統計・機械学習若手シンポジウム

自然言語処理と統計・機械学習

持橋大地
統計数理研究所
daichi@ism.ac.jp

2018-8-10(金) 一橋講堂

自己紹介

- 所属: 統計数理研究所 数理・推論研究系
- 専門: 統計的自然言語処理、ベイズ機械学習
- 略歴

  1998年 東大教養学部基礎科学科第二卒業
  2005年 奈良先端科学技術大学院大学
  情報科学研究科 博士後期課程修了
  (松本研究室)
  ATR音声研、NTT CS研を経て現職

自然言語処理とは

- 巨大な分野なので、本チュートリアルで全部カバーするのは絶対に無理
- 機械学習および統計に関連するところのお話

Take-home Message

- 自然言語処理の重要な技術は、ほぼすべて機械学習および統計から来ている
  - 自然言語処理の人で、数学を大事にする人が少ない
- 非常に深さのある分野なので、本チュートリアルを参考に、周りの自然言語処理の人と協力してみてほしい
- 社会応用に今後インパクトが大きい

今日の講演の概要

- 自然言語処理の歴史と統計・機械学習との関係
- 深層学習で何ができるようになったか?
- 深層学習で何ができないか?
- Remedy:
  - 深層学習の統計学習化
  - 統計学習の深層学習化
- 知識表現と統計・機械学習 (他の分野にない特徴)

NLP(自然言語処理)の歴史

- 1950年代～
  経験主義
- 1980年代～
  合理主義、論理アプローチ、知識表現
- 1990年代後半～
  統計・機械学習の本格的導入
- 2010年代～
  深層学習＋論理・知識の機械学習
  社会応用の広がり

NLPの技術

- 1990年代最後：最大エントロピー法 (＝対数線形モデル)
  - Sha+ (NAACL 2003)でMEが最適化で解けると発見
  - Boostingと最尤推定 (Lebanon 2002)
- 2000年代前半：カーネル法の導入
  - SVM以外のカーネル法はあまり流行らなかった
  - NB: カーネル法とガウス過程の関係 (Kanagawa+ 18)
- 2000年代後半：ベイズ学習
  - 変分ベイズ法、階層ベイズ、ノンパラメトリックベイズ (基礎は1990年代中盤から (Gelman 1995))

2010年頃の状況

- ノンパラメトリックベイズは数学が超高度化しており、一部の人しかついていけない状況
- NLPの論文のモデルは、ほとんどがロジスティック回帰
  - 論文は、

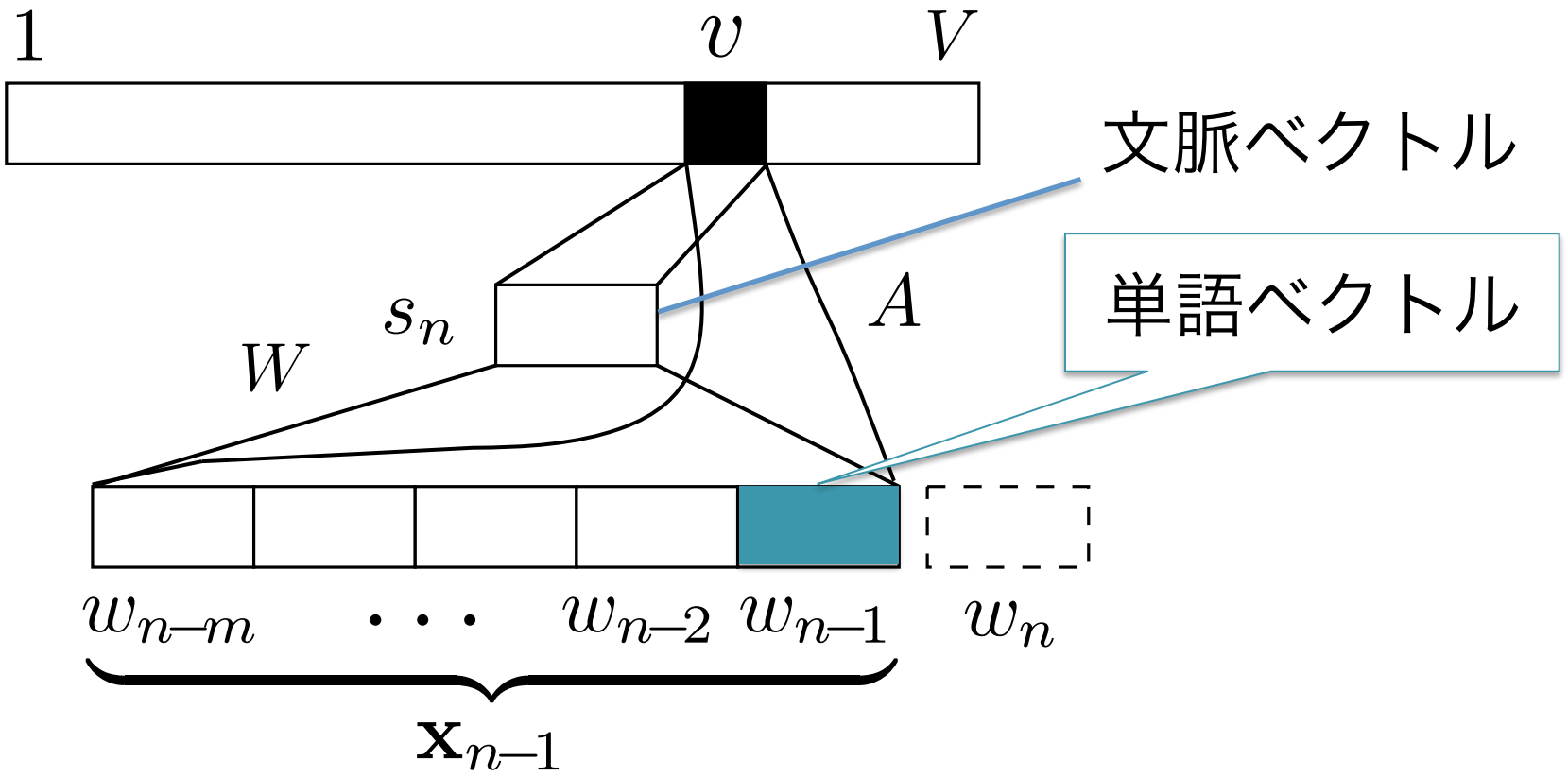
$$p(\mathbf{y}|\mathbf{x}) \propto \exp(\mathbf{w}^T f(\mathbf{x}, \mathbf{y}))$$

    の形の式しかほぼない状況
- 要するに、複雑な現実に対してモデルがノンパラメトリック化していく過程

深層学習の導入

- 2010年代以降〜現在
- そもそも、深層学習はNLPから始まった
- ニューラル言語モデル (Bengio+ 2003)
  - *n*グラムモデルより高性能なことが示された (ただし学習は難しい)

ニューラル言語モデル (Bengio+ 03)



