

# 判別分析と知識学習法による中期インド・アリアン韻律の 言語学者並み解析システム構築

逢坂雄美

仙台電波工業高等専門学校

従来の韻律解析では、基本的韻律スキームとテキストデータの韻律スキームとのパターンマッチングにより、該当する韻律名が識別される。当該古文献の最も重要な文献の韻律識別率は、平均70～80%であり、最も古いテキストではせいぜい20%台である。従来と全く異なる新手法、判別分析とニューラルネットワークを併用することにより、その識別率を大幅にアップできた。従来解析不能な韻律に対して、その解析を援用できる情報を提供する。標準語のサンスクリットから著しく異なる韻律を抽出する新手法を提供する。又この手法は半詩偈を極めて効率よくかつ正確に詩脚分割できる。当研究はニューラルネットワークの適用成功の一例である。

## A New Meter Analysis in Middle Indo-Aryan on the Basis of Neuro Network Assisted by Discriminant Analysis

Yumi Ousaka

Sendai National College of Technology

In the meter analysis so far, its meter name is identified by the pattern match with the meter scheme of the basic meter and the text data. The rate of the identification of the meter of the most important old canons runs from 70 to 80% on the average, and at most 20% level in the oldest one. The identification rate is able to be improved greatly by using a quite in the past different new technique, the neural network assisted by the discriminant analysis. We can obtain information on the meter that cannot be analyzed so far. A new technique for extracting a remarkably different meter from the standard Sanskrit of is developed. Moreover, this technique can divide a half verse into two pādas extremely efficiently and accurately. This research shows one application success in neural network.

### 1. 序論

中期インド・アリアン語(B. C. 5-A. D. 10)の古文献は、仏教(パーリ語)とジャイナ教(プラークリット語)の聖典がその例に挙げられるように、世界の文化の貴重な宝庫であり、現代文明の重要な源流の一つである。初期仏教・ジャイナ教は厳格なカースト制度のバラモン教に対抗する形で同時期に発生したので、お互いに関係を持ちながらその教理等を深化させた。それゆえ、今日では初期仏教の研究をする上でジャイナ教の研究は不可欠であると認識されている。

中期インド・アリアン語の研究には、韻律解析・文法解析・語彙と構文論等の系統的な研究が不可欠であることは周知の事実である。テキストの韻律解析は批判的校訂本作成に不可欠であり、単語の索引は翻訳に当たって非常な手助けとなり、単語の逆順索引は文法構造解析に必要であり、詩脚の正順また

は逆順索引はテキストの正しい読みの確立と並行詩脚の検索に当たって大変重要である。この目的達成に当たって大量のデータつまり多くのテキストを処理する必要がある。幸いにも、テキスト解析に計算機を使用することによって、テキストの系統的な研究が大幅に進展するものと期待されている。我々これまで上記のツールを開発し、研究に活用してきた[1]。

韻律解析プログラムでは、現在知られている基本的韻律スキームとテキストデータから計算された韻律スキームとのパターンマッチングにより、該当する韻律名が識別される。当該古文献の最も重要な文献の詩脚索引作成過程で韻律解析を実行した[2]が、平均的にはその70～80%を分類できた程度である(中には20%台の分類成功率のテキストもある)。言語研究進展の為には、この識別率を上げることが不可欠である。

本論文では、従来と全く異なる新手法、判別分析

と知識学習法を適用して、中期インド・アリアン韻律の言語学者並みに解析できるシステムの構築について議論する。

中期インド・アリアン語の韻律は、3種類、Akṣaracchandas, Mātrācchandas, Gaṇacchandas に分類される。その中の主要な韻律名として、Śloka, Triṣṭubh, Jagatī, Āryā, Vaitāliya, Aupacchandasaka がある。これらの韻律はすべて、特有な韻律スキームを持つ(次節参照)。

従来の韻律解析の手法で分類できなかった韻律は、大きく分けて2形式ある。一つは、現在知られている韻律では分類不能なものである。この韻文には、これまで未識別の新しい韻律(詩形破格=poetic license)が隠されている可能性がある。もう一つは、割り当て可能な韻律が何種類も現れるものである。これについてはこれまで、可能なすべての韻律パターンを表示(Alternative representation)するようにしている。正しい韻律分類はその表示の中から言語学者が選択するようになってきている。相当詳しい知識を有する言語学者のみが正しい選択をでき、この選択自体が言語学的に難しい問題である。

