The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

ノンパラメトリックベイズ二重分節解析器の高速化に関する研究

Acceleration of Nonparametric Bayesian Double Articulation Analyzer

尾崎 僚 *1　　谷口 忠大 *1
Ozaki Ryo　　Taniguchi Tadahiro

*1立命館大学
Ritsumeikan University

In segmentation of time series data using unsupervised learning, it is important to pay attention to the structure of data. Nonparametric Bayesian double articulation analyzer (NPB-DAA) is a method of assuming only the double articulation structure and directly estimating the language model and the acoustic model from the acoustic feature. However, there are problems that conventional NPB-DAA spend a lot of time to analyze parameters, and only experiments using small speech data are performed. In this paper, we propose NPB-DAA with lookup table that introduced a lookup table to conventional NPB-DAA. Therefore, in this paper, we improved the speed of conventional NPB-DAA by introducing a lookup table in conventional NPB-DAA. In the experiment, we analyzed real speech that time length is about 1 minute 20 seconds and show that NPB-DAA with lookup table can estimate parameters 10 times faster than conventional NPB-DAA.

1. はじめに

幼児の語彙獲得の過程において，連続音声信号からの単語分割が重要なタスクであると知られている．また，人間の幼児は月齢 8ヶ月の段階において，音声信号を単語ごとに分割ができる [Saffran 26]．音声言語は，単独では意味を持たない音素と，音素の組み合わせによって意味を持つ単語を構成する二重分節構造を持つ．二重分節構造を持つ時系列データを解析する手法の一つに，谷口らの開発したノンパラメトリックベイズ二重分節解析器 (Nonparametric Bayesian Double Articulation Analyzer: NPB-DAA) がある [Taniguchi 16a]．[Taniguchi 16a] では，母音のみで構成された小規模な音声データの分節化が実現された．しかしながら，NPB-DAA は計算コストが非常に大きく，大規模な音声データの解析には非常に長い時間を要するという問題があった．

本研究では，NPB-DAA の高速化を図る．NPB-DAA の高速化の実現により，より大規模な音声データの解析が可能になることが期待される．NPB-DAA の計算コストが大きい理由として，同じ結果が得られる計算を何度も繰り返し計算していることが挙げられる．そこで，本研究では一度計算して求めた値をメモリに保存しておき，再計算のコストを省くルックアップテーブルを導入することで，NPB-DAA の高速化を実現する．本研究では，推論計算過程の効率化により，計算量オーダを 3 次オーダから 2 次オーダに削減し，実行時間を 90%削減した．

2. 先行研究

Taniguchi らは教示者による非分節な動作を，学習者が自律的に分節化し，模倣するための手法として二重分節解析器を提案した [Taniguchi 11]．この二重分節解析器は Sticky Hierarchical Dirichlet Process Hidden Markov Model (sticky HDP-HMM) と Nested Pitman-Yor Language Model (NPYLM) [Mochihashi 09] に基づく教師なし形態素解析器により構成される．しかしながら，sticky HDP-HMM と NPYLM の組み合わせによる二重分節解析器は，sticky HDP-HMM の認識誤りによって NPYLM の性能低下を引き起こすという問題があった．そこで，Taniguchi らは時系列データの離散符号列化と分節化のプロセスを統合した階層ディリクレ過程隠れ言語モデル (Hierarchical Dirichlet Process Hidden Language Model: HDP-HLM) と，そのパラメータの推論手法を導出し，これらによりノンパラメトリックベイズ二重分節解析器 (NPB-DAA) を提案した．また，母音のみで構成された小規模な音声データからの語彙獲得実験を行い，sticky HDP-HLM と NPYLM の組み合わせによる二重分節解析器と比べて，高い単語分割性能を示した [Taniguchi 16a]．

また，[Taniguchi 16b] では単語分割性能の向上を図るため，深層学習の一種である Deep Sparse Auto-encoder (DSAE) を用いて入力データを低次元化した特徴量を NPB-DAA の入力データとして扱った．Tada らは，子音も含めた音声データへの適用のため，動的特徴量を含めた特徴量の抽出方法を検討し，子音を含めた音声データからの教師なし語彙獲得を可能とした [Tada 17]．

