by ふきや 8/7/2019

極秘 Classified Documents 文書

自律学習

Autonomous Learning Engine

AL エンジン

開発仕様書

正規版

Ver.1.3

by ふきや 8/7/2019

INDEX

8)SCR（SemanticConceptualDependency）ー係受空間

9)SCS（SemanticConceptualSP）ー意味タグ空間

10)SCU（SemanticConceptualUnknown）ー未知語推定空間

（2）RunTimeX−RDB/BinaryData

1)RTP（RunTimeMorpheme）+RTB（RunTimeBinary）ー単語データ群

2)CT （ConceptualTag)ー概念タグデータ群

3)ST（SemanticTag）ー意味タグデータ群

6. APPLICATION ………

（1）Chatbot（会話系質疑応答）

（2）SUMMARIZATION（自動要約+Index）

（3）HyperSemanticWeb（RDF、MetaDATA、MetaKnowledge）

◇◆このエンジンの特長◆◇

☆意味概念共役空間モデルの最先端性

→Topology や圏論、層の Cohomology、群論そして超階述語論理…定性化

→言語モデルに位相幾何学や群論などを採用して外挿を実現化

☆意味概念 SCM（Semantic Conceptual Map）生成法の独創性

→意味概念空間のイデアル生成にカテゴリー層の概念化

→Thesaurus や加法性、分布解析での自動概念生成化

☆概念タグ CT(Conceptual Tag)の差別性

→形態素意味概念階層構造のコホモロジーと SCM との共役性化

→文章の一意性や因果関係そしてコンパクト性化

☆意味タグ ST(Semantic Tag)の高度性

→各階層間の層別群化を圏論で表現化

→文に依存した語彙の意味変化や曖昧性を解消して意図の明確化

☆知識生成 KE と構造化 KS の超先進性

→超述語式知識の自動生成と構造化を超階述語論理化

→曖昧で複雑な知識を包括的で多角的構造化

☆知識推論は脳モデルを Simulating

→新しい知識をトポロジカル理論…で自律学習

→新知識や手法・推論の自律学習で動的知性エージェントが生成

※これは機械学習 Machine Learning でも、深層学習 Deep Learning でも、人工知能 Artificial Intelligence でもありません。新しい「自律学習 Autonomous Learning：AL エンジン」です。

1．Mathematical background and definition 数学的背景と定義

（1）言語空間 L と概念空間 SC の基本定義

□言語空間 L [1]を有限個の元 m（形態素）を要素（変数）とする空間とし、文や文章、文書を構成する文法函数 $G=\{g_1,\cdots,g_k\}$ をそれの系列（曲線）とみなせば、意味ある文や文章、文書とは多変数多項式の斉次方程式 $g_1(m_1,\cdots,m_n)=\cdots=g_k(m_1,\cdots,m_n)=0$ の解となり、この全体を $V(g_1,\cdots,g_k)=\{M=(m_1,\cdots,m_m)\in L^n \mid g_1(M)=\cdots=g_k(M)=0\}$ とすると、この空間は代数多様体[2]AlgebraicVariety となる。これは Syntax 上、正則な文集合である。また、この部分空間を複素空間 C 上と考えれば、この解の全体は無限遠点と極を有する genus 空間になり、文構成群の分類構造となる。

□ $\Re$ 上の $F:L\to SC$ で定義される意味概念空間 SC : SemanticConceptualSpace は、$m\in L$ の族集合となり、Bourbaki 位相が入り、意味概念位相空間になる。また同時に Hausdorff 位相からの視点も考慮され、mやK[3]の近傍で位相多様体 manifold の構造を持ち、特異点のない空間（超平面）となる。尚、上記の写像関数 F は、Abelian 群から Abelian 圏への導来関手とする。すなわち、以下の各関手の右導来関手で微分の概念を持つ長完全系列を生成するものである。

[redacted]

という可換環でもあり、共役関係な階層でもある SC となる。

---

[1] 言語空間 L とは、表記言語の形態素空間のことで、文（又は文章、文書）は助詞などを含む形態素列の曲線となる

[2] 代数多様体とは、代数的演算を包含する多様体のことで、位相多様体 Manifold とは相違する

[3] K とは Knowledge 知識のことで、空間上の曲線で相や態の完結された文のことを知識という

すなわち、

$$\phi \xrightarrow{\partial_{\phi}} DM \xrightarrow{\partial_{DM}} PG \xrightarrow{\partial_{PG}} ST \xrightarrow{\partial_{ST}} CL \xrightarrow{\partial_{CL}} PR \xrightarrow{\partial_{PR}} M \xrightarrow{\partial_{M}} 0$$

$$0 \xrightarrow{d_0} M \xrightarrow{d_M} PR \xrightarrow{d_{PR}} CL \xrightarrow{d_{CL}} ST \xrightarrow{d_{ST}} PG \xrightarrow{d_{PG}} DM \xrightarrow{d_{DM}} \phi$$

という長完全系列に拡張可能な循環系列になる。
これは完全系列なので、準同型定理によりホモロジーとコホモロジーが定義・構築される。

□いわゆる、$\partial_{ST} : ST \to CL$ を準同型写像とすると、$ST / Ker\partial_{ST} \cong \mathrm{Im}\partial_{ST}$ が成立する。

[redacted]る Ker 値（ジョルダン標準形: $Ker(A-\lambda E)$ ）との意味合いも存在するので、これは固有値問題の一般化になっている。そのうえ、これは単射とのズレ度合でもあり、基近傍でもあると考えられので、ハウスドルフ空間の ε 近傍とはまったく違った近傍の概念でもあり、言語概念解析の数学的背景としているのが「ふきや理論」である。
□係り受け関係も各階層別に成立する。

$$F(M) := \{(m_i \to m_j) \mid m \in M\}$$

$$F(PR) := \{(pr_i \to pr_j) \mid pr \in PR\}$$

……

$$F(DM) := \{(dm_i \to dm_j) \mid dm \in DM\}$$

上記、圏内の完全系列を導来関手 F によって意味概念空間 SC へ写像される。
□したがって、$F(DM) \to F(PG) \to F(ST) \to F(CL) \to F(PR) \to F(M)$ が SC 内で生成・構造化される。但し、$F = \{f_1, \cdots, f_k\}$ のように導来関手の集合になる。これが ST へ変換定義される。

□上記では、意味概念自体が位相構造を持ち、すなわち意味ある最小単位である形態素の「列」である意味概念が階層化された意味概念構造として構成される。これが知識 KE(超述語式 8 次元単体Δ：Simplex)へと変換される。

□Syntax 圏から Semantic 圏への写像で Syntax タグから Semantic タグ（ST という）への変換が

[redacted]

・完全な文の場合、形態素から文節、文節から連文節、連文節から文への射は、すべて全射になる。

・すなわち、形態素、文節、連文節、文の各集合への「係り受け」は、包含射と考えることができる。

Semantic 圏を仮にアーベル群 Ω とし、そこへの関手 F,G とする。

・Semantic 圏内の完全文は、意味概念同士の結び付きによる「意味の一意性」（対称性）による群と考えられる。

すると、Semantic 圏 Ω 内は、$F(U)\xleftarrow{F\psi}F(V)$ と $G(U)\xleftarrow{G\psi}G(V)$ という逆向きの係り受け（可換性）が成り立つことになる。但し、$\psi:V\to U$ という単射である。また、

関手 F,G に関して、$F\xrightarrow{\varphi}G$ を考えると、これは前層 $P=\Omega^{\tau^{\circ}}$ になる。φ の核 $Ker(-)$ も前層 P に入る。

[redacted]これは、表現可能関手であり、これで意味概念の一意性が証明される。

□今度は、Syntax 前層から Semantic 層への定義を考える。

◇Syntax 圏 τ 内の開集合 U に関し、イデアル $i \in I$ が存在し、被覆 $U = \bigcup U_i$ とする。

Semantic 圏 Ω 内で、$s_i \in F(U_i)$ とすると、$\rho_{U_i, U_i \cap U_j}(s_i) = \rho_{U_j, U_j \cap U_i}(s_j)$ となる。

但し、$i \neq j$ 、$U_i \cap U_j$ は共通開集合。$\rho(*)$ は写像関数であり、ここでは F を省略した。

これで「唯一」$F(U) = s$ が存在し、$\rho_{U, U_i}(s) = s_i$ 　がすべての $i \in I$ で成り立つことになり、関手

$F \xrightarrow{\varphi} G$ 　が層 Sheaf になる。これが関手間や層間（層を圏として）の関手の層 Sheaves の定義である。この空中戦（汎関数の一般化）が言語概念空間の構造（しくみ）を間接的に、俯瞰的に解析できるしくみである。

□視点を変えて層の定義を考えると、

◇位相空間 X の要素である α を含む開集合 U で $\alpha \in U \subset X$ を考える。

[redacted]

層 sheaves は、$O_X := \prod_{\alpha \in X} O_\alpha = \{(\alpha, f_\alpha) \mid f_\alpha \in O_\alpha, \alpha \in X\}$ 　と定義できる。

◇開集合 U の要素 α を中心とする等高線の断層を上から切断 f で落とすと茎になり、その交点が芽である。

従って、層とは、芽 $(f_\alpha, U\} \in O_\alpha$ と要素 $\alpha \in U \subset X$ との直和集合である。上式Πパイは直和の記号です。

関数どうしの多峰性の偏差 σ を求める式の一般化（離散）と考えても良い。

□複素関数論の曲線論でいうならば、複素関数曲線 X の開集合 U 内の点 p の近傍を正則とし、点 p を高々 m 次の極とした場合、$F(U)$ を層といい、$F(U) = O_X(mp)$ と書くことができる。簡単な例でいうと、$f = \frac{x^2}{(x-1)^3}$ 　2 次の零点と 3 次の極があり、無限遠点 $x = \infty$ で、$t = \frac{1}{x}$ とすると、$t = 0$ になり、$f = \frac{x^2}{(x-1)^3} = \frac{t}{(1-t)^3}$ なので、1 次の零点となる。従って、開集合 U の層間の層の定義を複素関数論（曲線論）でも表現できる。例題の二次の零点と一次の零点との違いは、複素空間だから相違することは判ると思うが、一次の零点になると虚軸ではなく、実数軸上の問題となることを意味している…これが重要です。

