卒業論文

ＳＥＩＵＮを用いた文字認識に関する基礎的研究

東北大学工学部情報工学科
丸岡(東)研究室４年

杉本秀昭

目 次

1 序論 3
1.1 本研究の背景 3
1.2 文字認識の手法 3
1.3 本研究の目的 4
1.4 本論文の構成 4

2 文字認識アルゴリズム 5
2.1 文字入力 5
2.2 前処理 5
2.2.1 ノイズ除去、スムージング 5
2.2.2 正規化 5
2.2.3 細線化 5
2.2.4 線素化 5
2.3 特徴量抽出 5
2.4 認識 6
2.4.1 辞書作成 6
2.4.2 評価値計算 6
2.5 候補出力 6

3 連想辞書の作成 10
3.1 連想辞書の構成 10
3.2 連想辞書の作成アルゴリズム 10
3.3 クラスタリング法 10
3.3.1 Ｌｉｎｄｅ法 10
3.3.2 階層的クラスタリング法 11
3.3.3 最大距離法 12
3.4 ＳＥＩＵＮを用いたクラスタリング 12
3.4.1 距離計算部”Ｄ” 12
3.4.2 上位選出&得点割り当て部”Ｐ” 12
3.4.3 クラスタリングプログラムのＳＥＩＵＮでの機能代行 . 13

4 実験 22
4.1 実験準備 22
4.2 クローズ実験 22
4.3 オープン実験 22
4.4 総合結果 22
4.5 実験結果に関する考察 22
4.5.1 認識率 22

4.5.2 速度比較 23

5 結論 43

1　序論

1.1　本研究の背景

近年、計算機の発達とそれによる情報処理が急速に広まってきている。また、大容量の補助記憶装置などの周辺機器の開発により計算機の扱う情報の量はますます増加している。しかしこの場合、現在でも計算機に対するデータの入力が主としてキーボードなどを介して人間の手によって行なわれていることを考えると、将来さらに倍増するであろうと予想されるデータの入力に対処できなくなることが予想される。そこで、人間にかわってデータを入力する方法として計算機による文字の自動認識に関する研究が３０年来行なわれてきた。

そのような情況の中、当研究室では、人間の右脳、左脳の機能分担をモデルにした高速高精度知的認識システム「ＳＥＩＵＮ」を富士通と協同開発した。このシステムでは印刷された活字文字の場合、読み取り速度は毎秒約２００文字、Ａ４版大の文書１枚を約１０秒で読み取る。またその認識率は９９．９パーセントを越える。

1.2　文字認識の手法

文字認識のアルゴリズムには大きく分けて二つの方法がある。一つは構造解析法であり、もう一つはパターン整合法である。前者の構造解析法は文字の線分関係や位置関係に着目し、その構造を特徴量とし、主成規則や表を用いて認識を行なう方法である。後者のパターン整合法は、パターン同士の重なり具合から類似度を基に認識を行なう。構造解析法は文字の変動に強く、認識の高精度化が期待できるが、処理量が膨大で線分の正確な対応づけなどの問題もあり、まだ実用化は困難である。パターン整合法はアルゴリズムが簡単で高速に処理を行なうことができるが、文字の構造情報を陽に評価していないため文字の変動等に弱い。

本研究ではパターン整合法を用いて文字認識を行なう場合を考える。

1.3 本研究の目的

当研究室で開発したＳＥＩＵＮはその認識アルゴリズムとしてパターン整合法の一つである連想整合法を採用している。連想整合法は認識を行なうためにクラスタリングによって作られた連想辞書という特別な辞書を必要とする。本研究ではＳＥＩＵＮを用いた文字認識システム開発の一環として、ＳＥＩＵＮの機能を活用した高速な連想辞書の作成のためのアルゴリズムについて考える。また連想整合法に最も適した連想辞書のクラスタリング方法についても検討する。

1.4 本論文の構成

第１章は序論である。第２章では当研究室で考案された連想整合法のアルゴリズムについて述べる。第３章では連想辞書作成のためのクラスタリング手法について述べる。第４章では第３章で挙げられたそれぞれのクラスタリング方法によって作成された連想辞書を用いて認識実験を行ない、その比較検討をする。第５章では結論を述べる。

2　文字認識アルゴリズム

2.1　文字入力

認識される文字データは、イメージスキャナから取り込まれる。読み込んだデータは白黒２値のデータとして出力され、各文字毎に切り出される（図２．１（ａ））。

2.2　前処理

切り出された文字データは、特徴量を求めるために次のような４段階の前処理が行なわれる。

2.2.1　ノイズ除去、スムージング

入力される原稿やイメージスキャナの性能等によって出力されたデータにはノイズや線分上の凸凹が発生している場合が多い。このため、まずノイズ除去を行ない次にスムージングを行なう（図２．１（ｂ））。

2.2.2　正規化

取り込んだ活字文字の大きさは全角や半角などさまざまなものがある。そこで文字の大きさを入力されたデータに対し一定にするために拡大、縮小を行なう。ここでは統一されたデータの大きさは６４×６４ドットとする（図２．１（ｃ））。

2.2.3　細線化

ここでは文字の線分の幅による違いを取り除くために、数ドットの幅をもつ文字の線分を１ドットの幅に縮める（図２．１（ｄ））。そのためのアルゴリズムとしてＨｉｌｄｉｔｃｈの細線化アルゴリズムを用いる。

2.2.4　線素化

２．２．３節で細線化された各ドットにその点での線分の方向を対応させる。線分の方向は"｜"、"ー"、"＼"、"／"の４つに分類される（図２．１（ｅ））。

2.3　特徴量抽出

当研究室では、これまで手書き文字の特徴量として方向密度ベクトルを用いてきた。その後それを参考にして活字特徴量として方向線素特徴量が考案された。方向線素特徴量は図２．２のように、まず６４×６４ドットの線素

化文字を8ドット間隔に縦横を分割する。次に左上から１６×１６ドットを半分づつ重複させて、４９（７×７）個のマスを作る。４９個のマスの並びは左上のマスを１番とし、左から右、上から下へと順番に番号を付けたものとする。1マスのベクトルは、"|"、"ー"、"／"、"＼"の４つのカウンタからなる４次元ベクトルでできており、マス内の重みは図２．２のようになっている。したがって１文字当りの次元数は１９６（４９×７）次元となる。図２．３に"千"をサンプルにした４番目のマスの抽出例を示してある。

2.4 認識

ここでは、認識手法として最も一般的な全数整合法で説明する。この手法は辞書ベクトルと各文字の特徴抽出でできた特徴ベクトルで距離を求め、近いものから順に識別する方法である。

2.4.1 辞書作成

各文字に対して１つの辞書ベクトルを作る。辞書作成の具体的方法については第３章で述べる。

2.4.2 評価値計算

評価値計算は辞書ベクトルと未知入力特徴ベクトルとから評価値を求める。一般には未知入力パターンを p，未知入力ベクトルを u とすると、

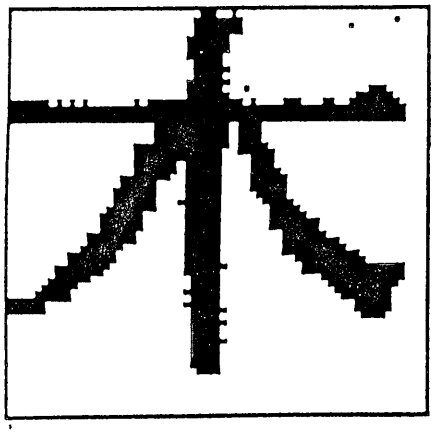
$$u = F(p) \in Z^N$$

となる。このとき評価値を $E(v^k, u) : Z^N \times Z^N \rightarrow Z$ と表わす。

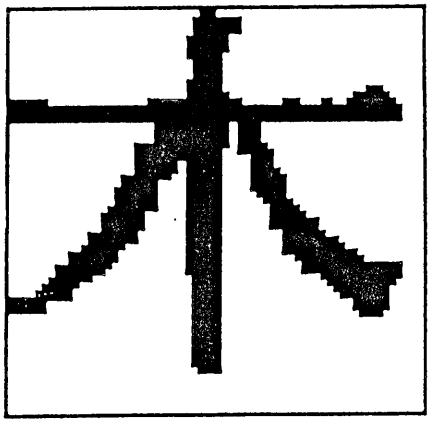
評価値計算によく使われるのは距離の公理を満たす距離値である。ここでは E としてユークリッド距離を用いる。

2.5 候補出力

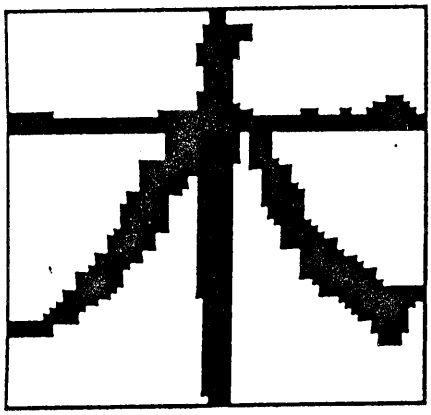
評価値計算をした結果に沿って１位候補、2位候補、…、と出力していく。



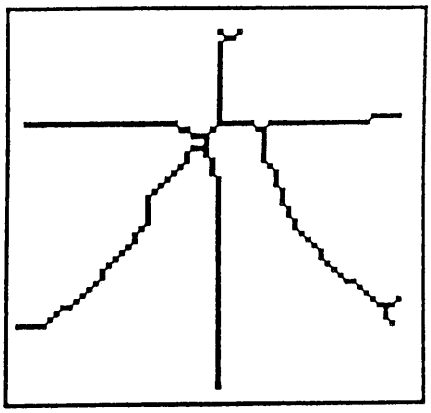
（a）　イメージデータ



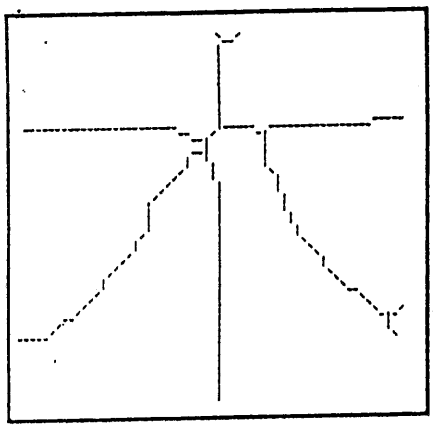
（b）　ノイズ除去・スムージング後



（c）　正規化後



（d）　細線化後

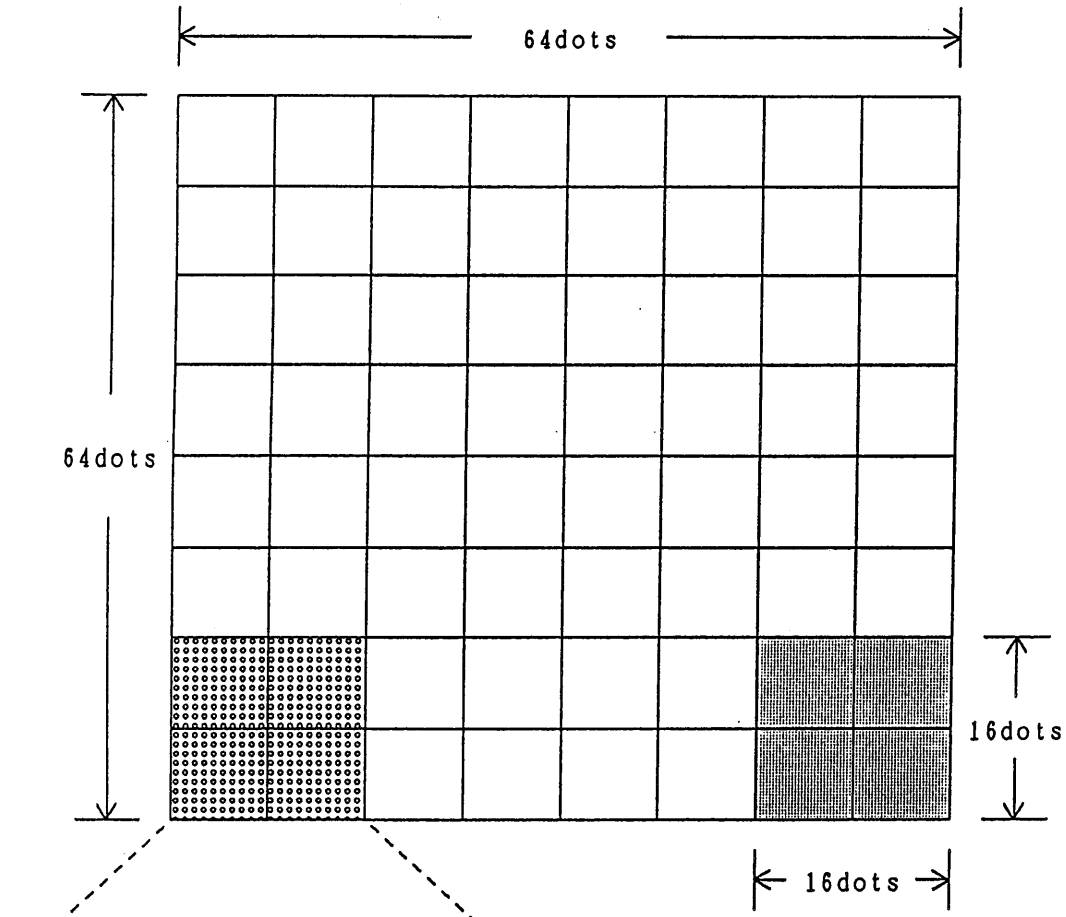


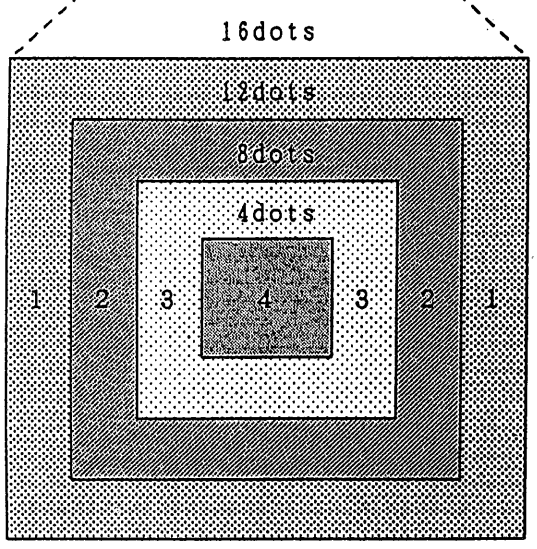
（e）　線素化後

図２．１：各処理後のイメージ

各マス中の方向ベクトル

“v_1” “v_2” “v_3” “v_4”





