カーネル法に基づく疎な言語表現のための共起尺度

横井 祥 [1,2]　小林 颯介 [3]　福水 健次 [4]　乾 健太郎 [1,2]
[1] 東北大学　[2] 理化学研究所 AIP センター　[3] 株式会社 Preferred Networks　[4] 統計数理研究所
{yokoi,inui}@ecei.tohoku.ac.jp　sosk@preferred.jp　fukumizu@ism.ac.jp

1 はじめに

ふたつの言語表現の間の共起の強さのモデル化は，自然言語処理における基本的タスクである．たとえばコロケーション獲得では，はじめにコーパスから bigram が集められ，とくに強く共起する bigram（たとえば “New York”）を見つけることが目的となる．あるいは対話の応答文選択では，はじめに〈発話文，応答文〉ペアの集合が正例として与えられ，発話文 x に対する最も適当な応答文 y—発話文 x との共起の強さがもっとも大きくなるような応答文 y—を候補集合から選択することが目的となる．いずれの場合も，はじめに言語表現のペアの集合 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ が与えられ，新しいペア (x, y) の共起の強さを計算・推定することになる．

自然言語処理では言語表現のペアの共起の強さをしばしば**自己相互情報量（PMI）**でモデル化する [2]．すなわち，与えられたペアの集合を適当な同時分布 p_{XY} からのサンプル

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p_{XY} \tag{1}$$

だと考えて，ペア (x, y) の共起の強さを

$$\mathrm{PMI}(x, y; X, Y) = \log \frac{p_{XY}(x, y)}{p_X(x) p_Y(y)} \tag{2}$$

で測る．与えられたデータ $\mathcal{D}$ から PMI を推定する方法をふたつ述べる．ひとつめは頻度による推定である：

$$\widehat{\mathrm{PMI}}_{\mathrm{MLE}}(x, y; \mathcal{D}) = \log \frac{n \cdot c(x, y)}{\sum_{y'} c(x, y') \sum_{x'} c(x', y)}. \tag{3}$$

ここで $c(x, y)$ はペア (x, y) の $\mathcal{D}$ 内の頻度を表す．(3)は計算が容易であり，コロケーション獲得[*1]など，特に単語同士の共起を測る際にしばしば用いられるが [2]，各 x や y が疎である場合（たとえば句や文である場合）は適用できない．PMI のふたつめの推定法は RNN を用いるものである．Li ら [9] は対話の応答文生成（各 x や y が文であり非常に疎な場合）に PMI を適用するため[*2]，$p(y)$ を RNN 言語モデル [12] で，$p(y|x)$ を条件付き RNN 言語モデル [13] でモデル化し，PMI を推定した：

$$\widehat{\mathrm{PMI}}_{\mathrm{RNN}}(x, y; \mathcal{D}) = \log \frac{\widehat{p}_{\mathrm{CondRNN}}(y|x)}{\widehat{p}_{\mathrm{RNN}}(y)}. \tag{4}$$

$\widehat{p}_{\mathrm{RNN}}(y)$ および $\widehat{p}_{\mathrm{CondRNN}}(y|x)$ を学習するコストは大きいが，(4)の推定法は文や句などの疎な言語表現に対しても適用できる．以上のように，頻度による PMI の推定(3)と RNN による PMI の推定(4)には計算コストと疎性への頑健性というトレードオフがある．我々が知る限り，疎な表現に適用できかつ高速に計算可能な共起の尺度は存在しない．

[*1] コロケーション獲得において単純に頻度 $c(x, y)$ の高い bigram を探すと，“of the” など高頻度な機能語のペアが上位に上がってきてしまい，連語を探す目的に合わない．

[*2] 対話の応答文生成において単純に条件付き確率 $\widehat{p}(y|x)$ の高い y を探すと，“I don’t know.” など，発話文 x の内容に関わらず用いることのできるありきたり回答（dull responses）が上位に上がってきてしまう．

