ以上のものを扱うことになる。

「オブジェクトではなく、オブジェクト上で定義された関数に注目しよう。」

このアプローチに沿ったものとして、次の三つが紹介されます。

- Vector Space
- presheaf と copresheaf
- enriched category

こうしたアプローチへの志向は、論文全体を一貫して流れています。

# Part 1 Agenda

## オブジェクト vs. オブジェクト上の関数

- Vector Space
- presheaf と copresheaf
- enriched category
- closed commutative monoidal category



# オブジェクト対オブジェクト上の関数 ベクトル空間

マルレク 言語の意味の数学的構造

あまり情報が与えられない、あるいは不完全な構造を持つ数学的対象 $X$ を考えてみよう。

この時、 $X$ を「 $X$ 上の関数」のようなものに置き換えることができる。そこには、通常、自然な埋め込み  $X \rightarrow Fun(X)$  がある。

だから、 $Fun(X)$ を扱うことで、 $X$ のすべてとそれ以上のものを扱うことになる。

# ベクトル空間 $k^X$

最初に思いつく例は  $k^X$  で、集合  $X$  上の関数で体  $k$  上に値を持つ関数の集合である。それは、ベクトル空間を形成する。

ベクトル空間での埋め込み  $X \rightarrow k^X$  は、 $x \in X$  を  $y \mapsto \delta_{xy}$  で定義される  $x$  の指示関数に送るものとして定義される。

$X$  が有限の時には、任意の  $f, g \in k^X$  に対して、内積  $\langle f | g \rangle$  が自然に定義できる。

$$\langle f | g \rangle = \sum_{x \in X} f(x)g(x)$$

内積は、ベクトル空間  $k^X$  をヒルベルト空間にする。

# ket 記法

物理学での、 $x \in X$ に対する指示関数を  $|x \rangle$  で表す“ket”記法は、 $x \in X$ から  $|x \rangle \in k^X$  をうまく区別してくれる。また、そこには内積があることを思い出させてくれる。

$k^X$ での $X$ の像(*image*)は、正規直交する、空間を張る集合を定義する。もし、 $X$ の要素が順序づけられていれば、ベクトル $\{|x \rangle: x \in X\}$ は、 $k^X$ の順序づけられた基底となる。

また、それぞれの基底ベクトル $|x \rangle$ は、one-hot 座標ベクトルを持つ。それは、 $x$ 番目の要素だけが1で、残りの要素全てが0の列ベクトルである。

# 量子情報理論

ところで、これが量子情報理論の出発点である。

古典bit  $\{0,1\}$  は、量子bit  $\{|0\rangle, |1\rangle\}$  で置き換えられる。

そしてそれが、二次元の複素ヒルベルト空間  $\mathbb{C}^{\{0,1\}}$  の正規直交基底を構成する。

集合  $X$  の中では、例えば、二つの要素の和や二つの要素の平均を求める方法が見つからないことがあるかもしれないが、これらの操作は、 $k^X$  の中では、確実に実行されることができる。

例えば、座標系では  $x \neq y$  なら、和  $|x\rangle + |y\rangle$  は、 $x$  番目と  $y$  番目の要素が1で残りは全て0となり、和  $|x\rangle + |x\rangle$  は、 $x$  番目の要素が2で残りは全て0となる。

# ベクトル空間 $\{0,1\}^X$

基本となる体 $k$ が、二つの要素を持つ $k = \{0,1\}$ の場合、ベクトル空間の構造は、少し弱くなるように見える。

ただ、スカラー倍は自明だが、 $\{0,1\}^X$ 上には、注目すべき構造がある。 $\{0,1\}^X$ の要素は、 $X$ の部分集合と考えることができる。

この対応は、 $X$ の特徴関数(characteristic function)とその台集合の対応である。

# Bool 代数

$\{0,1\}^X$  は、Bool代数の構造を持つ:

- 二つのベクトルの join  $v \vee w$  と meet  $v \wedge w$  は、 $v$ と $w$ で定義される二つの部分集合の unionとintersection に対応する。ここでは、meetもjoinもベクトルの和とは一致しない。
- すべてのベクトルは、「補完(complement)」をもつ。それは0と1を交換したもの  $0 \leftrightarrow 1$  として定義される。
- $\{0,1\}^X$  のベクトルは、包含関係で、半順序となる。
- $\{0,1\}^X$  のベクトルは、非ゼロの要素の数で定義される濃度を持つ。
- ...

## 参考資料

ベクトル空間については、2023年2月のマルレク「密度行列  $\rho$  で理解する確率の世界」を参照されたい。

<https://www.marulabo.net/docs/density2/>

このセミナーは、Tai-Danae Bradleyの2020年の論文 “At the Interface of Algebra and Statistics” 「代数と統計の境界で」

<https://arxiv.org/abs/2004.05631> に依拠したものである。

以下、そのスライドの一部を再掲する。

ケット記法については、2020年6月のマルレク「ケット  $|k\rangle$  で理解する量子の世界」も参照されたい。

<https://www.marulabo.net/docs/ket-talk/>

# $S$ 上のベクトル空間 $V$

$S$ を有限集合 $\{s_0, s_1, \dots, s_{n-1}\}$ とする時、 $S$ の要素上で定義され複素数 $\mathbb{C}$ に値を持つ関数 $\mathbb{C}^S: S \rightarrow \mathbb{C}$ の集まり $V$ を、 $S$ 上のベクトル空間という。ここでは有限次元の場合のみを扱う。

$$V = \mathbb{C}^S$$

内積  $\langle v|w \rangle$  が定義されたベクトル空間をヒルベルト空間という。

$$\langle v|w \rangle = \sum_{s \in S} \overline{v(s)} w(s)$$

# $|s\rangle$ は、 $S$ から $\mathbb{C}$ への関数 計算基底

$s \in S$ の時  $|s\rangle$  で $S$ から $\mathbb{C}$ への関数であるベクトルを表す。  
これを、ket 記法という。

$$S = \{s_0, s_1, \dots, s_{n-1}\} \text{ で } |s\rangle \in \mathbb{C}^S \text{ である}$$

$|s\rangle$  は、次の値を取ることができる。

$$s \mapsto 1, \quad s' \neq s \text{ に対して } s' \mapsto 0$$

これを計算基底と呼ぶ。

$$|s_i\rangle = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i \text{ 番目のエントリー}$$

# 基底 $|s\rangle$ が与えられた時の 任意のベクトル $|v\rangle \in V$ の表示

基底  $|s\rangle$  が与えられた時、任意のベクトル  $|v\rangle \in V$  は、次の形で表現できる。

$$|v\rangle = \sum_{s \in S} v(s) |s\rangle = \begin{bmatrix} v(s_1) \\ \vdots \\ v(s_n) \end{bmatrix} \quad v(s) \in \mathbb{C}$$

# $bra \langle v|$ の定義

全ての  $|v\rangle \in V$  について、 $V^* := \text{hom}(V, \mathbb{C})$  で、 $|v'\rangle \in V$  上の値が  $\langle v|v'\rangle$  である線形関数  $\langle v|$  が存在する。

$$\begin{array}{ccc} V & \xrightarrow{\langle v|} & \mathbb{C} \\ |v'\rangle & \longmapsto & \langle v|v'\rangle \end{array}$$

この線形関数  $\langle v| \in V^*$  を bra と呼ぶ。

$$V \cong V^*$$

# BraとKetの二対二対応

$$|v\rangle^\dagger = \langle v| \quad \langle v|^\dagger = |v\rangle$$

で次のような対応が存在する。

$$|v\rangle \longleftrightarrow \langle v|$$

$$V \cong V^*$$

$$\langle v| = \sum_{s \in S} \overline{v(s)} \langle s| = [\overline{v(s_1)} \quad \cdots \quad \overline{v(s_n)}] \quad v(s) \in \mathbb{C}$$

# テンソル積

$|v\rangle \in V, |w\rangle \in W$ の時、

テンソル積  $V \otimes W$  の要素を

$|v\rangle \otimes |w\rangle$  あるいは  $|vw\rangle$  と表す。

テンソル積  $V \otimes W^*$  の要素を、

$|v\rangle \langle w|$  あるいは  $|v\rangle \otimes \langle w|$  と表す。

$V \otimes W^*$  は、次のような  $V \rightarrow W$  の写像となる。

$$\begin{array}{ccc} V & \xrightarrow{|w\rangle \langle v|} & W \\ |v'\rangle & \longmapsto & |w\rangle \langle v|v'\rangle \end{array}$$



オブジェクト対オブジェクト上の関数  
(co)presheaf

マルレク 言語の意味の数学的構造

# 「オブジェクトからオブジェクト上の関数へ」の カテゴリー論バージョンを考える

「オブジェクトからオブジェクト上の関数へ」のカテゴリー論のバージョンを考えてみよう。

カテゴリー $C$ を、集合のカテゴリー $Set$ に値を取る $C$ 上の *functor*  $Set^{C^{op}}$  で置き換える。そうすると、重要なことにたくさんの構造を持つあるカテゴリーが得られる。

ここで $C$ の肩に乗っている“op”は、問題の *functor* の *variance* を表している。この例の場合には、この *functor* が、**contravariant functor** であることを表している。

*functor* の *variance* は、この論文では非常に重要なわけではないのだが、技術的ポイントとして正確さのために含めている。

# presheaf

$Set^{C^{op}}$  の functor  $F$  は、一般に  $C$  上の *presheaf* と呼ばれる。

**Yoneda Lemma** が、もとのカテゴリー  $C$  を、 $Set^{C^{op}}$  にその full subcategory として埋め込む  $C \rightarrow Set^{C^{op}}$  を提供する。

$C$  のオブジェクト  $x$  が与えられた時、Grothendieck は表現可能な presheaf  $h^x$  を指すものとして  $h^x := C(-, x)$  という表記をつかった。これは、presheaf を表現する  $y$  からオブジェクト  $x$  への射の集合  $C(y, x)$  によって定義されている。

この表記では、**Yoneda Embedding** は、オブジェクト  $x$  上  $x \mapsto h^x$  で定義される。

# topos

ベクトル空間の  $k^X$  が  $X$  以上の構造を持つのとちょうど同じように、  
カテゴリー  $Set^{C^{op}}$  は、 $C$  以上の構造を持つのである。

すべてのカテゴリー  $C$  に対して、presheafのカテゴリー  $Set^{C^{op}}$   
は、completeかつcocomplete である。

それは、カテゴリー  $Set^{C^{op}}$  には、カテゴリー論的 limitとcolimit  
という小さな図式が存在することを意味している。もっとある。

それらは、topos と言われるものの例でもある。

toposは、幾何学と論理学を行う自然な場所である。

# 離散カテゴリーのpresheaf

いくつか例をあげよう。

有限集合 $X$ を考える。それは、射が同一射のみからなる離散カテゴリーと考えることができる。

この場合、 $X = X^{op}$ であり、 $X$ 上のpresheaf  $F$ は、 $X$ のすべてのオブジェクト $x$ にある集合を割り当てる。

もし、 $X$ の要素が順序をもてば、 $F$ はその要素が集合で、 $x$ 番目の要素が $F(x)$ である列ベクトルと考えることができる。

表現可能なfunctor  $h^x$ は、 $x$ 番目の要素だけが、一点からなる集合 $*$ で、それ以外の要素はすべて空集合である列ベクトルと考えることができる。

空集合を0に、一点からなる集合 $*$ を1とすると、これは、ベクトル空間 $k^X$ を張るone-hot 基底ベクトルと同じ並びになる。

## coproduct $h^x \sqcup h^y$

注目すべきことは、カテゴリー $X$ には coproduct  $x \sqcup y$ は存在しないのだが、 $X$ 上のpresheaf カテゴリーでは、表現可能な functor  $h^x$ と $h^y$ について、coproduct  $h^x \sqcup h^y$  は存在する。

$x \neq y$ なら、 $h^x \sqcup h^y$  は、 $x$ 番目と $y$ 番目の要素が、一点からなる集合  $*$  で、それ以外はすべて空集合であるカラムである。

$x = y$ なら、 $h^x \sqcup h^x$  は、 $x$ 番目の要素が、二点集合  $* \sqcup *$  で、それ以外はすべて空集合であるカラムである。

## colimit of representables

ベクトル空間  $k^X$  では、指示関数(indicator function)がその基底を形成したように、すべてのfunctor  $X^{op} \rightarrow Set$  は、表現可能なfunctorによって構成される。

$X$ が有限集合の時、ベクトル空間  $k^X$  のすべてのベクトルは、基底ベクトルの線型結合で表現されるように、 $Set^{X^{op}}$  のすべてのpresheafは、representable のcolimit である。

## 参考資料

(co)presheafについては、

2023年12月のマルレク「大規模言語モデルの数学的構造 I」

<https://www.marulabo.net/docs/llm-math/>

2024年1月のマルレク「大規模言語モデルの数学的構造 II」

<https://www.marulabo.net/docs/llm-math2/>

を参照されたい。

ただし、これらの資料では、presheaf ではなく copresheaf について説明がなされている。今回の論文は、presheafについて論じている。

# presheafとcopresheaf

presheafとcopresheafは、dualな概念である。

いずれもあるカテゴリー $C$ から集合のカテゴリー $Set$ へのfunctorである。

$C$ 上のpresheafは、 $C^{op} \rightarrow Set$ のcontravariant functorで、  
 $C$ 上のcopresheafは、 $C \rightarrow Set$ のcovariant functorである。

二つのvarianceの違いから、そのrepresentable functorは、次のようになる。

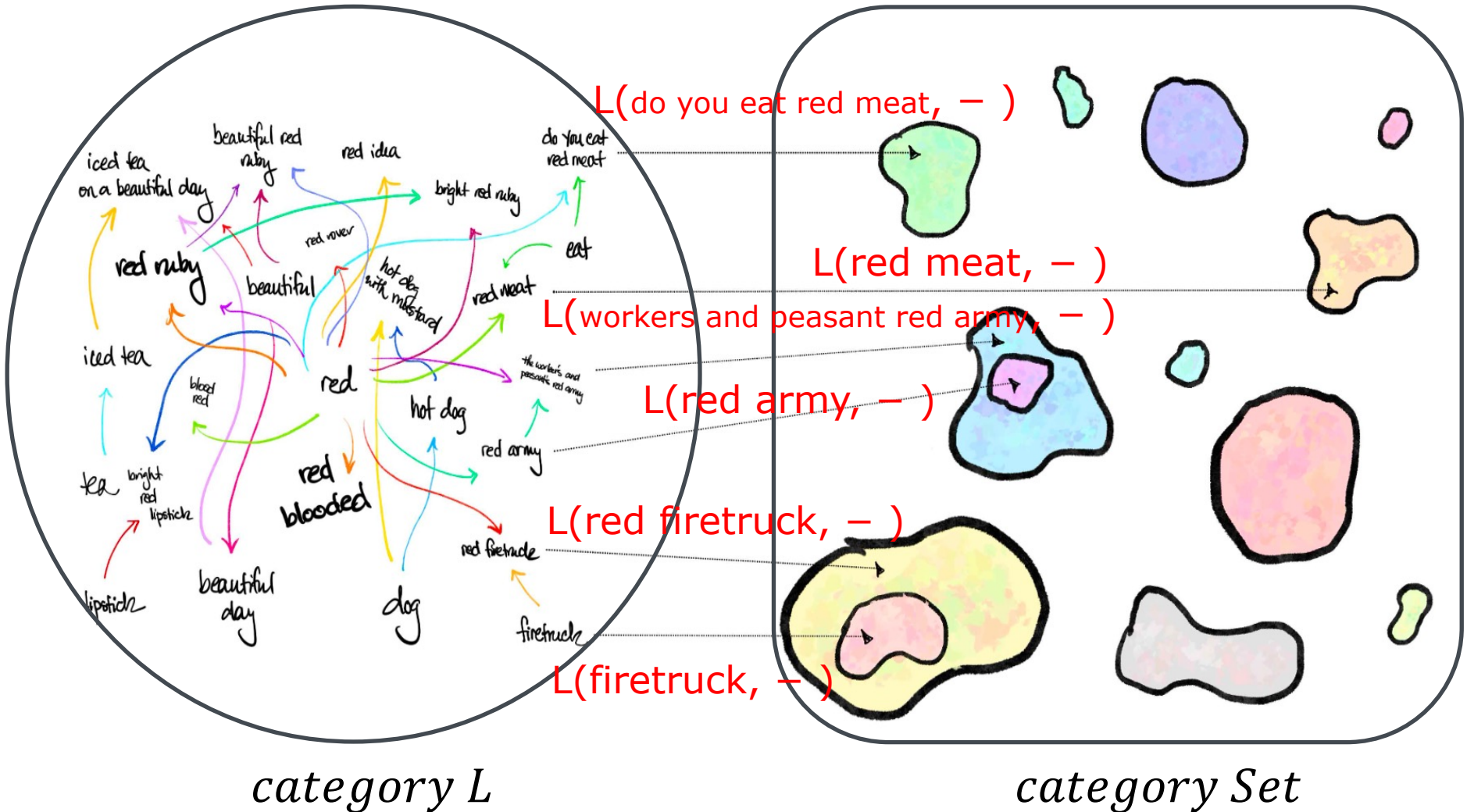
presheafのrepresentableは、 $h^x := C(-, x)$ で、  
copresheafのrepresentableは、 $h_x := C(x, -)$ である。

# $Set^L$ のオブジェクト $L(x, -)$ の例

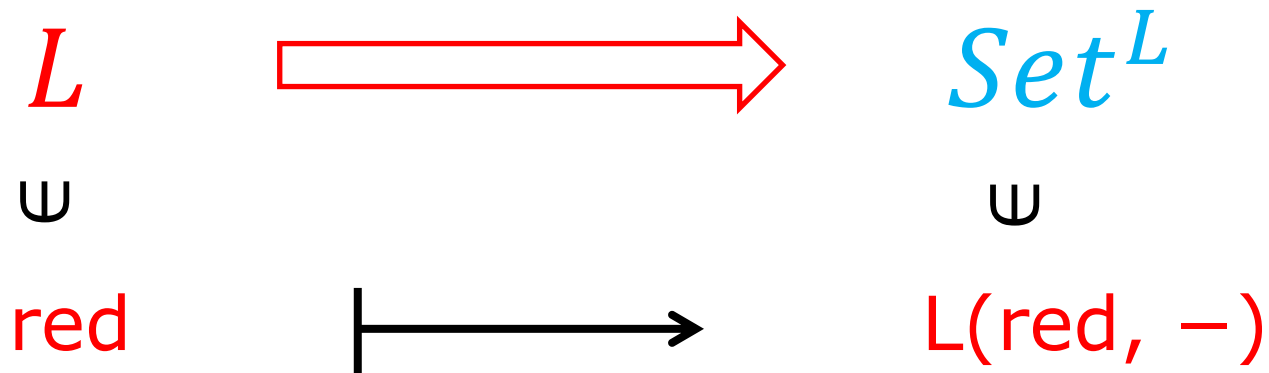
$L$



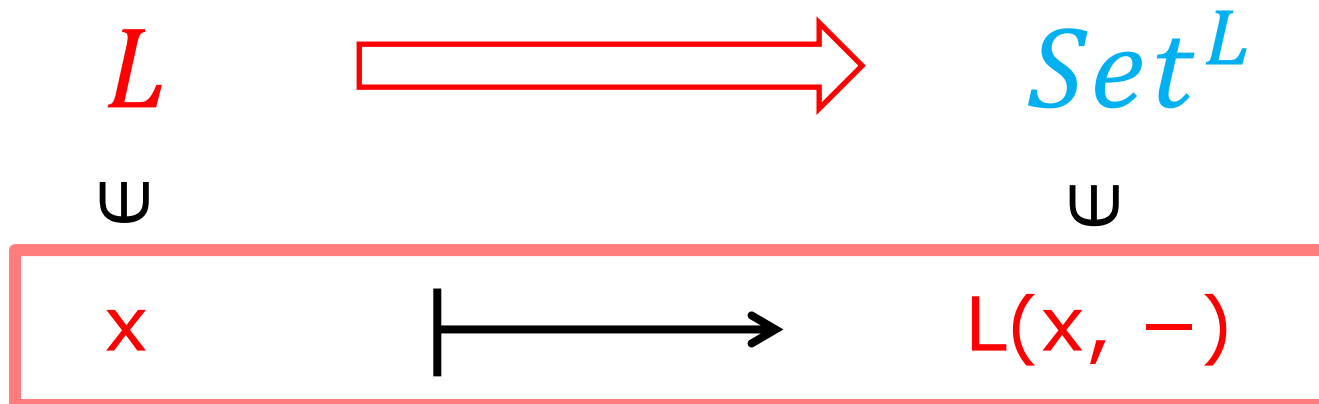
$Set$



category  $L$  とcategory  $Set^L$  の対応は、  
 $red$  と  $L(red, -)$  を対応づけるものです。



より一般的な、次の値の割り当てを、**Yoneda embedding** と言います。

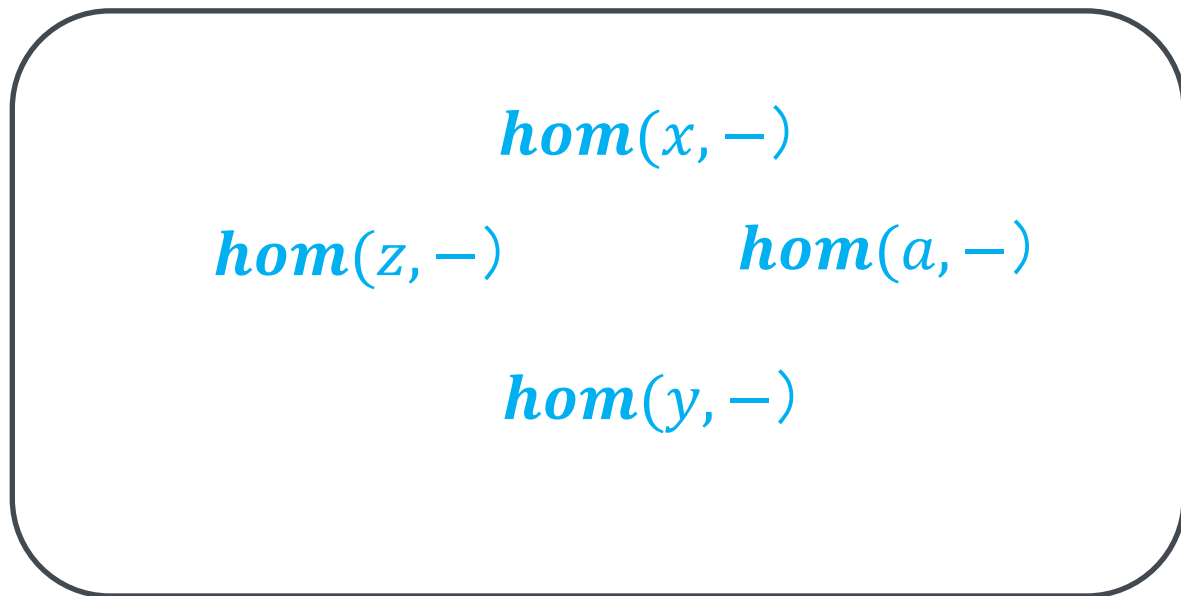


集合に値を持つ表現可能なfunctorである  
意味のcategory  $Set^L$ を、  
L上のcopresheafといいます。

意味のcategory  $Set^L$

$Set^L$

copresheaf



すべてのLのオブジェクトxについて  
 $hom(x, -)$ なるfunctorの集まり

“red or blue” の意味を表す

$$L(\text{red}, -) \sqcup L(\text{blue}, -)$$

$$L(\text{red}, -) \sqcup L(\text{blue}, -) \stackrel{\text{SORT OF}}{=} \left[ \begin{array}{l} * \\ * \\ * \\ \emptyset \\ ** \\ \emptyset \\ \vdots \end{array} \right] \begin{array}{l} \text{deep red Bing cherries} \\ \text{small blue marble} \\ \text{beautiful blue ocean} \\ \text{did you put the kettle on} \\ \text{red and blue fireworks} \\ \text{Sencha green tea} \\ \vdots \end{array}$$

# copresheafの表現可能性

## 定義 7

$[0,1]$ -category のすべてのオブジェクト  $x$  について、functor  $h^x := C(x, -)$  を  $x$  で表現された copresheaf と呼ぶ。

もし、ある  $x \in C$  について  $f = h^x$  なる  $f$  が、存在するならば copresheaf  $f: C \rightarrow [0,1]$  は表現可能であるという。



オブジェクト対オブジェクト上の関数

enriched category

マルレク 言語の意味の数学的構造

# enriched category theory

この論文では、2つのオブジェクト間の射の集合  $C(y, x)$  がもはや単なる集合でない場合の、カテゴリー論である enriched category theory について考察する。

そこでは、射の集合  $C(y, x)$  自身が、部分順序集合であったり、アーベル群であったり、位相空間であったり、他の何かであったりする。つまり、enrich 化されたカテゴリーでは、先の presheaf の構成  $Set^{C^{op}}$  で、 $Set$  を別のカテゴリーに置き換えることになる。

これはベクトル空間  $k^X$  のベースの体  $k$  を変えることに似ており、新しいベースのカテゴリーが十分に良い構造を持っていれば、 $C$  を  $Set^{C^{op}}$  で置き換えることについて述べたことは、ほとんど一語一語すべて通用する」。

## 2-enriched presheaf

例えば、 $\mathbf{Set}$  を、0と1という二つのオブジェクトを持ち、同一射以外の射は、 $0 \rightarrow 1$  一つだけである**2** というカテゴリーに置き換えると、それは、**2**-valued presheaves の  $2^{C^{op}}$  というカテゴリーになる。

この場合、 $C$  が  $X = X^{op}$  なる離散カテゴリー  $X$  だとすると、 $2^X$  の presheaf は、正確に  $X$  上の  $\{0, 1\}$  に値を取る関数に等しくなり、それは、 $X$  の部分集合に等しい。

$2^X$  の、それが**2**-enriched presheafであることによって与えられる構造は、以前に述べた部分集合上の Bool 代数である。

2-enriched presheaf  $f$  と  $g$  の、カテゴリー論的 coproduct は、join(union)  $f \vee g$  であり、そのカテゴリー論的 product は meet(intersection)  $f \wedge g$  である。

だから、どんな集合  $X$  に対しても、関数の集合  $\{0,1\}^X$  は、二つの要素からなる体  $F_2 = \{0,1\}$  上のベクトル空間と見ることもできるし、 $2 = \{0,1\}$  に値を取る、 $X$  上の enrich 化された presheaf とも見ることができる。

我々が、 $\{0,1\}$  を体と考えるか、あるいは、それを射  $0 \rightarrow 1$  が定義されたカテゴリーだと考えるのか、どちらの見方を取るかに応じて、異なる構造を得るのである。