しかしながら，NPB-DAA は，フレーム数を T，単語数の上限を N，1 単語の音素の最大数を L_{max}，1 単語の最大継続長を d_{max} としたとき，計算量が $\mathcal{O}(TN^2L_{max}d_{max}^3)$ と非常に大きい．実音声にして約 1 分 20 秒の音声データから得られた 3 次元・全 6311 フレームの特徴量からの教師なし語彙獲得に，Intel Xeon CPU E5-2630 v2 2.60GHz が 6 コア × 2CPU で構成された PC で 1 試行に約 2.7 時間必要であるため，音声認識の分野でよく用いられる TIMIT コーパス *1 といった大規模な音声データからの語彙獲得は行われてこなかった．そこで，本研究では NPB-DAA にルックアップテーブルを導入することで，NPB-DAA の高速化を実現する．

連絡先: 尾崎 僚，立命館大学 情報理工学研究科，滋賀県 草津市 野路東 1-1-1，ryo.ozaki@em.ci.ritsumei.ac.jp

*1 TIMIT Acoustic-Phonetic Continuous Speech Corpus: https://catalog.ldc.upenn.edu/ldc93s1

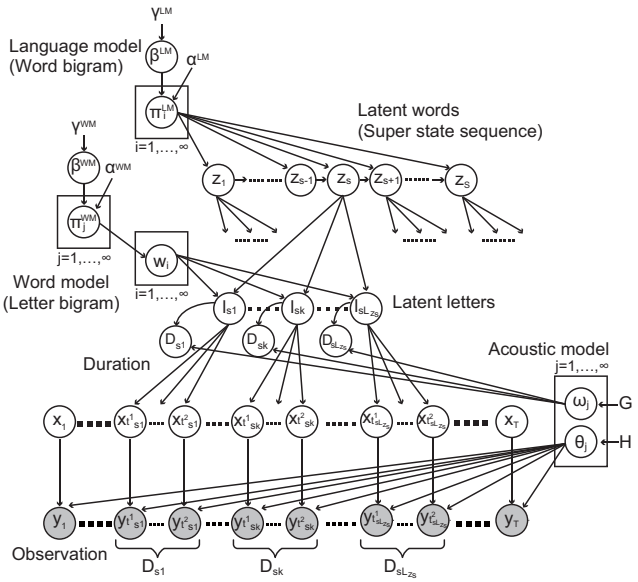

図 1: HDP-HLM のグラフィカルモデル [Taniguchi 16a]

3. NPB-DAA

3.1 階層ディリクレ過程隠れ言語モデル

HDP-HLM は，単語の生成確率を持つ言語モデル，文字列の生成確率を持つ単語モデル，音素信号列の生成確率を持つ音響モデルからなる生成モデルである．このモデルは，Matthew J らの提案した HDP-HSMM [Johnson 13] を拡張し，二重分節構造を含む系列データへの適用を可能としたモデルである．図 1 に HDP-HLM のグラフィカルモデルを示す．その他，生成過程等については， [Taniguchi 16a] を参照されたい．

3.2 ノンパラメトリックベイズ二重分節解析器

HDP-HLM のパラメータ推論手法として，NPB-DAA ではギブスサンプリングによるベイズ推定を行う．潜在変数のサンプリングでは，ブロック化ギブスサンプリングを用いる．また，潜在単語 w_i のサンプリングでは Sampling Importance Resampling (SIR) アルゴリズムを用いている．

3.2.1 潜在単語のサンプリング

潜在単語 z_s のサンプリングでは，HDP-HSMM の推論手法で用いられる backward filtering forward sampling の手続きを拡張して用いる．HDP-HLM における潜在単語 $z_s = i$ の backward message は下式で計算できる．

$$B_t(i) = \sum_j B^*_t(j) p(z_{s(t+1)} = j | z_{s(t)} = i) \quad (1)$$

$$B^*_t(i) = \sum_{d=1}^{T-t} B_{t+d}(i) p(D_{s(t+1)} = d | z_{s(t+1)} = i) \times p(y_{t+1:t+d} | i, d) \quad (2)$$

