

2015 年度 卒 業 論 文

TCG のカードのデザインと
テンプレート考案による制作支援

指導教員：加藤 秀行 助教
三上 浩司 准教授

メディア学部 ゲームサイエンス ゲームイノベーション プロジェクト

学籍番号 M0112248

隅田 仁士

2016 年 3 月

2015年度 卒業論文概要

論文題目

TCGのカードのデザインと
テンプレート考案による制作支援

メディア学部

氏

隅田 仁士

指導

加藤 秀行 助教

学籍番号：M0112248

名

教員

三上 浩司 准教授

キーワード

トレーディングカードゲーム (TCG)、カテゴリズ
レイアウト、製作補助ツール

近年、トレーディングカードゲーム(以下、TCG)は多くの種類が発売されており、多くのプレイヤーが存在する。TCGはコレクションが目的のトレーディングカードに対戦型カードゲームのゲーム性を組み合わせたものであり、そのゲームは対戦型のゲームである。対戦型のゲームとして、戦略を考え対戦し、勝利するのも楽しさの一つだが、トレーディングカードとしてコレクションをするのも楽しみの一つである。

TCGの熱心なファンの中にはTCGを自作する人も存在する。しかし、現状、TCGの制作には、ゼロからルールを構築し、それに合わせて、カードのレイアウトを考える必要があり、本職のデザイナーが作成するような、本格的なTCGを素人が作成するのは、非常に困難である。

そこで本研究では、素人がある程度完成度の高いTCGを容易に作成するための支援ツールを開発するために、TCGを調査し、分類分けを行った。TCGを分類することで、自作者の目的にあったレイアウトを提供することを目的とする。本研究では、既存のTCGには、本職デザイナーの経験やテクニック、ルールなどが反映されていると仮定のもと、データマイニングの手法である、階層クラスタ分析を用いて解析を行い、客観的にTCGの分類分けを行った。その結果、TCGの分類として、カードの名前とイラストのサイズで分類され、特殊なレイアウトを持つものは、特殊なレイアウトを持つもので分類でき、筆者の主観的方法に近い、分類をすることができた。

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	研究目的	2
第2章	TCG とそのデータ解析	3
2.1	TCG とは	3
2.2	TCG データ	4
2.3	データ解析	6
	2.3.1 データの数値化	6
	2.3.2 クラスタ分析	6
2.4	テンプレートレイアウトの決定法	18
2.5	分類結果	19
第3章	おわりに	33
3.1	まとめ	33
	謝辞	34
	参考文献	35
付録A章	コード集	38

目 次

2.1	TCG の各要素に関してまとめたデータ	5
2.2	類似した TCG	7
2.3	非類似 TCG	7
2.4	図 2.1 のデータを数値化したもの	9
2.5	カード分割図	10
2.6	距離行列	11
2.7	クラスタ間の距離の定義	13
2.8	図 2.6 の距離行列に対し最近距離法を適用した結果	14
2.9	図 2.6 の距離行列に対し最長距離法を適用した結果	14
2.10	図 2.6 の距離行列に対し群平均法を適用した結果	15
2.11	図 2.6 の距離行列に対し重心法を適用した結果	15
2.12	図 2.6 の距離行列に対しメディアン法を適用した結果	16
2.13	図 2.6 の距離行列に対しワード法を適用した結果	16
2.14	クラスタ分割の例	21
2.15	最短距離法での最適なクラスタ数	22
2.16	最長距離法での最適なクラスタ数	22
2.17	群平均法での最適なクラスタ数	23
2.18	ワード法での最適なクラスタ数	23
2.19	最短距離法のそれぞれのクラスタに属する TCG	24
2.20	最長距離法のそれぞれのクラスタに属する TCG	24
2.21	群平均法のそれぞれのクラスタに属する TCG	25
2.22	ワード法のそれぞれのクラスタに属する TCG	25
2.23	最短距離法の時のクラスタごとの平均値配置	26
2.24	最短距離法の時のクラスタごとの中央値配置	27
2.25	最長距離法の時のクラスタごとの平均値配置	28
2.26	最長距離法の時のクラスタごとの中央値配置	29

2.27 群平均法の時のクラスタごとの平均値配置	30
2.28 群平均法の時のクラスタごとの中央値配置	31
2.29 ウォード法の時のクラスタごとの平値配置	32
2.30 ウォード法の時のクラスタごとの中央値配置	32

第 1 章

はじめに

1.1 研究背景

近年、トレーディングカードゲーム（以下、TCG）は、非常に人気が高く、数多くの種類が発売されており、その国内市場規模は 80 億円 [1] にも及ぶ。TCG を楽しむ人の中には、自分で TCG を作成し、楽しむ人も存在する [2]。しかし、TCG を自作する場合、楽しむためのゲームのルールを自分で作成し、それに合わせてカードに配置する絵や文字、アイコンなどのレイアウトを自分で考えて決定する必要がある。したがって、作成経験が無い場合、ルールやカードの見易さを考慮したレイアウトを、気軽に作成するのは困難なものになっている。

一方、既存の TCG では、コレクションとしての見た目とカードゲームのゲーム性が両立されている。TCG の制作では、ゲームのルールにしたがい、カード上に必要なアイコンなどの要素をまとめ、その要素をカード上に配置する。そして、実際にプレイをしつつ、ルールの破綻やカードの使いにくさなどが無いかを確認し、修正を繰り返すことで、最終的なレイアウトを決定していく。例えば、TCG の中にはルール上、カードを重ねて置くものもある。カードを重ねて置くときに、下のカードに書かれた情報を参照する TCG もあるため、このような要素が隠れてしまわないように配置する必要がある。また、数枚のカードを手を持つ際、カードが重り見えなくなる部分に、

そのカードの使用に重要な情報が書かれていると、カードの確認に手間がかかることが想定される。TCG の作成では、このようにゲームのルールも考慮してレイアウトを考える必要がある。

これまでの TCG の研究の中には、コンピュータに戦略を学習させる研究 [3, 4] や、競技性の向上を目的とした研究 [5]、など、TCG のゲーム性に着目した研究がいくつかある。他にも、TCG にデジタル要素を取り入れた研究 [4] や、アンケート作成システムを利用して戦略の立案を支援するといった研究 [6] などもある。また、TCG 以外の、カードゲームシステムについての研究 [7] もある。しかし、TCG デザインのための要素レイアウトの研究は存在しない。

1.2 研究目的

既存の TCG には、本職のデザイナーの経験やテクニックが反映されているものと考えられる。そこで、既存の TCG を客観的手法を用い解析することで、TCG におけるレイアウトの特徴を抽出し、それをもとにレイアウトのテンプレートを提供することで、作成未経験者にも本職のデザイナーが作成したようなレイアウト性が高く、ゲームをプレイしやすい TCG の作成を補助できると考えられる。そのために、既存 TCG のデータをデータマイニングの手法の一つであるクラスタ分析により数種類のカテゴリーに分類し、この分類をもとに、TCG の要素を配置し、テンプレートを作成する手法を提案した。

第 2 章

TCG とそのデータ解析

2.1 TCG とは

TCG とはトレーディングカードゲームの略称で、収集が主目的であるトレーディングカードと、遊ぶことを目的としたカードゲームの要素を組み合わせたものである。代表的な TCG として「マジック・ザ・ギャザリング」[8] や「遊 戯 王オフィシャルカードゲーム」[9] , 「ポケモンカードゲーム」[10] などが存在する。多くの TCG は、定期的に同種の新たなパッケージ（商品）を販売し、トレードするカードが増えるため、コレクション要素が高い。

TCG の多くの銘柄では、デッキと呼ばれるカードの束を用いて対戦する。デッキはプレイヤーが収集したカードの中から規定枚数（通常、50 枚程度）になるように選びまとめたものである。細かいルールは TCG ごとに異なるが、一般的な TCG では、手札として 5 枚ほどのカードを、デッキから引き、規定のタイミングで手札からカードを指定の場に出し、モンスターを召喚したり、効果を発動していく。

一般的な TCG には、裏表が存在する。これは、デッキから手元に来るまで、どんなカードか分からないようにすためである。裏面は見た目を揃える必要があり、したがって TCG のカードの名前やイラストなどの要素は表面に集約される。

TCG の中には、アーケードゲーム版のトレーディングアーケードカードゲーム (TCAG) [11] やインターネットと連動してプレイする TCG も存在する。また、実際にカードを買うのではなく、インターネット上だけでプレイするソーシャルゲームやオンラインゲーム [12] も登場している。これらのアーケードゲーム版やインターネットと連動するタイプのカードゲームでは、一般的な TCG とは違い、カードの情報をコンピュータ上で読み取るため、裏表の制限がなく、表一面にイラストを配置し、裏面に効果などのカードの情報を載せることが可能になっている。本研究では、上記のようなアーケードゲームやインターネットと連動するような TCG は対象にせず、基本的なアナログの TCG のみを対象とする。

2.2 TCG データ

TCG のレイアウトには、カードの名前やイラスト、効果やカードの種類を表すアイコン、カードの効果を説明するためのテキストなどの様々な要素が存在する。これら要素は TCG の各銘柄ごとに配置する場所が異なる。ゲームのルールにより、カードを重ね合わせて置いたり、カードの向きを変えたりするため、それらを考慮し配置する必要がある。以下の図 2.1 は、64 銘柄に対し、20 個の要素を調査し、まとめた結果である。

タイトル	年代	会社	原作有無	メディア	カード名	絵	召喚コスト	種族	属性	種類	文字枠	カード枠	パワー	パワー-2	エキス/レシジョン	発生マナ	レベル	効果アイコン	フレーバー	※CMR/CN※
MTG	1993 8	WotC	オリジナル	漫画・ボード	上枠	名前下	名前右横	文字枠の上枠	右上	種族回枠	下	四角	右下		種族と同枠				あり・なし	あり
ポケモン	1996 10	ポケモン	あり	-	左端	上	名前上	文字枠の上枠	右上	種族回枠	下	四角	右下		種族と同枠		左上	あり	あり	あり
モンコレ	1997 8	ポケモン	あり	-	上	中心	名前上	文字枠の上枠	右上	種族回枠	下	なし	属性の隣		種族と同枠		左上	あり	あり	あり
遊戯王	1999 2	コナミ	あり/なし	漫画・アニメ	上枠	中心	中心	文字枠の上枠	名前右横	属性と同	下	四角	右下	右下	種族と同枠		左上	あり	なし	あり
ガンウオ	1999 2	バンダイ	あり	-	文字枠上	中心	中心	文字枠の上枠	名前右横	属性と同	下	四角	右下	右下	種族と同枠		左上	あり	あり	あり
アクエリ	1999 7	ブロッコリー	オリジナル	アニメ・本	下枠	中心	中心	名前下左	右上	種族回枠	下	なし	左下	右下	種族と同枠		左上	あり	あり	あり
DM	2002 5	タカラ/HotC	オリジナル	アニメ	上	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	なし	あり
ディメ・ゼ	2005 9	ブロッコリー	オリジナル	-	上	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
クルセイダ	2007 10	バンダイ	あり	-	文字枠上	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
WS	2008 3	ブシロード	あり	-	下枠	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
VS	2008 7	ブシロード	あり	-	下枠	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
バトスピ	2008 9	バンダイ	オリジナル	アニメ他	絵の下枠	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
ミラバト	2009 12	バンダイ	あり/なし	漫画	左端	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
カオス	2009 3	ブシロード	あり	-	下枠	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
ブレメ	2010 5	ブシロード	あり	-	文字枠上	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
ネグザ	2011 10	バンダイ	あり	-	文字枠上	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
VG	2011 2	ブシロード	オリジナル	アニメ	下枠	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
プリコネ	2011 9	セガ	あり	-	文字枠上	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
三国志大戦	2012 3	セガ	オリジナル	-	左枠	左端	左端	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
Z/X	2012 7	ブロッコリー	オリジナル	グッズ他	文字枠上枠	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
アンジュ	2013 10	角川	オリジナル	ゲーム・本	下枠上段	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
アンリミ	2013 3	エスエイジヤパン	あり	-	下枠	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
ジーク	2013 6	グリム	あり	ゲーム	テキスト下	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
ラスクロ	2013 9	ブシロード	オリジナル	WEB	絵の左下	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
BF	2014 1	ブシロード	オリジナル	アニメ	下枠	全体	全体	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
DQ	2014 3	スクエニ	あり	-	文字枠上	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
ウィークロス	2014 4	タカラトミー	オリジナル	アニメ	上	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
Lnオ	2014 4	タカラトミー	あり	-	文字枠上	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
妖怪(新)	2014 5	バンダイ	あり	-	上部	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
パズドラ	2015 1	角川	あり	-	下枠	中心	中心	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり
ドレット	2015 5	角川/ブルーバース	オリジナル	-	左枠	左端	左端	名前下左	右上	種族回枠	下	四角	左下	右下	種族と同枠		左上	あり	あり	あり

図 2.1 TCG の各要素に関してまとめたデータ

2.3 データ解析

図 2.2、2.3 のように、既存の TCG には、類似したものとそうでないものがある。このように、TCG の中には異なる TCG でも同じような配置を持つものや、まったく異なる配置を持つものがあり、この配置の位置によって TCG を分類できると考えられる。さらに、既存の TCG のレイアウトには、デザイナーの経験やテクニックが凝縮されていると考えられる。したがって、TCG データの解析により、デザイナーの経験やテクニックを抽出でき、これを利用することで、TCG を初心者が手軽に制作するためのテンプレートを提示できると考えた。データの解析には、オープンソースの統計解析ソフト R[13, 14] を用いた。

2.3.1 データの数値化

図 2.1 のデータは TCG により、データのない要素が含まれているため、最終的に解析に用いたデータは、27 銘柄、7 要素であった。これらを実数化したものを図 2.4 に示す。図 2.4 は、27 銘柄、7 要素（カードの名前、イラスト、召喚コスト、種族、カードの種類、効果について書かれたテキスト、攻撃力の数値）について、数値化を行ったものである。各要素を実数化するために、図 2.5 に示すように 1 枚のカードを 15 分割し、番号付けをした。本研究では、各要素の中心がどの位置に配置されているかにより、位置の判定を行った。このように実数化したデータを、R を用いて解析する。