◇ちなみに切断は、$f := (\varphi_1(\alpha), \varphi_2(\alpha), \cdots, \varphi_n(\alpha))$ 　という α での Section と定義できる。

この切断系列は、$0 \to F \to L_0 \to L_1 \to \cdots$ 　という軟弱 Flabby (injection) な完全系列

$H^0(U,F)=\Gamma(U,F)$ から $0\to\Gamma(U,L_0)\to\Gamma(U,L_1)\to\cdots$ となる。

この切断 Section は、語彙の意味概念のイデアル特定には重要なファクターで、語彙の意味概念である開集合の分布状況が明示的(等高線のように…)に判るものである。
その為に、層や茎、芽、スキームなどの手法が必要になり、これが意味概念層になる。

(2)Topological Autonomous Learning(TAL)基本推論学習法
[redacted]

[redacted]

[redacted]

[redacted]
これで局所的な**自律学習** Autonomous Learning ができる。
□特長は、任意のテキストからイデアルを抽出し、そのイデアルに依存される階層化を行うこと。例えば、製造技術書類から構造概念と機能概念を抽出し階層化することにより、多視的で多角的な質問・照会に対する解答・回答が得られるだけでなく、曖昧な質問・照会に対しては、問い返しや聞き返しなどガイドやナビが行えることが最大の特長である。

（3）従来の統計的確率モデルとの比較

口ディープラーニングとの考察

従来のディープラーニングのネタは、$W=(XX^T)^{-1}XY^T$ なので、これを導出してみれば違いが判る

線形関数　$y=f(x)=w^T x$　モデル

評価関数　$E[w]=\sum_i \|y_i-f(x_i)\|^2$　勾配法

目的関数　$w=\arg\min_w E[w]=\arg\min_w \sum_i \|y_i-w^T x_i\|^2$　最小二乗法

[redacted]

[redacted]　変数 K は層名、η はステップ数

$W^K \leftarrow W^K+\Delta W^K$　誤差逆伝播法 BP

蛇足

$(XX^T)^{-1}X=\delta$　係数と入力ベクトル

$\Delta W^K \propto \delta Y^T$　相似変換（共役）

以上、これが DL の処理フローと学習法である。

口では、準ニュートン法と比べてみよう。

・入力d次元ベクトルを $x=(x_1,\cdots,x_d)$ とし、出力凸関数を $f(x)\in\Re^d$ とする。

・次の点を $x_{n+1}=x_n+\Delta x$ とし、最適解を逐次的に求める。　…①

・差分 Δx を移動する方向と量として、これを求める。

・次点①の出力凸関数処理を Taylor 展開で近似する。

[redacted]

差分ベクトルの積とは、以下の L-BFGS 法で述べる。

・ここで見易くする為に、置き換えます。

[redacted] …⑤

・この Δx が探索方向量であるので、ステップ係数 α を求める

$$\alpha^* = \arg\min_{\alpha} f(x_n - \alpha\Delta x) \qquad \cdots⑥$$

・探索方向量 Δx とステップ係数 α が求まったので、①式にステップ係数を付加して更新する

$$x_{n+1} = x_n + \alpha\Delta x \qquad \cdots⑦$$

□上記は Newton 法だが、ふたつの問題点がある。

ひとつは、H_n^{-1} は正定値が条件で、半正定値では収束しない。

もうひとつは、行列なので次元の階乗になる為、メモリがオーバーフローする。

・そこで準ニュートン法である L–BFGS 法では、正定値行列 $B_n = H_n^{-1}$ に近似すると、⑤式は

[redacted]

もともとの式は、簡易式 $g_n \cdot g_n^T \approx H_n$ でも求めることができる。

…とすると、更新は、

$$B_{n+1} = (I - \rho_n y_n s_n^T) B_n (I - \rho_n s_n y_n^T) + \rho_n s_n s_n^T \qquad \cdots⑩$$

但し、$\rho_n = y_n s_n^{T^{-1}}$

・この⑩式は、⑦式と等価である。
・ディープラーニングも隠れ層である行列 B_n の更新と等価である。すなわち、特徴ベクトルを求める為のウエイト行列を探索方向量ベクトル Δx のナブラベクトル $g_n = \nabla f(x_n)$ から $B_n = H_n^{-1}$ を求めていくことと等価である。すなわち、ディープラーニングは準ニュートン法と等価である。■
□従来のディープラーニング DL と新推論学習法 TAL の違いは、DL が推論の隠れ層をヒューリスティック(人手)で固定するので収束シミュレーションに膨大な時間、経費そしてデータ量を必要とした。その上、学習には勾配法と最急降下法で、L2 局所劣モジュラ(凸集合)化を使っている為に、[redacted]節つ形態素の各階層にした意味概念位相空間 SemanticConceptualTopologicalSpace となる。
□各階層の隣り合う並び方(連結)によって「意味が特定(生成)」されている。従って、形態素も単語同様に隣り合う連結によって意味がなされており、形態素の文字列も前の文字に依存・影響[4]され非相関関係のモデルになる。これがマルコフ性であり、位相的 N_gram 法を使う所以である。但し、skip_gram などのスキップが大きければ、無相関になる。これをエルゴート性[5]という。すなわち、大局的には無相関な関係も、局所的には相関関係があるということを言語処理では使っている。
□離散系も局所的には連続性(Gibbs 分布…)にして計算するのが応用数学の常套手段である。これは顕在化されたデータ $D = \{d_1, \cdots, d_n\}$ の背景には、それを生成する複数の潜在化された要[redacted]

[redacted]

$e = \{e_1, \cdots, e_k\}$ を基底パラメータ[7]という。また、パラメータを生成するパラメータをハイパーパラメータといい、ディリクレ分布などで生成するのが、最近のトピックモデルである。事前分布のハイパーパラメータであるディリクレ事前分布を尤度確率分布と同じ分布(共役分布)としていることによ

4 これはマルコフ性なので非可換性という意味で非相関性と表記している

5 Ergodic とは時空での一様性をいい、既約非周期的有限マルコフ連鎖のこと。漸近分布や収束条件になっている

6 一般には N(μ,σ^2)でランダムウォークな雑音を表現するサンプリングを特徴ベクトルから生成する

7 この理論では基底という座標系のベクトルではなく、イデアルという位相空間上の部分空間を指す

って、事後分布を簡単にして求めている。

2. TAGGER　＜CORE_ENGINE＞

旧バージョンであるPerceptronsEngineの学習法は、文字N-gram(Bi_gram+PseudoTri_gram)で行う「分ち書き」処理と形態素Bi_gramでの品詞付けを生コーパス(文書データ)からの文字列の頻度数と順序系列を確率変数とした「教師なし」学習法であった。また、形態素解析moduleと構文解析moduleも別物であった。

本開発では、「教師なし」学習法の欠点であった収束に手間と時間が掛かることと精度に問題があった為、「半教師あり[8]」学習法で行う。表層空間であるテキストから深層空間、すなわち意味概念空間への写像とする「ふきやモデル」を採用することで、TotalPerformanceを上げることができる。

(1)形態素解析 Morpheme_module

以下の2つのModulesを開発する

①分ち書き処理：SegmentationModule(SM)

②品詞付け処理：PartModule(PM)＋CT付与

＜手法＞

①入力文を文単位の文字ベクトルVに変換　入力文ベクトル

②Seg(V)=XML　内部ストアは形態素別文

③$(\mathrm{W,T}) = \underset{w,t}{argmax} \sum_{i=1}^{N} p(t_i, t_{t-1}) + p(w_i, t_i)$　PATとRTP

$$= \underset{w,t}{argmax}\, p(T)p(W|T) \approx \underset{w,t}{argmax} \prod_{i=1}^{N} p(t_i|t_{i-1}) \cdot \prod_{i=1}^{N} p(w_i|t_i)$$

[redacted]

[redacted]

⑥$\mathrm{P}(w_i) = max_w P(w_i|w_j, w_k)$　Lattice構造をViterbiで

$\mathrm{P}(w_{i+1}) = max_w\{P(w_{i+1})|P(w_i)\}$　再帰函数

＜解説＞

①分ち書き処理：SegmentationModule(SM)

文字列から形態素を分割する法：「分ち書き三原則」

・最長一致法 The Longest Consistent Method(LCM)　既知語に効果的

---

8 SemiParametric法と等価だが、厳密にはHalfNonpara法という方が正しい

・助詞探索法 Particle Search Method（PSM）	文節に効果的
・字種連結法 Type of Binding Method（TBM）	未知語に効果的
◦P(Morpheme_w ｜ LCM,PSM,TBM)　分布処理	目的関数

最大値主義ではなく、分布主義なので最大値の近傍から 2 分岐^ステップ数になる

②品詞付け処理：PartModule（PM）―BP 品詞を特定

・RTP 照合で品詞特定	RTP 参照
・品詞列 PAT[9]で品詞特定（可変長 CRF マルコフモデル）	品詞列 PAT 参照
◦P(Morpheme_t ｜ $t_1 \cdots t_k$)	目的関数

③CT 付与処理：ConceptualTag（CT）

形態素ごとの CT 付与―RTP との照合で付与

・未知語の場合、前後の CT から推定	CT 列 CAT[10]

＜例1＞例文：「顔面のどこかが痛い。」

RTB={顔面、の、のど、どこか、が、痛い、か、どこ}が登録されている…とする。

四角で囲った形態素は「助詞」で、下線の形態素は「未知語」

「最長一致法」を優先すると、「顔面」+「のど」+「こ」+「か」+「が」+「痛い」

「[redacted]どかな」+「風景」+「を」+「見た」

「助詞」を優先：「の」+「ど」+「か」+「な」+「風景」+「を」+「見た」

※「最長一致法」を優先すると、正解が出る。

※禁則（文頭で助詞なし）があれば、「のど+かな+風景+を+見た」が候補になる

＜例3＞

例文：「悪いのはのどかな。」

[redacted]

---

9 PAT(品詞列)とは、Deep Learning でいう filter と同じような使い方をする（N-gram）

10 Conceptual Array Tag は CT 列で、通常の PAT は BP 列である

11 条件付き確率分布 CRF の行列を創っておく

比較助詞「の」の次に未知語「ど」が出たら Lattice で「$\boxed{の}$+ど+$\boxed{かな}$」と「のど+$\boxed{かな}$」のビタビで検証。
比較助詞「の」の次は「名詞」がビタビ最尤値だから P(ど,名詞)=低値だから棄却される。
この結果は、最長一致法の中に助詞法が内包されていることを示唆している。
従って、アルゴリズムとしては、