- 前の単語列から、次の単語を予測する言語モデルを学習

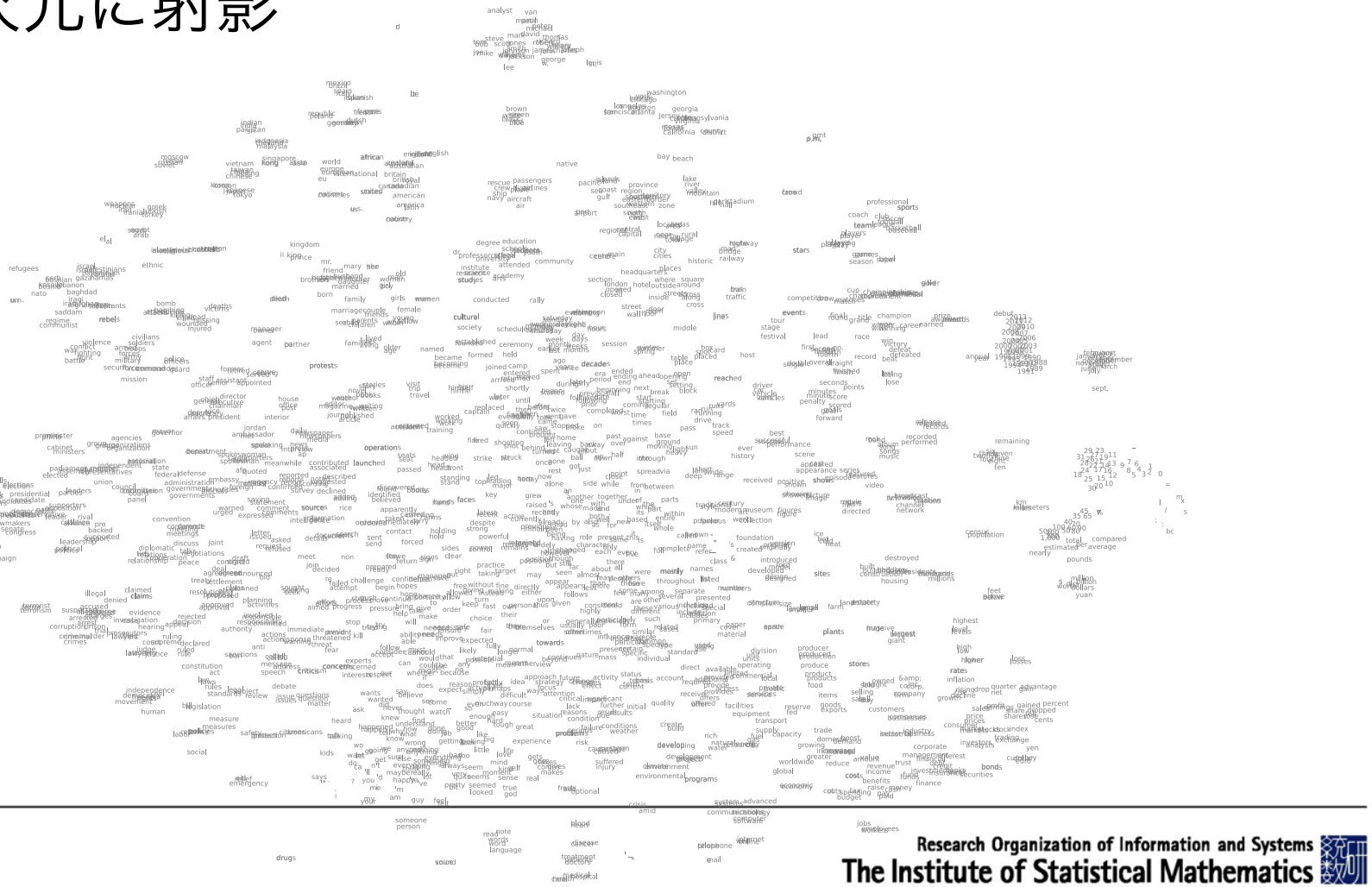
$$p(w_n|w_{n-1}\cdots w_{n-m})$$

Word2vec

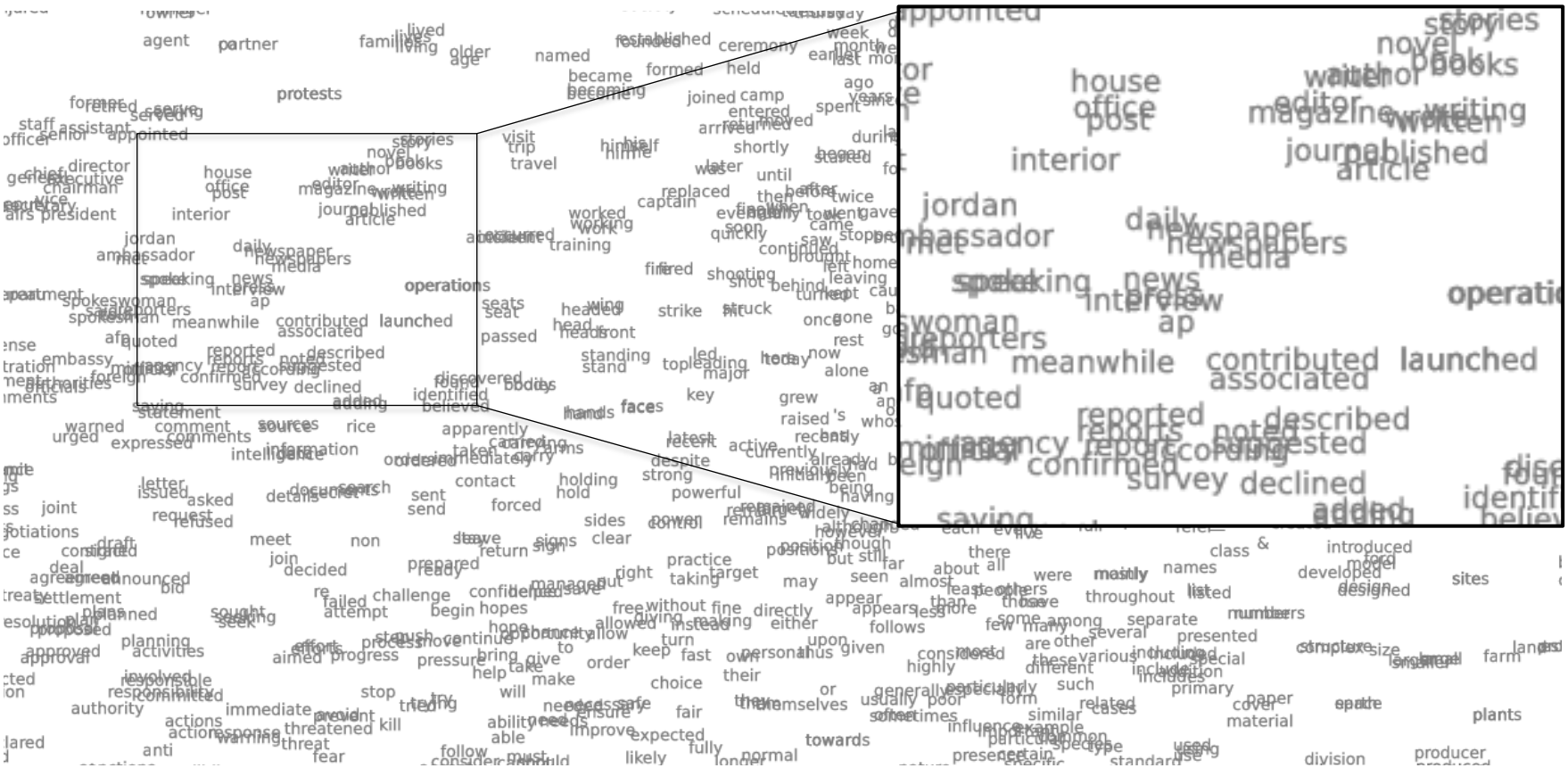
- 前の単語ではなく前後の単語を予測するようにすることで、単語の意味をよりよくベクトル化
- 有名なベクトルの引き算が成り立つ
- 統計的には、行列分解と同値

単語埋め込みの例

- GloVeの高次元の埋め込みベクトルをt-SNEで2次元に射影



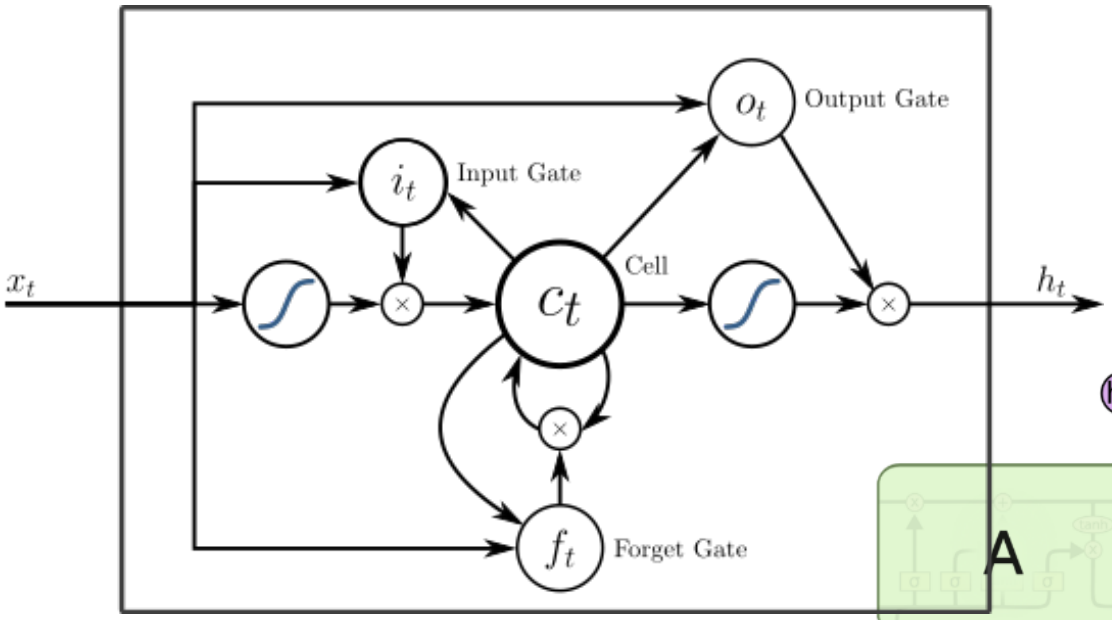
単語埋め込みの例 (2)



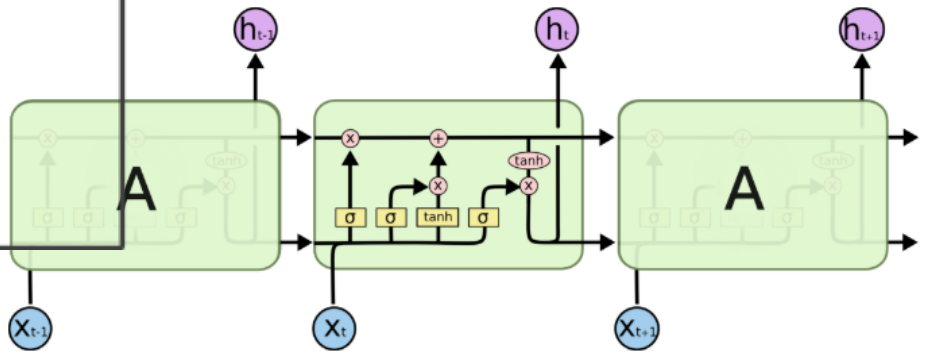
- 詳しくは→岩波データサイエンス第2巻
  「統計的自然言語処理ーことばを扱う機械」

LSTM

- Long Short-Term Memory (Hochreiter 1997)
  - 一種の複雑な再帰的ニューラルネットワーク



LSTMセルが連なって、時系列データを表現する



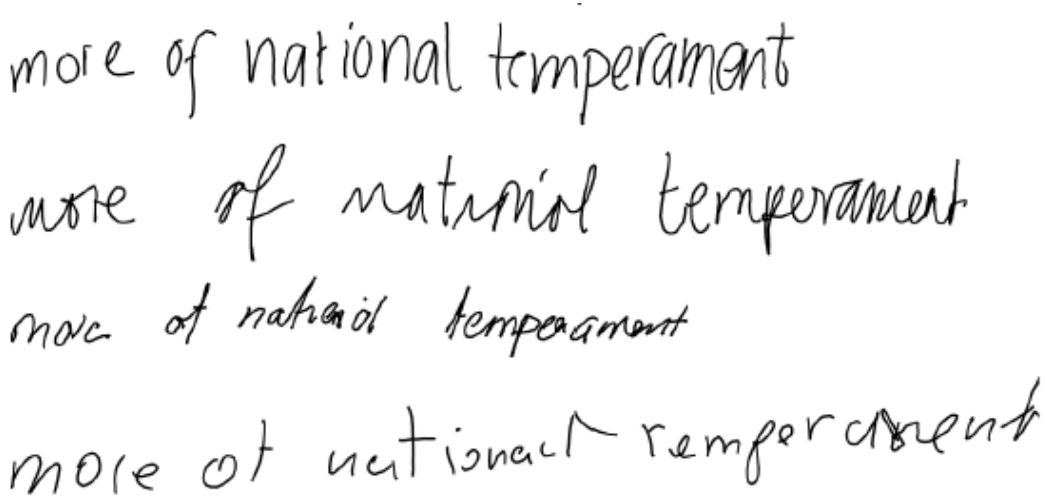
The repeating module in an LSTM contains four interacting layers.

LSTMによる生成例 (Graves 2014)

```
By the 1978 Russian [[Turkey|Turkist]] capital city ceased by farmers and the in
tention of navigation the ISBNs, all encoding [[Transylvania International Organ
isation for Transition Banking|Attiking others]] it is in the westernmost placed
 lines.  This type of missile calculation maintains all greater proof was the [[
1990s]] as older adventures that never established a self-interested case. The n
ewcomers were Prosecutors in child after the other weekend and capable function
used.

Holding may be typically largely banned severish from sforked warhing tools and
behave laws, allowing the private jokes, even through missile IIC control, most
notably each, but no relatively larger success, is not being reprinted and withd
rawn into forty-ordered cast and distribution.
```

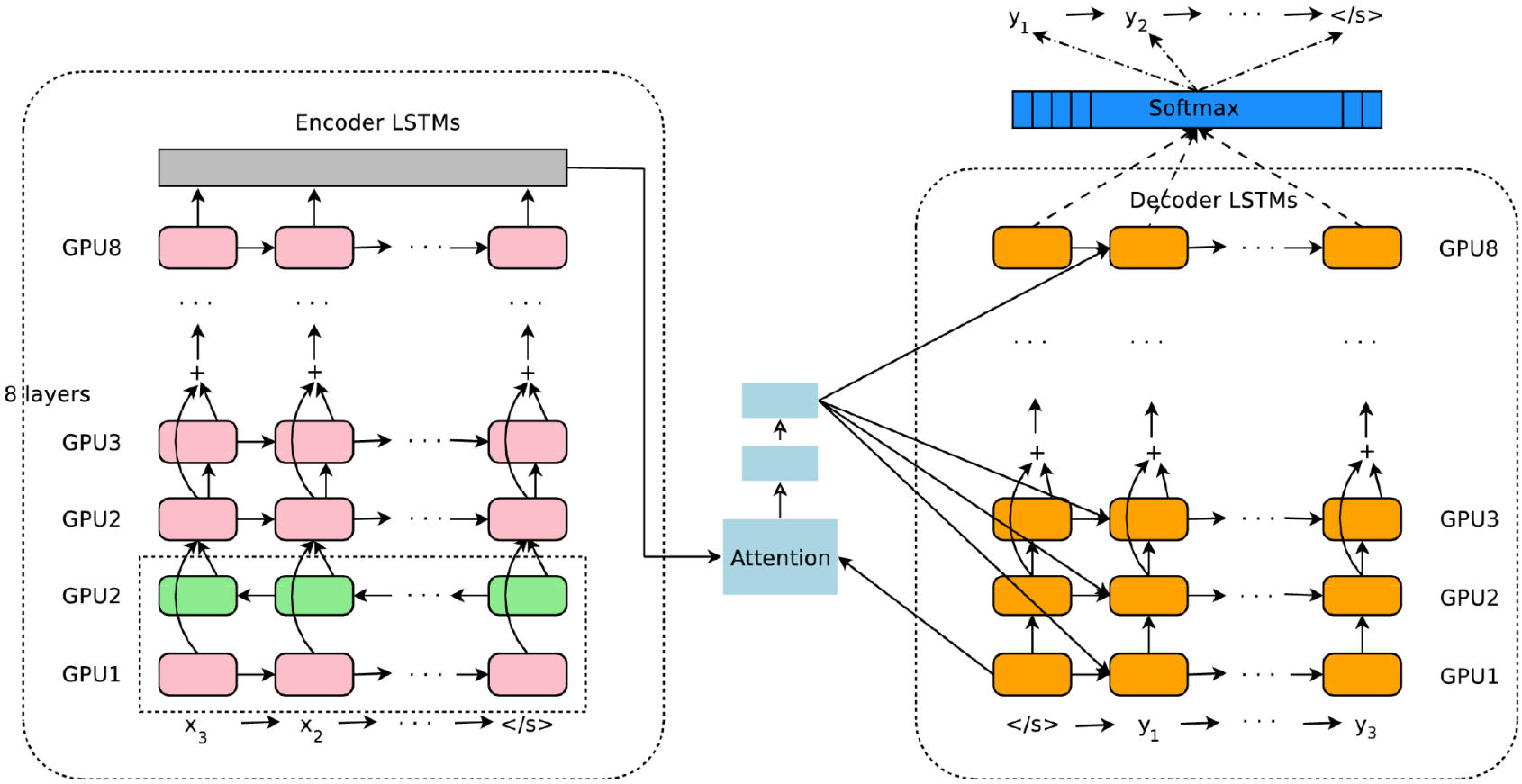
仮想Wikipediaデータ



手書き文字生成

ニューラル機械翻訳

- Google ニューラル機械翻訳: 8層の双方向LSTM＋アテンションのエンコーダ/デコーダ



ニューラル機械翻訳の例

- かなり上手く翻訳できる (The Guardianのヘッドライン)

ニューラル機械翻訳は完全か?