基本的な韻律スキームとテキストデータとのパターンマッチングを用いた従来の韻律解析における、2つの問題点(poetic license と Alternative representation)を解決するために、多量の韻律解析結果を印刷して、手作業で新しいpoetic license等解明に重要で適切な情報を抽出することを試みたが、不可能であった。この解決のためには、全く異なる新規手法開発の必要性を認識し、出力結果の分類にクラスター分析・主成分分析・ニューラルネットワークの技法を夫々単独で活用したが、分類の確定には至らなかった。

その後、基本的な韻律スキームの特性をより詳しく吟味した。適切な音節数を選ぶと、基本韻律スキームは小グループに分割され、その小グループの間には相関が弱くなり、近似的に直交している可能性がある、ことを見いだした。種々の多変量解析法の中から判別分析を試み、異なる小グループに属する基本韻律スキームは近似的に直交し、小グループに分類できる可能性がある、ことを突き止めた。この手法は結果を分かりやすい形に表・図示できる利点があり、かつてパーリ文献とジャイナ教文献の分類に活用した[3]。しかし残念ながら、この手法だけでは小グループ内での韻律形式の確定には至らなかつ

た。

小グループに分解した後では、ニューラルネットワークの知識学習法による分類の成功する可能性がより高くなると考えられる。教師あり学習ニューラルネットワークのプログラミング手法は、学習パターンの変化への対応が柔軟でネットワークの再学習を簡単にでき、学習精度を上げることも容易にできる。記憶し易い口伝形式である(基本)韻律スキームは10数個前後の0,1,2の数字列により表現できる。その解析は教師あり学習ニューラルネットワークに非常に適している。これらの特性を活用して、基本韻律スキームの学習について吟味したところ、ほぼ完全に学習できることを突き止めた。これにより、学習結果に基づき、各聖典の各韻律を効率よくかつ精度よく分析(予測)できる。当分析ではその結果が数値化されるので、基本韻律からの偏差を容易に評価できる。この結果、2つの問題点(poetic license と Alternative representation)を解決できる可能性がでてきた。一流の言語学者が、高度な知識を活用してようやく分類・把握可能な難しい韻律解析結果を、当該手法を使って得られる可能性がある。又、当該手法を確立し、結果の吟味をすることにより、古代インド文化の経験に基づく韻律分類は、最新の知識情報処理法で漸く分類できるほど高度に洗練されており、精神文化基盤がきわめて高度なことの一つの例証になるのではと、予測している。韻を踏んで記憶しやすい口伝形式の古来の他の言語分野の研究にも、当該手法を拡張できる可能性があるのではと思われる。

なおここで今後の議論を円滑にする為に、中期インド・アリアンの韻律研究の基本用語について簡単にまとめておく。中期インド・アリアンに属するアルダ・マガダ語のアルファベットは40文字からなる。1番目から8番目の文字は母音で、その内5文字 ā, i, ū, e, o は長母音、3文字 a, i, u は短母音である。残りの32文字は子音である。パーリ語・仏教混淆梵語は夫々42, 55文字であるが、その構成はアルダ・マガダ語とほぼ同様である。詩偈(しげ)は韻律の基本構成要素で、殆どが2行よりなり、各々を第1半詩偈・第2半詩偈と呼ぶ。また、各半詩偈は2つに分けられ、それらを詩脚(パーダ)と呼ぶ。韻律の研究には、まず最初に半詩偈ごとの韻律スキーム(音節パターン、長短音配列)を計算する。つまり長母音(‘-)と短母音(‘˘)配列を求める。本論

文では長母音を 2, 短母音を 1 に割り当て, その他空白を 0 として数値化して, 処理する. 韻律スキームは次のように決定される. (a) 全長音は, いつも長母音として読まれる. (b) 短母音の後に 2 つの子音を伴う場合は長母音として読まれる. (c) 特殊文字 'm' を後に伴う短母音はいつも長母音として読まれる. (d) 上の (b) と (c) の条件以外の短母音は常に短母音として読まれる. ところで, 音節数は長母音と短母音の総数を表わし, 音量数とは短母音を 1 とし長母音を 2 と数えた時の総数を示す. 音節数又は音量数が, 韻律の種類を規定するのに重要な役割を果たす.