2.3.2 クラスタ分析

クラスタ分析 [15, 16, 17] とは、分類対象集団において、類似したデータをまとめ、クラスタと呼ばれるデータの集合を作成することでデータを分類するデータマイニング手法のひとつである。クラスタ分析には、階層クラスタ分析と非階層クラスタ分析 [18, 19] が存在するが、本研究では、階層クラスタ分析を採用した。



図 2.2 類似した TCG

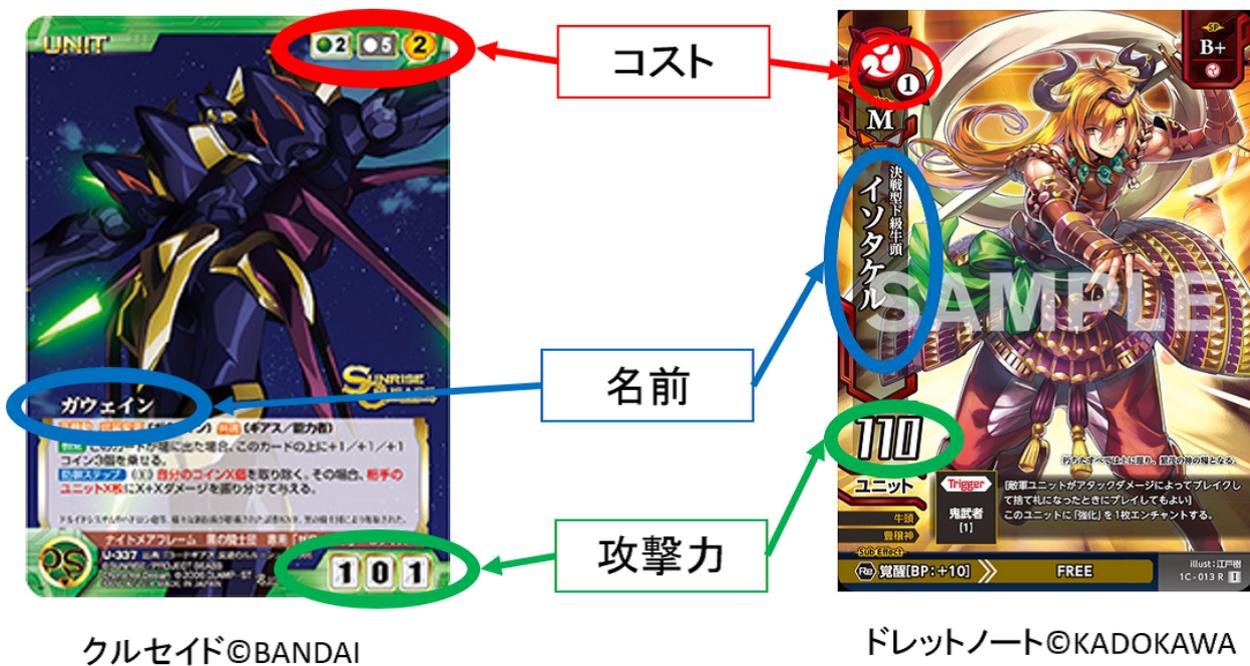


図 2.3 非類似 TCG

階層クラスタ分析は、最も類似したものから分類していく手法である。階層クラスタでは、データ間の距離情報を用い、クラスタを構築する。データ間の距離にはユークリッド距離、マハラノビス距離、マンハッタン距離、チェビシェフ距離、ミンコフスキ距離などが存在するが、本研究では最も基本的なユークリッド距離を用いた。TCG i と TCG j の間のユークリッド距離は、次式により定義される。

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_i^k - x_j^k)^2 + (y_i^k - y_j^k)^2} \quad (2.1)$$

ここで、 x_i^k 、 y_i^k は以下のように求める。図 2.5 のように分割し、数値化したデータは、1次元データであるため、距離を正確に計算するには、これを2次元にする必要がある。TCG i の k 番目のデータを z_i^k としたとき、分割を縦横 $N \times M$ の場合、 x_i^k の値は次式のように求めた。

$$x_i^k = z_i^k \bmod M \quad (2.2)$$

このとき 0 を M に置き換える。また y_i^k の値は次式のように求めた。

$$y_i^k = \lceil z_i^k / M \rceil \quad (2.3)$$

R では、`dist` 関数を用いることにより、全 TCG ペア間の距離情報を含む距離行列を求めることができる。図 2.4 のデータに対して、`dist` 関数（付録 3 のコードを参照）を適用した結果を図 2.6 に示す。

title	name	picture	Summon	ctribe	type	text	Power
MTG	2	5	3	8	8	11	15
pokemon	2	5	1	2	3	11	11
moncole	4	5	1	10	3	14	2
yuugiou	2	8	3	14	3	14	15
GUNDAM	11	8	3	14	1	11	15
DM	2	5	1	2	10	11	13
dimezero	2	8	1	10	1	11	13
Crusade	10	8	3	14	1	11	15
WS	14	8	1	15	13	14	13
Battle Spirit	8	5	1	9	7	11	10
Chaos	14	8	11	1	1	11	13
prememo	11	8	1	13	1	14	12
NEX-A	10	8	3	14	1	11	15
VG	14	8	1	15	13	11	13
pricone	11	8	1	13	10	14	12
sangoku	10	8	1	1	13	14	13
ZX	11	8	1	14	1	14	13
Ange	14	8	1	13	13	11	13
Unlimited	14	8	1	15	10	11	13
SiegKrone	14	8	1	15	14	11	13
Last Chron	10	8	1	7	7	11	13
BF	14	8	1	13	1	11	7
WIXOSS	2	8	1	12	1	14	13
level neo	11	5	10	1	11	14	12
youkai	2	8	3	1	1	14	10
puzzdra	14	8	1	1	1	11	10
dreadnough	7	8	1	13	13	14	10

図 2.4 図 2.1 のデータを数値化したもの

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

図 2.5 カード分割図

	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
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	3.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	5.57	4.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	3.32	4.90	5.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	4.36	6.00	5.83	3.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	3.74	3.87	6.56	6.24	6.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	4.00	4.12	5.00	3.87	4.36	4.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	4.47	6.08	5.74	3.87	1.00	6.63	4.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	6.00	7.55	7.28	6.71	5.20	6.00	6.16	5.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	3.87	4.24	5.10	5.29	4.47	3.32	3.87	4.58	3.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	6.40	5.83	7.21	7.21	5.66	6.08	5.92	5.74	6.86	5.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	5.10	5.74	4.58	4.36	2.65	6.48	4.00	2.83	5.10	4.36	5.74	2.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	4.47	6.08	5.74	3.87	1.00	6.63	4.47	0.00	5.29	4.58	5.74	2.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	5.92	7.48	7.35	6.78	5.10	5.92	6.08	5.20	1.00	3.74	6.78	5.20	5.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	4.80	6.48	5.48	5.29	4.00	5.74	5.00	4.12	3.32	4.00	6.48	3.00	4.12	3.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	5.48	5.92	7.14	7.42	6.56	3.74	6.00	6.48	4.69	4.12	5.39	6.16	6.48	4.80	4.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	5.29	5.74	5.39	4.58	3.00	6.00	3.46	3.16	4.24	3.61	5.39	2.45	3.16	4.36	3.87	5.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	5.92	7.48	7.07	6.78	5.10	5.92	5.74	5.20	2.24	4.24	6.48	4.80	5.20	2.00	2.83	4.36	4.36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	5.66	7.00	6.86	6.24	4.36	5.83	5.48	4.47	1.41	3.32	6.24	4.47	4.47	1.00	3.32	4.90	3.46	2.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	5.83	7.28	7.14	6.56	5.20	6.00	6.16	5.29	1.41	3.87	6.86	5.29	5.29	1.00	3.61	4.90	4.47	2.24	1.41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	4.58	5.10	5.66	5.66	4.24	4.12	3.87	4.12	3.87	2.83	4.47	3.87	4.12	3.74	3.46	3.00	3.32	3.16	3.32	3.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	6.24	6.32	4.69	5.83	3.74	6.86	4.58	3.87	5.00	4.24	5.48	2.65	3.87	4.90	4.00	6.24	2.65	4.47	4.12	5.00	3.74	0.00	0.00	0.00	0.00	0.00	0.00
23	4.12	4.24	5.29	3.74	4.47	4.58	2.24	4.58	5.74	3.46	6.32	4.36	4.58	5.83	5.29	6.24	3.32	6.16	5.20	5.92	4.47	5.10	0.00	0.00	0.00	0.00	0.00
24	5.48	5.57	6.56	7.14	6.56	5.10	6.63	6.63	6.16	5.00	4.36	6.00	6.63	6.24	5.20	4.24	6.48	5.92	6.16	6.16	4.80	6.56	6.86	0.00	0.00	0.00	0.00
25	4.12	3.46	5.48	5.10	5.66	4.12	3.87	5.74	7.55	4.69	5.29	5.74	5.74	7.62	6.48	5.57	5.57	7.35	7.14	7.68	4.90	6.16	4.24	6.08	0.00	0.00	0.00
26	6.00	4.80	5.92	6.86	5.20	5.29	5.10	5.29	6.16	4.12	3.32	4.69	5.29	6.08	5.57	4.47	4.47	5.74	5.48	6.16	3.32	4.12	5.57	5.10	4.58	0.00	0.00
27	5.10	6.71	5.74	5.92	5.39	5.10	4.90	5.29	3.16	3.87	7.00	4.69	5.29	3.32	2.65	4.24	4.47	2.65	3.46	3.46	3.32	4.80	5.20	5.83	6.40	6.16	0.00

图 2.6 距離行列

階層クラスタ分析には、クラスタ間の距離の定義により、以下のような6種類の方法(図 2.7)が提案されている。

- 最短距離法 一方のクラスタに属するデータともう一方のクラスタに属するデータ全ペアの距離において最小の距離をクラスタ間距離とする手法
- 最長距離法 一方のクラスタに属するデータともう一方のクラスタに属するデータ全ペアの距離において最大の距離をクラスタ間距離とする手法
- 群平均法 一方のクラスタに属するデータともう一方のクラスタに属するデータ全ペアの距離の平均をクラスタ間距離とする手法
- 重心法 各クラスタ内のデータの個数を重みとして、データの重心を求め、その重心間の距離をクラスタの間距離とする手法
- メディアン法 重心法の派生で、各クラスタの重心を求めるとき、重みを等しくし、重心間の距離をクラスタ間距離とする手法
- ウォード法 各クラスタ内のデータ間の距離の分散に対し、一方のクラスタに属するデータともう一方のクラスタに属するデータペアの距離の分散の比が最大となるようにクラスタを作成する手法

図 2.6 の距離行列に、上記各種を適用した結果を図 2.8–2.13 に示す。このようにクラスタリングされたデータは、デンドログラム(樹状図)により可視化することができる。重心法とメディアン法では、クラスタ間の距離の逆転が起こることがある(図 2.11、2.12)。

階層クラスタ分析では、デンドログラムにおけるしきい値により、クラスタ数が変化する。例えば、図 2.14 の青線の位置にしきい値を設けた場合、クラスタ数は6となる。一方、赤線の位置にしきい値を設けた場合、クラスタ数は24となる。このため、最適なクラスタ分類のためには、最適なクラスタ数を同定する必要がある。この最適クラスタ数の同定には、様々な手法が提