（1）　RTB に登録した形態素が複数ある場合、

[redacted]

（2）　RTB に登録した形態素がない場合、

① 「助詞」を探して、その前までを未知語（名詞）とする。
② 文頭に「助詞がある場合、無視して次の「助詞」を探す。

☞{上記例2}前者は、（形容詞（形容動詞）+名詞+格助詞+動詞）であり、
後者は、（助詞+「未知語」+格助詞+「未知語」+名詞+格助詞+動詞）となり、
これも「未知語」が含まれる方が不採用。

◇手法のまとめ：

最長一致法と助詞検出法を同時に行い、ラテス構造化し、PAT で最尤推定をする。

① 最長一致法で分かち書き候補を抽出する。
② その候補の中に助詞候補が入っていなければ、PAT 最尤推定で品詞確定する。
③ その候補の中に助詞候補が入っていれば、助詞検出法でラテス構造化する。
④ そして、ラテス構造候補を PAT 最尤推定で品詞確定する。

[redacted]

（2）構文解析 Syntax（Parser）

1）文節統合処理：PhraseModule（RM）
2）連文節統合処理：ClauseModule（CM）
3）係り受け処理：DependenceModule（DM）

連文節は五種類

・主部（主語を含む連文節）
・述部（述語を含む連文節）
・修飾部（目的語を含む連文節）

・独立部（感嘆語を含む連文節）

・接続部（接続詞を含む連文節）

係り受け三原則

・前から後ろに係る

・逆向きはない

・交叉はしない

[redacted]

・主述関係は、主部と述部で行う　　SP 処理は文節内

（3）文脈解析：Context

①照応解析 Anaphora+Cataphora

・照応詞→先行詞特定処理（AM）　　RTA 参照

・0 照応詞→先行詞特定処理（ZM）　　XML 参照

※RTA はデータ項を参照　　P(代名詞+P,先行詞 V)

※SCX の項を参照

（4）意味解析：Semantic analysis

①意味タグ付与（文節間関係子）　　RTS 参照

※RTS はデータ項を参照　　P(関係子,STvector)

3. KNOWLEDGE_ENGINE

1）GENERATOR　　知識 KE 生成器

・SKG：文 Sentence:S→KE　　Encoder

・KSG=SKG^{-1}：KE→文 Sentence:S^*　　Decoder

KE: KnowledgeExpression

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
					[redacted]		[redacted]	[redacted]

$E(S, S^*)=\Sigma\|S-S^*\|^2+\beta\Sigma KL(P\|P^*_j)$ Sparse 正規化学習

Where, $P^*_j=1/N\Sigma f(W_j x_n+b_j)$、$KL(P\|P^*_j)=P\ln(P/P^*_j)+(1-P)\ln(1-P/1-P^*_j)$

$E=-\Sigma(s_i\ln s^*_i+(1-s_i)\ln(1-s^*_i))$ 交差エントロピー学習

・UltraPredicateExpressionType(KE:KnowledgeExpression)→{KE}≡KS

KE≡Predicate(Subject,Aspect,Space,Human&Ancillary,Matter&Things,Reason&Cause,Means)

KE とは、[redacted]($\left|\Delta^{8-1}\right|=8$)という単体として定義される。

超述語式 KE の定義は、

KE≡Predicate(Subject,Aspect,Space,Human&Ancillary,Matter&Things,Reason&Cause,Means)

[redacted]

部品：知識システムをモジュール部品化

俯瞰：全体像と部分像を適切に表す事

　　その際、それぞれに実体(知識類義図)、モデル(KE 経路)による表現の両方がある。

連想：ある知識から、別の知識に辿りつけるか（アソシエーション）

関連：知識のネットワーク（関連）を表現出来ているか（俯瞰図と等価）

創造：検索分析から新しいシナリオの創造が出来るか（推論）

支援：個人別に異なる目的と観点で利用できるか（視点）

以上のすべてが実現されて初めて AI といえる。

2）INFERENCER 新 KE 生成

3）LEARNER KE 構造構築

4．DIALOGER—InteractiveShell+Editor

1）データ表示編集 SCX データ

2）木構造 XML 係り受け関係

・$p(P|S,M)=p(P|S)p(S)p(P|M)p(M)\propto \mathrm{Max}\,p(S|P)p(M|P)$ Bayesian

3）InteractiveShell 対話型学習器

4）Paraphrase CT 類似変換

・概念ベクトル同士の余弦定理などでの類似度判定

5．DATA—SCX＋RTX

1）SemanticConceptualX(Ten)

（1）単語概念多様体：SCM（SemanticConceptualManifold）

①ジャンル G 別単語概念多様体 SCMM　　[自立語：自立語]

生成：ジャンル別単語文行列 M[自立語：文]→M◦M^T＝SCMM

[redacted]

【理論】

□文書（Corpus：新聞記事等）の集合から作られる[単語：文]行列 $D^{n\times m}$ は、[単語:イデアル]行列 $A^{n\times k}$ と[文書:イデアル]行列 $B^{m\times k}$ に分割され、$D^{n\times m} = A\cdot B^T$ となることは行列の「ランク分割」として保証されている。ランク（階数） k とは、線形方程式系や線形変換の非退化を示すもので、非座標系のヒルベルト空間や Topology 空間ではイデアル数と同値である。以下、意味概念空間 SCM 内の各単語意味概念は、Hausdorff 系開集合で成り立っている Topology 空間といえるので、「イデアル:Ideal」概念という用語で統一する。

[redacted]

Where,

$$C_A = diag(c_{a1}^2,\cdots,c_{ak}^2) \in S_{++}^k$$ … 単語潜在パラメータ（対角共分散行列）

$$C_B = diag(c_{b1}^2,\cdots,c_{bk}^2) \in S_{++}^k$$ … 文書潜在パラメータ（対角共分散行列）

□モデルの定義ができたので、このモデルで求めるものは潜在パラメータ$C=\{C_A, C_B\}$と分散σ^2である。

最適解（凸集合の最小値）として求める為に、このモデルを局所劣モジュラ化することが必要になる。

□今では、パラメータの最適値を DeepLearning(DL)で求めることが流行っているが、このままだと大局的に非凸な関数である為、収束が発散するか極値に収束をしてしまうことになる。従って、テキストの DL は局所解として求めているのだが、その局所定義が曖昧でなかなか成果が出ていないのが現状である。IBM ワトソンも同様である。局所空間の接続から歪曲した大局空間を推定するには、大学院レベルの数学知識（情報幾何学）が必要となる。NLP はそこまで高度化されてきている。そこで新たな理論として最適解を求める為に、停留条件（上界の最小限）で収束させることを考えてみよう。

停留条件を設定する為に目的関数である自由エネルギーを定義する　　　　　　…イェンセンの一般化

$\hat{r} = \arg\min_r F(r)$　　　s.t　$r(A,B) = r_A(A) r_B(B)$　　　…　自由エネルギー最小化

[redacted]

$\propto \exp\left(-\frac{1}{2} tr(AC_A^{-1}A^T) - \frac{1}{2\sigma^2} \left\langle \left\| D - AB^T \right\|_{Fro}^2 \right\rangle_{r_B(B)} \right)$　　　…　正規項の省略

$\propto \exp\left(-\frac{1}{2} tr(AC_A^{-1}A^T + \frac{1}{\sigma^2} \left\langle -2D^T BA^T + AB^T BA^T \right\rangle_{r_B(B)}) \right)$　　　…　展開

$$\propto \exp\left(-\frac{1}{2}tr(A\left(\frac{\left\langle B^T B\right\rangle_{r_B(B)}}{\sigma^2}+C_A^{-1}\right)A^T-\frac{2D^T\left\langle B\right\rangle_{r_B(B)}A^T}{\sigma^2}\right) \quad \cdots \text{ 因子分解}$$

$$\propto \exp\left(-\frac{tr\left(A-\hat{A}\right)\Sigma_A^{-1}\left(A-\hat{A}\right)^T}{2}\right) \quad \cdots \text{ まとめ}$$

[redacted]

[redacted]

となることがわかる。

同様に $r_B(B)$ も同じになるので途中は省略し、結果だけを書くと

$$r_B(B) \propto \exp\left(-\frac{tr\left(B-\hat{B}\right)\Sigma_B^{-1}\left(B-\hat{B}\right)^T}{2}\right)$$

Where,

$$\hat{B}=\sigma^{-2}D^T\left\langle A\right\rangle_{r_A(A)}\hat{\Sigma}_B \quad 、 \quad \hat{\Sigma}_B=\sigma^2\left(\left\langle A^T A\right\rangle_{r_A(A)}+\sigma^2 C_B^{-1}\right)^{-1}$$

したがって、

$$r_B(B)=\prod_{m=1}^{M} Norm_K(\tilde{b}_m;\tilde{\hat{b}}_m,\hat{\Sigma}_B)=\frac{tr\left((B-\hat{B})\Sigma_B^{-1}(B-\hat{B})^T\right)}{(2\pi)^{MK/2}\,|\hat{\Sigma}_B|^{M/2}} \quad \cdots \text{ 文書相関分布}$$

□すなわち、モデル尤度の「条件付き共役性」で変分ベイズ事後分布 $r_A(A), r_B(B)$ と事前分布 $p(A|C_A), p(B|C_B)$ が同じガウス分布[12]になっていることに注目する。裏には A と B の独立性が隠れており、大局的には独立性が証明できない場合でも、局所的な独立性を定義することで局所劣モジュラを定義することができる。局所劣モジュラが定義できれば、これらを「貼り合わせ」ていけば大局的な独立性の近似同型となる。但し、この「貼り合わせ」は、Topology 上の位相多様体のコホモロジーの知識[13]が必要になってくる。依存性を考慮するなら p(A)+p(B) = 1 として定義しても可能である。