表1: 提案尺度 Pointwise HSIC（PHSIC）の位置づけ．

確率変数間の依存性の尺度 X と Y の依存の度合い （p_{XY} と $p_X p_Y$ の異なりの度合い）	**サンプル間の共起の尺度** ペア (x, y) の 「X と Y の依存の度合い」への貢献度
相互情報量 $I(X, Y) = \mathrm{KL}[p_{XY} \| p_X p_Y]$ $= \mathop{\mathbf{E}}_{x,y}\left[\log \frac{p_{XY}(x, y)}{p(x)_X p_Y(y)}\right]$	自己相互情報量（PMI） $\mathrm{PMI}(x, y; X, Y)$ $= \log \frac{p(x, y)}{p(x)p(y)}$
HSIC $\mathrm{HSIC}(X, Y; k, \ell) = \|m_{XY} - m_X m_Y^\top\|_{\mathrm{HS}}^2$ $= \mathop{\mathbf{E}}_{(x,y)}\left[\mathop{\mathbf{E}}_{(x',y')}[\widetilde{k}(x, x')\widetilde{\ell}(y, y')]\right]$ $= \mathop{\mathbf{E}}_{(x,y)}\left[(\phi(x) - m_X)^\top C_{XY} (\psi(y) - m_Y)^\top\right]$	Pointwise HSIC（PHSIC） $\mathrm{PHSIC}(x, y; X, Y, k, \ell)$ $= \mathop{\mathbf{E}}_{(x',y')}[\widetilde{k}(x, x')\widetilde{\ell}(y, y')]$ $= (\phi(x) - m_X)^\top C_{XY} (\psi(y) - m_Y)^\top$

表2: 本稿で用いる記号．

$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	$\mathcal{X}$ 上の正定値カーネル（直感的には類似度関数）
$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$	$\mathcal{Y}$ 上の正定値カーネル
$\phi(x) := k(x, \cdot)$	k が定める特徴空間における $x \in \mathcal{X}$ の特徴ベクトル
$\psi(y) := \ell(y, \cdot)$	ℓ が定める特徴空間における $y \in \mathcal{Y}$ の特徴ベクトル
$\widetilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$	確率分布 X で中心化したカーネル k（$\widetilde{\ell}$ も同様） $\widetilde{k}(x, x') := k(x, x') - \mathbf{E}_{X'}[k(x, x')] - \mathbf{E}_X[k(x, x')] + \mathbf{E}_{X,X'}[k(x, x')]$

本稿では，句や文などの疎な言語表現に適用でき，しかも計算コストの小さい新しい共起尺度を提案する．2 節で提案尺度である **Pointwise HSIC（PHSIC）**の形式的な定義を与える．PMI が「ペア (x, y) の相互情報量への貢献度」であることに対応させ，PHSIC は「ペア (x, y) のカーネル法に基づく依存性尺度 HSIC への貢献度」として定義する (表 1)．3 節で PHSIC が疎性に頑健であることの直感的な説明を与える．PMI は新しいペアとすでに観測されたペアを完全一致で照合するが，PHSIC はカーネル関数でこれを平滑化した量と見ることができる．4 節で PHSIC の効率的な推定法を示す．5 節で対話の応答文選択の実験をおこない，PHSIC が既存尺度に比べ学習速度が約 50 倍高速（2 時間 30 分 ↦3 分）で，かつデータ数が少ないときにも予測精度の劣化の程度が小さいことを示す．

問題設定・記号　X, Y をそれぞれ $\mathcal{X}, \mathcal{Y}$ に値をとる確率変数とする．本稿では，入力として言語表現のペアの集合 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \overset{\text{i.i.d.}}{\sim} p_{XY}$ を受け取り，与えられたペア $(x, y) \in \mathcal{X} \times \mathcal{Y}$ の “共起の強さ” を測るタスクを扱う．たとえば対話の応答選択タスクがこれに該当する．また提案手法はカーネル法に基づくため，カーネル法に関連する記号をいくつか用意する (表 2)．

2 Pointwise HSIC

PMI は「ペア (x, y) の相互情報量への貢献度」と捉えることができる．これに対応させ，提案尺度 PHSIC は「ペア (x, y) の HSIC への貢献度」として定義する (表 1)．

2.1 確率変数間の依存性の尺度

確率変数 X と Y が独立であるとは，同時分布 p_{XY} と周辺分布の積 $p_X p_Y$ が等しいことを言う．ふたつの確率変数 X, Y がどの程度依存しているかは，なんらかの観点で「p_{XY} と $p_X p_Y$ がどの程度異なるか」を測れば良い．相互情報量や HSIC は，このようにして定められた確率変数間の依存性の尺度である．

2.2 相互情報量と自己相互情報量

確率変数 X と Y の**相互情報量**は次で定義される [4] ：

$$I(X,Y) = \mathrm{KL}[p_{XY} \| p_X p_Y]. \tag{5}$$

すなわち相互情報量は p_{XY} と $p_X p_Y$ の違いを KL ダイバージェンスの観点で測った依存性尺度である．

相互情報量は，同時分布 p_{XY} による期待値の形，つまりペア $(x,y) \in \mathcal{X} \times \mathcal{Y}$ 全体での総和の形でも表現することができる：

$$I(X,Y) = \mathop{\mathbf{E}}_{(x,y)}\left[\log \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}\right]. \tag{6}$$