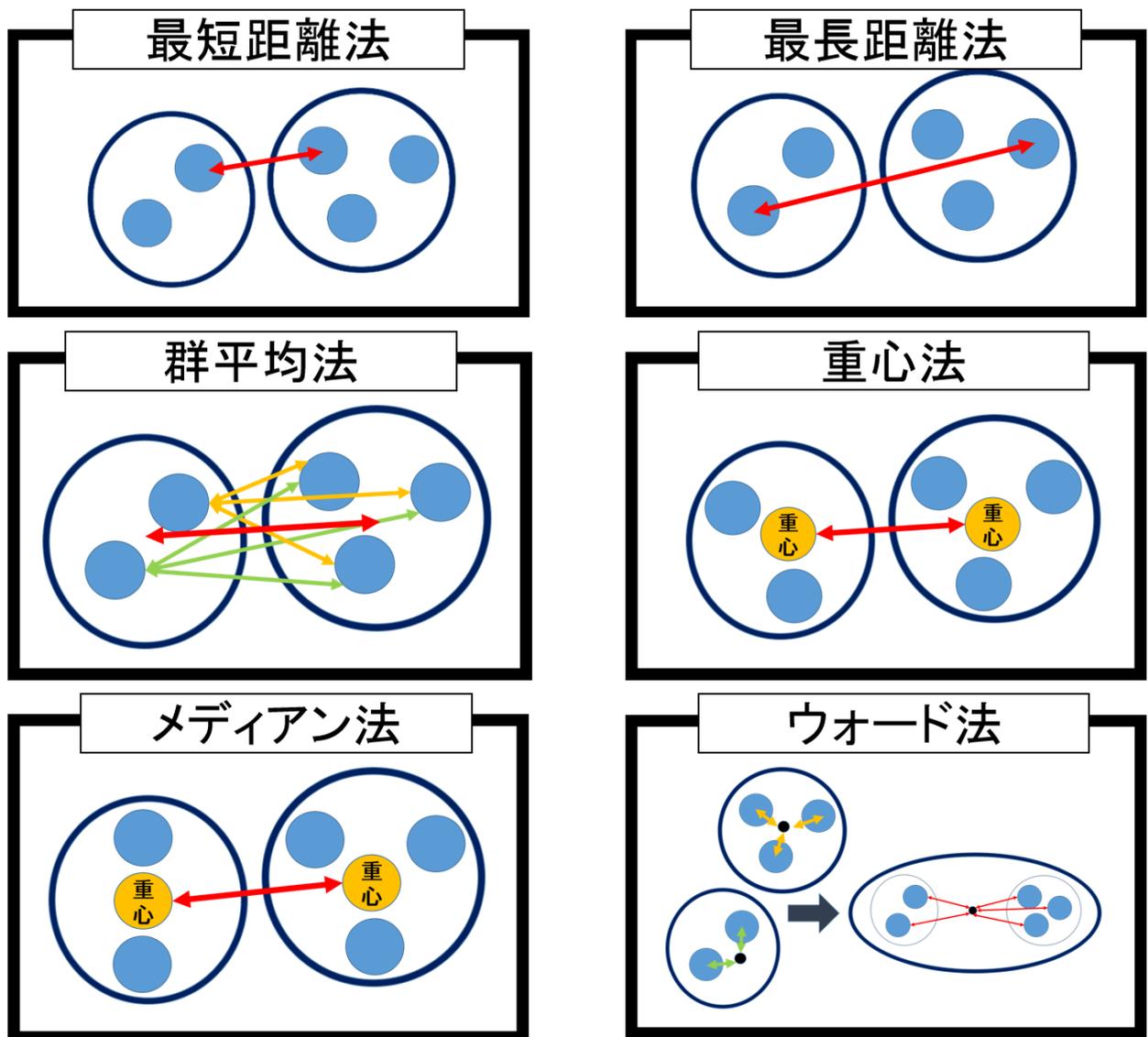


図 2.7 クラスタ間の距離の定義

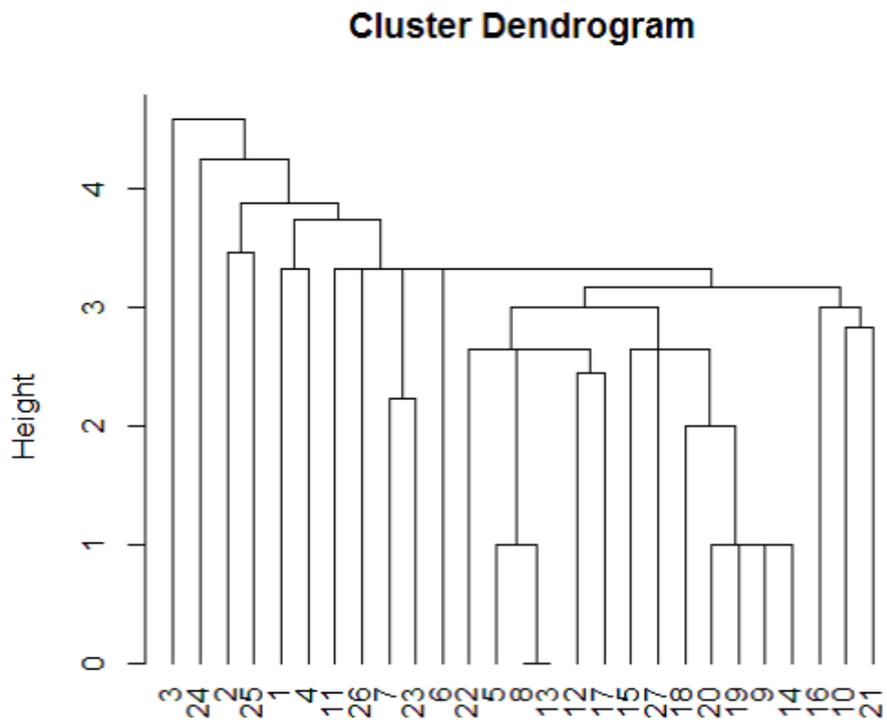


図 2.8 図 2.6 の距離行列に対し最近距離法を適用した結果

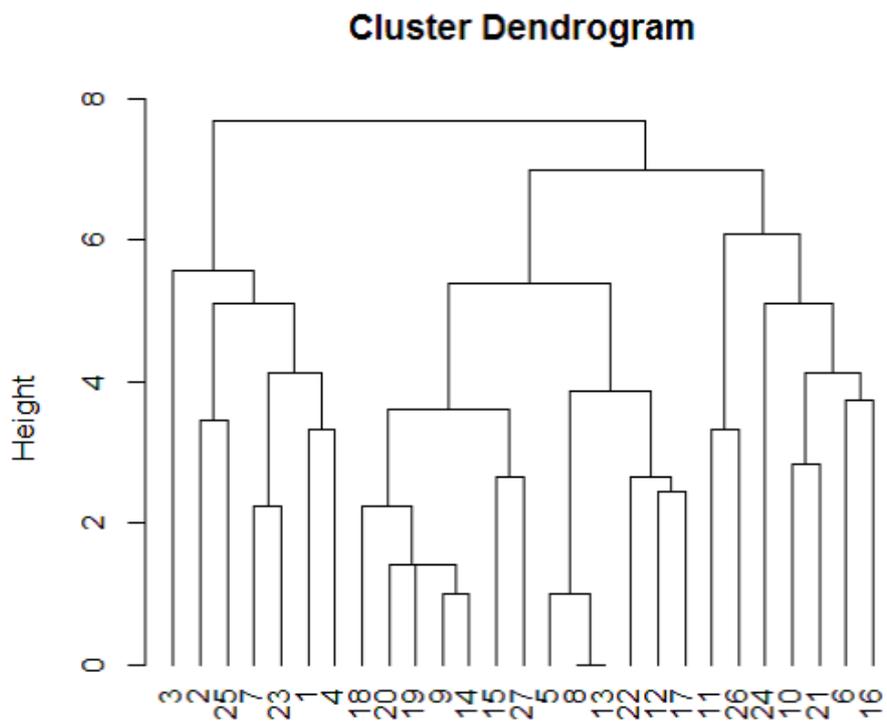


図 2.9 図 2.6 の距離行列に対し最長距離法を適用した結果

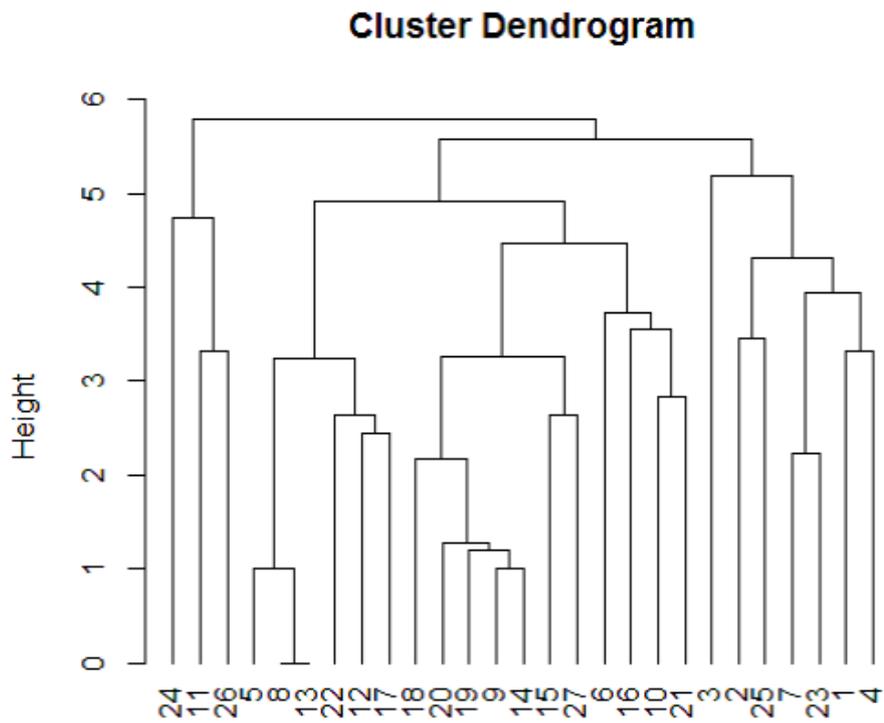


図 2.10 図 2.6 の距離行列に対し群平均法を適用した結果

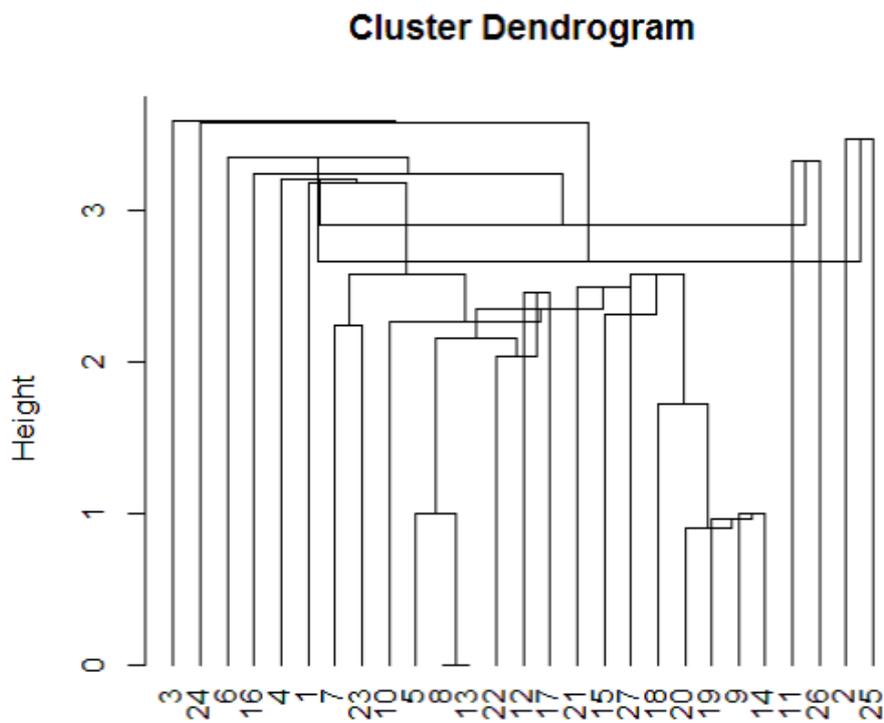


図 2.11 図 2.6 の距離行列に対し重心法を適用した結果

Cluster Dendrogram

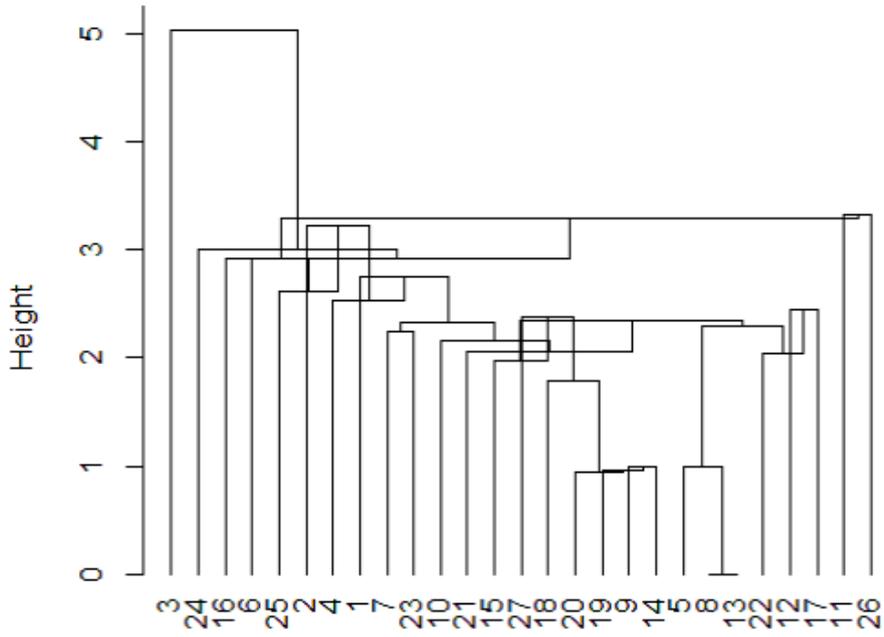


図 2.12 図 2.6 の距離行列に対しメディアン法を適用した結果

Cluster Dendrogram

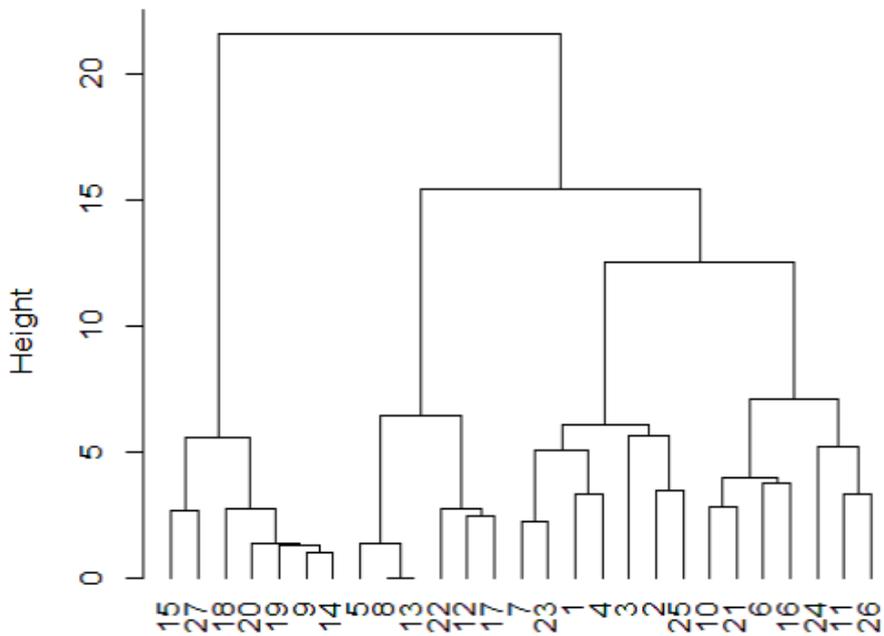


図 2.13 図 2.6 の距離行列に対しワード法を適用した結果

案されている [20, 21, 22, 23, 24]。その 1 つの手法は、コントラスト C である。コントラスト C [22, 23, 24] では、 k 個のクラスタに分類されている場合のコントラスト $C(k)$ は、以下のよう
に定義される。

$$C(k) = \frac{D_{\text{in}}(k) - D_{\text{out}}(k)}{D_{\text{in}}(k) + D_{\text{out}}(k)} \quad (2.4)$$