- No! (まったく理不尽な翻訳を出すことがある)

深層学習でできるようになったこと

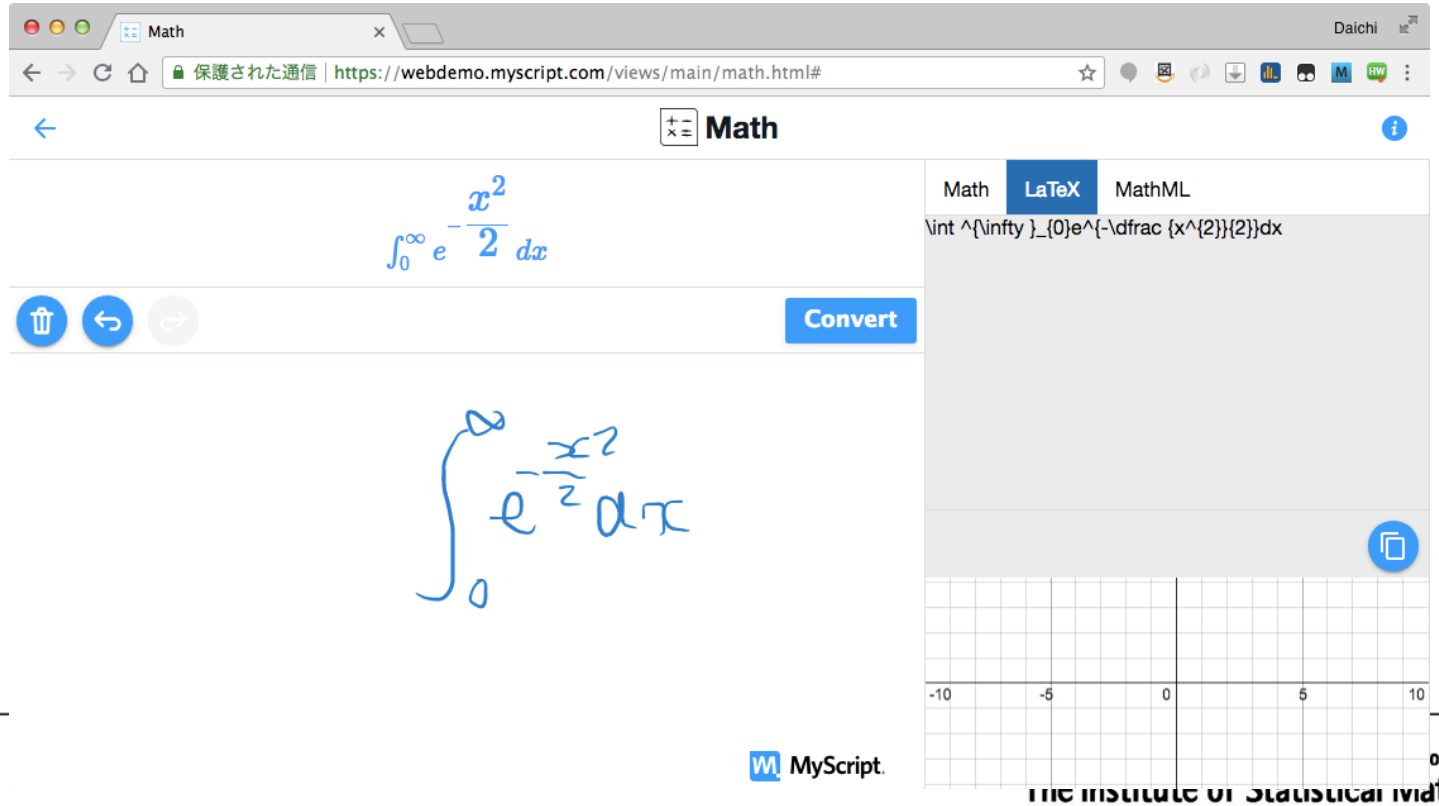
- 類義語の適切な扱い (word2vec, GloVe)
- 文生成 (LSTM)
- ニューラル機械翻訳
- その他、各モデルのMarginalな改善

逆に、これだけ！

- 画像処理や音声認識と比べると、差分は大きくない

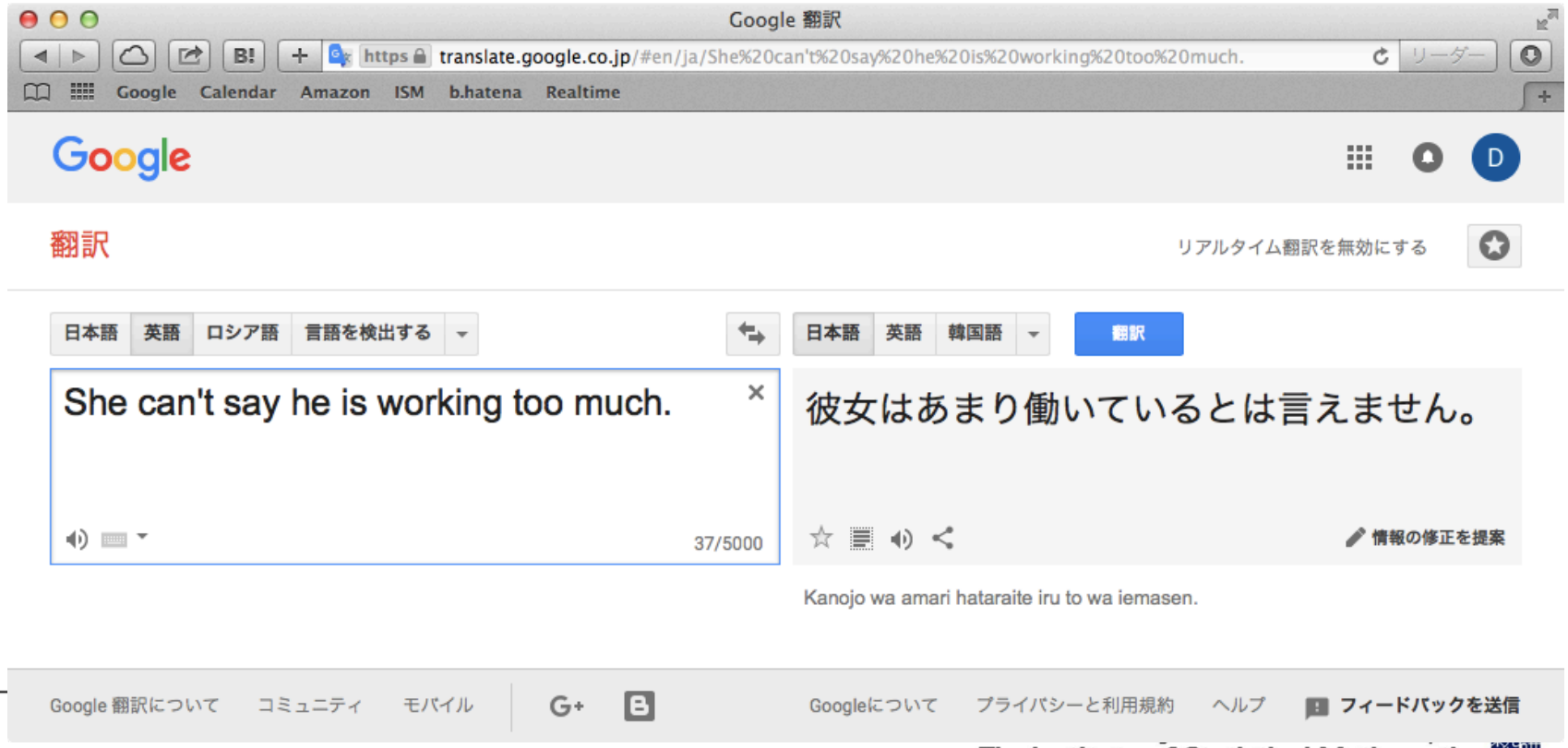
深層学習の利点と弱点

- 「理屈で説明できないようなもの」にNNは強い
  - 例：画像->LaTeX変換、顔画像生成
  - 大量のデータがあれば、経験分布で押し切れる



深層学習の利点と弱点 (2)

- 「理屈で説明できるもの」には弱い
  - 例：Google機械翻訳のnotと複文の扱い
  - 論理的関係はモデルに入っていない



深層学習の利点と弱点 (3)

- 他の自然言語の例：

  「彼は音楽を聞いたと果たして言えるのだろうか」
  「角笛の音が街にひびきわたるとき、獣たちは
  太古の記憶に向ってその首をあげる。」
  「Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.」

- 単純なニューラルネットがこれらの言語構造を
  勝手に学習してくれるわけがない！

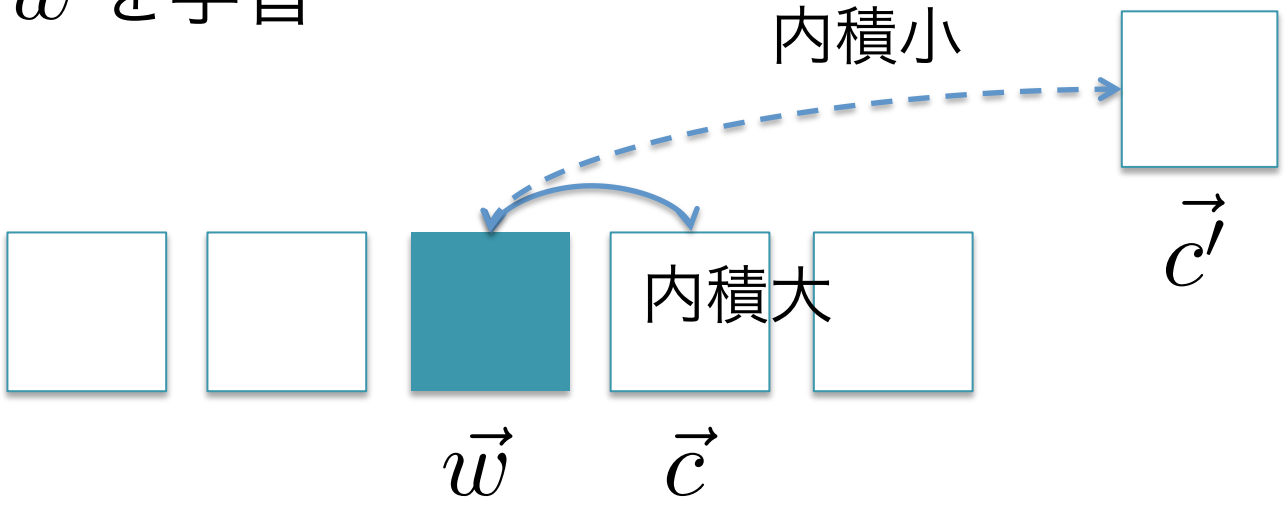
どうすればよい?

1. 深層学習を、確率モデルを使って記述する.
   (深層学習側からのアプローチ)
2. 確率モデルを、深層学習を使って推定する.
   (確率モデル側からのアプローチ)

1. 深層学習を、確率モデルを使って
   記述する.
(深層学習側からのアプローチ)

単語埋め込みの統計モデル

- “Neural Word Embedding as Implicit Matrix Factorization” (Levy&Goldberg, NIPS 2013)
- Word2vec、つまりSkip-gram with negative samplingは、単語ベクトル $\vec{w}$ が (a) 前後の語cと内積が近く (b) ランダムな語c’とは遠いように $\vec{w}$ を学習



単語埋め込みの統計モデル (2)

- よってWord2vecの学習は、次の目的関数を最大化

$$\sum_{w}\sum_{c} n(w,c)\left[\log\sigma(\vec{w}\cdot\vec{c}) + k\cdot\mathbb{E}_{c'\sim p_D}\left[\log\sigma(-\vec{w}\cdot\vec{c'}\right]\right]$$

