

TR-IT-0134

音声認識誤り発生過程の
モデル化に関する研究
A study on the modeling
of recognition error occurrence

森川 恵美
Morikawa Emi

中村 篤
Nakamura Atsushi

1995.9.

音声パターンの多様化などのため音声の自動認識は困難である。従って現在の音声認識システムによる音声認識結果には認識誤りが含まれている。この誤り発生過程を何らかの単純なモデルで捉え、それに基づく擬似環境を構築することにより認識理解系各部の所要性能明確化、簡便な実験環境の確保などが期待できる。そこで本研究では音声認識システムにおけるエラー発生過程のモデル化を行い音素ラベル列から実際の音声認識動作をシミュレートすることを考える。本稿ではキーワードスポッティングにおけるエラー発生過程のモデル化と、得られたモデルを用いた擬似スポッティング結果発生プログラム、及びその実行結果について述べる。

©ATR 音声翻訳通信研究所

©ATR Interpreting Telecommunications Research Laboratories

1 はじめに

音声パターンの多様化などのため音声の自動認識は困難である。従って現在の音声認識システムによる音声認識結果には認識誤りが含まれている。この誤り発生の過程を何らかの単純なモデルで捉え、それに基づく擬似環境を構築することにより認識理解系各部の所要性能明確化、簡便な実験環境の確保などが期待できる。そこで本研究では音声認識システムにおけるエラー発生過程のモデル化を行い音素ラベル列から実際の音声認識動作をシミュレートすることを考える。

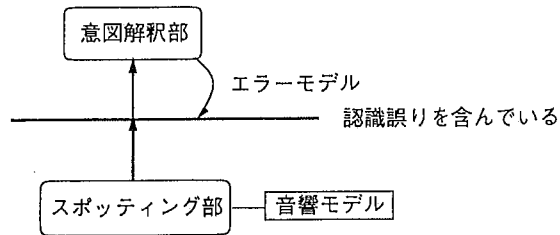


図 1: 意図解釈部から見たエラーモデル

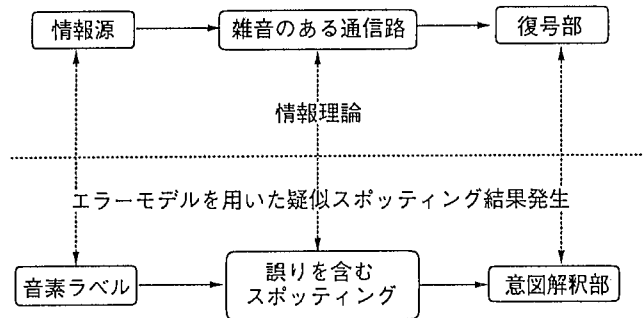


図 2: 疑似スポッティング結果発生過程の情報理論的解釈

キーワードスポッティングにおけるエラー発生過程のモデル化を行い、音素ラベル列から実際の音声認識動作をシミュレートすることにより

- 意図解釈部の評価実験データの生成
- スポッティング誤り特性の可視化

などが可能になる。

本稿ではキーワードスポッティングにおけるエラー発生過程のモデル化および得られたモデルを用いて作成した疑似スポッティング結果発生プログラムとその実行結果について述べる。

2 モデル

2.1 エラー発生過程

キーワードスポッティングにおけるエラーとしては次の2種類が考えられる。

- キーワード脱落誤り
本来認識されるはずであるキーワードが認識されない
- キーワード湧き出し誤り
本来認識されるはずのないキーワードが認識されてしまう

これらの認識誤り率はそれぞれ次のように評価される。

まず、キーワード脱落誤り率は

$$\text{脱落誤り率} = \frac{\text{脱落誤り数}}{\text{キーワード出現数}}$$

で評価され、脱落誤りはキーワード出現毎にある確率で発生する。これに対し湧き出し誤り率は

$$\text{湧き出し誤り率} = \frac{\text{湧き出し誤り数}}{\text{キーワード数}} / \text{単位時間}^1$$

で評価され、入力音声の音響的な特徴に依存して発生する。これをランダムな発生として近似するとポアソン発生モデルの適用が可能となる。ポアソン発生モデルはランダム事象を取り扱うための代表的なモデルであり、単位時間毎の事象発生数がポアソン分布にしたがい、事象発生間隔が指数分布にしたがう特徴をもっている。

ポアソン発生モデルは

- トラフィック工学
- 待ち行列理論
- 信頼性工学

の分野に一般的に応用されている。

2.2 score の推定

$$\text{score} = \sum_s \left\{ \sum_i^N \log \frac{1}{\sqrt{2\pi}\sigma_i} \right\} - \sum_s \left\{ \sum_i^N \frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right\}$$

N : 特徴ベクトルの次元数 (= 34)

S : Viterbi path 上の状態系列
(= フレーム数)

¹通常、単位時間=1時間

ここでまず、第2項について考える。

$$\text{第2項} = \sum_s \left\{ \sum_i^N \frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right\}$$

まず前提として、正解仮説において $X_i \sim N(\mu_i, \sigma_i^2)$ とする
確率変数 $Y_i = \frac{(X_i - \mu_i)^2}{2\sigma_i^2}$ の密度関数は

$$f(y_i) = \frac{e^{-y_i}}{\sqrt{\pi y_i}} \quad (y_i > 0)$$

となりモーメントから

$$\text{平均} : E[Y_i] = 0.5$$

$$\text{分散} : E[Y_i^2] - (E[Y_i])^2 = 0.5$$

が得られる。

ここで、 Y_i の分布について再生性を仮定すると

$$\sum_i^N Y_i (N = 34) \text{ において} \quad \begin{aligned} \text{平均} &: 0.5 \times 34 = 17 \\ \text{分散} &: 0.5 \times 34 = 17 \end{aligned}$$