方向線素特徴量

１マス：　１６ドット（横）×１６ドット（縦）

マス数：　７（横）×７（縦）

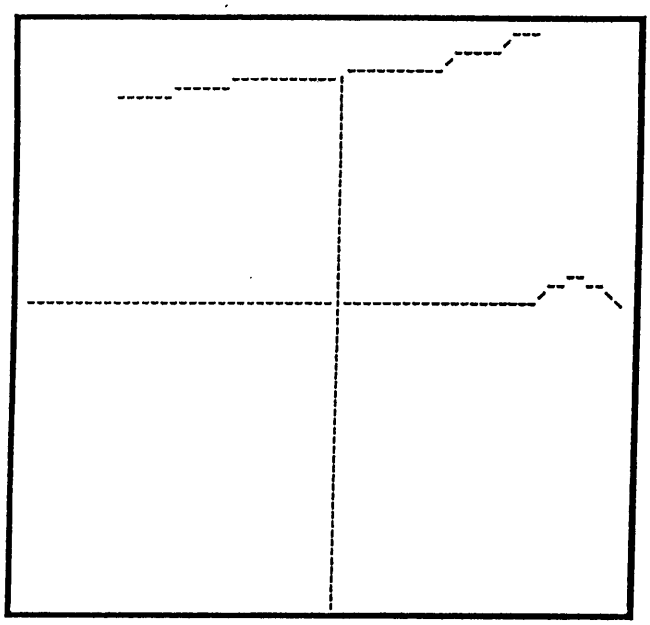
＝４９個

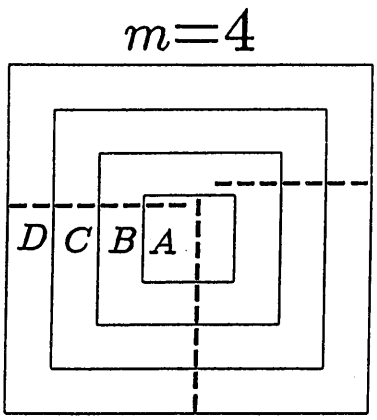
次元数：　４９マス×４次元
＝１９６次元

重みのかけ方：

右の枠内の数字は枠内線素に対する重み。

図２．２：方向線素特徴量





千

$$V_{41} = \underbrace{\mathbf{4} * 4}_{A} + \underbrace{\mathbf{3} * 2}_{B} + \underbrace{\mathbf{2} * 2}_{C} + \underbrace{\mathbf{1} * 2}_{D} = 28$$

$$V_{42} = 4 * 2 + \mathbf{3} * 5 + \mathbf{2} * 4 + \mathbf{1} * 4 = 35$$

$$V_{43} = 0$$

$$V_{44} = 0$$

$$\underline{\mathbf{V}_4 = (28, 35, 0, 0)}$$

図２．３：方向線素特徴量の抽出例

3　連想辞書の作成

3.1　連想辞書の構成

連想整合法では総当たり法と違い、全ての文字と照合するのではなくて、４９（７×７）個それぞれのマス毎に、マスの代表ベクトルのみと照合すればよい。このそれぞれのマス毎の代表ベクトルを求め、またその代表ベクトルと似たベクトルの組み合せを関連づけた連想整合法の標準パターンのことを連想辞書という。図３．１にその例を示す。

3.2　連想辞書の作成アルゴリズム

連想辞書は、次のように構成される。

全字種$y_1 \sim y_N$（Nは字種数）を各マス毎にクラスタリングし、L個のクラス

$$C_1^{(m)}, C_2^{(m)}, \ldots, C_L^{(m)}, m = 1 \cdots 49$$

を得る。各クラスの代表ベクトル$\overline{x}_i^{(m)}$を、

$$\overline{x}_i^{(m)} = E(x), x \in C_i^{(m)}, i = 1 \cdots L, m = 1 \cdots M$$

のように定める。

本研究ではクラスタリングの方法として今まで使用してきたｌｉｎｄｅ法の他に新たに二つの方法、階層的クラスタリング法と最大距離法でクラスタリングを行なう。

3.3　クラスタリング法

3.3.1　Ｌｉｎｄｅ法

Ｌｉｎｄｅ法はK－ｍｅａｎｓ法の一種であるので、ここではまずK－ｍｅａｎｓ法について説明する。

この方法は以下のようなステップから成る。

N個のサンプルパターンの集合$L = x_1, x_2, \cdots, x_N$が与えられ、K個のグループにクラスタリングするとすると、

1）K個の初期クラスタ中心$z_1(1), z_2(1), \ldots, z_K(1)$を適当に決める。
2）k回目の繰り返しステップで、サンプルxを次の方法でK個のクラスタに分類する。
すべての$i = 1, 2, \ldots, K (i \neq j)$について、

$$\|x - z_j(k)\| < \|x - z_i(k)\|$$

であれば、$x \in S_j(k)$とする。ここに$S_j(k)$は$z_j(k)$をクラスタ中心とするサンプル集合である。
3）2）で得られた$S_j(k)$のあたらしいクラスタ中心を$z_j(k+1)$として、

$$J_j = \sum_{x \in S_j(k)} \|x - z_j(k+1)\|^2, j = 1, 2, \ldots, K$$

を最小にするように$z_j(k+1)$を決める。これは、

$$z_j(k+1) = \frac{1}{N_j} \sum_{x \in S_j(k)} x, j = 1, 2, \ldots, K$$

とすることである。ここでN_jは$S_j(k)$の数を示す。
4）全ての$j = 1, 2, \ldots, K$に対して$z_j(k+1) = z_j(k)$となれば、アルゴリズムは収束したものとして終了する。そうでなければ2）に戻る。

図3．2にその例を示す。

以上のアルゴリズムがK－ｍｅａｎｓ法であるが、Ｌｉｎｄｅ法ではK×2個の数にクラスタリングを行なう場合の1）の初期クラスタ中心を決めるさいにK個でクラスタリングして求めたクラスタ中心を用いる。具体的には、図3．3のようにする。

3.3.2 階層的クラスタリング法

本研究では階層的クラスタリングの内で重心間の最隣接距離にあるクラスタ毎にまとめていく方法を取る。この方法は以下のようなステップから成る。

1）$k = n, c_i = x_i, i = 1, 2, \ldots, n$
2）もし$k = K$ならば終了。
3）クラスタc_iとc_jとの間の距離$d(c_i, c_j)$の最小のc_i, c_jの組を見つける。
4）c_iとc_jとを一つのグループc_iにまとめてクラスタ中心を決める。
5）c_jは取り除く。$k = k - 1$として2）に戻る。

図3．4にその例を示す。

このアルゴリズムでは本来ならdの最小を見つけるために最初に$n(n-1)\div 2$組の計算を行ない、その後クラスタを一つまとめる毎に、$n-2, n-3, \ldots$回ずつの計算を行なわなければならない。その計算量は膨大なものになるが本研究ではＳＥＩＵＮのベクトル量子化の機能を使うことで高速化する（後述）。

また、こうして作られるクラスタは図３．５のようなツリー構造を成す。

3.3.3 最大距離法

この方法は以下のようなステップから成る。

1) Ｎ個のサンプルパターン集合$X = x_1, x_2, \ldots, x_N$から任意に一つクラスタ中心を持ってきて$z_1$とする。
２）z_1から最も遠い距離にある点を集合Xから求めz_2とする。
３）$D_{x_i} = min\|x_i - z_j\|, i = 1, 2, \cdots, N, j = 1, 2, \cdots, k$
４）D_{x_i}の最大値を与えるx_iを新たなクラスタ中心z_kとする。
５）$k = K$ならば終了。そうでなければ３）に戻る。

図３．６にその例を示す。

3.4 ＳＥＩＵＮを用いたクラスタリング

クラスタリングで最も時間がかかるのは距離計算と計算距離に沿ったソートである。本研究ではこの二つの処理をＳＥＩＵＮで代行することでクラスタリングを高速に行う。具体的にはＳＥＩＵＮの距離計算部”Ｄ”で距離計算を、上位選出&得点割り当て部”Ｐ”でソートを行なう。

3.4.1 距離計算部”Ｄ”

距離計算部ではサンプルパターンベクトルの集合Xを格納する辞書バッファ（本研究の実験では３３０３文字分）が４９（７×７）本、距離計算のための中心ベクトルzを格納するための特徴ベクトルバッファが４９本の辞書バッファ毎に一つづつ、計算された距離を格納する距離バッファが４９本それぞれ容易されている（図３．７）。

3.4.2 上位選出&得点割り当て部”Ｐ”

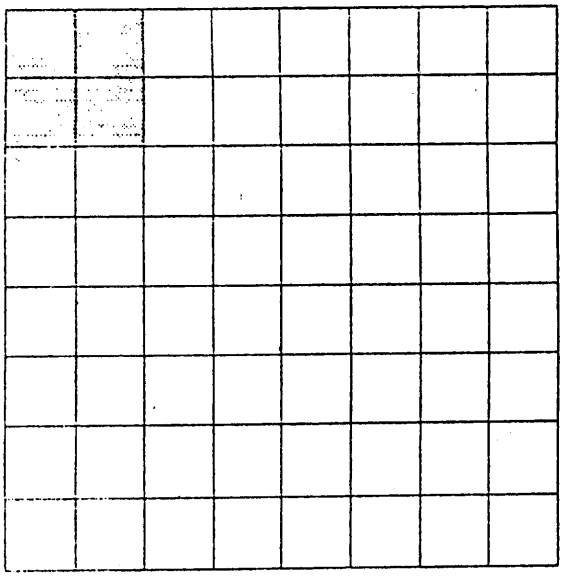
上位選出&得点割り当て部では、”Ｄ”部で計算した距離値にしたがって距離バッファ内の値をソートする（図３．８）。

3.4.3　クラスタリングプログラムのＳＥＩＵＮでの機能代行

３節で説明したクラスタリング方法の距離計算とソートの部分をアルゴリズムの違いから、次のように代行させる。

・Ｌｉｎｄｅ法…距離計算部分を”Ｄ”部で処理。
・最大距離法…距離計算部分を”Ｄ”部で処理。
・階層的クラスタリング法…距離計算部分を”Ｄ”部で、ソート部分を”Ｐ”部で処理。

	クラス内文字	字数	代表ベクトル(v1, v2, v3, v4)
クラス 1	亜 盈 乙 祁 圭 歳 益 吾 工 牙 …	188	(5, 19, 1, 3)
クラス 2	唖 鴨 斡 栢 唄 挽 押 構 格 椀 …	217	(32, 37, 0, 3)
クラス 3	娃 憶 快 怪 恢 懐 恨 傑 姐 核 …	320	(36, 14, 1, 2)
クラス 4	阿 因 蠣 賛 囲 回 慰 巨 躯 凹 …	231	(42, 31, 0, 3)
クラス 5	哀 茜 嘉 華 蓋 殻 鼓 芦 芋 雨 …	318	(1, 46, 0, 0)
クラス 6	葵 訣 講 詠 詰 該 課 詣 醤 勲 …	114	(0, 68, 0, 0)
クラス 7	穐 穏 稽 轄 袈 瑚 欧 貫 欺 磯 …	135	(22, 45, 0, 1)
クラス 8	渥 央 激 源 沿 渦 漁 汚 河 汽 …	310	(2, 3, 0, 2)
クラス 9	鰺 卸 鏡 鍵 綱 餌 宜 継 鎌 鑑 …	77	(32, 13, 10, 3)
クラス 10	宛 雲 窮 移 稼 寡 穫 稀 稿 科 …	165	(21, 33, 3, 2)
クラス 11	虻 蛙 佳 采 渠 伊 俄 侃 帰 果 …	145	(28, 5, 1, 3)
クラス 12	飴 竿 竈 或 衣 壱 引 下 希 堯 …	201	(0, 35, 0, 0)
クラス 13	絢 紺 惟 悦 峨 現 恰 慣 悟 慌 …	79	(43, 9, 1, 2)
クラス 14	鮎 蟹 官 筋 鰻 憲 宰 箇 笠 緻 …	167	(22, 23, 6, 4)
クラス 15	閣 駅 靭 鞍 靴 鞠 畦 監 顕 堅 …	33	(45, 44, 0, 2)
クラス 16	依 丘 懸 穴 岳 貴 汗 景 見 液 …	146	(13, 6, 3, 4)
クラス 17	尉 駒 敬 関 杭 期 勤 勘 競 軽 …	57	(38, 52, 0, 3)
クラス 18	飲 鉱 街 径 釧 鍬 鈷 館 行 糞 …	47	(11, 12, 21, 1)
クラス 19	鬱 鵠 際 鷗 隣	5	(59, 31, 1, 7)
クラス 20	歌 警 顧 繋 習 覆 碧 蒙 融 襲	10	(18, 71, 2, 1)