## § 2 基本的韻律スキームの数値化と判別分析

中期インド・アリアンの韻律は Akṣaracchandās は, 音節数によって分類される. 例えばその中の主要な韻律名, Śloka は半詩偈に 16 音節を含み, Triṣṭubh と Jagatī はパーダにそれぞれ 11, 12 音節を含む韻律である. Gaṇacchandās は, 半詩偈中の音量数によって規定され, その典型的な韻律, Āryā の音量数は, 26, 27, 29, 30 である. その音節数は一つに固定されるのではなく, ある範囲に分布する. Mātrāchandās は, パーダに含まれる音量数によって規定される. 半詩偈の音量数が 28~30 のものを Vaitāliya, 30~34 のものを Aupacchandāsaka と呼ぶ. これらの韻律はすべて, 特有な韻律スキームを持つ (表 1 参照).

判別分析あるいはニューロネットワークを適用するには, これらのスキームを数値化する必要がある. 長母音を 2, 短母音を 1 に割り当て, その他に空白を 0 として数値化する. 本論文では, 半詩偈ではなくパーダのスキームを数値化して, 以下の解析に使用する. パーダごとに数値化することにより, パーダの組み合わせ状況を調べることができる. この組み合わせはパーダのパッチワークとして知られている. 3 大韻律の各韻律において, その構造は異なるが同じ音節数を持つ場合がある. 同一の音節数の中に入っている基本的韻律ごとに組み分けの番号 (表 2 の Classification No.1) を割り当て, この番号を識別標識に採用する. 表 2 から分かるように全部で 33 の識別番号がある.

種々の多変量解析法の中から判別分析を試みた. この手法では結果を, 分かりやすい形に表・図示できる利点がある. 表・図形上で分かり易く結果を表現できると, 今後の論理展開も容易になる. 表 1 を数値化

§ 2 では, 基本的韻律スキームの数値化と, それに判別分析を適用した時に得られる小グループ分割について議論する. § 3 では小グループに分割された基本スキームのニューラルネットワークによる分類・学習について議論する. § 4 では得られた学習データに基づく実際の聖典の韻律の, ニューラルネットワークによる分類について議論する. この分析では, 各聖典の実際の韻文を入力する. プログラム中で長短の韻律スキームを求め, その後にデータ解析用に 0, 1, 2 の数値列に変換したうえで, 自動的に分類が実行される. § 5 では, 今後の問題点等について議論する.

し, 重複しているものを省くと, 全部で 1140 個分の韻律スキームのデータが作成される. 判別分析への入力データとしては, それぞれの韻律スキームに識別番号を付加したものが使用される.

判別分析には統計解析ソフト SPSS を使用した. 得られた結果を表 3 に示す. 表の 1 列目は元の入力データの識別番号を表し, 1 行目は予測グループ割当の識別番号を表している. 2 行 3 列目の数値データ 5 は, 識別番号 2 の 5 つの入力データが識別番号 3 の韻律スキームに分類されることを示している. この表から分かるように, 7 行 4 列目と 19 行 15 列目に夫々 1 つのデータが存在するが, 33 の識別番号をほぼ 6 個の小グループに分類できる. 判別分析ではうまく分類ができないこの 2 個のデータは, 次節のニューラルネットワーク処理において, 何らの異常を引き起こさないことを確認している. それぞれの小グループは以下の識別番号と音節数より構成されている: 小グループ (1~5) には音節数 6 から 9 の韻律が所属する; (6~10) には音節数 10 の韻律が所属する; (11~16) には音節数 11 の韻律が所属する; (17~22) には音節数 12 の韻律が所属する; (23~25) には音節数 13 の韻律が所属する; (26~33) には音節数 14 から 17 の韻律が所属する. 音節数が本質的に重要な役割を果たしているといえる. このように, 判別処理を適用して, 異なる小グループに属するスキーム (Original group) を, 近似的に直交するように整理できた. 表 3 の Sum の欄の数字はそれぞれの識別番号グループのデータ数を表す.