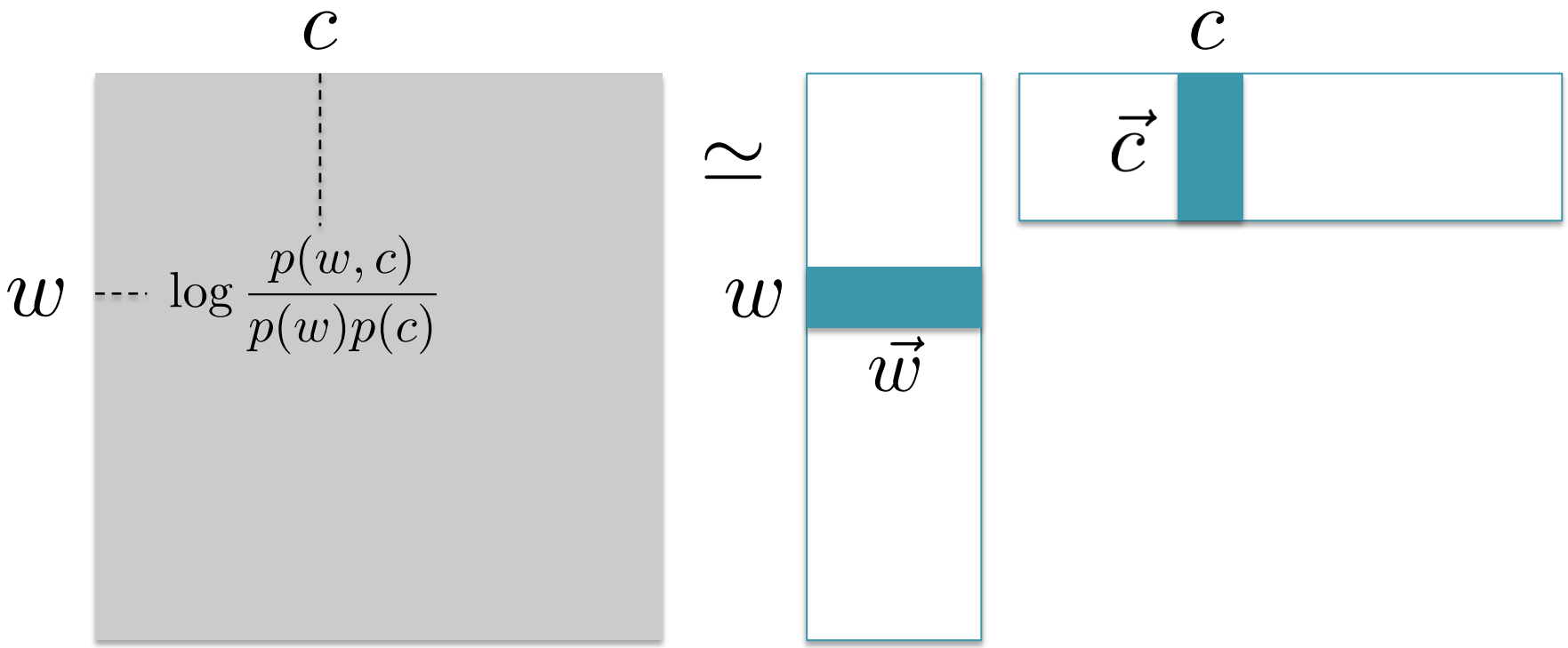
- wで微分して0とおき、しばらく計算するとこの目的関数を最大化する解は下をみたす

$$\vec{w}\cdot\vec{c} = \log\left(\frac{n(w,c)}{n(w)n(c)}\right) - \log k$$

→ 自己相互情報量 (Pointwise Mutual Information, PMI)！

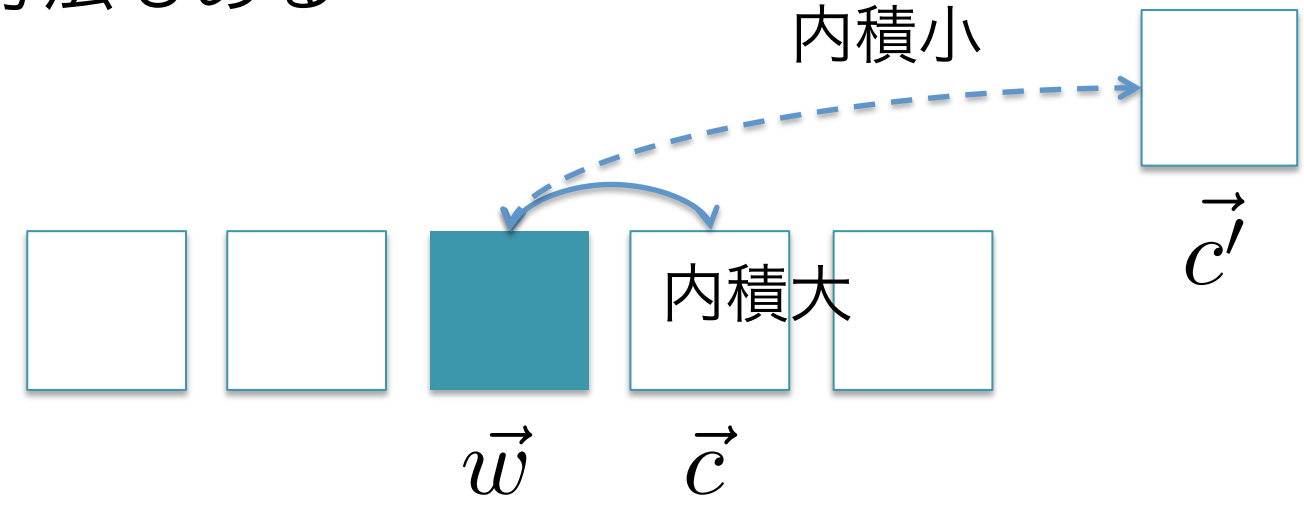
単語埋め込みの統計モデル (3)

- Word2vecは、PMIの行列を行列分解で低ランク近似していることが分かった



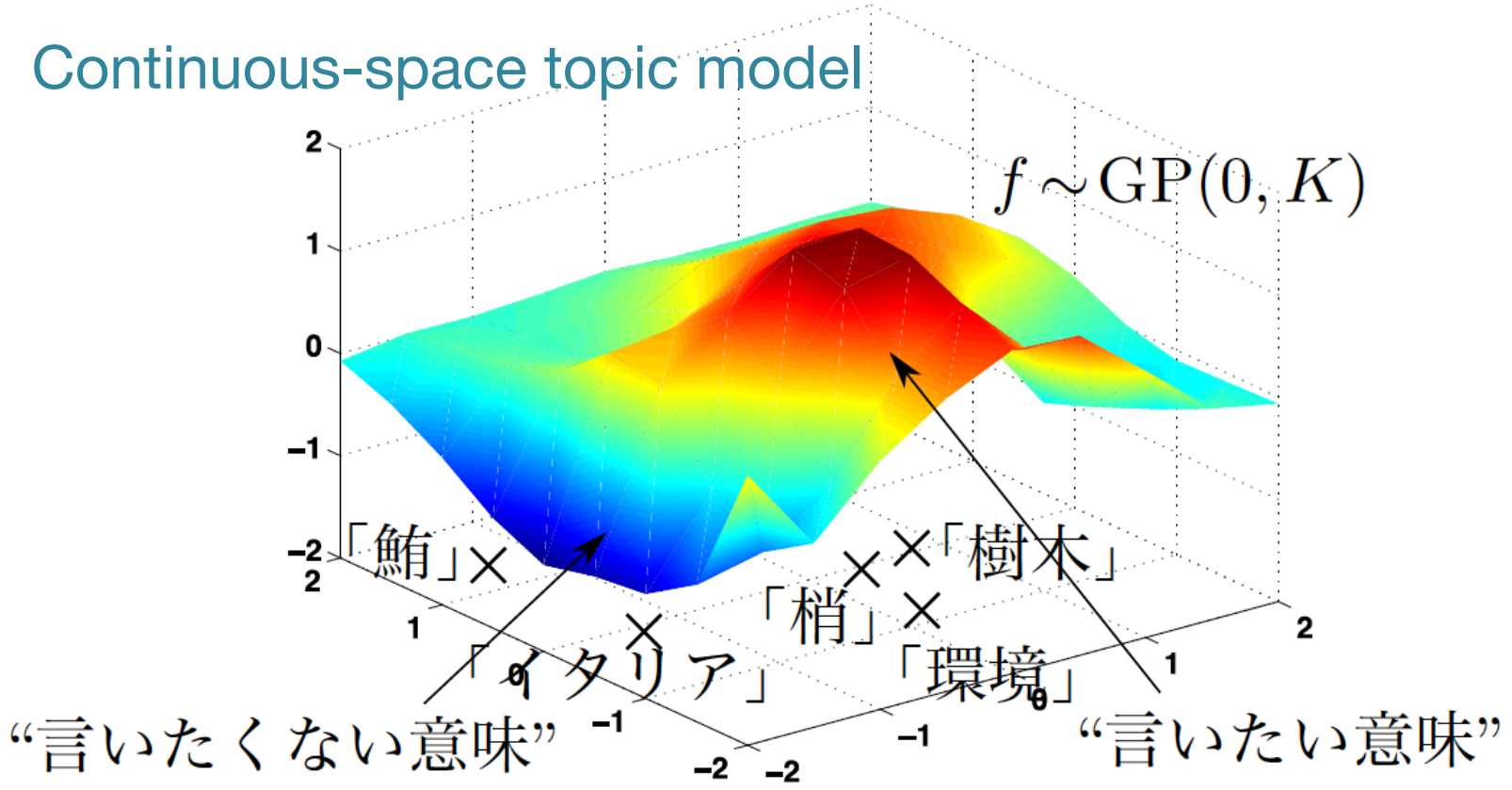
単語埋め込みの統計モデル

- 問題：word2vecの図のような学習基準は、まだヒューリスティック
  → 何かの形で、PMIが自然に導かれるようにできないか？
- (注) GloVe (Pennington+ 2014)という別の統計的な方法もある



おまけ: CSTM (M 2013)

Continuous-space topic model



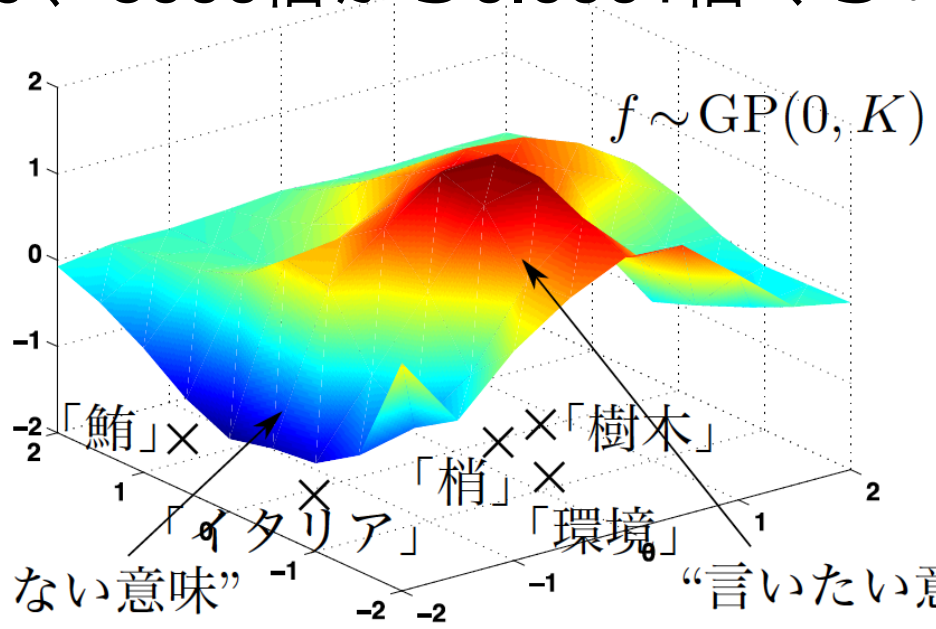
- 単語wは*d*次元の潜在座標 $\phi(w) \sim \mathrm{N}(0, I_d)$ をもつ
- この上に、ガウス過程 $f \sim \mathrm{GP}(0, K)$ を生成

CSTM: 基本モデル

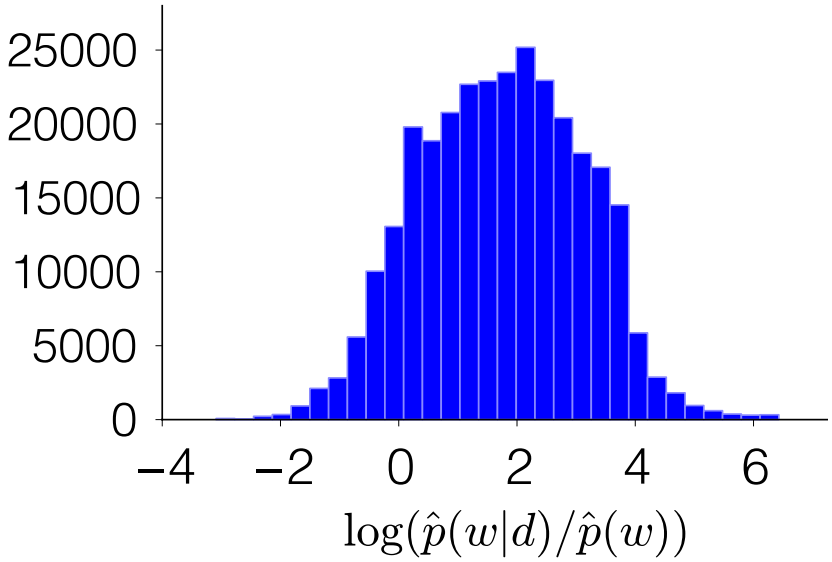
- 単語の平均的な確率(最尤推定) $G_0(w)$ を、ガウス過程 $f(w)$ で Modulate

$$p(w|d) \propto e^{f(w)} G_0(w) = \frac{e^{f(w)} G_0(w)}{\sum_w e^{f(w)} G_0(w)}$$