12 分布が非対称性で汎用性を求めるならディリクレ分布を使えばよい

13 Cohomology や homomorphism など

□ガウス分布と判っているので、1 次、2 次モーメントの式から事後分布の平均と分散 $(\hat{A},\hat{\Sigma}_A,\hat{B},\hat{\Sigma}_B)$ を求める

$$\langle A \rangle_{r_A(A)} = \hat{A} \qquad \langle A^T A \rangle_{r_A(A)} = \hat{A}^T \hat{A} + N\hat{\Sigma}_A$$ … 単語分布の期待値

$$\langle B \rangle_{r_B(B)} = \hat{B} \qquad \langle B^T B \rangle_{r_B(B)} = \hat{B}^T \hat{B} + N\hat{\Sigma}_B$$ … 文書分布の期待値

[redacted]

これで局所解が求まりました。

□上記の目的関数として定義した自由エネルギー $F(r)$ を $r_A(A)$ 及び $r_B(B)$ の汎関数としてではなく、

変分パラメータ $(\hat{A},\hat{\Sigma}_A,\hat{B},\hat{\Sigma}_B)$ の関数として陽に書くと（展開は省略して結果だけ書く）

[redacted]

$$\mathit{Given} \quad C_A, C_A \in D_{++}^K \qquad \sigma^2 \in \Re_{++}$$ …パラメータ

$$\min_{\hat{A},\hat{B},\hat{\Sigma}_A,\hat{\Sigma}_B} F \qquad s.t. \quad \hat{A} \in \Re^{N\times K}, \hat{B} \in \Re^{M\times M}, \hat{\Sigma}_A, \hat{\Sigma}_B \in S_{++}^K$$ …目的関数と制約条件

□従って、最終パラメータである事前共分散 C_A, C_B と σ^2 を偏微分と停留条件によって求める

$$\frac{\partial F}{\partial c_{a_k}^2} = \frac{M}{2}\left(c_{a_k}^{-2} - c_{a_k}^{-4} \left(\frac{\|\hat{a}_k\|^2}{M} + (\Sigma_A)_{k,k} \right) \right) \qquad \therefore \quad c_{a_k}^2 = \|\hat{a}_k\|^2 / M + (\hat{\Sigma}_A)_{k,k}$$

[redacted]

$$\therefore \quad \sigma^2 = \frac{\|D\|_{Fro}^2 - tr(2D^T \hat{A}\hat{B}^T) + tr\left((\hat{A}^T\hat{A} + M\hat{\Sigma}_A)(\hat{B}^T\hat{B} + N\hat{\Sigma}_B)\right)}{NM}$$

以上で、求めるパラメータと分散は求まった。この分散σ^2 は大局空間の曲率、捩率などを考慮したものであり、途中でラグランジュ未定乗数を入れて局所制限を加えていないのが特徴である。

□これまでの論理の展開は、潜在変数を含む多くの確率モデルに適用されている大サンプル極限での変分ベイズ自由エネルギーの振る舞いを評価する理論を使い、ベイズ事後分布への近似誤差評価及び超パラメータに関する変分ベイズ解の相転移現象で解明をする漸近理論を使って精製[14]をしてきた。また、非漸近理論を使った Topology 空間上の圏空間での層を使った意味概念空間構築と知識構造化である新理論の有意性は別紙にて説明します。

【追加】

□上記のモデルを視点を変えて固有値と特異値分解で考えてみよう。固有値問題(EigenValueProblem;eigen(＊))

$D^{n\times n}x^n = \lambda x^n$ や特異値分解(SingularValueDecomposition;SVD) $D^{n\times m} = U^{n\times k}\Sigma^{k\times k}V^{k\times m}$ の解である固有値 λ と固有ベクトル x 、特異値 Σ とユニタリ行列 U, V は、イデアルをスカラー値にしたものとベクトル化、行列化したものと等値である。多変量解析での因子分析や主成分分析は、これらのことを言い換えただけである。但し、上記の行列 D は正方行列、長方行列どちらでもよいのが特徴だ。

□上記行列 D から行列 D'[単語:単語]を生成した場合、単語の概念ベクトルのイデアルは「文書」になる。その「文書」は、意図があっての文書であるので、イデアルは「意図」になる。すなわち、「意図」に沿った単語が散りばめられている文書/文章をイデアルとして成り立っている概念ベクトルが表記されていることになる。この文書/文章を「無限：∞」に近く集めれば、意図分布の山である混合分布が大数の法則で一つの凸集合（劣モジュラ）になる。これらから単語概念ベクトルを抽出すれば、偏差の無い、ゴミの少ない単語概念ベクトル収束が保証される。

□もう少し代数幾何学や位相幾何学からみると、イデアルを体や環 k とし、有限個の k の元を係数とする多項式による方程式 $f_1(x_1,\cdots,x_n) = \cdots = f_m(x_1,\cdots,x_n) = 0$ の解(イデアル単語)の全体は、

[redacted]

の各点の近傍で多様体(Manifold)の構造をもち特異点のない曲線(Curve)になっているので、無限の文書/文章を学習更新していけば、劣モジュラになり、イデアルに収束することが証明される。

<補説：(解)とは、斉次方程式となり、常に一次独立な n 組の解をもち、任意の解はそれの線形結合となる。列を一次独立な n 組の解αとして、行をφ(α)として行列で表せる。一般解は $x = \varphi(\alpha)c$ で表せる。 c は定数ベクトル>

---

[14] 通常「ゴミ取り」とか「ノイズ取り」と言われる

by ふきや 8/7/2019

[redacted]

□高精度 Anaphora（Cata）Module

・目的関数　P(Antecedent ｜ Anaphor=CT,Predicate,Position,Topics）

・EnvironmentAgent（KE）　S＝主語＜人称代名詞＞　誰が

P＝φ

時系列＝日時　いつ

※空間認識＝位置情報　どこで(Position)

人間(集団…)＝＜人称代名詞＞　誰と

※主題/目的＝トピックス　何を（Topics）

原因＝(トピックスの)理由　何故

手段＝交通機関　どうやって

文書 D の KE 列を行列でストア→あらすじ、ダイジェスト→トピックの推移推定

・アルゴリズム　P(Antecedent ｜ Anaphor=CT,Predicate,Position,Topics）

①照応詞 CT（場所、事物、方角：ACT）と距離 CT（近称、中称、遠称、不定称：DCT）を付与する

②EnvironmentAgent（KE）行列を LookUp して、該当（空間認識、トピックス）を選択する

〇考察：

代名詞とは、前述を受けて「同じ語彙を繰り返す」ことによっての「文章の煩雑さ」を無くす為のものであり、照応詞の先行詞は前述文に存在する。前述文に先行詞がない照応詞は、常識やある集団の暗黙の了解などで先行詞が判り切っている場合であり、これも先行詞が存在する。従って、先行詞が存在しない照応詞は存在せず、お伽話に出てくる「あるところにおばあさんとおじいさんが…」のいう「ある（ところ）」は一般代名詞として BP1.7 に分類している。また、

簡易先行詞特定では、述語に共起する RTA 行列から先行詞を特定をしたが、これは文書の「あらすじ」を無視した述語と単語との共起関係だけで特定したもので、文書によっては精度は低くなる。やはり「あらすじ：Digest」を用いた先行詞の特定をするべきとした。「あらすじ」の要素を照応詞の属性 CT から推定して空間認識（位置情報）とトピックスに絞った。

□SCM の概念処理と知識処理

・概念処理

◦単語概念ベクトルで同義/類義/関連語を抽出＜ベクトルの分布と分散で可能＞

但し、SCM だけでは関連語⊇類義語までしか抽出できない。類義語の中に同義語は含まれているので、その類義語を含む文と類義性の高い文(主に SP 関係)を抽出することで、同義語を推定できる。

◦ベクトル加法性で、共起関係や階層関係の抽出

◦主題やトピックスの抽出

・知識処理

◦S(経済、景気)、S(経済、消費)、S(景気、消費)の類似度が明示

◦P(経済、景気)＝P(経済|景気)P(景気)=P(景気|経済)P(経済) 行列を参照

◦P(経済|景気,消費)＝P(経済|景気)×P(経済|消費)　　グラフィカルモデルを参照

◦cos θ ＝＜経済,景気＞/||経済||･||景気||　　余弦定理から類似度推定

◦上記の P(経済,景気)＝cos θ（経済,景気）　　検証

【技術】

この SCM 構築法は、パラレル・コンピュータ（GPU）などを使用することを前提にしている。精度を求め、日々更新を続けることで実用的精度を維持することが可能となる。特に、ここでは Corpus として新聞記事（2 紙、6 年分）を使う。但し、この Corpus は、異常値（外れ値、ゴミ）という隠れた語彙が散布されているので、これらを効率良く削除して、正規化されたイデアル（基、基底）を求めることが必要である。

①Corpus➡新聞記事２紙、6 年分+α

②分野別記事➡分野別に記事をまとめる

③TAGGER 処理➡XML 出力（S：文単位）

④SW(StopWords)処理➡XML－SW＝WE　＜KW 抽出＞

1．SW の定義（接頭辞、接尾辞…）に注意する。

2．例：「東京湾」は、接尾辞「湾」と「東京」を Chunking して「東京湾」(固有名詞)となるが、

WE では、（東京+湾）の形態素の結合にすること。

⑤データ行列➡$SM_g = [WE:S]$を構築する。但し、S は文単位、 g はジャンル。

⑥ジャンル別行列➡ $SM_g \bullet SM_g^T = SMT_g$　　g ジャンルの一記事の単語・単語行列

[redacted]

[redacted]

⑨ $SMT_g \equiv \frac{1}{N} SMT_g H_N SMT_g^T \equiv \frac{1}{N}\sum_{n=1}^{N}(we^{(n)} - \overline{w})(we^{(n)} - \overline{w})^T$

⑩正則化行列処理➡ $(SMT_g + \lambda I)^{-1} \bullet SMT_g = SCM_g$

λは、固有値。g ジャンルの単語・単語行列の逆行列

注： $SMT_g^T \neq SMT_g^{-1}$　転置行列と逆行列は違うので注意

[redacted]

[redacted]1,000 イデアル,ジャンル別行列⇒500 イデアル、語彙 300 イデアル抽出

⑬自律学習➡SCM⇔RTP⇔CT　　　SCM の∀語彙を RTP へ登録　　SCM の概念 V を CT へ登録

【STOP__WORDS】の定義

ジャンル別 TOP500

■取得する単語

BP=1 or 1.1、1.2、1.3、1.4、1.10

（名詞、普通名詞、空間名詞、固有名詞、サ変名詞）

■STOP　WORDS

BP=1.3、1.5、1.6、1.7、1.8、1.9

（時相名詞、指示代名詞、人称代名詞、一般代名詞、数詞、形式名詞）

BP=2 or 2.x　（動詞）

BP=3 or 3.x　（形容詞）

[redacted]