これが、正規分布に近似可能と考えると

$$\sum_i^N Y_i \sim N(17, 17)$$

したがって、

$$\begin{aligned} \text{第2項} &= \sum_s \left\{ \sum_i^N \frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right\} \\ &\sim \sum_s N(17, 17) \end{aligned}$$

となる。

第2項 $\sim \sum_s N(17, 17)$ を利用すると score は以下のように近似できる。

$$\begin{aligned} \text{score} &\sim \sum_s \left\{ \sum_i^N \log \frac{1}{\sqrt{2\pi\sigma_i}} \right\} - \sum_s N(17, 17) \\ &\sim N\left(\sum_s \left\{ \sum_i^N \log \frac{1}{\sqrt{2\pi\sigma_i}} - 17 \right\}, 17\right) \end{aligned}$$

また、湧き出し誤りとして付加した keyword の score は

$$\text{score} \sim N\left(\sum_s \left\{ \sum_i^N \log \frac{1}{\sqrt{2\pi\sigma_i}} - 17 - k \right\}, 17\right)$$

k : 与えられた score の補正值

として評価する。

3 擬似スポッティング結果発生プログラム

第2章のエラーモデルを用いて音素ラベルファイルからエラーを含んだ擬似スポッティング結果を発生させるプログラムについて考える。

音素ラベルファイルに対して音素ラベル列照合した結果を元に、与えられた脱落誤り率、湧き出し誤り率に従って誤りを付加する。したがって、このプログラムの入力は以下のようになる。

実行ファイル

キーワード辞書 (salient_word.JT_tr)	} score 計算用
Path Length (read_dist.exe の出力)	
音響モデル (HMnet_filled.1.400)	
キーワードリスト (salient_word.list)	
脱落誤り率 湧き出し誤り率	
湧き出しに対するスコアの補正	
入力ファイル (Exe.get_word_label の出力+各発話の終了時間)	

ここで、入力ファイルの種類によって次の3種類に分類して考えていく。
まず、準備としてどの場合にも標準出力される flag の値についてまとめておく。

表 1: 各入力に対する flag の意味

case	flag	
1	0	正しいスポッティング結果
	2	エラーモデルを元に付加した湧き出し誤り
2	0	正しいスポッティング結果
	2	エラーモデルを元に付加した湧き出し誤り
3	0	音素ラベルファイルに対して音素ラベル列照合をしたスポッティング結果
	2	エラーモデルを元に付加した湧き出し誤り

3.1 case 1

入力ファイル： 形態素解析結果に対して音素ラベル列照合をした結果

標準出力 :

1. ファイル名
2. 開始時間, keyword, 終了時間, score, flag

標準エラー出力 :

1. 対話総時間
2. 脱落誤り率 (与えられた誤り率)
3. 各 keyword の湧き出し数, 湧き出し率 (/hr)
4. 湧き出し率 (与えられた湧き出し率)

3.2 case 2

入力ファイル : 音素ラベルファイルに対して音素ラベル列照合をした結果と形態素解析結果に対して音素ラベル列照合をした結果

標準出力 :

1. ファイル名
2. 開始時間, keyword, 終了時間, score, flag

標準エラー出力 :

1. 対話総時間
2. 同一音素系列による湧き出し数
3. 脱落誤り率 (与えられた誤り率)
4. 各 keyword の湧き出し数, 湧き出し率 (/hr)
5. 湧き出し率 (与えられた湧き出し率)

3.3 case 3

入力ファイル : 音素ラベルファイルに対して音素ラベル列照合をした結果

標準出力 :

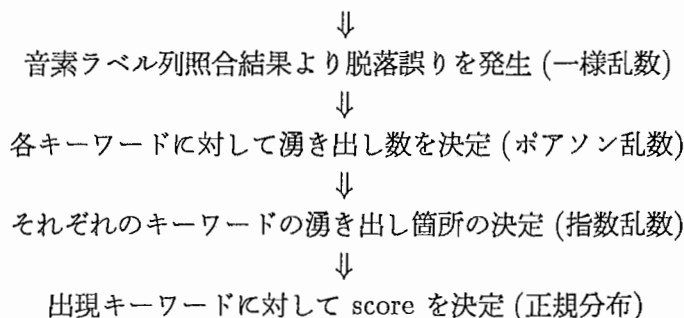
1. ファイル名
2. 開始時間, keyword, 終了時間, score, flag

標準エラー出力 :

1. 対話総時間
2. 脱落誤り率 (与えられた誤り率)
3. 各 keyword の湧き出し数, 湧き出し率 (/hr)
4. 湧き出し率 (与えられた湧き出し率)

3.4 擬似スポッティング結果発生アルゴリズム

正しいスポッティング結果と同一音素系列による湧き出し誤りとの区別 (case 2 のみ)



4 実行例

第 3.1 節, 第 3.2 節, 第 3.3 節のそれぞれの場合の擬似スポッティング結果例を挙げる.

使用コーパスはホテル予約などをタスクとする音声対話の書き起こしコーパスで, TAC70015 という対話のうちの 3 発話目の擬似スポッティング結果のみを以下の例に挙げる.