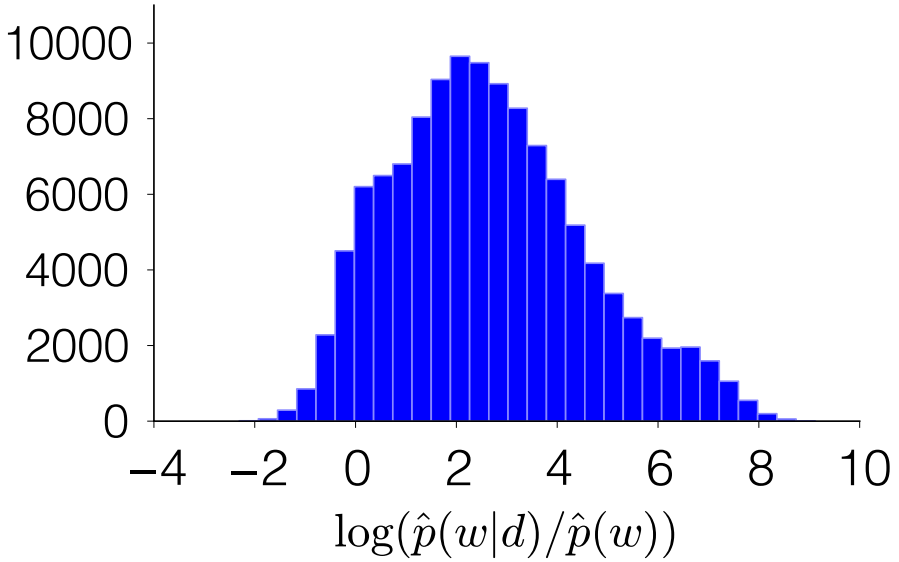
  - $e^{f(w)}$ は、8000倍から0.0001倍くらいの値



Empirical Evidence



Brownコーパス



Cranfield コーパス

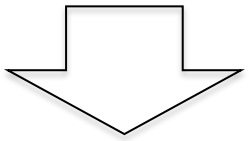
- $p(w|d) \propto e^{f(w)} p(w) \Longleftrightarrow f(w) \propto \log\left(\frac{p(w|d)}{p(w)}\right)$ を最尤推定で計算してプロット
  - $\hat{p}(w|d) = n(w,d)/\sum_w n(w,d),\ \hat{p}(w) = n(w)/\sum_w n(w)$
- 確率の比はほぼGaussianで分布している!

学習

- ガウス過程から生成した関数fは文書ごとに無限次元
  → 学習不可能
- DILN (Paisley+ 2012)と同様に、補助変数uを導入
  - 単語座標の行列を $\Phi = (\phi(w_1), \cdots, \phi(w_V))$ とする
  - $u \sim \mathrm{N}(0, I_d)$ のとき、$f = \Phi u$ はuを積分消去して

$$f|\Phi \sim \mathrm{N}(0, \Phi^T \Phi) = \mathrm{N}(0, K)$$

  - これは、線形カーネル $k(w_i, w_j) = \phi(w_i)^T \phi(w_j)$ を使ったGPと等価なことを意味する



- $\alpha(w) = \alpha_0 G_0(w) e^{u^T \phi(w)}$ として、u と $\phi(w)$ の学習問題!

学習 (2)

- 通常のMetropolis-Hastingsで、単語と文書の潜在座標
  を学習
  - For j = 1 .. J,
    - for i = randperm(1 .. D),
      - Draw u' ~ N(u,σ^2) & MH-accept(u'); Update Z
    - For w = randperm(1 .. W),
      - Draw ϕ'(w) ~ N(ϕ(w),σ^2) & MH-accept(u'); Update Z_1..Z_N
    - z ~ N(0,σ^2); α_0' = α_0 ・ exp(z)
      - If MH-accept(α_0') then α_0=α_0'
  - 実際は、uとϕ(w)の更新をランダムに混合
- 単語間に強い相関があるため、勾配法では局所解

CSTM in retrospective

- CSTMの単語生成モデル (Cox比例ハザードモデル)

$$p(w|d) \propto p(w) \exp(f(w,d))$$

- 変形すると、

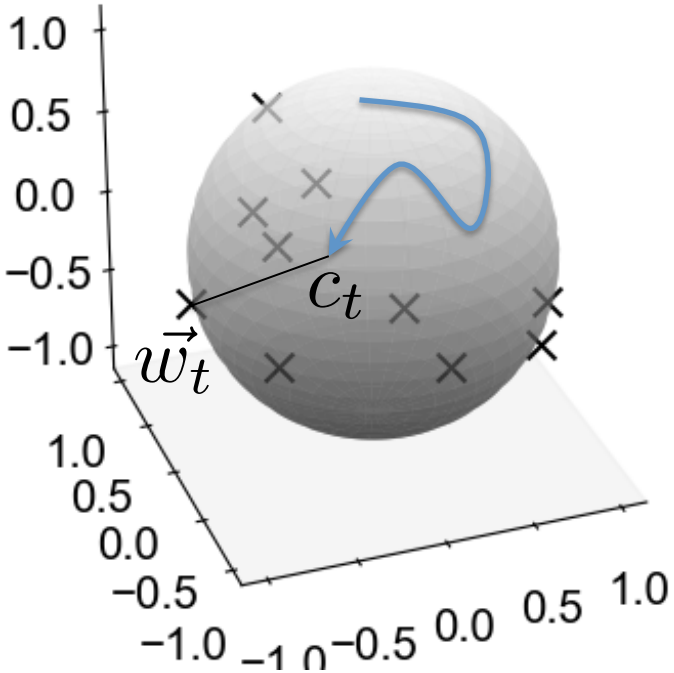
$$\exp(f(w,d)) \propto \frac{p(w|d)}{p(w)}$$

$$\therefore \quad f(w,d) \propto \log \frac{p(w,d)}{p(w)p(d)}$$

  - 結局、$f(w,d) \approx \vec{w} \cdot \vec{d}$ は自己相互情報量だった！

Random Walk in Discourse model

- Arora+ (TACL 2016)で提唱, Mnih&Hintonの Log-linear言語モデルがもと



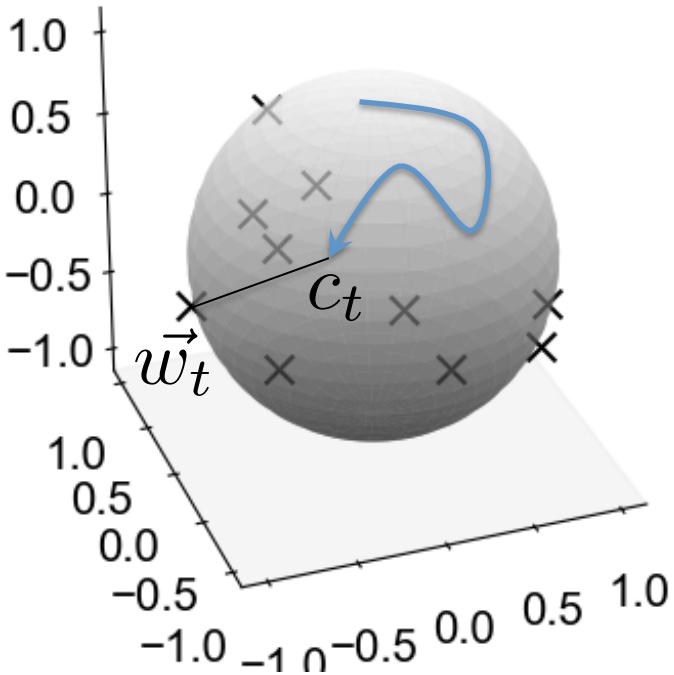
高次元超球面上で、文脈ベクトル c_t がゆっくりランダムウォークしているとする。このとき、単語w_tが出力される確率は

$$p(w_t|c_t) = \frac{\exp(\vec{w}_t \cdot c_t)}{Z}$$

とする

RAND-Walk (Arora+ 2016)

- 定理: このモデルの下で、意味的に近い単語はベクトルが近くなるので



$$\mathrm{PMI}(v,w) = \frac{\vec{v}\cdot\vec{w}}{d} \pm O(\epsilon)$$

単語ベクトルの内積がPMIに対応!
分配関数Zは数万個あるすべての単語についての和で、文脈毎に異なるため、定理の証明はかなり面倒
(注: Self-normalization)

脱線: 文の埋め込み (Arora+ 2017)

- 1つの文の中でc_tが一定だと仮定すれば、そのc_tを求めれば文の埋め込みになるのでは?
  → 元モデルでは、単語ベクトルを単に平均すればc_tの最尤推定量になる
- 問題あり：
  - “the”や“a”などのベクトルも足されてしまう
    → 単語の重みづけが必要
  - 意味というより文法や慣例によるノイズに引きずられる (例: san francisco)
- どんな重みやノイズ除去にすればよい?

Improved Random walk model

- 単語は文脈ベクトルc_t'との近さ以外に、確率αでランダムにユニグラム分布から出力される

$$p(w_t|c_t') = \alpha p(w_t) + (1-\alpha)\frac{\exp(\vec{w}_t \cdot c_t')}{Z}$$

- 文脈ベクトルc_t'は、定常ベクトルc_0と時変成分c_tに分けられる

$$c_t' = \beta c_0 + (1-\beta)c_t, \quad c_0 \perp c_t$$

  - c_0は、文法的な単語の出やすさを支配
  - 最近の研究でも同じ観察 (“All-but-the-top”, ICLR 2018) : こちらの方が深い議論をしているのでお薦め

Estimation

- 文 $s = w_1 w_2 \cdots w_M$ に対して、対応するcを推定したい
- 最尤推定を考えると、

$$\log p(s|c) = \sum_{w \in s} \log \left[\alpha\, p(w) + (1-\alpha) \frac{\exp(\vec{w} \cdot c)}{Z} \right]$$

- Σの中をf(w,c)とおくと、テイラー展開により

$$\begin{aligned} f(w,c) &\approx f(w,0) + \nabla f(w,0)^T c \\ &= \frac{(1-\alpha)/(\alpha Z)}{p(w) + (1-\alpha)/(\alpha Z)} \langle c, \vec{w} \rangle + (\text{constant}) \end{aligned}$$

Estimation (2)

- よって、近似的に

$$\mathrm{argmax}_c \sum_{w \in s} f(w,c) \propto \sum_{w \in s} \frac{a}{p(w)+a} \vec{w}, \quad a = \frac{1-\alpha}{\alpha Z}$$

  - cの推定量は、文の単語ベクトルの重み付き和
  - Smoothed Inverse Frequency (SIF) weighting

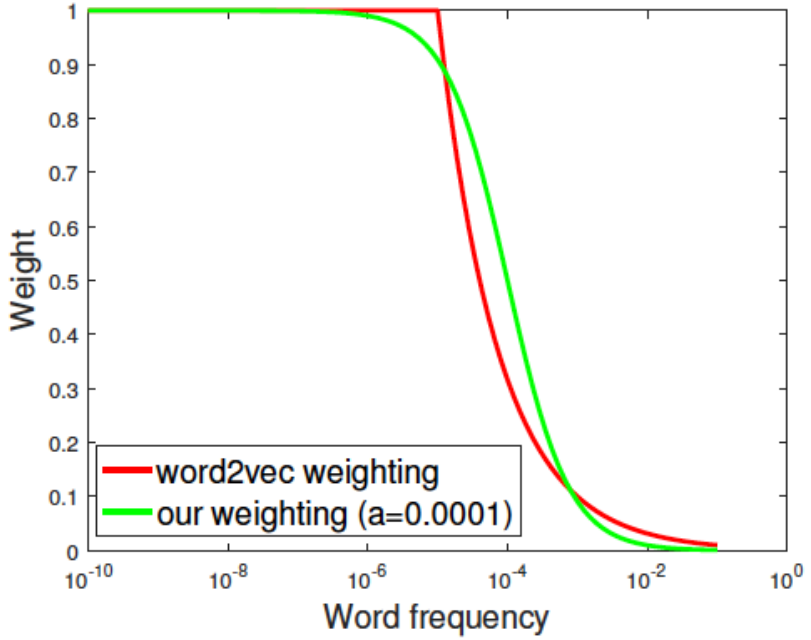
- 注: |c|=1 なので、上の導出には

$$\max_{c:|c|=1} \langle c, g \rangle + C = g/|g|$$

  を使っている

word2vecとの関係

- word2vecは、近傍語wを選ぶときに $1/\sqrt{p(w)}$ に比例して選んでいる
- これは、単語に重み $q(w) = \min\left(1, \sqrt{\frac{10^{-5}}{p(w)}}\right)$ をかけていることに相当
- SIFとq(w)は非常に似ている



Algorithm for sentence embedding

Algorithm 1 Sentence Embedding

Input: Word embeddings $\{v_w : w \in \mathcal{V}\}$, a set of sentences $\mathcal{S}$, parameter a and estimated probabilities $\{p(w) : w \in \mathcal{V}\}$ of the words.