ここで、 $D_{\text{in}}(k)$ はクラスタ内の全 TCG ペアの類似度の平均値を表して、 $D_{\text{out}}(k)$ は、異なるク
ラスタに属する全 TCG ペアの類似度の平均値を表している。これらの値は、式 (2.5)、(2.6) のよ
うに記述される。

$$D_{\text{in}} = \frac{1}{k} \sum_{l=1}^k \frac{2}{N_l(N_l - 1)} \sum_{i \in \mathbf{C}_l} \sum_{(j \neq i) \in \mathbf{C}_l} s_{ij} \quad (2.5)$$

$$D_{\text{out}} = \frac{2}{k(k-1)} \sum_{l=1}^{k-1} \sum_{m=l+1}^k \frac{1}{N_l N_m} \sum_{i \in \mathbf{C}_l} \sum_{j \in \mathbf{C}_m} s_{ij} \quad (2.6)$$

ただし、 N_l は l 番目のクラスタ \mathbf{C}_l に含まれる TCG の数を表している。また、 s_{ij} は TCG i と
TCG j の類似度であり、以下の式で与えられる。

$$s_{ij} = \frac{1}{1 + d_{ij}} \quad (2.7)$$

さらに、求めた $C(k)$ の最大値 C_{max} を以下のように定義する。

$$C_{\text{max}} = \max_{1 \leq k \leq N} C(k) \quad (2.8)$$

ただし、 N は、全 TCG 数とする。これを用いて最適なクラスタ数 k_{opt} は、以下のように書ける。

$$k_{\text{opt}} = \sum_{i=1}^N k \delta(C_{\text{max}} - C(k)) \quad (2.9)$$

ここで、 $\delta(\cdot)$ は、ディラクのデルタ関数である。

次に Jain-Dubes 法 [25] では、 k 個のクラスタに分類されている場合の Jain-Dubes 法は、以下

のように定義される。

$$p(k) = \frac{1}{k} \sum_{i=1}^k \max_{1 \leq j \leq k} \left\{ \frac{\eta_i + \eta_j}{\xi_{ij}} \right\} \quad (2.10)$$

ここで、

$$\eta_j = \frac{1}{n_j} \sum_{i=1}^{n_j} D(x_j^{(i)}, m_j) \quad (2.11)$$

$$\xi_{ij} = D(m_i, m_j) \quad (2.12)$$

である。 m_j はクラスタ j の重心で、 ξ_{ij} はクラスタ i とクラスタ j の重心の距離となる。 $x_j^{(i)}$ はクラスタ j 内の i 番目の TCG で、 n_j はクラスタ j 内の TCG の数を表している。 η_j は、クラスタ j に含まれる各 TCG と重心との平均距離であり、これはクラスタ j の半径を表している。 $p(k)$ の値がある範囲内で最小になるときが最適なクラスタ数となる。しかし、この定義のままでは、 $p(k)$ が 0 になる、すなわち、全ての TCG がそれぞれ単独のクラスタになる場合、 k の範囲を限定しない場合、最適な結果となる。そのため、Jain-Dubes 法では、スタージェスの公式を用いて、 $2 \leq k \leq 1 + \log_2 N$ とする。本研究では、Jain-Dubes 法を用いて、最適なしきい値を求めた。

2.4 テンプレートレイアウトの決定法

テンプレートレイアウトを求めるため、TCG をクラスタリングし、最適なクラスタ数を求めた。各クラスタごとに、TCG の要素位置の平均値と中央値を求めることで、レイアウトのテンプレートを作成した。

平均値は、以下のように記述される。

$$a_j(i) = \frac{1}{N_i} \sum_{k \in C_i} b_{jk}(i) \quad (2.13)$$

ただし、 $b_{jk}(i)$ は、クラスタ C_i に属する TCG k の j 番目の要素である。

中央値は、 $b_{j1}(i), b_{j2}(i), \dots, b_{jN_i}(i)$ を実数とし、小さい順に並べ替えたものを $b'_{j1}(i), b'_{j2}(i), \dots, b'_{jN_i}(i)$

とするとき、以下のように記述される。

$$a_j(i) = \begin{cases} b'_{j, \frac{N_i+1}{2}}(i) & (\text{if } N_i \text{ is odd.}) \\ \frac{1}{2}(b'_{j, \frac{N_i}{2}}(i) + b'_{j, \frac{N_i+2}{2}}(i)) & (\text{if } N_i \text{ is even.}) \end{cases} \quad (2.14)$$

2.3.2 章で説明した、クラスタの同定は R を用いて行い、これらのデータを保存し、最適なクラスタ数の計算は、これらデータから Processing[26] 上で行った。

2.5 分類結果

最適なクラスタ数を求めた結果を、図 2.15–2.18 に示す。また、各手法ごとに、それぞれのクラスタに属する TCG をまとめたものを図 2.19–2.22 に示す。

最短距離法 (図 2.15、2.19) では、最適なクラスタ数を求めた際に、最後まで単独でクラスタになっているものが 2 つあり、その他多くの TCG がひとまとめにクラスタリングされているものもあり、TCG をうまく分類することが出来ていない。この単独でクラスタになっている TCG を見てみると、他の TCG に比べ、独特なレイアウトを持っていることが分かる。

最長距離法 (図 2.16、2.20) では、カードの名前が下にあり、イラストが全体に描かれているものと、カードの名前が上にあり、イラストの描かれる位置が全体ではないものと分かれている。クラスタに属する TCG の数が少ないものを見ると、独特な TCG のレイアウトを持ったものが一つのクラスタに属している。

群平均法 (図 2.17、2.21) では、最長距離法と同じように、カードの名前の位置とイラストのサイズで分けられているが、カードの名前が下にあるものの中で、特殊なものが分けられている。単独でクラスタになるものもあるが、最短距離法と異なり、これは最後まで単独ではなく、途中まで単独になっていた。最長距離法と同じように、クラスタに属する TCG の数が少ないものを見ると、独特な TCG のレイアウトを持ったものが一つのクラスタに属している。

ワード法 (図 2.18、2.22) では、カードの名前が下にあり、イラストが全体的に描かれている

ものを特に同じクラスタに分類する傾向があり、それ以外の多数がひとまとめになっているクラスタもある。クラスタの一つは群平均法でのクラスタと同じ TCG が属しているクラスタもあり、ここに属している TCG は類似した配置を持つ TCG といえる。どの手法でも単独のクラスタになるものや、3 つ以下のクラスタになるものは同じ TCG で、これらは他の TCG と比べて特殊なレイアウトを持っていた。

それぞれの手法で、各クラスタにおける TCG 要素の配置を、それぞれ平均値と中央値を利用して作成した結果を図 2.23–2.30 に示す。クラスタ内の配置を決定するのに、平均値と中央値を比べると、中央値の方が配置に崩れが無いため、有用だといえる。以上の結果から、最長距離法と中央値を用いた場合が筆者の主観的方法での分類、配置に近く、テンプレートレイアウトを作成するのに良好な結果を示したが、実用化のためには、さらなる検討が必要である。

