ベイズ全域木による文書クラスタリング

岡野原　大輔　　東京大辻井研

- 大規模文書集合の高速な類似度計算手法
  - クラスタリング等に利用可能
- 提案手法
  1. 文書dから文書d'が生成されるモデル p(d|d')
  2. 文書集合中の疎な有向グラフを構築
  3. 疎な有向グラフから有向全域木を推定
  4. 有向全域木を用いたアプリケーション

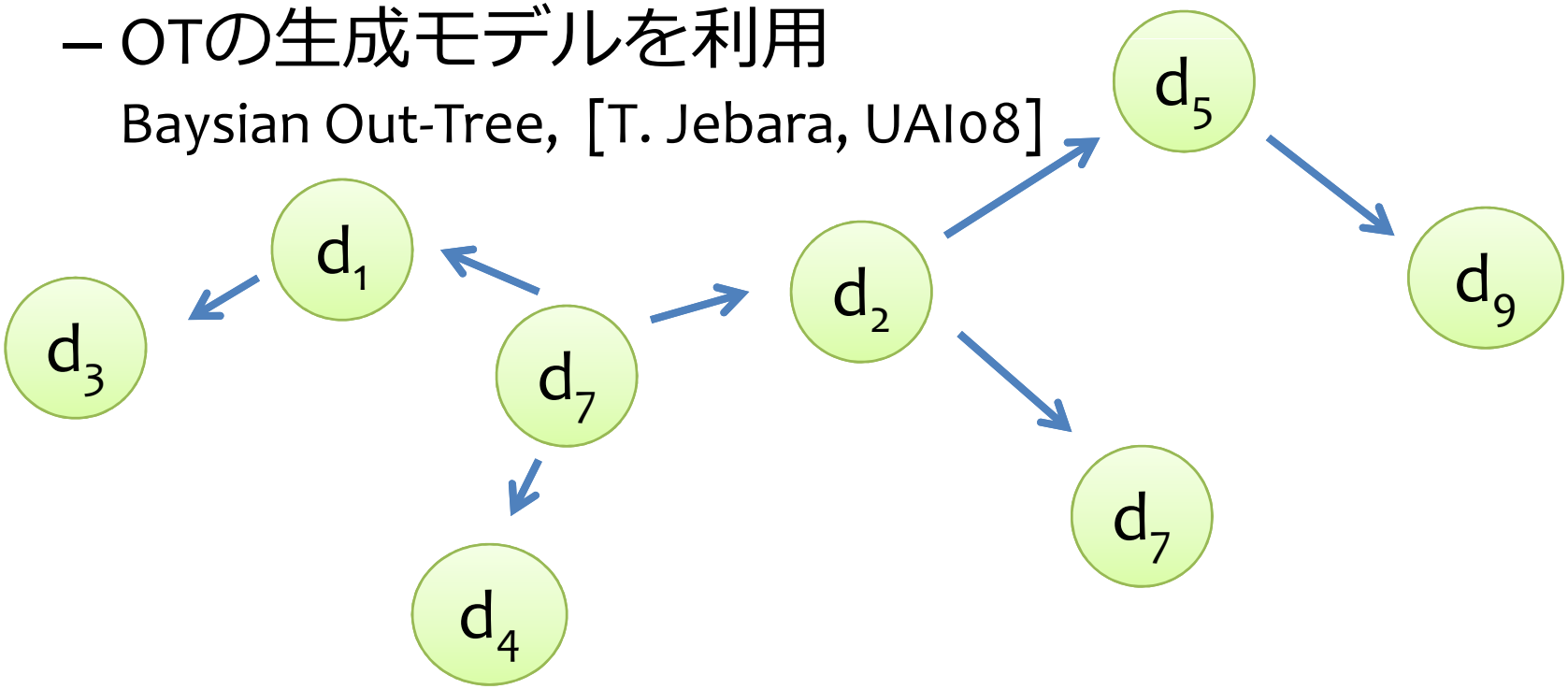
準備：文書クラスタリング

- 文書集合 $D = \{d_1 d_2 \ldots d_n\}$
- D を似た文書同士に分割
- 文書数が増えるに連れ計算量が増大
- k-means / 混合分布
  - クラスタ数に計算量が比例 $O(nk)$
  - 細かい粒度($k \propto n$)で解析ができない
  - 例：この文書と似た文書を10個列挙

提案手法

- 文書集合を有向全域木(OT:Out-Tree) で解析
  - 各文書には親が唯一つ存在　& 閉路無
  - c.f. 交差有のMSTを用いた係り受け解析
  - OTの生成モデルを利用

    Baysian Out-Tree, [T. Jebara, UAI08]