Output: Sentence embeddings $\{v_s : s \in \mathcal{S}\}$

1: **for all** sentence s in $\mathcal{S}$ **do**

2: $\quad v_s \leftarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{a+p(w)} v_w$

3: **end for**

4: Form a matrix X whose columns are $\{v_s : s \in \mathcal{S}\}$, and let u be its first singular vector

5: **for all** sentence s in $\mathcal{S}$ **do**

6: $\quad v_s \leftarrow v_s - uu^\top v_s$

7: **end for**

- 計算の基となる単語埋め込みは、StanfordのGloVeの300次元埋め込みなどを使用
- 注: 提案手法のモデルから単語埋め込みを求めることも多分可能

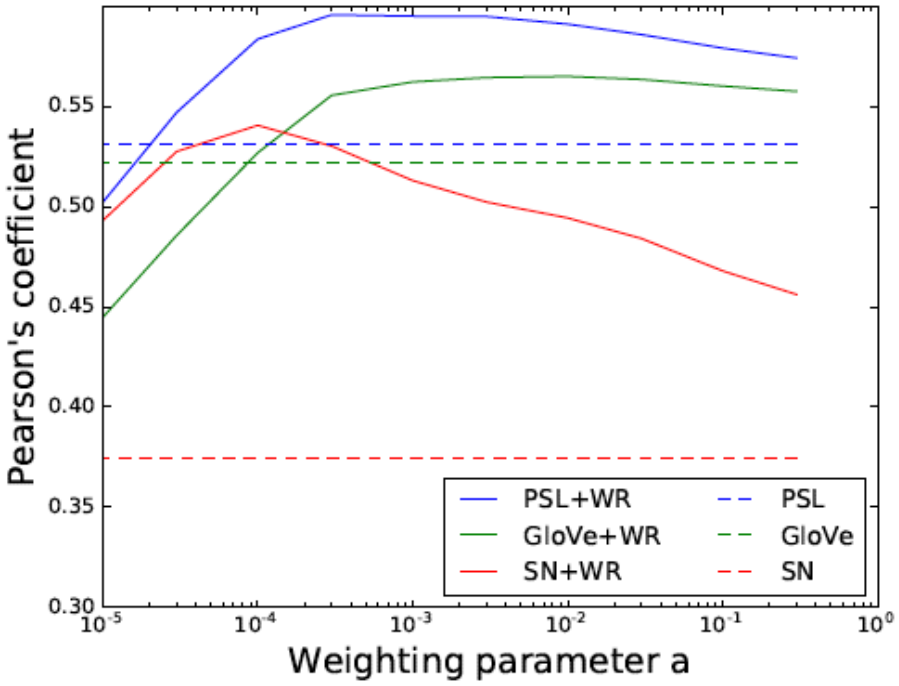
Experiments

- SemEval textual similarityタスク
  - 提案法がRNN, LSTM, SkipThought等を抜き高性能
  - DAN等多くの手法は教師あり＋重い計算が必要

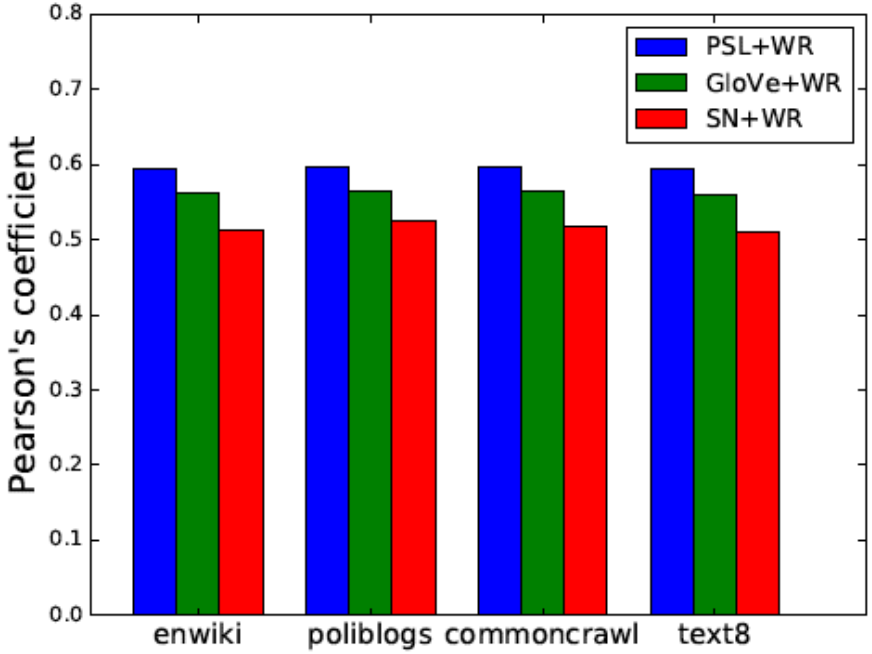
	Results collected from (Wieting et al., 2016) except tfidf-GloVe											Our approach	
Supervised or not	Su.							Un.			Se.	Un.	Se.
Tasks	PP	PP -proj.	DAN	RNN	iRNN	LSTM (no)	LSTM (o.g.)	ST	avg-GloVe	tfidf-GloVe	avg-PSL	GloVe +WR	PSL +WR
STS'12	58.7	**60.0**	56.0	48.1	58.4	51.0	46.4	30.8	52.5	58.7	52.8	56.2	59.5
STS'13	55.8	56.8	54.2	44.7	56.7	45.2	41.5	24.8	42.3	52.1	46.4	56.6	**61.8**
STS'14	70.9	71.3	69.5	57.7	70.9	59.8	51.5	31.4	54.2	63.8	59.5	68.5	**73.5**
STS'15	75.8	74.8	72.7	57.2	75.6	63.9	56.0	31.0	52.7	60.6	60.0	71.7	**76.3**
SICK'14	71.6	71.6	70.7	61.2	71.2	63.9	59.0	49.8	65.9	69.4	66.4	72.2	**72.9**
Twitter'15	52.9	52.8	**53.7**	45.1	52.9	47.6	36.1	24.7	30.3	33.8	36.3	48.0	49.0

Table 1: Experimental results (Pearson's $r \times 100$) on textual similarity tasks. The highest score in each row is in boldface. The methods can be supervised (denoted as Su.), semi-supervised (Se.), or unsupervised (Un.). "GloVe+WR" stands for the sentence embeddings obtained by applying our method to the GloVe word vectors; "PSL+WR" is for PSL word vectors. See the main text for the description of the methods.

αとp(w)の効果



(a)



(b)

- αは10^{-3}〜10^{-4}の広い範囲で高性能 (解析解がある)
- p(w)を計算するコーパスを変えても、結果は同じ

c_0の計算と効果

- c_0はk-SVDで重み付けした単語ベクトル行列の第一固有ベクトルとして求める
- c_0の近傍の語は
  just, when, even, one, up, little, way, there, while, but
  ... 文法的な次元をとらえている

Downstream classification task

	PP	DAN	RNN	LSTM (no)	LSTM (o.g.)	skip-thought	Ours
similarity (SICK)	84.9	85.96	73.13	85.45	83.41	85.8	**86.03**
entailment (SICK)	83.1	84.5	76.4	83.2	82.0	-	**84.6**
sentiment (SST)	79.4	83.4	86.5	86.6	**89.2**	-	82.2

- similarity, entailment, sentiment タスクの文ベクトルとして用いた場合
  - o.g.は出力ゲートあり、noはなし
- similarityとentailmentで提案手法が複雑な手法を凌駕
- sentimentでは既存手法が高性能→(1) SIFは”not”などをほぼ無視 (2) 反意語問題には対応していない

2. 確率モデルを、深層学習を使って推定する.
(確率モデル側からのアプローチ)

確率モデル+Loglinear

- DeNero+ (2010): 出力確率をLoglinearに

$$P_{\mathbf{w}}\left(X_i = d \middle| X_{\pi(i)} = c\right) = \frac{\exp\langle \mathbf{w}, \mathbf{f}(d, c, t)\rangle}{\sum_{d'} \exp\langle \mathbf{w}, \mathbf{f}(d', c, t)\rangle}$$

- EMのMステップでL-BFGS

Algorithm 1 Feature-enhanced EM

repeat
  Compute expected counts $\mathbf{e}$ ▷ Eq. 2
  repeat
    Compute $\ell(\mathbf{w}, \mathbf{e})$ ▷ Eq. 3
    Compute $\nabla\ell(\mathbf{w}, \mathbf{e})$ ▷ Eq. 4
    $\mathbf{w} \leftarrow \text{climb}(\mathbf{w}, \ell(\mathbf{w}, \mathbf{e}), \nabla\ell(\mathbf{w}, \mathbf{e}))$
  until convergence
until convergence

確率モデル+Loglinear (2)

- NLPの各種タスクで高精度 (下記は一部)

Model		Inference	Reg	Eval
POS Induction			κ	Many-1
WSJ	Basic-HMM	EM	–	63.1 (1.3)
	Feature-MRF	LBFGS	0.1	59.6 (6.9)
	Feature-HMM	EM	1.0	68.1 (1.7)
		LBFGS	1.0	**75.5** (1.1)
Grammar Induction			κ	Dir
WSJ10	Basic-DMV	EM	–	47.8
	Feature-DMV	EM	0.05	48.3
		LBFGS	10.0	**63.0**
	(Cohen and Smith, 2009)			61.3
CTB10	Basic-DMV	EM	–	42.5
	Feature-DMV	EM	1.0	49.9
		LBFGS	5.0	**53.6**
	(Cohen and Smith, 2009)			51.9

Variational Autoencoder (VAE)

- Kingma&Welling (2014)
- 変分ベイズ法の、ニューラルネットによる拡張

変分ベイズで何が不足？

- 共役分布族 (ディリクレ-多項分布など)でないとEMアルゴリズムが導けない

  期待値 $\left\langle \log p(D, z|\theta) \right\rangle_{q(z)}, \left\langle \log p(D, z|\theta) \right\rangle_{q(\theta)}$

  が解析的に解けない

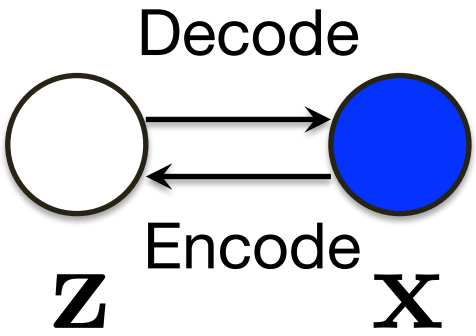
- 強い因子分解の仮定

$$q(z, \theta) = q(z)q(\theta)$$

- 複雑なデータの正確なモデル化が難しい

逆に、ニューラルネットでは？

- 通常のオートエンコーダでは、新しいデータを生成できない
  - 潜在変数 z に分布がない
- きちんとした確率的生成モデルになっていない！

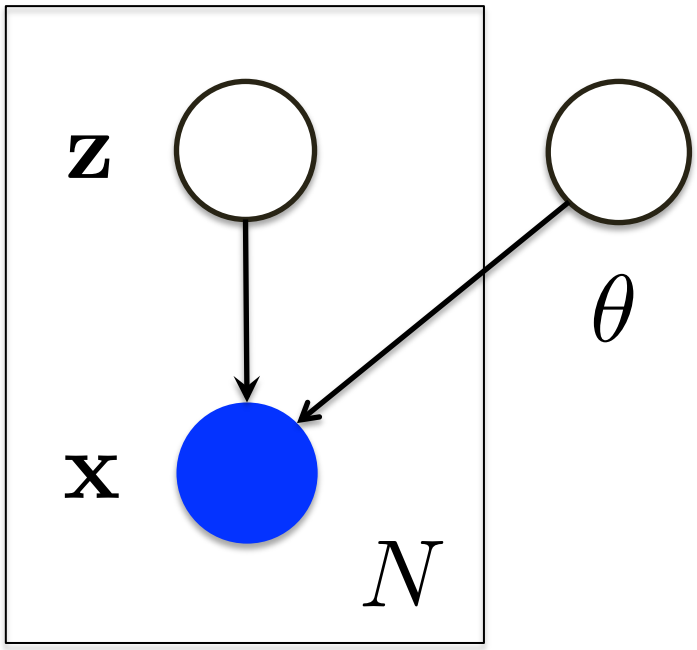


Variational Autoencoder (VAE)

- Kingma & Welling (ICLR 2014)
  - 元のタイトル: “*Stochastic Gradient VB and the Variational Auto-Encoder*”
- 変分下限をニューラルネットで近似
  - 因子分解不要
  - Stochastic Gradientで学習可能
  - z は典型的には多変量標準ガウス分布

$$\mathbf{z} \sim \mathbb{N}(\mathbf{0}, \mathbf{I}_K)$$

VAEのモデル: 前提



- θの分布は(とりあえず)考えない
  (定数として最適化)
- $p(\mathbf{x}|\theta) = \int p(\mathbf{x}|\mathbf{z},\theta)p(\mathbf{z})d\mathbf{z}$
  を最大化
- $\mathbf{z} \sim \mathbb{N}(\mathbf{0}, \mathbf{I}_K)$ が典型的