なお, 表 3 の結果を詳しく吟味すると, 最後のグループはより小さな 4 つのグループに分割されるこ

表1. 中期インド・アリアン韻律スキーム

	Metrical scheme	Syllables or morae number (SN or MN)	Representation
Śloka	$uuuu v\text{---}u uuuu v\text{---}u $ $u\text{---}u uuuu uuuu v\text{---}u $ & $uu\text{---} uuuu uuuu v\text{---}u $ $u\text{---}u uuuu uuuu v\text{---}u $ , $u\text{---}u u\text{---}u uuuu v\text{---}u $	SN in a h.v. =16	Śloka1 Śloka2 Śloka3, Śloka4 ŚlokaM, ŚlokaEE, ŚlokaE4, ŚlokaC
	Miscellaneous		
Modified Śloka	Modification of 17 syllables into 16 syllables by taking into account of prefixes	SN in a h.v. =17	Śloka1-4, ŚlokaM ŚlokaEE, ŚlokaE4
	Modification of 17 syllables into 16 syllables by transformation of the consecutive two short vowels into one long vowel	SN in a h.v. =17	Śloka1-4, ŚlokaM, Examine1
Triṣṭubh	$u\text{---}u u\text{---}u u\text{---}u u\text{---}u $ , $u\text{---}u u\text{---}u u\text{---}u u\text{---}u $ $u\text{---}u u\text{---}u u\text{---}u u\text{---}u $ , $u\text{---}u u\text{---}u u\text{---}u u\text{---}u $ $u\text{---}u uuu u\text{---}u u\text{---}u $	SN in a pāda =11	Triṣṭubh
Jagatī	$u\text{---}u u\text{---}u u\text{---}u u\text{---}u $ or $uuu uuu uuu uuu $	=12	Jagatī
	Their combinations	SN in a h.v.=23	Triṣṭubh+Jagatī, etc.
	Miscellaneous	=22, 23 and 24	Tcheck, etc.
Āryā (in units of a h.v.)	$uu   uu\text{---}u   uu    uu\text{---}u   uu   u   uu   u   $	MN in a h.v. =26	Aryā2
	$uu   uu\text{---}u   uu    uu\text{---}u   uu   u   uu   -   $	=27	Aryā2
	$uu   uu\text{---}u   uu    uu\text{---}u   uu   uu   uu   u   $	=29	Aryā1
	$uu   uu\text{---}u   uu    uu\text{---}u   uu   uu   uu   -   $	=30	Aryā1
	Miscellaneous	26, 27, 29 and 30	Aryā3, Examine2
Āryā (in units of two h.v.'s)	Metrical pattern of Āryā 2 to the 1st h.v. and Āryā 1 to the 2nd h.v.	MN=(29 or 30) or (26 or 27)	New Āryā
	Metrical pattern of Āryā 2 to both of the 1st h.v. and the 2nd h.v.	MN=(29 or 30)	Gīti
	Metrical pattern of Āryā 1 to the 1st h.v. and Āryā 2 to the 2nd h.v.	MN=(26 or 27) or (29 or 30)	Udgīti
	Metrical pattern of Āryā 1 to both of the 1st h.v. and the 2nd h.v.	MN=(26 or 27)	Upagīti
Old Āryā	$uu   uu\text{---}u   uu    u\text{---}   uu   uu   uu   u   $ $uu   uu\text{---}u   uu    uu\text{---}u   uu   uu   uu   u   $ $uu   uu\text{---}u   uu    u\text{---}u   uu   uu   uu   u   $	MN= (28 or 29) (29 or 30) (30 or 31)	Old Āryā
	Miscellaneous	(28-31)	OA.Examine
Vaitāliya	(Odd pāda) $w uuu uuuu $ (Even pāda) $w\text{---} uuu uuuu $	MN in a pāda =13-16	(Classical Sanskrit) C.Vait. + C.Vait.
Aupacch.	$w uuu uuuu $ $w\text{---} uuu uuuu $	=14-18	C.Aupa.+ C.Aupa.
	Their combinations	MN in a h. v. =26-34	C.Vait. + C.Aupa., reverse of it, etc.
	Miscellaneous	=26-34	CA.Exam., etc.
Vaitāliya	(Odd pāda) 6 morae  $uuuu$   (Even pāda) 8 morae  $uuuu$	MN in a h. v. =28-30	(Middle Indo-Aryan) Vait.
	Aupacch. 6 morae  $uuuu$   8 morae  $uuuu$	=30-34	Aup.
Their combinations	6 morae  $uuuu$   8 morae  $uuuu$	=29-32	Vait. + Aup.,
	6 morae  $uuuu$   8 morae  $uuuu$	=29-32	Aup. + Vait.
	Miscellaneous	=28-34	A.Exam.+Aup., etc.