$$B_T(i) = 1 \quad (3)$$

ここで，$z_{s(t)}$ は時刻 t における潜在単語 z_s，$D_{s(t)}$ は時刻 t における潜在単語の持続時間を表す変数である．式 (2) における $p(y_{t+1:t+d}|i,d)$ は潜在単語 $z_s = i$ から観測系列 $y_{t+1:t+d}$ が得られる確率を表しており，潜在単語 $z_s = i$ の潜在音素列 w_i を $w_i = (l_1, l_2, \ldots, l_{L_i})$ と置くと，$p(y_{t+1:t+d}|i,d)$ は下式のようにして計算できる．

$$p(y_{t+1:t+d}|i,d) = \sum_{r \in R^{(L_i,d)}} \prod_{k=1}^{L_i} p(r_k|l_k) \times \prod_{m=1}^{r_k} p(y_{t+m+\sum_{k'=1}^{k-1} r_{k'}} | l_k) \quad (4)$$

$$R^{(L_i,d)} = \left\{ r \in \{1,2,\ldots\}^{L_i} \,\middle|\, \sum_{k=1}^{L_i} r_k = d \right\} \quad (5)$$

ここで，$R^{(L_i,d)}$ は L_i 次元で要素の和が d になるような自然数ベクトル r の集合を示す．式 (4) は動的計画法を用いて効率的に計算することができる．簡単のため，式 (4) の $y_{t+1:t+d}$ を $y_{1:T}$ と置き換えて考えると，下式によって再帰的に求められる．また，下式を forward message と定義する．

$$\alpha_t(k) = \sum_{d'=1}^{t-k+1} \alpha_{t-d'}(k-1) p(d'|l_k) \prod_{t'=1}^{d'} p(y_{t-t'+1}|l_k) \quad (6)$$

$$\alpha_0(0) = 1 \quad (7)$$

以上より，$B_t(i), B^*_t(i)$ を用いて，潜在単語 $z_{s(t+1)}$ および潜在単語 $z_{s(t+1)}$ の継続時間 $D_{s(t+1)}$ をサンプリングできる．詳しくは， [Taniguchi 16a] を参照されたい．

ここで，単語数を N，観測データ y の長さを T，単語の最大継続長を d_{max}，単語の最大音素数を L_{max} としたとき，forward message $\alpha_t(k)$ の計算量は式 (6) より $\mathcal{O}(L_{max} d^3_{max})$ である．また，backward message の計算量は $\mathcal{O}(TN^2 L_{max} d^3_{max})$ となる．

3.2.2 潜在単語の潜在音素列のサンプリング

同じ潜在単語の領域としてセグメント化された各領域を $y^{1:K} = \{y^1, y^2, \ldots, y^K\}$ としたとき，観測データ $y^{1:K}$ の生成モデルは HDP-HSMM とみなすことができる．そのため，推定された単語の潜在音素のサンプリングでは HDP-HSMM を用いる．しかし，HDP-HSMM における潜在音素のサンプリングは確率的であるため，同じ潜在単語としてセグメント化された観測系列 $y^{1:K}$ の潜在音素列が同じになるとは限らない．そのため，SIR に基づいた推定手順を導入する．$y^{1:K}$ における観測データに同じ潜在音素列を共有させるため，各観測データ $y^{1:K}$ からの潜在音素列 w の事後確率 $p(w|y^{1:K})$ は下式のように定義される．

$$p(w|y^{1:K}) \propto p(w) p(y^{1:K}|w) \quad (8)$$

$$= p(w|y^j) p(y^j) \prod_{i \neq j}^{K} p(y^i|w) \quad (9)$$

ここで，$p(y^j)$ は観測データの尤度を表しており，HDP-HSMM を用いて容易に求められる．$p(y^j|w)$ は式 (4) と同じ手順で求められる．また，$p(w|y^j)$ の推定も HDP-HSMM によって行うことができる．以上より，$p(w|y^j)$ を提案分布，$p(y^j) \prod_{i \neq j}^{K} p(y^i|w)$ を重みとして考えることで，SIR と同様の方法でサンプリングが可能となる．

ここで，同じ単語としてセグメント化された領域の数を K，提案分布から得られた音素列の最大音素数を L_{max}，単語の最大継続長を d_{max} としたとき，SIR の計算量は $\mathcal{O}(L_{max} d^3_{max} K^2)$ となる．