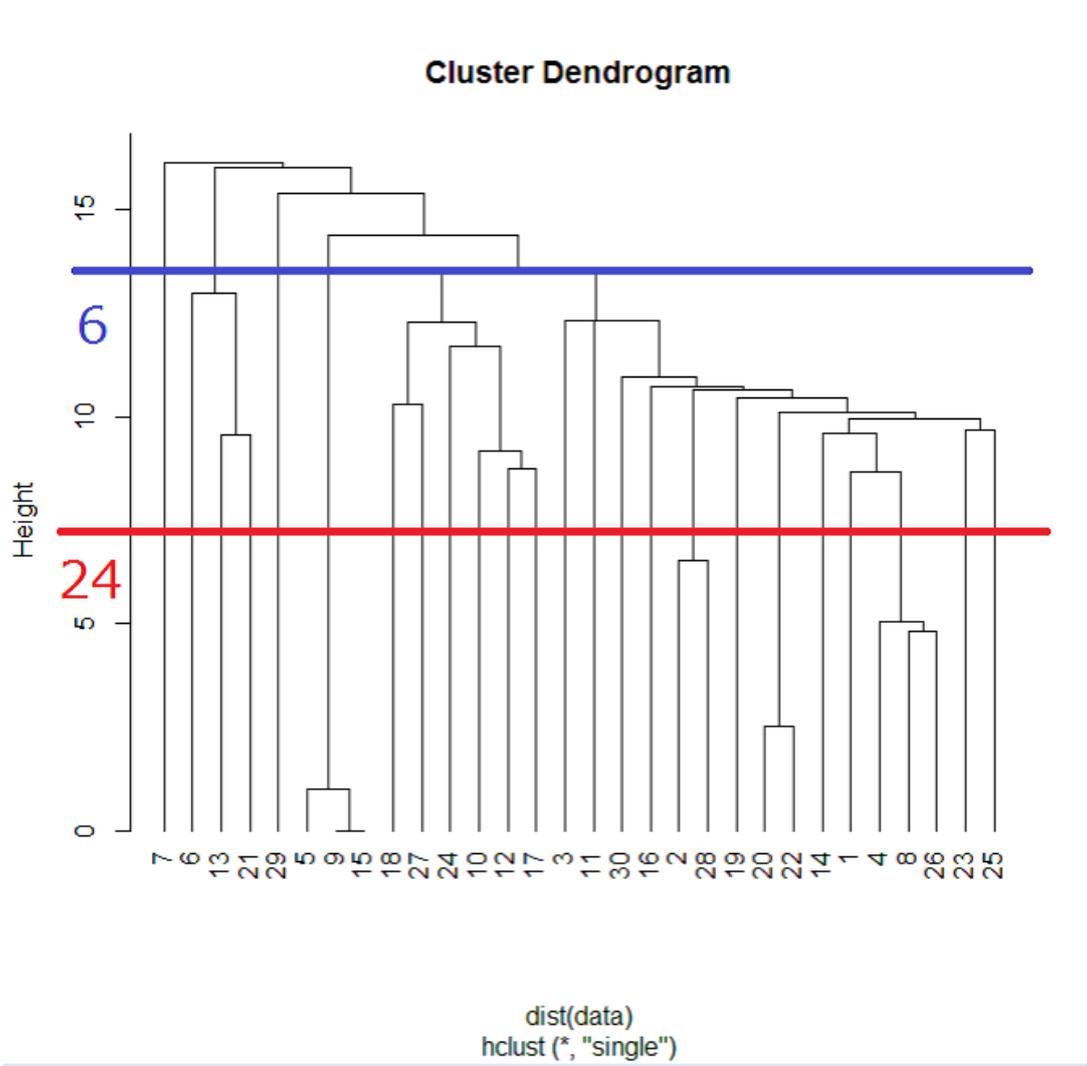


図 2.14 クラスタ分割の例

Cluster Dendrogram

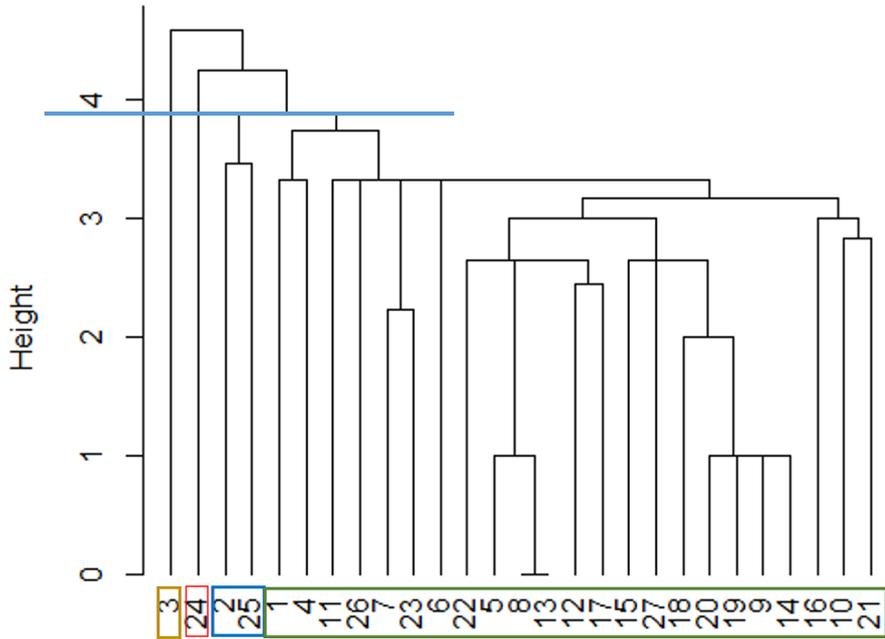


図 2.15 最短距離法での最適なクラスタ数

Cluster Dendrogram

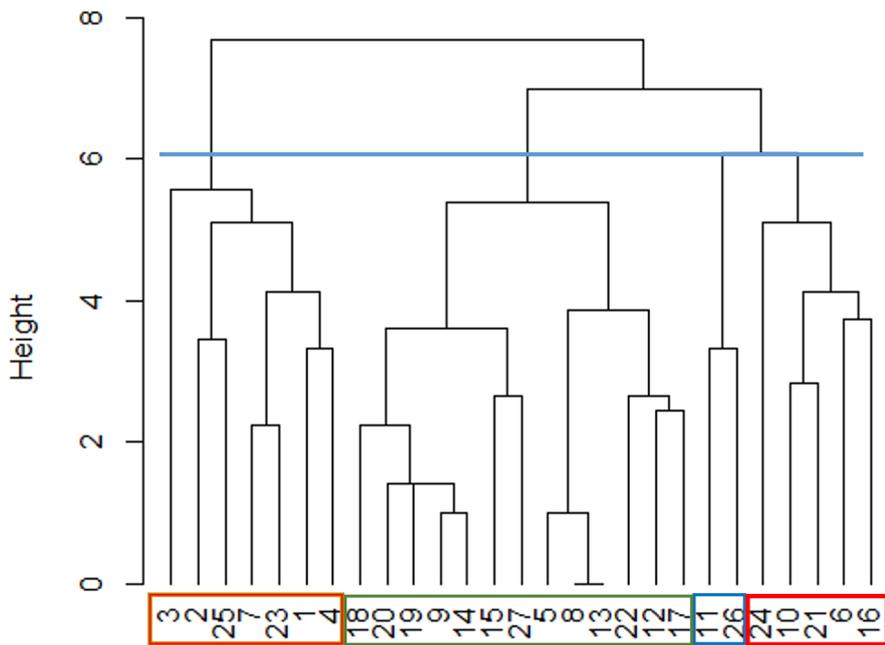


図 2.16 最長距離法での最適なクラスタ数

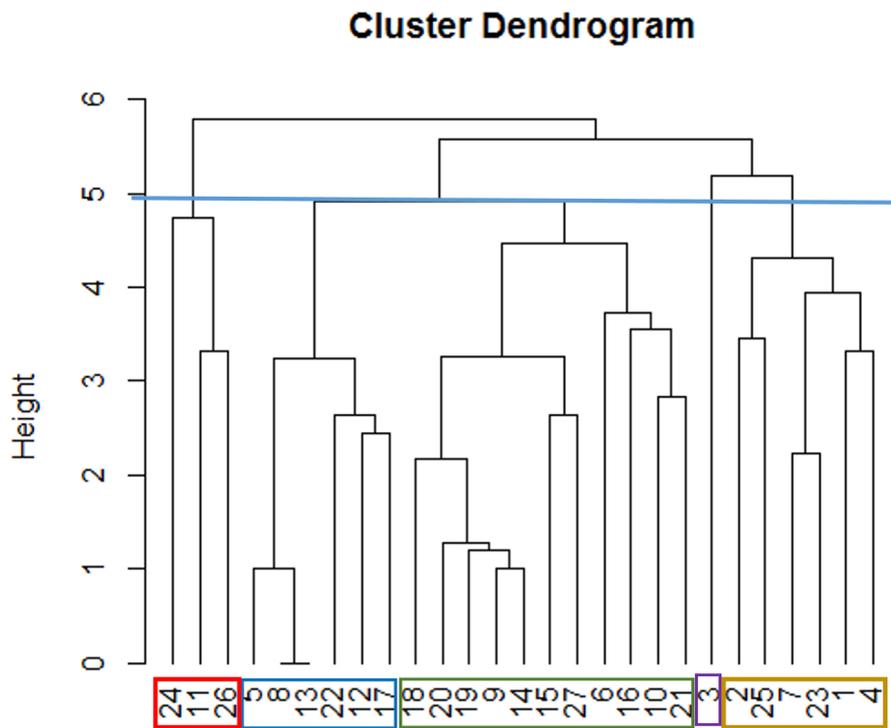


図 2.17 群平均法での最適なクラスタ数

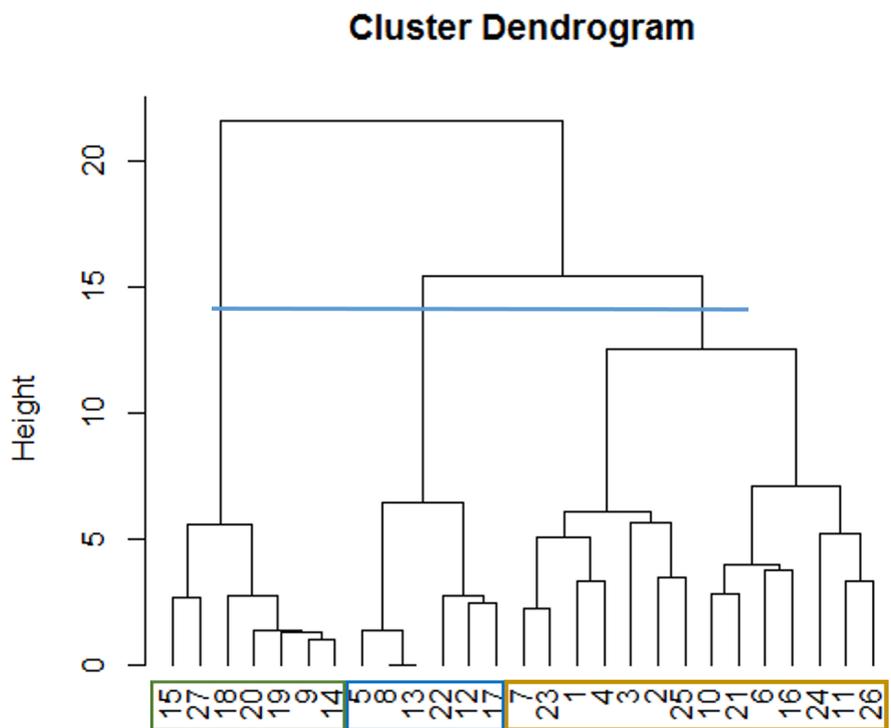


図 2.18 ウォード法での最適なクラスタ数

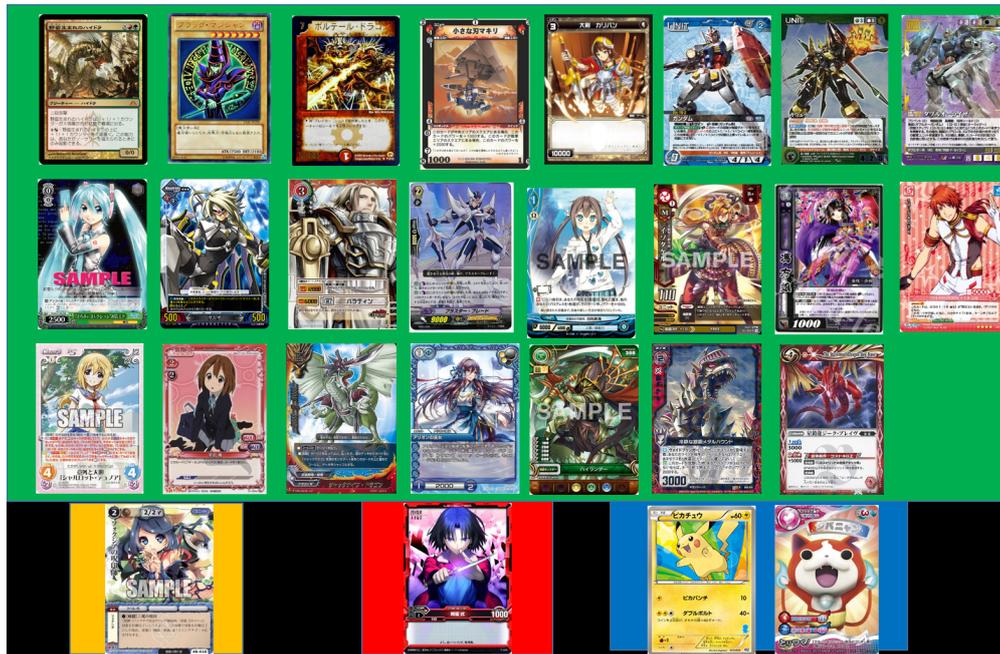


図 2.19 最短距離法のそれぞれのクラスターに属する TCG

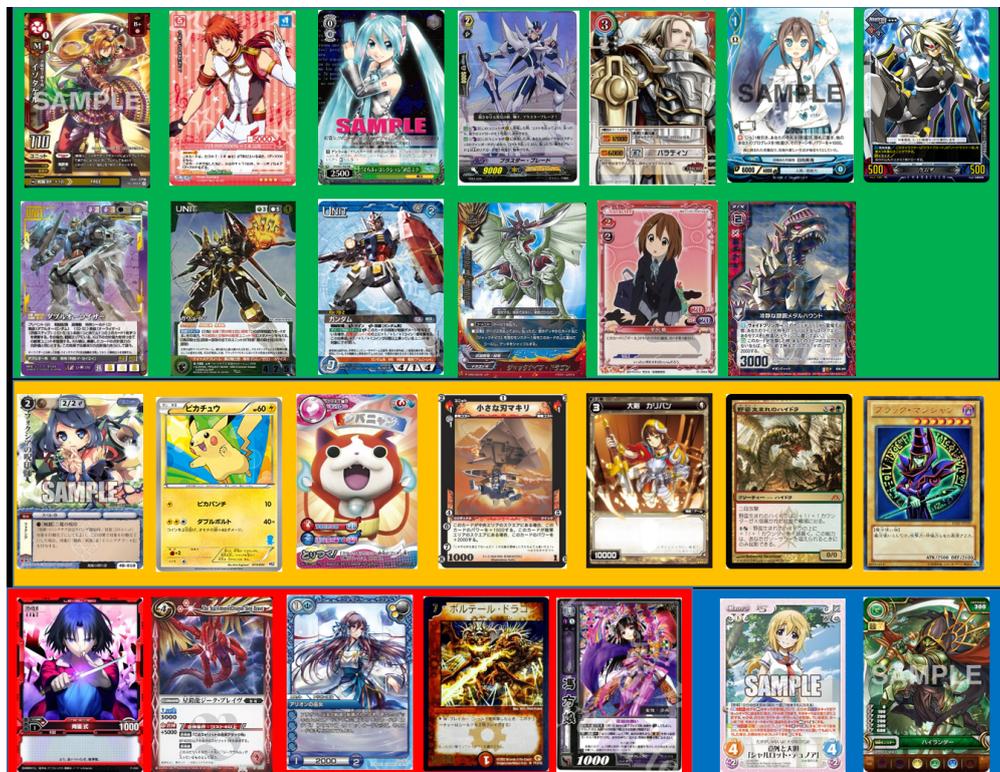


図 2.20 最長距離法のそれぞれのクラスターに属する TCG

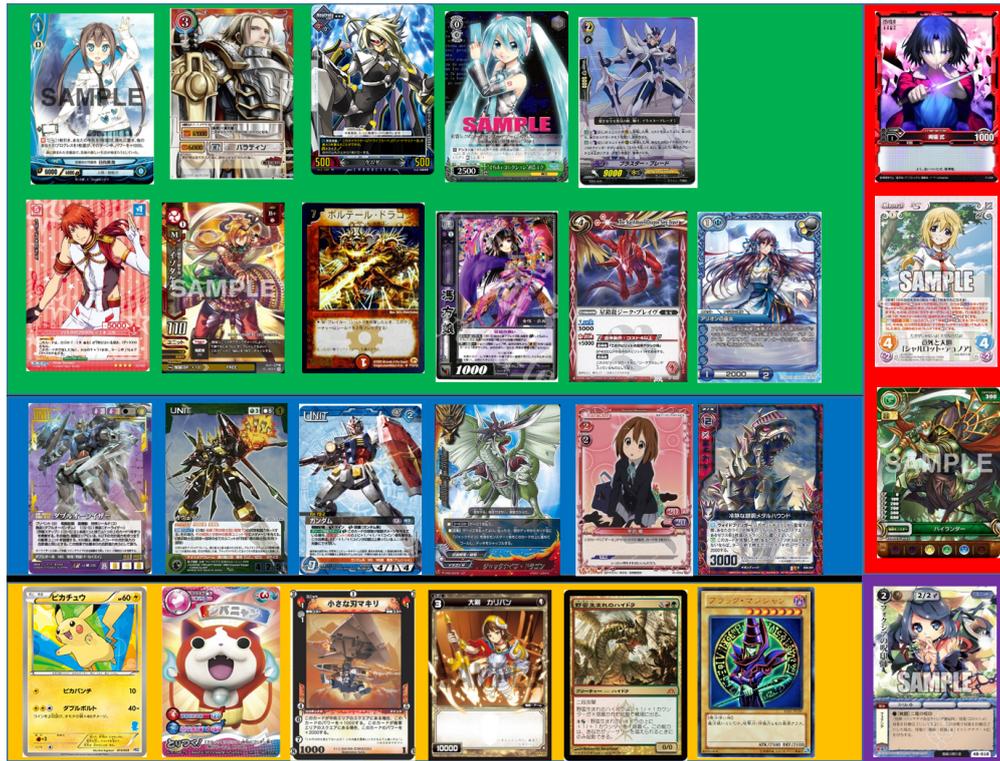


図 2.21 群平均法のそれぞれのクラスに属する TCG

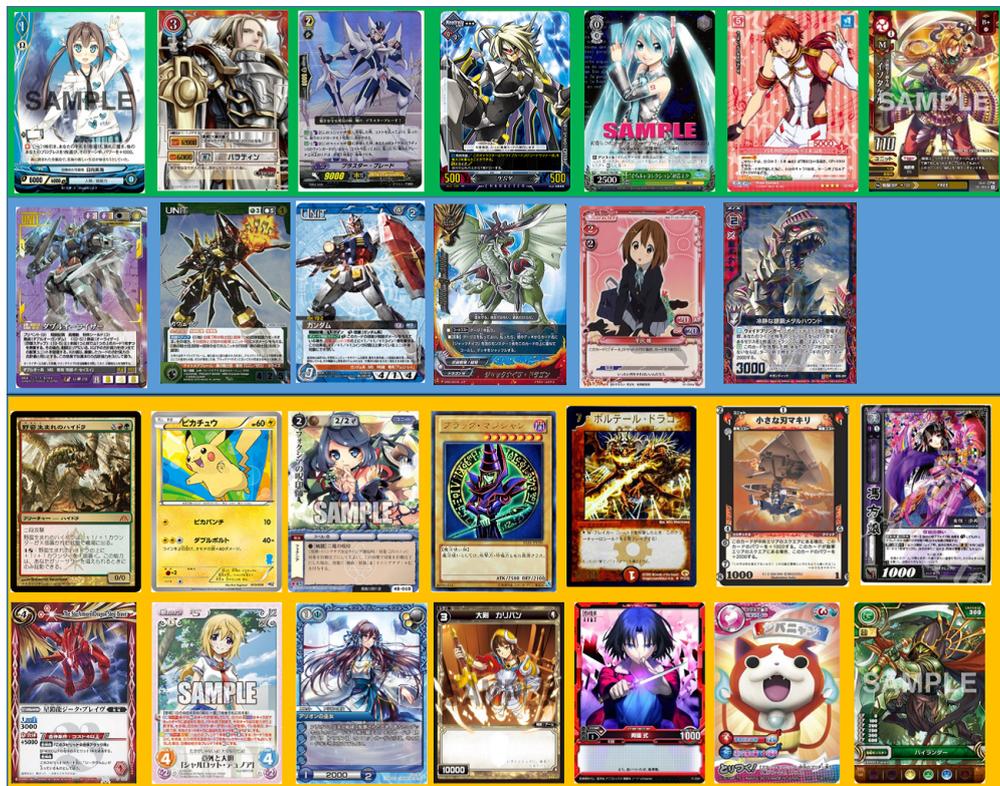


図 2.22 ウォード法のそれぞれのクラスに属する TCG

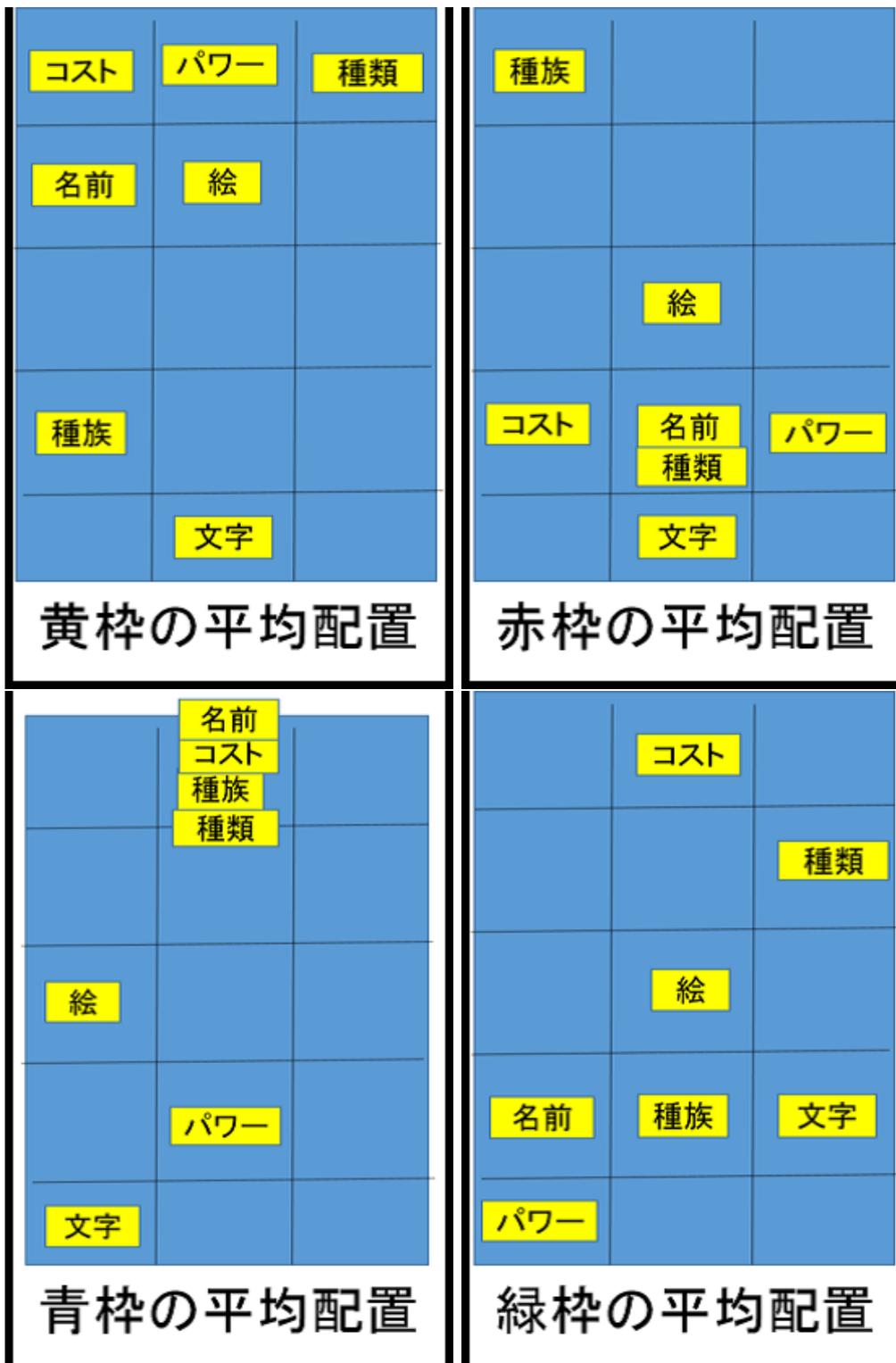


図 2.23 最短距離法の時のクラスタごとの平均値配置

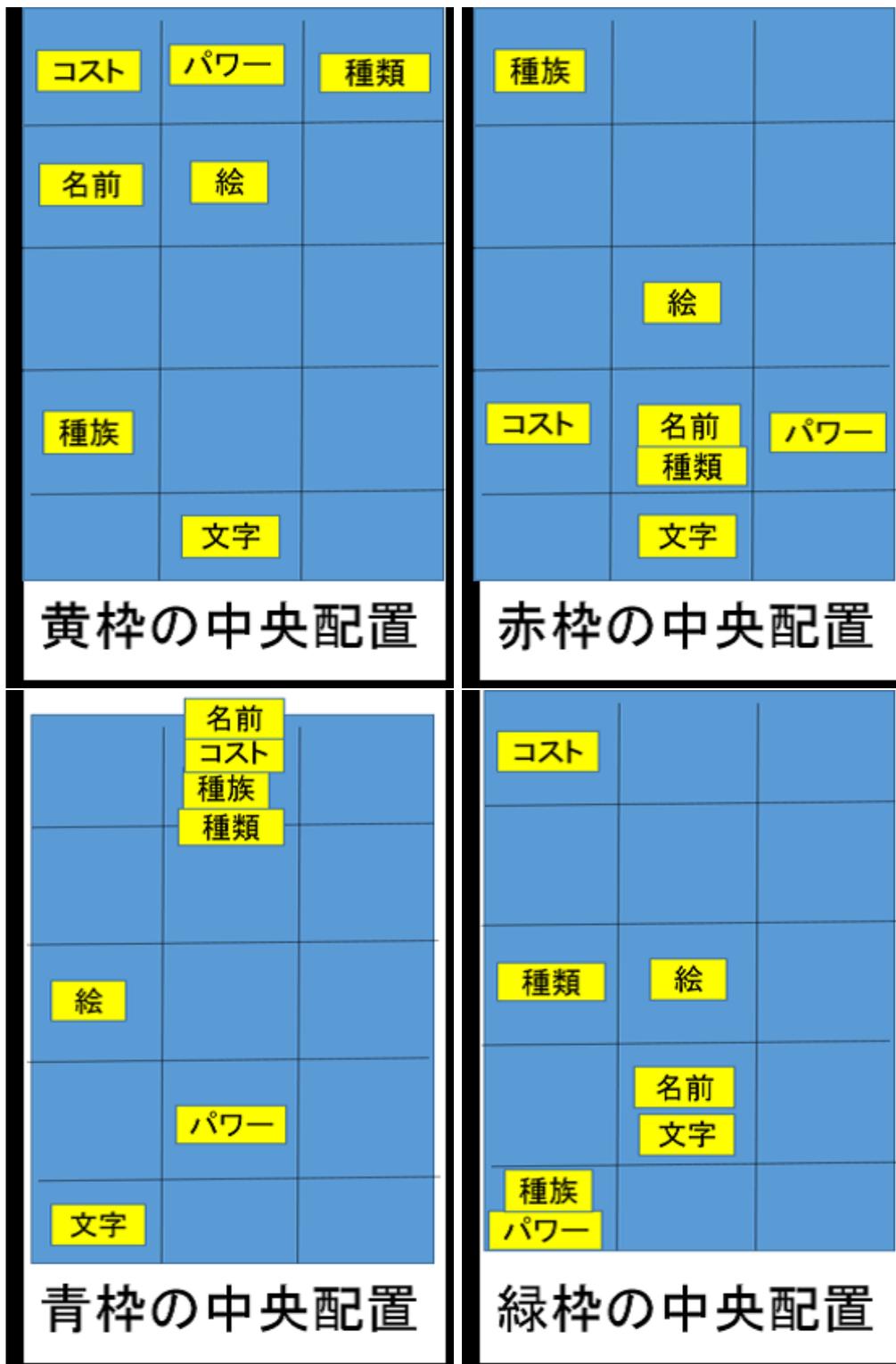


図 2.24 最短距離法の時のクラスごとの中央値配置

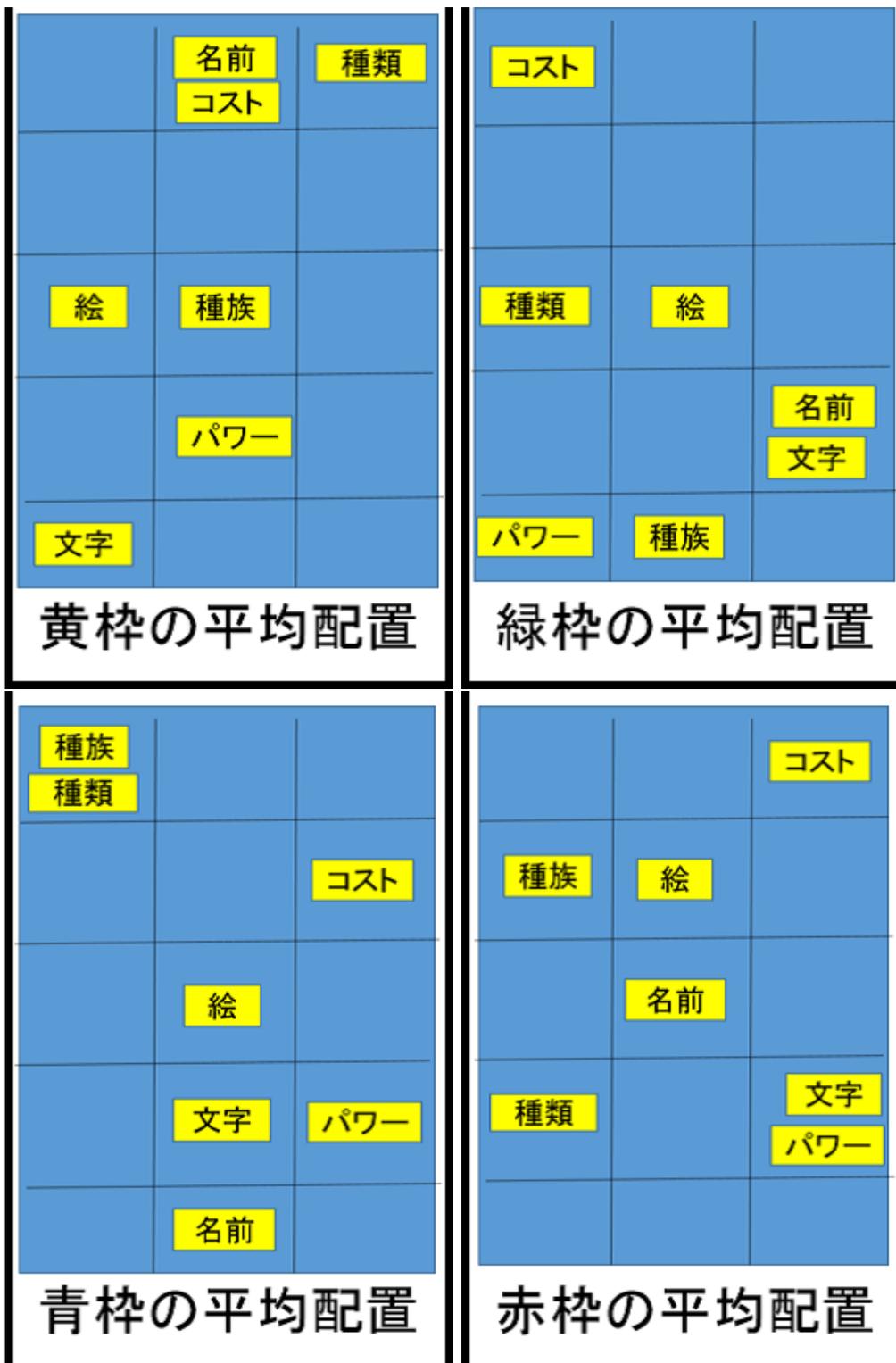


図 2.25 最長距離法の時のクラスタごとの平均値配置

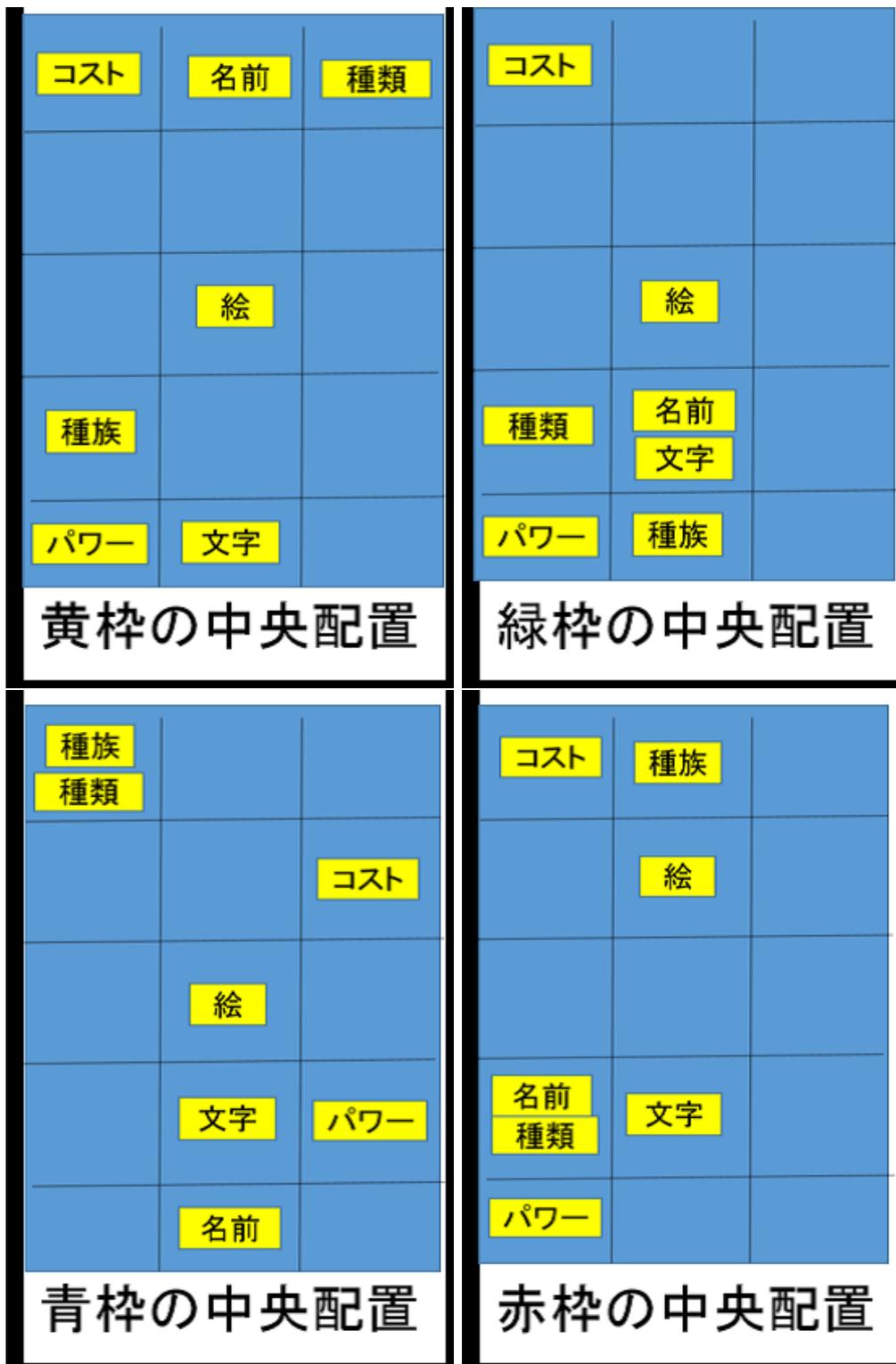


図 2.26 最長距離法の時のクラスタごとの中央値配置

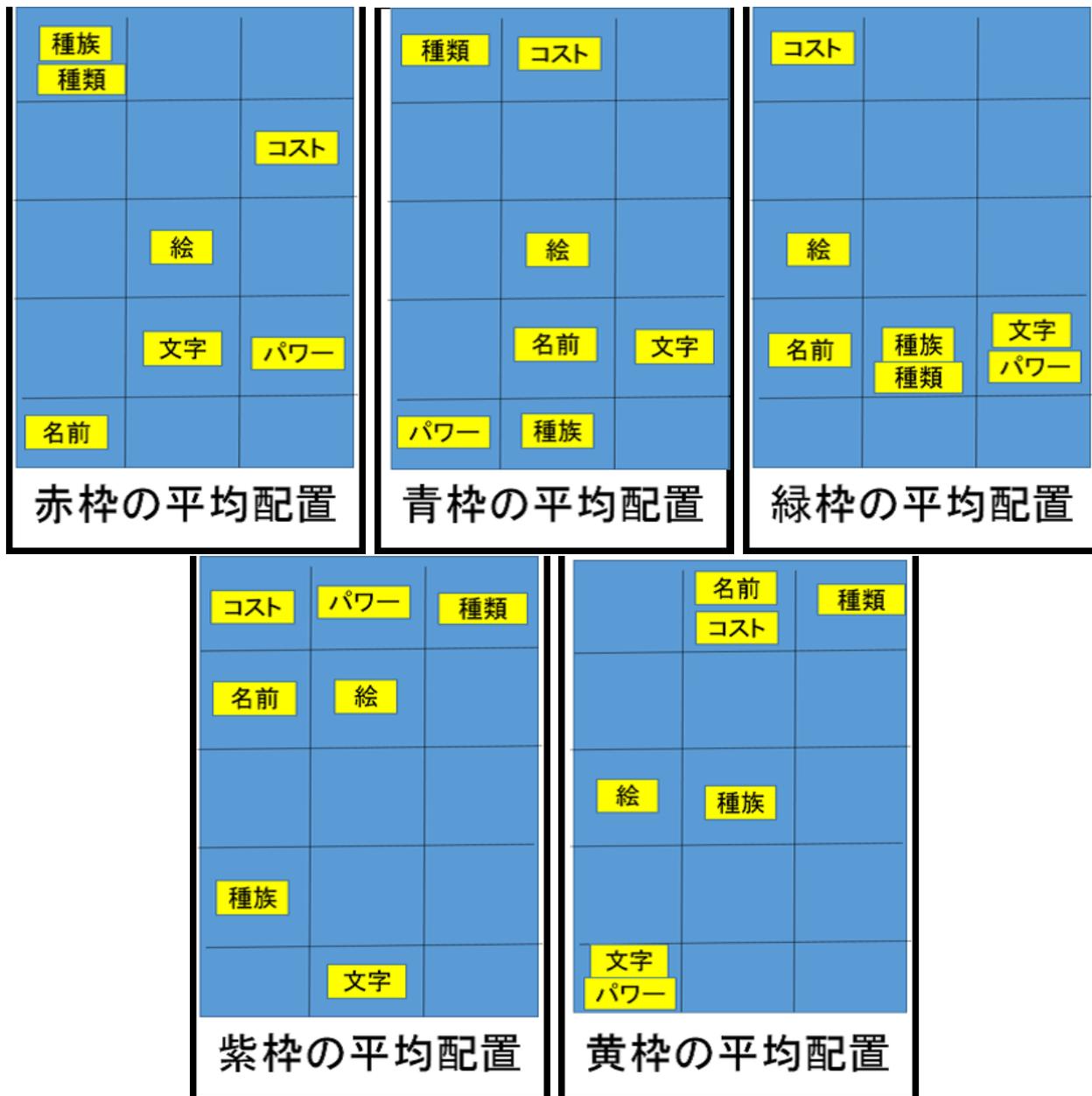


図 2.27 群平均法の時のクラスごとの平均値配置

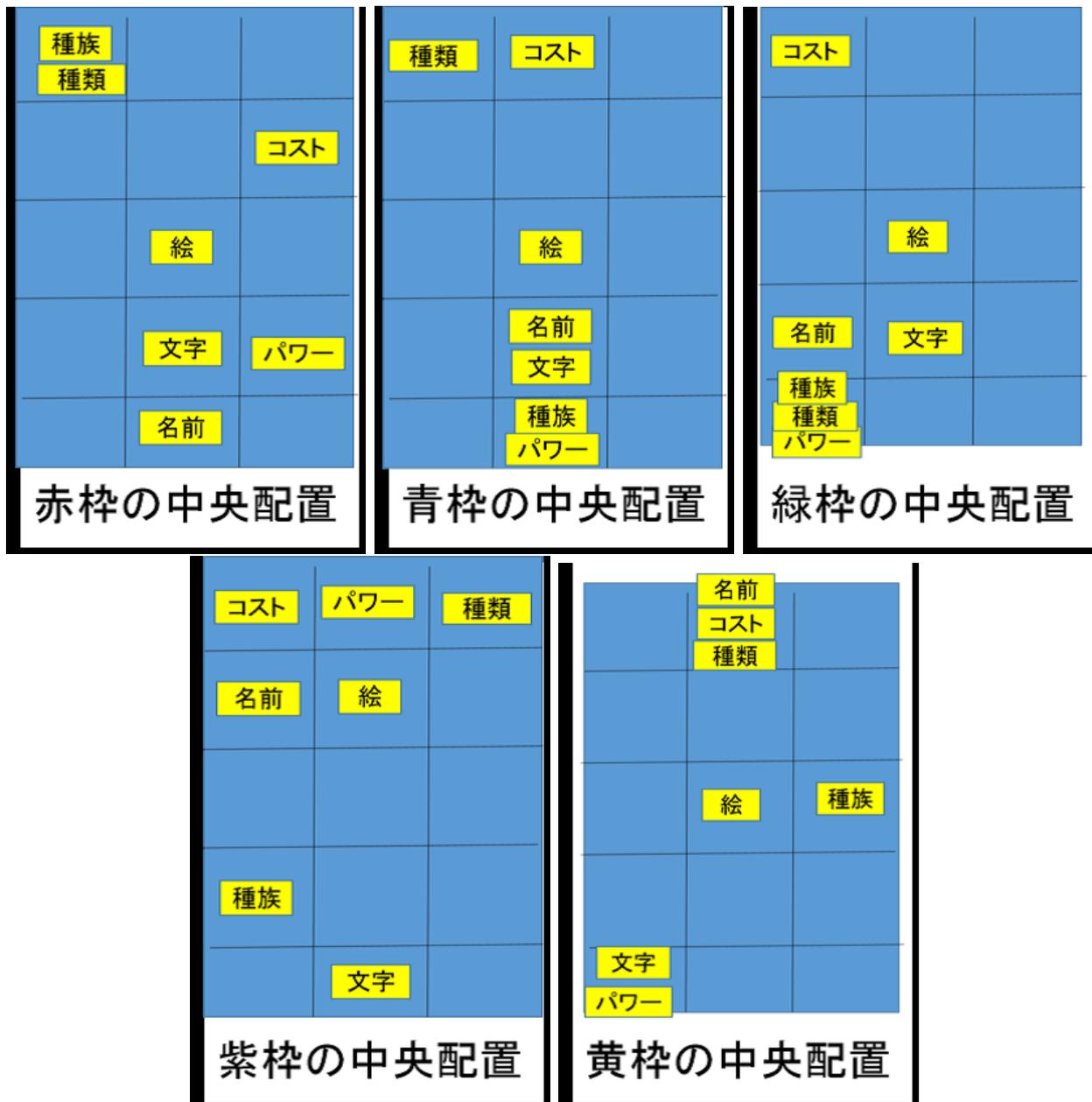


図 2.28 群平均法の時のクラスタごとの中央値配置

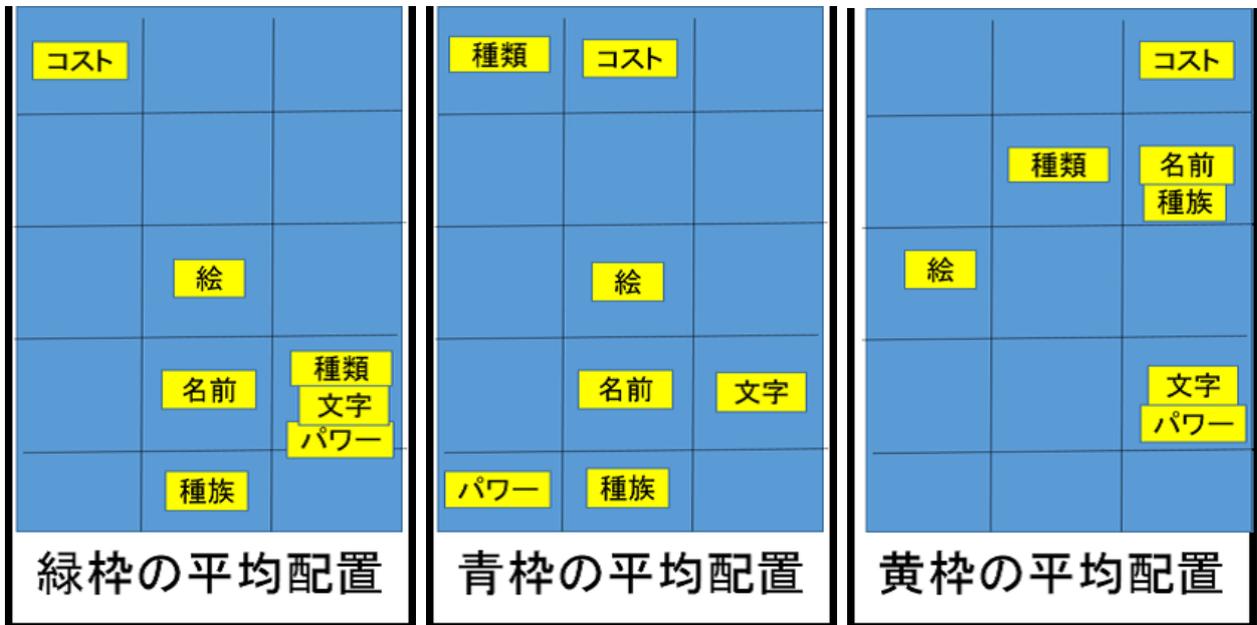


図 2.29 ウォード法の時のクラスラごとの平値配置

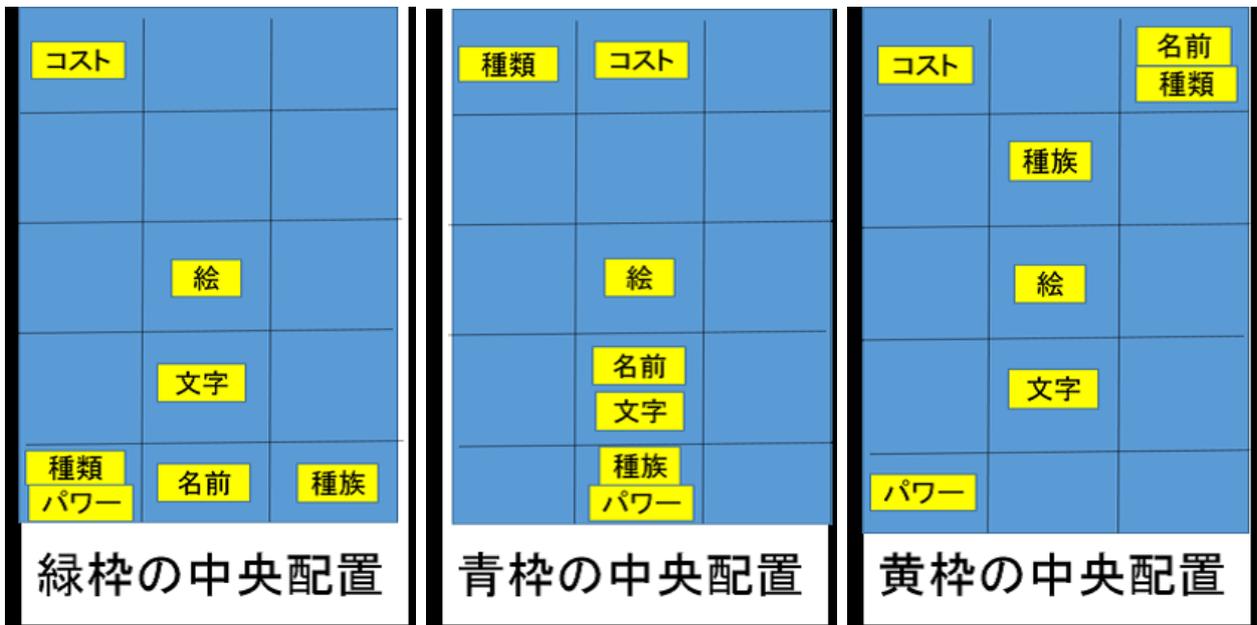


図 2.30 ウォード法の時のクラスラごとの中央値配置

第 3 章

おわりに

3.1 まとめ

本研究では、TCG の作成において、作成未経験者でも、本職のデザイナーが作成したようなデザイン性が高く、ゲームをプレイしやすい TCG 作成を補助することを目的とし、TCG 作成支援ツールを開発するための、TCG のレイアウトの分類を行った。そのために、既存 TCG のデータを収集し、これらデータを階層クラスタ分析を用い解析することで、レイアウトに含まれる要素の位置情報から TCG を分類した。

TCG の分類として、カードの名前とイラストのサイズで分類され、特殊なレイアウトを持つものは、特殊なレイアウトを持つもので分類され、概ね主観的に類似していると考えられるものに分類でき、一定の成果を得ることができた。しかし、分割の仕方や要素の配置の決定法などは、他にも多くの手法が考えられ、まだまだ検討の余地がある。

謝辞

本論文制作にあたり、プログラミングの指導、論文添削をして頂いた加藤秀行 助教、論文のテーマ決めの際に、意見を下さった渡辺大地 講師、三上浩司 准教授に、感謝いたします。またアドバイスをしてくれた研究室の方々にも感謝いたします。

参考文献

- [1] Spring 's Diary. 「2014~2015 年の TCG 市場」. <http://d.hatena.ne.jp/matuda-purehabu/20151201/1448939285>. 参照:2016.1.17.
- [2] おいしいたにし. カードゲームを自作する 7 【印刷データを作る】. <http://tanishi.org/?p=1180>. 参照:2016.1.17.
- [3] 藤井 叙人, 片寄 晴弘 . 戦略型トレーディングカードゲームのための戦略獲得手法. 情報処理学会論文誌, Vol. 50, pp. 2796 – 2806, 2009.
- [4] 中村 杏 . モンテカルロ木探索に基づいたトレーディングカードゲームプレイヤー. 法政大学大学院情報科学研究科, 2014.
- [5] 野瀬 彰大, 深川 大路 . Tcg におけるシャッフル手法に関する計算機実験を用いた考察. 研究報告ゲーム情報学 (GI) , Vol. 4, pp. 1 – 8, 2011.
- [6] 藏富徳彦, 湯浅康史, 白松俊, 大園忠親, 新谷虎松. Web を用いたアンケート立案支援システムの実現とその応用. 第 74 回全国大会講演論文集, Vol. 1, pp. 727 – 728, 2012.
- [7] 神谷享昌, 築地立家 . Arduino を用いた通信対戦用のカードゲームシステムの製作. 第 74 回全国大会講演論文集, Vol. 1, pp. 155 – 156, 2012.

- [8] マジック：ザ・ギャザリング 日本公式ウェブサイト. HOME. <http://mtg-jp.com/#>. 参照:2016.1.7.