網掛け部がまさに**自己相互情報量（PMI）**

$$\mathrm{PMI}(x,y;X,Y) = \log \frac{p(x,y)}{p(x)p(y)} \tag{7}$$

であることに注意すると，$\mathrm{PMI}(x,y)$ は「ペア (x,y) の，相互情報量 $I(X,Y)$ への貢献度」と見ることができる．

2.3 HSIC と PHSIC

HSIC[6] はカーネル法に基づく確率変数間の依存性尺度で，次で定義される：

$$\mathrm{HSIC}(X,Y;k,\ell) = \|m_{XY} - m_X m_Y^\top\|_{\mathrm{HS}}^2. \tag{8}$$

ここで $m_X := \mathop{\mathbf{E}}_{x}[\phi(x)]$ は，確率変数 X の特徴空間における期待値で，X のカーネル平均埋め込みと呼ばれる．すなわち HSIC は同時分布 p_{XY} と周辺分布の積 $p_X p_Y$ の違いをカーネル平均埋め込みの観点で測った量である．

HSIC も，相互情報量と同様に，同時分布 p_{XY} による期待値の形で表現することができる：

$$\mathrm{HSIC}(X,Y;k,\ell) = \mathop{\mathbf{E}}_{(x,y)}\left[\mathop{\mathbf{E}}_{(x',y')}[\widetilde{k}(x,x')\widetilde{\ell}(y,y')]\right] \tag{9}$$

$$= \mathop{\mathbf{E}}_{(x,y)}\left[(\phi(x)-m_X)^\top C_{XY}(\psi(y)-m_Y)^\top\right]. \tag{10}$$

C_{XY} は X と Y の特徴空間での相互共分散を表す：

$$C_{XY} := \mathop{\mathbf{E}}_{(x,y)}[(\phi(x)-\mathop{\mathbf{E}}_{x'}[\phi(x')])(\psi(y)-\mathop{\mathbf{E}}_{y'}[\psi(y')])^\top]. \tag{11}$$

(9)はデータ空間での表現であり，(10)は特徴空間での表現である[*3]．**Pointwise HSIC（PHSIC）**を(9)および(10)の網掛け部により定義する：

$$\mathrm{PHSIC}(x,y;X,Y,k,\ell) := \mathop{\mathbf{E}}_{(x',y')}[\widetilde{k}(x,x')\widetilde{\ell}(y,y')] \tag{12}$$

$$= (\phi(x)-m_X)^\top C_{XY}(\psi(y)-m_Y)^\top. \tag{13}$$

すなわち $\mathrm{PHSIC}(x,y)$ は「ペア (x,y) の，$\mathrm{HSIC}(X,Y)$ への貢献度」である．本節の内容をまとめると表 1の通り．

3 PMI と PHSIC の違い：カーネルによる平滑化

PHSIC がデータの疎性に頑健な理由として，PHSIC は「PMI をカーネルで平滑化した量」だと捉えられることを述べる．まず PMI の最尤推定量(3)は次のように書き換えられる：

$$\widehat{\mathrm{PMI}}_{\mathrm{MLE}}(x,y) = \log \frac{n \cdot \sum_{i=1}^n \mathbb{I}[x=x_i \wedge y=y_i]}{\sum_{i=1}^n \mathbb{I}[x=x_i]\sum_{i=1}^n \mathbb{I}[y=y_i]}. \tag{14}$$

ここで $\mathbb{I}[\mathrm{cond}]$ は指示関数を表し，条件 cond が満たされるときは 1 を，そうでないときは 0 を返す．(14)によれば，PMI の推定量は「与えられた (x,y) を観測済みの $(x_i,y_i) \in \mathcal{D}$ たちと順に照合し，(x,y) と (x_i,y_i) が完全に一致すれば加点，x 側もしくは y 側のどちらか一方のみが一致する場合は減点」を繰り返した量である (表 3，上側)．

一方，PHSIC(12)の最尤推定量は次の通り：

$$\widehat{\mathrm{PHSIC}}_{\mathrm{kernel}}(x,y) = \tfrac{1}{n}\sum_{i=1}^n \widetilde{\widetilde{k}}(x,x_i)\widetilde{\widetilde{\ell}}(y,y_i). \tag{15}$$