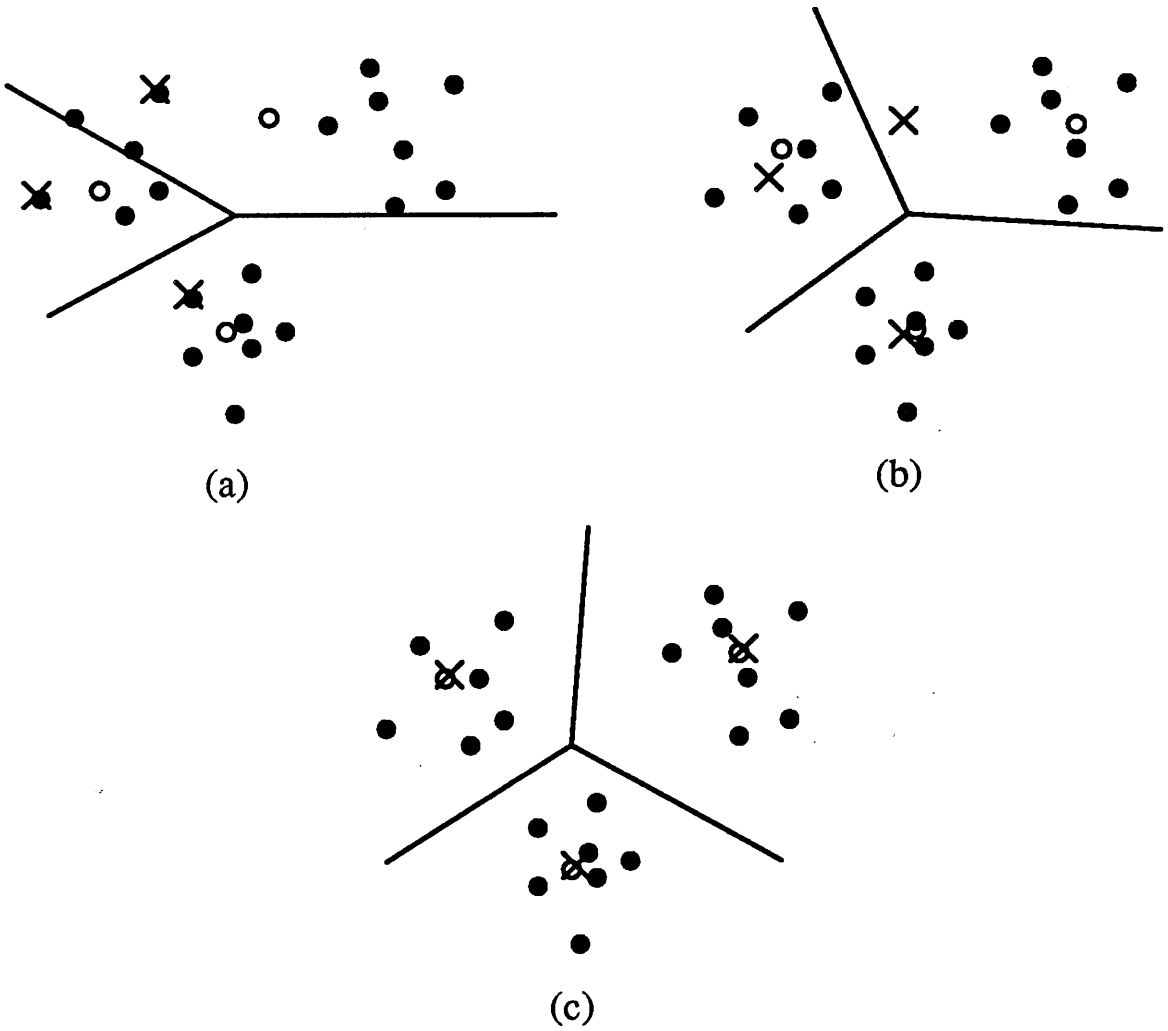


マス番号:　１番（左図の網部分）

クラス数:　２０クラス／１マス

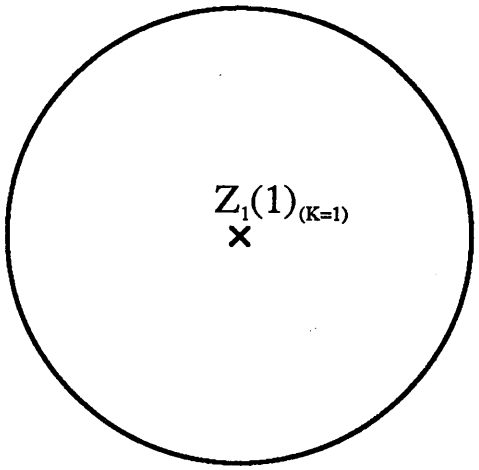
“…”は以後省略を表す

図３．１：連想辞書の例

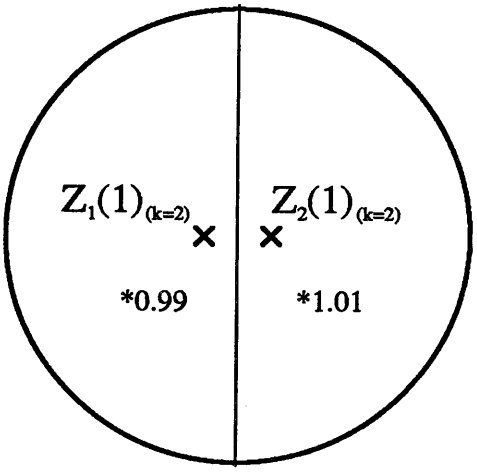


（ａ）最初のクラスタ中心（×）
　　分類後の中心（○）
（ｂ）２回目のクラスタリング
（ｃ）３回目のクラスタリング

図３．２：Ｋ－ｍｅａｎｓ法



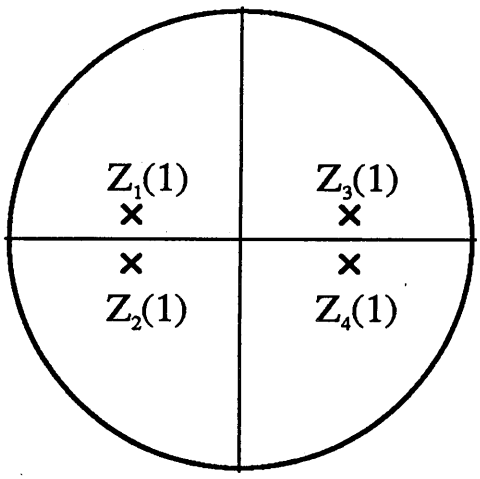
K=1



K=2

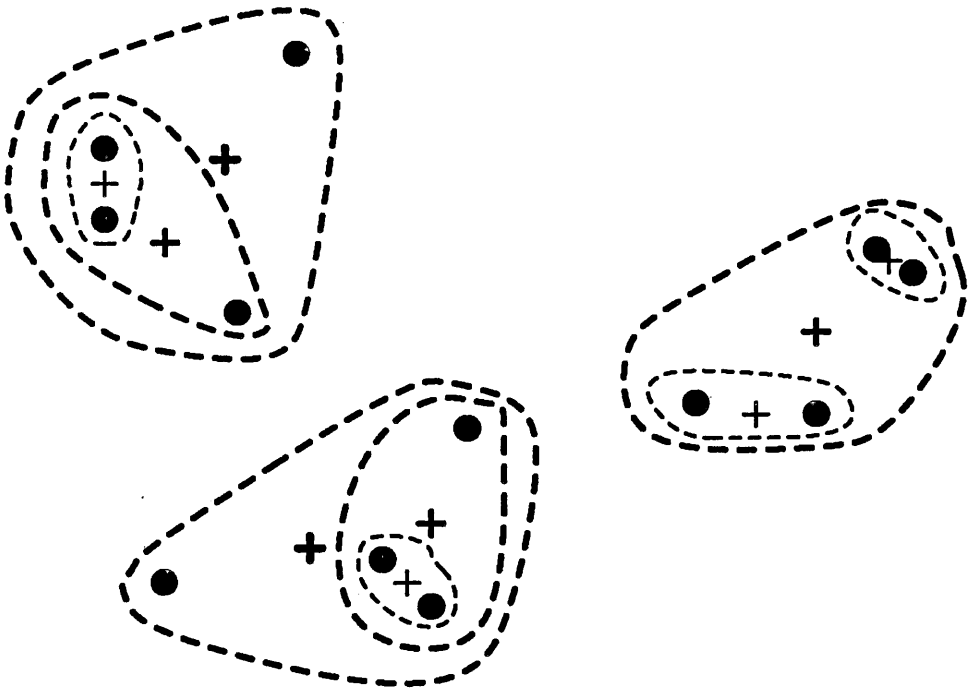
$$Z_1(1)_{(k=2)} = Z_1(1)_{(K=1)} * 0.99$$

$$Z_2(1)_{(k=2)} = Z_1(1)_{(K=1)} * 1.01$$



K=4

図３．３：初期クラスタ中心の決定



（●）サンプルパターンx_i
（＋）新しいクラスタ中心

図３．４：階層的クラスタリング法

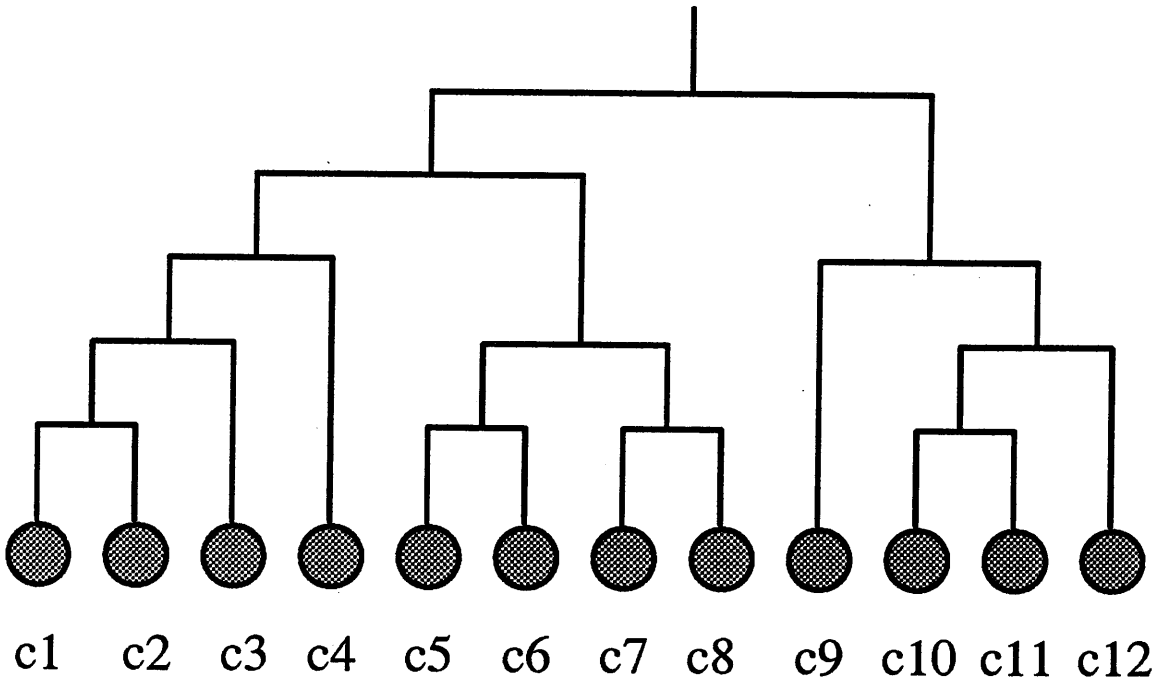
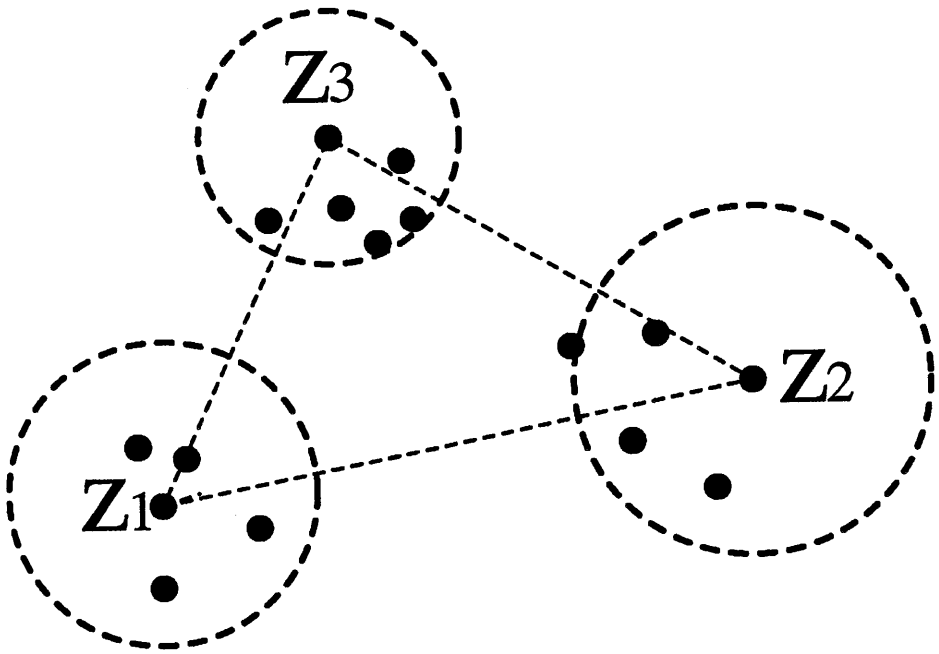