- [9] KONAMI. 遊戯王アーク・ファイブ オフィシャルカードゲーム. <http://www.yugioh-card.com/japan/>. 参照:2016.1.7.
- [10] ポケモンカードゲーム公式ホームページ. HOME. <http://www.pokemon-card.com/>. 参照:2016.1.7.
- [11] セガグループ 製品情報サイト. トレーディングカードゲーム. https://sega.jp/arcade/trading_card/. 参照:2016.1.16.
- [12] BATTLENET. ハーフストーン. <http://us.battle.net/hearthstone/ja/>. 参照:2016.1.17.
- [13] 統計解析フリーソフト R の備忘録頁 ver.3.1. HOME. <http://cse.naro.affrc.go.jp/takezawa/r-tips/r.html>. 参照:2015.12.20.
- [14] 日常メモ. 【R によるデータサイエンス】クラスター分析. <http://d.hatena.ne.jp/graySpace/20140430/1398864610>. 参照:2015.12.20.
- [15] 山本義郎, 藤野友和, 久保田貴文. R によるデータマイニング入門. オーム社, 日本, 2015.
- [16] 兼子 毅. R で学ぶ多変量解析. 日科技連出版社, 日本, 2011.
- [17] MACROMILL. クラスター分析とは. <http://www.macromill.com/landing/words/b003.html>. 参照:2015.12.20.
- [18] Albert Analytical technology. データマイニング 8. クラスター分析の手法 (階層クラスター分析). http://www.albert2005.co.jp/technology/mining/method3_2.html. 参照:2015.12.20.
- [19] Toshihiro Kamishima. クラスタリング (クラスター分析). <http://www.kamishima.net/jp/clustering/>. 参照:2015.12.20.

- [20] Pan Zhang, Cristopher Moore, and M. E. J. Newman. Community detection in networks with unequal groups. *Physical Review E*, Vol. 93,iss,1, pp. 727 – 728, 2016.
- [21] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, Vol. 99,no,12, pp. 7821 – 7826, 2002.
- [22] Yuriy Lyakh and Vitaliy Gurianov and Oleg Gorshkov and Yuriy Vihovanets. Estimating the number of data clusters via the contrast statistic. *Biomedical Science and Engineering*, Vol. 5, pp. 95 – 99, 2012.
- [23] John M. Beggs and Dietmar Plenz. Web を用いたの実現とその応用. Neuronal Avalanches Are Diverse and Precise Activity Patterns That Are Stable for Many Hours in Cortical Slice Cultures, Vol. 24(22), pp. 216 – 229, 2004.
- [24] Shuhei Oono, Hideyuki Kato, Tohru Ikeguchi. Shuhei oono and hideyuki kato and tohru ikeguchi. In *Proceedings of the Nonlinear Theory and its Applications (NOLTA2011)*, pp. 589 – 592, 2011.
- [25] A. K. Jain and R. C. Dubes. Algorithms for clustering data. *Prentice-Hall*, 1988.
- [26] Processing - Official Site. Processing. <https://processing.org/>. 参照:2016.1.17.

付録 A

コード集

R のプログラムコード

```
data<-read.csv("card.csv")

datax0<-data %%3
datax0[datax0 == 0]<-3
datax<-datax0

datay0<-data/3
datay<-ceiling(datay0 )

dataxy<-cbind ( datax, datay )

data.d<-dist(dataxy)

data.d<-dist(dataxy)
(data.hc<-hclust(data.d))

summary(data.hc)
data.hc$merge
data.hc$height
data.hc$labels
data.hc$method
data.hc$call
data.hc$dist.method

plot(hclust(dist(dataxy),"single"),hang=-1)