発話: はい, ありがとうございます. [えー] 最初にお客様のお名前を頂けますでしょうか.

4.1 case 1

形態素解析結果に対する音素ラベル列照合結果

...
TAC70015.0030.B.WLB
1560.00 e,e 1905.00
1905.00 s,a,i,sh,j,o 2190.00
2320.00 k,j,a,k,u 2500.00
2885.00 n,a,m,a,e 3180.00
3845.00 sh,j,o,o 4030.00
#4255.0
...

実行結果

...
TAC70015.0030.B
6 zh,j,u,u,sh,j,o 506 3730.052246 37.300522 2
747 o,n,a,k,a 1147 2769.187012 34.614838 2
843 p,u,r,a,s,u 1293 3098.384277 34.426491 2
1045 j,u,u,g,a,t,a 1595 4141.731934 37.652107 2
1270 s,u,t,a,ng,d,a,a,d,o 2020 5408.276367 36.055176 2
1433 sh,i,n,a,k,a,z,u 2033 4351.379395 36.261494 2
1560 e,e 1905 3271.973389 47.419903 0
1905 s,a,i,sh,j,o 2190 2706.567871 47.483646 0
2320 k,j,a,k,u 2500 1734.699097 48.186085 0

2390 sh,j,o,o,m,e,ng 2890 3878.174805 38.781750 2
2622 n,o,r,i,k,a,e 3172 4019.473877 36.540672 2
2885 n,a,m,a,e 3180 2758.178467 46.748787 0
3288 b,e,ts,u 3588 2192.409424 36.540157 2
3845 sh,j,o,o 4030 1765.133667 47.706314 0
4055 sh,j,u,r,u,i 4505 3418.904541 37.987827 2
4097 j,a,ts,u 4397 2269.538574 37.825642 2
#

...

ここで、脱落誤り率 0.3, 湧き出し誤り率 20 を与え、実際の脱落誤り率は 0.33, 湧き出し誤り率は 25.8 となった。score の補正は 10 とした。

4.2 case 2

形態素解析結果に対する音素ラベル列照合結果

...

TAC70015.0030.B.WLB
1560.00 e,e 1905.00
1905.00 s,a,i,sh,j,o 2190.00
2320.00 k,j,a,k,u 2500.00
2885.00 n,a,m,a,e 3180.00
3845.00 sh,j,o,o 4030.00
#4255.0

...

音素ラベルファイルに対する音素ラベル列照合結果

...

TAC70015.0030.B.LBL
490.00 a,t,o 630.00
650.00 g,o 695.00
825.00 i,m,a 945.00
910.00 a,s,u 1165.00
1560.00 e,e 1905.00
1905.00 s,a,i,sh,j,o 2190.00
1905.00 s,a,i 2090.00
2320.00 k,j,a,k,u 2500.00
2885.00 n,a,m,a,e 3180.00
3010.00 m,a,e 3180.00
3655.00 a,s,u 3770.00
3845.00 sh,j,o,o 4030.00
3905.00 j,o,o 4030.00
#4255.0

...

実行結果

...

TAC70015.0030.B

229 r,a,n,g,d,o,r,i,i,s,a,a,b,i,s,u 1379 8381.829102 2
303 h,i,d,a,r,i,t,e 903 4285.456543 2
484 zh,i,t,e,n,g 834 2625.676270 2
490 a,t,o 630 1350.200317 1
549 e,n,g,ch,j,o,o 999 3434.789795 2
650 g,o 695 435.759552 1
825 i,m,a 945 1166.050781 1
910 a,s,u 1165 2469.123047 1
1356 r,o,s,a,n,g,z,e,r,u,s,u 2156 5799.988770 2
1439 a,sh,i,t,a 1839 2916.060547 2
1560 e,e 1905 3339.825684 0
1625 g,e,n,g,k,i,n,g 2025 3064.651123 2
1702 i,q,k,a,t,s,u 2152 3190.608154 2
1838 b,u,n,g,j,a 2188 2436.407471 2
1905 s,a,i 2090 1767.628540 1
1905 s,a,i,sh,j,o 2190 2738.515137 0
2177 t,a,i,sh,j,o 2627 3217.743652 2
2320 k,j,a,k,u 2500 1703.537476 0
2778 zh,i,n,g,z,a,i 3228 3376.760498 2
2885 n,a,m,a,e 3180 2838.690186 0
2904 k,a,i,g,a,i 3404 3821.120117 2
3010 m,a,e 3180 1631.119507 1
3120 k,o,n,g,z,a,t,s,u 3620 3664.009521 2
3212 k,a,s,a,n,g 3562 2606.791260 2
3309 k,j,o,h,i 3659 2375.424072 2
3655 a,s,u 3770 1094.822754 1
3704 s,a,i,sh,j,u,u,t,e,k,i 4554 6532.818359 2
3845 sh,j,o,o 4030 1787.289917 0
3855 h,o,t,e,r,u,m,e,e 4555 5281.756348 2
3905 j,o,o 4030 1142.909302 1

#

...

ここで、脱落誤り率 0.3, 湧き出し誤り率 20 を与え、実際の脱落誤り率は 0.25, 湧き出し誤り率は 25.8 となった。score の補正は 10 とした。

4.3 case 3

音素ラベルファイルに対する音素ラベル列照合結果

...