オブジェクト対オブジェクト上の関数

closed commutative  
monoidal category

さて、さらに議論を進める前に、オブジェクトのfree constructionでオブジェクトを置き換えることでは、もとのオブジェクトについて明らかになることは、直接的には多くはないことを確認しておこう。

$k^X$ に帰着する有限集合  $X$  上の自由なベクトル空間であれ、カテゴリー  $C$  の自由な cocompletion であれ、得られる構造は自由だが、基礎であるオブジェクトを単にindex set と利用しているに過ぎない。

$X$ 上の“関数”の構造は、本質的に、その関数が何に値を持つかという構造に負っている。

例えば、 $Set^{C^{op}}$ のcomplete性やcocomplete性は、集合のカテゴリのcomplete性やcocomplete性に源を持っている。

同様に、 $k^X$ でのベクトルの加法やスカラー倍は、体 $k$ の加法や乗法から生まれたものである。

ポイントは、 $X$ のfree constructionに移行することで、 $X$ を調べるためのいくつかの余地が生まれることである。例えばベクトル空間の理論では、線形変換が絡むと物事が面白くなる。

別の例をあげれば、有限群 $G$ から自由ベクトル空間 $\mathbb{C}^G$ の移行は、群の要素を演算子 $\mathbb{C}^G \rightarrow \mathbb{C}^G$ として考えるまでは、その群について多くを語るわけではない。

その結果、 $G$ の正則表現が得られる。この正則表現は多くの美しい性質を持ち、すべての既約表現が意味のある多重度を持つ項として含まれる既約表現の直和に分解される。

ここで、このセクションの最初の行で提案された戦略に戻る。

オブジェクト $X$ の内部構造について少ししか知られていない場合、より多くを学ぶためのアプローチは、 $X$ を $X$ 上の関数のようなもので置き換え、 $X$ の限られた既知の構造が、 $X$ の関数上で自由に定義された構造とどのように相互作用するかを研究することである。

特に、何の中で関数を評価し、また、その対象を数学的にどう見るかという選択が必要である。

この論文の残りのセクションは、言語学的に重要な例で、関数を何で評価するかという自然な選択肢のいくつかについて、詳細を扱うものと解釈できる。

## 参考資料

enriched category については、

2024年1月のマルレク「大規模言語モデルの数学的構造 II」

<https://www.marulabo.net/docs/llm-math2/>

を参照されたい。

# *V-enriched category*

## 定義 2

$(V, \leq, \otimes, 1)$  を、commutative monoidal preorder とする。  
この時、 $V$  で enrich 化されたカテゴリー（単純に、 $V$ -カテゴリーと呼んでもいい）を、次の様に定義する。

- $V$ -カテゴリーは、 $C$  と同じオブジェクトを持つ。
- また、 $V$ -カテゴリーは、その任意のオブジェクトのペア  $x, y$  について、 $V$ -hom オブジェクトと言われる  $C(x, y) \in V$  を持つ。
- この  $V$ -hom オブジェクトは、すべての  $x, y, z \in V$  について、次の条件をみたす。

$$\begin{aligned} 1 &\leq C(x, x) \\ C(y, z) \otimes C(x, y) &\leq C(x, z) \end{aligned}$$

# 単位区間 $[0,1]$ は commutative monoidal preorder である

monoidal product  $\otimes$  を普通の実数間の掛け算とし、  
monoidal unit を 1 とし、preorderの順序関係 $\leq$ を通常の実  
数間の大小関係とすればいい。

この単位区間  $[0,1]$  で enrich化されたカテゴリー、 **$[0,1]$ -カテ  
ゴリー**は、 $C$ と同じオブジェクトを持ち、すべての $x, y \in C$ で定義さ  
れた $[0,1]$ に値を取る関数 $(x, y) \mapsto C(x, y)$ を持つ。

この $C(x, y)$ は、次の条件を満たす。

すべての $x \in C$  について、 $C(x, x) = 1$

すべての $x, y, z \in C$  について、 $C(y, z)C(x, y) \leq C(x, z)$

# closedなcommutative monoidal preorder

## 定義 3

commutative monoidal preorder  $V$ は、すべての  $x, y \in V$  に対して、次の条件を満たす  $[x, y] \in V$  が存在する時、closed であると言う。

$$x \otimes y \leq z \text{ if and only if } x \leq [y, z]$$

この  $[x, y] \in V$  を、 $V$  の **internal hom** という。

$(x, y) \in V$  なら  $[x, y] \in V$ 、すなわち、このinternal homについて  $V$  が閉じている時、 $V$  はclosed commutative monoidal preorder と言われるということである。

# enriched copresheaf の定義

$V$ がclosed commutative monoidal preorder の時、 $V$ は自分自身でenrich化されたカテゴリーになります。

先の、enrich化されたfunctor  $C \rightarrow D$  で  $D = V$ とした enrich化したfunctor  $C \rightarrow V$ を考えます。

## 定義 6

$C$ を、closed commutative monoidal preorder  $V$ 上でenrich化されたカテゴリーとする。

enrich化されたcopresheaf  $V^C$ は、 $C$ のすべてのオブジェクト  $x, y$ について  $C(x, y) \leq V(fx, fy)$ を満たす関数で定義される。





## Part 2

# 自然言語処理での語の埋め込み

言語の意味の数学的構造

# 語のベクトル表現

ここ10年、計算言語学と自然言語処理(NLP)の研究者たちは、当初はアルファベット順に並べられた集合の構造しか持たなかった単語を、ベクトルに置き換えるというステップを踏んできた。

単語には構造があるように感じられる。

- 単語は目的と意味を持って言語の中で使われているように見える
- 辞書は各単語を他の単語と関連付ける
- 単語には品詞のラベルを付けることができる
- などなど

しかし、単語の構造とその使い方の正確な数学的性質は明らかではない。

言語は、前節で説明した構造を明らかにする戦略の重要な実世界でのテストになると思われる。

単語をベクトルで置き換えるというステップは、人工知能分野における現在の進歩の主な原動力のひとつであるが、集合をその集合上の関数で置き換えるという例としては、すぐには認識できない。

というのも、単語をベクトルに置き換えることは、通常、NLPツールによって暗黙のうちに行われるためである。

画像処理や音声処理などの領域で得られた驚くほど良好な結果に続き、2010年代には自然言語の処理に取り組む研究者たちが、ディープニューラルネットワーク(DNN)モデルの応用に関心を寄せるようになった。

## Part 2 Agenda

### 自然言語処理での語の埋め込み

- DNN と word embedding
- embedding algorithm



自然言語処理でのembedding

DNN と word embedding

マルレク 言語の意味の数学的構造

# DNN – Deep Neural Network

DNNは、その最も基本的な形では、明示的に関数の合成で定義される関数  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$  として記述することができる:

$$f: \boxed{\mathbb{R}^n} \xrightarrow{f_1} \mathbb{R}^{n_1} \xrightarrow{f_2} \mathbb{R}^{n_2} \xrightarrow{f_3} \dots$$
$$\dots \xrightarrow{f_K} \mathbb{R}^{n_K} \xrightarrow{g} \boxed{\mathbb{R}^m}$$

$$f_i(\mathbf{x}) = a(\mathbf{M}_i \mathbf{x} + b_i),$$

この関数  $f_i(x) = a(\mathbf{M}_i x + b_i)$  で、 $M_i$  は  $n_i \times n_{i-1}$  の行列、 $b_i \in \mathbb{R}^{n_i}$  は「バイアス」、 $a$  は非線形の「活性化関数」である。  
また、上の図で、 $g$  は出力関数と呼ばれる。

# DNNの学習

DNNの学習は、モジュライ空間を通して、適切な $M_i$ と $b_i$ を探索するプロセスである。

DNNの学習は、通常はモジュライ空間上で定義されるコスト関数を最小化することにより、目的のタスクを実行する望ましい関数を見つけるプロセスである。

このプロセスは、自然言語タスクの場合、通常、言語データをモデルに入力し、最適化の目的を実際出力と意図された出力との誤差の最小化に設定し、最適化は勾配降下法の形式で実行される。

# 最初の選択

## 1-hot ベクトルとしての語の埋め込み

重要なことは、こうした設定では、言語データ(典型的には「語」である)は、ベクトルとして表現されているだろうことである。DNNのドメインはベクトル空間なのである。

実務家にとって自然な最初の選択は、単語を 1-hotベクトルとして表現することであった。したがって、例えば30,000個の単語からなる語彙 $D$ がある場合、 $D \rightarrow \mathbb{R}^D$ は30,000次元の実数ベクトル空間における標準基底ベクトルとして単語を埋め込む。

aardvark  $\mapsto (1, 0, 0, 0, \dots, 0, 0)$

aardwolf  $\mapsto (0, 1, 0, 0, \dots, 0, 0)$

⋮

zyzzyva  $\mapsto (0, 0, 0, 0, \dots, 0, 1).$

## ある驚くべき発見

同様に、出力が単語としてデコードされる場合、出力関数 $g$ はターゲット語彙の確率分布として解釈される。

当時、DNNはエンド・ツー・エンドのプロセスとして考えられていた。入力と出力の間に起こることはすべてブラックボックスとして扱われ、最適化アルゴリズムが処理するように任されていた。しかし、驚くべき事態が発生した。

ある言語タスク用に学習を済ましたモデルの第1層(先の図の $f_1$ )を、別の言語タスクを対象とした別のDNNの第1層として使用すると、この二つ目のタスクの性能が大幅に向上するのである。

## 低次元なベクトルへの埋め込み

その後間もなく、研究者たちはその単一層を、文脈中の他の単語から単語を予測するモデルのユニークな隠れ層として独立に訓練し始めた。その単一層を  $\sigma$  とすると、次のようになる。

$$D \hookrightarrow \mathbb{R}^D \xrightarrow{\sigma} \mathbb{R}^{n_1},$$

$\sigma$  は、 $D$  をもっと低い次元のベクトル空間、典型的には、2~300次元のベクトルに埋め込んだ。例えば、embedding projector <https://projector.tensorflow.org/> のベクトル表現では、単語 aarark が 200 個の成分を持つベクトルにマッピングされている。

$$\text{aardvark} \mapsto (0.632, 0.370, -0.620, \dots, -0.475).$$

# 1-hot ベクトルから密なベクトルへの埋め込み

このようにして、 $\sigma$ のもとで、最初の1-hot 基底ベクトルのイメージは、複数のDNNモデルの入力として供給される低次元の密な語ベクトル表現として使用することができる。

ここでの「密な」という言葉は専門用語ではなく、 $\mathbb{R}^D$ の元の1-hot ワード・ベクトルが、1つの1を除いてすべてのエントリーが0であるという意味で「疎な」ものであるのと対比して使われている。一方、 $\mathbb{R}^{n_1}$ の単語ベクトルは一般に非ゼロのエントリを持つ。

$$\begin{array}{ccc} D \hookrightarrow \mathbb{R}^D & \xrightarrow{\sigma} & \mathbb{R}^{n_1}, \\ \text{aardvark} \mapsto (1, 0, 0, 0, \dots, 0, 0) & & \\ & & \text{aardvark} \mapsto (0.632, 0.370, -0.620, \dots, -0.475). \end{array}$$

## 内積で意味の類似性を表すことができる

このようにして生成された語ベクトル表現の集合(「語埋め込み」とも呼ばれる)は、いくつかの驚くべき能力を持つことがわかった。

異なるタスクにおけるモデルの性能が大幅に向上しただけでなく、埋め込まれた単語ベクトルのベクトル空間演算に予期せぬ言語的意義が見出されたのである。

特に、**2つのベクトル間の内積は意味的類似度と高い相関を示す。**例えば、*tubulidentata*, *orycteropus*, *anteaters*, *shrews*, *pangolins* といった語のベクトル表現と語 *aardvark* のベクトル表現の内積の値は最も大きなものとなる。

# ベクトルの足し算と引き算が意味を表す

さらに驚くべきことに、埋め込み空間におけるベクトルの足し算と引き算は、それらが表す単語間の類似関係と相関している。

例えば、BerlinのベクトルからGermanyのベクトルを引いたものは、ParisのベクトルからFranceのベクトルを引いたものに、数値的には非常に近いものになる。[MSC+13]

これらのことから、単語のベクトルは、個々の単語が点として埋め込まれた意味空間のようなものに存在していることがわかる。

BerlinのベクトルからGermanyのベクトルを引いても、辞書の単語に対応するベクトルにはならない。むしろ、これらのベクトルの差は「capital city」という概念に近いものであり、意味空間の別の場所にある「capital」という単語のベクトルと混同してはならない。

## 純粋な構文特性の形式的操作から 意味特性が抽出できるということ

このような語のベクトル表現は、現在注目されているLLMを含め、現在のニューラル言語モデルの標準的な入力となっている。

そして、これらの知見から示唆されるように、純粋な構文特性の形式的操作から意味特性が抽出できるという事実、すなわち純粋な形式から意味が出現するという事実は、間違いなく現代で最も刺激的なアイデアの一つである。

このような考え方は決して新しいものではなく、少なくとも1世紀以上前から言語思想の中に存在していたものであることは、後ほど説明しよう。

$\mathbb{R}^D$  の1-hotベクトルには、意味が存在しない

しかしその前に、なぜ語の埋め込みが、前節で紹介した $X$ から  $\text{Fun}(X)$ への移行の有用性を示すのかを理解することが重要である。

$\mathbb{R}^D$ に含まれる1-hotベクトルには、語埋め込みの意味的性質は存在しない。

実際、aardvarkとtubulidentata の1-hotベクトルの内積は0であり、 $\mathbb{R}^D$ でのベクトル空間演算は言語的意味を持たない。

GermanyとBerlinの1-hotベクトルの差は、FranceとParis の1-hotベクトルの差から遠く離れている。salamiとtherefore の差も、他のどんな二つの1-hot ベクトルの差も同じである。

言語学的に重要な特性は、ニューラル最適化アルゴリズムによって得られた埋め込みマップ  $\sigma : \mathbb{R}^D \rightarrow \mathbb{R}^{n_1}$  と合成した後に初めて現れる。

このことは、典型的に解釈の難しい問題である。



自然言語処理でのembedding  
embedding algorithm

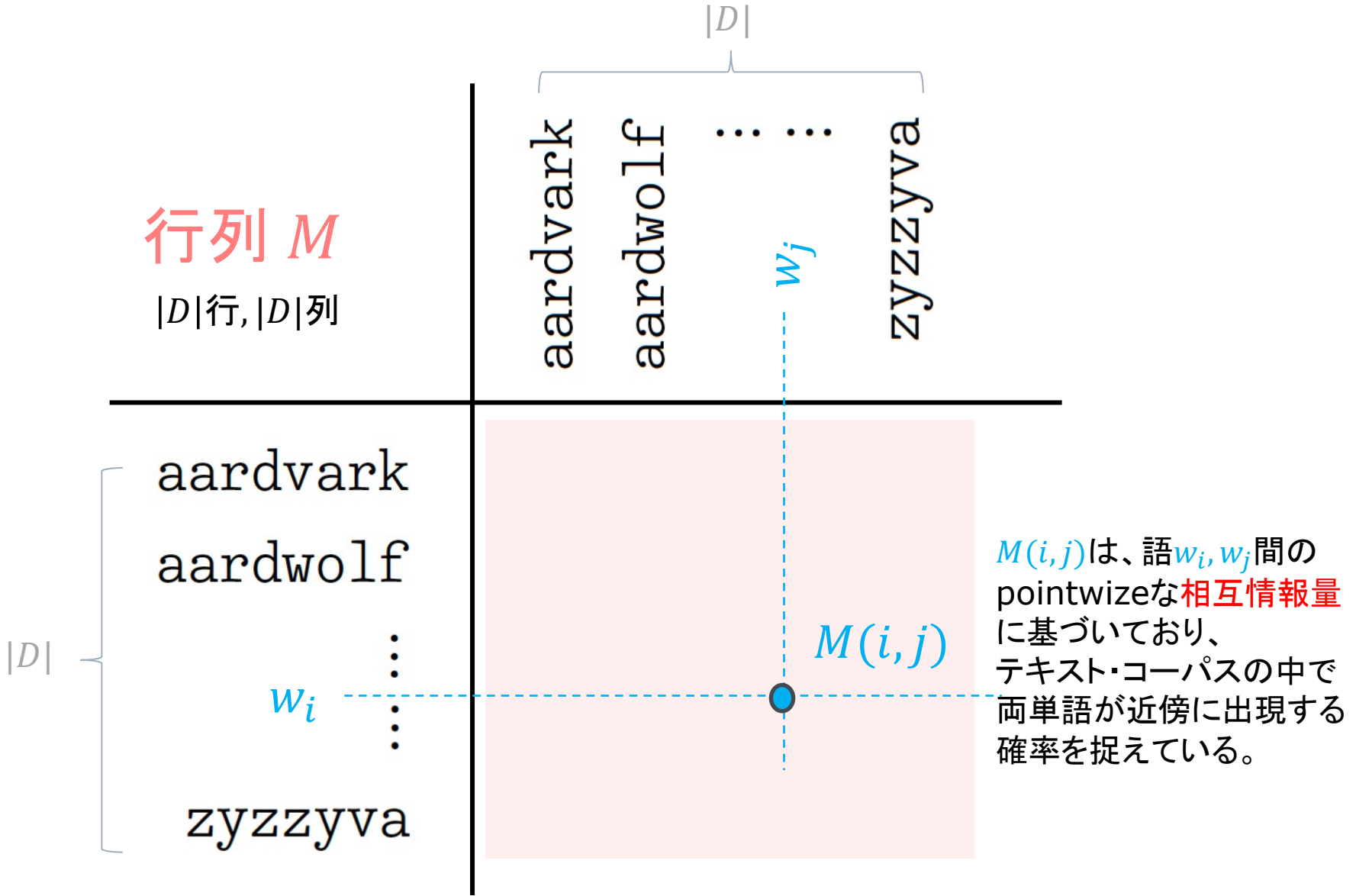
マルレク 言語の意味の数学的構造

# embedding algorithm

しかし、語埋め込みという特殊なケースでは、アルゴリズムが精査され、単語が言語中でどのように使われるかという情報から構成される行列の暗黙の因数分解を実行していることが示されている。

少し詳しく述べれば、語埋め込みの最適化の目的は、 $|D| \times |D|$ のある行列 $M$ の因数分解と同値であることを示すことができる。

この行列はその  $i, j$  番目のエントリーが、語 $w_i$ と語 $w_j$ のあいだの語と文脈の関連性を表わす、言語学的に意味のある尺度となっている。この尺度は、両単語間のpointwiseな相互情報量に基づいており、テキスト・コーパスの中で両単語が近傍に出現する確率を捉えている。



$M(i, j)$ は、語 $w_i, w_j$ 間の pointwiseな相互情報量に基づいており、テキスト・コーパスの中で両単語が近傍に出現する確率を捉えている。

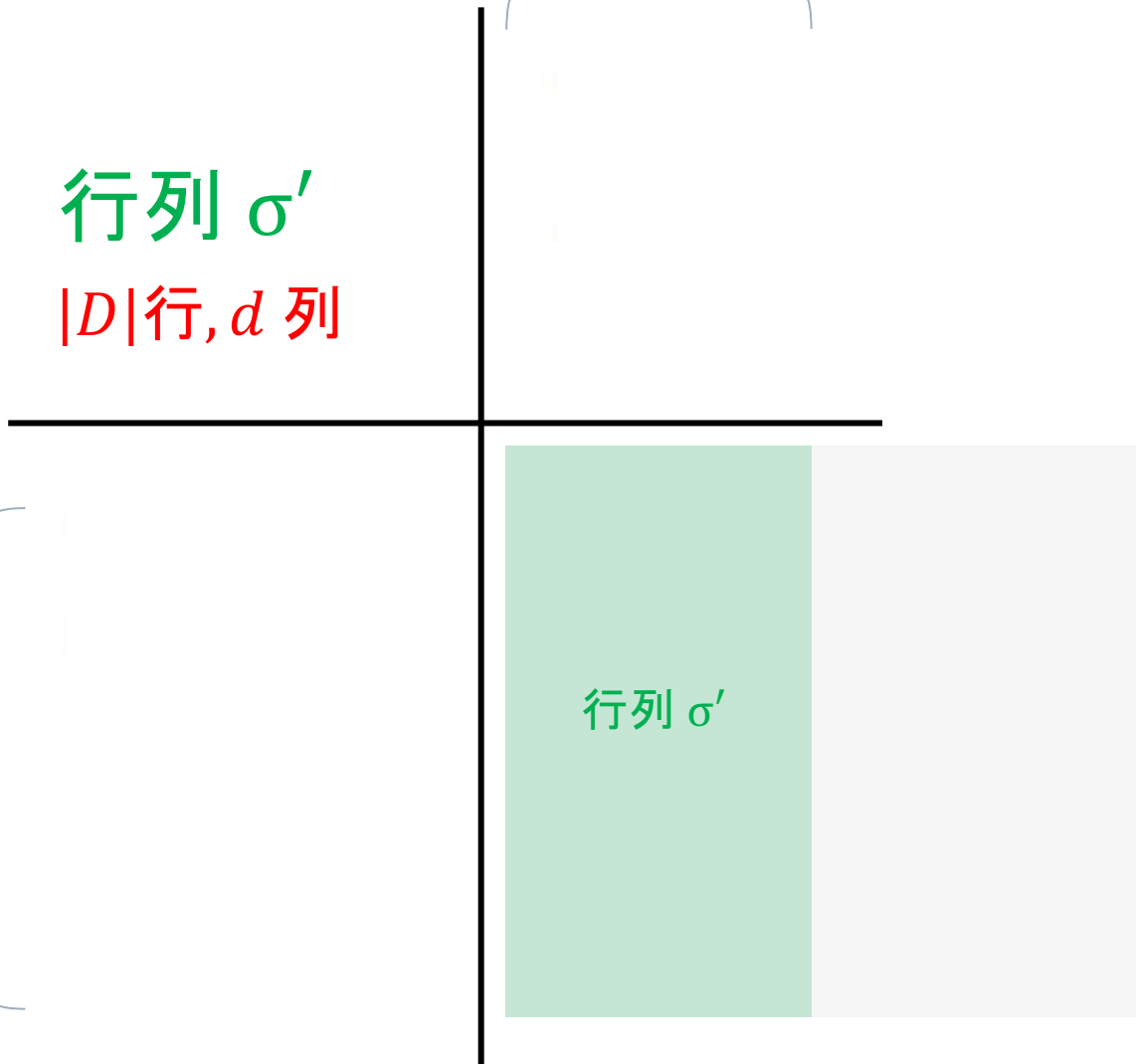
$d$

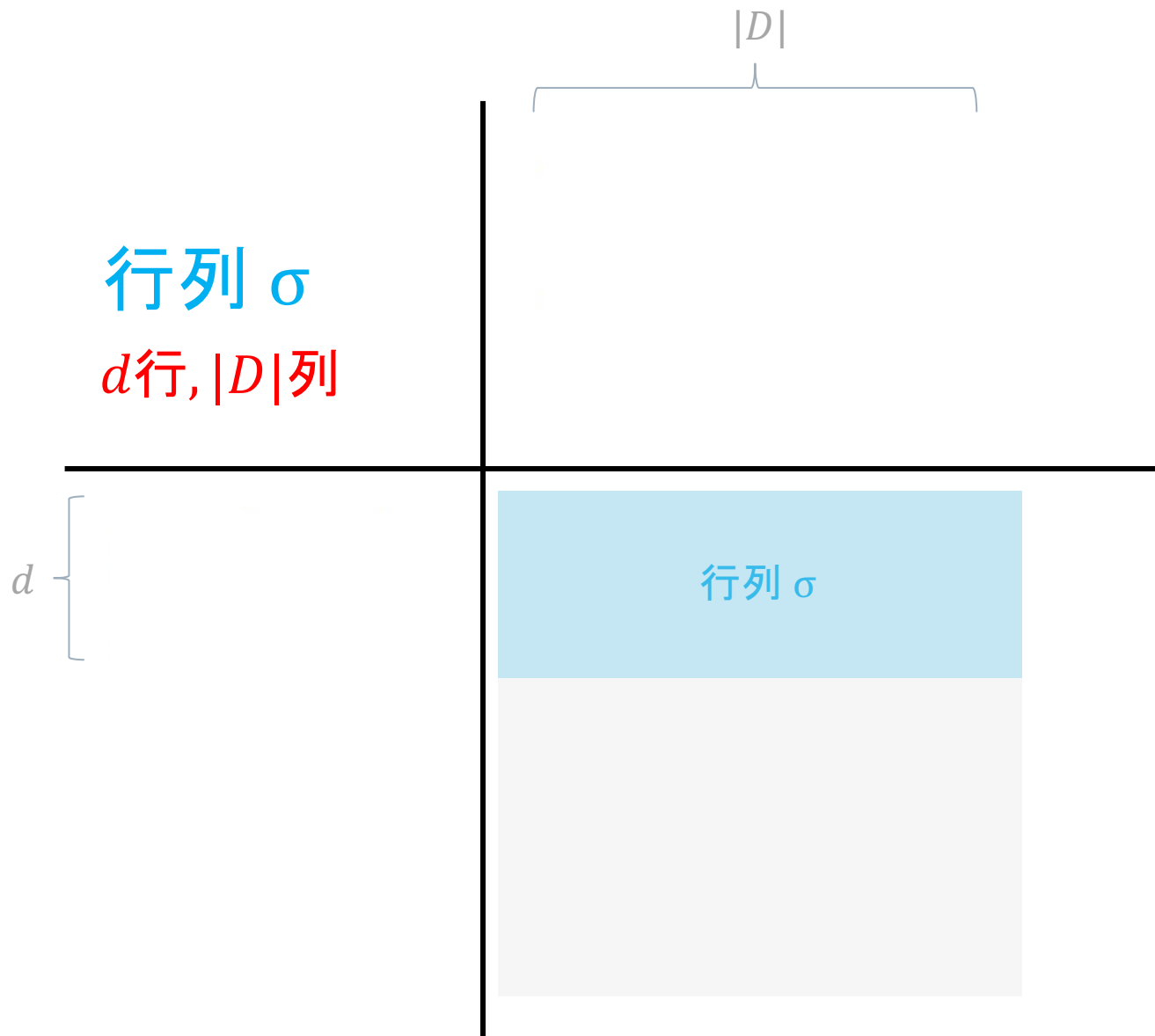
行列  $\sigma'$

$|D|$ 行,  $d$ 列

行列  $\sigma'$

$|D|$





行列  $\sigma'$

$|D|$ 行,  $d$ 列



$\times$

行列  $\sigma$

$d$ 行,  $|D|$ 列



$=$

行列  $\sigma'\sigma$

$|D|$ 行,  $|D|$ 列



$\|M - \sigma'\sigma\|$ が最小となるような、 $\sigma'$ と $\sigma$ を見つける。

# 埋め込みは $M$ の列の低次元近似

写像  $\sigma$  は  $M$  の最適な低ランクの近似である。

つまり、を  $d \ll |D|$  で、それぞれのサイズが  $|D| \times d$  と  $d \times |D|$  である  $\sigma'$  と  $\sigma$  で、 $\|M - \sigma'\sigma\|$  が最小となるような、 $\sigma'$  と  $\sigma$  を見つける。

要するに、ニューラル埋め込みは  $M$  の列の低次元近似に過ぎないということである。

したがって、埋め込みが示す驚くべき特性は、ニューラル・モデルの魔法のような特性の結果というよりも、テキスト・コーパスに見られる言語データの根底にある代数的構造の結果である。

実際、 $M$ の列、またはその低次元近似を明示的な単語ベクトル表現として直接使用することで、ニューラル単語埋め込みと同等の結果が得られることが示されている。

興味深いことに、もう1つの因子 $\sigma'$ は、学習されたDNNの第2層として読むことができるが、関連する言語情報が含まれているにもかかわらず、通常は破棄される。

まとめると、語埋め込みの数学的ストーリーは次のように進む。

まず、語彙の集合 $D$ から自由ベクトル空間 $\mathbb{R}^D$ に移行する。

$\mathbb{R}^D$ には意味のある言語情報はないが、 $D$ の構造に関する限定的な情報を配置できる大きな構造集合を提供する。

具体的には、この限られた情報とは、 $|D| \times |D|$  行列  $M$  であり、テキストのコーパスの中で単語が他の単語とどのように組み合わさっているかについてのたまかな統計データから構成される。

$$D \hookrightarrow \mathbb{R}^D \rightarrow \mathbb{R}^d$$

$$D \hookrightarrow \mathbb{R}^D$$

aardvark  $\mapsto (1, 0, 0, 0, \dots, 0, 0)$

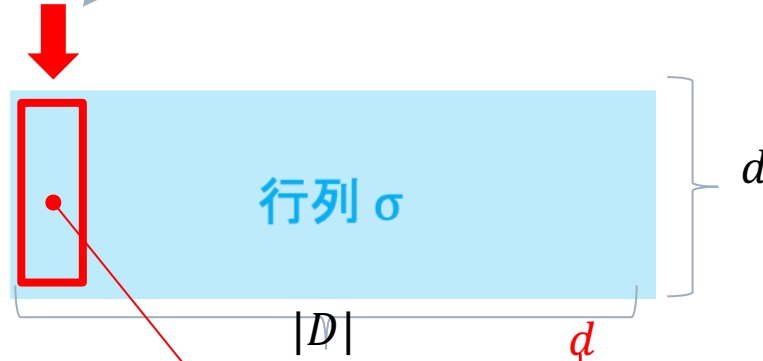
aardwolf  $\mapsto (0, 1, 0, 0, \dots, 0, 0)$

$\vdots$

zyzzyva  $\mapsto (0, 0, 0, 0, \dots, 0, 1)$ .