BP=10 or 10.x　（接尾辞）

BP=11 or 11.x　（特殊）

BP=12 or 12.x　（連語）

単語別 TOP300

■取得する単語

BP=1 or 1.1、1.2、1.3、1.4、1.10

（名詞、普通名詞、時相名詞、空間名詞、固有名詞、サ変名詞）

但し、接頭辞と接尾辞は Chunking された複合名詞にする

BP=2 or 2.1、2.2、2.3、2.5

（動詞、自動詞、他動詞、可能動詞、補助動詞、自他動詞）

BP=3 or 3.1、3.2、3.3

（形容詞、連体形、連用形、その他）

■STOP　WORDS

BP=1.5、1.6、1.7、1.8、1.9

（指示代名詞、人称代名詞、一般代名詞、数詞、形式名詞）

[redacted]

BP=10 or 10.x　（接尾辞）　単独の接尾辞は除外

BP=11 or 11.x　（特殊）

BP=12 or 12.x　（連語）

＜検証手法＞

①ジャンル別のコーパスを入手

$$Corpus(Doc) = \{d_1, \cdots, d_n\}、d = (c_j, d_i)$$ 但し、 $j = 1, \cdots, l$ genres

②コーパスから単語抽出→頻度情報

[redacted]

④余弦定理にて単語相関関係を補正→⑤を先に処理しても OK！

$$\cos\theta = \frac{(w_i, w_j)}{\|w_i\| \bullet \|w_j\|}$$ 但し、(w_i, w_j) は内積

⑤再度、余弦定理でクラスタリング→上記①サンプルがジャンル別になっている。

①のサンプルと余弦定理で求めたクラスタを比較→精度アップ

①がなくとも、余弦定理でクラスタ

余弦定理でのクラスリングは、小数点以下 3 桁か 1,000 倍して分散σ^2 から分類

◇サンプル偏差補正の為にディリクレ分布を使う。→全体頻度分布からの出現率補正

全単語 W～Dir（W｜α）

$$p(W) = Dir(W \mid \alpha) = \frac{\Gamma(\sum_i^n \alpha_i)}{\prod_i^n \Gamma(\alpha_i)} \prod_i^n w_i^{\alpha_i - 1}$$

但し、$\alpha = (\alpha_1, \cdots, \alpha_n)$ は頻度数、$W = \{w_1, w_2, \cdots, w_n\}$ は各単語の出現率

解説：出現率の頻度数乗は、k 面サイコロ生成器補正。ベイジアンを使うと、共役事前分布として、無限のμが無限の混合比nを生成すると考えると DP(α、p(μ))になる。p(μ)は、事前分布の基底測度 G0 で、k 値がバラつき。

＜多項分布法＞

□最近の文献や理論などは、Watson でも使われている「Bag of Words」（略称：BOW）という文書の単語（主に名詞や形容詞）だけを形態素解析器で抽出し、文書を「単語の集合」という塊（かたまり）で意味を解析するやり方。

→数学的背景は、集合論上での統計学と確率論

□その代表的な方法が、単語間の関係を無視して、個別にカウントするモデル、すなわち独立性を前提条件とした多項分布（Multinomial distribution）である。

□上記の「単語の集合」を M 次元ベクトル $x = (x_1, \cdots, x_M)$ とすると、各単語の出現確率は、$\theta_1, \cdots, \theta_M$ で表現できる。

□これを多項分布で書くと、

$$Multi(x \mid \theta) = \frac{(x_1 + \cdots + x_M)!}{x_1! x_2! \cdots x_M!} \theta_1^{x_1} \cdots \theta_M^{x_M} \qquad \sum_{i=1}^{M} \theta_i = 1$$

通常、$M = 2$ の時を 2 項分布といい、多項分布は、「頻度」を表現するため自然な分布になると言われているのは、

分布が $\theta_i^{x_i} * (x_i!)$ という積の形になり、各 x_i ごとに分離しているからである。

[redacted]

となり、θ^1 の第 i 成分も同様で、結局

$$\hat{\theta}_i^0 = \frac{\sum_{n \in D^0} x_i^{(n)}}{\sum_{j=1}^{M} \sum_{n \in D^0} x_j^{(n)}} \qquad \hat{\theta}_i^1 = \frac{\sum_{n \in D^1} x_i^{(n)}}{\sum_{j=1}^{M} \sum_{n \in D^1} x_j^{(n)}}$$

が得られる。これは、「たくさん現れた単語ほど出現確率が高く見積もられる」という直観的に納得できる式。

□ここで問題になるのが、一度も出現しなかった単語のパラメータが、ゼロになってしまい計算処理や精度に問題が生ずることである。そこで、スムージング（Smoothing）処理をします。

$$\hat{\theta}_i^y = \frac{N_i^y + \gamma}{|D^y| + M\gamma} \qquad \gamma > 0$$

という「ゲタ」をはく簡単なバックオフ・スムージング（Back-off smoothing）の式だが、グッド・チューリング（Enhanced Good-Turing estimation）を使うと、

$$P(w_2 \mid w_1) = \frac{C^*(w_1 w_2)}{C(w_1)} = \frac{C^*(w_1 w_2)}{C(w_1 w_2)} \frac{C(w_1 w_2)}{C(w_1)} = d_{C(w_1 w_2)} \frac{C(w_1 w_2)}{C(w_1)}$$

但し、$d_{C(w_1 w_2)}$ は、ディスカウント係数。

□グッド・チューリング推定値 $C^*(\bullet)$ は、

$$C^*(w_1 w_2) = r^* \approx (r+1)\frac{N_{r+1}}{N_r} \qquad \text{但し、} |w_1 w_2| = r$$

というように、真の頻度（出現回数） r^* が求められる。

r 回出現した単語列の数 N_r との「比」の積、すなわち頻度を分布として一つ多い頻度との比で補正をしている。

□ここで新たな問題として、最尤推定法はデータ D を完全に信用して処理したものなのだが、実際は、データ D は「話し半分」に聞く方が汎用性を高められるし、その方が自然である。

それがディリクレ分布（Dirichlet distribution）を用いる方法である。(これは賛否両論がある…独立性に疑問が残る)

[redacted]

[redacted]後分布（Posterior distribution）を求めることになる。

□最大事後確率推定とは、

$$\theta^* = \arg\max_{\theta} \ln\left[p(D|\theta)p(\theta)\right]$$

として最適パラメータ θ^* を求める。これを MAP 推定ともいう。

□上の多項分布とディリクレ分布を使ってスムージングの式を最大事後確率推定で導くと、

$$p(\theta^0, \theta^1) = Dir(\theta^1 | \alpha^1) Dir(\theta^0 | \alpha^0)$$

そして、対数尤度 L を

[redacted]

□そして、

$$\hat{\theta}_i^y = \frac{N_i^y + \alpha_i^y - 1}{|D^y| + \sum_{j=1}^{M}(\alpha_i^y - 1)}$$

というスムージング解が得られる。

□最後に経験分布（Empirical distribution）を説明します。

$$p_{emp}(x \mid D) = \frac{1}{N}\sum_{n=1}^{N}\delta(x - x^{(n)})$$

というディラックのデルタ関数を使う。

$$p(x') \times |V_M(\varepsilon, x')| \approx \int_{x \in V_M(\varepsilon, x')} p_{emp}(x \mid D)dx$$

という「確率密度」×「体積」≒「確率」の確率密度関数が定義できる。

右辺の積分を実行すると、

$$p(x') \approx \frac{k}{N\,|V_M(\varepsilon, x')|}$$

となる。k は領域 $V_M(\varepsilon, x')$ に含まれる D の要素数。

□ε-nearest neighbors alogorithm とは、

$$a(x') = -\ln p(x') = -\ln k + M \ln \varepsilon + C$$

で、最近傍要素だけに縮退する。

逆は、ｋ最近傍法という。

[redacted]

[redacted]

□これは、ゴミを含まないように、

$$\psi_1^{(n)}(A) = \sum_{i \in N^{(n)}} d_A^2(x^{(n)}, x^{(i)})$$

というように標的近傍（Target neighbor）を求め、

$$\psi_2^{(n)}(A) = \sum_{j \in N^{(n)}} \sum_{l=1}^{N} \mathrm{sgn}[y^{(l)} \neq y^{(n)}]\left[1 + d_A^2(x^{(n)}, x^{(j)}) - d_A^2(x^{(n)}, x^{(l)})\right]_+$$

とで、「条件が破られている度合」との式で、まとめると

$$\Psi(A)=\frac{1}{N}\sum_{n=1}^{N}\left[(1-\mu)\psi_1^{(n)}(A)+\mu\psi_2^{(n)}(A)\right]$$ → 最小化

と最適化問題が定義できる。

□勾配法による最適化も同様に、

$$A \leftarrow A-\eta\frac{\partial\Psi(A)}{\partial A}$$

[redacted]

$C^{(i,j)}=(e_i-e_j)(e_i-e_j)^T$ の M×M 行列である。

■結論として、

①今回のプロトタイピングは、低スペックのマシーンで並列処理を使わないことを前提としていることと、

②SCM は、単なる集合ではなく、単語間の関係を無視していない相関行列であること、

③各単語のベクトル基（ベクトル要素）が、数万以上になること、

などを考慮して、以下の超高速処理法にする。但し、精度は劣化するが表示系内は相当。

◇SCM の各要素は、頻度数を表示している。そのまま使用する。

◇単語間のベクトル要素間の積を求め、ゼロ以上かどうかで判別する。

$$dist(w_i,w_j)=-\ln p(cor(w_i,w_j))$$ $dist(*,*)$:= 距離関数

[redacted]

上の例だと、第 3 要素がヒットし、{+ 1} になる。

$$p(x)=\frac{cor(w_i,w_j)}{|w_{*l}|}$$ $p(x)$:= 確率関数

$\ln x$:= 自然対数　　$|w_{*l}|=\sum_l w_{*l}$:= 単語ベクトル要素総数

従って、

$$dist(w_i, w_j) = -\ln p(cor(w_i, w_j)) = \ln \frac{1}{p(cor(w_i, w_j))} = \ln \frac{|w_{*l}|}{cor(w_i, w_j)}$$