関数 $\widetilde{\widetilde{k}}(\cdot,\cdot)$，$\widetilde{\widetilde{\ell}}(\cdot,\cdot)$ はデータで中心化した類似度関数である．(15)によれば，PHSIC の推定量は「与えられた (x,y) を観測済みの $(x_i,y_i) \in \mathcal{D}$ たちと順に照合し，x 側と y 側がともに平均より似ている場合（$\widetilde{\widetilde{k}}(x,x_i) > 0$ かつ $\widetilde{\widetilde{\ell}}(y,y_i) > 0$ の場合）に加点[*4]，x 側と y 側のどちらか一方が似ていてもう一方が似ていない場合は減点」を繰り返した量である (表 3，下側)．

表3: PMI と PHSIC の「与えられた (x,y) と観測済みの $(x_i,y_i) \in \mathcal{D}$ たちとの照合の仕方」という観点での比較．

	加点	減点
PMI	(x,y) $\| \quad \|$ $\mathcal{D} = \{\ldots,(x_i,y_i),\ldots\}$	$(x,y) \quad (x,y)$ $\| \quad \nparallel$ $\{\ldots,(x_i,y_i),\ldots,(x_i,y_i),\ldots\}$
PHSIC	$(x,y) \quad (x,y)$ $\approx \quad \approx$ $\{\ldots,(x_i,y_i),\ldots,(x_i,y_i),\ldots\}$	$(x,y) \quad (x,y)$ $\approx \quad \nparallel$ $\{\ldots,(x_i,y_i),\ldots,(x_i,y_i),\ldots\}$

以上のように，PMI と PHSIC の最尤推定量を「与えられた (x,y) と観測済みの $(x_i,y_i) \in \mathcal{D}$ たちとの照合の仕方」という観点で比較すると，PMI は完全一致で照合する一方，PHSIC はこれをカーネル（類似度関数）を用いてスムージングしていることが分かる．PHSIC がデータの疎性に頑健であることがこの点からも期待できる．

3.1 既存の類似度関数の活用

また自然言語処理には正定値カーネルの形の類似度関数が潤沢に存在する．PHSIC はこれらの資産をそのまま活用できる．

- **単語ベクトル**の和（加法構成性 [11]）や**文の分散表現** [8] など，分布仮説に基づく文や句の分散表現が多数提案されており，学習済みのモデルも種々配布されている．これら同士のコサインが簡便で高精度な類似尺度として利用できる．
- SVM 時代に考案された**構造カーネル** [3] が多数存在する．
- 正定値カーネルの和や積も正定値カーネルであるため，これらの類似度関数を自在に**組合せ**ることができる．

4 PHSIC の推定法と計算量

特徴空間での推定：線形カーネルの場合　一般に，カーネル $k(\cdot,\cdot)$ が定める特徴ベクトル $\phi(\cdot)$ は未知ないし超高次元であり特徴空間での推定は難しい（これをデータ空間で代替するのがカーネル法の特徴のひとつであった）．ところが，昨今の自然言語処理で利用される句や文の類似度は，コサインを用いた**線形**

[*3] 一般にカーネル法では，データ空間におけるカーネル $k(x,x')$ で特徴空間における内積 $\phi(x)^\top\phi(x')$ を計算できることを用いて，データ空間側で統計処理をおこなう．

[*4] x 側と y 側がともに平均より似ていない場合，すなわち $\widetilde{\widetilde{k}}(x,x_i) < 0$ かつ $\widetilde{\widetilde{\ell}}(y,y_i) < 0$ の場合も PHSIC の推定値が増加する．

カーネルで与えられるケースが多々ある (3.1 節)．コサインとともに用いられる文 x のベクトル表現を $\boldsymbol{x} \in \mathbb{R}^d$ とすれば，これは特徴空間での表現 $\phi(x) = \boldsymbol{x}/\|\boldsymbol{x}\|_2$ が陽に与えられていることに等しい（$k(x,x') = \phi(x)^\top \phi(x') = \cos(\boldsymbol{x},\boldsymbol{x}')$）．このような場合，PHSIC を特徴空間側で推定することが可能である．

PHSIC の特徴空間での表現(13) の最尤推定量は次の通り：

$$\widehat{\mathrm{PHSIC}}_{\mathrm{RKHS}}(x,y) = (\phi(x) - \overline{\phi(x)})^\top \widehat{C}_{XY} (\psi(y) - \overline{\psi(y)}) \quad (16)$$

$$\widehat{C}_{XY} := \frac{1}{n}\sum_{i=1}^n \phi(x_i)\psi(y_i)^\top - \overline{\phi(x)}\,\overline{\psi(y)}^\top. \quad (17)$$