$\mathbb{R}^D$

$$\sigma: \mathbb{R}^D \rightarrow \mathbb{R}^d$$



aardvark  $\mapsto (0.632, 0.370, -0.620, \dots, -0.475)$

$M$ の列、あるいは、もっと良ければ $M$ の低ランク因数分解の列が、ベクトル空間構造と相互作用することで、単語集合 $D$ に含まれないしは隠されていた統語的・意味的情報が明らかになる。

統計データの行列は、例えば、群の乗法表を表す行列よりも数学的には表面的に見えるが、 $D$ が持つかもしれない構造については何も仮定しないという点に魅力がある。

むしろ、それは $D$ が特定のコーパスでどのように使われてきたかの純粋な目撃証人である。それは、観察結果の集合が入力で、より形式的な構造が出力であるようなものだ。

だから、単語の埋め込みが、単語が生きている言語学的に意味のある空間を見つけるという重要なステップを達成したのであれば、次のステップは、その空間の根底にある構造が何であるかをよりよく理解することである。

ベクトルの引き算が、意味のある類似性を反映していると事後的に認識できることは、さらに多くの発見があり得ることを示唆している。

この次の議論では、行列 $M$ の低ランク因数分解には、Singular value decomposition (SVD) を用いた厳密解があることを理解することが重要である。このアナロジーを十分に理解するために、初歩的な観点から行列を復習しておくといだろう。

## 參考資料

1. Distributed Representations of Words and Phrases and their Compositionality  
*Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, 2013*  
<https://arxiv.org/abs/1310.4546>
2. Neural word embedding as implicit matrix factorization  
*Omer Levy and Yoav Goldberg, 2014*  
[https://papers.nips.cc/paper\\_files/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf](https://papers.nips.cc/paper_files/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf)





# Part 3

意味の空間から意味の構造へ

言語の意味の数学的構造

# この章の構成

意味の空間から意味の構造へと進むためには、すこし準備が必要である。ここでは、いくつかのツールが提示され、そのカテゴリー論的位置付けが行われている。

同時に、この章では、embeddingの基礎となった線形代数的アプローチとカテゴリー論的なアプローチ「**パラレル**」な関係を示すことが、一つの目標となっている。

具体的には、体 $k$ 上に値を取る集合 $X$ 上の関数と、集合のカテゴリーSet または別のenrich化されたカテゴリー上に値を取るfunctorとの比較に注意してほしい。

## Part 3 Agenda

### 意味の空間から意味の構造へ

- singular value decomposition
- profunctor
- Isbell adjunction
- Formal concept



意味の空間から意味の構造へ  
singular value decomposition

この節では、体 $k$ 上に値を取る集合 $X$ 上の関数と、集合のカテゴリリーSet または別のenrich化されたカテゴリリー上に値を取る functorとの比較を念頭に置いて欲しい。

## 行列: $X \times Y \rightarrow k$

さて、行列について考えてみよう。

有限集合  $X, Y$  が与えられたとしよう。体  $k$  に値を取る  $X$ - $Y$  行列は、関数  $m: X \times Y \rightarrow k$  である。

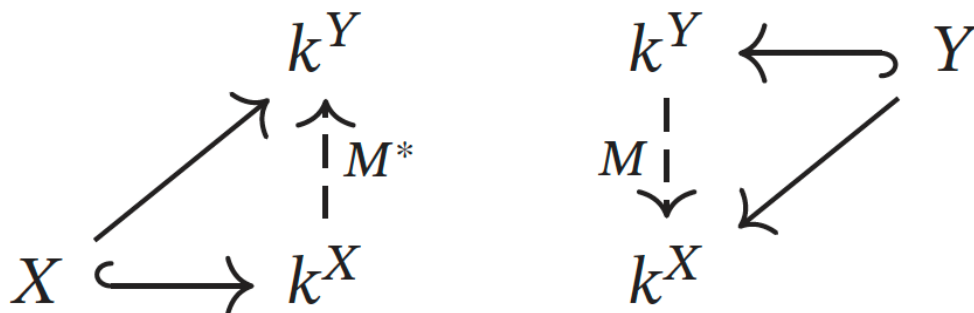
簡単な currying ( $X \rightarrow (Y \rightarrow k), Y \rightarrow (X \rightarrow k)$ ) で、 $x \mapsto m(x, -)$  と  $y \mapsto m(-, y)$  で定義される割り当てで、 $m$  は関数  $X \rightarrow k^Y$  と  $Y \rightarrow k^X$  を定義することがわかる。

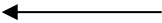
$X$  と  $Y$  の要素を順序づければ、関数  $m$  は、 $|X|$  個の行と  $|Y|$  個の列からなる長方形の数字で表現される。ここで、 $x$  行  $y$  列の数字は、 $m(x, y)$  である。

この時、関数  $m(x, -) \in k^Y$  は、行列  $m$  の  $x$  番目の行と同一視できる。この関数は、 $Y$  の要素と同じ数のエントリーを持ち、 $y$  を行の中の  $y$  番目のエントリーに送る関数を定義する。

同様に、 $m$  の  $y$  列は、関数  $m(-, y) \in k^X$  を表現する。

$X \subset k^X, Y \subset k^Y$  なので、写像  $X \rightarrow k^Y$  と  $Y \rightarrow k^X$  の定義域を線形に  $k^X, k^Y$  に拡大すると、線型写像  $M^*: k^X \rightarrow k^Y$  と  $M: k^Y \rightarrow k^X$  が得られる。この線型写像は、もちろん、行列  $M$  と、その転置行列  $M^*$  に関連している。



$$M^*M$$


$M$ と $M^*$ の合成、 $MM^*: k^X \rightarrow k^X$ 、 $M^*M: k^Y \rightarrow k^Y$ は、特別な性質を持っている線形変換である。


$$MM^*$$

$M$ と $M^*$ の合成、 $MM^*: k^X \rightarrow k^X$ 、 $M^*M: k^Y \rightarrow k^Y$ は、特別な性質を持っている線形変換である。

## $M^*M$ 、 $MM^*$ の性質

$M = U\Sigma V^*$  とSVD分解できたとしよう。

$M^* = (U\Sigma V^*)^* = V\Sigma^*U^*$  で、 $U^*U = V^*V = I$  だから、

$$M^*M = V\Sigma^*U^*U\Sigma V^* = V(\Sigma^*\Sigma)V^*$$

$$MM^* = U\Sigma V^*V\Sigma^*U^* = U(\Sigma\Sigma^*)U^*$$

右辺は、左辺の固有値分解である。よって、

- $V$  の列ベクトル(右特異ベクトル)は  $M^*M$  の固有ベクトルである。
- $U$  の列ベクトル(左特異ベクトル)は  $MM^*$  の固有ベクトルである。
- $\Sigma$  の非ゼロ要素は  $M^*M$  または  $MM^*$  の非ゼロ固有値の平方根である。

# singular value decomposition

もし基礎にある体 $k$ を実数 $\mathbb{R}$ に固定すると(実行列は $AA^* = A^*A$ )  
だから $normal$ である)、**スペクトル定理**から、それぞれ $MM^*$ と  
 $M^*M$ の固有ベクトルからなる  $k^X$ の正規直交基底 $\{u_1, \dots, u_m\}$ と $k^Y$   
の正規直交基底 $\{v_1, \dots, v_n\}$ を得ることができる。それは、  
 $r = \min(m, n)$ とすると、非負の固有値  $\{\lambda_1, \dots, \lambda_r, 0, \dots, 0\}$ を共有  
している。

このデータは、**行列 $M$ を  $M = U\Sigma V^*$ と因数分解する**のに、利用で  
きる。この分割は、 $M$ のsingular value decomposition と言  
われるものだ。

$\{u_i\}$ は $U$ の列ベクトルで、 $\{v_j\}$ は $V^*$ の行ベクトルである。そして、 $\Sigma$ は、 $m \times n$ の対角行列で、その $i$ 番目のエントリは $\sigma_i = \sqrt{\lambda_i}$ である。行列 $U$ と $V$ は、 $U^*U = I, V^*V = I$ を満たす。

SVDの用語では、非負の実数 $\sigma_i$ は、 $M$ のsingular value と呼ばれ、ベクトル $\{u_1, \dots, u_r\}, \{v_1, \dots, v_r\}$ は、それぞれ $M$ の左あるいは右のsingular ベクトルと呼ばれる。

また、 $k^X \times k^Y$ に属するベクトル $\{(u_1, v_1), \dots, (u_r, v_r)\}$ は次のようにお互いに関連している。

$$M^*u_j = \sigma_j v_j \text{ かつ } Mv_j = \sigma_j u_j$$

さらに、これらのペアは、対応するsingular value が $\sigma_i \leq \sigma_j$ を満たす時、 $(u_i, v_i) \leq (u_j, v_j)$  と順序づけられる。

最後に、rankが大きくても $s$ で Frobenius normで行列 $M$ に最も近い行列 $M'$ は、 $M' = U\Sigma'V^*$ で与えられる。ここで、 $\Sigma'$ は対角行列で対角線上の非負のsingular valueの最大のものは $s$ だけである。

$\Sigma'$ のすべての0エントリーのせいで、積 $U\Sigma'V^*$ に寄与しない  
 $U, \Sigma', V^*$ の部分を削除することで、 $M' = U'\Sigma''V'^* \approx M$ を得る。

ここで、 $U'$ は $m \times s$ 行列で、 $V'^*$ は $s \times n$ 行列、 $\Sigma''$ は対角線上に $M$ の最大のsingular value  $s$ を持つ  $s \times s$ の対角行列である。

Principal component analysis (PCA「主成分分析」)は、行列の低ランクのコンポーネントへの分解のこの近似を、高次元のデータの次元の削減に利用している。

## 参考資料

- Tai-Danae Bradley,  
At the Interface of Algebra and Statistics  
<https://arxiv.org/abs/2004.05631>
- マルレク「密度行列  $\rho$  で理解する確率の世界」  
<https://www.marulabo.net/docs/density2/>

# Singular Value Decomposition

全ての $m \times n$ の行列 $M$ は、次の形に分解できる。

$$M = V \Sigma U^\dagger$$

ここに、 $V$ は $m \times m$ の、 $U$ は $n \times n$ ユニタリ行列

$\Sigma$ は、 $D$ を $m \times m$ の対角行列とした時、 $[D|0]$ の形で表される。

$\Sigma$ の0の作用で、 $U^\dagger$ の一部は機能しないのでそれを省いたもの $U_0^\dagger$ とすると

$$M = V D U_0^\dagger$$

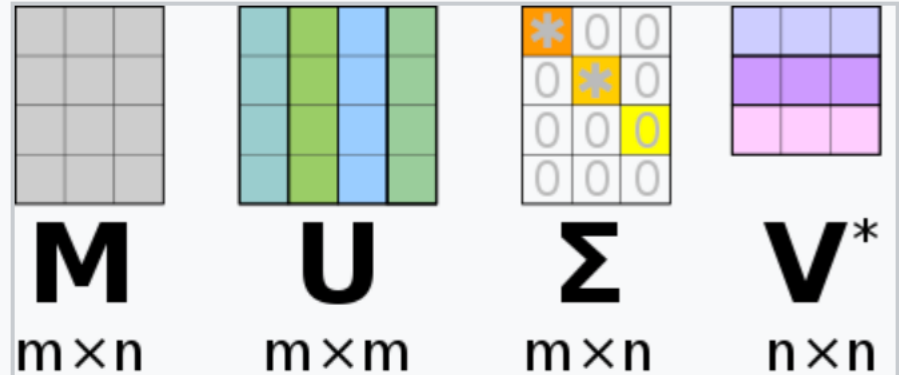
$$\begin{bmatrix} * & * & * \\ * & * & * \end{bmatrix} = \begin{bmatrix} * & * \\ * & * \end{bmatrix} \begin{bmatrix} * \\ * \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}$$

$M = V D U_0^\dagger$

# SVDのvariants

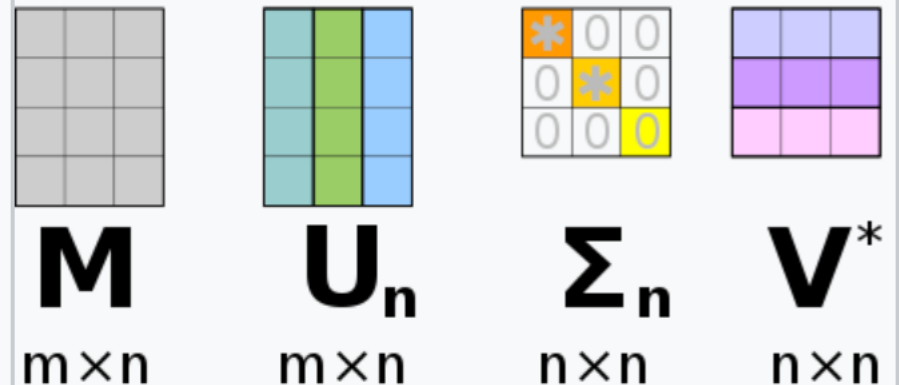
$$m > n$$

## Full SVD



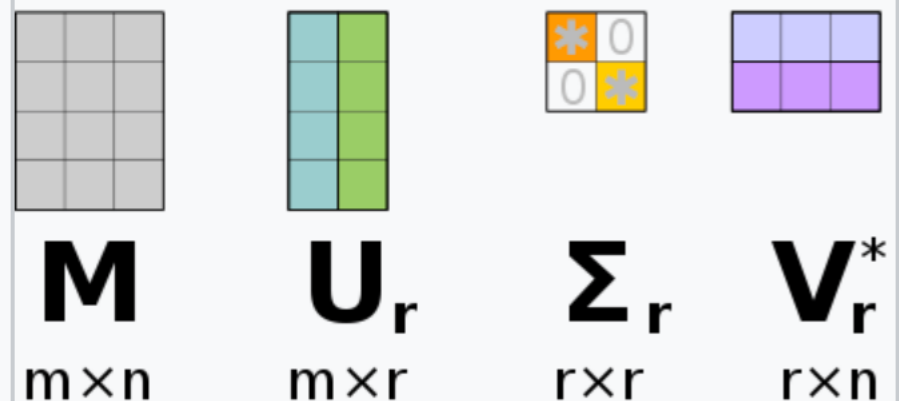
## Thin SVD

remove columns of U not corresponding to rows of  $V^*$



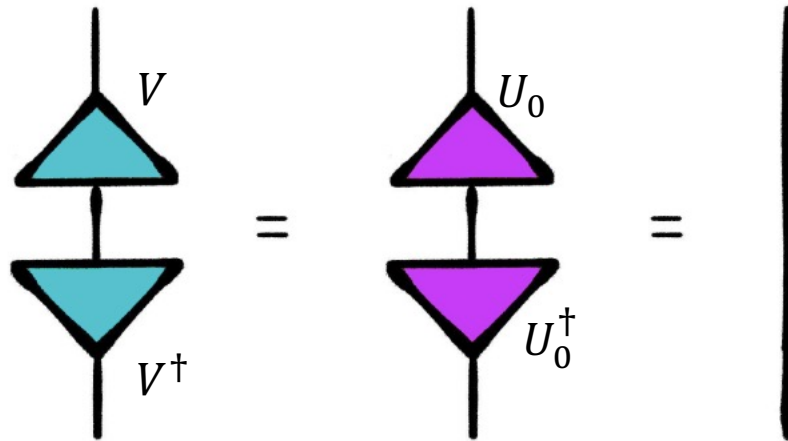
## Compact SVD

remove vanishing singular values and corresponding columns/rows in U and V

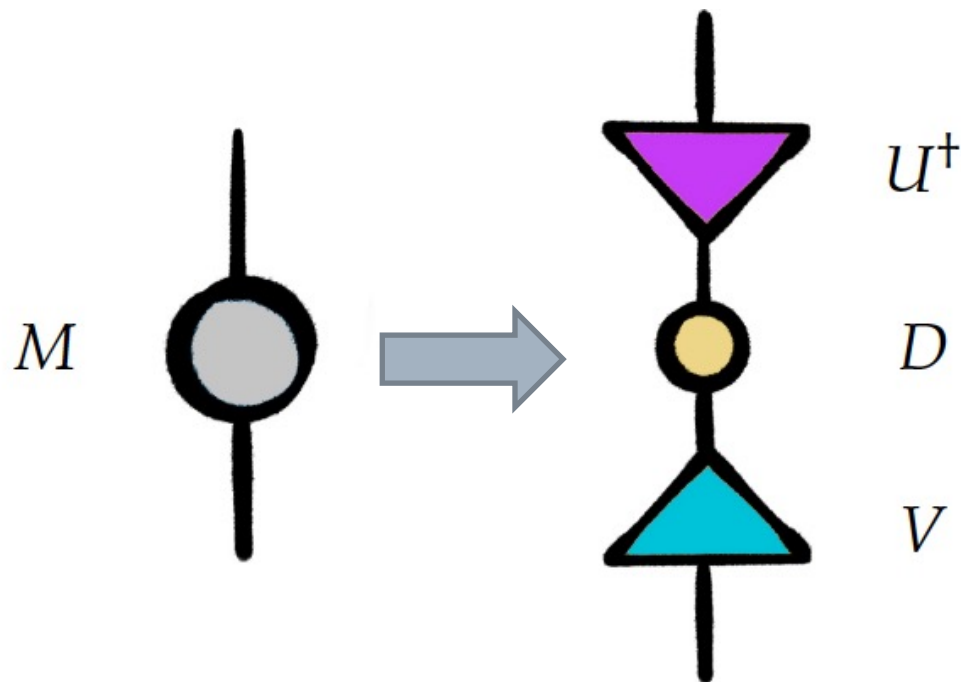


# $U_0$ は isometry, $V$ はunitary

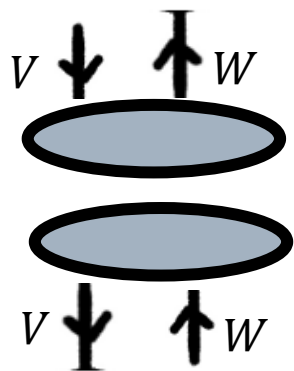
$U_0$  はもはやユニタリではないが、 $U_0 U_0^\dagger = id$  で isometry である。  
また、 $V$ はユニタリであるので、次の図式が成り立つ。



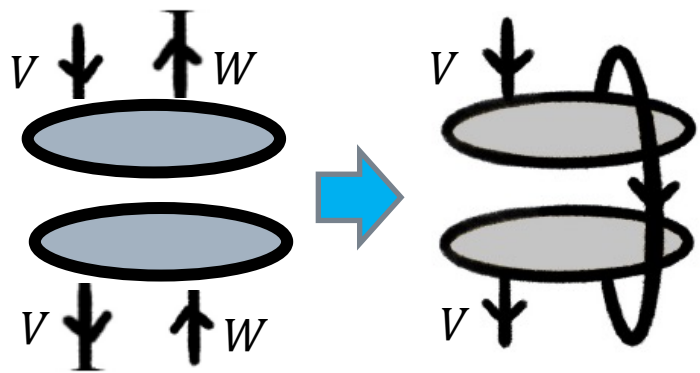
# Singular Value Decomposition の tensor network での表現



# $\rho_V, \rho_W$ を tensor network で表示する

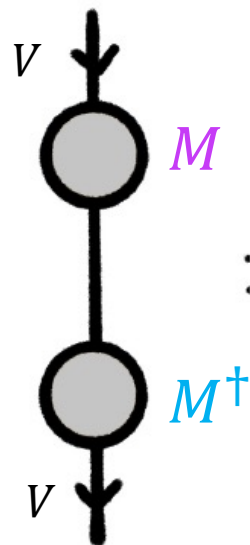
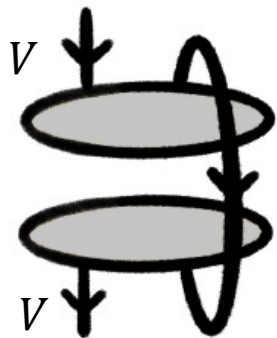
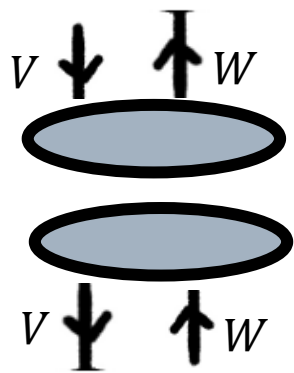