TAC70015.0030.B.LBL

490.00 a,t,o 630.00
650.00 g,o 695.00
825.00 i,m,a 945.00
910.00 a,s,u 1165.00
1560.00 e,e 1905.00
1905.00 s,a,i,sh,j,o 2190.00
1905.00 s,a,i 2090.00
2320.00 k,j,a,k,u 2500.00
2885.00 n,a,m,a,e 3180.00
3010.00 m,a,e 3180.00
3655.00 a,s,u 3770.00
3845.00 sh,j,o,o 4030.00
3905.00 j,o,o 4030.00
#4255.0

...

実行結果

...

TAC70015.0030.B
490 a,t,o 630 1396.812256 0
793 g,a,q,k,o,o 1243 3415.612793 2
817 a,r,e 1067 1725.348511 2
825 i,m,a 945 1161.416382 0
1082 t,o,k,i 1382 2065.744629 2
1187 j,o,o,sh,j,o,k,u 1787 4278.214355 2
1415 d,a,i,t,a,i 1915 3756.306885 2
1560 e,e 1905 3311.532227 0
1569 n,i,q,t,e,e 2019 3410.342529 2
1679 n,a,k,a 1979 2088.667969 2
1693 t,e,m,a,e 2093 3024.718506 2
1902 t,o,m,a,t,o,zh,j,u,u,s,u 2802 6671.986816 2
1905 s,a,i,sh,j,o 2190 2825.381104 0
1919 i,ch,i,r,i,ts,u 2469 4064.557129 2
2089 i,r,o,i,r,o 2589 3455.927002 2
2119 m,u,r,i 2419 2099.494873 2
2153 sh,j,u,r,u,i 2603 3409.116699 2
2219 b,o,t,a,ng 2569 2586.265625 2
2320 k,j,a,k,u 2500 1729.160156 0
2530 w,a,r,i,m,a,sh,i 3130 4395.763672 2
2729 i,i 2929 1573.064331 2
2885 n,a,m,a,e 3180 2872.242432 0
3010 m,a,e 3180 1628.762939 0
3026 sh,i,ng,s,a,i,b,a,sh,i 3776 5636.924805 2
3086 k,a,e,r,i 3486 3154.629150 2

3551 i,ng,g,u,r,i,q,sh,j,u,s,u,t,a,i,r,u 4801 9214.046875 2
 3806 t,a,ng,i 4106 2243.042480 2
 3845 sh,j,o,o 4030 1764.004028 0
 3905 j,o,o 4030 1247.100464 0
 4033 i,sh,j,a 4333 2184.770996 2
 4247 r,i,j,o,o 4647 3046.866211 2
 #

...

ここで、脱落誤り率 0.3, 湧き出し誤り率 20 を与え、実際の脱落誤り率は 0.28, 湧き出し誤り率は 27.4 となった。score の補正は 10 とした。

4.4 考察

第 4.1 節, 第 4.2 節, 第 4.3 節の例において実行結果として得られる湧き出し誤り率が与えられたものよりもかなり大きくなっている。これは、この例で用いている対話の対話時間が 107215 [msec] と短いため各キーワードに対し湧き出し誤りはあるかないかとなってしまっているためであると考えられる。従って、総対話時間を長くするか与える湧き出し率を大きくすれば与えられた湧き出し誤り率に近い結果が得られる。以下の表に対話時間を変化させた場合と与える湧き出し誤り率を変化させた場合の実際に得られた脱落誤り率と湧き出し誤り率を示す。

表 2: 対話時間と湧き出し誤り率

対話時間 [msec]	与えられた 湧き出し誤り率	実際の 湧き出し誤り率
107215 (1 対話)	20	25.78
	20	27.05
	50	49.17
	50	48.54
	100	101.10
	100	100.31
451105 (5 対話)	20	20.47
	20	20.29
	50	49.51
	50	50.08
	100	100.32
	100	100.15

5 おわりに

以上のように、キーワードスポッティングにおけるエラー発生過程のモデル化を行ない、上記モデルを用いて、音素列に対しスポッティング結果のシミュレーションを行なうプログラムを作

成した。

ただし、上記のプログラムにおいて

- 脱落させた正しいキーワードが湧き出し誤りとして認識される可能性
- 湧き出し誤り発生の際の音響的特徴の類似性(ラベルデータに加えて音素タイプライタの出力結果などを用いることにより考慮可能であると考えられる)

については考慮していない。

今後、音響的特徴の類似性を考慮して湧き出し誤りを発生させることにより、実際の音声認識結果により近いエラーを付加することができると考えられる。

6 謝辞

快適な実習環境を提供して頂いた ATR 音声翻訳通信研究所第一研究室の연구원諸氏に感謝の意を表します。また、本実習の機会を与えて下さった ATR 音声翻訳通信研究所の山崎泰弘社長、並びに ATR 音声翻訳通信研究所第一研究室の匂坂芳典室長に深く感謝致します。