ただし $\overline{\phi(x)} = \frac{1}{n}\sum_{i=1}^n \phi(x_i)$ は特徴空間での標本平均．$\overline{\psi(y)}$ も同様．(16)の計算は行列計算に帰着されており，しかも計算量も小さい．学習時はベクトル $\overline{\phi(x)}, \overline{\psi(y)} \in \mathbb{R}^d$ の計算，行列 $\widehat{C}_{XY} \in \mathbb{R}^{d\times d}$ の計算が必要であり，時間計算量は $\mathcal{O}(nd^2)$，空間計算量は $\mathcal{O}(nd)$．予測時は $\phi(x), \psi(y) \in \mathbb{R}^d$ の構成および(16)の計算に $\mathcal{O}(d^2)$ の時間計算量を要する．

データ空間での推定：非線形カーネルの場合　木カーネル [3] 等の非線形カーネルを用いる場合は PHSIC をデータ空間で推定する必要がある．行列分解を介する簡単な工夫で，データ空間でも(16)と同等の計算量で推定が可能である．詳細は付録 A.

5　実験

PHSIC が疎なデータに適用でき (3 節) しかも計算量が小さい (4 節) ことを，対話の応答文選択タスクを通して大規模実データで検証する．提案尺度は RNN を用いた PMI の推定に比べ 50 倍程度高速に学習でき，また訓練データ数が少ない場合も予測精度の劣化が少ないことがわかった．

タスク　対話の応答文選択タスクをおこなう．訓練データとして〈発話文，応答文〉のペアの集合 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ が与えられ，評価データの発話文 x 毎に応答文 y を選択肢から選ぶ．

データ　インターネットリレーチャットにおける対話ペアを抽出したコーパスを用いる [10][*5]．訓練データの正例ペア数 n は約 50 万．検証・評価データの正例ペア数はそれぞれ約 2 万で，各発話文 x に対してコーパス全体から乱択で選択された負例 y' が 9 件ずつ割り当てられている（10 択問題となる）．

訓練データサイズの違いが学習時間・予測精度へ与える影響を確かめるため，訓練データ全体を用いる実験のほか，乱択した $\{10^2, 10^3, 10^4\}$ ペアを訓練データとした実験もおこなう．

計算機環境　CPU：Xeon E5-1650-v3 (3.5GHz, 6 Cores)，GPU：GTX 1080 (8GB)．

5.1　提案尺度：PHSIC

PHSIC は任意の正定値カーネルを用いることができる．今回は，もっとも簡単な類似尺度のひとつである「文を構成する単語ベクトルの和同士のコサイン」

$$k(x,x') = \cos(\textstyle\sum \mathrm{wordvecs}(x), \sum \mathrm{wordvecs}(x')) \quad (18)$$

を用いる．$\ell(\cdot,\cdot)$ も同様．単語ベクトルは，文字 n-gram を活用した単語ベクトル学習アルゴリズムである `fastText`[1] によりウェブクロールデータ[*6]から学習済みの公開資源[*7]を利用する．ベクトルは 300 次元，語彙サイズは 200 万．

[*5] https://github.com/rkadlec/ubuntu-ranking-dataset-creator
[*6] http://commoncrawl.org/
[*7] https://fasttext.cc/docs/en/english-vectors.html

表4: 各モデルの**学習時間**，単位は秒．各行はモデル，各列は訓練データサイズ n．RNN の各モデルに付記されているのは次元数および単語埋め込みの初期化法．PHSIC の計算時間は，内数として特徴関数の計算時間 (feat map) と行列計算時間 (mat calc) を付記した．

Models				10^2	10^3	10^4	5×10^5
RNN	300	random	$\widehat{p}(y)$	4.1	11.2	66.6	3131.2
			$\widehat{p}(y\|x)$	8.0	24.5	112.2	5172.4
	300	fastText	$\widehat{p}(y)$	2.6	6.4	40.1	2386.8
			$\widehat{p}(y\|x)$	5.7	12.4	75.6	4192.3
	1200	random	$\widehat{p}(y)$	5.4	19.1	88.9	5733.4
			$\widehat{p}(y\|x)$	12.4	29.0	142.4	12511.5
	1200	fastText	$\widehat{p}(y)$	4.5	12.0	67.5	5672.7
			$\widehat{p}(y\|x)$	19.1	29.4	145.6	9414.1
PHSIC	fastText		total	4.4E−2	3.7E−1	3.8	187.0
			(feat map)	3.6E−2	3.6E−1	3.8	184.4
			(mat calc)	8.1E−3	1.1E−2	5.0E−2	2.6