$$\rho = |\psi\rangle\langle\psi|$$



$$\rho = |\psi\rangle\langle\psi|$$

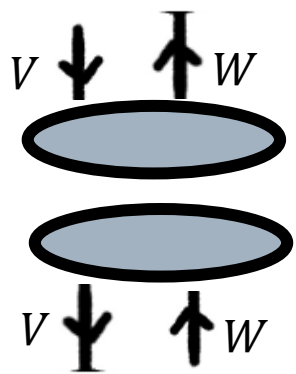
$$\rho_V$$



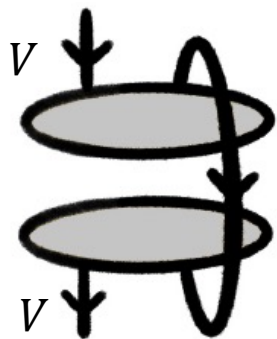
$$\rho = |\psi\rangle\langle\psi|$$

$$\rho_V$$

$$M^\dagger M$$

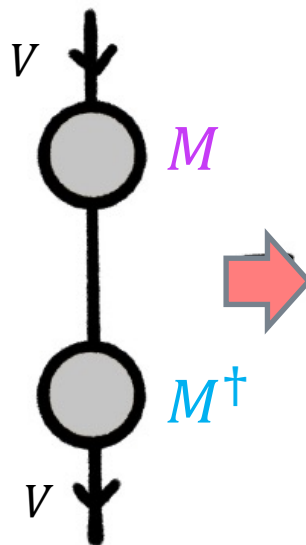


$$\rho = |\psi\rangle\langle\psi|$$

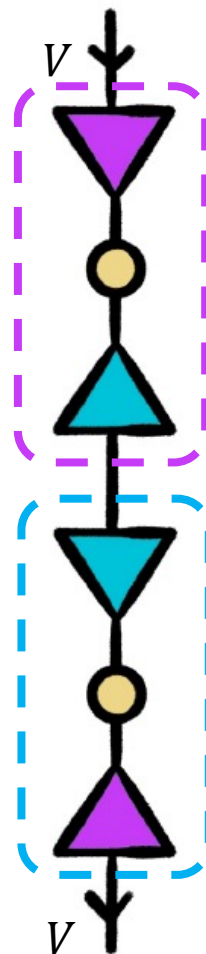


$$\rho V$$

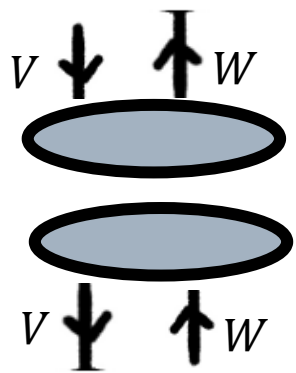
=



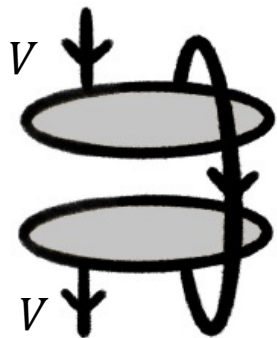
$$M^\dagger M$$



$$UD^\dagger V^\dagger VDU^\dagger$$

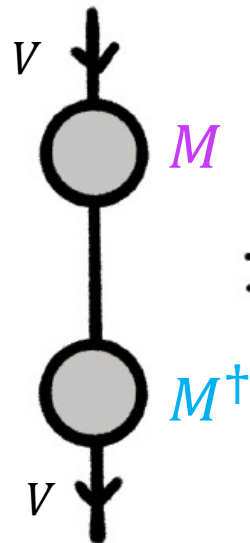


$$\rho = |\psi\rangle\langle\psi|$$



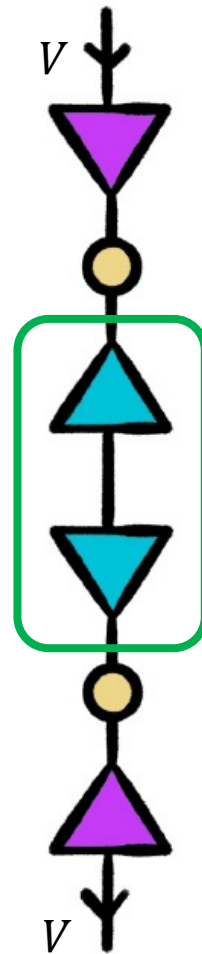
$$\rho V$$

=



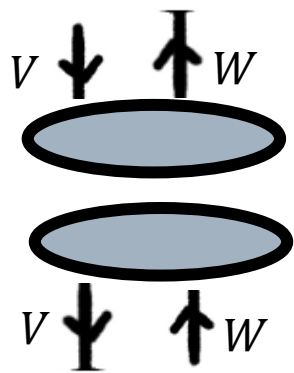
$$M^\dagger M$$

=

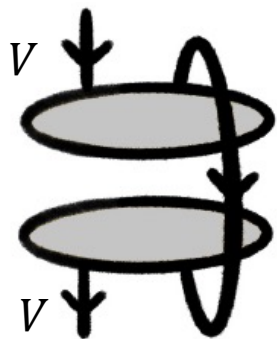


$$U D^\dagger V^\dagger V D U^\dagger$$

$$\rho_V = M^\dagger M = UD^2U^\dagger$$

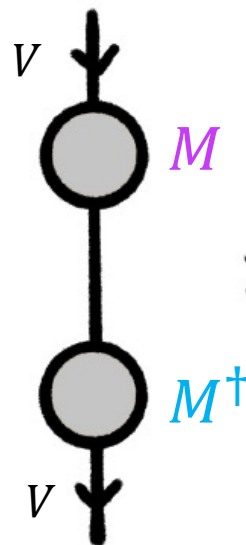


$$\rho = |\psi\rangle\langle\psi|$$



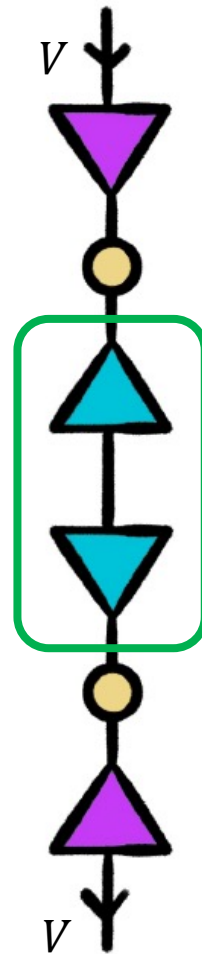
$$\rho_V$$

=

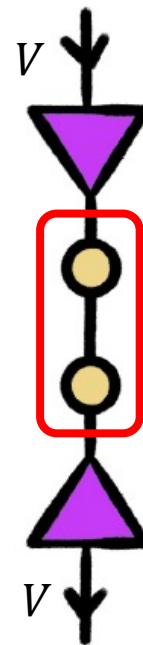


$$M^\dagger M$$

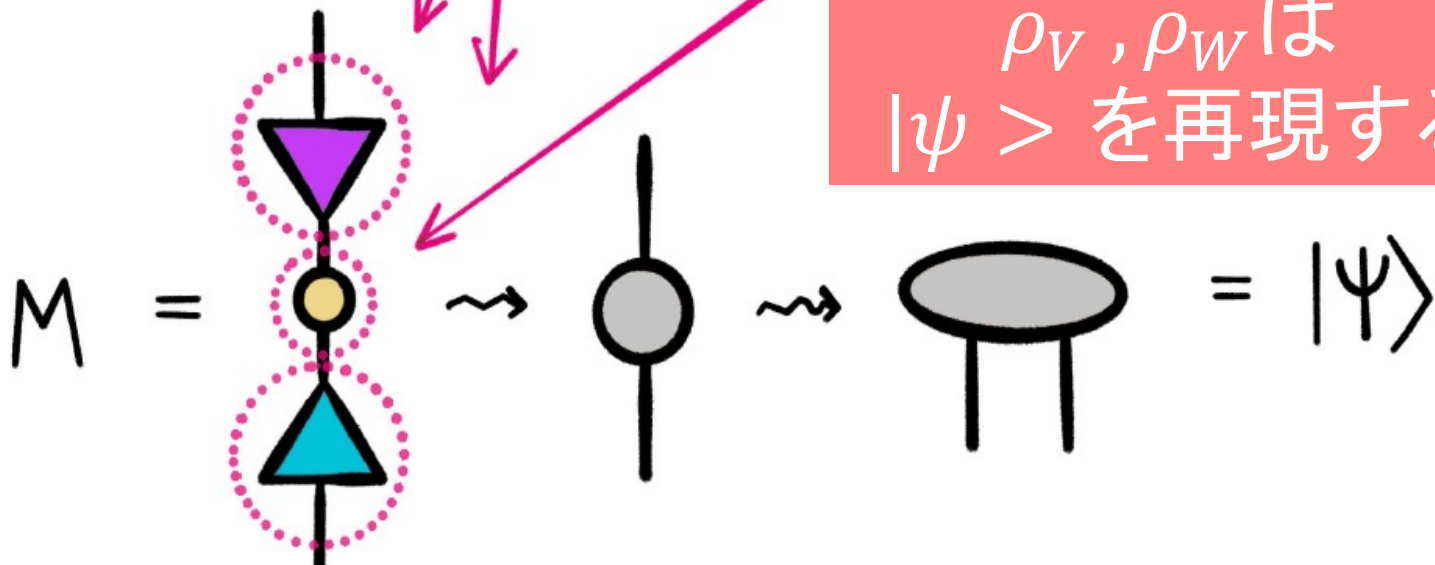
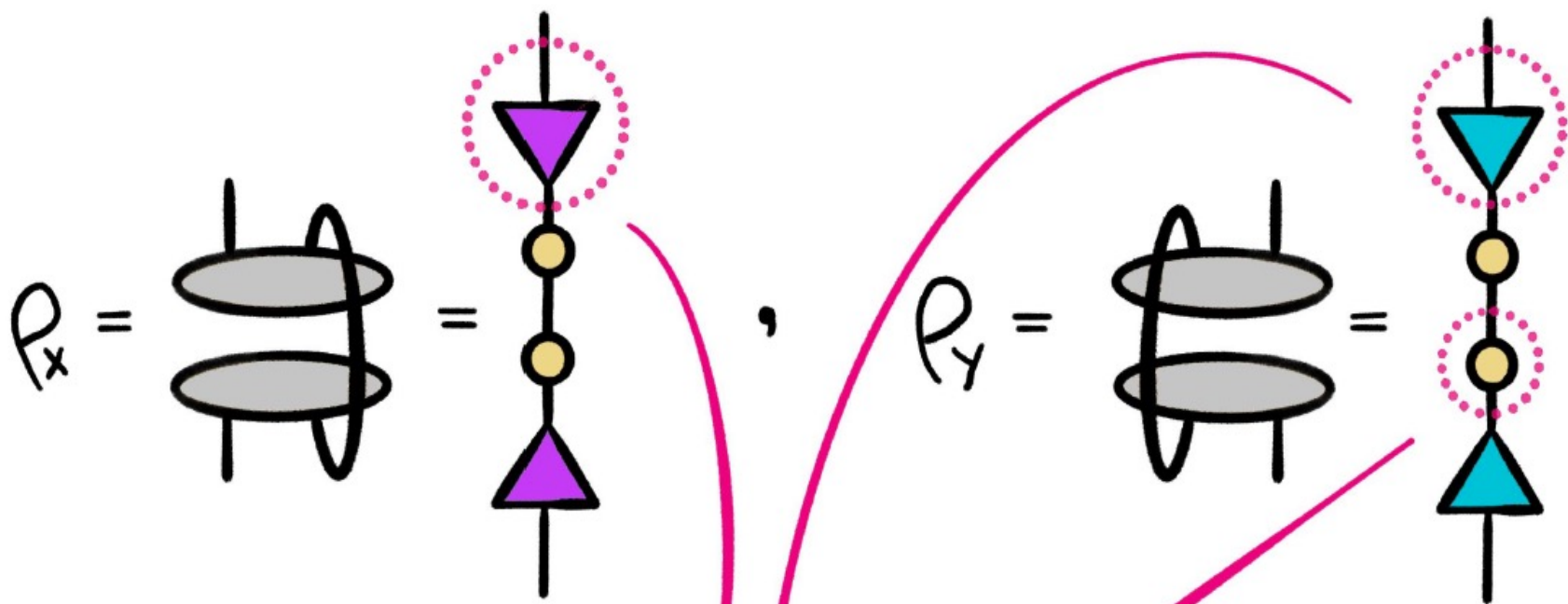
=



$$UD^\dagger V^\dagger V D U^\dagger$$



$$UD^2U^\dagger$$





意味の空間から意味の構造へ

profunctor

マルレク 言語の意味の数学的構造

# カテゴリー論への移行 行列からprofunctorへ

線形代数からカテゴリー理論に移ると、驚くほどよく似た話が見えてくる。

2つのカテゴリー $C$ と $D$ が与えられたとき、 $C$ - $D$ 行列の類比は **profunctor** と呼ばれるもので、集合のカテゴリー  $Set$  に値を取る functor  $f: C^{op} \times D \rightarrow Set$  となる。

前述したように、ここでも「op」はfunctorのvarianceを示すために使われ、正確を期すために必要であるが、初見では無視できる。専門家であれば、このopが興味深い数学的対称性に関与する状況を知っているはずだが、ここで説明する線形代数とのアナロジーでは、行と列の間の一種の転置を示すものと考えることができる。

profunctor  $f: C^{op} \times D \rightarrow Set$

もしprofunctorの定義域の二つのカテゴリー  $C^{op}$  と  $D$  が離散カテゴリーとして見て有限集合であるならば、profunctorは単に二つの要素のペアでindexされた集合の集まりである。それは数字の代わりに集合をエントリーとする行列である。

簡単なcurrying で、 $C, D$  上のオブジェクト  $c, d$  に対する定義、 $c \mapsto f(c, -), d \mapsto f(-, d)$  で、profunctor  $f$  は、ペアのfunctor  $C \rightarrow (Set^D)^{op}$  と  $D \rightarrow Set^{C^{op}}$  を定義することがわかる。

$$f(c, -): C \rightarrow (Set^D)^{op} \text{ と}$$
$$f(-, d): D \rightarrow Set^{C^{op}}$$

線形代数の場合と同様に、functor  $f(c, -)$  は行列  $f$  の  $c$  行目の集合として描くことができる。このfunctorは functor  $D \rightarrow Set$  を定義し、このfunctorで  $D$  の  $j$  番目のオブジェクトは、行  $f(c, -)$  の  $j$  番目のオブジェクトの集合にマップされる。

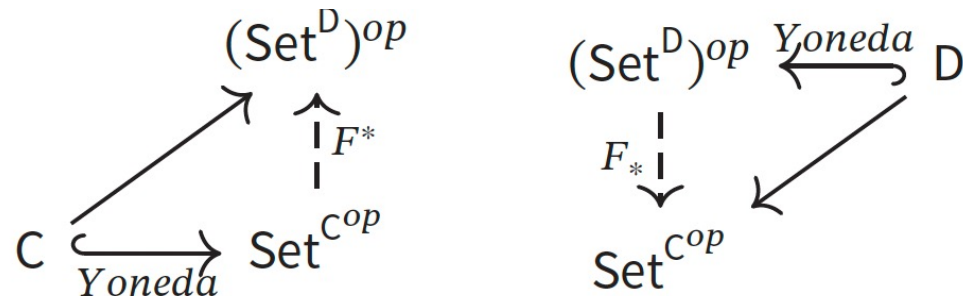
同様に、functor  $f(-, d)$  は行列  $f$  の  $d$  列目の集合を表す。このfunctorは functor  $C \rightarrow Set^{op}$  を定義し、このfunctorで  $C$  の  $i$  番目のオブジェクトは、列  $f(-, d)$  の  $i$  番目のオブジェクトの集合にマップされる。

# presheaf と Yoneda embedding

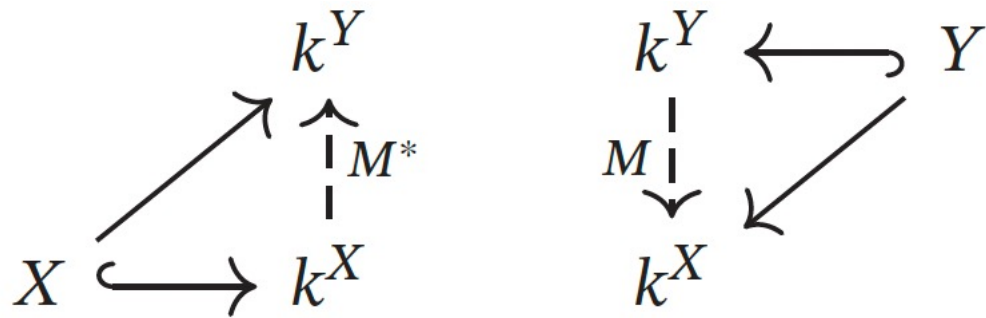
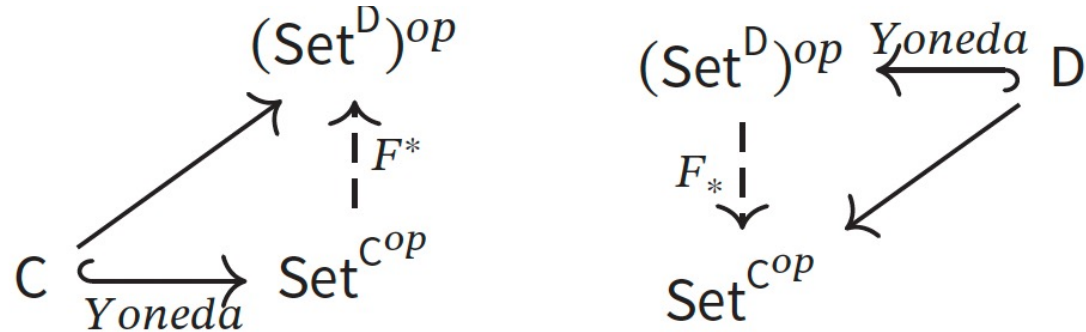
カテゴリーが、Yoneda (あるいは co-Yoneda) embedding で presheaf のカテゴリーに埋め込まれると考えれば、functor  $C \rightarrow (Set^D)^{op}$  と  $D \rightarrow Set^{C^{op}}$  は、一意に、次の functor  $F^*, F_*$  に拡張できる。それらは、それぞれ colimit と limit を保存する。

$$F^*: Set^{C^{op}} \rightarrow (Set^D)^{op}$$
$$F_*: (Set^D)^{op} \rightarrow Set^{C^{op}}$$

図で表すと次のようになる。



# カテゴリー論の場合



# 線形代数の場合

# $F^*F_*$ と $F_*F^*$ の性質

$$F^*F_*: (\text{Set}^D)^{op} \rightarrow (\text{Set}^D)^{op}$$
$$F_*F^*: \text{Set}^{C^{op}} \rightarrow \text{Set}^{C^{op}}$$

opを外してわかりやすくすると

profunctorの場合

$$F^*F_*: \text{Set}^D \rightarrow \text{Set}^D$$
$$F_*F^*: \text{Set}^C \rightarrow \text{Set}^C$$



行列の場合

$$MM^*: k^X \rightarrow k^X$$
$$M^*M: k^Y \rightarrow k^Y$$



意味の空間から意味の構造へ

Isbell adjunction

マルレク 言語の意味の数学的構造

# Isbell adjunction

線型写像  $M$  とその transpose  $M^*$  の合成で定義される線型写像が特別な意味を持つように、随伴する functor  $F^*$  と  $F_*$  は特別な性質を持っている。

この随伴関係は、Isbell adjunctionとして知られている。

$$F^* : \mathit{Set}^{C^{op}} \rightleftarrows (\mathit{Set}^D)^{op} : F_*$$

John Baez は、この Isbell adjunction を「数学の宝石」と呼んだ。(本誌2023年1月号のコラム記事)

# profunctorのnuclei

合成されたfunctor  $F^*F_*$  と  $F_*F^*$  の下で同型なオブジェクトは profunctor  $f$  のnuclei「核」と呼ばれ、行列の左右の特異ベクトルに類似している。

このnucleiを  $C^{op} \times D$  のオブジェクトのペア  $(c_i, d_i)$  にまとめることができる。この時、次が成り立つ。

$$F^*c_i \cong d_i \text{ かつ } F_*d_i \cong c_i$$



$$M^*u_j = \sigma_j v_j \text{ かつ } Mv_j = \sigma_j u_j$$

( 行列のSVDの場合には、次の式だった。 )

nuclei  $\{(c_i, d_i)\}$  自身、重要な構造を持つ。それは、あるカテゴリーを、complete で cocompleteなものに組織する。

行列の特異ベクトルのペア  $\{(u_i, v_i)\}$  もある構造を持っている。それらは、特異値の大きさの順に並べられる。特異値の大きさ自身、とても重要である。(これらのペアは、対応するsingular value が  $\sigma_i \leq \sigma_j$  を満たす時、 $(u_i, v_i) \leq (u_j, v_j)$  と順序づけられる。)

profunctorのnuclei  $\{(c_i, d_i)\}$ は、それとは異なった、もっと入り組んだ複雑な構造を持っている。

なぜなら、このペアの結合は、さまざまな代数的な方法で可能であり、ペアの図式の、さまざまなカテゴリー論的な limit と colimit を得ることができるからだ。

言語学的な文脈では、このことは重要である。

なぜなら、profunctorのnuclei は、抽象的な単位のようなもので、カテゴリー論的な limit と colimit は、それら进行操作する方法を提供するからである。

このことは、次のセクションで具体的に示される。

今のところ、行列の特異ベクトルから得られる単語の埋め込みは、意味にベクトル空間の構造を重ねる方法として解釈する。

意味的類似性のような言語の特定の意味的側面については、ベクトル空間の構造が適しているが、ベクトル空間の構造を重ね合わせることで、他の意味を覆い隠してしまう可能性がある。

Isbell adjunctionは、言語の他の構造的特徴を明らかにするのに役立つかもしれない別の構造を提供する。

## 参考資料

IsBell adjunctionは、Isbell dualityともいう。

本文で参照されていたJohn Baez の論文は、“Isbell Duality”  
である。<https://www.ams.org/journals/notices/202301/rnoti-p140.pdf>

その要旨を紹介する。

第一に、任意のカテゴリリー $C$ に対して、カテゴリリー $[C^{op}, Set]$ へ写像が存在する。(  $[C^{op}, Set]$ は、 $C^{op}$ から $Set$ へのfunctor カテゴリリーを表す表記である)  
このカテゴリリーのオブジェクトは、カテゴリリー $C^{op}$ からカテゴリリー $Set$ へのfunctorであり、その射はfunctor間のnatural transformationである。

この写像は、任意の $X \in C$ を取ったとき、次のfunctorを返す。

$$\text{hom}(-, X) : C^{op} \rightarrow Set$$

このfunctorは、すべての $A \in C$ に対して、 $A$ から $X$ へのすべての射 $\text{hom}(A, X)$ の集合に移す。

# Yoneda embedding

この写像は、Yoneda embedding と呼ばれ、それ自身functorである。

$$y: C \rightarrow [C^{op}, Set]$$
$$X \mapsto \text{hom}(-, X)$$

Yoneda embeddingは、カテゴリー論では基本的である。

哲学的には、それは、あるオブジェクトは、それに入ってくる射の振る舞いで知ることができることを述べている。

それを実践的なツールとしてどのように使えばいいかを学ぶには少し時間があるかもしれないが、現代の教科書では、うまく説明されている。[3,4]

一つの洞察は、次のようなものである。

我々は、ある集合をとって、それをベースとしてベクトル空間を作ることがある。そのとき、あるカテゴリーCをより大きなカテゴリー  $[C^{op}, Set]$  の内部に存在しているものと考えると便利なのがよくある。

その理由は、 $[C^{op}, Set]$  は “colimit” を持つからである。

“colimit” は、ベクトル空間の線型結合に似ている。

例えば、 $F, G \in [C^{op}, Set]$  を次のように和を取ることができる。

$$(F + G)(X) = F(X) + G(X) \quad \forall X \in C^{op}$$

ここで右辺の、通常の実数の disjoint union である。

実際、 $[C^{op}, Set]$  は、カテゴリーC上のcolimitを持つ free カテゴリーなのである。

# co-Yoneda embedding

ただ、この話全体にdualなバージョンがあるのである！

あるオブジェクトは、そこから出ていく射の振る舞いによっても知ることができるのだ。

この事実は、co-Yoneda embeddingで把握できる。

$$z: C \rightarrow [C, Set]^{op}$$
$$X \mapsto \text{hom}(X, -)$$

colimitにdualな概念は、“limit”である。colimitが和の一般化であるように、limitは積の一般化である。

驚くべきことではないが、 $[C, Set]^{op}$ は、カテゴリC上の limitを持つfree カテゴリなのである。

1960年、Isbell [2] は、Yoneda embeddingとco-Yoneda embeddingの間に、素敵なつながりがあることに気付いた。その結果は、その後多くの著者によって明確にされてきた。[1]にそのレビューがある。

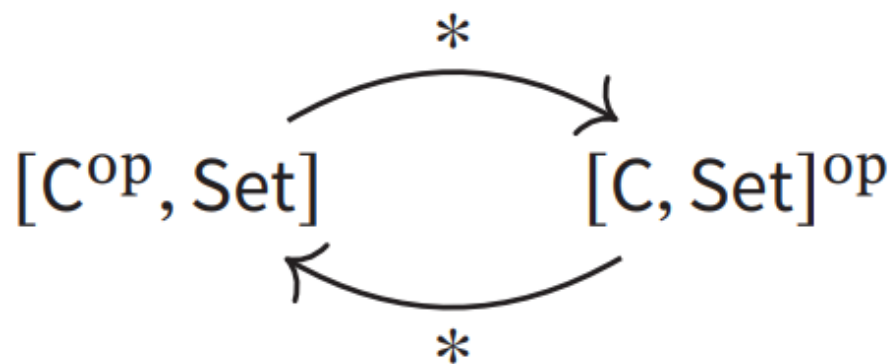
すべてのfunctor  $F: C^{op} \rightarrow Set$ は、次の式で与えられる Isbell conjugate  $F^*: C \rightarrow Set$ を持つのである。

$$F^*(X) = \text{hom}(F, y(X))$$

同様に、すべてのfunctor  $G: C \rightarrow Set$ は、次の式で与えられる Isbell conjugate  $G^*: C^{op} \rightarrow Set$ を持つ。

$$G^*(X) = \text{hom}(z(X), G)$$

Isbell conjugate の二つのバージョンは、次の図のように、行ったり戻ったりする二つのfunctor を与えるのである。



$$(\mathcal{O} \dashv \text{Spec}): \text{CoPresheaves} \overset{\mathcal{O}}{\underset{\text{Spec}}{\rightleftarrows}} \text{Presheaves}$$

その代わりに、Isbell dualityは、それらがadjoints (随伴)であるといっているのだ。すなわち、すべての  $F \in [C^{op}, Set]$  と  $G \in [C, Set]^{op}$  について、 $\text{hom}(F^*, G)$  と  $\text{hom}(F, G^*)$  が、自然に同型になるというのだ。

このことは、必ずしも有限次元とは限らないベクトル空間での次のような状況とにている。

$\text{Vect}$ を与えられた体上のすべてのベクトル空間のカテゴリーとするとき、dualを取ることが、 $\text{Vect}$ と $\text{Vect}^{op}$ の間を行ったり来たりするadjoint functorを定義する。

Isbell dualityは、さらなる盛大な発展の舞台をセットしている。  
例えば、[1]。

(中略)

しかしながら、Isbell dualityには、依然として広大な未開拓の地が残されている。おそらく、問題の一つは単純なものだ。それは、この数学の宝石が、いまだ広く知られてないということだ。

# References

- [1] T. Avery and T. Leinster, Isbell conjugacy and the reflexive completion, *Theory Appl. Categ.* 36 (2021), 306–347. Also available as arXiv:2102.08290.
- [2] J. R. Isbell, Adequate subcategories, *Illinois Jour. Math.* 4 (1960), 541–552.
- [3] T. Leinster, *Basic Category Theory*, Cambridge U. Press, Cambridge, 2014. Also available as arXiv:1612.09375.
- [4] E. Riehl, *Category Theory in Context*, Dover, New York, 2016. Also available as <http://www.math.jhu.edu/~eriehl/context.pdf> .
- [5] S. Willerton, Tight spans, Isbell completions and semitropical modules. Available as arXiv:1302.4370.

# Isbell duality

## duality between geometry and algebra

A general abstract adjunction

$$(\mathcal{O} \dashv \text{Spec}): \text{CoPresheaves} \underset{\text{Spec}}{\overset{\mathcal{O}}{\rightleftarrows}} \text{Presheaves}$$

Under the interpretation of presheaves as generalized spaces and copresheaves as generalized quantities modeled on ([Lawvere 86](#), see at space and quantity), **Isbell duality is the archetype of the duality between geometry and algebra** that permeates mathematics (such as Gelfand duality, Stone duality, or the embedding of smooth manifolds into formal duals of R-algebras).