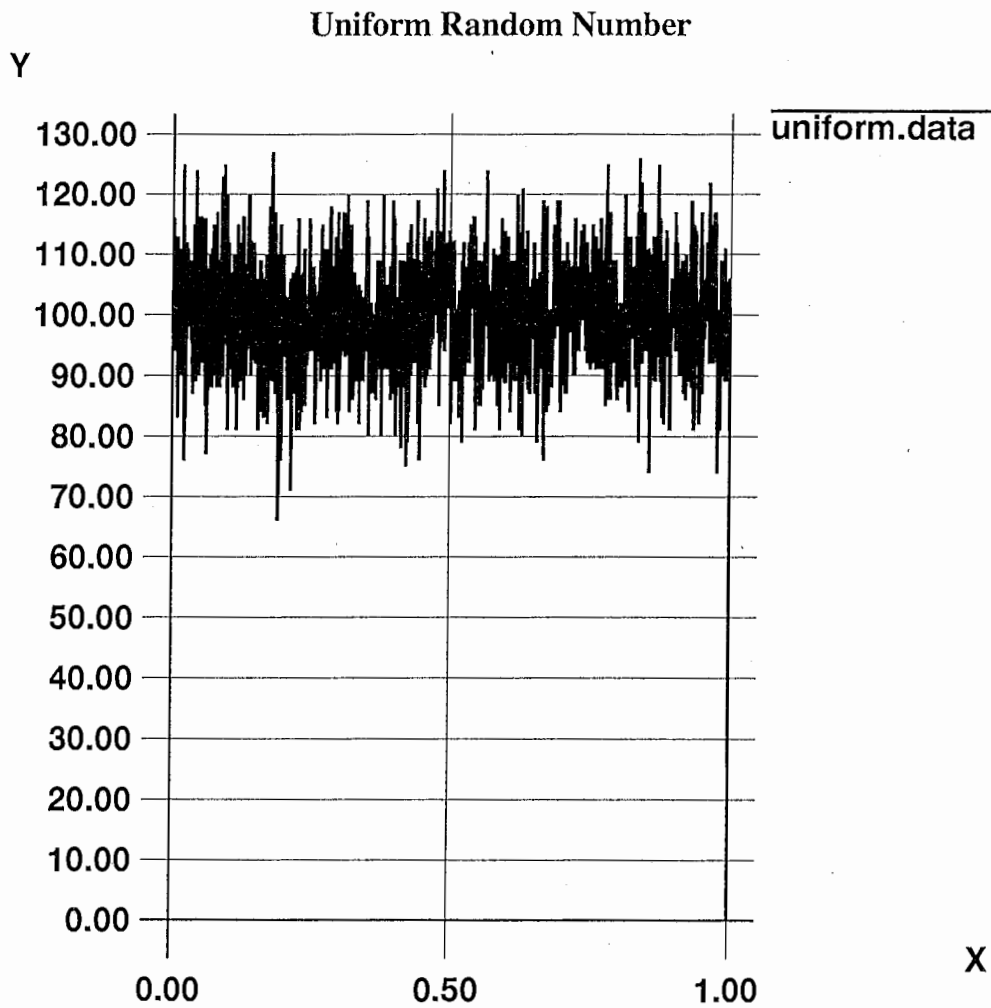
参考文献

- [1] E.Kreyszig 著 (田栗正章 訳): “Advanced Engineering Mathematics 6”, (邦題 技術者のための高等数学 6 「確率と統計」), 培風館, 1988.
- [2] 中川 聖一: “確率モデルによる音声認識”, コロナ社, 1988.
- [3] 中川 聖一, 大黒 慶久, 村瀬 功: “連続音声認識システムの評価法 — タスクの複雑性と文認識率との関係 —”, 電子情報通信学会論文誌 D-II Vol. J73-D-II No.5 pp.683-693, 1990.
- [4] 東京大学教養学部統計学教室編: “統計学入門”, 東京大学出版, 1994.

付録

A 使用擬似乱数の検定

A.1 一様乱数



一様乱数の分布 (100,000 回発生)