plot(hclust(dist(dataxy),"complete"),hang=-1)
plot(hclust(dist(dataxy),"average"),hang=-1)
plot(hclust(dist(dataxy),"centroid"),hang=-1)
plot(hclust(dist(dataxy),"median"),hang=-1)
```

```

plot(hclust(dist(dataxy),"ward"),hang=-1)
The "ward" method has been renamed to "ward.D"; note new "ward.D2"
plot(hclust(dist(dataxy),"ward.D"),hang=-1)

par(mfrow=c(2,3))
plot(hclust(dist(dataxy),"single"),hang=-1)
plot(hclust(dist(dataxy),"complete"),hang=-1)
plot(hclust(dist(dataxy),"average"),hang=-1)
plot(hclust(dist(dataxy),"centroid"),hang=-1)
plot(hclust(dist(dataxy),"median"),hang=-1)
plot(hclust(dist(dataxy),"ward"),hang=-1)

sei.single<-hclust(dist(dataxy),"single")
for(i in 1:27) {
cutree(sei.single,k=i)
}

sei.complete<-hclust(dist(dataxy),"complete")
for(i in 1:27) {
cutree(sei.complete,k=i)
}

sei.complete<-hclust(dist(dataxy),"average")
for(i in 1:27) {
cutree(sei.average,k=i)
}

sei.complete<-hclust(dist(dataxy),"centroid")
for(i in 1:27) {
cutree(sei.centroid,k=i)
}

sei.complete<-hclust(dist(dataxy),"median")
for(i in 1:27) {
cutree(sei.median,k=i)
}

sei.complete<-hclust(dist(dataxy),"ward")
for(i in 1:27) {
cutree(sei.ward,k=i)
}

```

Processing のプログラムコード

```

int N=27; int M=7; int m = 5;

String filename[] = {"average.txt","centroid.txt","complete.txt","median.txt",
"single.txt","ward.txt"};
String[][] data = new String[filename.length][N][N];

```

```

int [][][] A = new int[filename.length][N][N];

String distname[] = {"dist.txt"};
String[][] dist = new String[N][N];
double[][] s = new double[N][N];

String cardname[] = {"card.txt"};
String[][] card2 = new String[N][M];
int[][] card = new int[N][M];

int K [] = new int[filename.length];

double ck [][][] = new double [filename.length][N][N][M];
int n [][][] = new int [filename.length][N][N][M];

int dk [][][] = new int [filename.length][N][N][M];

double dm [][][] = new double [filename.length][N][N];
double nu [][][] = new double [filename.length][N][N];
double eps [][][] = new double [filename.length][N][N][N];
double S [][][] = new double [filename.length][N][N][N];
double p [][] = new double [filename.length][N];
int k_opt [] = new int [filename.length];

void setup()
{
    for (int i=0; i<filename.length; i++)
    {
        String file[] = loadStrings(filename[i]);
        for (int j=0; j<N; j++)
        {
            String tmp[] = split(file[j], " ");
            for (int k=0; k<N; k++)
            {
                data[i][j][k] = tmp[k];