意味の空間から意味の構造へ

Formal concept

マルレク 言語の意味の数学的構造

# Formal Concept Analysis

[https://en.wikipedia.org/wiki/Formal\\_concept\\_analysis](https://en.wikipedia.org/wiki/Formal_concept_analysis)

**Formal context** は次の三つ組  $K = \langle G, M, I \rangle$  である。

ここで、**Gはオブジェクトの集合、Mは属性の集合**であり、 $I \subseteq G \times M$ は、どのオブジェクトがどの属性を持つかを表す incidence と呼ばれる二項関係である。

オブジェクトの部分集合  $A \subseteq G$  と属性の部分集合  $B \subseteq M$  に対して、次のように2つの演算を定義する:

$$A' = \{m \in M \mid (g, m) \in I \text{ for all } g \in A\}$$

Aのすべてのオブジェクトが共有するすべての属性の集合

$$B' = \{g \in G \mid (g, m) \in I \text{ for all } m \in B\}$$

Bのすべての属性を共有するすべてのオブジェクトの集合。

Example for a formal context: "bodies of water"

オブジェクト

bodies of water		属性					
		<i>temporary</i>	<i>running</i>	<i>natural</i>	<i>stagnant</i>	<i>constant</i>	<i>maritime</i>
objects	canal		✓			✓	
	channel		✓			✓	
	lagoon			✓	✓	✓	✓
	lake			✓	✓	✓	
	maar			✓	✓	✓	
	puddle	✓		✓	✓		
	pond			✓	✓	✓	
	pool			✓	✓	✓	
	reservoir				✓	✓	
	river		✓	✓		✓	
	rivulet		✓	✓		✓	
	runnel		✓	✓		✓	
	sea			✓	✓	✓	✓
	stream		✓	✓		✓	
	tarn			✓	✓	✓	
	torrent		✓	✓		✓	
trickle		✓	✓		✓		

$A = \{\text{lagoon, puddle, sea}\}$  とすると

bodies of water		attributes					
		<i>temporary</i>	<i>running</i>	<i>natural</i>	<i>stagnant</i>	<i>constant</i>	<i>maritime</i>
objects	canal		✓			✓	
	channel		✓			✓	
	<b>lagoon</b>			✓	✓	✓	✓
	lake			✓	✓	✓	
	maar			✓	✓	✓	
	<b>puddle</b>	✓		✓	✓		
	pond			✓	✓	✓	
	pool			✓	✓	✓	
	reservoir				✓	✓	
	river		✓	✓		✓	
	rivulet		✓	✓		✓	
	runnel		✓	✓		✓	
	<b>sea</b>			✓	✓	✓	✓
	stream		✓	✓		✓	
	tarn			✓	✓	✓	
	torrent		✓	✓		✓	
	trickle		✓	✓		✓	

$A' = \{ \text{temporary, running, natural, stagnant, constant, maritime sea} \}$

bodies of water		attributes					
		<i>temporary</i>	<i>running</i>	<i>natural</i>	<i>stagnant</i>	<i>constant</i>	<i>maritime</i>
objects	canal		✓			✓	
	channel		✓			✓	
	lagoon			✓	✓	✓	✓
	lake			✓	✓	✓	
	maar			✓	✓	✓	
	puddle	✓		✓	✓		
	pond			✓	✓	✓	
	pool			✓	✓	✓	
	reservoir				✓	✓	
	river		✓	✓		✓	
	rivulet		✓	✓		✓	
	runnel		✓	✓		✓	
	sea			✓	✓	✓	✓
	stream		✓	✓		✓	
	tarn			✓	✓	✓	
	torrent		✓	✓		✓	
trickle		✓	✓		✓		

B = {temporary, maritime }とすると

bodies of water		attributes					
		<i>temporary</i>	<i>running</i>	<i>natural</i>	<i>stagnant</i>	<i>constant</i>	<i>maritime</i>
objects	canal		✓			✓	
	channel		✓			✓	
	lagoon			✓	✓	✓	✓
	lake			✓	✓	✓	
	maar			✓	✓	✓	
	puddle	✓		✓	✓		
	pond			✓	✓	✓	
	pool			✓	✓	✓	
	reservoir				✓	✓	
	river		✓	✓		✓	
	rivulet		✓	✓		✓	
	runnel		✓	✓		✓	
	sea			✓	✓	✓	✓
	stream		✓	✓		✓	
	tarn			✓	✓	✓	
	torrent		✓	✓		✓	
trickle		✓	✓		✓		

$B' = \{\text{lagoon, puddle, sea}\}$

bodies of water		attributes					
		<i>temporary</i>	<i>running</i>	<i>natural</i>	<i>stagnant</i>	<i>constant</i>	<i>maritime</i>
objects	canal		✓			✓	
	channel		✓			✓	
	<b>lagoon</b>			✓	✓	✓	✓
	lake			✓	✓	✓	
	maar			✓	✓	✓	
	<b>puddle</b>	✓		✓	✓		
	pond			✓	✓	✓	
	pool			✓	✓	✓	
	reservoir				✓	✓	
	river		✓	✓		✓	
	rivulet		✓	✓		✓	
	runnel		✓	✓		✓	
	<b>sea</b>			✓	✓	✓	✓
	stream		✓	✓		✓	
	tarn			✓	✓	✓	
	torrent		✓	✓		✓	
trickle		✓	✓		✓		

# closure operator

$$A \mapsto A'' = (A')' \text{ for } A \subseteq G \text{ (extent closure)}$$
$$B \mapsto B'' = (B')' \text{ for } B \subseteq M \text{ (intent closure).}$$

先の例では、

$A''$  は、すべてのオブジェクトからなる集合で

$B''$  は、running以外のすべての属性を含む集合となる。

# formal concept

Willeは、formal conceptのエレガントな定義を与えた:

(A,B)の組は、次の条件を満たせば、context (G,M,I)のformal conceptである:

$$A \subseteq G, B \subseteq M, A' = B, \text{ and } B' = A.$$

直観的に言えば、(A,B)がformal conceptとなるのは、AのすべてのオブジェクトがBのすべての属性を持つ場合である。

# Concept lattice of a formal context

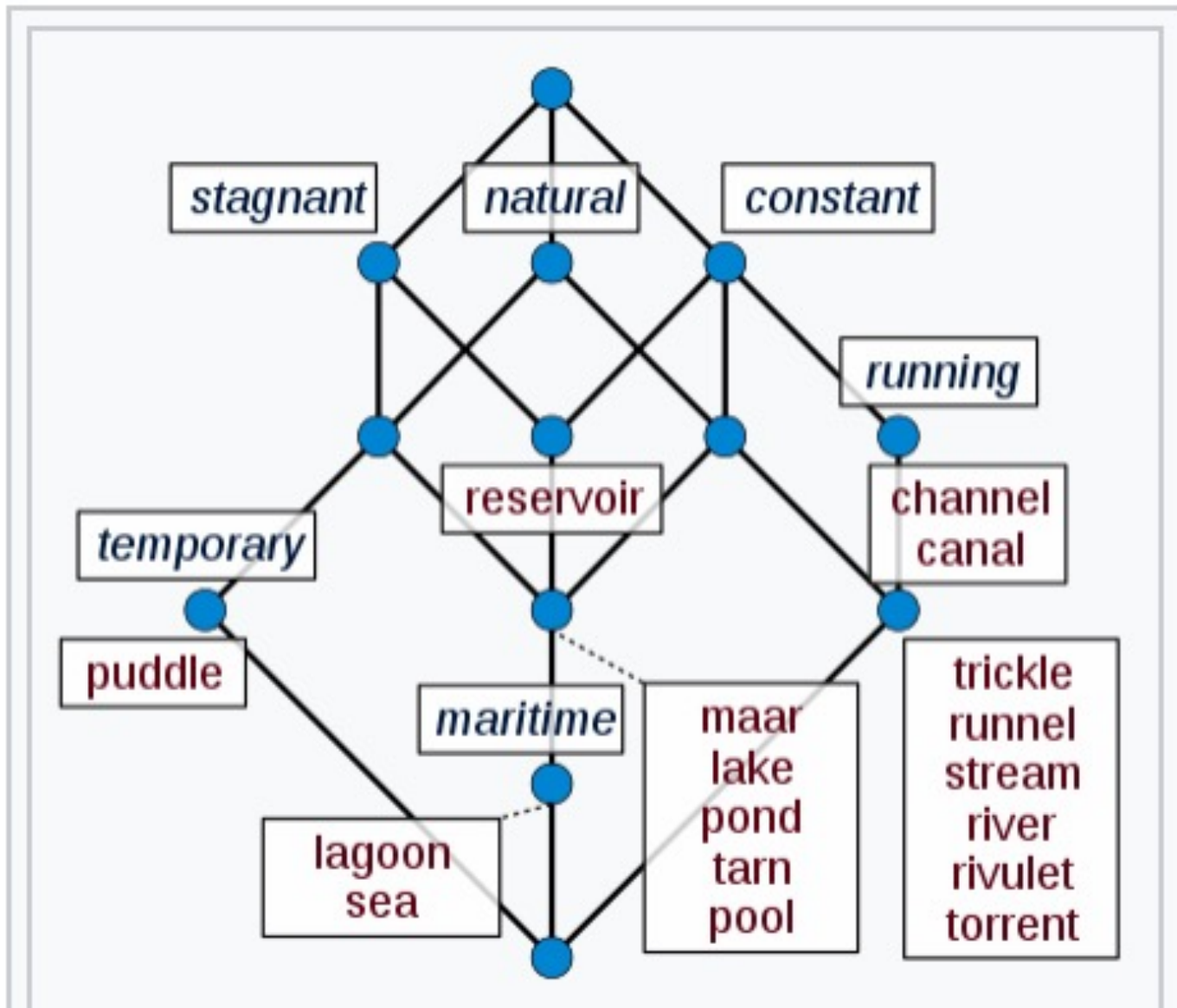
context  $K$ のformal concept  $(A_i, B_i)$ は、外延(オブジェクト)の包含、または同等に、内包(属性)の逆の包含によって(部分的に)順序付けることができる。

$K$ の任意の2つのformal concept  $(A_1, B_1)$ と $(A_2, B_2)$ に対して、 $A_1 \subseteq A_2$ のとき、 $(A_1, B_1) \leq (A_2, B_2)$ と言う。  
同様に、 $(A_1, B_1) \leq (A_2, B_2)$ は、 $B_1 \supseteq B_2$ のときである。

この順序では、formal conceptのすべての集合は、最大共通下位概念、つまりmeetを持つ。その外延は、その集合のすべての外延に共通するオブジェクトで構成される。

同様に、すべてのformal conceptの集合は最小共通の上位概念を持ち、その内包は、その概念の集合のすべてのオブジェクトが持つすべての属性から構成される。

これらのmeet操作とjoin操作はlattice(束)を定義する公理を満たす。逆に、すべての完全束は、あるformal contextのconcept latticeであることを示すことができる。



# Formal Concept

## Isbell adjunction の単純な例

柔軟さのため、enriched カテゴリーの観点からIsbell adjunction を 見ることは有用である。

ベースのカテゴリーを  $Set$  から、カテゴリー  $2 = \{0,1\}$  だとしよう。

この時、二つの有限集合  $X$  と  $Y$  の間の profunctor は、カテゴリー  $2$  上で enrich 化された 離散カテゴリー としてみれば、単なる関数  $r: X \times Y \rightarrow \{0,1\}$  で、それは、 $X \times Y$  上の 関係に他ならない。

## $R^*R_*$ と $R_*R^*$ の固定点

functor  $R^*: 2^X \rightarrow 2^Y$ と $R_*: 2^Y \rightarrow 2^X$ は、Formal Concept Analysis 理論でよく知られた対象である。

関数  $R^*$  は、 $X$ の部分集合  $A \subseteq X$ を次のように $Y$ の部分集合に移す。

$$R^*(A) = \{y \in Y \mid \forall x \in A. R(x, y) = 1\}$$

関数  $R_*$  は、 $Y$ の部分集合  $B \subseteq Y$ を次のように $X$ の部分集合に移す。

$$R_*(B) = \{x \in X \mid \forall y \in B. R(x, y) = 1\}$$

$R^*R_*$ と $R_*R^*$ の固定点は、formal concept と呼ばれる。

それらは、 $(A_i, B_i) \subset X \times Y$  のペアとして組織され、次が成り立つ。

$$R^*(A_i) = B_i \text{ かつ } R_*(B_i) = A_i$$

すべての formal concept の集合  $\{(A_i, B_i)\}$  には、

$$A_i \subseteq A_j \text{ の時かつその時に限り } (A_i, B_i) \leq (A_j, B_j)$$

という条件で、半順序関係が導入できる。

この条件は、 $B_i \supseteq B_j$  に等しい。

さらに、集合  $\{(A_i, B_i)\}$  は、complete lattice を構成する。

行列の特異値ベクトルのように、そこには、最大あるいは最小の formal concept があるのである。もっとある。

例えば、formarl concept の product と coproduct が、次のように定義できる。

$$(A_i, B_i) \wedge (A_j, B_j) := (A_i \cap A_j, R^* R_*(B_i \cup B_j))$$

$$(A_i, B_i) \vee (A_j, B_j) := (A_i \cup A_j, R_* R^*(B_i \cap B_j))$$

ポイントは、formal conceptのlimitとcolimitには、単純で有限な公式があるということだ。

それは、集合のunionとintersectionに似ているが、正確には違う。それは、profunctorのnucleiに見られるような代数的構造のアイデアを与えてくれる。

formal conceptのlimitとcolimitには、単純で有限な公式がある。それは、集合のunionとintersectionに似ているが、正確には同じではない。

ポイントは、それが、profunctorのnucleiでみてきたように、ある種の代数的構造のアイデアを与えるということである。

## 参考資料

先のセッションでは、彼女のIsbell adjunctionの議論を補足するものとして、BaezのIsbell Dualityの論文を紹介した。

今回は、彼女のFormal Conceptの議論を補足するものとして、Lawvereの論文“Adjointness in Foundation”を紹介する。基本的には、数学におけるFormalなものConceptualなものとの双対性についてAdjointnessという視点から論じたものだ。

1969年の古いものだが、カテゴリー論では重要な論文の一つだと思う。

<https://github.com/mattearnshaw/lawvere/blob/master/pdfs/1969-adjointness-in-foundations.pdf>

# 第一節：数学とその基礎における 形式と概念の双対性について

数学と呼ばれる正確な知識の追求は、形式的なものと概念的なものと呼ばれる2つの側面を本質的に含んでいるように思われる。

例えば、私たちは代数的に多項式を操作し、幾何学的に対応する曲線を視覚化する。あるいは、ある瞬間には群論の公理から定理を導くことに集中し、次の瞬間には、その定理が参照する実際の群のクラスについて考える。

このように、概念的なものはある意味で形式的なものの主題なのである。

ここでいう「基礎」とは、数学において普遍的なものを研究することである。したがって、この意味での「基礎」は、数学の「出発点」や「正当化」と同一視することはできない。

しかし、このように定義された「基礎」の他の果実の中には、数学のある分野から別の分野への移行や、どのような研究方向が関連しそうかをある程度判断するための指針が含まれていると思われる。

それ自体が数学の一部であるため、数学の基礎はまた形式的-概念的な双対性の一部である。

その形式的な側面において、基礎論はしばしば数学の形式的な側面に集中し、論理学を生み出した。より最近では、普遍性の探求は、カテゴリー論の形で概念的な転回も行っている。

カテゴリー論は、与えられた種(species)  $A$ の数学的対象のすべての射の総体を新たな数学的対象と見なし、これらの新たな数学的対象はすべて、 $A$ から独立した共通の非自明な種 $C$ に属することを認識することから始まった。

自然なことに、形式化された理論 $T$ の意味論そのものを別の形式化された理論 $T_0$ と見なしたり、あるいは、カテゴリーの研究を形式化しようとする試みのように、数学の概念的な側面を扱うこともできる。

一方、「基礎論」は数学の形式的側面を概念化し、ブール代数、cylindric代数、polyadic代数、そして以下に述べるような構造を生み出す。

本論文の目的の一つは、カテゴリー論の概念的領域で最初に分離され命名されたものだが、論理学にも浸透しているように思われる **adjointness** の概念が普遍的なものであることの証拠を与えることである。

具体的には、第III節において、関数性の理論と命題論理の間の既知の類似性(証明論で利用されることもある)を明示するのに適切な抽象構造であると思われる cartesian closed category の概念について述べる。

cartesian closed categoryの構造はすべてadjointnessによって与えられ、「メタ理論(hyperdoctrine)」の構造も同様である。通常、論理学や証明論の領域とはかけ離れていると考えられている数学の領域にも、正確に類似した「量化子(quantifiers)」が存在する。

我々が指摘したように、(少なくとも自然数上の)recursionも、全面的に、適切なadjointによって特徴づけられる。

したがって、それは、大まかに言って、直観主義的な高階の整数論の証明理論を与えることが可能である。そして、その理論の中では、(論理的または数学的に)すべての重要な公理が、adjointnessの概念のインスタンスとして表現される。

「概念」、「形式」、「基礎」という上述の概念は、本稿では数学的な役割を果たしていない。これらは、カテゴリー論全般、特に本稿と、普遍的な傾向を持つ他の研究との関係を見るための、一つの可能な視点を提供するために、この序章に含まれたに過ぎない。

しかし、これらの概念を真摯に受け止めようとするならば、「基礎」を形式化する試みの本質的な特徴は、形式的なものとの間の「双対性」の記述であろう。

後者の点については、第IV節の最後は、第III節とは異なるが、同様に普遍的な意味を持つと思われる一群のadjointの状況の議論に費やされている。これらは、ある種のグローバル化された Galois connectionとして短く記述されるだろう。

# 第四節：代数幾何と数学の基礎における 大域化されたGalois Connection

$$\text{Theories}^{\text{op}} \begin{array}{c} \xleftarrow{\text{semantics}} \\ \xrightarrow{\text{structure}} \end{array} (\text{Cat}, [\text{Sets}^I])$$

in which the right hand side denotes a category whose morphisms are commutative triangles

$$\begin{array}{ccc} \mathbf{C} & \xrightarrow{\quad} & \mathbf{C}' \\ & \searrow & \swarrow \\ & \text{Sets}^I & \end{array}$$

of functors with  $\mathbf{C}$  and  $\mathbf{C}'$  more or less arbitrary categories. The invariant notion of theory here appropriate has, in all cases considered by the author, been expressed most naturally by identifying a theory  $T$  itself with a category of a certain sort, in which case the semantics (categories of models) of  $T$  is a certain subcategory of the category of functors  $T \longrightarrow \text{Sets}^I$ . There is then a further adjoint situation

$$\text{Formal} \begin{array}{c} \xleftarrow{\quad} \\ \xrightarrow{\quad} \end{array} \text{Theories}$$

describing the presentation of the invariant theories by means of the formalized languages appropriate to the species. Composing these two adjoint situations, and tentatively identifying the Conceptual with categories of the general sort  $(\text{Cat}, [\text{Sets}^I])$ , we arrive at a family of adjoint situations

$$\text{Formal}^{\text{op}} \begin{array}{c} \xleftarrow{\quad} \\ \xrightarrow{\quad} \end{array} \text{Conceptual}$$

(one for each species of theory) which one may reasonably hope constitute the fragments of a precise description of the duality with which we began our discussion.

# この章のポイント profunctor vs. 行列

profunctorの場合

$$F^*F_*: \text{Set}^D \rightarrow \text{Set}^D$$
$$F_*F^*: \text{Set}^C \rightarrow \text{Set}^C$$



行列の場合

$$MM^*: k^X \rightarrow k^X$$
$$M^*M: k^Y \rightarrow k^Y$$

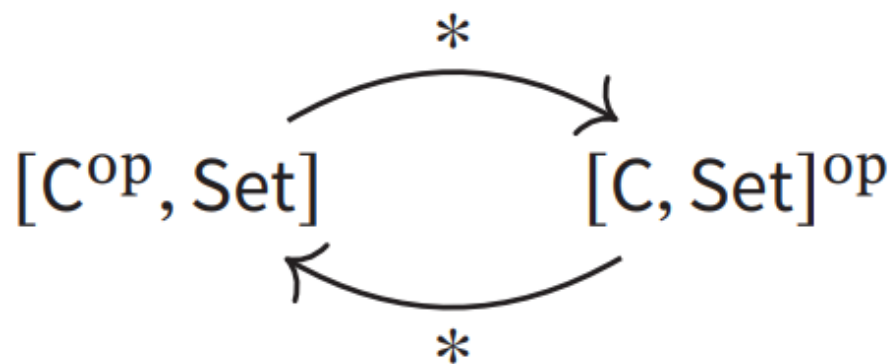
$$F^*c_i \cong d_i \text{ かつ } F_*d_i \cong c_i$$



$$M^*u_j = \sigma_j v_j \text{ かつ } Mv_j = \sigma_j u_j$$

( 行列のSVDの場合には、次の式だった。 )

Isbell conjugate の二つのバージョンは、次の図のように、行ったり戻ったりする二つのfunctor を与えるのである。



$$(\mathcal{O} \dashv \text{Spec}): \text{CoPresheaves} \overset{\mathcal{O}}{\underset{\text{Spec}}{\rightleftarrows}} \text{Presheaves}$$





# Part 4

## ニューラル言語モデル批判

言語の意味の数学的構造

## Part 4 Agenda

### ニューラル言語モデル批判

- Tai-Danaeの言語思想とニューラル言語モデル批判
- 現実の経験的データからの構造の抽出の試み
- 語の構造の抽出
- 数学的展望



# Tai-Danaeの言語思想 ニューラル言語モデル批判

1 マルレク 言語の意味の数学的構造

# Tai-Danaeの言語思想

ここでは、この論文の後半に集中して記述されている彼女の言語思想を、先に紹介しようと思います。

それは、言語思想の大きな流れの中で、現在のニューラル言語モデルの批判を、意図したものです。

彼女のそうした意図を理解することが、彼女の提起している数学的アプローチの意味をよりよく理解することになると考えています。

# ニューラル言語モデルの高い性能

ニューラル技術が言語に対するより原理的なアプローチよりも優れている点があるとするれば、それは経験的な言語データに対して驚くほど高い性能を発揮できることである。

今後、どのような形式言語モデルがくるにせよ、現実世界におけるその品質と妥当性を判断することが、決定的な意味を持つことは間違いない。

# テキストデータに内在する意味の構造

考えてみれば、文字列としてのコーパスはsyntacticsそのものであるから、言語のsyntacticalな特徴をテキスト・コーパスから抽出できることは驚くべきことではないかもしれない。

しかし、より広範で哲学的な疑問は残る。

もしそうだとしたら、意味の重要な側面が純粹な形式から生まれるということはあるのだろうか？

最近のLLMの進歩に伴い、この問題はますます重要になってきている。

この論文は、この問題について実質的な哲学的な扱いを提供する場ではないが、ここで紹介する数学が、意味はテキストデータに内在する複数の形式的次元と不可分であるという考えを支持していることを指摘しておくことは重要だと思われる。

# 意味と形式は分離できない

意味と形式は切り離せないという考え方は新しいものではないが、現在のAIをめぐる哲学的な議論には浸透していない。

厳密な哲学的見地から言えば、カントとヘーゲルの影響力のある著作は、形式と内容は排他的なものではないという原則に立っており、この考え方は、分析哲学の父であるフレーゲの思想の中核にも見出すことができる。

# ソシュールの構造主義言語学

さらに重要なことに、形式と意味は独立していないという視点は、フェルディナン・ド・ソシュールの研究と、近代言語学の出現の動機となった構造主義革命によって、言語学の中心となった。

重要な論点は、形式と意味、シニフィアンとシニフィエの両方が、共通の構造的特徴によって同時に決定されるということである。

20世紀半ば、言語形式と意味は一枚の紙の裏表のように密接に関係しているというソシュールの考え方が言語学の分野で支配的であった。

# チョムスキーの生成言語学

1950年代後半にチョムスキーが革新的な生成言語学を導入したことで、構造主義的なプログラムは劇的に減速したのだが、**チョムスキーのプログラムのもとでも、構造的な特徴が言語の性質を定義するものとしてとらえられ続けられた。**

# 言語学での経験的アプローチの復活

20世紀末の言語学への経験的アプローチの復活は、この進化に新たな変化をもたらした。

コネクショニズム、コーパス言語学、潜在的意味解析、その他の言語学習可能性へのアプローチは、経験的データからあらゆる種類の構造的特性を引き出そうとする新たな努力を象徴しており、意味論と統語論を結びつける無数の概念的・技術的手段を提供している。

言語の形式にしかアクセスできない言語モデルにおいて意味論が問題となることは驚きかもしれないが、ここで重要なのは、形式から意味が生まれるという理論は、言語思想の広範かつ確立された伝統の一部であるということである。

# ニューラル言語モデルの不十分さ

そして、このような伝統が、特にその構造主義的バージョンにおいて私たちに語っているのは、構文対象の分析において意味が問題になるとすれば、それはすべて言語の形式に反映された構造的特徴に起因するということである。

しかし、現在のニューラル言語モデルが不十分なのは、まさにこの点である。というのも、ニューラル言語モデルは、そのタスクを実行する際に必然的に働く構造的特徴を明らかにしていないからである。



# 現実の経験的データからの 構造の抽出の試み

1 マルレク 言語の意味の数学的構造

# 現実の経験的データから構造を抽出する

このセッションでは、線形代数とenrichedカテゴリー理論のツールが実際にどのように機能するかを説明し、結論では、ここで紹介したenrichedカテゴリー理論の経験的能力が、より多くのことを行うためにどのように使用できるかを共有する。

始めに、2022年3月時点の英語版ウィキペディアの全記事からなる英語版ウィキペディア・コーパスを考えてみよう。

このコーパスを言語構造を仮定しない純粋な構文的対象として考えると、コーパスは有限個の独立したトークンまたは文字の長いシーケンスとして現れる。

物事を単純化するために、そのコーパスの中で最も出現頻度の高い40文字(句読点を除く)に限定してみよう。つまり、初期集合  $X$  は以下の要素を含む:

$$X = \{-, /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, =, a, b, c, d, e, f, \\ g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \acute{e}\}.$$

ここで、 $Y = X \times X$  とし、これまで見てきたのと同様に、 $X, Y$  要素間の関連性を言語学的に表す **行列  $m : X \times Y \rightarrow \mathbb{R}$**  を考える。

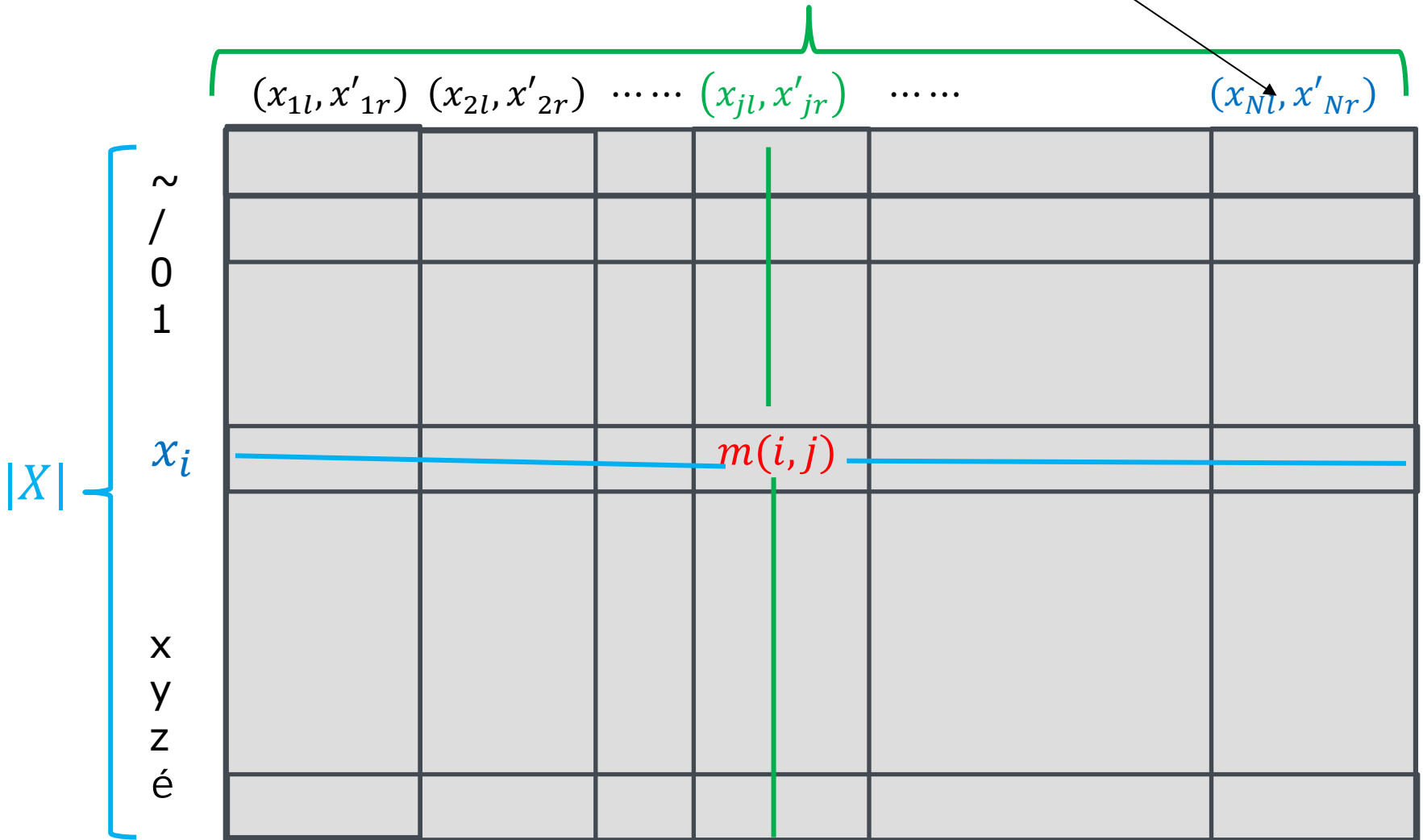
行列  $m$  についての単純な選択は、文字  $x \in X$  が与えられた時、 $x$  の左に  $y_l$ 、右に文字  $y_r$  が現れることを示す  $(y_l, y_r) \in Y$  の経験的確率を行列の値として考えることである。