3.2.3 HDP-HLM のパラメータ更新

各観測データに対応した潜在単語列と潜在単語に対応した音素列がそれぞれサンプリングされた後，π^{LM} 等の言語モデルの各パラメータを潜在単語列に基づいて更新する．同様に，単語モデルの各パラメータについても潜在音素列に基づいて更新する．音響モデルのパラメータについては，それぞれの観測データ y_t によって決定された隠れ状態 x_t から更新する．

4. NPB-DAA with lookup table

4.1 尤度計算におけるメッセージ計算

従来の NPB-DAA では，潜在音素列 $w_i = (l_1, l_2, \ldots, l_{L_i})$ から観測 $y_{1:T}$ が得られる確率を forward message を用いて計算していた．この時，式 (6) の $\alpha_{t-d'}(k-1)$，$p(r_k = d'|\omega_{l_k})$，$\prod_{t'=1}^{d'} p(y_{t-t'+1}|\theta_{l_k})$ を $d' = 1$ から $d' = t-k+1$ まで縦に並べたような列ベクトルを $\mathbb{A}_t^k$，$\mathbb{D}_t^k$，$\mathbb{L}_t^k$ と置く．

$$\mathbb{A}_t^k = \left[\begin{array}{ccc} \alpha_{t-1}(k-1) & \cdots & \alpha_{k-1}(k-1) \end{array} \right]^T \quad (10)$$

$$\mathbb{D}_t^k = \left[\begin{array}{ccc} p(1|\omega_{l_k}) & \cdots & p(t-k+1|\omega_{l_k}) \end{array} \right]^T \quad (11)$$

$$\mathbb{L}_t^k = \left[\begin{array}{ccc} p(y_t|\theta_{l_k}) & \cdots & \prod_{t'=1}^{t-k+1} p(y_{t-t'+1}|\theta_{l_k}) \end{array} \right]^T \quad (12)$$

この時，式 (6) は下式で表せる．

$$\alpha_t(k) = \left[\begin{array}{ccc} 1 & \cdots & 1 \end{array} \right] \cdot \left(\mathbb{A}_t^k \circ \mathbb{D}_t^k \circ \mathbb{L}_t^k \right) \quad (13)$$

$$(14)$$

ここで，演算子 $\circ$ は要素毎の積を表す．また，演算子 $\cdot$ は内積演算を表す．この時 $\alpha_{t+1}(k)$ は，下式で表される．

$$\alpha_{t+1}(k) = \left[\begin{array}{ccc} 1 & \cdots & 1 \end{array} \right] \cdot \left(\mathbb{A}_{t+1}^k \circ \mathbb{D}_{t+1}^k \circ \mathbb{L}_{t+1}^k \right) \quad (15)$$

ここで，$\mathbb{A}_{t+1}^k, \mathbb{D}_{t+1}^k, \mathbb{L}_{t+1}^k$ それぞれに注目すると，

$$\mathbb{A}_{t+1}^k = \left[\begin{array}{c} \alpha_t(k-1) \\ \mathbb{A}_t^k \end{array} \right] \quad (16)$$

$$\mathbb{D}_{t+1}^k = \left[\begin{array}{c} \mathbb{D}_t^k \\ p(t-k+2|\omega_{l_k}) \end{array} \right] \quad (17)$$

$$\mathbb{L}_{t+1}^k = p(y_{t+1}|\theta_{l_k}) \left[\begin{array}{c} 1 \\ \mathbb{L}_t^k \end{array} \right] \quad (18)$$

より，それぞれ $\alpha_t(k)$ の計算で求めた $\mathbb{A}_t^k, \mathbb{D}_t^k, \mathbb{L}_t^k$ を再利用できる．

よって，従来は forward message の計算量が $\mathcal{O}(L_{max} d_{max}^3)$ であったのに対し，計算量を $\mathcal{O}(L_{max} d_{max}^2)$ に削減できる．

