

## これからの数理言語モデル

### □PoincareEmbeddings を使った Taxonomy 「概念階層化」

双曲空間上のポアンカレ球モデルでリーマン多様体を表現するもの

$$d(u, v) = \operatorname{arccosh} \left( 1 + 2 \frac{\|u-v\|^2}{(1-\|u\|^2)(1-\|v\|^2)} \right) \quad \text{逆双曲線余弦関数}$$

$$\text{where, } g_x = \left( \frac{2}{1-\|x\|^2} \right)^2 g^E \quad \text{リーマン計量テンソル}$$

これで概念ベクトル間の指数距離が求まるので、階層構造化がし易くなる。

詳しくは、当 HP 「技術情報」 > 「参考文献」 を参照してください。☺

### □勾配積分 Integrated Gradients による 「知識構造化」

これは画像診断時に、学習画像と入力画像との中間画像を勾配積分によって m 枚自動生成する式です。Morphing とは「似て非なるもの」であることは式をみれば判りますね。

$$\text{勾配積分} \quad M^c \approx \sum_{t=1}^m \operatorname{ReLU} \left\{ \frac{1}{m} \sum_{k=1}^N \sum_{i,j} \frac{\partial y^c(r(\frac{t}{m}))}{\partial A_{ij}^k(r(\frac{t}{m}))} \right\} \Delta \left( r(\frac{t}{m}) \right)$$

これを言語モデル化するには強化学習 RL の価値推定(報酬)される状態行列の自動生成に勾配積分を使う。これを simulating することによって知識構造化が構築できて、Chatbot などで使われる「問い返し」が自動生成できる。従来の深層学習 DL は階層化が不得意な技術なので、これと併用することを推奨します。👍

### □Persistent Homology で 「概念構造解析」

位相空間上の位相多様体(概念ベクトル...)などは連続した変形に伴い、その性質や構造が捉え難いが、パーシステント・ホモロジーは位相不変量の穴 Genus のフィルトレーションで概念構造を解析する。

$$H_k(\mathfrak{X}) = \frac{\ker \partial_k}{\operatorname{Im} \partial_{k+1}} \cong \left( \bigoplus_{j=1}^s (z^{b_j}) / (z^{d_j}) \right) \oplus \left( \bigoplus_{j=s+1}^{s+t} (z^{b_j}) \right)$$

時系列データでもクィーバーの表現論で対処ができ、ランダムトポロジーへの研究が進んでいる。詳しくは文献を参照ください。📖

### □Sparse Modeling で 「見える層」 ホワイト box 化

$L_1$  ノルム lasso 回帰は  $L_2$  ノルム ridge 回帰に負けて忘れられた正則化手法であったが、CNN が流行ったことで研究が進み、今や深層学習の欠点である「隠れ層の見える化」の最有力候補にまでなった。

$$l(\Lambda, \Psi) = \log |\Lambda \Lambda^T + \Psi| + \operatorname{tr} \{ (\Lambda \Lambda^T + \Psi)^{-1} S \} \quad \text{スパース因子分析}$$

$$\min_{A, B} \|X - XBA^T\|_F^2 + \lambda \|B\|_F^2 + \sum_{k=1}^K \lambda_k \|b_k\|_1 \quad \text{subject to } A^T A = E_K \quad \text{スパース主成分分析}$$