例えば、 $m(h, (t, e)) \approx 0.3836$  であるのに対して、 $m(h, (r, o)) \approx 0.3836$  であり、 $h$  を中心文字とすると、 $rho$  よりも  $the$  の方が、100 倍以上確率が高いことがわかる。

行列  $m : X \times Y \rightarrow \mathbb{R}$

$X = \{-, /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, =, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \acute{e}\}$ .

$$|Y| = |X \times X| = |X| \times |X| (= N)$$



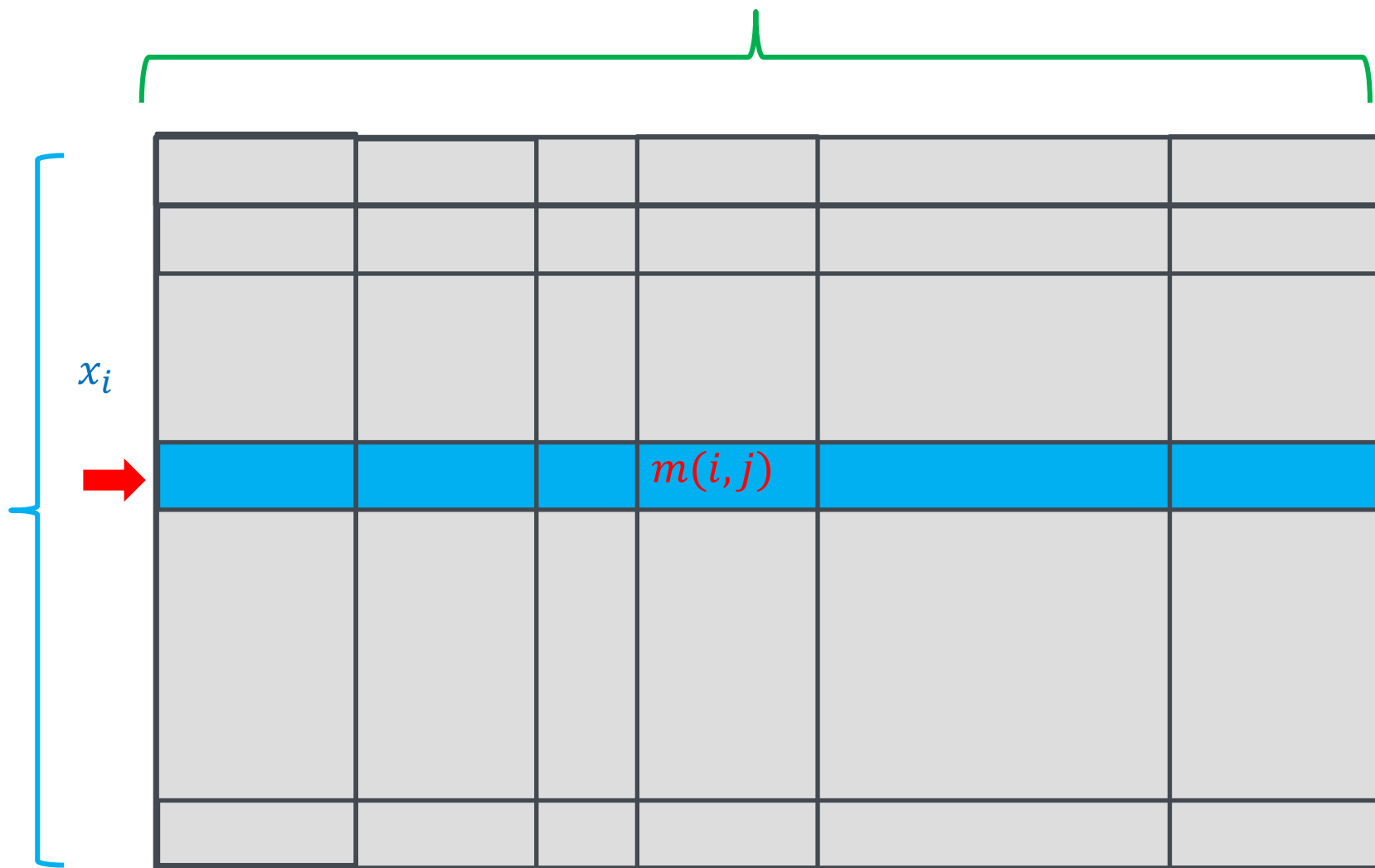
$X$ の要素として考えると、各文字は独立しており、他のすべての文字とは異なる。しかし、 $X \rightarrow \mathbb{R}^Y$ を行列 $m$ を介して埋め込み、コーパスに反映された具体的な言語運用の中でそれらが示す関係を活用することで、構造的な特徴が明らかになる。

実際、誘導された演算子 $M^*: \mathbb{R}^X \rightarrow \mathbb{R}^Y$ に対してSVDを行えば、各文字のベクトル表現を得ることができる。

(本論文の範囲外の理由から、SVDを行う前にエンタリーの平方根をとり、行列を0を中心に置くと便利である。)

行列  $M^* : \mathbb{R}^X \rightarrow \mathbb{R}^Y$

Xのベクトルから  
Yのベクトルへの写像



# Low-rank matrix approximation

行列  $\sigma'$

$|D|$ 行,  $d$ 列

行列  $\sigma$

$d$ 行,  $|D|$ 列

行列  $\sigma'\sigma$

$|D|$ 行,  $|D|$ 列

×

行列  $\sigma$

=

行列  $\sigma'$

行列  $\sigma'\sigma$

行列  $M$ と  
同じ大きさ

$\|M - \sigma'\sigma\|$ が最小となるような、 $\sigma'$ と $\sigma$ を見つける。

# 低いランクの行列での近似

実用的なアプリケーションの中には、行列 $M$ を、特定のランク $r$ を持つ別の行列 $\tilde{M}$ で近似する問題を解く必要があるものもある。

近似が  $\text{rank}(\tilde{M}) = r$  という制約のもとで、 $M$  と  $\tilde{M}$  の差のフロベニウス・ノルムの最小化に基づく場合解は  $M$  の SVD で与えられることがわかる。

すなわち

$$\tilde{M} = U\tilde{\Sigma}V^*$$

ここで、 $\tilde{\Sigma}$ は $\Sigma$ と同じ行列であるが、最大の $r$ 個の特異値のみを含む（他の特異値は0に置き換えられる）。

# truncated SVD

多くのアプリケーションでは、非ゼロ特異値の数 $r$ が大きいため、コンパクトSVDでさえ計算が非現実的なものになる。

このような場合、最小の特異値を切り捨てて、 $t \ll r$ 個の非ゼロ特異値のみを計算する必要があるかもしれない。

切り捨てSVDは、もはや元の行列 $M$ の厳密な分解ではないのだが、固定ランク $t$ の任意の行列による**最適な低ランク行列近似 $M$ を提供する。**

$$D \hookrightarrow \mathbb{R}^D \rightarrow \mathbb{R}^d$$

$$D \hookrightarrow \mathbb{R}^D$$

aardvark  $\mapsto (1, 0, 0, 0, \dots, 0, 0)$

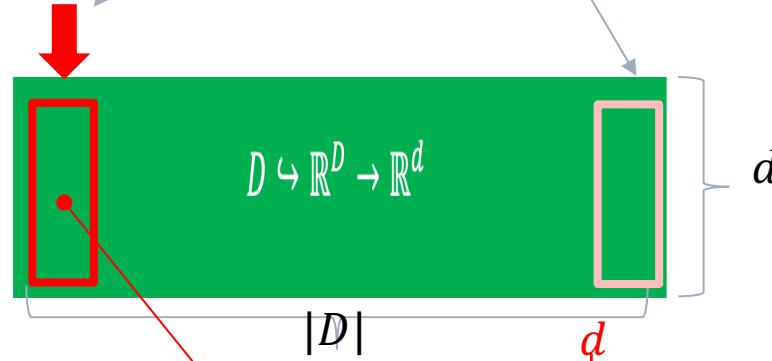
aardwolf  $\mapsto (0, 1, 0, 0, \dots, 0, 0)$

$\vdots$

zyzzyva  $\mapsto (0, 0, 0, 0, \dots, 0, 1)$ .

$\mathbb{R}^D$

$$\sigma: \mathbb{R}^D \rightarrow \mathbb{R}^d$$



aardvark  $\mapsto (0.632, 0.370, -0.620, \dots, -0.475)$

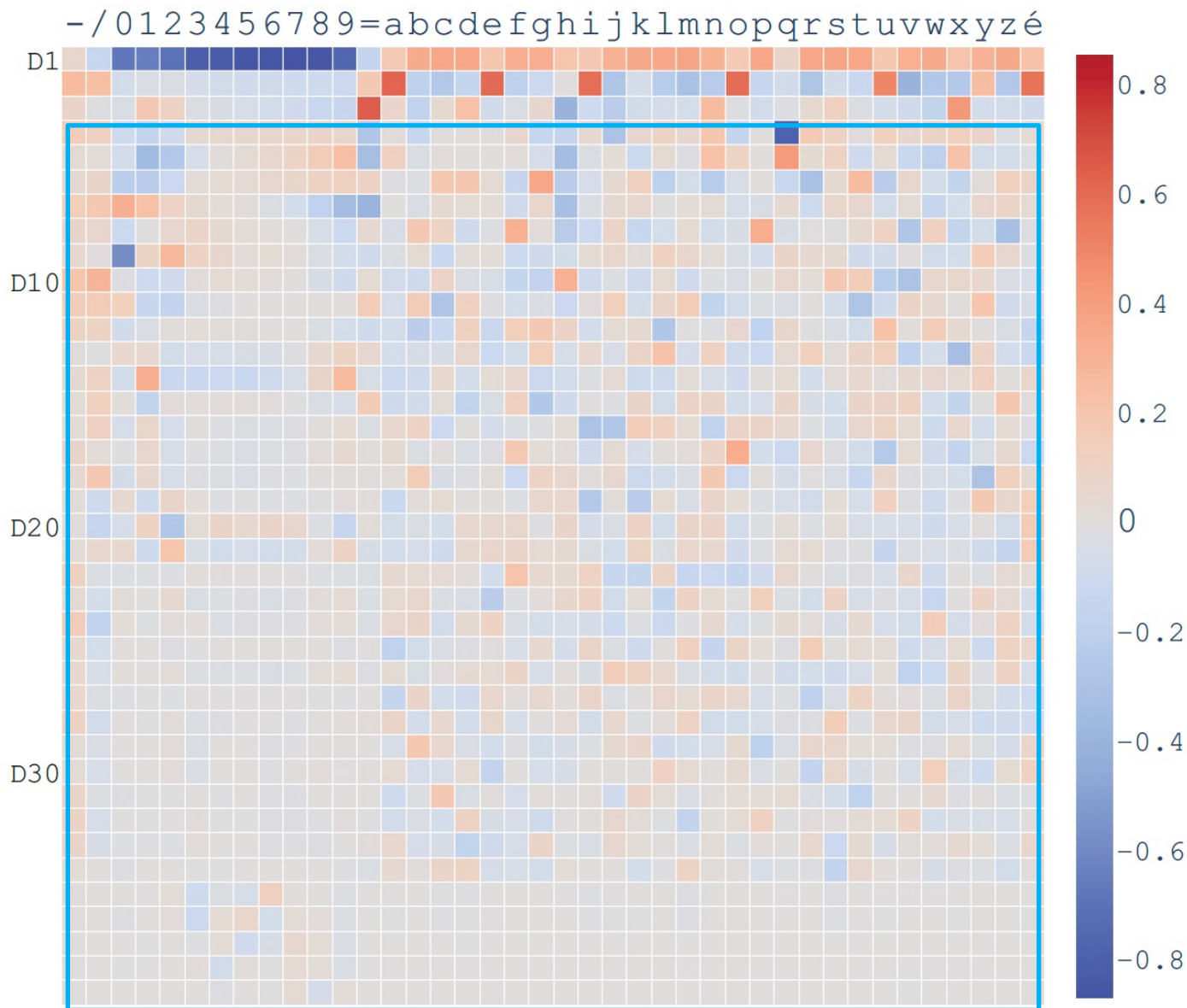
誘導された演算子  $M^*: \mathbb{R}^X \rightarrow \mathbb{R}^Y$  に対してSVDを行えば、各文字のベクトル表現を得ることができる。

上位三つの特異ベクトル  $D_1, D_2, D_3$  について、truncated SVD を実行する。

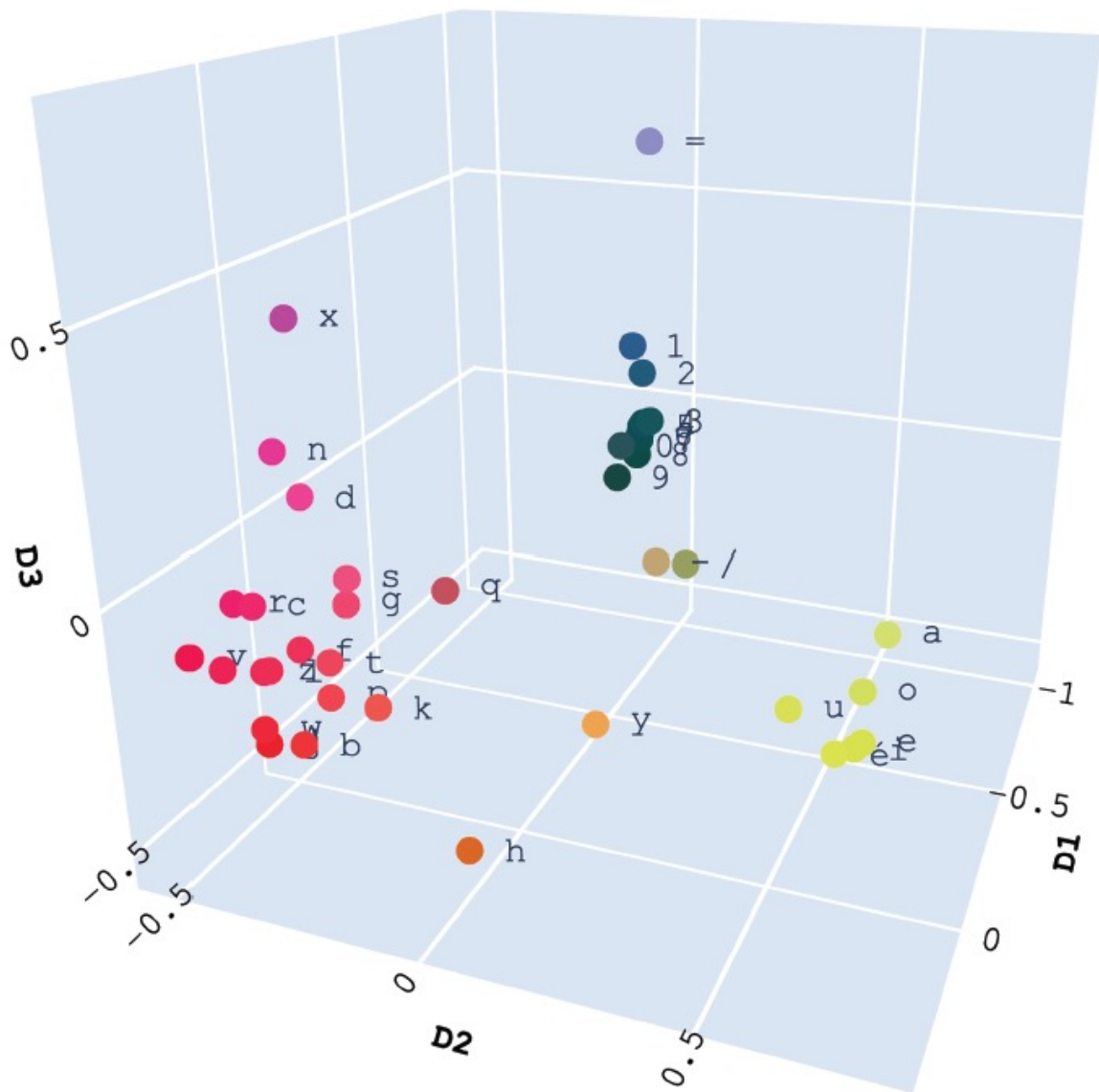


最初の3つの特異ベクトルを見ると、  
1つ目の特異ベクトルは数字と文字を識別し、  
2つ目の特異ベクトルは母音とそれ以外を識別し、  
3つ目の特異ベクトルは特殊文字を識別することがわかる。

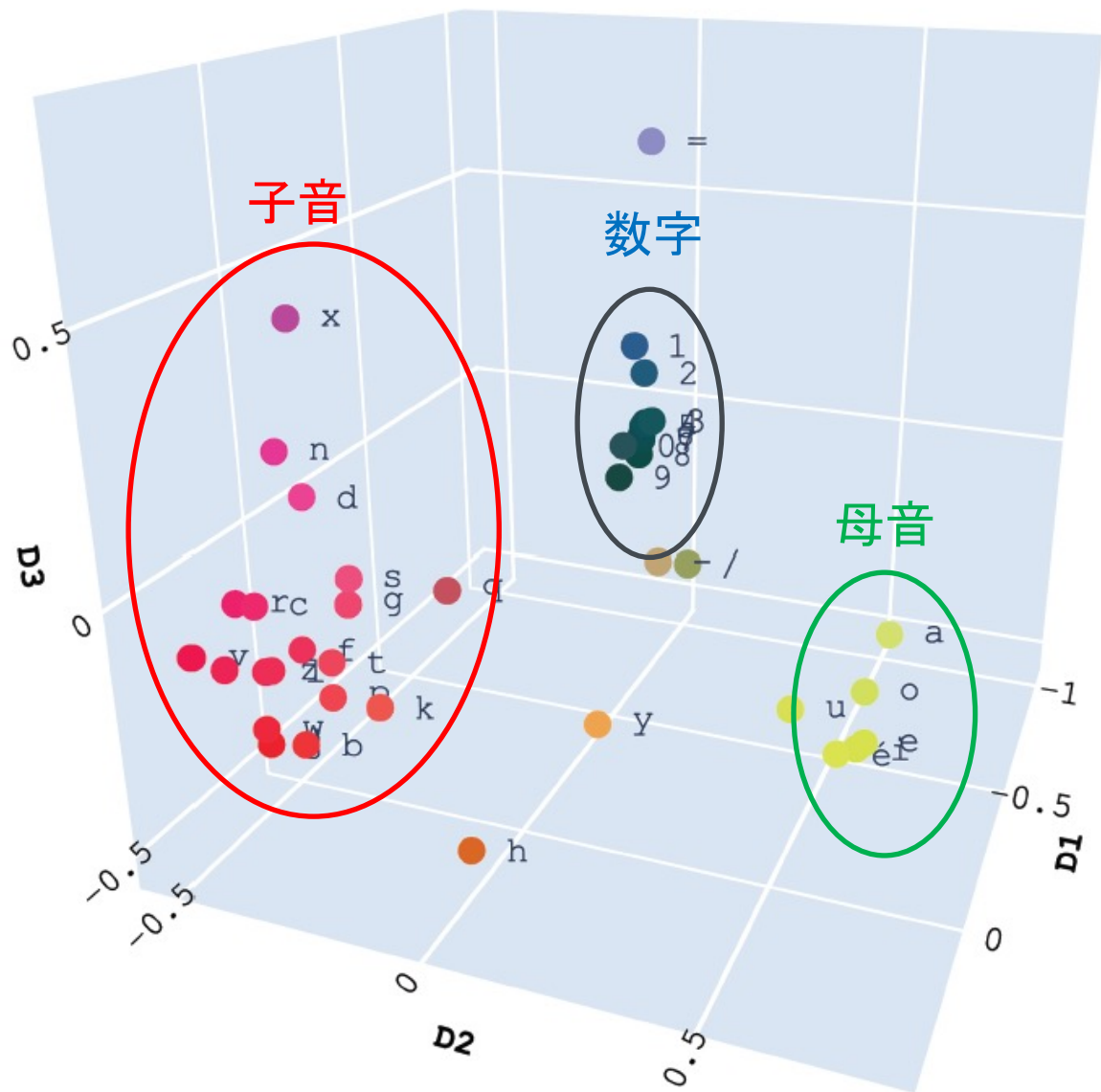
連続する特異値の急激な減衰は、3次元以上の次元では  
わずかな区別しか追加されないことを示している



図は、 $X$ のすべての文字を3次元空間の点としてプロットしたもので、座標は3つの最大の特異値に対応する特異ベクトルで与えられ、その特異値でスケールされている。つまり、要素は母音、子音、数字、特殊文字として区別される。



図は、 $X$ のすべての文字を3次元空間の点としてプロットしたもので、座標は3つの最大の特異値に対応する特異ベクトルで与えられ、その特異値でスケールされている。つまり、要素は母音、子音、数字、特殊文字として区別される。



# Formal Concept Analysisの応用

特異値とベクトルへの分解は重要な構造的特徴を明らかにした。  
各特異ベクトルは合理的な方法で要素を識別し、対応する特異値は要素を異なるタイプにクラスタリングする働きをする。

ただ、この狭い文脈での線形代数 SVDの応用は、行き詰まるように思われる。

しかし、前節で議論したように、集合  $X$  と  $Y$  をカテゴリーまたは enriched カテゴリーとみなすことで、さらに異なる構造的洞察を得ることができる。

## Cutoff値の設定で、 $\{0, 1\}$ に値を取る行列を作る

プリミティブな例では、 $X$ と $Y$ を二つのオブジェクトからなるカテゴリ-2 =  $\{0, 1\}$ 上でenrich化された離散カテゴリとみなしてきた。

そのためには、次のステップでは、 $\{0, 1\}$ に値を取る行列  $r: X \times Y \rightarrow \{0, 1\}$ が必要である。

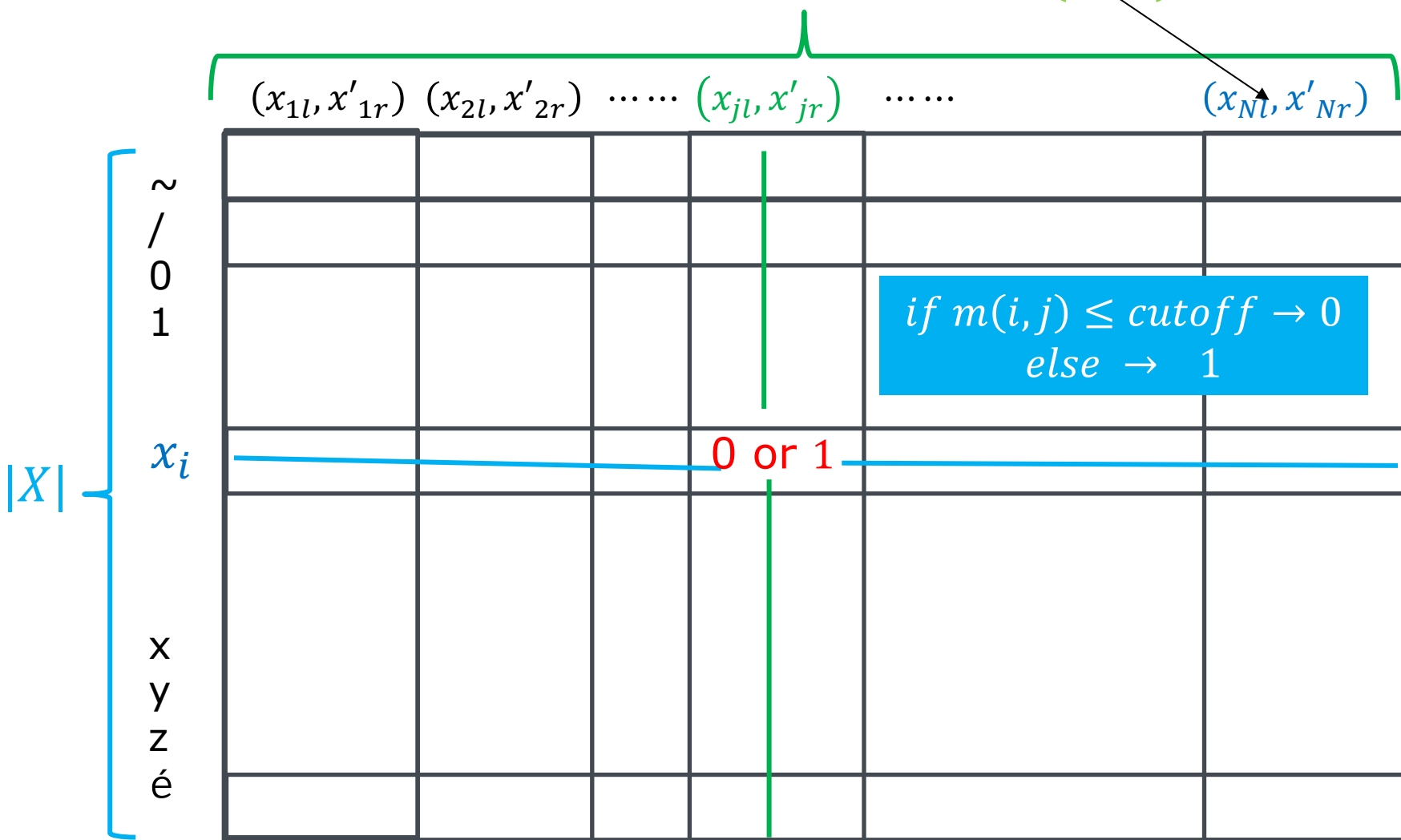
単純で地味な選択としては、cutoff 値(0.001など)を設定し、先のSVDで使用した行列 $M$ を $\{0, 1\}$ に値を取るの行列に変更することである。