4.2 SIR の重み計算

潜在音素列 w のサンプリングでは，SIR が用いられており，式 (9) と定義されている．この時，各観測系列 $y^{1:K}$ から提案分布を用いてサンプリングされた w の候補をそれぞれ $\overline{w}^{1:K}$ とする．また，それぞれの候補 $\overline{w}^{1:K}$ の重みは下式で求められた．この時，$p(y^i|\overline{w}^j)$ を $\mathbb{W}_{j,i}$ と置き，$\mathbb{W}_{j,i}$ を $i, j = 1, \ldots, K$ まで並べた行列を $\mathbb{W}$ と置く．

$$\mathrm{weight}(\overline{w}^j) = p(y^j) \prod_{i \neq j}^{K} \mathbb{W}_{j,i} \quad (19)$$

$$\mathbb{W}_{j,i} = p(y^i|\overline{w}^j) \quad (20)$$

$$\mathbb{W} = \left[\begin{array}{ccc} \mathbb{W}_{1,1} & \cdots & \mathbb{W}_{1,K} \\ \vdots & \ddots & \vdots \\ \mathbb{W}_{K,1} & \cdots & \mathbb{W}_{K,K} \end{array} \right] \quad (21)$$

ここで，$\mathbb{W}_{j,i}$ の各要素はそれぞれ (6) の再帰式によって求められる．また，行列 $\mathbb{W}$ において，重み計算に用いられる要素は $i \neq j$ の時のみであるため，行列 $\mathbb{W}$ のうち計算が必要な要素数は $K(K-1)$ 個である．しかし，候補 $\overline{w}^a$ と $\overline{w}^b$ が同じ音素列であった時，それぞれの重みは

$$\mathrm{weight}(\overline{w}^a) = p(y^a) \mathbb{W}_{a,b} \prod_{i \neq a,b}^{K} \mathbb{W}_{a,i} \quad (22)$$

$$\mathrm{weight}(\overline{w}^b) = p(y^b) \mathbb{W}_{b,a} \prod_{i \neq a,b}^{K} \mathbb{W}_{b,i} \quad (23)$$

として求められる．また，$\overline{w}^a = \overline{w}^b$ より $\prod_{i \neq a,b}^{K} \mathbb{W}_{a,i} = \prod_{i \neq a,b}^{K} \mathbb{W}_{b,i}$ となるため，計算結果を再利用できる．これにより，従来は計算回数が $K(K-1)$ 回であったのに対して，$\overline{w}^{1:K}$ のうち重複しないものの個数を n 個，重複するものの種類の数を u 個とすると，計算回数を $n(K-1)+uK$ 回まで削減できる．

よって，forward message の計算量を $\mathcal{O}(L_{max} d_{max}^2)$ としたとき，従来では計算量が $\mathcal{O}(L_{max} d_{max}^2 K^2)$ であったのに対し，計算量を $\mathcal{O}(L_{max} d_{max}^2 K(n+u))$ に削減できる．

5. 実行時間の比較実験

本実験では，従来の NPB-DAA と NPB-DAA with lookup table を用いて，実行時間の比較を行う．また，本実験で用いた NPB-DAA with lookup table のソースコードは GitHub[*2] に公開した．

5.1 実験条件

実験に用いるデータは，母音のみで構成された，実音声にして約 1 分 20 秒程度の音声データから抽出された，全 6311 フレームの 3 次元の特徴量を用いる．

また，本実験は Intel Xeon CPU E5-2630 v2 2.60GHz が 6 コア × 2CPU で構成された計算機を用いる．本実験では，weak-limit 近似における最大単語数および最大音素数をそれぞれ 10 個とした．

上記の条件において，ギブスサンプリング 100 イテレーションを 1 試行とし，NPB-DAA および NPB-DAA with lookup table それぞれ独立に 20 回試行する．

5.2 実験結果

表 1 に NPB-DAA および NPB-DAA with lookup table の実行時間の平均および最大値と最小値をそれぞれ示す．表 1 より，NPB-DAA と NPB-DAA with lookup table を比較すると，実行時間を約 90%削減できていることが確認できる．

*2 https://github.com/EmergentSystemLabStudent/NPB_DAA/tree/ozakiDevelop

表 1: 実行時間の比較

Method		Execution time[sec] iteration	trial
NPB-DAA with lookup table	MIN	**2.8**	**732.2**
	AVG	8.7	869.3
	MAX	18.9	1136.1
NPB-DAA	MIN	36.7	6237.6
	AVG	98.7	9869.8
	MAX	310.9	13417.7