となるので、この式で表示する距離を求める。

また、相関関数から相関行列は、事前に作成しておくことで、より高速処理ができる。

□　上記 SCM を余弦定理により単語相関関係を補正した…ということは、一次独立でなく、一次従属であることを示している。すなわち、$w_k = k_1 w_1 + k_2 w_2 + \cdots + k_n w_n$ が成立することになる。行（列）の基本変形で零ベクトル（零行列：行列式）$\det(w_k) = 0$ になることと同等である。単語は分布 $w_k = \{w_1, \cdots, w_k\}$ だから、ベクトルとして $\vec{w}_k = (\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_n)$ ともかける。これは、Rank(SCM)が求められることになる。

[redacted]

k 項演算関数に拡張する。値域の分割は、k=(k,k-1,…,1)になり、

有界変動ボレル測度 $\int_{R^d} f(x) d\lambda(x) = \int_{R^d} \int_{R^d} f(x+y) d\mu(x) d\nu(y)$ で定義される。

□　意味タグ（作用子、操作関数）として、単語を意味タグで群化することによって、文の意味識別が容易になる。

□　行列 A の n 乗を求めたい場合、A^n とすると、要素の n 乗になるので、行列の n 乗を求めるためには、A%＊%A …%＊%A と n 個を並べることになるが、面倒なので、固有値問題を使って、eigen（A）を使う。

eigen（A）＄vectors　%＊%　t（eigen（A）＄vectors）

とすると、正則な行列ならば単位行列になるのだが、solve (A) がエラーなら、ならない。SCM も同じだ。その時は、上記 A'の solve（A'）で再度計算をすると、単位行列になる。数値を整数値にするには、zapsmall（ｘ）を使う。ライブラリをインストールして…。要するに、SCM で固有値や固有ベクトルを使うなら、固有ベクトルの二度逆行列を求めなければならない。

A＝eigen(A)$vectors %*% diag(eigen(A)$values) %*% t(eigen(A)＄vectors)

で、固有値と固有ベクトルで元の行列 A を復元してみたら、非正則行列は復元できず。

しかし、

A＝eigen(A)$vectors %*% diag(eigen(A)$values) %*% solve(eigen(A)＄vectors)

としたら、完全復元できた。

…ということは、固有ベクトルは、非対称行列だから、ユニタリー行列（直行行列）ではないが、正則行列であることが証明できた。しかし、行列 A[123 456 789]（３×３）は、元々正則行列ではなかった（逆行列がなかった）ので、

それを正則行列に直し、スカラー値（係数）で表示するのが eigen( )である。ある体上の同じサイズの正則行列の全体は一般線型群 GL と呼ばれる群をなし、線形代数群や行列群と呼ばれる代数群の一種。正則行列の行列式はゼロでない…ということは、ランク（階数）が重複しないことであるので、非正則行列の階数を整理する関数。試しに det(A)=0 [redacted]

```
> x
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> x %*% x
     [,1] [,2] [,3]
[1,]   30   66  102
[2,]   36   81  126
[3,]   42   96  150
> eigen(x)$vectors %*% diag((eigen(x)$values))^2 %*% solve(eigen(x)$vectors)
     [,1] [,2] [,3]
[1,]   30   66  102
[2,]   36   81  126
[3,]   42   96  150
```

[redacted]vd(x)$v)

は、同じ x =[123　456　789]に再現ができた。

※すなわち、SVD は完[redacted]正則行列。

```
> x
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> svd(x)$u %*% diag(svd(x)$d) %*% t(svd(x)$v)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> svd(x)$d[1:2]
[1] 16.84810  1.06837
> svd(x)$d
[1] 1.684810e+01 1.068370e+00 5.543107e-16
> svd(x)$u[1:3,1:2] %*% diag(svd(x)$d[1:2]) %*% t(svd(x)$v[1:3,1:2])
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

上記のように、特異[redacted]

```
> eigen(x)
$values
[1]  1.611684e+01 -1.116844e+00 -5.700691e-16

$vectors
           [,1]       [,2]       [,3]
[1,] -0.4645473 -0.8829060  0.4082483
[2,] -0.5707955 -0.2395204 -0.8164966
[3,] -0.6770438  0.4038651  0.4082483

> eigen(x)$vectors %*% diag(eigen(x)$values) %*% solve(eigen(x)$vectors)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> eigen(x)$vectors[1:3,1:2] %*% diag(eigen(x)$values[1:2]) %*% solve(eigen(x)$vectors[1:3,1:2])
 以下にエラー solve.default(eigen(x)$vectors[1:3, 1:2]) :
   'a' (3 x 2) は正方行列である必要があります
> eigen(x)$vectors[1:3,1:2] %*% diag(eigen(x)$values[1:2]) %*% t(eigen(x)$vectors[1:3,1:2])
         [,1]     [,2]     [,3]
[1,] 2.607477 4.037383 5.467290
[2,] 4.037383 5.186916 6.336449
[3,] 5.467290 6.336449 7.205607
```

このように、固有値分解では、有効固有値と固有値ベクトルを使って、元の行列に戻すことができない。理由は、最後の項をみれば判る通り、正方行列にならないので、逆行列が成立できない為である。代わりに転置行列で行っても、元の[redacted]う斉次方程式が一般的に成り立つのはご存じのとおりで、上記の 2×2 行列の固有多項式は、λ^2-tr(A)λ+det(A)が成り立つ。3×3 行列なら、λ^3-tr(A)λ^2+c2λ-det(A)と表すことができる。但し、c2 は、主小行列式の総和とする。

```
> y
     [,1] [,2] [,3] [,4]
[1,]    2    3    4    2
[2,]    1    3    5    6
[3,]    2    4    2    4
> svd(y)
$d
[1] 11.588767  2.321990  2.075772

$u
           [,1]       [,2]        [,3]
[1,] -0.4676791  0.4891831  0.73619031
[2,] -0.7134120 -0.7006360  0.01234927
[3,] -0.5218425  0.5194315 -0.67666189

$v
           [,1]       [,2]       [,3]
[1,] -0.2323332  0.5670106  0.0633047
[2,] -0.4858708  0.6216077 -0.2220999
[3,] -0.5592882 -0.2185991  0.7964188
[4,] -0.6301964 -0.4942847 -0.5589108

> prcomp(y)
Standard deviations:
[1] 2.196105e+00 1.475508e+00 2.886449e-16

Rotation:
             PC1         PC2        PC3
[1,]  0.25925147 -0.06494033 0.69565910
[2,]  0.09183949 -0.36663695 0.60405812
[3,] -0.44293046  0.79821422 0.38075507
[4,] -0.85332690 -0.47351258 0.07872601
```

□このように解（根）の２次方程式を創り、その解を求めるということを解空間上の関数といい、これはガロアなどが[redacted]る。これはスペクタクルとして無限に生成できるものである。また、K[α]=L として、L/K というガロア拡大体から位数を論じることにより、5 次方程式の解方程式は不可能であることを、このシンメトリックから証明している。

```
> zapsmall(t(svd(y)$u) %*% svd(y)$u)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
> zapsmall(t(svd(y)$v) %*% svd(y)$v)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
> svd(y)$u %*% diag(svd(y)$d) %*% t(svd(y)$v)
     [,1] [,2] [,3] [,4]
[1,]    2    3    4    2
[2,]    1    3    5    6
[3,]    2    4    2    4
```

```
> prcomp(y)
Standard deviations:
[1] 2.196105e+00 1.475508e+00 2.886449e-16

Rotation:
             PC1         PC2        PC3
[1,]  0.25925147 -0.06494033 0.69565910
[2,]  0.09183949 -0.36663695 0.60405812
[3,] -0.44293046  0.79821422 0.38075507
[4,] -0.85332690 -0.47351258 0.07872601
> zapsmall(t(prcomp(y)$rotation) %*% prcomp(y)$rotation)
    PC1 PC2 PC3
PC1   1   0   0
PC2   0   1   0
PC3   0   0   1
```

```
> x
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> eigen(x)
$values
[1]  1.611684e+01 -1.116844e+00 -5.700691e-16

$vectors
           [,1]       [,2]       [,3]
[1,] -0.4645473 -0.8829060  0.4082483
[2,] -0.5707955 -0.2395204 -0.8164966
[3,] -0.6770438  0.4038651  0.4082483

> eigen(x)$vectors %*% diag(eigen(x)$values) %*% t(eigen(x)$vectors)
         [,1]     [,2]     [,3]
[1,] 2.607477 4.037383 5.467290
[2,] 4.037383 5.186916 6.336449
[3,] 5.467290 6.336449 7.205607
> t(eigen(x)$vectors) %*% diag(eigen(x)$values) %*% eigen(x)$vectors
          [,1]      [,2]      [,3]
[1,]  3.114206  6.457657 -3.577078
[2,]  6.457657 12.499376 -6.027652
[3,] -3.577078 -6.027652  1.941578
> eigen(x)$vectors %*% diag(eigen(x)$values) %*% solve(eigen(x)$vectors)
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
> k
     [,1] [,2] [,3]
[1,]    3    1   -1
[2,]    1    0    2
[3,]    2    1   -1
> det(k)
[1] -2
> svd(k)
$d
[1] 4.1153381 2.2398472 0.2169732

$u
            [,1]        [,2]       [,3]
[1,] -0.80526029 -0.01675492 -0.5926847
[2,] -0.06746479 -0.99051962  0.1196636
[3,] -0.58907077  0.13634572  0.7964957

$v
           [,1]        [,2]       [,3]
[1,] -0.8896929 -0.34292204 -0.3014149
[2,] -0.3388132  0.05339239  0.9393374
[3,]  0.3060262 -0.93784526  0.1636894

> kk <- t(svd(k)$u) %*% k %*% svd(k)$u
> kk
           [,1]       [,2]        [,3]
[1,] 2.30056136  1.6141725  3.00630534
[2,] 1.84786634 -0.3919996 -1.20359528
[3,] 0.01799156 -0.1959406  0.09143824
> sum(diag(k))
[1] 2
> sum(diag(kk))
[1] 2
```

上記は[redacted]い、群や体の特徴である。

面倒なので、tr()関数も創った。

```
> tr <- function(x){
+ return(sum(diag(x)))
+ }
> tr(k)
[1] 2
> tr(kk)
[1] 5
> tr(l)
[1] 5
```