生成モデル

- OTを生成し、それに従い文書を順に生成
- $p(\mathbf{h})$：（非観測の）OTの生成
  - $\mathbf{h} = \{h_1\, h_2 \ldots h_n\}$：$h_i$は$d_i$の親の文書番号
  - $p(\mathbf{h})$は、一様分布を仮定
- $p(D) = \sum_{\mathbf{h}} p(\mathbf{h}) \prod_{i=1\ldots n} p(d_i \mid d_{hi})$ (1)

  $\fallingdotseq \prod_{i=1\ldots n} p(d_i \mid d_{h^*i})$ (2)
  - $\mathbf{h}^* = \mathrm{argmax}_h\ p(d_i \mid d_{hi})$
  - (1)はMatrix Tree定理で$O(n^3)$時間で求められるが、今回はmaxで近似

文書から文書の生成モデル(1/2)

[P. Haider+ ICML09]のモデル設計を利用

- $\Phi(d) \in \{0,1\}^m$: dの特徴ベクトル(e.g. BOW)
- $p(d|\theta) \propto \Pi_k p_{ber}(\Phi_k(d) | \theta_k)$
  - 各次元が独立な多次元ベルヌーイ分布
  - $p_{ber}(x_k | \theta_k) = \theta_k^{xk}(1-\theta_k)^{(1-xk)}$
- $p(\theta) \propto p_{beta}(\theta|\alpha, \beta)$
  - θはベータ分布（p_{ber}と共役）から生成
  - $p_{beta}(\theta_i | \alpha_i, \beta_i) = \theta_i^{(\alpha i-1)}(1-\theta_i)^{(\beta i-1)}$

直感的には $\alpha_k : \Phi_k(d) = 1$ の成り易さ（α_k ～1)
$\beta_k : \Phi_k(d) = 0$ の成り易さ(β_k ～100)

文書から文書の生成モデル(2/2)

$$P(d|d') = \int_{\boldsymbol{\theta}} P(d|\boldsymbol{\theta})p(\boldsymbol{\theta}|d')d\theta$$

θを積分消去

$$= \prod_{k:\phi_k(d)=1} \frac{\alpha_k + \phi_k(d')}{\alpha_k + \beta_k + 1} \prod_{k:\phi_k(d)=0} \frac{\beta_k + 1 - \phi_k(d')}{\alpha_k + \beta_k + 1}$$

各特徴の確率への貢献度

	$\Phi_k(d')=1$	$\Phi_k(d')=0$
$\Phi_k(d)=1$	A:$(\alpha_k + 1)/C$	B: α_k /C
$\Phi_k(d)=0$	C: $(\beta_k)/C$	D: $(\beta_k+1)/C$

大差

小差

$C = \alpha_k+\beta_k+1 \fallingdotseq \beta_k$
$(\beta_k >> \alpha_k)$

新しい特徴が発火するのには大きなコスト
消えるのならどちらでも

疎な文書関係の抽出 (1/4)

- 全文書間で$p(d|d')$を求めるのは困難
  - $|D| \sim 10^6$　$O(|D|^2)$は不可能
- $p(d|d')$を高精度かつ高速に近似する
  - $\Phi(d)$, $\Phi(d) \times \Phi(d')$が疎であることを利用
- $\log p(d|d') =$

$$\underbrace{\sum_{i \in A} \log(\alpha_i + 1)}_{\text{d, d'の両方で発火}} + \underbrace{\sum_{i \in B} \log(\alpha_i)}_{\text{dのみで発火}} +$$

$$\underbrace{\sum_{i \in C} \log(\beta_i)}_{\text{d'のみで発火}} + \underbrace{\sum_{i \in D} \log(\beta_i + 1)}_{\text{d, d'ともに発火していない}} + \text{const.}$$

疎な文書関係の抽出 (2/4)

- $\sum_{i \in C} \log(\beta_i) + \sum_{i \in D} \log(\beta_i + 1)$
  $= B - \sum_{i \in A \cup B} \log(\beta_i)$
  - $\log(\beta_i) \fallingdotseq \log(\beta_i + 1)$ $(\beta_i >> 1)$より
  - $B = \sum_i \log(\beta_i)$
- $\sum_{i \in A} \log(\alpha_i + 1) + \sum_{i \in B} \log(\alpha_i)$
  $= \sum_{i \in A} \log(1 + 1/\alpha_i) + \sum_{i \in A \cup B} \log(\alpha_i)$
- $\log p(d|d') = \sum_{i \in A} \log(1 + 1/\alpha_i)$
  $\underbrace{-\sum_{i \in A \cup B} \log(\alpha_i/\beta_i)}_{d\text{のみに依存：}q(d)} + B$

疎な文書関係の抽出 (3/4)