図３．５：ツリー構造の例



(●) サンプルパターンx_i

図３．６：最大距離法

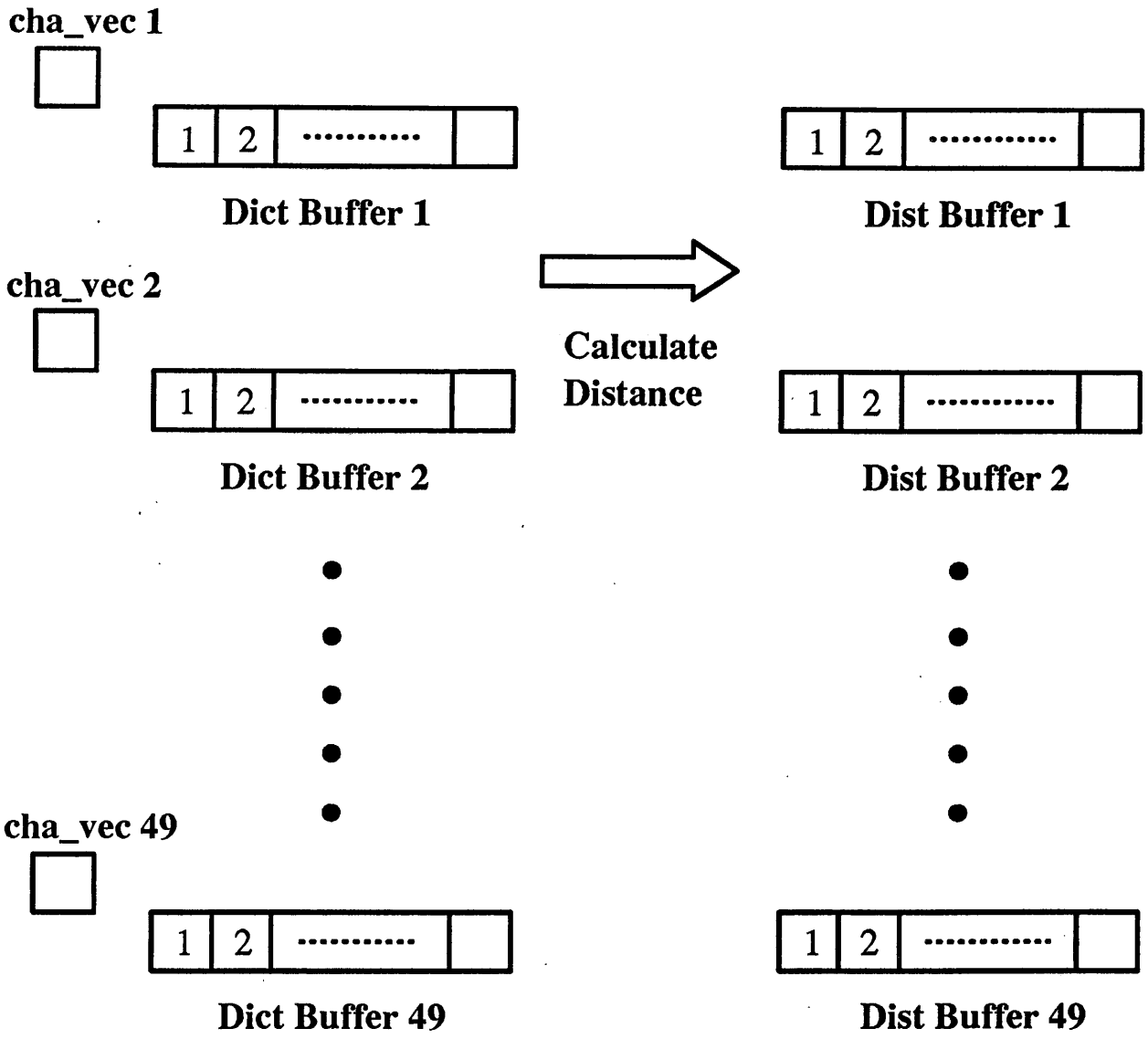

cha_vec 1
1
2
Dict Buffer 1
Dist Buffer 1
cha_vec 2
Calculate
Distance
Dict Buffer 2
Dist Buffer 2
cha_vec 49
Dict Buffer 49
Dist Buffer 49

図３．７：距離計算部”Ｄ”

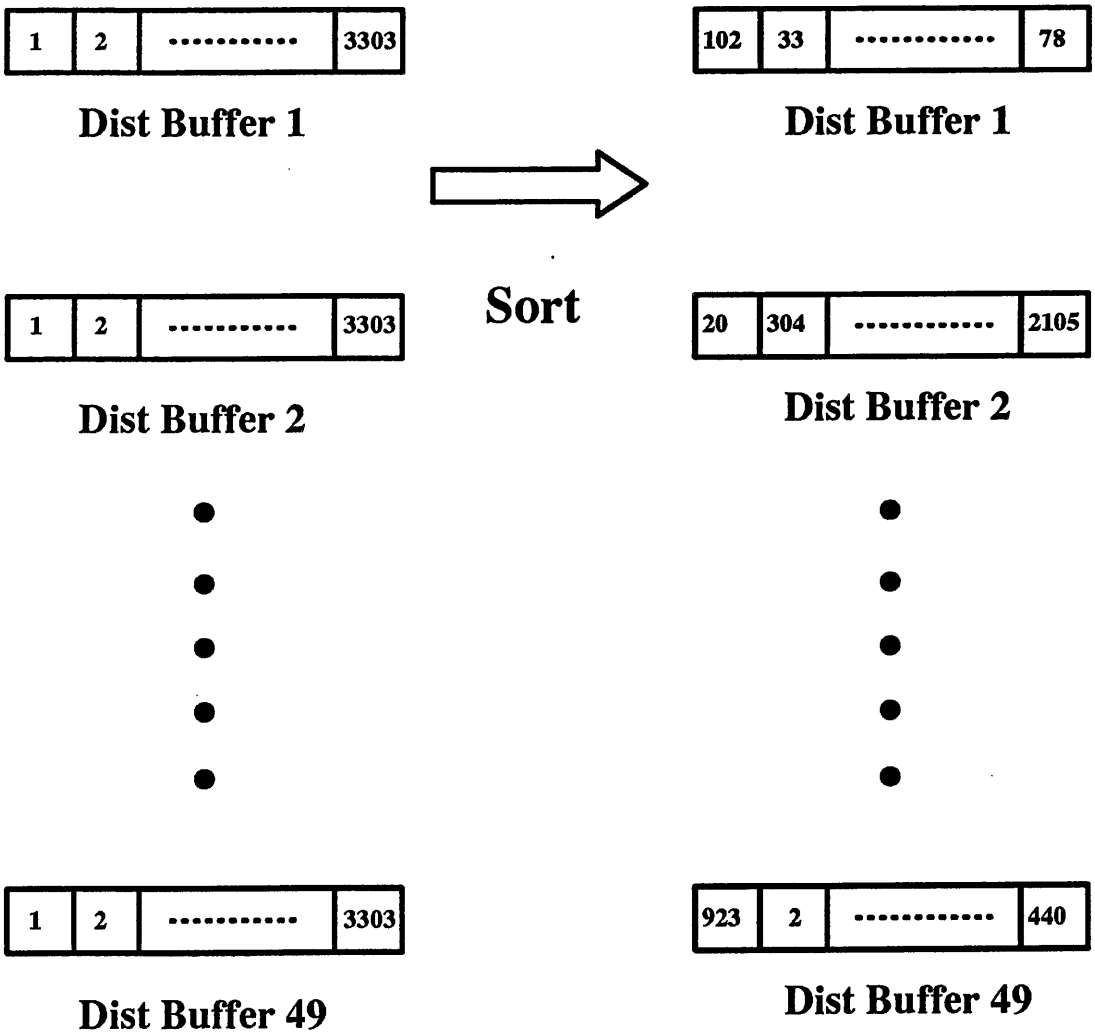


図３．８：上位選出&得点割り当て部”Ｐ”

4 実験

4.1 実験準備

第3章で述べた3種のアルゴリズムでそれぞれにクラスタリングした連想辞書を用いて認識実験を行なう。認識のプログラムは記号や誤認識しやすい文字の細分類は行なわず、ＳＥＩＵＮでの認識結果のみを認識率として使用している。連想辞書作成用ベクトルデータとしては表４．１に示した活字文字のイメージデータ（３３０３文字) ×１０セットの平均ベクトルを取り使用した。またグラフや表中の省略名に関しては、表４．２、表４．３、表４．４に示してある。

4.2 クローズ実験

この実験では、連想辞書作成に使用したイメージデータの内から４種類を選んで認識実験を行なった。表４．２に使用したイメージデータを示す。表４．５（ａ)、表４．５（ｂ)、表４．５（ｃ）にそれぞれのクラスタリング法で作成した連想辞書毎の１位認識率から３位認識率までを示す。括弧の中の数字は３３０３文字中何文字認識したかを表す。また、グラフ４．１（ａ)、グラフ４．１（ｂ)、グラフ４．１（ｃ）にはそれぞれ１位認識率から３位認識率毎のクラスタリング方法での比較結果を示してある。

4.3 オープン実験

この実験では、連想辞書作成に使用したイメージデータ以外のデータを４種類使用して認識実験を行なった。表４．３に使用したイメージデータを示す。表４．６（ａ)、表４．６（ｂ)、表４．６（ｃ）にそれぞれクローズ実験と同様な表を示す。また、グラフ４．２（ａ)、グラフ４．２（ｂ)、グラフ４．２（ｃ）にも同様な比較結果を示す。

4.4 総合結果

クローズ実験とオープン実験での総合認識結果を表４．７とグラフ４．３（ａ)、グラフ４．３（ｂ)、グラフ４．３（ｃ）に示す。

4.5 実験結果に関する考察

4.5.1 認識率

まず、それぞれのクラスタリング法での認識率の傾向を見てみる。Ｌｉｎｄｅ法では最も認識率が良かったのがクラスタ数１６のときであった。また認識率のクラスタ数の変化による認識率の差は三つもうちで最も大きく変動

している。階層的クラスタリング法では最も認識率が良いのがクラスタ数３２のときで、クラスタ数の変化による認識率の差が三つのうちで最も少ない。最大距離法ではクラスタ数１６のときが良く、変化の差は中間である。

結果としては認識率で最も良いのがＬｉｎｄｅ法と最大距離法となった。ただし実際には誤認識された文字は記号がほとんどで、漢字や平仮名、片仮名、アルファベットなどは誤認識はほとんどされなかった。誤認識の例は表４．８に示した。したがって三つの方法のうちどの方法でも十分であると言える。

また今回の実験ではオープン実験の方がクローズ実験よりも認識率がよくなっているが、本来ならばクローズ実験の方がよくなるはずである。これは、クローズ実験では一つレーザープリンタの縦倍の活字の認識率が非常に悪く、オープン実験の認識実験のときのレーザープリンタの活字が全角だったことに起因すると思われる。

4.5.2 速度比較

今回の実験ではＳＥＩＵＮとのデータをやりとりするドライバ部分がまだ完全ではなかったため、クラスタリングプログラムをワークステーション上で動かして連想辞書を作成した。そこで理論的にはどのクラスタリング方法がどの程度早くなるかを考えてみる。

まず、Ｌｉｎｄｅ法と最大距離法ではアルゴリズム上３３０３文字分の距離値をＳＥＩＵＮとワークステーション上（ＬＵＮＡ８８Ｋ）で何度もやりとりをしなくてはならず、両者の間のデータ転送速度がかなり遅いため３０パーセント程度の速度向上しか望めない。また、Ｌｉｎｄｅ法と最大距離法とを比較すると、最もよい認識率を示したクラスタ数１６個でクラスタリングする場合、最大距離法の方が計算量が少なくてすむ。これはＬｉｎｄｅ法ではクラスタ中心が収束するまでに距離計算を何度も繰り返さなければならないからである。

階層的クラスタリング法ではＳＥＩＵＮとのデータのやりとりは前の二つの方法より比較的少なくてすみ、ソートの機能まで代行させることが出来る。したがってＳＥＩＵＮの機能を十分に発揮した高速クラスタリングを行なうという点では階層的クラスタリング法がよいと思われる。

表４．１：連想辞書作成用の活字文字のイメージデータ

1）エプソンプリンター印刷文字 ４倍角 濃

2）エプソンプリンター印刷文字 ４倍角 淡

3）エプソンプリンター印刷文字 縦倍角 コピー１回

4）エプソンプリンター印刷文字 全角 淡

5）富士通プリンター印刷文字 ４倍角 Ｎｏ．２

6）富士通プリンター印刷文字 横倍角 Ｎｏ．２ コピー１回

7）レーザープリンター印刷文字 ４倍角 コピー２回

8）レーザープリンター印刷文字 縦倍角 Ｎｏ．２

9）レーザープリンター印刷文字 全角 Ｎｏ．１

１０）ＮＥＣプリンター印刷文字 ４倍角

表４．２：クローズ実験用の活字文字のイメージデータ

１）エプソンプリンター印刷文字 ４倍角 濃（４ｂａｉ＿ｈｉ）
２）富士通プリンター印刷文字 横倍角 Ｎｏ．２ コピー１回
（ｆｍｙｂ２ｃ１）
３）レーザープリンター印刷文字 縦倍角 Ｎｏ．２
（ｌａｔｂａｉｇ２）
４）ＮＥＣプリンター印刷文字 ４倍角 （ｎｅ４ｂ１ｃ０）

表４．３：オープン実験用の活字文字のイメージデータ

１）エプソンプリンター印刷文字 ４倍角 コピー１回
（４ｂａｉｃｐ）
２）富士通プリンター印刷文字 全角 Ｎｏ．２ コピー１回
（ｆｍｚｎ２ｃ１）
３）レーザープリンター印刷文字 全角 Ｎｏ．２
（ｌａｚｅｎｇ２）
４）ＮＥＣプリンター印刷文字 縦倍角 Ｎｏ．１
（ｎｅｔｂ１ｃ０）

表４．４：グラフ中の略称

ＬＢＧ … Ｌｉｎｄｅ法
Ｔｒｅｅ … 階層的クラスタリング法
Ｄｉｓｔ … 最大距離法

表４．５（ａ）：Ｌｉｎｄｅ法、クローズ実験

		4bai_hi	fmyb2c1	latbaig2	ne4b1c0	average
2	1	83.833(2769)	72.328(2389)	54.829(1811)	84.771(2800)	73.925(2442)
	2	91.750(3029)	82.531(2726)	66.061(2182)	92.643(3060)	83.235(2749)
	3	94.672(3127)	87.133(2878)	71.178(2351)	95.247(3146)	87.058(2876)
4	1	98.517(3254)	95.833(3167)	83.439(2756)	98.395(3250)	94.059(3107)
	2	99.728(3294)	97.941(3235)	88.314(2917)	99.576(3289)	96.390(3184)
	3	99.818(3297)	98.305(3247)	89.585(2959)	99.637(3291)	96.836(3199)
8	1	99.546(3288)	97.709(3230)	87.860(2902)	99.273(3279)	96.117(3175)
	2	99.849(3298)	98.728(3261)	90.221(2980)	99.728(3294)	97.132(3208)
	3	99.849(3298)	98.910(3267)	90.615(2993)	99.818(3297)	97.298(3214)
16	1	99.485(3286)	98.486(3253)	87.769(2899)	99.364(3282)	96.276(3180)
	2	99.788(3296)	99.092(3273)	90.282(2982)	99.758(3295)	97.230(3212)
	3	99.818(3297)	99.304(3280)	90.675(2995)	99.758(3295)	97.389(3217)
32	1	99.425(3284)	98.305(3247)	87.587(2893)	99.304(3280)	96.155(3176)
	2	99.758(3295)	98.940(3268)	90.070(2975)	99.728(3294)	97.124(3208)
	3	99.788(3296)	99.122(3274)	90.584(2992)	99.788(3296)	97.321(3214)
64	1	99.152(3275)	98.214(3244)	85.165(2813)	98.910(3267)	95.360(3150)
	2	99.606(3290)	99.122(3274)	88.783(2931)	99.455(3285)	96.730(3195)
	3	99.637(3291)	99.183(3276)	89.949(2971)	99.606(3290)	97.094(3207)
128	1	98.456(3252)	96.716(3196)	81.895(2705)	98.635(3249)	93.869(3101)
	2	99.304(3280)	98.395(3250)	87.103(2877)	99.152(3275)	95.989(3171)
	3	99.455(3285)	98.910(3267)	88.647(2928)	99.394(3283)	96.602(3191)