```
> k
     [,1] [,2] [,3]
[1,]    3    1   -1
[2,]    1    0    2
[3,]    2    1   -1
> l
     [,1] [,2] [,3]
[1,]    3    1   -1
[2,]    1    1    2
[3,]    2    1    1
> tr(svd(k)$u %*% k %*% t(svd(k)$u))
[1] 2
> tr(svd(l)$u %*% k %*% t(svd(l)$u))
[1] 2
> tr(svd(l)$u %*% l %*% t(svd(l)$u))
[1] 5
```

そして、

```
> k
     [,1] [,2] [,3]
[1,]    3    1   -1
[2,]    1    0    2
[3,]    2    1   -1
> zapsmall(eigen(k)$vectors %*% diag(eigen(k)$values)  %*% solve(eigen(k)$vectors))
     [,1] [,2] [,3]
[1,]    3    1   -1
[2,]    1    0    2
[3,]    2    1   -1
> zapsmall(eigen(k)$vectors %*% k  %*% solve(eigen(k)$vectors))
           [,1]       [,2]       [,3]
[1,]  1.5355648  1.9033852  0.4021551
[2,]  2.8808812 -0.7320077 -2.1915808
[3,] -0.8230144  0.6000610  1.1964428
> tr(eigen(k)$vectors %*% k  %*% solve(eigen(k)$vectors))
[1] 2
```

上記の SVD(k)と同じ母斑が Eigen(k)でも付いたが、 k の行列と内容が違う。任意のユニタリ行列で挟めば、同じ行列 k でも違った行列になるが、類（母斑）は同じ。

□ 行列 A の特異値分解は、下図の行列 B の固有値問題に帰着される。

[redacted]すか？これは B の X 軸での鏡映変換σx となり、固有値問題が主成分分析 prcomp( )と同じように回転によって特徴（固有）を強調するためのものであることが判る。

[redacted]単語の基底 b_k 個に関し、$\chi_k = 1s(b_k)$ とす[redacted]

	単位	回転	240°	鏡映χ1	χ2 軸	χ3 軸
G	E	C3	C3^2	σv	σv'	σv"
Gχ1	χ1	χ2	χ3	χ1	χ3	χ2
Gχ2	χ2	χ3	χ1	χ3	χ2	χ1
Gχ3	χ3	χ1	χ2	χ2	χ1	χ3
Gχ4	χ4	χ4	χ4	χ4	χ4	χ4

これを一般系にすると、すなわち G を A に変えると、

[redacted]

関数セット$\{\chi_1, \chi_2, \chi_3, \chi_4\}$を基底ベクトルのセット$\{e_1, e_2, e_3\}$

と同等に扱う。基底関数という。

この係数行列$\{A_{\mu\nu}\}$は、$\hat{A}\{\chi_1, \chi_2, \chi_3, \chi_4\} = \{\chi_1, \chi_2, \chi_3, \chi_4\}A$ とし、

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix}$$ となる。

すなわち、基底関数から基底関数への変換（移動）する空間対称操作 A を定義することが重要。ここで

$$\hat{A}(\sigma_v) = \begin{Bmatrix} \hat{A}\chi_1 = \chi_1(A_{11} = 1) \\ \hat{A}\chi_2 = \chi_3(A_{32} = 1) \\ \hat{A}\chi_3 = \chi_2(A_{23} = 1) \\ \hat{A}\chi_4 = \chi_4(A_{44} = 1) \end{Bmatrix}$$ となるので、

$$D(\sigma_v) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$ となる。

[redacted]行列が $D(C_i)$ になり、n 個のクラスタでも関数基底の個数による正方行列の模様、

by ふきや 8/7/2019

すなわ[redacted]である。

□単語概念 $w_i = (w_1, \cdots, w_n) \in W = \{w_1, \cdots, w_n\}$ には、必ず基底 $w_i = (b_1, \cdots b_k)$ が存在する。但し、$w_i \in b_i$ である。（一次独立性や直交性、正則性が証明できる基底であるので、SVD や PRCOMP、EIGEN などで抽出できる）。例をあげると、色、形状、味、匂い(臭い)、…などのことで、これらは独立して成立している概念である。そして、これらは基[redacted]、階層循環化されている。基底数（基底関数の数）は、それほど多くはないので、行列計算も速い。単語概念 $w_i \chi_i$ を考えると、

$$w_i(\chi_1, \cdots, \chi_k) = [\chi_i \cdots \chi_j] = (\chi_1, \cdots, \chi_k)\begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$ となる。

（中辺の基底関数の順[redacted]、二つの単語ベクトルとの関係をみる場合、内積 $\check{u} \bullet \check{v} = u_1^* v_1 + u_2^* v_2 + \cdots + u_n^* v_n = \sum_{i=1}^{n} u_i^* v_i$ となる。

単語ベクトルと同じように、単語既約表現行列 $D_{11}^{P}(W) = \{111-1-1-1\}$ なのとなっていた場合、内積をとればゼロにな[redacted]り受け）に関する概念処理では、基底（関数）を決めて表現行列とすれば、指標も類も直交性も安易に求まることになる。

□ちなみに、固有値問題の固有値別の固有ベクトルどうしを掛けることを内積と

```
> k
     [,1] [,2] [,3]
[1,]    3    1   -1
[2,]    1    0    2
[3,]    2    1   -1
> kk
     [,1] [,2] [,3]
[1,]    3   -1    1
[2,]    1    2    0
[3,]    2   -1    1
> kkk
     [,1] [,2] [,3]
[1,]    1    3   -1
[2,]    0    1    2
[3,]    1    2   -1
> det(k)
[1] -2
> det(kk)
[1] 2
> det(kkk)
[1] 2
```

いい、これはゼロになる。直交だから…従って、直交行列かどうかの証明に使用する。

```
> rei
     [,1] [,2]
[1,]    2    1
[2,]    1    2
> eigen(rei)
$values
[1] 3 1

$vectors
          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068

> t(eigen(rei)$vectors[,1]) %*% eigen(rei)$vectors[,2]
     [,1]
[1,]    0
```

□行列 A の[redacted]

n 次元空間 SCM 上の単語ベクトル X と Y の複合単語の近傍系を直積 X＊Y とすると、底空間へのファイバー束 f(X*Y)としての写像とし、それからの切断 g(x)を考えれば…良い。

□複合単語の近傍系を求めるもうひとつの法：MoorePenrose を使った正則でない行列の逆行列を使って求める。