5.2　ベースライン尺度：RNN 言語モデル

訓練データで RNN 言語モデル [12] および条件付き RNN 言語モデル [13] を，1 層 LSTM[7]，フレームワークは `Chainer`[14] を用いて学習し，(1) 応答文の尤度 $\widehat{p}(y)$，(2) 応答文の条件付き確率 $\widehat{p}(y|x)$，および (3)PMI（$\widehat{p}(y|x)/\widehat{p}(y)$）をベースライン尺度として用いる．隠れ層の次元，および単語埋め込み層の初期化については，以下の設定の組合せで学習する：

- 隠れ層の次元：300，1200，
- 初期化：$[-0.1, 0.1]$ の一様ランダム，`fastText`．

ほか設定は以下の通り：

- 語彙：頻度 10（$n = 5\times10^5$），頻度 2（ほか）以上の語を利用，
- ドロップアウト率：0.1（300 次元），0.3（1200 次元），
- バッチサイズ：16（$n = 10^2$），64（$n = \{10^3, 10^4, 5\times10^5\}$），
- 最大エポック：30（$n = \{10^2, 10^3, 10^4\}$），5（$n = 5\times10^5$）[*8]．

5.3　予測精度の評価

PHSIC やベースライン尺度は，各ペア (x,y) に対してなんらかの共起スコア $\mathrm{score}(x,y)$ を与える．共起スコアの良さを，`Recall@{1,2,5}`，すなわち発話文 x に対する 10 個の応答文候補 y_i たちを $\mathrm{score}(x,y_i)$ の順に並べたときにトップ $\{1,2,5\}$ 個までに正解が含まれる確率で評価する．

5.4　実験結果：計算時間

計算時間に関する実験結果は表 4の通り．約 50 万の訓練データの学習に，既存尺度は，検証セットパープレキシティが最適となるモデル（1200 次元，`fastText`）で約 2 時間 30 分，PHSIC と条件を揃えた場合（300 次元，`fastText`）でも約 1 時間 10 分を要したのに対し，PHSIC の学習は約 3 分で完了した．なお，PHSIC の学習時間のほとんど全ては特徴ベクトルの計算（カーネルの計算）であり，行列計算に帰着させたパラメータ行列・ベクトルの計算は約 2.5 秒で完了している．

5.5　実験結果：予測精度

予測精度に関する実験結果は表 5の通り．約 50 万件の学習データをすべて使ったとき，PHSIC は，RNN を用いた PMI の推定量 4 種の大凡中間の値（既存手法とほぼ同等の予測精度）を示した．

一方，訓練データの数を減らしていくと，既存尺度の予測精

[*8] 最大エポック終了時まで，0.5 エポック（$n = 5\times10^5$）または 1 エポック（ほか）毎に検証セットパープレキシティを計測し，検証セットパープレキシティが最良のモデルでの予測精度，および（最大エポックまでに要した時間ではなく）当該モデルを獲得するまでの学習時間を報告する．

表5: 各モデルの**予測精度**（Recall@{1,2,5}）.

Models			10^2	10^3	10^4	5×10^5
Chance Level			.10,.20,.50	.10,.20,.50	.10,.20,.50	.10,.20,.50
$\widehat{p}_{\mathrm{RNN}}(y)$	1200	fastText	.09,.20,.51	.10,.20,.51	.10,.20,.51	.10,.20,.51
$\widehat{p}_{\mathrm{RNN}}(y\|x)$	1200	fastText	.10,.20,.51	.10,.21,.51	.11,.22,.52	.11,.22,.53
$\widehat{\mathrm{PMI}}_{\mathrm{RNN}}(x,y)$	300	random	.10,.20,.50	.10,.21,.50	.10,.20,.50	.11,.22,.54
		fastText	.10,.20,.50	.10,.21,.51	.14,.27,.58	.27,.46,.78
	1200	random	.10,.20,.50	.10,.20,.51	.11,.22,.53	.15,.30,.65
		fastText	.10,.20,.51	11,.21,.52	.12,.24,.55	.26,.44,.76
PHSIC	fastText		.16,.30,.61	.23,.39,.69	.26,.42,.72	.26,.42,.72

度は大幅に劣化するのに対し，PHSIC の予測精度の劣化スピードははるかに緩慢である．すなわち，訓練データや教師データの少ないタスクに対して，PHSIC は予測精度の意味でも有用であることが期待できる．