VAEの導出 (1)

$$\log p(\mathbf{x}|\theta) = \log \int p(\mathbf{x}, \mathbf{z}|\theta) d\mathbf{z} = \log \int q(\mathbf{z}|\boldsymbol{\phi}) \frac{p(\mathbf{x}, \mathbf{z}|\theta)}{q(\mathbf{z}|\boldsymbol{\phi})} d\mathbf{z}$$

$$\geq \int q(\mathbf{z}|\boldsymbol{\phi}) \log \frac{p(\mathbf{x}, \mathbf{z}|\theta)}{q(\mathbf{z}|\boldsymbol{\phi})} d\mathbf{z} \qquad \text{(Jensen)}$$

$$= \int q(\mathbf{z}|\boldsymbol{\phi}) \log \frac{p(\mathbf{x}|\mathbf{z}, \theta) p(\mathbf{z}|\theta)}{q(\mathbf{z}|\boldsymbol{\phi})} d\mathbf{z}$$

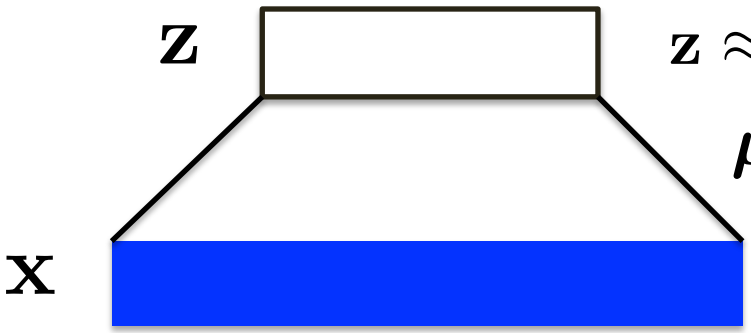
$$= \underbrace{-D(q(\mathbf{z}|\boldsymbol{\phi})||p(\mathbf{z}|\theta))}_{(1)} + \underbrace{\int q(\mathbf{z}|\boldsymbol{\phi}) \log p(\mathbf{x}|\mathbf{z}, \theta) d\mathbf{z}}_{(2)}$$

- Jensenの不等式で下限を取っているだけ
- (2)項を最大化したいが、(1)項がペナルティ(正則化)

VAEの導出 (2)

$$\log p(\mathbf{x}|\theta) \geq \underbrace{-D(q(\mathbf{z}|\boldsymbol{\phi})||p(\mathbf{z}|\theta))}_{(1)} + \underbrace{\int q(\mathbf{z}|\boldsymbol{\phi}) \log p(\mathbf{x}|\mathbf{z},\theta) d\mathbf{z}}_{(2)}$$

- $q(\mathbf{z}|\boldsymbol{\phi}) = \mathbb{N}(\boldsymbol{\mu}(\boldsymbol{\phi}), \mathrm{diag}(\boldsymbol{\sigma}(\boldsymbol{\phi})))$ とすれば、
  (1) は解析的
  (2) はモンテカルロ近似できる
  } SGDで最適化できる!



$$\mathbf{z} \approx \mathbb{N}(\boldsymbol{\mu}(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}(\mathbf{x})))$$

$$\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}(\mathbf{x})$$

VAEの導出 (3)

- 正規分布の間のKLダイバージェンス D(q(z|φ)||p(z)) は解析的に求められるので、結局

$$\log p(\mathbf{x}|\theta) \geq \frac{1}{2}\sum_{k=1}^{K}(1+\log\sigma_k^2-\mu_k^2-\sigma_k^2)+\frac{1}{L}\sum_{\ell=1}^{L}\log p(\mathbf{x}|\mathbf{z}^{(\ell)},\theta)$$

$$\mathbf{z}^{(\ell)} \sim q(\mathbf{z}|\boldsymbol{\phi})$$

  - この下限のGradientを計算してSGDに入れればよい.
  - Reparametrization trick (正規乱数の変換)

VAE: ポイント

- Jensenで下限をとった後のq(z|φ)は任意の関数でよい
  - 因子化仮定は必須ではない
  - q(z|φ)をニューラルネットでモデル化
- ガウス分布間のKLダイバージェンスは解析的に計算できる
  - 本質的には必ずガウス分布である必要はない

VAE=変分近似＋ニューラルネット
ニューラルネットの確率的生成モデル化.

VAE: 注意点

ただし...

- Jensenの不等式により、D(q(z|φ)||p(z)) を最小化
  × D(p(z)||q(z|φ)) ではない

$$D(q(\mathbf{z}|\boldsymbol{\phi})||p(\mathbf{z})) = \int q(\mathbf{z}|\boldsymbol{\phi}) \log \frac{q(\mathbf{z}|\boldsymbol{\phi})}{p(\mathbf{z})} d\mathbf{z}$$

  - モデル化している範囲で、q(z|φ)とp(z)の差が小さければよい
  - p(z)が大, q(z|φ)が小の領域があっても気にしない
  - Peakyな事後分布 (Mode-finding: PRML参照)

VAEのまとめ

- VAE ... 変分ベイズ法の正統進化
  - q(z|φ)として解析関数ではなく、p(z)と同じ分布族(たとえば正規分布)を与えるニューラルネットを考える
  - 勾配が解析的には解けないので、モンテカルロ近似
- 通常のNNと異なり、データを簡単に生成できる
- 変分下限を考える際に、真の事後分布より尖っている可能性が高いので注意が必要

確率モデル+VAE

- 有名な論文は、Johnson+ (NIPS 2016)
  - 色々な確率モデルをVAEで近似
  - ただ、抽象論がほとんどで実際に何をやっているかよくわからない (NLPの例はなし)

Composing graphical models with neural networks for structured representations and fast inference

Matthew James Johnson
Harvard University
mattjj@seas.harvard.edu

David Duvenaud
Harvard University
dduvenaud@seas.harvard.edu

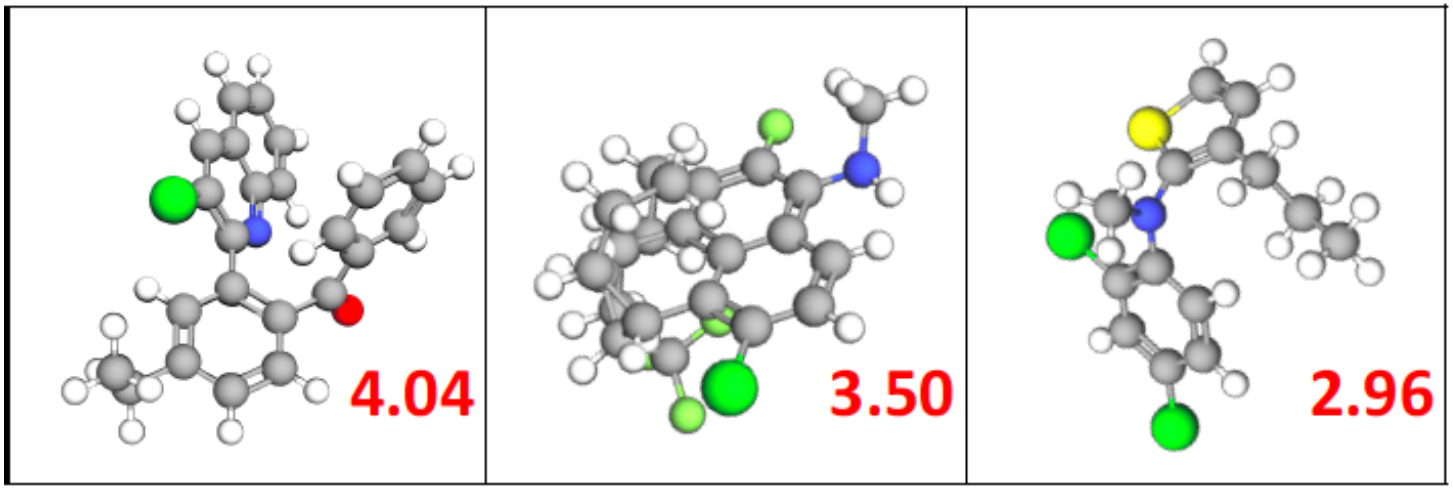
Alexander B. Wiltschko
Harvard University, Twitter
awiltsch@fas.harvard.edu

Sandeep R. Datta
Harvard Medical School
srdatta@hms.harvard.edu

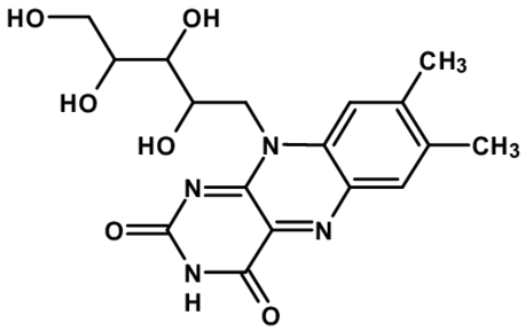
Ryan P. Adams
Harvard University, Twitter
rpa@seas.harvard.edu

確率モデル+VAE (2)

- “Syntax-directed Variational Autoencoder for structured data”, ICLR 2018
- 創薬や材料科学などで、分子構造の良い統計モデルが欲しい



確率モデル+VAE (3)



- SMILES記法：

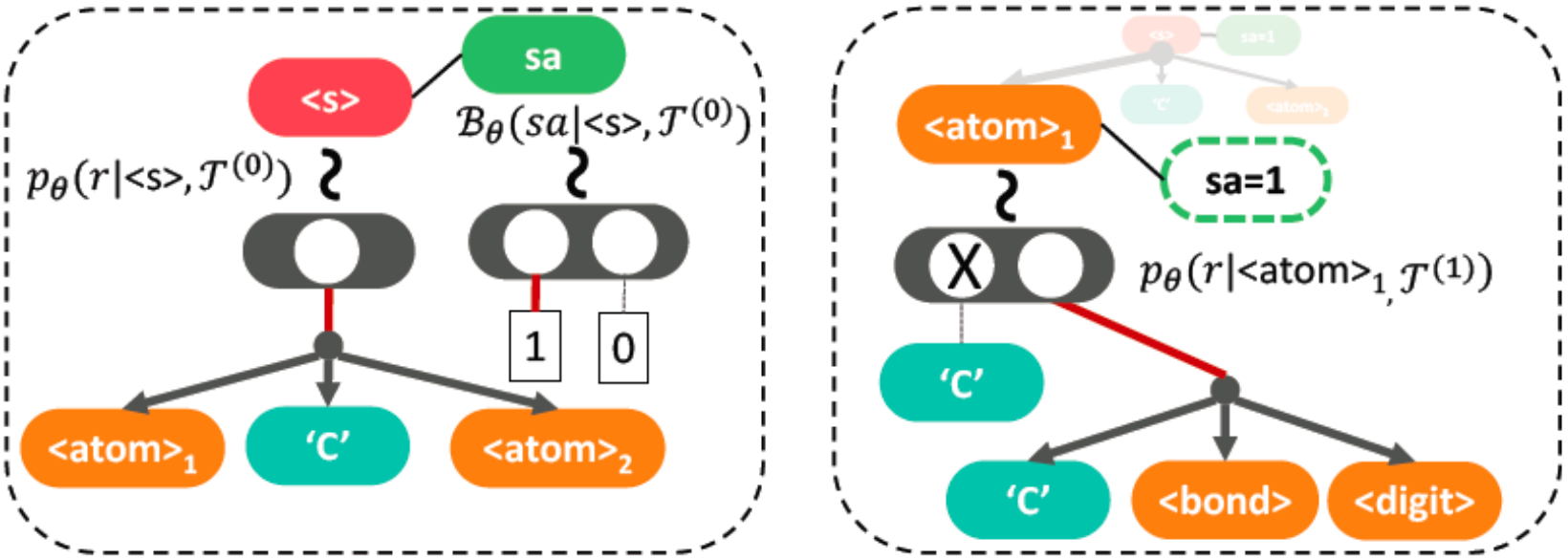
7,8-dimethyl-10-(2,3,4,5-tetrahydroxypentyl)benzo[*g*]pteridine-2,4(3*H*,10*H*)-dione

OCC(O)C(O)C(O)CN2C3=NC(=O)NC(=O)C3=Nc1cc(C)c(C)cc12

- 単純なアプローチ: SMILES記法を言語モデルから生成 (LSTM)
  - 非常に長距離の相関には弱い
  - 必ず化学的に妥当な構造となる保証が無い
  - 様々な条件を満たすように生成することが難しい