表 2: 単語数上限の変化に伴う実行時間の変化

Method		Execution time per trial[sec] Maximum number of words 10	20	30
NPB-DAA with lookup table	MIN	**732.2**	**1365.5**	**2247.4**
	AVG	869.3	1652.8	2464.3
	MAX	1136.1	1897.4	2678.0
NPB-DAA	MIN	6237.6	15704.3	22596.3
	AVG	9869.8	16316.6	24757.9
	MAX	13417.7	17211.9	26286.0

表 3: 音素数上限の変化に伴う実行時間の変化

Method		Execution time per trial[sec] Maximum number of letters 10	20	30
NPB-DAA with lookup table	MIN	**732.2**	**739.2**	**885.3**
	AVG	869.3	992.1	1003.5
	MAX	1136.1	1158.7	1145.6
NPB-DAA	MIN	6237.6	9151.9	9871.0
	AVG	9869.8	10285.2	10435.5
	MAX	13417.7	11804.6	11064.9

次に，weak-limit 近似における単語数および音素数の上限値の変化による実行時間の変化を調査するため，単語数上限および音素数上限をそれぞれ 20 個・30 個と変化させたものをそれぞれ 3 試行ずつ計測した結果を表 2 および表 3 に示す．表 2 より，単語数上限の変化に対して，NPB-DAA および NPB-DAA with lookup table ともに大きな増加傾向が確認された．表 3 より，音素数上限の変化に対して，NPB-DAA および NPB-DAA with lookup table ともに比較的小さな増加傾向がみられた．表 2 および表 3 より，NPB-DAA と NPB-DAA with lookup table を比較すると，実行時間を約 90%削減できていることが確認できる．

6. まとめ

本研究では，NPB-DAA にルックアップテーブルを導入した NPB-DAA with lookup table を提案し，実行時間の削減を行った．ルックアップテーブルの導入により，従来は計算量が $\mathcal{O}(TN^2L_{max}d_{max}^3)$ であったのに対し，計算量を $\mathcal{O}(TN^2L_{max}d_{max}^2)$ に削減できた．また，実時間において実行時間を約 90%削減できているを確認した．今後の方針として，TIMIT コーパスを用いた教師なし語彙獲得を行い，NPB-DAA with lookup table の性能調査等を行う．また，現状の NPB-DAA および NPB-DAA with lookup table はバージョンの古いライブラリに強く依存しており，移植性が低いことが問題点として挙げられる．そのため，より広く NPB-DAA を用いてもらうため，各種ライブラリのバージョンアップと整理が今後の課題である．

参考文献

[Johnson 13] Johnson, Matthew J. and Willsky, Alan S.: Bayesian Nonparametric Hidden Semi-Markov Models, Journal of Machine Learning Research, Vol.14, pp.673–701, Feb.2013

[Mochihashi 09] Daichi Mochihashi, Takeshi Yamada and Naonori Ueda: Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling, Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1, pp.100–108, 2009

[Saffran 26] Saffran, Jenny R., Aslin, Richard N. and Newport, Elissa L.: Statistical Learning by 8-Month-Old Infants, American Association for the Advancement of Science, Vol.274, No.5294, pp.1926–1928, 1996.

[Tada 17] 多田 裕貴，幸 優佑，林　楓，萩原 良信，谷口 忠大: ノンパラメトリックベイズ二重分節解析器の TIDIGITS コーパスへの適用, The 31st Annual Conference of the Japanese Society for Artificial Intelligence, 2017

[Taniguchi 11] Tadahiro Taniguchi and Shogo Nagasaka: Double articulation analyzer for unsegmented human motion using Pitman-Yor language model and infinite hidden Markov model, 2011 IEEE/SICE International Symposium on System Integration, pp.250–255, Dec.2011

[Taniguchi 16a] Tadahiro Taniguchi, Shogo Nagasaka and Ryo Nakashima: Nonparametric Bayesian Double Articulation Analyzer for Direct Language Acquisition From Continuous Speech Signals, IEEE Transactions on Cognitive and Developmental Systems, Vol.8, No.3, pp.171–185, Sep.2016

[Taniguchi 16b] Tadahiro Taniguchi, Ryo Nakashima, Hailong Liu and Shogo Nagasaka: Double articulation analyzer with deep sparse autoencoder for unsupervised word discovery from speech signals, Advanced Robotics, Vol.30, No.11–12, pp.770–783, Apr.2016