なお，これらの性質は，評価尺度を ROC-AUC および MRR（平均逆順位）にした場合でも全く同様の傾向が見られた．

6 おわりに

自然言語処理でデファクトとなっている共起尺度である PMI は疎なデータに適用しようとすると大きな学習時間が必要であった (1 節)．我々は Pointwise HSIC（PHSIC）というカーネル法に基づく新しい共起尺度を提案した (2 節)．PHSIC は，疎なデータに適用でき (3 節)，さらに行列計算に基づく効率的な推定が可能である (4 節)．対話の応答選択の実験の結果，PHSIC は既存尺度の約 50 倍高速に学習が完了し，しかも同等の予測精度を示した．さらに，データ数が少ない場合の予測精度の劣化の程度も小さいことが分かった (5 節)．

今後の研究方針を述べる．(1)(16)式を用いると，与えられた $\phi(x)$ に対するもっとも適切な $\psi(y)$ を（定数倍の不定性を除いて一意に）**生成**することができる．深層学習ベースの文のデコーダを用いれば $\psi(y)$（特徴ベクトル）から y（文）を復元できることが期待できるため，「強い共起を持つ文の高速生成」が実現する可能性がある．(2) 文の埋め込み手法の性能は，それを深層学習器の入力に用いてさらに学習をおこなった予測器を介して，外部タスクの精度で計られることが多い．すなわち予測器の性能と分散表現の性能が混ざった状態でしか評価をおこなうことができない．一方提案手法はノンパラメトリックなカーネル法であるため，予測器の性能を除いた状態で分散表現の性能を評価できる可能性がある．(3)PHSIC は幅広い類似度尺度を用いることができる (3.1 節)．今回はもっとも簡単な類似度尺度を用いたが，様々なタスクに対してタスクに見合った多様なカーネルを用いた実験も魅力的な今後の研究課題である．

謝辞 東京大学の胡緯華氏からの有益な助言に感謝します．本研究の一部は文部科学省科研費 15H01702 の支援を受けた．

参考文献

[1] Bojanowski, P. et al. “Enriching Word Vectors with Subword Information”. In: *TACL* 5 (2017), pp. 135–146.
[2] Church, K. W. and Hanks, P. “Word Association Norms, Mutual Information, and Lexicography”. In: *ACL*. 1989, pp. 76–83.
[3] Collins, M. and Duffy, N. “Convolution Kernels for Natural Language”. In: *NIPS*. 2002, pp. 625–632.
[4] Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. John Wiley & Sons, 2006.
[5] Fine, S. and Scheinberg, K. “Efficient SVM Training Using Low-Rank Kernel Representations”. In: *JMLR* 2.Dec (2001), pp. 243–264.
[6] Gretton, A. et al. “Measuring Statistical Dependence with Hilbert-Schmidt Norms”. In: *ALT*. 2005, pp. 63–77.
[7] Hochreiter, S. and Schmidhuber, J. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
[8] Kiros, R. et al. “Skip-Thought Vectors”. In: *NIPS*. 2015, pp. 3294–3302.
[9] Li, J. et al. “A Diversity-Promoting Objective Function for Neural Conversation Models”. In: *NAACL-HLT*. 2016, pp. 110–119.
[10] Lowe, R. et al. “The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems”. In: *SIGDIAL*. 2015, pp. 285–294.
[11] Mikolov, T. et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *NIPS*. 2013, pp. 3111–3119.
[12] Mikolov, T. et al. “Recurrent neural network based language model”. In: *Interspeech*. 2010, pp. 1045–1048.
[13] Sutskever, I., Vinyals, O., and Le, Q. V. “Sequence to Sequence Learning with Neural Networks”. In: *NIPS*. 2014, pp. 3104–3112.
[14] Tokui, S. et al. “Chainer: a Next-Generation Open Source Framework for Deep Learning”. In: *LearningSys*. 2015.