この擬似乱数が一様分布に従うことは χ^2 検定

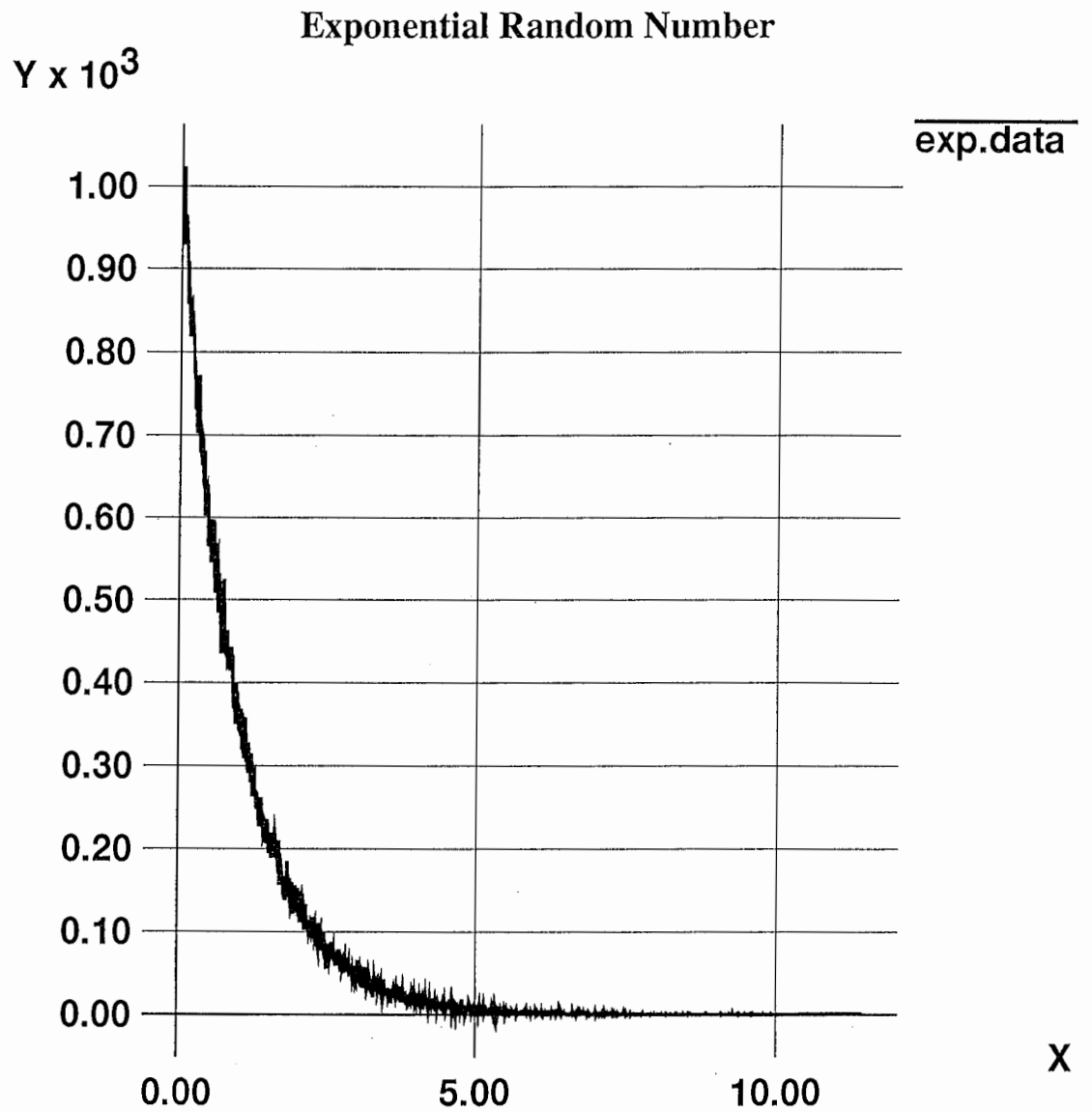
自由度 : $k - 1 = 101 - 1 = 100$

標本数 : $N = 1000$

有意水準 : $\alpha = 0.05$

において採択されている。

A.2 指数乱数



指数乱数の分布 (100,000 回発生)