```
> x
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> det(x)
[1] 0
> ginv(x)
           [,1]          [,2]       [,3]
[1,] -0.6388889 -5.555556e-02  0.5277778
[2,] -0.1666667 -5.551115e-17  0.1666667
[3,]  0.3055556  5.555556e-02 -0.1944444
> k
     [,1] [,2] [,3]
[1,]    3    1   -1
[2,]    1    0    2
[3,]    2    1   -1
> zapsmall(ginv(k))
     [,1] [,2] [,3]
[1,]  1.0  0.0 -1.0
[2,] -2.5  0.5  3.5
[3,] -0.5  0.5  0.5
> solve(k)
     [,1] [,2] [,3]
[1,]  1.0  0.0 -1.0
[2,] -2.5  0.5  3.5
[3,] -0.5  0.5  0.5
```

左記例は、正方行列だったが、そうでなくても…OK!ライブラリ MASS のインストールを忘れずに…。

□SCM 簡易構築法

形態素 m 次[redacted]

[redacted]

出力クラス $Y=(x_1,\cdots,x_k)\in\Re^k$

$X\bullet X^T=W\in\Re^{m\times m}$　いままでの[redacted]

[redacted]

$X\bullet H=\widetilde{X}\in\Re^{m\times n}$　共分散行列　→cov(X)　R なら…

但し、もう少し工夫して、文章長に於ける頻度数の調整をすること。

$x^j = \sum_i x_i / l \in \Re^m$ j 番目の文章長 l の正規化された頻度数にする。

[redacted]

$\hat{W} \bullet \hat{X} = \dot{X} \in \Re^{m \times n}$ 再々新入力

$\dot{X} \bullet \dot{X}^T = \dot{W} \in \Re^{m \times m}$ 再々SCM

…

どこまでやれば…？これは n 次元多項式である。

正準相関分析

$f \equiv \alpha^T x = \alpha_1 x_1 + \cdots + \alpha_m x_m \in \Re$ $\alpha, x \in \Re^m$

[redacted]

$$\sum fg = \frac{1}{n}\sum_{j=1}^{n} \alpha^T (x^j - \bar{x})(y^j - \bar{y})^T \beta = \alpha^T \left\{ \frac{1}{n}\sum_{j=1}^{n} (x^j - \bar{x})(y^j - \bar{y})^T \right\} \beta = \alpha^T \sum_{xy} \beta$$

標本共分散

{・}は、m 次元変数 x と l 次元変数 y の間の共分散行列に対応するもので、$\sum_{xy}$ と表す。

■対角化　→eigen(固有値)、svd(特異値)$v,u 等長変換(回転)ユニタリ、$Σ伸縮変換(長径、短径)=特異値

n×n 行列 A と n×n 非特異行列 Q（正則行列∍直交行列）で、$D = QAQ^{-1}$ という対角行列が存在する。

[redacted]

☆$W=(XX^T)^{-1}XY^T$ の導出は…SCM 生成時に出てくる

線形関数　$y=f(x)=w^T x$

評価関数　$E[w]=\sum_i \left\| y_i - f(x_i) \right\|^2$　勾配法

目的関数　$w=\arg\min_w E[w]=\arg\min_w \sum_i \left\| y_i - w^T x_i \right\|^2$　最小二乗法

[redacted]

$XY^T = XX^T W$　行列表示

$\therefore W=(XX^T)^{-1}XY^T$　行列と列行列

そして、最急降下法の学習

$\Delta W^K = -\eta \nabla_{W^K} E[W]$　多変数と K は層名、ηはステップ数

[redacted]

■GibbsSampling∈MarkovChainMonteCarlo：MCMC

劣モジュラ SCM の収束は、近似ではなく事後分布からのサンプリング法

[redacted]

[redacted]例数、 $z_d^{(s)}$ は s 番目にサンプリングした変数 Z_d の事例（例：Genre、Topics）

D は変数 Z_d のクラス数、δは Dirac'sDeltaFunction

また、関数 $f(z_1,\cdots,z_D)$ の期待値は、

[redacted]

周辺化してパラメータを減らすことで、より効率的に推定が可能になるが、この背景には有限体 FiniteField というキツイ条件が付加されるので要注意…なのだが、拡大体 K⊂L に存在する構造体を解析することも必要なので、K[X]に属する多項式の根は、既約多項式ならば、素イデアル、極大イデアル、整域、体（剰余環も体）の存在があることが前提になる。

文書集合 W とトピック集合 z の同時分布

$$p(W, z \mid \alpha, \beta) = p(z \mid \alpha)\, p(W \mid z, \beta)$$

[redacted]

$$
= \frac{\Gamma(\beta V)^K}{\Gamma(\beta)^{VK}} \prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(N_{kv} + \beta)}{\Gamma(N_k + \beta V)}
$$

Were, N_{kv} はトピック k が割り当てられた文書における語彙 v の出現回数

$N_k = \sum_{v=1}^{V} N_{kv}$ はトピック k が割り当てられた文書における総単語数

ここで GibbsSampling に必要な条件付き確率であるベイズの定理を用いて

$$
p(z_d = k \mid W, z_{\backslash d}, \alpha, \beta) \propto p(z_d = k \mid z_{\backslash d}, \alpha) p(w_d \mid W_{\backslash d}, z_d = k, z_{\backslash d}, \beta)
$$

[redacted]

右辺第二項は

$$
p(w_d \mid W_{\backslash d}, z_d = k, z_{\backslash d}, \beta) = \frac{p(W \mid z_d = k, z_{\backslash d}, \beta)}{p(W_{\backslash d} \mid z_{\backslash d}, \beta)}
$$

$$
= \frac{\dfrac{\prod_{v=1}^{V} \Gamma(N_{kv\backslash d} + N_{dv} + \beta)}{\Gamma(N_{k\backslash d} + N_d + \beta V)} \prod_{k' \neq k} \dfrac{\prod_{v=1}^{V} \Gamma(N_{k'v\backslash d} + \beta)}{\Gamma(N_{k'\backslash d} + \beta V)}}{\prod_{k'=1}^{K} \dfrac{\prod_{v=1}^{V} \Gamma(N_{k'v\backslash d} + \beta)}{\Gamma(N_{k'\backslash d} + \beta V)}} = \frac{\Gamma(N_{k\backslash d} + \beta V)}{\Gamma(N_{k\backslash d} + N_d + \beta V)} \prod_{v: N_{dv} > 0} \frac{\Gamma(N_{kv\backslash d} + N_{dv} + \beta)}{\Gamma(N_{kv\backslash d} + \beta)}
$$

従って

[redacted]

ている

第二項は、文書 d がトピック k に割り当てられたときの尤度（ポリヤ分布）であり、文書 d がトピック k に割り当てられた文書集合と似ているほど、トピック k になり易くなっている

CRF と同じ

混合ユニグラムモデルはハイパーパラメータ α, β は不動点反復法を用いて文書集合 W とトピック集合 z の同時確率 $p(W, z \mid \alpha, \beta)$ を最尤推定することができる

更新式は

$$\alpha^{new} = \alpha \frac{\sum_{k=1}^{K} \Psi(D_k + \alpha) - K\Psi(\alpha)}{K\Psi(D + \alpha K) - K\Psi(\alpha K)}$$

[redacted]

潜在変数 z をサンプリングし、データ W とサンプル z の同時確率 $p(W, z \mid \alpha)$ をパラメータ α に関し最大化することを繰り返す➡確率的 EM アルゴリズム StochasticEMalgorithm

普通の EM は

①E ステップで潜在変数の帰属確率である負担率を計算し、　＜期待値＞

②M ステップで負担率の期待値をとって同時確率を最大化する　＜最大値＞

この確率的 EM は

①E ステップで潜在変数をサンプリングし、→マルコフ連鎖モンテカルロ法

②M ステップでサンプル潜在変数を用いて期待値をとった同時確率を最大化する

以上が SCM 構築法の理論と技術である。

※※※イデアルと汎用性と多義性と視点性の考え方※※※

例として SCM の汎用性は TF-IDF で求める頻度数も考慮すること。広義的汎用性

$\hat{w} \in [SCM]$ これは SCM の単語ベクトルなので、同義/類義/関連語などなど

$A \circ A^T$＝SCM なので、同義/類義語も記事単位内に存在する可能性大

$\overline{w} = [sentence]$ これは単語が入っている文の集合ベクトルでイデアルに近い

$\tilde{w} = \hat{w} - \overline{w}$ これは関連語や共起語など周辺語彙

$\overline{w} = \hat{w} - \tilde{w}$ SCM からイデアルを求めるのが目的

[redacted]

$\frac{\tilde{w}}{\overline{w}} = \frac{\hat{w}}{\overline{w}} - 1 =$ 拡張率　〇拡張率は、単語の視点性を示唆している

$\frac{\overline{w}}{\tilde{w}} = \frac{\hat{w}}{\tilde{w}} - 1 =$ 縮小率　〇

by ふきや 8/7/2019

以下はグラフィカルモデルができない

$$p(\overline{w} \mid \hat{w}) = \frac{p(\hat{w} \mid \overline{w})\, p(\overline{w})}{p(\hat{w})} = \frac{p(\hat{w} \mid \overline{w})\, p(\overline{w})}{\int p(\hat{w} \mid \overline{w})\, p(\overline{w}) d\overline{w}} \propto p(\hat{w} \mid \overline{w})\, p(\overline{w})$$

○SCM からイデアルを推定

[redacted]

[redacted]

後は上記①の処理へ　　分類分布は確率値 θ 処理

③総合単語行列 SCMT　　[自立語：自立語]

生成：$\Sigma_{\forall}$SCMM＝SCMT

用途：　IDF 値抽出（ジャンル判定）

④G 別文間行列 SCMS　　[文：文]

生成：$M^{T} \circ M$＝SCMS

用途：クラスタリング、文生成、

（2）G 別照応解析行列 SCA（Anaphora）　　[照応詞：先行詞]

生成：TAGGER(D)=P(Anaphora_CT, P(Antecedent_CT))

用途：TAGGER の文脈解析（照応解析）

（3）G 別意味タグ行列 SCT（ST）　　[関係子：ST]

生成：Cluster(KE ｜ Predicate)

用途：TAGGER の意味解析（ST 自動生成）

※詳細は、別紙「ST 付与 TK_Module」を参照

（4）G 別目的解析行列 SCO（Object）[動詞：目的語]

生成：TAGGER(D)=P(Predicate,P(Object))

用途：要約、主題、目的語抽出

（5）G 別因果関係行列 SCC（Coherence）　　[動詞(結果)：先行詞(原因)]

生成：TAGGER(D)=P(Predicate,Coherence(Antecedent))

用途：TAGGER の意味解析（因果関係 ST 付与）

（6）五感定量測度行列 SCG（Senses）[連体修飾語：連体詞]、[連用修飾語：連用詞]

生成：TAGGER(D)=P(SubstantivesModifier,Substantives)

用途：感情 ST（笑い、泣き、怒り、悲しみ、…）

（7）品詞列行列 SCP（→PAT）　　[品詞 BP：品詞 BP]

生成：TAGGER(D)=P(BP｜BP)

用途：未知語 BP 推定

（8）主述行列 SCS（SP）　　[主語：述語]

生成：TAGGER(D)=P(S,P)

用途：照応解析（ゼロ代名詞）の主述推定

（9）文字列行列 SCC（Character）　　[文字：文字]

生成：TAGGER(D)=P(C｜C)

用途：未知語推定（Bi_gram,Tri_gram）

（10）行列 SC

生成：

用途：

2）RunTimeConceptualData

[redacted]

[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]

[redacted]

⑤CT　　形態素概念体系

⑥ST　　意味タグ(関係子)

※※※イデアルと汎用性と多義性と視点性の求め方※※※

汎用性は TF-IDF で求める頻度数も考慮すること。広義的汎用性

$\hat{w} \in [SCM]$　これは SCM の単語ベクトルなので、同義/類義/関連語などなど

$A \circ A^T$＝SCM なので、同義/類義語も記事単位内に存在する可能性大

$\overline{w} = [sentence]$ これは単語が入っている文の集合ベクトルでイデアルに近い

$\tilde{w} = \hat{w} - \overline{w}$ これは関連語や共起語など周辺語彙

$\overline{w} = \hat{w} - \tilde{w}$ SCM からイデアルを求めるのが目的

[redacted]

[redacted]

[redacted]

$\frac{\overline{w}}{\tilde{w}} = \frac{\hat{w}}{\tilde{w}} - 1 =$ 縮小率 〇

以下はグラフィカルモデルができない

$$p(\overline{w} \mid \hat{w}) = \frac{p(\hat{w} \mid \overline{w})\, p(\overline{w})}{p(\hat{w})} = \frac{p(\hat{w} \mid \overline{w})\, p(\overline{w})}{\int p(\hat{w} \mid \overline{w})\, p(\overline{w}) d\overline{w}} \propto p(\hat{w} \mid \overline{w})\, p(\overline{w})$$

〇SCM からイデアルを推定

$$\int p(\overline{w} \mid \hat{w}) d\hat{w} = p(\overline{w})$$

〇イデアルの SCM 周辺化

6. APPLICATION

1）Chatbot

・文書 D のクラス化で回答/返答へのナビ/ガイドが誘導される

・ナビゲーションは、回答へクラス誘導

・ガイドは、問い返しをしながら

[redacted]

・問い返しとは、「質問」に対する曖昧/複数性の質問

・聞き返しとは、挨拶/平文に対する確認文

2）SUMMARIZATION

①単語クラス行列 Σ[単語:クラス]←Corpus から単語文行列 M[単語:文]

・クラスベクトル C{0,1}→Σ・C=P(w)でクラスcの単語分布 P(w)が抽出できる

[redacted]

SCM の元データ行列を素性行列で、SCM の単語確率はユニグラム

P(p(w)|s)P(s)=P(s|p(w))P(p(w))→wは単語、s は文で SCM の素性行列

[redacted]

３）HyperSemanticWeb

・社内文書意味概念検索→文書へ ST 付与

・WebRelationalST 検索→Web へ ST 付与