表2. 判別分析とニューラルネットワーク分析のための識別データ. 韻律名, その簡略表記号, パーダの音節数、識別番号、ニューラルネットワーク分析結果の精度 (2つの教師データの判別精度), 計算打ち切り総誤差, 当該小グループの入力データ数を示している.

Metre Name	Abbreviation in calculation	Syllable No in pada	Classification No.1	Precision for Estimation	Precision for Estimation of Morae	Precision for Calculation	Input Nos. (Total 1140)
Sloka (a few Old Aryās)	Sloka	8	1	(-.053,.062)	(-.06,.066)	0.00005	159(-1)
Arya	Arya0	6-9	2				
Arya	Arya1_2	9	3				
Vait, & Aup.	Vait_Aupa	9	4				
Vait, & Aup. in MIA	Middle_Vait_Aupa	9	5				
Arya	Arya0	10	6	(-0.02, 0.02)	(-0.039,0.036)	0.00001	111
Arya	Arya1_2	10	7				
Old Arya	Old_Arya	10	8				
Vait, & Aup.	Vait_Aupa	10	9				
Vait, & Aup. in MIA	Middle_Vait_Aupa	10	10				
Tristubh	Tristubh	11	11	(-.04, .04)	(-1.003,.1003)	0.00002	219
Arya	Arya0	11	12				
Arya	Arya1_2_3_4	11	13				
Old Arya	Old_Arya	11	14				
Vait, & Aup. in both CS & MIA	Vait_Aupa_or_Middle_Aupa	11	15				
Vait, & Aup. in MIA	Middle_Vait_Aupa	11	16				
Jagati	Jagati	12	17	(-.014, .013)	(-.0193,.145)	0.00002	269
Arya	Arya0	12	18				
Arya	Arya(or_Old_Arya)	12	19				
Old Arya	Old_Arya	12	20				
Vait. in CS and, Aupa. In MIA	Vait_or_Middle_Aupa	12	21				
Vait, & Aup. in MIA	Middle_Vait_Aupa	12	22				
Arya & Old Arya	Arya_or_Old_Arya	13	23	(-.0017, .0014)	(-.069,.054)	0.00002	172
Old Arya	Old_Arya	13	24				
Vait, & Aup. in MIA	Middle_Vait_Aupa	13	25				
Arya & Old Arya	Arya_or_Old_Arya	14	26	(-.03, .03)	(-.063,.063)	0.00001	210
Old Arya	Old_Arya	14	27				
Vait, & Aup. in MIA	Middle_Vait_Aupa	14	28				
Arya & Old Arya	Arya_or_Old_Arya	15	29				
Old Arya	Old_Arya	15	30				
Arya & Old Arya	Arya(or_Old_Arya)	16	31				
Old Arya	Old_Arya	16	32				
Arya	Arya	17	33				

表3. 判別分析による基本韻律スキームの分類 (小グループへの分類)

Original group	Predicted group																																	Sum
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
1	76	4	0	0	0																													80
2	15	21	5	3	4																													48
3	0	0	11	1	2																													14
4	0	0	0	2	0																													2
5	0	2	2	0	11																													15
6					19	0	0	0	0																								19	
7				1	25	2	4	6																									39	
8				0	0	3	0	0																									3	
9				0	2	0	4	0																									6	
10				3	4	1	11	25																									44	
11									48	0	2	1	5	0																			56	
12									0	7	0	0	0	0																			7	
13									1	4	39	13	5	6																			68	
14									0	0	0	10	2	0																			12	
15									0	0	0	0	6	2																			8	
16									8	1	7	6	9	37																			68	
17															80	0	0	0	0	0	0												80	
18															0	1	0	0	0	0	0												1	
19															0	14	40	23	2	10													90	
20															0	0	0	22	2	6													30	
21															0	0	0	0	4	0													4	
22															10	0	0	4	0	50													64	
23																					50	36	18										104	
24																					0	38	2										40	
25																					2	0	26										28	
26																					59	23	10	0	0	0	0	0	0	0	0	0	92	
27																					0	30	0	0	0	0	0	0	0	0	0	0	30	
28																					0	0	4	0	0	0	0	0	0	0	0	0	4	
29																					0	0	0	42	10	0	0	0	0	0	0	0	52	
30																					0	0	0	0	12	0	0	0	0	0	0	0	12	
31																					0	0	0	0	0	14	2	0	0	0	0	0	16	
32																					0	0	0	0	0	0	0	2	0	0	0	0	2	
33																					0	0	0	0	0	0	0	0	0	0	0	0	2	2