この擬似乱数が指数分布に従うことは χ^2 検定

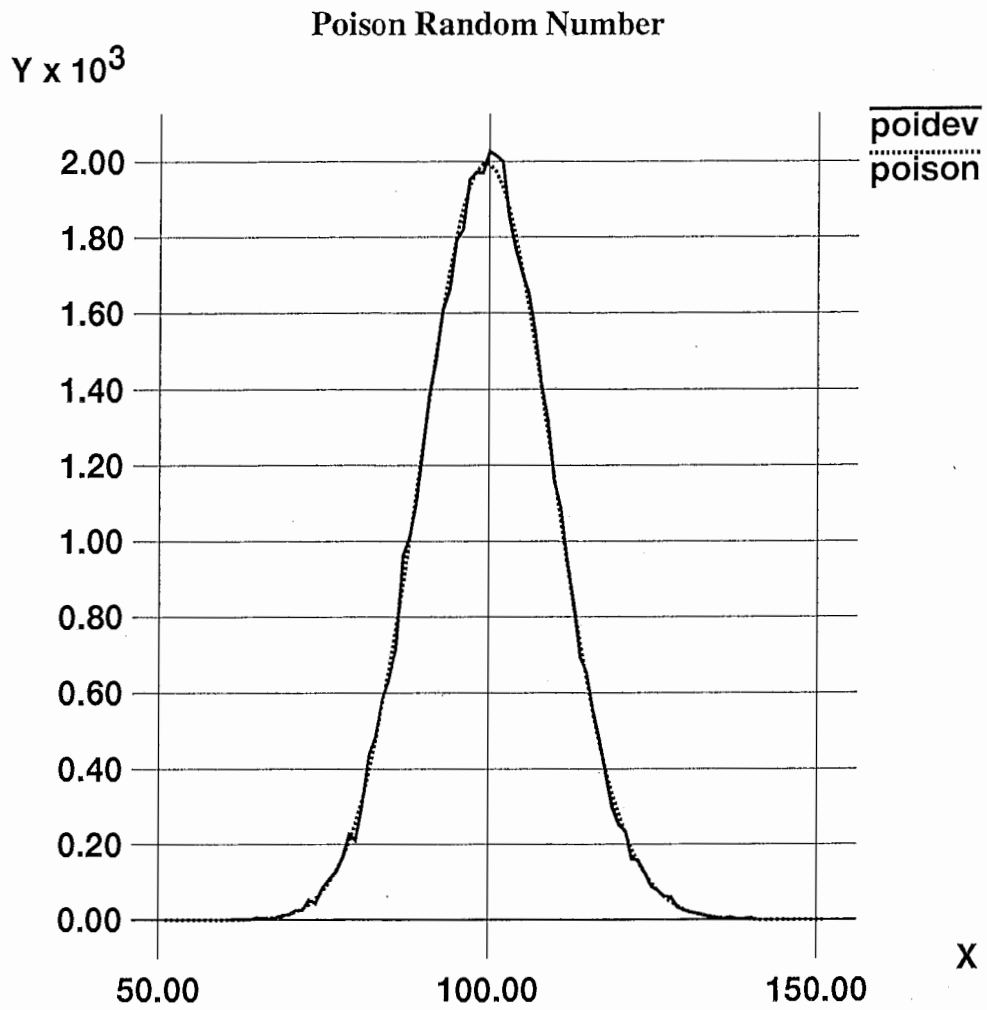
自由度 : $k - 1 = 101 - 1 = 100$

標本数 : $N = 1000$

有意水準 : $\alpha = 0.05$

において採択されている。

A.3 ポアソン乱数



この擬似乱数がポアソン分布にしたがうことは χ^2 検定

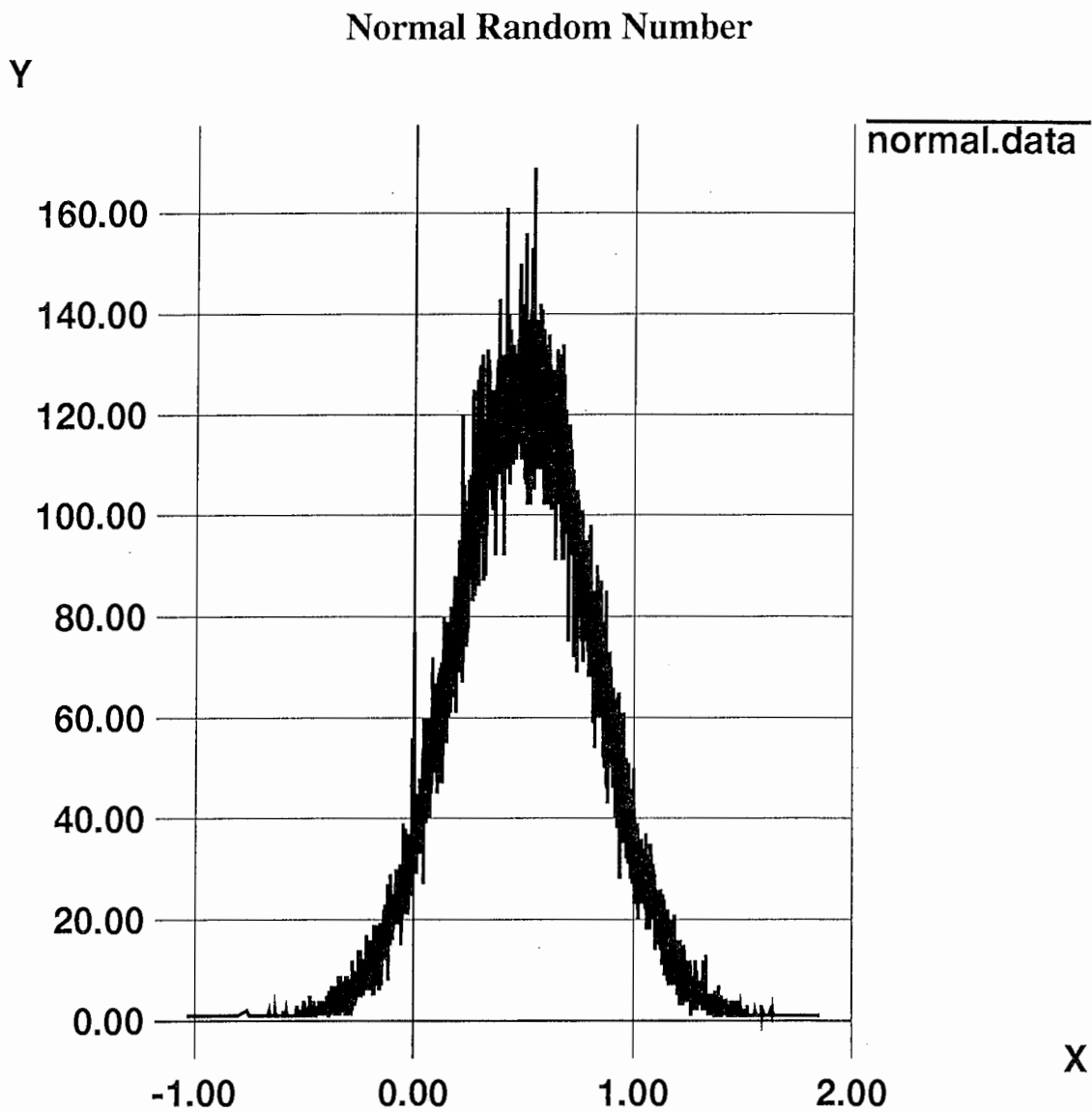
自由度 : $k - 1 = 51 - 1 = 50$

標本数 : $N = 10000$

有意水準 : $\alpha = 0.05$

において採択されている。

A.4 正規乱数 $N(0.5, 0.1)$



正規乱数の分布 (100,000 回発生)

この擬似乱数が指数分布に従うことは χ^2 検定

自由度 : $k - 1 = 9 - 1 = 8$

標本数 : $N = 10000$

有意水準 : $\alpha = 0.05$

において採択されている。

付録

B プログラム解説

spot_w_error

使用法: spot_w_error キーワード辞書ファイル path_length を求めたファイル
音響モデルファイル キーワードリストファイル
脱落誤り率 湧き出し誤り率 湧き出し誤りのスコアの補正 (/frame)
入力ファイル名