正則化項の行列対角成分の尾根(リッジ)を膨らませて正則化する基本的アイデアでいろいろな見える化スパースモデルが創れます。📖

### □Cohomology 群による 「因果推論」

概念ベクトルを n 次元多面体 |C| として、その有向面の貼り合わせ depend によって鎖群というコホモロジー群を文と定義する。

$C_*: \partial_l[k_{l-1}:k_l] \cdot C_l[k_l] = C_{l-1}[k_{l-1}]$	ホモロジー鎖体行列演算
$C^*: \delta^{l-1}[k_l:k_{l-1}] \cdot C^{l-1}[k_{l-1}] = C^l[k_l]$	コホモロジー鎖体行列演算
$H^l(C^*) = Ker \delta_l / Im \delta_{l-1}$	$l$ 次元コホモロジー群
形態素 $\subseteq$ 文節 $\subseteq$ 連文節 $\subseteq$ 文 $\subseteq$ 文章 $\subseteq$ 文書	文書鎖体の包含関係
形態素 $\xrightarrow{\delta}$ 文節 $\xrightarrow{\delta}$ 連文節 $\xrightarrow{\delta}$ 文 $\xrightarrow{\delta}$ 文章 $\xrightarrow{\delta}$ 文書	貼り合わせ $\delta$ でのコチェイン複体

「文や文章に意味がある」とは、意味空間上のイデアルを基とする商群になっていること。そこで文間の因果関係を Coherence 干渉で推論する。☞

### □QuantumAI「量子深層学習」で勾配逆伝搬(自動微分)

非テンソル積で統計が不成立な Entangle 状態(量子もつれ)「(時空を超えた)不気味な遠隔作用」を利用した Q コンが実用化され始めている。いま求められているのは論理ゲート操作を創る QAI エンジニアです。

$$|B_{00}\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B) \quad \text{ベル状態}$$

$$|T_L\rangle = e^{-i\frac{\pi}{8}Z} \frac{1}{\sqrt{2}}(|0_L\rangle + |1_L\rangle) \quad \text{魔法状態}$$

$$\frac{d\langle H(\theta) \rangle}{d\theta} = \frac{1}{2} \{ \langle H(\theta + \frac{\pi}{2}) \rangle - \langle H(\theta - \frac{\pi}{2}) \rangle \} \quad \text{エネルギー勾配}$$

$$\frac{df(\theta)}{d\theta} := \frac{d}{d\theta} \langle \psi | U_G^\dagger(\theta) A U_G(\theta) | \psi \rangle = r \{ f(\theta + \frac{\pi}{4r}) - f(\theta - \frac{\pi}{4r}) \} \quad \text{observable 勾配}$$

TensorFlowQuantum では有効逆伝搬ハミルトニアンで損失関数の期待値勾配を古典ニューラルネットワーク NN で処理するようになってきているが、量子 CNN(MERA)や量子 RNN(QAOA)などの改良版を創れる QAI エンジニアが望まれる。☺

### □IUT 宇宙際タイヒミュラー理論で synonym「類義性」

$$\{ \underline{q}^{j^2} \}_{j=1, \dots, l^*} \mapsto \underline{q} \quad \text{言語体 } F \text{ の自己同型を仮定する}$$

自己同型は、束の次数を保持するので右辺の絶対値は、左辺の次数の絶対値と比べて小さいという不等式が帰結されるが、このような自己同型は存在しない。しかし、「環・スキーム論=正則構造」と見做すと「同義反復的解決」がなされる。これを相異なる正則構造を持つリーマン面間の擬等角写像といい、この部分的解体で生ずる「歪み」を計算するのが IUTeich である。

すなわち、文書や単語の意味概念位相空間上の類義性は、位相 Topology の連続性と加群 Group の対称性、コチェイン複体のアーベル群列 Cohomology の貼り合わせ、圏 Category 間の関手の層で構成される空間際イデアルで計算される。■

・補足説明・

<タイヒミュラー空間>

面積と向きを保存する相似写像とは等長写像

$$\text{曲面の種数 } g \quad \chi(S) = 2 - 2g \quad g \geq 2$$

タイヒミュラー空間  $\tau_g$  は、種数  $g$  の双曲面空間

<ガウス-ボネの定理>

$$\text{コンパクト双曲面 } S \text{ の双曲面積} \quad -2\pi \chi(S) \quad \text{三角形分割}$$

$$\text{オイラー特性数} \quad \chi(S) = F - E + V \quad F = \text{面数}, E = \text{辺数}, V = \text{頂点数}$$