付録 A データ空間での PHSIC の推定

最尤推定 PHSIC のデータ空間での表現(12)の最尤推定量は，

$$\widehat{\mathrm{PHSIC}}_{\mathrm{kernel}}(x,y) = (\boldsymbol{k}-\tfrac{1}{n}K\mathbf{1})^\top(\tfrac{1}{n}H)(\boldsymbol{\ell}-\tfrac{1}{n}L\mathbf{1}). \tag{19}$$

ここで $\boldsymbol{k} := (k(x,x_1),\dots,k(x,x_n))^\top \in \mathbb{R}^n$，$\boldsymbol{\ell}$ も同様．しかしこの推定方法は計算コストが大きい．学習時は縦ベクトル $K\mathbf{1}$ および $L\mathbf{1} \in \mathbb{R}^n$ の計算が必要で，時間計算量は $\mathcal{O}(n^2)$，空間計算量は $\mathcal{O}(n)$．予測時はベクトル $\boldsymbol{k},\boldsymbol{\ell} \in \mathbb{R}^n$ の構成および H を掛け合わせる計算に $\mathcal{O}(n)$ の時間計算量を要する．

不完全コレスキー分解を介した推定 推定量(19)の計算コストが大きかったのは，グラム行列 K,L の構成が必要だったからであった．カーネル法ではグラム行列を陽に構成せずにその近似を低コストで計算する方法がいくつか提案されており，不完全コレスキー分解 [5] もそのひとつである．不完全コレスキー分解により，$\{x_1,\dots,x_n\} \subseteq \mathcal{X}$ および正定値カーネル $k\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ から，行列 $A = (\boldsymbol{a}_1,\dots,\boldsymbol{a}_n)^\top \in \mathbb{R}^{n\times d}$ $(d \ll n)$ を時間計算量 $\mathcal{O}(nd^2)$ で得ることができる．これにより，グラム行列 K 全体を構成することなく，$\boldsymbol{a}_i \in \mathbb{R}^d$ たちによってカーネルを近似的に計算できるようになる（$\boldsymbol{a}_i^\top\boldsymbol{a}_j \approx k(x_i,x_j)$，$AA^\top \approx K$ ）．

HSIC にも不完全コレスキー分解を介した推定法が提案されている [6]：

$$\widehat{\mathrm{HSIC}}_{\mathrm{ICD}}(X,Y) = \tfrac{1}{n^2}\|(HA)^\top B\|_{\mathrm{F}} \tag{20}$$

ただし $A = (\boldsymbol{a}_1,\dots,\boldsymbol{a}_n)^\top \in \mathbb{R}^{n\times d}$ は不完全コレスキー分解により計算した $AA^\top \approx K$ なる行列であり，B も同様（$BB^\top \approx L$）．(20)はデータによる平均の形でも表現することができる：

$$\widehat{\mathrm{HSIC}}_{\mathrm{ICD}}(X,Y) = \tfrac{1}{n}\textstyle\sum_{i=1}^n \Big[(\boldsymbol{a}_i-\overline{\boldsymbol{a}})^\top \widehat{C}_{\mathrm{ICD}}(\boldsymbol{b}_i-\overline{\boldsymbol{b}}) \Big]. \tag{21}$$

ただし $\widehat{C}_{\mathrm{ICD}} := \frac{1}{n}(HA)^\top B$．また $\overline{\boldsymbol{a}}$ は $\{\boldsymbol{a}_i\}_{i=1}^n$ の平均，$\overline{\boldsymbol{b}}$ も同様：

$$\overline{\boldsymbol{a}} := \tfrac{1}{n}A^\top\mathbf{1} = \tfrac{1}{n}\textstyle\sum_{i=1}^n \boldsymbol{a}_i. \tag{22}$$

PHSIC(x,y) はペア (x,y) の HSIC(X,Y) への貢献度であったので，PHSIC は(21)の網掛け部により推定できる：

$$\widehat{\mathrm{PHSIC}}_{\mathrm{ICD}}(x,y) = (\boldsymbol{a}-\overline{\boldsymbol{a}})^\top \widehat{C}_{\mathrm{ICD}}(\boldsymbol{b}-\overline{\boldsymbol{b}}). \tag{23}$$

ここで新しい $x \in \mathcal{X}$ に対応するベクトル $\boldsymbol{a} \in \mathbb{R}^d$ は，不完全コレスキー分解のアルゴリズムを「途中から行う」ことで計算できる．分解中に採用された支配的な x_i たちを順に $x^{(1)},\dots,x^{(d)}$ とすると，$\boldsymbol{a}$ の第 j 成分 $\boldsymbol{a}(j)$ は以下で計算できる（$\boldsymbol{b}$ も同様）．

$$\boldsymbol{a}(j) = [k(x,x^{(j)}) - \textstyle\sum_{m=1}^{j-1}\boldsymbol{a}(m)A_{jm}]/A_{jj}. \tag{24}$$

不完全コレスキー分解を介した PHSIC の推定(23)はナイーブな推定(19)に比べて極めて効率的であり，計算量は線形カーネルを用いた特徴空間での推定(16)と同等．