表４．５（ｂ）：階層的クラスタリング法、クローズ実験

		4bai_hi	fmyb2c1	latbaig2	ne4b1c0	average
2	1	21.344(705)	17.469(577)	12.050(398)	21.859(722)	18.181(600)
	2	30.397(1004)	25.219(833)	19.164(633)	30.518(1008)	26.325(869)
	3	36.300(1199)	30.518(1008)	23.373(772)	35.876(1185)	31.517(1041)
4	1	90.160(2978)	82.804(2735)	67.847(2241)	90.796(2999)	82.902(2738)
	2	96.246(3179)	90.675(2995)	78.777(2602)	96.397(3184)	90.524(2990)
	3	97.941(3235)	93.279(3081)	82.501(2725)	98.062(3239)	92.946(3070)
8	1	98.607(3257)	95.792(3164)	84.499(2791)	98.486(3253)	94.346(3116)
	2	99.516(3287)	97.972(3236)	89.161(2945)	99.576(3289)	96.556(3189)
	3	99.758(3295)	98.456(3252)	90.009(2973)	99.728(3294)	96.988(3204)
16	1	99.304(3280)	97.699(3227)	87.193(2880)	98.940(3268)	95.784(3164)
	2	99.788(3296)	98.940(3268)	89.979(2972)	99.606(3290)	97.078(3206)
	3	99.818(3297)	99.183(3276)	90.554(2991)	99.728(3294)	97.321(3215)
24	1	99.334(3281)	98.305(3247)	87.254(2882)	99.213(3277)	96.027(3172)
	2	99.849(3298)	99.061(3272)	90.221(2980)	99.697(3293)	97.207(3211)
	3	99.879(3299)	99.213(3277)	90.615(2993)	99.788(3296)	97.374(3216)
32	1	99.425(3284)	98.607(3257)	86.830(2868)	99.243(3278)	96.026(3172)
	2	99.818(3297)	99.213(3277)	90.160(2978)	99.697(3293)	97.222(3211)
	3	99.879(3299)	99.243(3278)	90.463(2988)	99.788(3296)	97.343(3215)
40	1	99.516(3287)	98.456(3252)	86.800(2867)	99.152(3275)	95.981(3170)
	2	99.818(3297)	99.061(3272)	89.858(2968)	99.667(3292)	97.101(3207)
	3	99.849(3298)	99.183(3276)	90.554(2991)	99.758(3295)	97.336(3215)
48	1	99.576(3289)	98.426(3251)	86.376(2853)	99.304(3280)	95.921(3168)
	2	99.788(3296)	99.092(3273)	89.737(2964)	99.697(3293)	97.079(3207)
	3	99.849(3298)	99.243(3278)	90.433(2987)	99.758(3295)	97.321(3250)
56	1	99.394(3283)	98.395(3250)	86.073(2843)	99.213(3277)	95.769(3163)
	2	99.788(3296)	99.243(3278)	89.585(2959)	99.728(3294)	97.086(3207)
	3	99.818(3297)	99.273(3279)	90.251(2981)	99.788(3296)	97.283(3213)
64	1	99.364(3282)	98.486(3253)	85.801(2834)	99.183(3276)	95.709(3161)
	2	99.728(3294)	99.273(3279)	89.464(2955)	99.728(3294)	97.056(3206)
	3	99.818(3297)	99.273(3279)	90.130(2977)	99.788(3296)	97.252(3212)
128	1	99.273(3279)	98.002(3237)	83.470(2757)	98.850(3265)	94.900(3135)
	2	99.546(3288)	98.971(3269)	88.102(2910)	99.546(3288)	96.541(3189)
	3	99.637(3291)	99.183(3276)	89.161(2945)	99.637(3291)	96.905(3201)

表４．５（ｃ）：最大距離法、クローズ実験

		4bai_hi	fmyb2c1	latbaig2	ne4b1c0	average
2	1	70.754(2337)	54.556(1802)	44.838(1481)	70.269(2321)	60.104(1985)
	2	80.805(2669)	66.031(2181)	55.313(1827)	81.108(2679)	70.814(2339)
	3	85.014(2808)	71.935(2376)	61.490(2031)	85.680(2830)	76.030(2511)
4	1	97.003(3204)	93.249(3080)	80.503(2659)	96.882(3200)	91.909(3036)
	2	99.092(3273)	96.185(3177)	86.527(2858)	98.850(3265)	95.164(3143)
	3	99.304(3280)	97.154(3209)	88.162(2912)	99.304(3280)	95.981(3170)
8	1	99.213(3277)	97.003(3204)	87.103(2877)	98.971(3269)	95.573(3157)
	2	99.758(3295)	98.214(3244)	90.312(2983)	99.637(3291)	96.980(3203)
	3	99.818(3297)	98.395(3250)	90.645(2994)	99.788(3296)	97.162(3209)
16	1	99.516(3287)	98.365(3249)	88.374(2919)	99.394(3283)	96.412(3184)
	2	99.818(3297)	99.061(3272)	90.342(2984)	99.728(3294)	97.237(3212)
	3	99.818(3297)	99.152(3275)	90.705(2996)	99.728(3294)	97.351(3215)
24	1	99.546(3288)	98.395(3250)	88.223(2914)	99.273(3279)	96.359(3183)
	2	99.788(3296)	99.061(3272)	90.342(2984)	99.728(3294)	97.230(3211)
	3	99.818(3297)	99.183(3276)	90.584(2992)	99.788(3296)	97.343(3215)
32	1	99.516(3287)	98.214(3244)	87.526(2891)	99.213(3277)	96.117(3175)
	2	99.758(3295)	99.061(3272)	90.191(2979)	99.728(3294)	97.185(3210)
	3	99.758(3295)	99.304(3280)	90.493(2989)	99.788(3296)	97.336(3215)
40	1	99.546(3288)	98.426(3251)	87.133(2878)	99.334(3281)	96.110(3175)
	2	99.758(3295)	99.243(3278)	90.070(2975)	99.697(3293)	97.192(3210)
	3	99.758(3295)	99.425(3284)	90.403(2986)	99.758(3295)	97.336(3215)
48	1	99.455(3285)	98.517(3254)	86.588(2860)	99.273(3279)	95.958(3170)
	2	99.758(3295)	99.183(3276)	89.888(2969)	99.697(3293)	97.132(3208)
	3	99.758(3295)	99.304(3280)	90.372(2985)	99.758(3295)	97.298(3214)
56	1	99.394(3283)	98.537(3255)	86.346(2852)	99.122(3274)	95.852(3166)
	2	99.758(3295)	99.122(3274)	89.797(2966)	99.667(3292)	97.086(3207)
	3	99.758(3295)	99.304(3280)	90.282(2982)	99.728(3294)	97.268(3213)
64	1	99.485(3286)	98.426(3251)	85.952(2839)	99.122(3274)	95.746(3162)
	2	99.728(3294)	99.122(3274)	89.616(2960)	99.667(3292)	97.033(3205)
	3	99.758(3295)	99.304(3280)	90.433(2987)	99.728(3294)	97.306(3214)
128	1	98.971(3269)	97.366(3216)	83.863(2770)	98.789(3263)	94.747(3130)
	2	99.546(3288)	98.638(3258)	87.829(2901)	99.334(3281)	96.337(3182)
	3	99.697(3293)	99.061(3272)	89.071(2942)	99.546(3288)	96.844(3199)

表４．６（ａ）：Ｌｉｎｄｅ法、オープン実験

		4baicp	fmzn2c1	lazeng2	netb1c0	average
2	1	82.804(2735)	67.575(2232)	59.764(1974)	72.147(2383)	70.573(2331)
	2	91.341(3017)	79.443(2624)	72.025(2379)	83.742(2766)	81.638(2696)
	3	94.429(3119)	83.893(2771)	77.869(2572)	87.860(2902)	86.013(2841)
4	1	97.911(3234)	94.429(3119)	90.160(2978)	96.488(3187)	94.747(3129)
	2	99.455(3285)	97.669(3226)	96.094(3174)	98.971(3269)	98.047(3239)
	3	99.606(3290)	98.244(3245)	97.275(3213)	99.243(3278)	98.592(3256)
8	1	99.122(3274)	96.700(3194)	95.247(3146)	98.728(3261)	97.449(3219)
	2	99.758(3295)	98.517(3254)	98.638(3258)	99.546(3288)	99.115(3274)
	3	99.788(3296)	98.789(3263)	99.213(3277)	99.606(3290)	99.349(3281)
16	1	99.304(3280)	97.972(3236)	95.640(3159)	98.971(3269)	97.972(3236)
	2	99.667(3292)	98.910(3267)	98.698(3260)	99.576(3289)	99.213(3277)
	3	99.728(3294)	99.061(3272)	99.213(3277)	99.606(3290)	99.402(3283)
32	1	99.425(3284)	97.820(3231)	94.399(3118)	98.819(3264)	97.616(3224)
	2	99.697(3293)	98.880(3266)	97.972(3236)	99.425(3284)	98.994(3270)
	3	99.728(3294)	99.092(3273)	98.668(3259)	99.485(3286)	99.243(3278)
64	1	99.061(3272)	96.791(3197)	90.433(2987)	98.062(3239)	96.087(3174)
	2	99.546(3288)	98.668(3259)	95.792(3164)	99.152(3275)	98.290(3247)
	3	99.667(3292)	98.759(3262)	97.336(3215)	99.243(3278)	98.751(3262)
128	1	98.183(3243)	94.338(3116)	83.924(2772)	96.004(3171)	93.112(3075)
	2	99.546(3288)	97.275(3213)	91.371(3018)	98.668(3259)	96.647(3192)
	3	99.425(3284)	97.972(3236)	93.763(3097)	99.001(3270)	97.540(3222)

表４．６（ｂ）：階層的クラスタリング法、オープン実験

		4baicp	fmzn2c1	lazeng2	netb1c0	average
2	1	21.344(705)	15.471(511)	13.654(451)	17.075(564)	16.886(558)
	2	30.487(1007)	22.919(757)	20.678(683)	24.463(808)	24.587(812)
	3	36.300(1199)	28.005(925)	26.067(861)	29.549(976)	29.980(990)
4	1	89.888(2969)	79.110(2613)	73.660(2433)	85.044(2809)	81.926(2706)
	2	96.246(3179)	88.374(2919)	84.893(2804)	92.946(3070)	90.615(2993)
	3	98.002(3237)	91.735(3030)	89.979(2972)	95.459(3153)	93.794(3098)
8	1	98.305(3247)	94.611(3125)	92.522(3056)	97.215(3211)	95.663(3160)
	2	99.576(3289)	97.184(3210)	97.427(3218)	99.183(3276)	98.343(3248)
	3	99.728(3294)	97.911(3234)	98.426(3251)	99.394(3283)	98.865(3266)
16	1	98.971(3269)	96.882(3200)	94.823(3132)	98.214(3244)	97.222(3211)
	2	99.606(3290)	98.365(3249)	98.153(3242)	99.425(3284)	98.887(3266)
	3	99.667(3292)	98.789(3263)	98.971(3269)	99.546(3288)	99.243(3278)
24	1	99.152(3275)	97.639(3225)	94.672(3127)	98.880(3266)	97.586(3223)
	2	99.667(3292)	98.698(3260)	98.214(3244)	99.425(3284)	99.001(3270)
	3	99.728(3294)	98.940(3268)	99.031(3271)	99.576(3289)	99.319(3280)
32	1	99.183(3276)	97.578(3223)	95.005(3138)	98.850(3265)	97.654(3226)
	2	99.697(3293)	98.819(3264)	98.062(3239)	99.425(3284)	99.000(3270)
	3	99.697(3293)	99.122(3274)	98.880(3266)	99.546(3288)	99.311(3280)
40	1	99.183(3276)	97.639(3225)	94.278(3114)	98.819(3264)	97.480(3220)
	2	99.697(3293)	98.880(3266)	97.699(3227)	99.516(3287)	98.948(3268)
	3	99.697(3293)	99.061(3272)	98.789(3263)	99.546(3288)	99.273(3279)
48	1	99.364(3282)	97.669(3226)	93.430(3086)	98.850(3265)	97.328(3215)
	2	99.697(3293)	98.880(3266)	97.639(3225)	99.455(3285)	98.918(3267)
	3	99.697(3293)	99.122(3274)	98.607(3257)	99.516(3287)	99.236(3278)
56	1	99.364(3282)	97.578(3223)	93.158(3077)	98.638(3258)	97.185(3210)
	2	99.697(3293)	98.880(3266)	97.427(3218)	99.455(3285)	98.865(3266)
	3	99.697(3293)	99.092(3273)	98.274(3246)	99.516(3287)	99.145(3275)
64	1	99.273(3279)	97.548(3222)	92.461(3054)	98.607(3257)	96.972(3203)
	2	99.697(3293)	98.880(3266)	96.972(3203)	99.425(3284)	98.744(3261)
	3	99.697(3293)	99.092(3273)	97.941(3235)	99.516(3287)	99.062(3272)
128	1	98.940(3268)	95.792(3164)	87.345(2885)	97.850(3232)	94.982(3137)
	2	99.637(3291)	98.305(3247)	93.975(3104)	99.092(3273)	97.752(3229)
	3	99.667(3292)	98.638(3258)	96.185(3177)	99.304(3280)	98.449(3252)