とが分かる。分割数を増やした場合、その結果として目的とする韻律解析が進展するならば、分割数を増やす意義がある。しかし、そのような状態にはなっていない。この分割数の状態で、ニューラルネットワーク処理による学習が十分に収束する。それゆえ、本研究では基本的な韻律スキーム全体を6個の小グループに分割し、次節以下の処理を適用する。

これまで、韻律スキーム全体に対してニューラルネットワークを適用しても、収束する結果が得られなかった。しかし、近似的に直交する小グループに分解した後で、ニューラルネットワークの知識学習を活用すると、韻律スキームをほぼ完全に学習できる可能性が高くなる。

### §3 ニューラルネットワークによる基本的韻律スキームの学習

本節では、前説で求めた近似的に直交する6個の小グループに対して、ニューラルネットワークの知識学習法を活用する。小グループごとの学習により、ニューラルネットワークの結合強度ベクトル・閾値を評価する。これまで藤木らが定式化・作成したニューラルネットワーク・プログラム（使用言語はCまたはC++）を使用する[4]。当手法は、その構造や学習アルゴリズムが単純で、C（またはC++）言語で書かれていることもあって、種々のOSのパーソナルコンピュータ(PC)へ容易に移植できる。

本論文で使用しているニューラルネットワークは、図1に示すように入力層、中間層1層、出力層の計3層からなる階層型構造である。教師データは2種類使用する。1つは、前節の判別分析でも使用した、表2の識別番号(Classification No.1)である。もう一つは基本韻律スキームのモーラ数である。このモーラ数はスキームごとに簡単に確定する数値であり、ニ

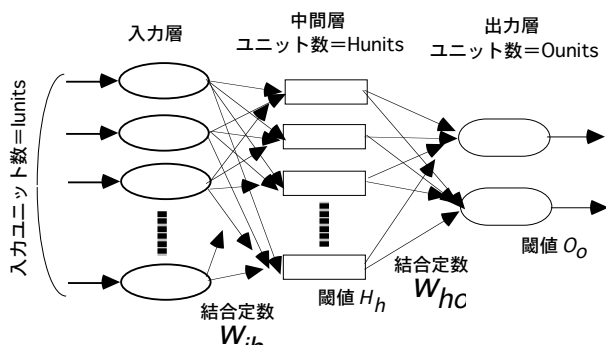


図1. 階層型ニューラルネットワーク  
ニューラルネットワークの収束状況判定には良い指標

である。この結果として、出力層のユニット数は2になる。なお、中間層及び出力層からのニューラル出力は双曲線関数 $\tanh(x)$ を使用している。

各小グループごとの入力データ数、入力ユニット数(Iunits)、中間層ユニット数(Hunits)を表4に示す。学習打ち切りの誤差の2乗、中間層ユニット数は試行錯誤の結果、最適なものを選択した。各小グループごとの採択した誤差の2乗は、表3のPrecision for Calculationの項目に示している。又、各小グループにおける2つの教師データ（識別番号とモーラ数）に対する認識の誤差の範囲を表3の第5,6列目に示している。この数値範囲から分かるように、整数値の教師データを十分よい精度で識別できる。なお、基本韻律スキームの学習において、第1グループの1つのデータは、このグループ認識においてその特性が特異すぎるので除外した。表3と4の(-1)はこのことを意味している。

表4. ニューラルネットワークデータ

Small group	Input data	Iunits	Hunits
1	159(-1)	9	15
2	111	10	15
3	219	11	15
4	269	12	15
5	172	13	15
6	210	17	17

これらの小グループごとのニューラルネットワークの学習において、十分な精度に達するまでに数十万回程度の学習回数を要する。実際の入力テキストごとに、認識の為の学習をするのはあまり実用的でない。それゆえ、各小グループごとに、入力ユニット数・中間層ユニット数、入力層の結合定数、中間層の結合定数と閾値、出力層の閾値(図1参照)をファイル出力しておき、実際のテキスト認識にこのファイル出力されたデータを引用する。言語学者による認識ソフトウェア使用を考えた場合、この方が実用に適する。

### §4 ニューラルネットワークによる韻律解析

予測処理のプログラムを、ニューラルネットワーク本体のプログラムからは切り離し、単体にする。前節の段階で得られた結合強度ベクトルデータ等を取り込むことにより、この予測処理プログラムを作成できる。