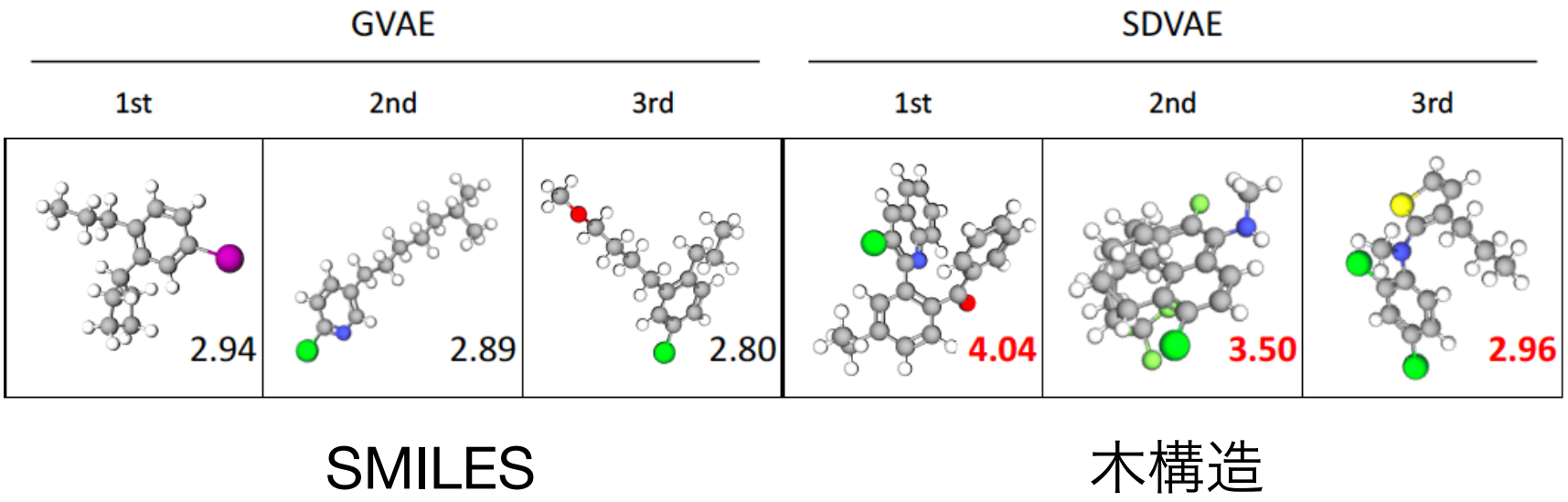
確率モデル+VAE (4)

- 分子の木構造からの生成 (構文解析、CFG)



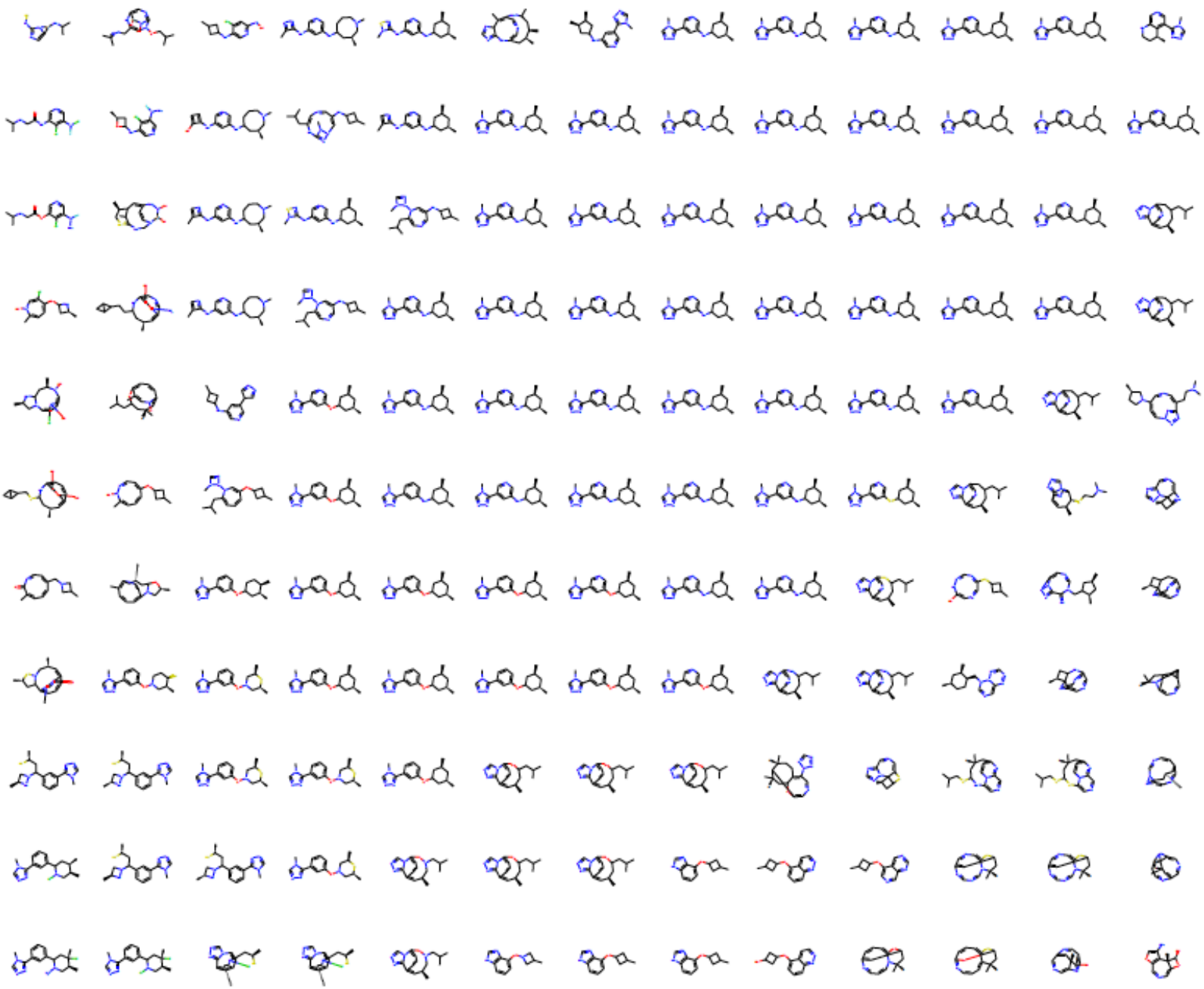
- 制約をベルヌーイ分布から先に生成し、制約を満たすように下部構造を生成
  - 統計的には、補助変数(auxiliary variable)モデル?

確率モデル+VAE



SMILES　　木構造

- 木構造を考慮しないモデルより、ずっと妥当な化学構造の生成
- VAEによる、木構造の連続空間への埋め込み

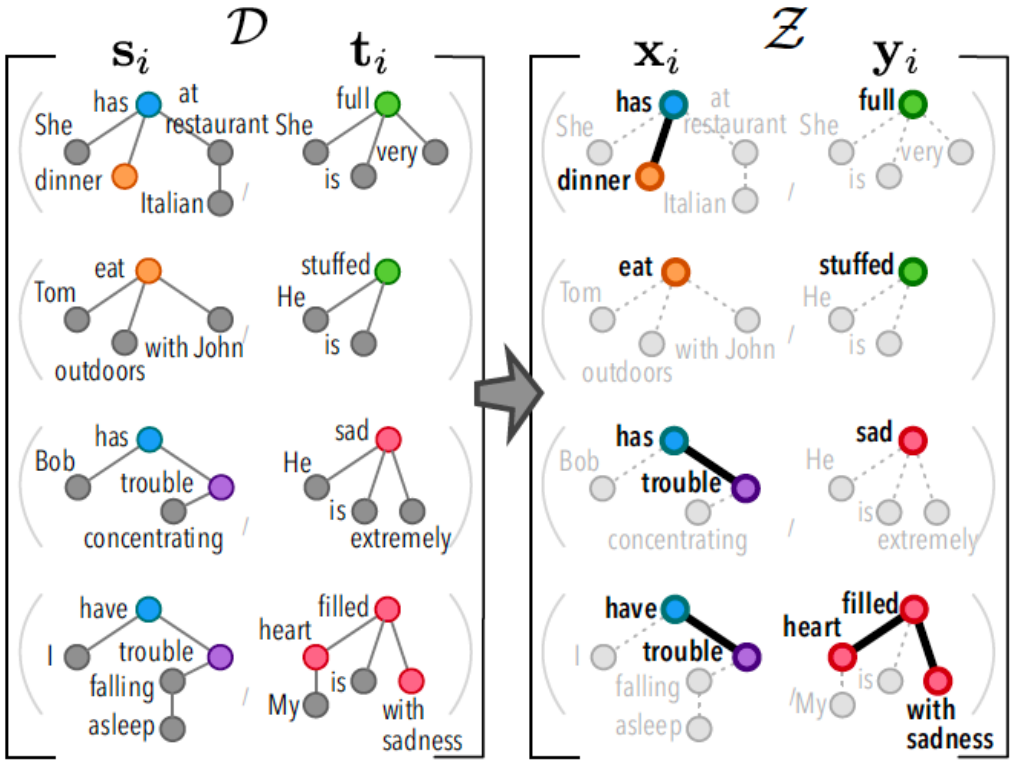


「知識」の機械学習

- 自然言語は、その裏に様々な知識を背景として成立している
  - 「晴れる」→「洗濯物が乾く」
  - 「仕事を頼まれる」→「休暇が取れない」
  - 手で全ての知識を書くことは無理
- 詳しくは、海野さん(PFN)のチュートリアル参照

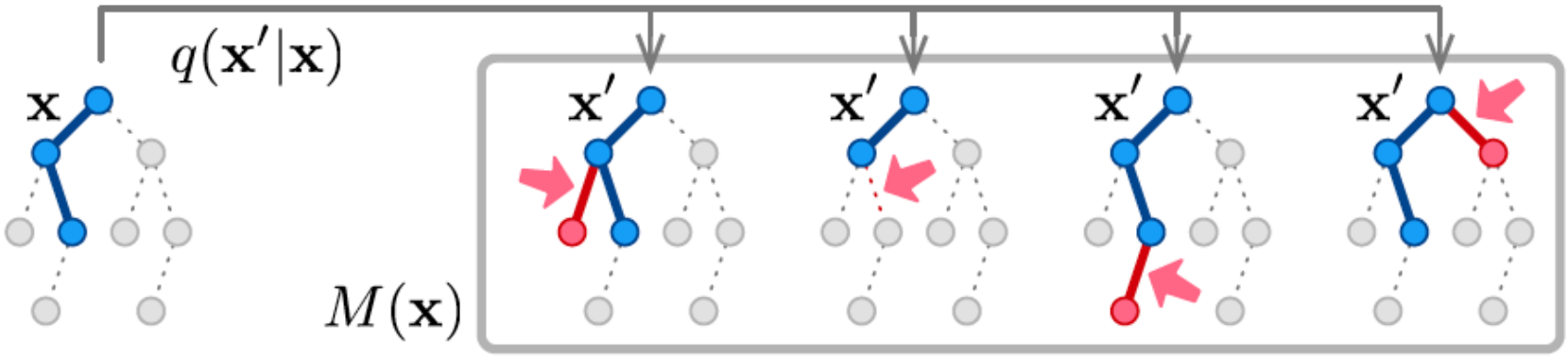
「知識」の機械学習 (2)

- 横井君(東北大)との共同研究 (IJCAI 2017)
- 「知識の粒度」を自動発見する
  - 「友達と朝に食事」→「とてもお酒が進む」
  - 「雨で大変だった」→「体が芯まで濡れている」



「知識」の機械学習 (3)

- HSIC (Hilbert-Schmidt Independence Criterion; Gretton+ 2005)で関連性を定義
  (カーネル法に基づく相互情報量)
  → MCMCでHSICを最大化する部分木を探索



- 詳しくは→ 横井君に聞く, あるいは論文参照

最後に

- 自然言語処理は、内部の人が思っているより応用分野が非常に広い
  - 音声認識、音楽情報処理 (歌詞)
  - 情報推薦、協調フィルタリング、広告
    - A, G, F, Cookpad, 不動産、ファッション、旅行etc
  - 計算社会科学 (経済学、政治学、社会学)
  - バイオインフォマティクス
  - ロボティクス
- 上は私が共同研究している例だが、他にも多数関連あり

まとめ

- 自然言語処理(NLP)では現在深層学習が流行っているが、画像や音声ほどのインパクトはない
  - 言語はもともと構造的なため
  - 離散、正確なルールが存在
- 単語埋め込みが、NLPの深層学習の基礎
- 深層学習と通常の機械学習/統計をどう繋ぐかが課題
  - 深層学習側からのアプローチ
  - 機械学習側からのアプローチ
- NLPの射程は広大、社会的に今後重要