表４．６（ｃ）：最大距離法、オープン実験

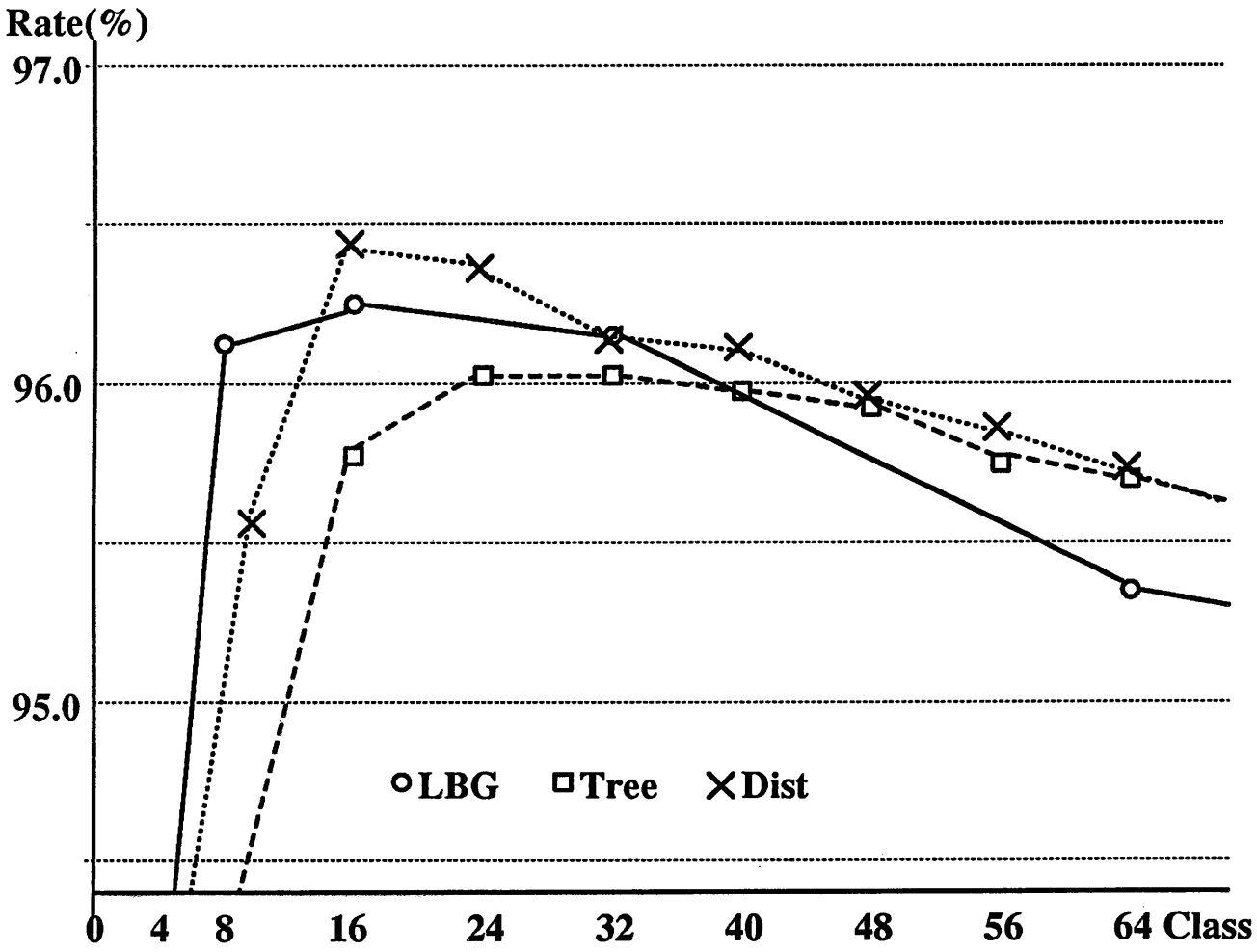
		4baicp	fmzn2c1	lazeng2	netb1c0	average
2	1	67.847(2241)	46.624(1540)	41.932(1385)	58.432(1930)	53.709(1774)
	2	78.989(2609)	57.826(1910)	54.223(1791)	70.360(2324)	65.350(2158)
	3	83.167(2747)	63.367(2093)	60.672(2004)	75.023(2478)	70.557(2331)
4	1	96.730(3195)	90.009(2937)	87.133(2878)	94.520(3122)	92.098(3042)
	2	99.001(3270)	95.065(3140)	93.854(3100)	97.790(3230)	96.428(3185)
	3	99.304(3280)	96.397(3184)	95.610(3158)	98.698(3260)	97.502(3220)
8	1	99.001(3270)	95.852(3166)	94.429(3119)	98.183(3243)	96.866(3199)
	2	99.667(3292)	98.062(3239)	98.214(3244)	99.394(3283)	98.834(3264)
	3	99.758(3295)	98.305(3247)	98.577(3256)	99.576(3289)	99.054(3272)
16	1	99.304(3280)	97.548(3222)	95.731(3162)	98.789(3263)	97.843(3232)
	2	99.758(3295)	98.607(3257)	98.638(3258)	99.516(3287)	99.130(3274)
	3	99.788(3296)	98.910(3267)	99.122(3274)	99.546(3288)	99.342(3281)
24	1	99.183(3276)	97.487(3220)	95.489(3154)	98.728(3261)	97.722(3228)
	2	99.728(3294)	98.850(3265)	98.062(3239)	99.516(3287)	99.039(3271)
	3	99.758(3295)	99.152(3275)	98.940(3268)	99.576(3289)	99.357(3282)
32	1	99.304(3280)	97.850(3232)	94.581(3124)	98.728(3261)	97.616(3224)
	2	99.667(3292)	99.061(3272)	98.002(3237)	99.425(3284)	99.039(3271)
	3	99.728(3294)	99.213(3277)	98.668(3259)	99.516(3287)	99.281(3279)
40	1	99.304(3280)	97.699(3227)	93.854(3100)	98.638(3258)	97.374(3216)
	2	99.697(3293)	98.880(3266)	97.760(3229)	99.455(3285)	98.948(3268)
	3	99.697(3293)	99.061(3272)	98.547(3255)	99.546(3288)	99.213(3277)
48	1	99.364(3282)	97.639(3225)	93.249(3080)	98.698(3260)	97.238(3212)
	2	99.697(3293)	98.789(3263)	97.275(3213)	99.394(3283)	98.789(3263)
	3	99.697(3293)	98.971(3269)	98.335(3248)	99.546(3288)	99.137(3275)
56	1	99.425(3284)	97.215(3211)	92.038(3040)	98.668(3259)	96.837(3199)
	2	99.637(3291)	98.728(3261)	96.761(3196)	99.425(3284)	98.638(3258)
	3	99.697(3293)	98.971(3269)	98.093(3240)	99.516(3287)	99.069(3272)
64	1	99.334(3281)	97.275(3213)	91.644(3027)	98.517(3254)	96.693(3197)
	2	99.637(3291)	98.850(3265)	96.549(3189)	99.364(3282)	98.600(3257)
	3	99.667(3292)	99.001(3270)	97.820(3231)	99.516(3287)	99.001(3270)
128	1	98.759(3262)	95.671(3160)	86.891(2870)	97.639(3225)	94.740(3129)
	2	99.516(3287)	97.941(3235)	93.521(3089)	99.152(3275)	97.533(3221)
	3	99.637(3291)	98.426(3251)	95.792(3164)	99.273(3279)	98.266(3246)

表４．７：総合認識率

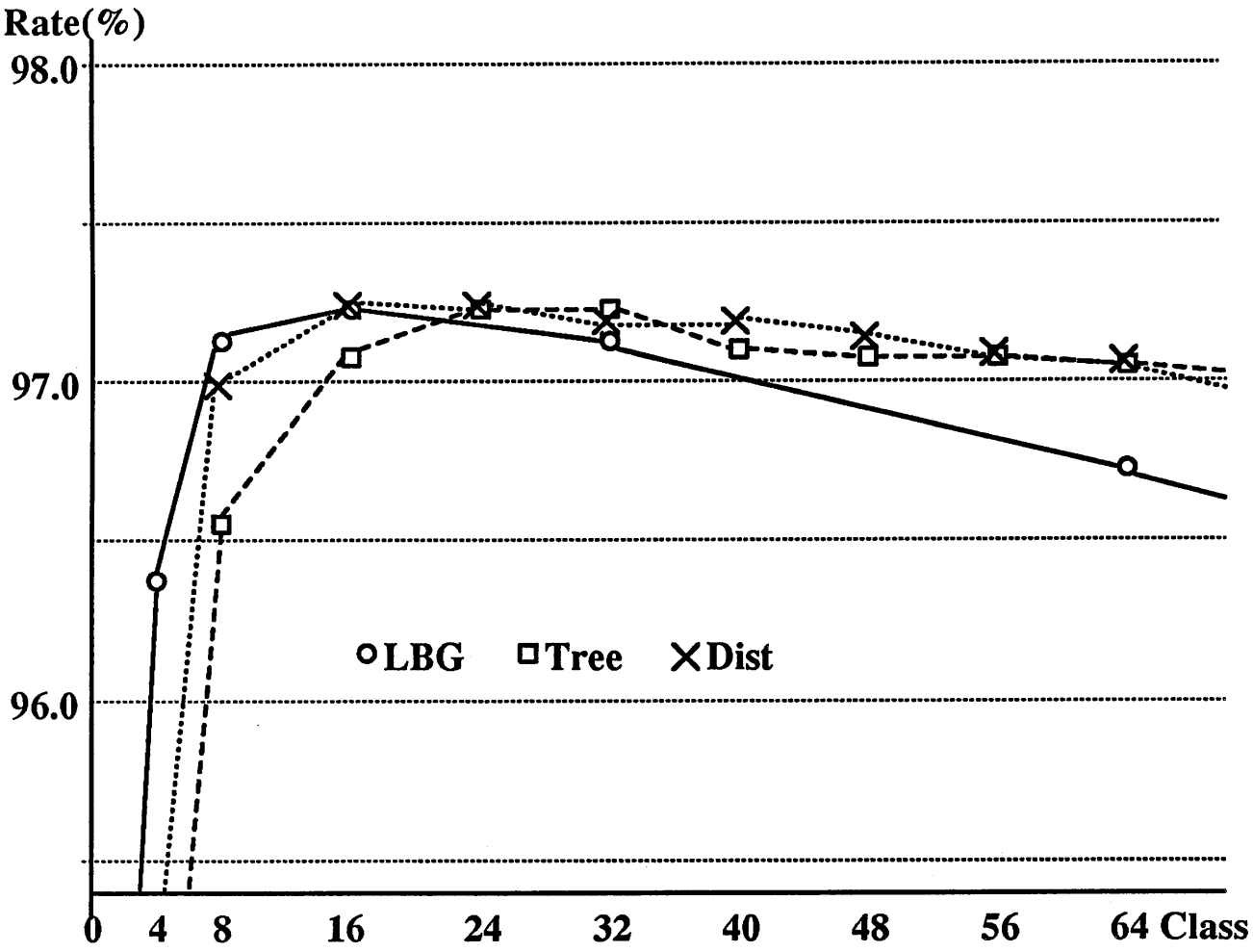
		LBG	Tree	Dist
2	1	72.249(2386)	17.534(579)	56.907(1880)
	2	82.437(2723)	25.456(841)	68.082(2249)
	3	86.536(2858)	30.749(1016)	73.294(2421)
4	1	94.403(3118)	82.414(2722)	92.004(3039)
	2	97.219(3211)	90.570(2992)	95.796(3164)
	3	97.714(3227)	93.370(3084)	96.742(3195)
8	1	96.783(3197)	95.005(3138)	96.220(3178)
	2	98.124(3241)	97.450(3219)	97.907(3234)
	3	98.324(3248)	97.927(3235)	98.108(3241)
16	1	97.124(3208)	96.503(3187)	97.128(3208)
	2	98.222(3244)	97.983(3236)	98.184(3243)
	3	98.400(3250)	98.282(3246)	98.347(3248)
24	1		96.807(3198)	97.041(3205)
	2		98.104(3240)	98.135(3241)
	3		98.347(3248)	98.350(3249)
32	1	96.886(3200)	96.840(3199)	96.867(3200)
	2	98.059(3289)	98.111(3241)	98.112(3241)
	3	98.282(3246)	98.327(3248)	98.309(3247)
40	1		96.731(3195)	96.742(3195)
	2		98.025(3238)	98.070(3239)
	3		98.305(3247)	98.275(3246)
48	1		96.625(3192)	96.598(3191)
	2		97.999(3236)	97.961(3236)
	3		98.279(3246)	98.218(3244)
56	1		96.477(3187)	96.345(3182)
	2		97.976(3236)	97.862(3232)
	3		98.214(3244)	98.169(3243)
64	1	95.724(3162)	96.341(3182)	96.220(3178)
	2	97.510(3221)	97.900(3234)	97.817(3231)
	3	97.923(3234)	98.157(3242)	98.154(3242)
128	1	93.491(3088)	94.941(3136)	94.744(3129)
	2	96.318(3181)	97.147(3209)	96.935(3202)
	3	97.071(3206)	97.677(3226)	97.555(3222)