予測処理プログラムのファイル入出力ダイアログを図2に示す。最初のダイアログボックスには、解析

の為のテキストが格納されたファイルを入力し、2番目のボックスには出力結果を格納するファイル名を入力する。使用しているプラットフォームを選択し、Runボタンをクリックすると、解析プログラムが実行される。従来の我々の開発したツールと同様に、多くのプラットフォーム(Windows PC, Mac PC, Linux PC)で使えるように、プログラミング言語はジャバを使用している。この図を見て分かるように非常に分かりやすいインターフェースを備えている。

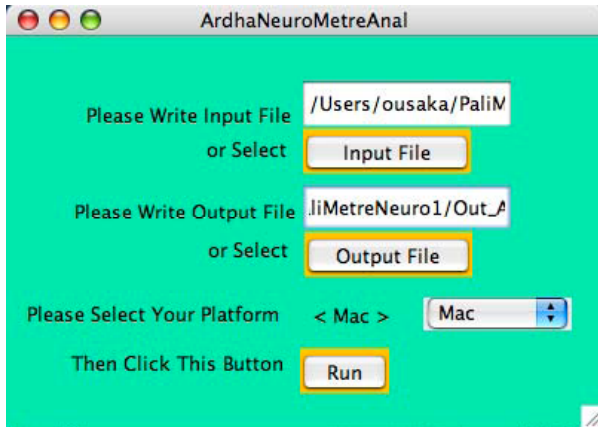


図2. 韻律解析ダイアログ

実際の入力ファイル例を図3に示す。これは言語学的に普通に作成・使用されているテキスト形式であり、何ら特別のものではない。プログラムでの内部処理で、各詩偈を長短の韻律スキームに変換し、各長短記号を数値化したうえで、韻律数の評価等がされて、予測処理が実行される。入力テキストを韻律に変換する処理では、従来の韻律解析ツールをニューロネットワーク解析用に変形している。その後ニューラルネットワークの予測処理が実行される。予測処理の際に、各詩偈を前節で評価した小グループへ適切に割り当てることは、自動的になされる。以上のようにプログラム使用者が、予測処理の為にテキストを作成すること以上の特別な作業を必要としない。

```
1. aṇupuvveṇa vimohāim | jāim irā samāsajja /
   vasumanto maimanto | savvaṃ naccā aṇeṭisaṃ //8.1//
2. duvīhaṃ pi viittāṇaṃ | bud ā ammassa pāragā /
   aṇupuvvie saṃkhāe | kammaṇāo tiuttāi. //8.2//
3. kasāe payaṇue kiccā | app' āhāro tiikkhae; /
   aha bhikkhū gilāejjā | āhārass' eva antiyam. //8.3//
4. jīviyaṃ nābhikaṅkhejjā | maraṇaṃ no vi patthae: /
   duhao vi na sajjejjā | jīvie maraṇe tahā. //8.4//
.....
.....
```

図3. 入力例

chap.81			
"EXAMINE1"	υ υ υ υ υ υ υ υ υ υ υ	17	28
"O.EXAMINE3"	υ υ υ υ υ υ υ υ υ υ υ	17	28
1 aṇupuvveṇa vimohāim   jāim dhīrā samāsajja /			
ŚlokaM	υ υ υ υ   υ υ υ υ   υ υ υ υ   υ υ υ υ	16	26
vasumanto maimanto   savvaṃ naccā aṇeṭisaṃ //81//			