- $\log p(d|d') = q(d) + \sum_{i \in A} \log(1 + 1/\alpha_i)$
  - $q(d) := \sum_{i:\Phi i(d)=1} \log(\alpha_i/\beta_i)$
- $\log(1 + 1/\alpha_i)$が大きいもの順に加える
  - α_iはiのdfに比例（α_iの求め方は後述）
  - $\log(1 + 1/\alpha_i)$ はidfと考えられる
  - $\log(1 + 1/\alpha_i)$ が大きい⇔α_iが小さい
    ⇔特徴iが発火している文書数が少ない
  - 高速に計算可能

疎な文書関係の抽出 (4/4)

- L(i) : 特徴iが発火している文書集合
  - 特徴番号は|L(i)|が小さい順に並んでいるとする
- α_i , β_i : パラメータ（決定的に求まる）
- C : 閾値

```
for (i = 1; |L(i)| <C; ++i)
   for d, d' ∈ L(i)
         S[d, d'] += log(1 + 1/α_i)
   end for
end for
log p(d|d') =q(d) + S[d, d']
```

$O(|L(i)|^2)$
L(i)は非常にskew
c.f. Zip'f 法則

$q(d) := \sum_{i \in \Phi(d)} \log(\alpha_i/\beta_i)$

OTの推定

- log(d|d')をd'からdへの枝の重みとする
- 最大重みとなる有向全域木を求める
  - CLE algorithm [Chu+ 65, Edmonds, 67]を利用
    c.f. 交差有係り受け解析で利用 [Mcdonald+ 04]
- 枝の張られ方の傾向
  - （珍しい）単語を共有すると枝が張られる
  - 単語の消失方向に大きいペナルティ
    p(d|d')とp(d'|d)を比較すると差はq(d), q(d')

ハイパーパラメータ推定

- パラメータは(ほぼ)全自動で求める
  [P. Haider+, ICML 09]
  - α, βはパラメータを共有
  - $\alpha_i = \mu_i\sigma \quad \beta_i = (1-\mu_i)\sigma$
    - σはprecisionと呼ばれ，分散の逆数に比例
  - μ_i は$p_{beta}(\alpha_o, \beta_o)$に従い生成
  - $\mu_i = \mathrm{argmax}(\mu \mid d)$
    $= (\alpha_o + f_i - 1)/(\alpha_o + \beta_o + n - 2)$
  - パラメータ $\alpha_o, \beta_o, \sigma$は幅広い範囲でロバスト
    - $\alpha_o=1.1,\ \beta_o=100,\ \sigma=1$

OTを用いたアプリケーション

- 系統樹/カテゴリ生成
- 文書間の連想パス
  - 二つの文書を与えた上でそれらを連想ゲームのように結びつける
  - 複数の文書の中心文書などを発見する
- 概念距離
  - 近傍グラフ上の距離は多様体上の距離の良い近似であることが知られている

実験データ

- 日本語Wikipedia
  - 文書数：1086582
  - 単語数：196736
  - 総出現単語数：37659990 （平均 34単語/文書）
- 英語 Wikipedia
  - 文書数：4029545
  - 単語数：26627436
  - 総出現単語数：341707137（平均 84単語/文書）

IPA辞書にはてなキーワード，Wikipediaのタイトルで補強し抽出

分かち書きされた結果をnormalize

実験設定

- 内積を類似度とする手法と比較
  - 特徴ベクトルはtf-idf
  - 個別に計算する手法とまとめて計算する手法
- 計算時間の計測
  - 内積による手法はランダムサンプリングによる推定値（N=1000）

実験結果 (1/2)

- 日本語Wikipedia

	全類似度　計算時間 (s)	一要素あたり(ms)
個別の内積計算	2.27×10^6	209
まとめて内積計算	1.70×10^6	157
提案手法による類似度計算（$\gamma = 3$）	11.5	0.011
提案手法による類似度計算（$\gamma = 10$）	425	0.395

実験結果 (2/2)

- 英語Wikipedia

	全類似度　計算時間 (s)	一要素あたり(ms)
個別の内積計算	4.07×10^7	1010
まとめて内積計算	3.83×10^7	950
提案手法による類似度計算（$\gamma=3$）	13.2	0.003
提案手法による類似度計算（$\gamma=10$）	84.9	0.021

まとめ・今後の予定

- 大規模文書の疎な関係を抽出
  - 高速な近似法を利用可能
- クラスタリングなどの応用が可能
- 文書以外のデータにも適用可能
  - 単語、画像、系統樹解析
- モデルの精緻化
  - 特徴ベクトルで組み合わせ [Okanohara+ 09]
- Q & A、情報検索の新しいアプリ？
  - 例：二つ文書を与えたらその二つのまとめ文書？
    OR 経路上の文書を返す