表4．8：誤認識の例(クローズ実験、４ｂａｉ＿ｈｉ）

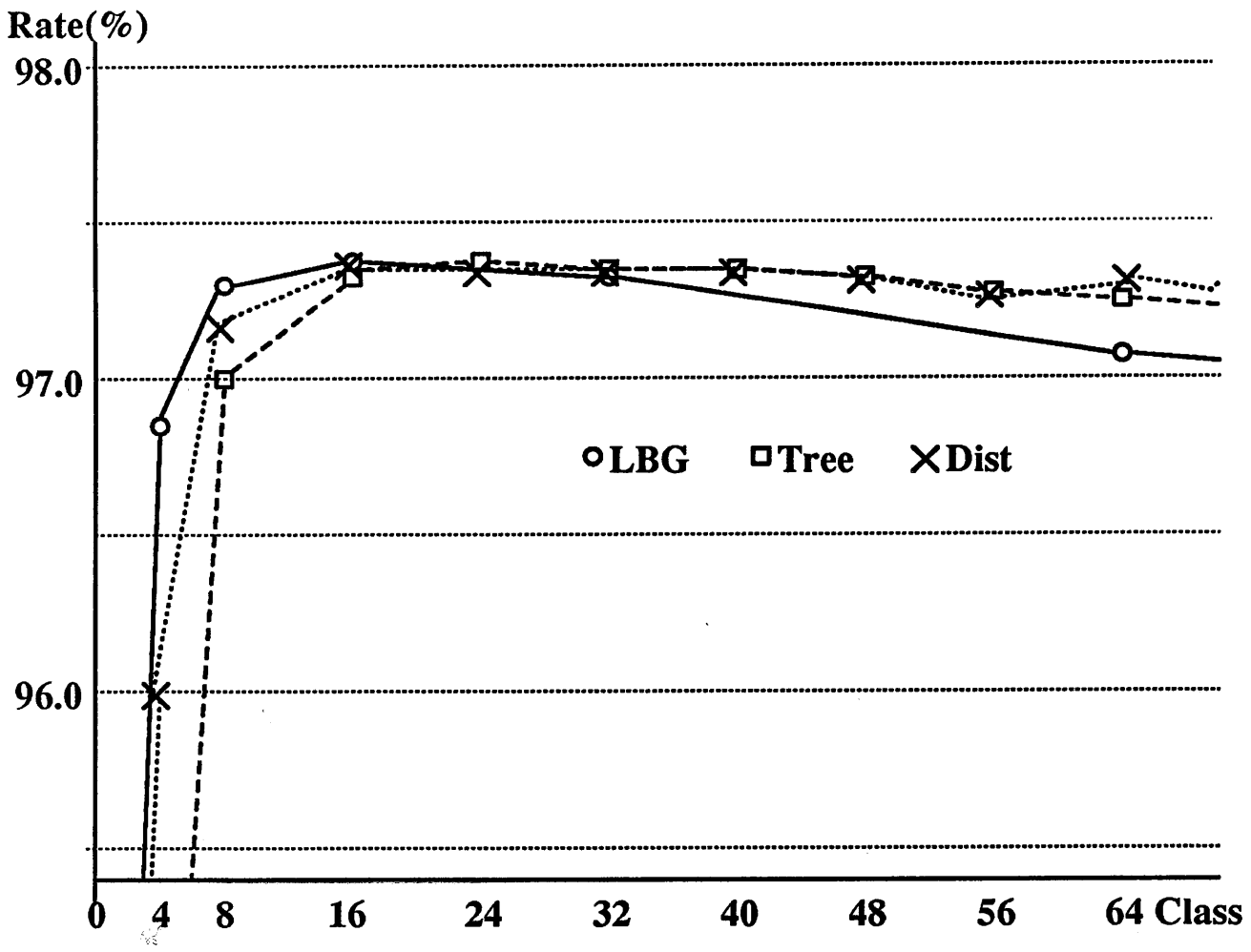
	誤認識した文字の例
Ｌｉｎｄｅ法	’、、、 ‾ 、＿、‥、一、＝、○、‐、０、ず、メ、ュ、太
階層的クラスタリング法	｡、,、、 ‾ 、＿、・、ゞ、一、‐、ケ、ュ
最大距離法	｡、,、、・、＿、一、｜、‐、〈、一、０、イ、ド、ュ、平