図4. 従来の解析結果

図4は従来の解析の1例である。解析結果の表示では、一行目に割り当てられた韻律名、求められた韻律スキーム、音節数、音量数が表示され、その下の行には韻文テキストを表示する。図の第1半詩偈は、現在知られている韻律によって分類不能なものであることを表示している。また、第2半詩偈ではślokaMと分類されているが、通常のślokaと異なっている。韻律スキームにおける特殊マーク“|”はパーダ仕切りを表す。このような状態のパーダをニューラルネットワークで識別すると、図5ようになる。第1半詩偈の後半のパーダは通常のślokaになる。これが序論で指摘したAlternative representationに対する一つの解答であろうまた正しい詩脚分割結果が得られた。従来の手法では、第2半詩偈にślokaMの表示がなされているが、本手法では第2半詩偈はAryāとślokaに分類できる可能性があることを示している。

```
.....
1. aṇupuvveṇa vimohāim | jāim dhīrā samāsajja /
.....
pada1 = aṇupuvveṇa vimohāim   pada2 = jāim dhīrā samāsajja
Syll = υ υ υ υ υ υ υ υ υ υ : Syll. No = 9 Syll = υ υ υ υ υ υ υ υ υ : Syll. No = 8
Neuro Value1 for pada 1 = 1.2384336;           for pada 2 = 1.0335243
Neuro Value2 for pada 1 = 14.03074;           for pada 2 = 13.976429;
Neuro Metre for pada 1 = None! ; for pada 2 = Sloka

.....
vasumanto maimanto | savvaṃ naccā aṇeṭisaṃ //8.1//
.....
pada1 = vasumanto maimanto   pada2 = savvaṃ naccā aṇeṭisaṃ
Syll = υ υ υ υ υ υ υ υ υ υ : Syll. No = 8   Syll = υ υ υ υ υ υ υ υ υ υ : Syll. No = 8
Neuro Value1 for pada 1 = 1.9917974;           for pada 2 = 1.0211112
Neuro Value2 for pada 1 = 11.99625;           for pada 2 = 13.962445;
Neuro Metre for pada 1 = Arya0;   for pada 2 = Sloka
```

図5. 解析結果

異なる韻律の組み合わせ、つまり韻律のパッチワークの可能性を明示している。このように、従来の韻律解析が不能な例においても、本手法を使うと有益な情報を得ることができる。

なお、当然のことながら韻律の識別率が上がり、高々20%台の識別率であったAyarangaでは、80%台まで上昇した。又、他のテキストについて、今後詳しく吟味していく予定である。その解析が進むことにより、序論で指摘したpoetic licenseについても、より詳しい知見が得られる可能性がある。

## § 5 討論

従来の韻律解析では、基本的韻律スキームとテキストデータの韻律スキームとのパターンマッチングにより、該当する韻律名が識別される。当該古文書の最も重要な文献の韻律識別率は、平均70~80%であり、最も古いテキストではせいぜい20%台である。従来と全く異なる新手法、判別分析とニューラルネットワークを併用することにより、その識別率を大幅にアップできた。従来解析不能な韻律に対して、その解析を援用できる情報を提供する。「歴史的背景と異なる地域性を反映して、標準語のサンスクリットから著しく異なる韻律を抽出できる」新たな手法を提供する可能性がある。又この手法は半詩偈を極めて効率よく、かつ正確に詩脚分割できる。この新手法で開発した解析ツールは、言語学者自身が抵抗感なく使用できるようなインターフェースを備えている。得られた解析ツールを、刊行物・インターネット等で公表して、当該分野の研究のさらなる発展を期する。

本論文で議論された新手法は、ニューラルネットワークの言語解析への確かな適用成功の1例であろう。当該言語分野では、これらの手法は世界でもまだ手つかずの全く新しいものであり、かつ同族のインド・ヨーロッパ語族（ギリシャ語・アルメニア語等）への拡張可能性も含んでいる。このことは、この手法の適用範囲の拡張にも繋がっていく可能性を示している。

## 謝辞

この研究遂行に当たって適切な助言をしていただきました仙台電波高専・藤木なほみ教授に感謝いたします。本研究は2005年度日本学術振興会・科学研究費補

助金、基盤研究(B) (No. 16320009)の助成の下に遂行したことを付記し、感謝致します

## 参考文献

- [1] Y. Ousaka, Automatic Analysis of the Canon in Middle Indo-Aryan by Personal Computer II in both Japanese and English, The Chuo Academic Research Institute (Tokyo, 2005) 85pp.
- [2] 逢坂雄美, 「中期インド・アリアン聖典の統計解析」, 科学研究費・特定領域研究『人文科学とコンピュータ』数量的分析計画研究班 (1998年度) シンポジウム「人文科学における数量的分析(4)」, pp. 31-38
- [3] Y. Ousaka and M. Yamazaki, Genealogical Classification of Saddharmapundarika Manuscripts Based on Many-Variable Analysis, Literary and Linguistic Computing, Vol.17, No.2 (2002) pp.193-206
- [4] S. Fujiki and N. Fujiki, A Learning of Stochastic Feed-Forward Neural Network, J. Phys. Jpn., 164 (1995) 757., 馬場庸二, [カルバック測度を用いたエラーバックプロパゲーション学習則の学習及び認識能力の評価], 平成1年度仙台電波高専専攻研究