カットオフ以下のエントリーはすべて0に置き換えられ、カットオフ以上のエントリーはすべて1に置き換えられる。

行列  $r: X \times Y \rightarrow \{0,1\}$

$X = \{-, /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, =, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \acute{e}\}$ .

$$|Y| = |X \times X| = |X| \times |X| (= N)$$



次に、それを拡張し関数  $R^*: 2^X \rightarrow 2^Y$  と  $R_*: 2^Y \rightarrow 2^X$  を得て、 $R_*R^*$  と  $R^*R_*R^*$  の固定オブジェクトを見る。

それは、先に述べたように、高度に構造化された**完全束 complete lattice** を構成する formal concept  $\{(A_i, B_i)\}$  に編成される。

束(lattice)の全体を視覚化することは、静的な二次元画像では難しい。

このページでは、1つの文字を選択し、 $A$ が選択した文字を含む concept  $\{(A,B)\}$  だけを見ることによって定義された部分束を見ることができる。図4では、 $a$ と3の文字が選択され、さらに画像の複雑さを減らすために、 $|B| \geq 20$ という多数のコンテキストを表すノードだけが描かれている。

{a,e,i,é}

{a,e,o,é}

{a,e,i,o}

{g\_n,m\_n,  
m\_t,t\_l,...}

{g\_n,m\_n,  
m\_r,m\_t,...}

{g\_n,m\_n,  
m\_s,m\_t,...}

{a,u,é}

{a,e,u}

{a,i,u}

{a,o,u}

{g\_l,m\_l,  
m\_n,p\_t,...}

{d\_l,g\_n,  
m\_n,m\_s,...}

{m\_l,m\_n, {d\_n,g\_n,  
m\_s,p\_t,...}m\_n,m\_s,...}

{a,y}

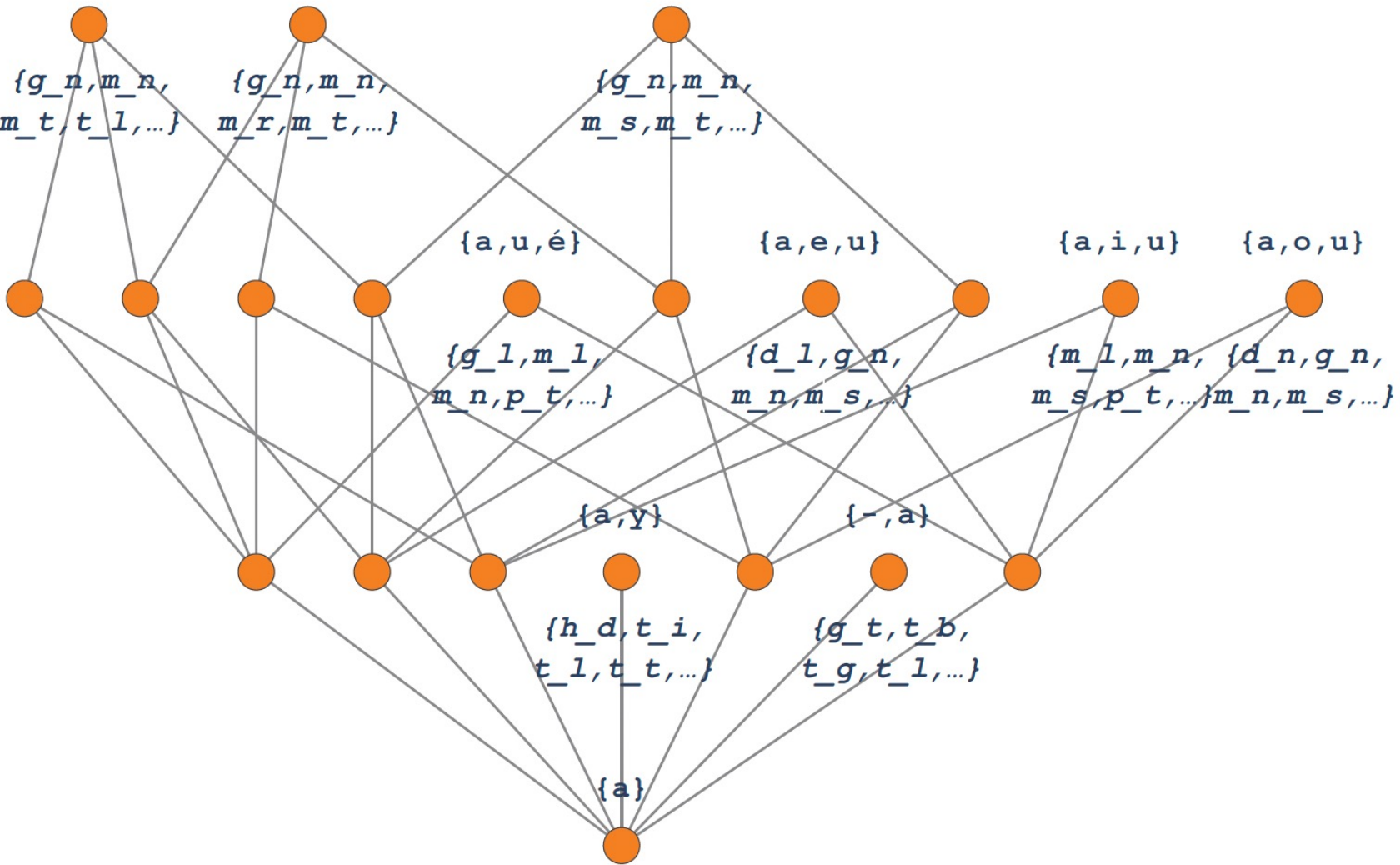
{-,a}

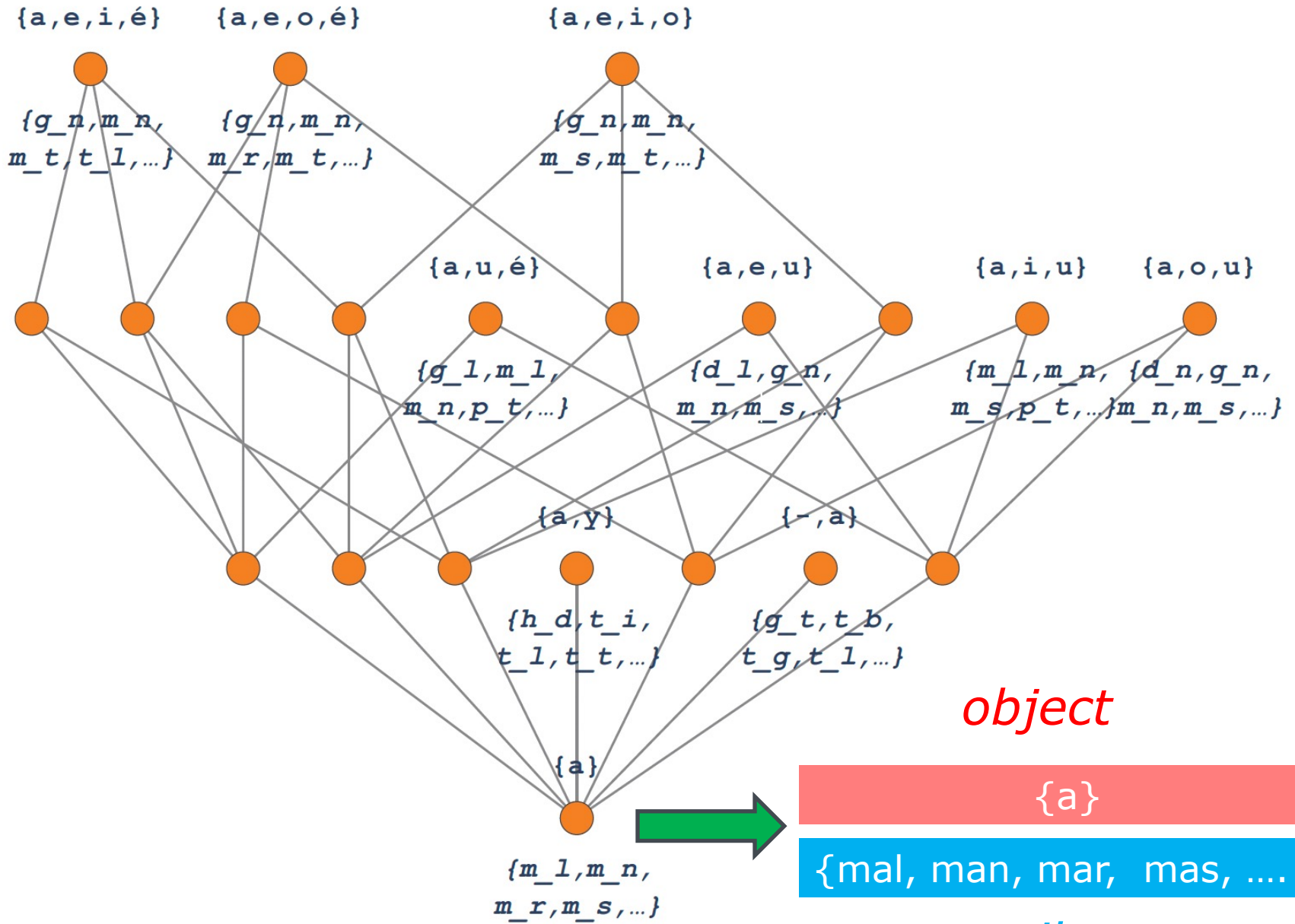
{h\_d,t\_i,  
t\_l,t\_t,...}

{g\_t,t\_b,  
t\_g,t\_l,...}

{a}

{m\_l,m\_n,  
m\_r,m\_s,...}





*object*



*attribute*



束はすぐに、数字と文字の区別を明確にするが、それだけではない。

各文字の集合 $A_i$ は、明示的なコンテキストのdualな集合 $B_i$ に関連付けられ、構成性の原理を示唆する。すなわち、対応するクラスの要素は、コーパスに属するシーケンスを生成するために自由に構成することができる。

このような構成は、言語学的観点から関連する特徴を明らかにする。数字が他の数字や特殊文字と構成される傾向があるのに対し、母音は子音と構成されることが多い。

厳密に言えば、SVD 解析でも同様の二重性が存在する。

なぜなら、左特異ベクトルは右特異ベクトルと(一般的に)一対一で対になっており、Yの中のコンテクストをそのコンテクストとなる文字に基づいて区別するからである。

一方、formal conceptは  $R_*R^*$  と  $R^*R_*$  の固定オブジェクトとして、大きな、しかし離散的な文字のクラス間の二元性を示す。

数字、子音、文字はすべて区別されるが、より細かい区別もなされる。さらに、このような双対クラスの集合は単なる独立した要素の集合ではなく、前述の演算  $\vee$  と  $\wedge$  を持ち、概念レベルで代数演算を行うことができる。

# 語の構造の抽出

1 マルレク 言語の意味の数学的構造

## 形式から生まれる内容

考えてみれば、文字列としてのコーパスはsyntacticsそのものであるから、言語のsyntacticalな特徴をテキスト・コーパスから抽出できることは驚くべきことではないかもしれない。

結局のところ、言語学的な観点からは、子音、母音、および数字は、それ自体に意味を持たない、純粹にsyntacticalな単位なのである。

しかし、文字について言えることは、より高いレベルの言語単位についても言えるのである。

このことを、 $X$ をBritish National Corpus の1,000語の最頻単語として、示すことができる。

このコーパスに含まれる単語 $x$ の左に現れる語を $y_l$ 、右に現れる語を $y_r$ とした時、その語の出現の経験的確率を使って、 $XY$ 行列 $M$ を作り、前と同じ計算を繰り返す。

10個の最大の特異値に対応する  $M$ の特異ベクトルは、名詞、動詞(過去と現在)、形容詞、副詞、場所、量詞、数字、国など、単語のあらゆる構文的・意味的特徴を捉えている。

これらの10個の特異ベクトルは、図5で降順に描かれている。読みやすくするために、1000個のエントリのうち8個、つまり最大4個と最小4個だけが表示されている。

図5. BNCコーパスに含まれる単語で, 最初の10個の特異ベクトルについて, 大きい順に4つ, 小さい順に4つの値に対応するもの.

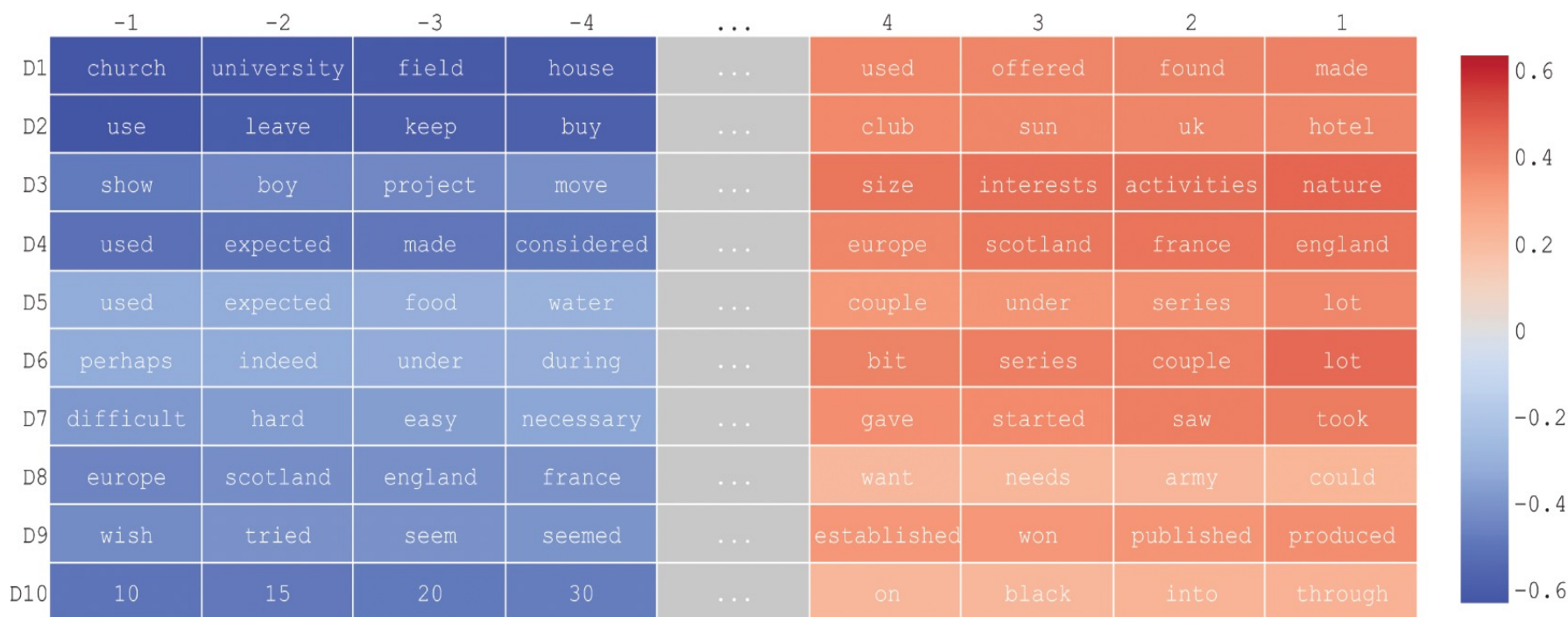


Figure 5. Words in the BNC corpus corresponding to the four greatest and four least values for the first 10 singular vectors in decreasing order.

	-1	-2	-3	-4	.
D1	church	university	field	house	.
D2	use	leave	keep	buy	.
D3	show	boy	project	move	.
D4	used	expected	made	considered	.
D5	used	expected	food	water	.
D6	perhaps	indeed	under	during	.
D7	difficult	hard	easy	necessary	.
D8	europe	scotland	england	france	.
D9	wish	tried	seem	seemed	.
D10	10	15	20	30	.

	-1	-2	-3	-4	.
D1	church	university	field	house	名詞
D2	use	leave	keep	buy	動詞現在
D3	show	boy ?	project	move	動詞現在
D4	used	expected	made	considered	動詞過去
D5	used	expected	food ?	water ?	動詞過去
D6	perhaps	indeed	under	during	副詞
D7	difficult	hard	easy	necessary	形容詞
D8	europe	scotland	england	france	場所
D9	wish	tried	seem	seemed	動詞?
D10	10	15	20	30	數字

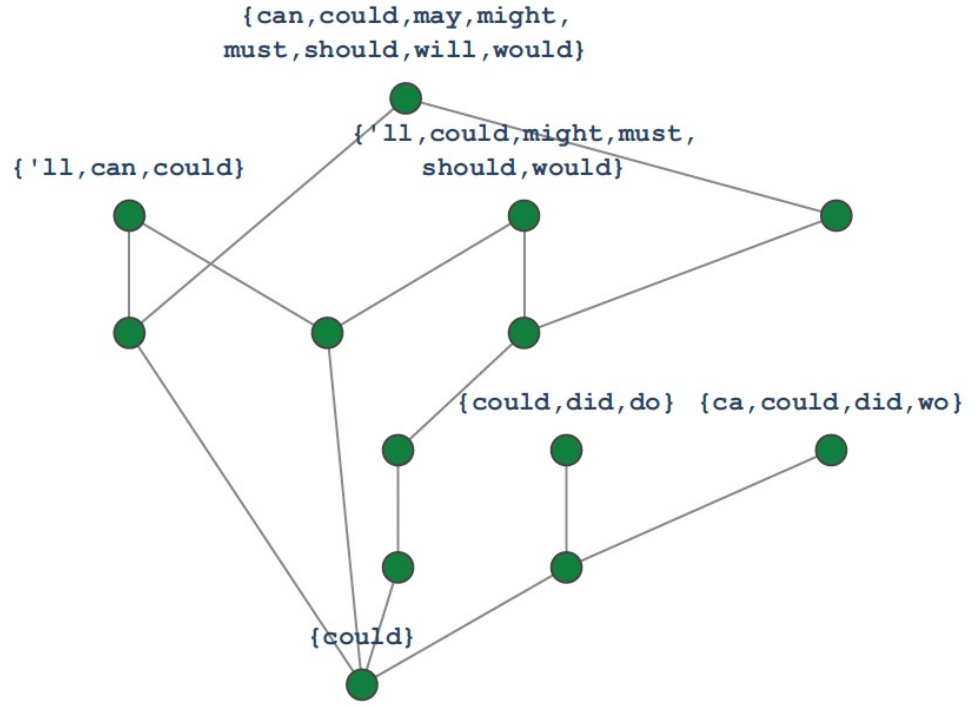
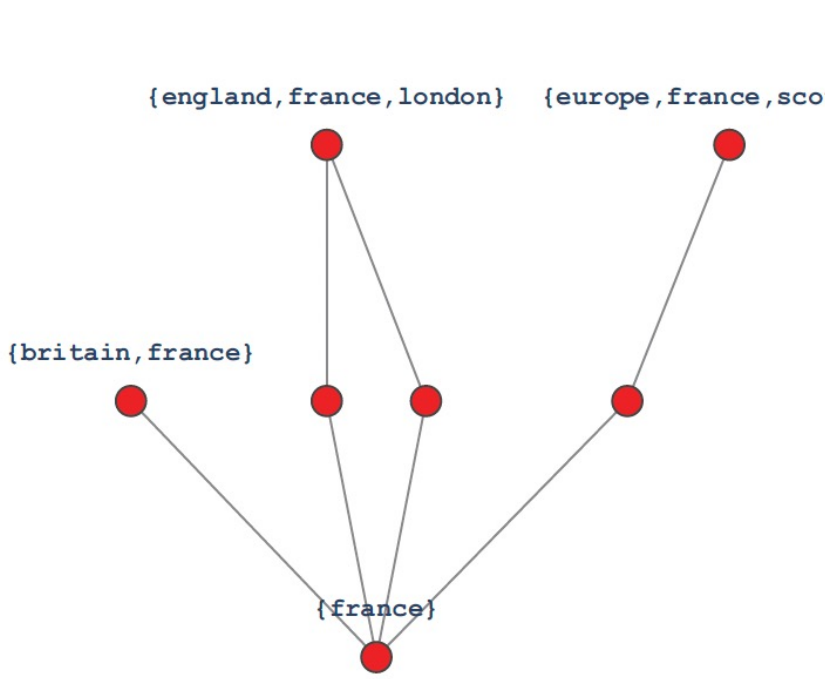
	4	3	2	1
	used	offered	found	made
	club	sun	uk	hotel
	size	interests	activities	nature
	europe	scotland	france	england
	couple	under	series	lot
	bit	series	couple	lot
	gave	started	saw	took
	want	needs	army	could
	established	won	published	produced
	on	black	into	through

# Formal Concept Analysis

文字行列の場合と同じように、ブール行列 $M$ を作成するためにカットオフ(ここでは0.01)を選択することで、特異ベクトルに現れる語に関するさらなる情報を得ることができ、これらの10,00語についてformal conceptのlatticeを抽出することができる。

説明のために、図5に示した最も重要な特異ベクトルのエントリからいくつかの単語(france, could, 10)を選択し、対応するformal conceptの部分latticeを図6に示す。

線形代数はこれらの単語が重要であることを強調し、それらをクラスタリングする。カットオフを選択してformal conceptに渡すと、これらの単語が属する構文クラスと意味クラスが明らかになり、興味深い、より洗練された構造的特徴が現れる。



しかし、より広範で哲学的な疑問は残る。

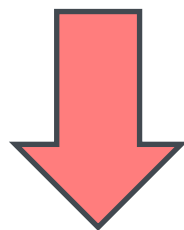
もしそうだとしたら、意味の重要な側面が純粋な形から生まれるということはあるのだろうか？

最近のLLMの進歩に伴い、この問題はますます重要になってきている。

よく繰り返される考え方のひとつに、純粋な言語形式(つまり生のテキスト)しか利用できない言語モデルは、意味とは何の関係も持たないし、持つこともできないというものがある。

この考え方は、意味を「言語形式と伝達意図の関係」と理解することに基づいている。

この論文は、この問題について実質的な哲学的な扱いを提供する場ではないが、ここで紹介する数学が、意味はテキストデータに内在する複数の形式的次元と不可分であるという考えを支持していることを指摘しておくことは重要だと思われる。



前々回のセッション

「Tai-Danaeの言語思想とニューラル言語モデル批判」  
に続く

# 参考資料

## 語の意味

2023/06/30 マルレク「プロンプトで遊ぶ-- GPT-4 との対話 --」  
での「辞書」を作る実験

<https://www.marulabo.net/docs/prompt/>

GPT-4は、「語の意味」について、  
どのような情報を持っているか

# GPT-4は、「語の意味」について、 どのような情報を持っているか

ここでは、GPT-4が「語の意味」について、どのような情報を持っているのかを探る。

具体的には、ある語を与えた時に、**ミニWordNet-like な辞書項目を作成する**プロンプトを作成する。

WordNetが持つ情報を、基本的にはGPT-4も持っていることを示すことができると思う。

# WordNetとはなにか？

WordNetは、Princeton大学が公開している、次のような特徴を持つ英語の包括的なネットワーク上の語彙データベースである。

<https://wordnet.princeton.edu/>

「WordNetは、英語の大規模な語彙データベースである。名詞、動詞、形容詞、副詞は、それぞれ異なる概念を表す認知的同義語の集合(synset)にグループ化されている。synsetは、概念的意味的關係と統語論的關係によって相互にリンクされる。その結果、意味的に関連する単語と概念のネットワークは、ブラウザでナビゲートすることができる。」

# WordNetとはなにか？

「WordNetは、単語の意味に基づいてグループ化するという点で、表面的にはシソーラスに似ている。しかし、いくつかの重要な相違点がある。

第一に、WordNetは、単語の形(文字列)だけでなく、単語の特定の意味もリンクしている。その結果、ネットワーク上で互いに近接する単語は、意味的に区別されるようになる。

第二に、WordNetは単語間の意味的関係をラベル付けしている。一方、シソーラスにおける単語のグループ化は、意味の類似性以外の明確なパターンに従っていない。」

今回作成したプロンプトの

「上位概念」はWordNetのhypernymで、  
「下位概念」はWordNetのhyponymである。

# WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations  
Display options for sense: (gloss) "an example sentence"

## Noun

- **S: (n) money** (the most common medium of exchange; functions as legal tender) *"we tried to collect the money he owed us"*
  - [direct hyponym](#) / [full hyponym](#)
  - [direct hypernym](#) / [inherited hypernym](#) / [sister term](#)
  - [part holonym](#)
  - [derivationally related form](#)
- **S: (n) money** (wealth reckoned in terms of money) *"all his money is in real estate"*
- **S: (n) money** (the official currency issued by a government or national bank) *"he changed his money into francs"*

## Noun

- S: (n) **money** (the most common medium of exchange; functions as legal tender) *"we tried to collect the money he owed us"*
  - direct hyponym / full hyponym
    - S: (n) appropriation (money set aside (as by a legislature) for a specific purpose)
    - S: (n) fund, monetary fund (a reserve of money set aside for some purpose)
    - S: (n) boodle, bread, cabbage, clams, dinero, dough, gelt, kale, lettuce, lolly, lucre, loot, moolah, pelf, scratch, shekels, simoleons, sugar, wampum (informal terms for money)
    - S: (n) shinplaster (paper money of little value issued on insufficient security)
    - S: (n) subsidization, subsidisation (money (or other benefits) obtained as a subsidy)
    - S: (n) token money (coins of regular issue whose face value is greater than their intrinsic value)
  - direct hypernym / inherited hypernym / sister term
  - part holonym
  - derivationally related form

- **S: (n) money** (the most common medium of exchange; functions as legal tender) *"we tried to collect the money he owed us"*
  - direct hyponym / full hyponym
  - direct hypernym / **inherited hypernym** / sister term
    - **S: (n) medium of exchange, monetary system** (anything that is generally accepted as a standard of value and a measure of wealth in a particular country or region)
      - **S: (n) standard, criterion, measure, touchstone** (a basis for comparison; a reference point against which other things can be evaluated) *"the schools comply with federal standards"; "they set the measure for all subsequent work"*
        - **S: (n) system of measurement, metric** (a system of related measures that facilitates the quantification of some particular characteristic)
          - **S: (n) measure, quantity, amount** (how much there is or how many there are of something that you can quantify)
            - **S: (n) abstraction, abstract entity** (a general concept formed by extracting common features from specific examples)
              - **S: (n) entity** (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