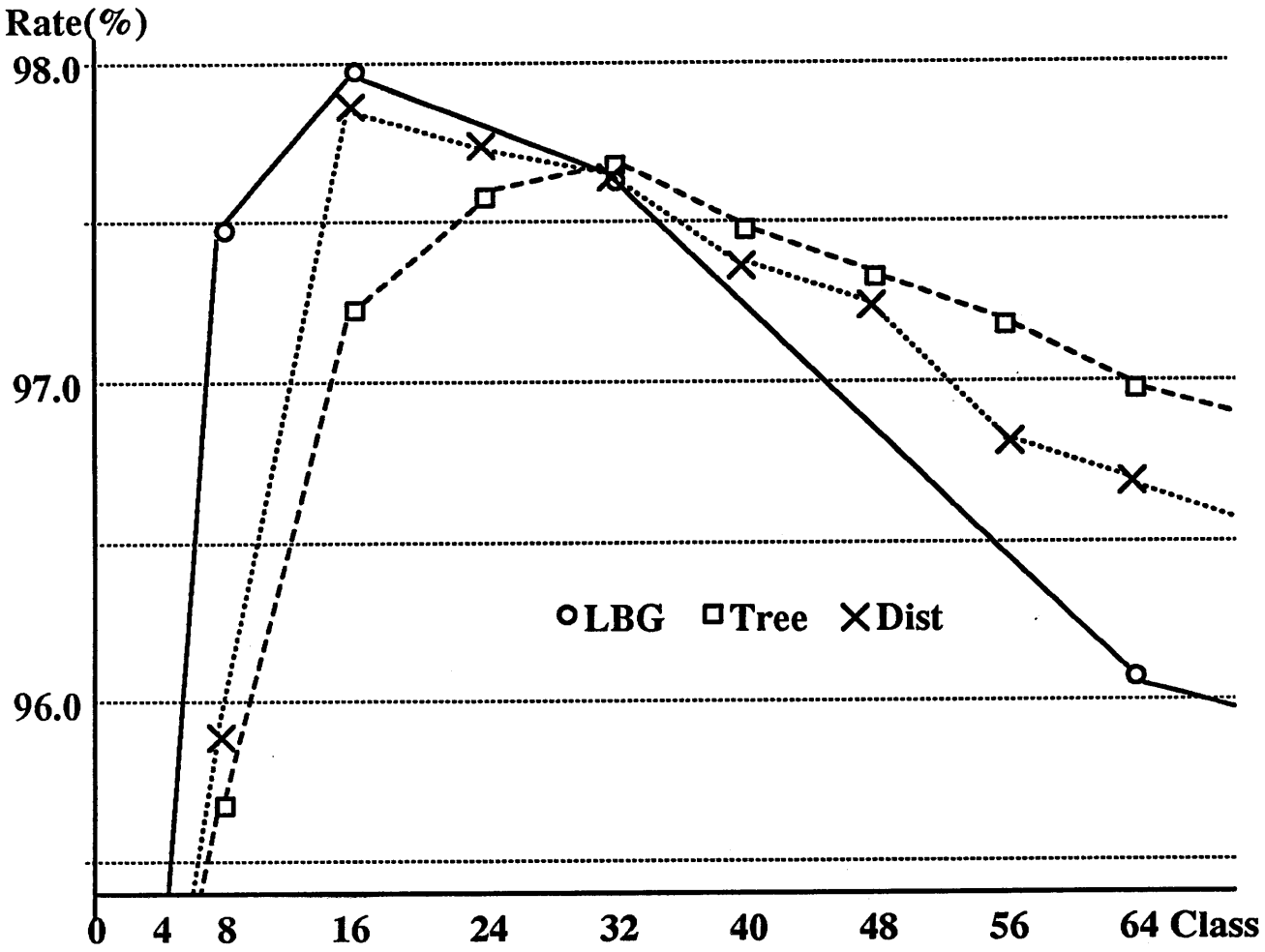
グラフ4．1（a）：一位認識率、クローズ実験



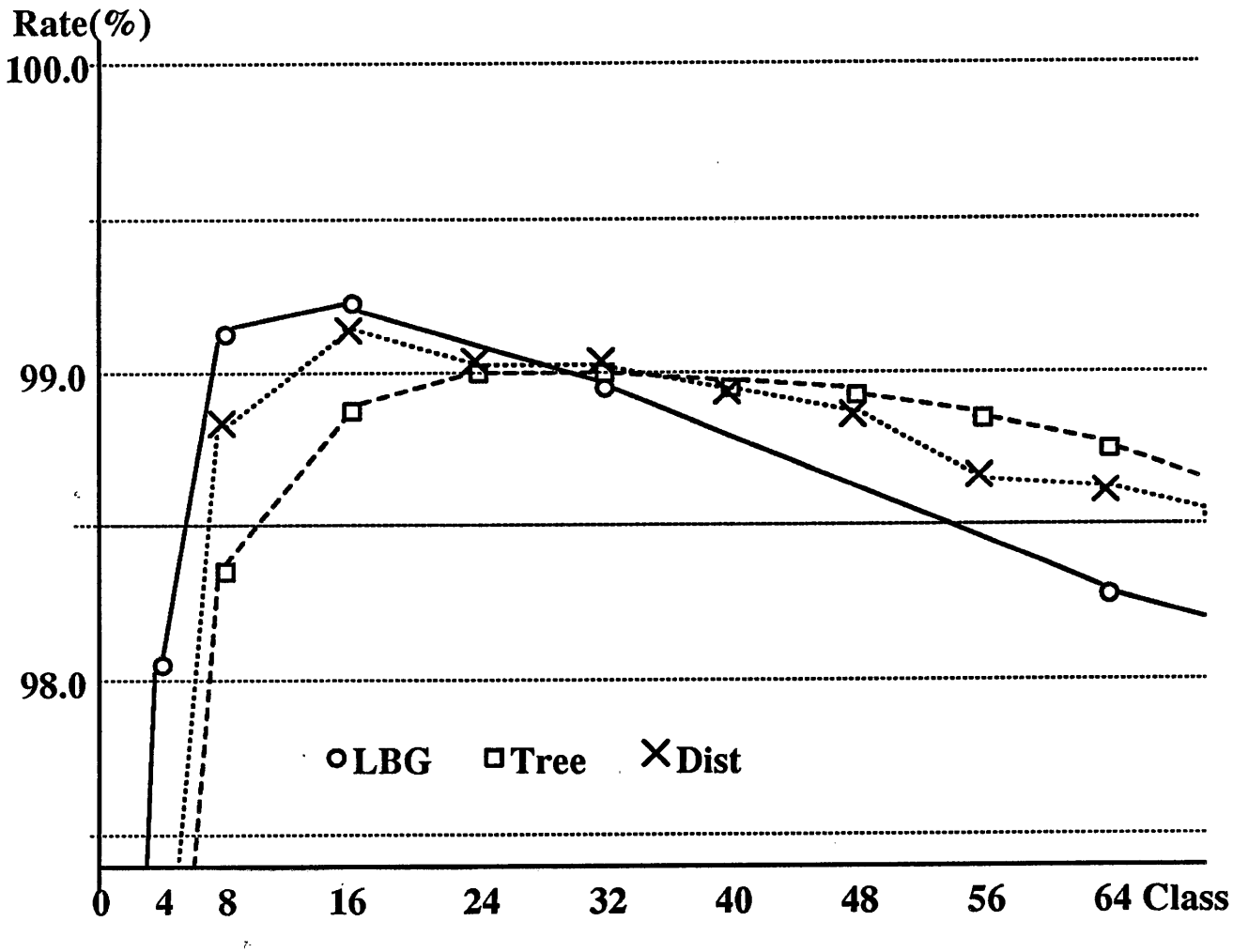
グラフ４．１（ｂ）：二位までの認識率、クローズ実験



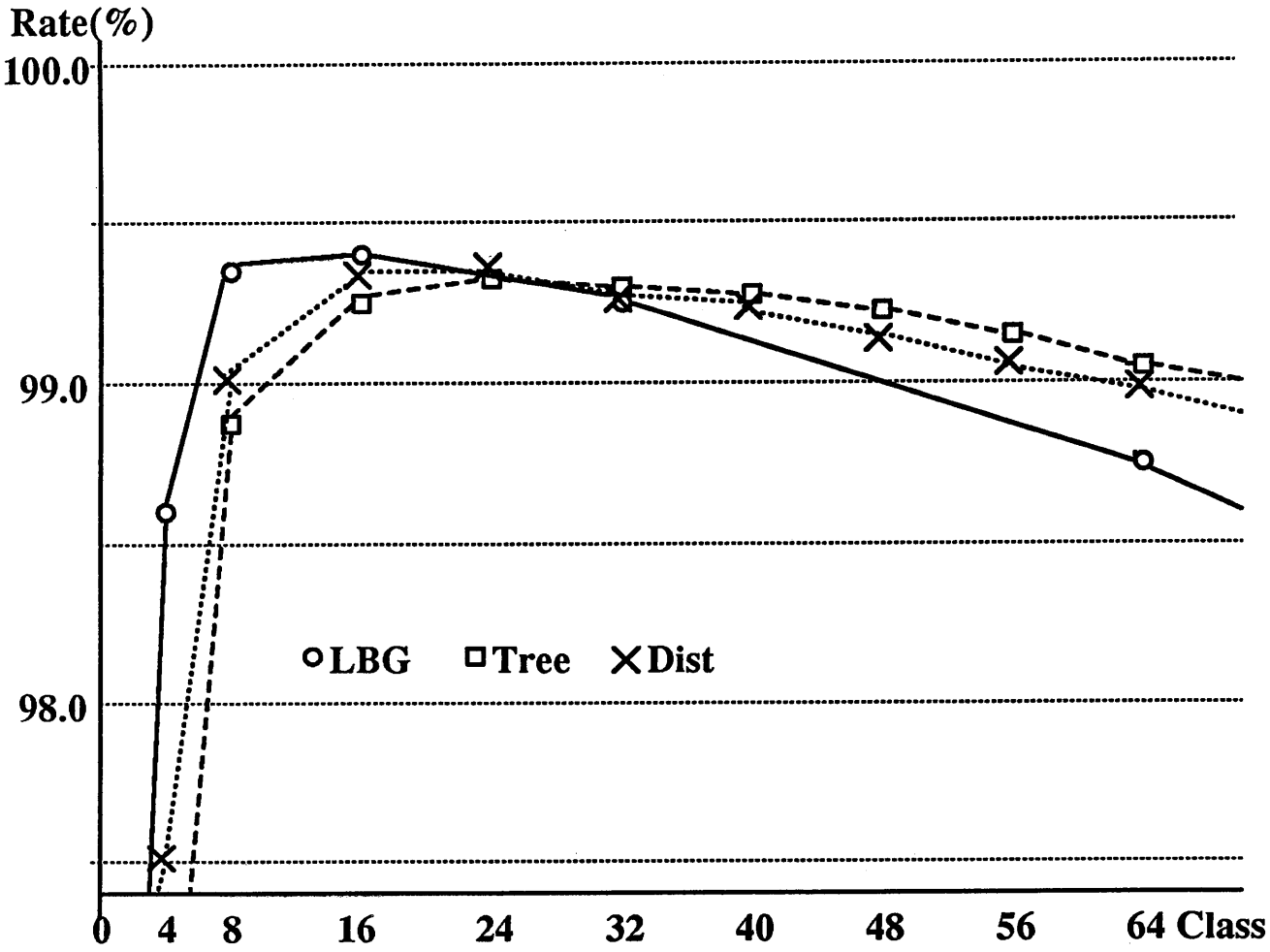
グラフ４．１（ｃ）：三位までの認識率、クローズ実験



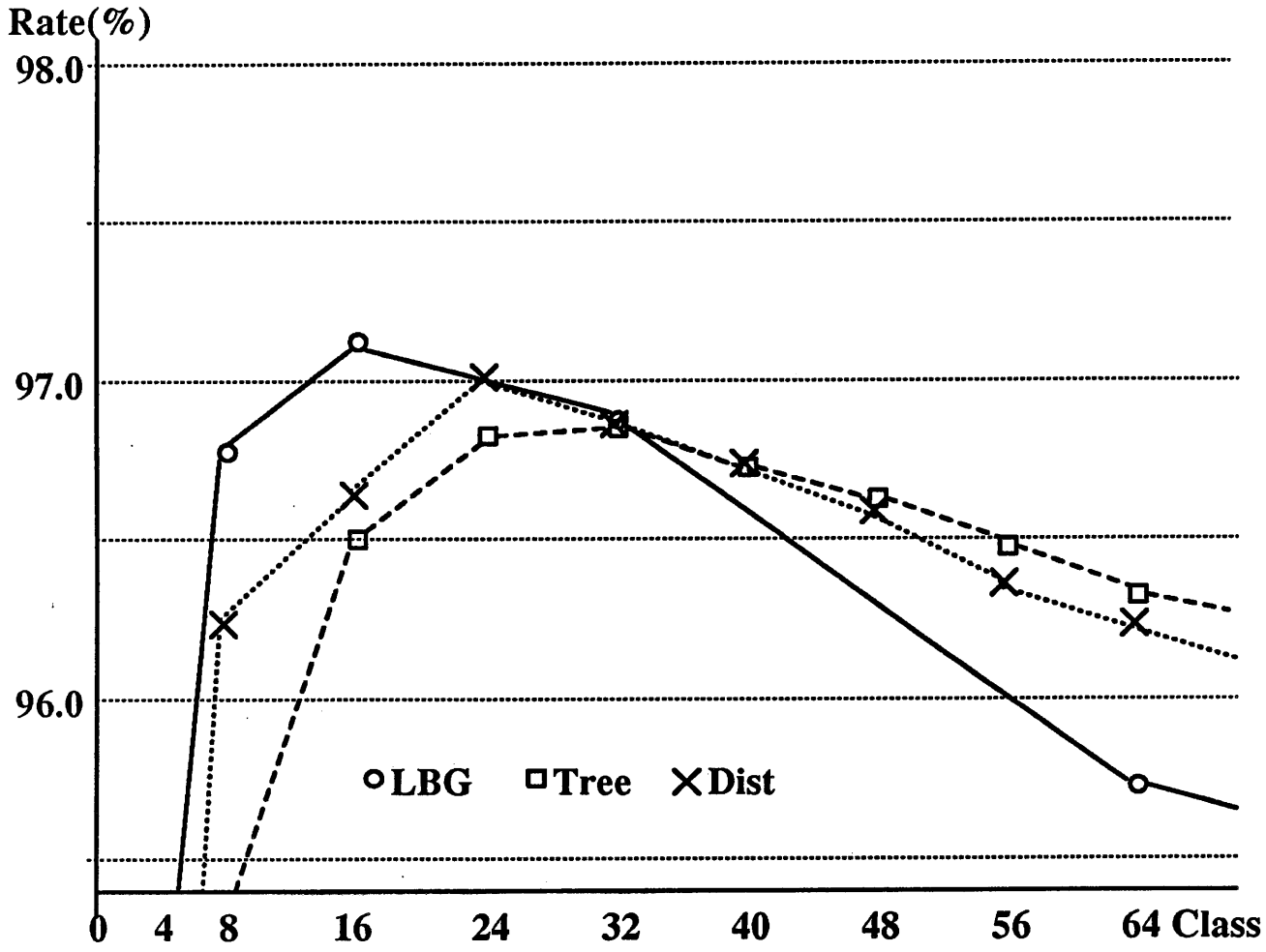
グラフ4．2（a）：一位認識率、オープン実験



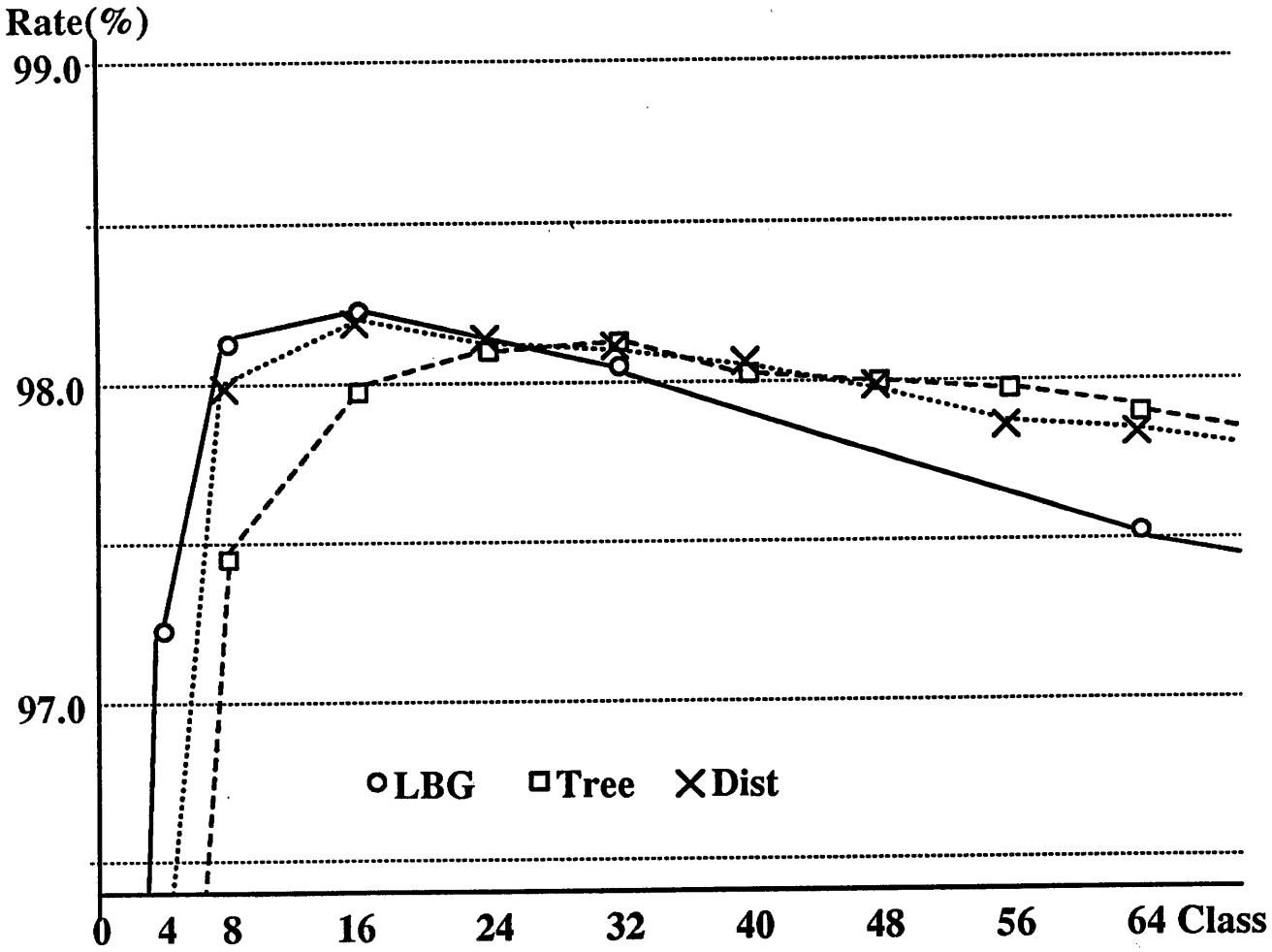
グラフ４．２（ｂ）：二位までの認識率、オープン実験



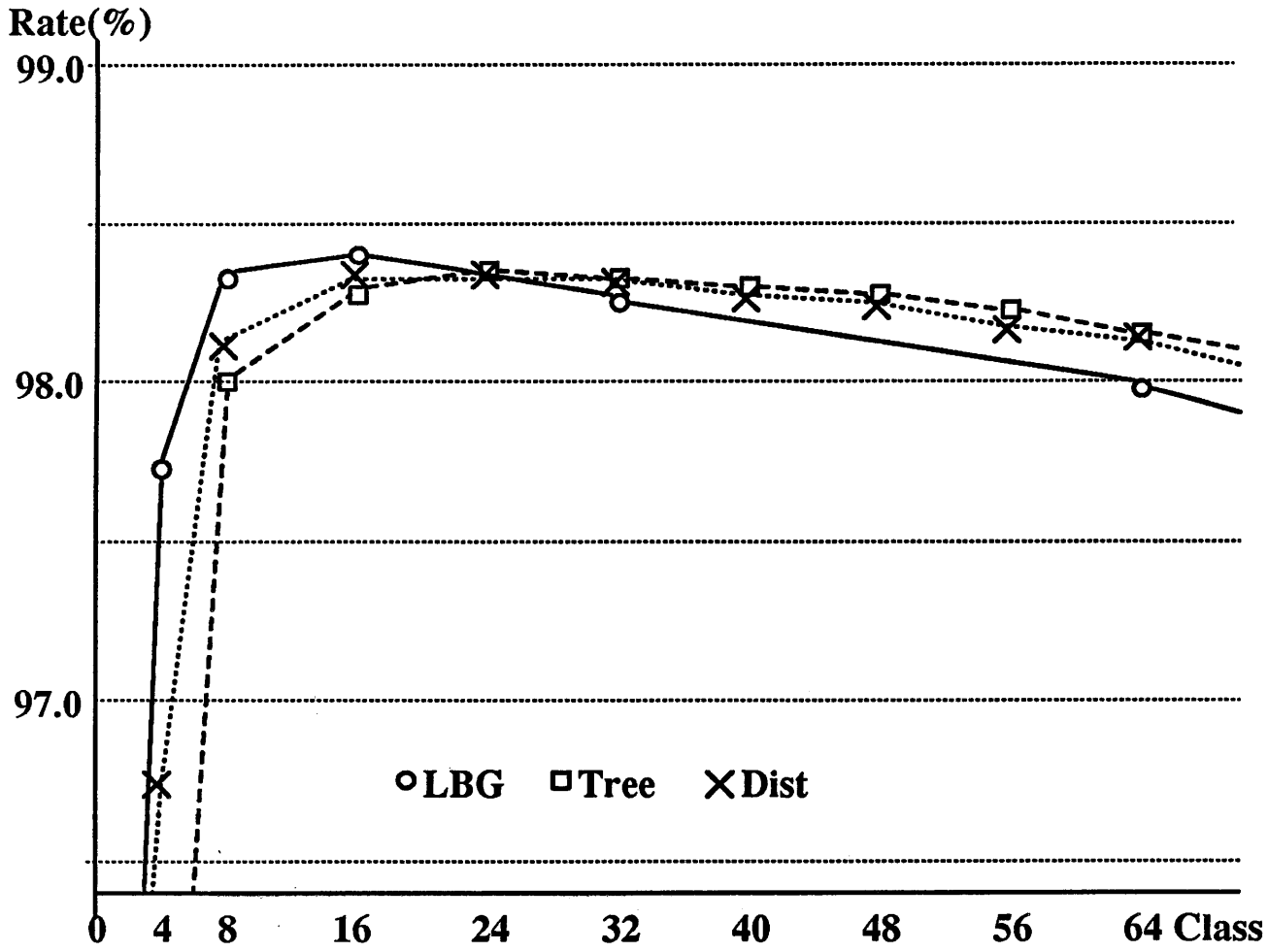
グラフ４．２（ｃ）：三位までの認識率、オープン実験



グラフ４．３（ａ）：一位認識率、クローズとオープンの平均



グラフ４．３（ｂ）：二位までの認識率、クローズとオープンの
平均



グラフ４．３（ｃ）：三位までの認識率、クローズとオープンの
平均

5　結論

本研究では、連想辞書作成をＳＥＩＵＮを用いて高速化するためのアルゴリズムについて考え、三種類のクラスタリング方法で辞書を作成しそれぞれ認識実験を行なった。また、本研究をまとめると次のようになる。

（１）各クラスタリングの方法の最も認識率のよいクラスタ数が判明した。

（２）（１）で判明したクラスタ数でクラスタリングを行なう場合、最大距離法がワークステーション上では計算時間が最も短く高速に連想辞書を作成できる。

（３）ＳＥＩＵＮの機能を利用した場合には速度的にはＬｉｎｄｅ法、最大距離法が高速化でき、階層的クラスタリングの場合にはかなりの高速化が期待できると思われる。

（４）実際の認識プログラムでは記号等の細分化を行なって認識するため、三種のクラスタリング方法の内どの方法でも十分な認識率が得られると考えられる。したがってＳＥＩＵＮの機能を利用した場合に最も高速となるクラスタリング方法を使用すれば良いと考えられる。

また、今後の課題としては、実際にＳＥＩＵＮを使ってクラスタリングを行なった場合の計算速度の比較すること、より高速で記号なんどの認識にも強いクラスタリング方式を考えることなどがあげられる。

謝辞

本研究を進めるに当たり、数多くの御助言、御指導を賜わりました、阿曽弘具教授、堀口進助教授、下平博助手に深く感謝致します。

また、大町真一郎氏には本研究全般に渡り親身な御指導、御助言を頂きました。ここに深く感謝致します。

また、日頃から数多くの御助言、御指導を頂きました、越後和徳氏、池田啓明氏に心から感謝致します。

最後に、御討論、御協力を頂き、また日頃の生活においてもお世話になりました丸岡(東)研究室の皆様に感謝致します。

参考文献

(1) 孫寧:「文字認識の高速化と高精度化に関する研究」
東北大学大学院工学研究科情報工学専攻 博士学位論文 平成2年度

(2) 木村正行ほか:「イメージ型と論理型情報処理を統合した高速・高精度の知的文字システムの研究開発」
平成2年度科学研究費補助金(特別推進研究)研究成果報告書

(3) 長尾真:「パターン情報処理」
コロナ社 電子情報通信学会編 電子情報通信学大会シリーズ1－4