case1 および case3 のシェルスクリプト例

```
#!/bin/csh -f

set ROOTDIR = '~xmorikaw/jisshu/'
set TMP_WORK_FILE = '~/tmp_work_file'
set TMP_DIC_FILE = '~/tmp_dic_file'

pushd . > null

cd $ROOTDIR/read_dist/

read_dist.exe >& $TMP_DIC_FILE

popd >null

echo "" >> $TMP_WORK_FILE
foreach LBLFILE ( `ls -m $ROOTDIR/error_model/TAC70015_WLB/TAC7*.WLB | sed 's/,//g'` )
#echo "" >> $TMP_WORK_FILE
echo $LBLFILE:t >> $TMP_WORK_FILE
    $ROOTDIR/word_label/Exe.get_word_label \
    -ky $ROOTDIR/keyword/salient_word.list -L $LBLFILE -lm 1\
    >> $TMP_WORK_FILE

    awk '$3!=""{s=$3}END{print s}' $LBLFILE >> $TMP_WORK_FILE

end

# spot_w_error キーワード辞書 path_length を求めたファイル
#             音響モデル キーワードリスト
```

```
#          脱落誤り率 湧きだし誤り率 湧きだし誤りのスコアの補正 (/frame)
#          入力ファイル名
```

```
spot_w_error \  
salient_word.JT_tr $TMP_DIC_FILE \  
HMnet_filled.1.400 salient_word.list \  
0.3 20 10 $TMP_WORK_FILE
```

```
rm $TMP_WORK_FILE  
rm $TMP_DIC_FILE  
rm null
```

```
exit 0
```

case2 のシェルスクリプト例

```
#!/bin/csh -f
```

```
set ROOTDIR = '~xmorikaw/jisshu/'  
set TMP_WORK_FILE1 = '~/tmp_work_file1'  
set TMP_WORK_FILE2 = '~/tmp_work_file2'  
set TMP_DIC_FILE = '~/tmp_dic_file'
```

```
pushd . > null  
cd $ROOTDIR/read_dist/
```

```
read_dist.exe >& $TMP_DIC_FILE
```

```
popd >null
```

```
echo "" >> $TMP_WORK_FILE1  
foreach LBLFILE ( `ls -m $ROOTDIR/error_model/TAC70015_WLB/TAC7*.WLB | sed 's/,//g'` )  
#echo "" >> $TMP_WORK_FILE1  
echo $LBLFILE:t >> $TMP_WORK_FILE1  
    $ROOTDIR/word_label/Exe.get_word_label \  
    -ky $ROOTDIR/keyword/salient_word.list -L $LBLFILE -lm 1\  
>> $TMP_WORK_FILE1
```

```
awk '$3!=""{s=$3}END{print s}' $LBLFILE >> $TMP_WORK_FILE1
```

```
end
```

```

foreach LBLFILE ( `ls -m $ROOTDIR/error_model/TAC70015_LBL/TAC7*.LBL | sed 's/,//g'` )
echo "" >> $TMP_WORK_FILE2
echo $LBLFILE:t >> $TMP_WORK_FILE2
    $ROOTDIR/word_label/Exe.get_word_label \
        -ky $ROOTDIR/keyword/salient_word.list -L $LBLFILE -lm 1\
    >> $TMP_WORK_FILE2
end

# spot_w_error キーワード辞書 path_length を求めたファイル
#             音響モデル キーワードリスト
#             脱落誤り率 湧きだし誤り率 湧きだし誤りのスコアの補正 (/frame)
#             入力ファイル名1 入力ファイル名2

    spot_w_error \
    salient_word.JT_tr $TMP_DIC_FILE \
    HMnet_filled.1.400 salient_word.list \
    0.3 20 10 $TMP_WORK_FILE1 $TMP_WORK_FILE2

rm $TMP_WORK_FILE1
rm $TMP_WORK_FILE2
rm $TMP_DIC_FILE
rm null

exit 0

```

上記シェルスクリプトにおいて \$TMP_WORK_FILE* および \$TMP_DIC_FILE をあらかじめ用意しておけば実行時間は大幅に削減される。

付録

C ディレクトリ構造とファイルの解説

~/JMORDIR/ 形態素解析結果ファイルを格納

~/JTEXTDIR/ 日本語書き起こしファイルを格納

~/LBLDIR/ 音素ラベルファイルを格納

~/TeX/ 実習報告書, OHPのLaTeXソースファイルおよび図

~/WLBDIR/ 単語境界を与えられた音素ラベルファイルを格納

~/jisshu/error_model/ 実習で作成したプログラム関係のファイルを格納

- ./SYSTEM/ : 擬似スポッティング結果発生プログラムの実行ファイル, シェルスクリプトおよび入力ファイル例を格納
 - ./SYSTEM/SPOT.CSH : 第3章の case 1 および case 3 の場合のシェルスクリプト例
 - ./SYSTEM/SPOT_w_overlap.CSH : 第3章の case 2 の場合のシェルスクリプト例
 - ./SYSTEM/spot_w_error : 擬似スポッティング結果発生プログラムの実行ファイル
 - ./SYSTEM/HMnet_filled.1.400 : 音響モデルファイル
 - ./SYSTEM/salient_word.JT_tr : キーワード辞書ファイル
 - ./SYSTEM/salient_word.list : キーワードリストファイル
 - ./SYSTEM/keywd.path.length : path length ファイル
(xmorikaw/jisshu/read.dist の出力)
 - ./SYSTEM/test.onso : 形態素解析結果に対する音素ラベル列照合結果ファイル
 - ./SYSTEM/test.right : 音素ラベルファイルに対する音素ラベル列照合結果ファイル
 - ./SYSTEM/OUT/ : 実行結果例を格納
 - ./SYSTEM/C/ : プログラムのソースファイルを格納
- ./random/ : 使用乱数の検定プログラムを格納