語から辞書項目を作るプロンプトを作る

# 語から辞書項目を作るプロンプトの構成

- [語の辞書項目]
  - [意味]
  - [Semantic Category]
    - [同義語]
    - [類義語]
    - [上位概念]
    - [下位概念]
  - [Lexical Category]
    - [近接語]
    - [直前語]
    - [直後語]

# 語から辞書項目を作るプロンプト

## [語の辞書項目]と「意味」項目の定義

**[語の辞書項目]** は、ある語Aが与えられた時、次のような項目からできているとします。以下のプロンプトで、語AまたはAは、与えられた具体的な文字列です。

- **[意味]** 語Aの意味を短いことばで述べたものです。もし、その語の意味が[品詞]によって分かれるのなら、[品詞]と[通し番号]のペアをラベルとして先頭に置いてから、[意味]を記述します。この時、対応する意味を持つ語Aを含む例文を、"(" + 例文 + ")"の形式で付け加えてください。その後改行します。この[意味]項目のラベルを "【「 + A + 」の意味】"とします。

## [Semantic Category]の項目の定義

- [Semantic Category]は、次のような項目からできています。

この[Semantic Category]項目のラベルを "【「 + A +」の Semantic Category 】"とします。

- [同義語] 語Aと同じ意味を表す語を、';' で区切って並べます。並びの長さは10程度とします。出力の際、先頭には "[同義語]"というラベルをつけます。

- [類義語] 語Aと意味的に近い意味を表す語を、';' で区切って並べます。並びの長さは10程度とします。出力の際、先頭には "[類義語]"というラベルをつけます。

## [Semantic Category]の項目の定義の続き

- **[上位概念]** 語Aの上位概念を表す語を一つ選んで、概念の包含関係で小さいものから大きなものへの順に、' < ' で区切って並べます。並びの長さは5程度とします。出力の際、先頭には "[上位概念]" というラベルをつけます。たとえば、生物学的な上位概念を一つ選べば、"人間"の上位概念は、人間 < 哺乳類 < 動物 < 生物 です。先にあげた[類義語]は、ある上位概念に共通に所属する語の集まりと考えることができます。

- **[下位概念]** 語Aの上位概念を表す語を一つ選んで、概念の包含関係で大きいものから小さいものへの順に、' > ' で区切って並べます。並びの長さは5程度とします。出力の際、先頭には "[下位概念]" というラベルをつけます。たとえば、地理的なエリアで下位概念を選べば、"日本"の下位概念は、日本 > 北海道 > 稚内です。

## [Lexical Category]の項目の定義

- [Lexical Category]は、次のような項目からできています。この[Lexical Category]項目のラベルを "【「 + A +」の Lexical Category 】"とします。
  - [近接語]ある文の中で、その語Aに近接して現れる確率が高い語を、';' で区切って並べます。並びの長さは15程度とします。出力の際、先頭には "[近接語]" というラベルをつけます。
  - [直前語]ある文の中で、その語Aの直前に現れる確率が高い語を、';' で区切って並べます。並びの長さは15程度とします。出力の際、先頭には "[直前語]" というラベルをつけます。
  - [直後語]ある文の中で、その語Aの直後に現れる確率が高い語を、';' で区切って並べます。並びの長さは15程度とします。出力の際、先頭には "[直後語]" というラベルをつけます。

# プロンプトの動作サンプル

語「歩く」の辞書項目を作成してください。

わかりました。語「歩く」の辞書項目を作成してみます

【「歩く」の意味】:

動詞1: 両足を交互に前に進めて移動する。(例:彼は公園を歩いた。)

動詞2: 一定の目的や方向に進む。(例:彼は成功へと歩いている。)

【「歩く」のSemantic Category】:

[同義語] 歩行する; 徒歩で行く; 徘徊する; 散歩する; 闊歩する; 行進する; 練り歩く; 遊歩する; 歩く; うろつく

[類義語] 走る; ジョギングする; 進む; 旅する; 移動する; 歩みを進める; 旅行する; 遊ぶ; 散策する; 探検する

[上位概念] 移動 < 行動 < 生物の活動 < 生物

[下位概念] 歩く > 超高層ビルを歩く > 空中を歩く > ワイヤーを歩く

### 【「歩く」のLexical Category】:

[近接語] 道; 公園; 散歩; ペース; 歩幅; 速度; 歩数; 遠く; 距離; 目的地; 環境; 風景; エリア; 足元; 自然

[直前語] ゆっくりと; 速く; 楽しく; 遠くまで; 道を; 一緒に; 独りで; 公園を; 自然を; 環境を

[直後語] を開始する; を終える; を続ける; と感じる; のスピード; の距離; のペース; で移動する; で旅行する; で探検する

以上が「歩く」の辞書項目となります。

語「知能」の辞書項目を作成してください。

了解しました。「知能」の辞書項目を以下に作成します。

【「知能」の意味】:

名詞1: 論理的思考、学習、理解、自己認識、認知、通信、問題解決などの能力。(例: 彼は非常に高い知能を持っている。)

【「知能」のSemantic Category】:

[同義語] インテリジェンス; 知性; 知覚; 知恵; 知識; 認識能力; 知覚力; 知悟; 心知; 知覧

[類義語] 賢さ; 慧眼; 洞察力; 了解; 理解力; 創造性; 技巧; 能力; 資質; 知識力

[上位概念] 知能 < 心 < 生物 < 生命体 < 存在

[下位概念] 知能 > 人間の知能 > 認識能力 > 論理的思考

## 【「知能」のLexical Category】:

[近接語] 人工; 高い; 低い; 感情; 能力; 学習; 理解; 認識;  
システム; ロボット; テスト; 研究; 開発; 進化; 技術

[直前語] 高い; 低い; 人工の; 驚異的な; 進化した; 現代の;  
認知的な; システムの; ロボットの; 複雑な

[直後語] を持つ; を開発する; を試す; がある; のテスト; の研  
究; の進化; の比較; を超える; の未来

以上が「歩く」の辞書項目となります。

語「愛憎」の辞書項目を作成してください。

了解しました。「愛憎」の辞書項目を以下に作成します。

【「愛憎」の意味】:

名詞1: 愛情と憎しみ、つまり極めて強い好き嫌いを表す。(例: 彼の行動は人々から激しい愛憎を受けた。)

【「愛憎」のSemantic Category】:

[同義語] 愛恨; 愛妬; 恩讐; 恩怨; 好悪

[類義語] 感情; 情緒; 激情; 情感; 情操; 気分; 情状; 気持ち;  
情意; 心情

[上位概念] 愛憎 < 感情 < 心 < 生物 < 生命体 < 存在

[下位概念] 愛憎 > 激しい愛憎 > 愛憎の対象 > 個別の愛憎

## 【「愛憎」のLexical Category】:

[近接語] 激しい; 深い; 複雑な; 混ざった; 交錯する; 紛れもない; 二元的な; 相反する; 明確な; 難解な

[直前語] 人々の; 彼の; 彼女の; 彼らの; 複雑な; それらの; 混ざった; 深い; 激しい; 個々の

[直後語] が深い; を持つ; が激しい; が交錯する; を感じる; が強い; を見せる; が顕著である; が存在する; を示す

以上が「愛憎」の辞書項目となります。

語「King」の辞書項目を作成してください。

### 【「King」の意味】:

名詞1: 一国を治める男性の君主を指す。(例: The king signed the decree.)

名詞2: チェスやカードゲームなどで最も重要な役割を果たすピースまたはカード。(例: He moved his king.)

名詞3: ある分野や領域で最も優れた、または支配的な人や物を指す。(例: He is known as the king of pop.)

### 【「King」のSemantic Category】:

[同義語] Monarch; Ruler; Sovereign; Emperor; Tsar

[類義語] Queen; Prince; Duke; Lord; Leader;  
Emperor; Tsar; Baron; Count; Sultan

[上位概念] King < Monarch < Ruler < Human < Mammal < Animal

[下位概念] King > Country's King > King's Subjects

### 【「King」のLexical Category】:

[近接語] rule; reign; throne; kingdom; queen; prince; royal; crown; sovereign; authority

[直前語] the; a; his; her; this; old; young; great; powerful; beloved

[直後語] of; is; was; has; rules; reigned; orders; declared; ascended; died

語「Hallucination」の辞書項目を作成してください。

【「Hallucination」の意味】:

名詞1: 病状または薬物の影響で起こる、存在しないものを見たり聞いたりする体験。(例: The patient was having hallucinations.)

【「Hallucination」のSemantic Category】:

[同義語] Delusion; Mirage; Illusion; Phantasm; Vision; Apparition; Fantasy; Dream; Reverie; Trip

[類義語] Paranoia; Delirium; Insanity; Madness; Confusion; Deception; Trickery; Deceit; Dizziness; Vertigo

[上位概念] Hallucination < Perception Disorder < Mental Disorder < Disorder < Medical Condition

[下位概念] Hallucination > Visual Hallucination >  
Auditory Hallucination > Olfactory Hallucination >  
Tactile Hallucination

### 【「Hallucination」のLexical Category】:

[近接語] visual; auditory; perception; reality;  
disorder; schizophrenia; psychedelic; delusion;  
experience; drug

[直前語] a; the; her; his; severe; induced; visual;  
auditory; terrifying; vivid

[直後語] of; is; was; caused; experienced; due;  
resulting; accompanied; suggesting; involving

# 「語の意味」の階層構造

「語の意味」の階層構造・ネットワーク構造を、  
GPT-4は把握できる。

語"human"の可能な上位概念を30個あげよ。

その時、どういう視点から、この上位概念が選ばれたかを説明せよ。

以下は、「human」の可能な上位概念を30個列挙したもので、それぞれの視点を説明します。

Primate(霊長類) - 進化学的な分類

Mammal(哺乳類) - 生物学的な分類

Vertebrate(脊椎動物) - 生物学的な分類

Animal(動物) - 生物学的な分類

Organism(生物) - 生物学的な分類

Life form(生命体) - 生物学・哲学的な分類

Biological entity(生物の実体) - 生物学・哲学的な分類

Entity(実体) - 哲学的な分類

Being(存在) - 哲学的な分類

Sentient being(知覚する存在) - 哲学・心理学的な分類

Biped(二足歩行者) - 物理的特徴

Homo sapiens - 人類学的な分類

Carnivore(肉食者) - 食性に基づく分類

Terrestrial animal(地球上の動物) - 生息地に基づく分類

Earth dweller(地球生物) - 生息地に基づく分類

Natural entity(自然の存在) - 哲学的な分類

Eukaryote(真核生物) - 細胞学的な分類

Multicellular organism(多細胞生物) - 細胞学的な分類

Chordate(脊索動物) - 生物学的な分類

Omnivore(雑食者) - 食性に基づく分類

Endotherm(恒温動物) - 生物学的な分類

Hominid(ヒト科) - 人類学・進化学的な分類

Anthropoid(類人猿) - 進化学的な分類

Anthropoid(類人猿) - 進化学的な分類

Creature(生き物) - 生物学・哲学的な分類

Life(生命) - 哲学的な分類

Fauna(動物界) - 生物学的な分類

Physical entity(物理的実体) - 物理学的な視点

Biological being(生物的存在) - 生物学・哲学的な分類

Homo genus(ホモ属) - 人類学・進化学的な分類

Earth inhabitant(地球の生物) - 生息地に基づく分類

注意:これらの上位概念は、物理的な存在(生物学、進化論)、行動(食

There was an error generating a response

語"human"の可能な上位概念を30個あげよ。

その時、どういう視点から、この上位概念が選ばれたかを説明せよ。

「human」の可能な上位概念を選ぶ視点は多種多様で、生物学、社会学、哲学、心理学など様々な学問領域の視点から考えることができます。以下にそれぞれの視点から選ばれた「human」の上位概念を30個挙げます。

生物学的視点(進化学、分類学):

Primate - ヒトは霊長類に分類されます。

Mammal - ヒトは哺乳類に分類されます。

Vertebrate - ヒトは脊椎動物に分類されます。

Animal - ヒトは動物界に分類されます。

Eukaryote - ヒトの細胞は真核細胞です。

Organism - ヒトは生物(有機体)です。

Multicellular organism - ヒトは多細胞生物です。

Hominid - ヒトはホミニッド(人類)に分類されます。

Biped - ヒトは二足歩行の生物です。

Homo genus - ヒトはホモ属に属します。

### 社会科学学的視点(社会学、人類学):

Social being - ヒトは社会的な存在です。

Cultural being - ヒトは文化を持つ存在です。

Rational being - ヒトは理性を持つ存在です。

Political animal - ヒトは政治的な動物(アリストテレスの表現)です。

Family member - ヒトは家族の一員とされます。

### 哲学的視点(存在論、形而上学)

Being - ヒトは存在するものです。

Sentient being - ヒトは感覚を持つ存在です。

Moral agent - ヒトは道徳的に行動する主体です。

Conscious being - ヒトは意識を持つ存在です。

Physical entity - ヒトは物理的な存在です。

### 心理学的視点(認知科学、神経科学):

Cognitive agent - ヒトは認知的な主体です。

Emotional being - ヒトは感情を持つ存在です。

Learning being - ヒトは学習する存在です。

Perceiving being - ヒトは知覚する存在です。

Communicative being - ヒトはコミュニケーションを取る存在です。

### 環境的視点(地球環境、生態学):

Terrestrial animal - ヒトは地球上の生物です。

Earth inhabitant - ヒトは地球に住む生物です。

語“人間”の可能な上位概念を30個あげよ。

その時、どういう視点から、この上位概念が選ばれたかを説明せよ。

「人間」の上位概念を選ぶ視点は多種多様で、それは生物学的な視点、社会学的な視点、哲学的な視点、心理学的な視点、文化的な視点などから選ぶことが可能です。以下にそれぞれの視点から選ばれた「人間」の上位概念を30個挙げます。

生物学的視点(進化学、分類学):

ヒト属 - 人間はヒト属に分類されます。

ヒト科 - 人間はヒト科に属します。

霊長類 - 人間は霊長類に分類されます。

哺乳類 - 人間は哺乳類に分類されます。

脊椎動物 - 人間は脊椎動物に分類されます。

動物界 - 人間は動物界に分類されます。

真核生物 - 人間の細胞は真核細胞です。

生物 - 人間は生物です。

多細胞生物 - 人間は多細胞生物です。

### 社会学的視点:

社会生物 - 人間は社会的な生物です。

文化的存在 - 人間は文化を持つ存在です。

理性的存在 - 人間は理性を持つ存在です。

政治的存在 - 人間は政治的な存在です。

家族の一員 - 人間は家族の一部です。

### 哲学的視点:

存在 - 人間は存在するものです。

意識的存在 - 人間は意識を持つ存在です。

倫理的存在 - 人間は倫理的な存在です。

自由意志の持ち主 - 人間は自由意志を持つ存在です。

## 心理学的視点:

認知的存在 - 人間は認知する存在です。

感情的存在 - 人間は感情を持つ存在です。

学習する存在 - 人間は学習する存在です。

知覚的存在 - 人間は知覚する存在です。

## 文化的視点:

言語使用者 - 人間は言語を使用する存在です。

信仰の持ち主 - 人間は信仰を持つ存在です。

伝統の保持者 - 人間は伝統を保つ存在です。

## 環境的視点:

地球の生物 - 人間は地球に生息する生物です。

環境の一部 - 人間は自然環境の一部です。

地球の支配者 - 人間は地球上の主要な支配者ともいえます。

大気呼吸者 - 人間は大気中の酸素を利用して呼吸します。  
生態系の一部 - 人間は生態系の一部です。

これらの視点は、人間が生物学的にどのように分類されるか、社会や文化の中でどのように位置づけられるか、哲学的にどのような特性を持つか、心理学的にどのような能力を持つか、そして地球上の生物としてどのような役割を果たしているかを反映しています。



# 数学的展望

1 マルレク 言語の意味の数学的構造

## 本論文のまとめ

- 線形代数のよく知られたツールを使って単語の埋め込みを理解し、Formal Concept Analysisをカテゴリー的な用語で枠組みすることによって、純粹にsyntacticalな入力から言語の構造的特徴を理解するパラレルな物語を見出すことができる。
- より具体的には、実世界のデータに見られるsyntacticalな関係をエンコード化した実数値行列を用いると、線形代数を用いて、何らかの意味情報と構造を示す意味の空間に渡すことができる。

- $\{0,1\}$ 値行列を得るためにカットオフを導入することで、構文から生じる意味構造を明らかにするためにFormal Concept Analysisを使用することができる。
- 本稿で使用する線形代数とenrichedカテゴリー理論からのよく知られたツール(主成分分析とFormal Concept Analysis)には目新しいものはないが、両ツールのセットを取り巻く平行な物語はあまり知られていない。

## 課題と展望

しかし、物語を伝えることよりも重要なのは、enriched カテゴリー論の枠組みが、統語論から意味論がどのように生まれるかについての理解を深め、意味論の構造を研究するために、線形代数にヒントを得た新しいツールを提供できる可能性である。

# 線形代数とFormal Concept Analysis を より近づける

すぐに思いつくアプローチとしては、線形代数とFormal Concept Analysis をより近づける方法がある。

拡張された実数 $[-\infty, \infty]$ または単位区間 $[0, 1]$ は、closed monoidal categoryの構造を与えることができ、他のカテゴリーをenrich化するための適切な基底カテゴリーとなる。

つまり、 $[-\infty, \infty]^X$ は、拡張された実数上に値を取る集合 $X$ 上の関数であり、 $[-\infty, \infty]$ でenrich化されたpresheafのカテゴリーと見なすことができる。そして、行列 $M$ またはその変形は、拡張された実数のカテゴリー上でenrich化されたprofunctorとみなすことができる。

# nucleiの構造

nucleiの構造は、 $\{0, 1\}$ 行列を得るためのカットオフを導入することなく、Formal Concept Analysisに匹敵する方法で直接研究することができる。

このアイデアは、10年ほど前から特定の数学界で広まっていた。[Pav12、Pav21、Wil13、Ell17、Bra20、BTV22]とその中の参考文献を参照されたい。

このアプローチは、すべてのカットオフ値に対するFormal Conceptのlatticeを、実数によって複雑に変調されたFormal Conceptという1つの数学的対象に統一する方法と考えることができる。

# 離散カテゴリーからenriched カテゴリーへ

もう一つの重要な点は、線形代数の議論が、要素上の順序以上の構造を持たない集合 $X$ と $Y$ から始まったことである。

カテゴリー的な議論を可能な限り平行なものにするため、 $X$ と $Y$ は同一射以外の射を持たない離散カテゴリーとみなされた。

しかし、 $X$  と $Y$  に射すなわちenrich化する射を導入することは可能であり、それらの射の存在は、説明される構成全体を通して、最終的には、任意の  $X$ - $Y$  profunctorのnucleiに反映される。

# 文法のカテゴリー論的再編成とDisCoCat

これらの情報を既存の線形代数のツールで説明する方法は存在しないのだが、最近の博士論文[dF22]は、正則文法、文脈自由文法、pregroup文法など、多くの言語学的文法モデルをカテゴリー理論の言語で再構成しており、Coeckeらの言語の構成的分布モデル[CCS10]の広い文脈に適合している。

2010年代まで遡るこれらのモデルでは、文の意味は、文法を構成する単語の意味と、それらの単語が文法の規則に従ってどのように構成されるかから生じるものと提唱されている。

この関係は、選択された文法カテゴリーから、有限次元ベクトル空間のような分布情報を捉えるカテゴリーへのfunctorによってモデル化される。

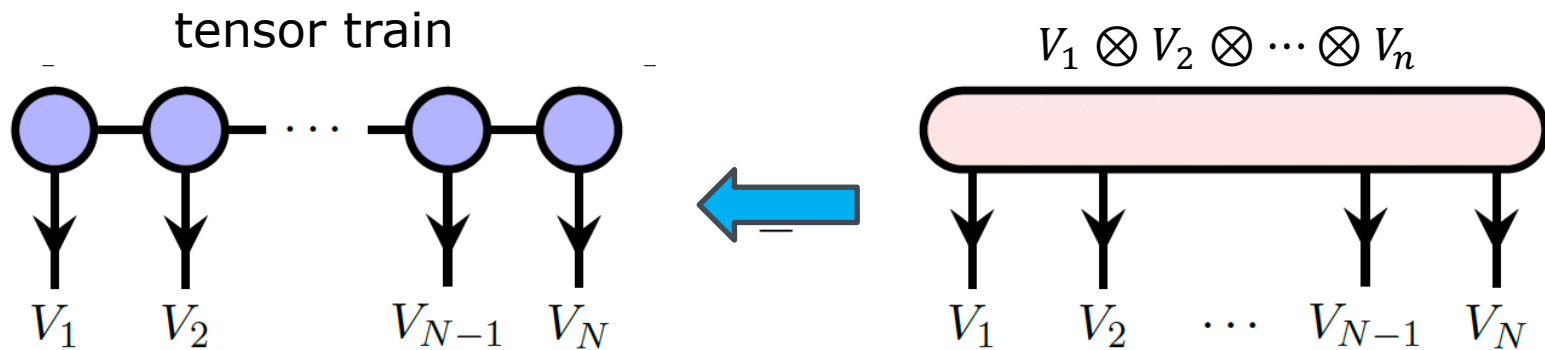
このようなモデルは構文論から意味論への通路と考えることもできるが、文法の選択に大きく依存する。

しかし、ここでのポイントは、もし $X$ が集合以上の構造を持つことから始めたいのであれば、この論文で説明した数学的な物語を崩すことなく、enriched カテゴリー論がその方法を提供するという事である。

# 多重線形代数とテンソル・ネットワーク

線型代数の道具が、enriched カテゴリー理論における類似の道具よりもさらに発展している場所のひとつが、多重線型代数である。

例えば、ベクトル空間  $V_1 \otimes V_2 \otimes \dots \otimes V_n$  のテンソル積のテンソルを **tensor train**、あるいは **matrix product state** と呼ばれるものに因数分解することは、 $n - 1$  個の互換性のある切り捨て **SVD** の列として解釈できる。



カテゴリ  $C_1 \times C_2 \times \dots \times C_n$  の積上の functor に対する「互換性のある nuclei の列」についての同様の理論を我々は知らない。

テキストデータは、単なる語とコンテキストのペアというよりも、長い文字列と見なすのが自然であることを考えると、このようなオブジェクトのカテゴリーを強化したものが、言語において構文的なものから多層的な意味構造がどのように生まれるかを理解するのに適していると考えるのが妥当であろう。

# AI技術の数学的基礎の革新を目指して

今後の数学的展望を示したのが、このセクションです。

「重要なのは、enriched カテゴリー論の枠組みが、統語論から意味論がどのように生まれるかについての理解を深めることである。」

その上で、

「意味論の構造を研究するために、線形代数にヒントを得た新しいツールを提供できる可能性がある。」

と語ります。

前者の指摘については、前回・前々回のセミナー「大規模言語モデルの数学で構造」で紹介した、enriched カテゴリーを用いて Syntaxカテゴリーに確率概念を導入し、再構成された copresheaf 意味論を導くという展開が、今後の基本的な枠組みになるということだと思います。

後者の「線形代数にヒントを得た新しいツール」の構想は、今回の論文での幾つかの実験から、何を教訓として学ぶのかということだと思います。

この点では、いくつかのアイデアと課題が提案されています。

# 線形代数とFormal Concept Analysisを より近づける

ひとつは、「線形代数とFormal Concept Analysisをより近づける」ことが示唆されています。

具体的には、実数 $X$ 上に値を取る関数を $[-\infty, \infty]$ でenrich化されたpresheafのカテゴリーと見なし、行列 $M$ またはその変形は、拡張された実数のカテゴリー上でenrich化されたprofunctorとみなすことができるということです。

profunctor上のnuclei(それは行列の固有値・固有ベクトルの対応物と考えることができます)の構造を直接研究することで、それは可能になると言います。

線形代数の議論は、要素上の順序以上の構造を持たない集合 $X$ と $Y$ から始まっているのです( $m$ 行 $n$ 列というように)が、そこもenrich化でき、そのenrich化は任意の $X$ - $Y$  profunctorのnucleiに反映されるといいます。

# 文法のカテゴリー論的再編成とDisCoCat

意味に構造を見出そうとする彼女の問題意識は(それは、フレーゲ的な「構成性の原理」と同じものです)、あらためて「文法」概念の再構成に向かうのかもしれない。

「文法」のカテゴリーから意味を捉える有限ベクトル空間への functor というフレームは、Coeckeらの(そして、かつての彼女自身の)DisCoCatのアプローチに他なりません。

それは、特定の文法カテゴリーの選択に依存するという大きな問題を抱えているのですが、その弱点は、enrich化で乗り越えられるかもしれないと言います。

「SyntaxからSemanticへの道」は、「もし $X$ が集合以上の構造を持つことから始めたいのであれば、この論文で説明した数学的な物語を崩すことなく、enriched カテゴリー論がその方法を提供するということである。」

# 多重線形代数とテンソル・ネットワーク

彼女が、注目しているもう一つフレームは、線形代数の拡大としての多重線形代数です。

「例えば、ベクトル空間  $V_1 \otimes V_2 \otimes \dots \otimes V_n$  のテンソル積のテンソルを tensor train、あるいは matrix product state と呼ばれるものに因数分解することは、 $n - 1$  個の互換性のある切り捨てSVDの列として解釈できる。」

こうした手法は、彼女の以前の論文、“Modeling Sequences with Quantum States: A Look Under the Hood”  
<https://arxiv.org/pdf/1910.07425.pdf> で説明されています。

一年前の丸山のセミナー「密度行列  $\rho$  で理解する確率の世界」  
<https://www.marulabo.net/docs/density2/> は、こうしたアプローチを紹介したものです。

テンソル・ネットワークへの注目の理由ですが、彼女が次のように述べていることは重要だと思います。

「テキストデータは、単なる語とコンテキストのペアというよりも、長い文字列と見なすのが自然であることを考えると、このようなオブジェクトのカテゴリーを強化したものが、言語において構文的なものから多層的な意味構造がどのように生まれるかを理解するのに適していると考えるのが妥当であろう。」

copresheaf 意味論は、語、フレーズ、文、文の連続 ... と続く「長い文字列」を対象にしたものでしたから。

# AI技術の数学的基礎の探究は続く

なかなか壮大な展望を持っていることがわかります。

現在のAI技術の生み出した言語観は暫定的なもので、その根拠となる数学理論も完全なものではありません。

Tai-Danaeの「ニューラル言語モデル」の批判はその言語観への批判から発したものです。AI技術そのものの数学的基礎の探究の深化の突破口となることを期待しています。

こうした進行は、同時に、新しい科学的発見・技術的发展にインスパイアされて、数学自体が変化・发展することを示しています。

こうした展開が意味するのは、直接的には、人間の認識能力についての認識が发展するという事なのですが、それが、人間という存在に対する認識の深化に実を結ぶことを望んでいます。