                int AA = Integer.parseInt(data[i][j][k]);
                A[i][j][k] = AA;
            }
        }
    }

    String d[] = loadStrings("dist.txt");
    for(int i=0; i < N; i++)
    {
        String tmpd[] = split(d[i], " ");
        for(int j=0; j < N; j++)
        {
            dist[i][j] = tmpd[j];
        }
    }
}

```

```

        double DDa = Double.parseDouble(dist[i][j]);
        s[i][j] = 1.0/(DDa + 1.0);
    }
}

String cd[] = loadStrings("card.txt");
for(int i=0; i < N; i++)
{
    String tmp[] = split(cd[i], " ");
    for(int j=0; j < M; j++)
    {
        card2[i][j] = tmp[j];
        int ca = Integer.parseInt(card2[i][j]);
        card[i][j] = ca;
    }
}

for (int a=0; a<filename.length; a++)
{
    for (int b=1; b < N-1; b++)
    {
        int Nin = 0; int Nout = 0;
        double Din =0; double Dout = 0;
        for(int i = 0; i < N; i++)
        {
            for(int j=i+1; j < N; j++)
            {
                if(A[a][b][i] == A[a][b][j])
                {
                    Din += s[j][i];
                    Nin++;
                }
                else
                {
                    Dout += s[j][i];
                    Nout++;
                }
            }
        }
        Din/=Nin;
        Dout/=Nout;
        C[a][b] = (Din - Dout)/(Din + Dout);
    }
}

for (int f=0; f<filename.length; f++)
{
    for (int k=0; k<N; k++)
    {

```

```

    for (int i=0; i<N; i++)
    {
        for (int j=0; j<M; j++)
        {
            ck[f][k][A[f][k][i]-1][j] += card[i][j];
            n[f][k][A[f][k][i]-1][j]++;
        }
    }
}

for (int f=0; f<filename.length; f++)
{
    for (int k=0; k<N; k++)
    {
        for (int i=0; i<N; i++)
        {
            for (int j=0; j<M; j++)
            {
                if (n[f][k][i][j]!=0) ck[f][k][i][j] /= n[f][k][i][j];
            }
        }
    }
}

for (int f=0; f<filename.length; f++)
{
    for (int k=0; k<N; k++)
    {
        for (int i=0; i<N; i++)
        {
            for (int j=0; j<M; j++)
            {
                if ( ck[f][k][i][j]-(int)ck[f][k][i][j]>=0.5)
                    dk[f][k][i][j] = (int)ck[f][k][i][j]+1;
                else
                    dk[f][k][i][j] = (int)ck[f][k][i][j];
            }
        }
    }
}

for (int f=0; f<filename.length; f++)
{
    for (int k=0; k<N; k++)
    {
        for (int i=0; i<N; i++)
        {
            double dd = 0;

```

```

    for (int j=0; j<M; j++)
    {
        dd += (card[i][j] - ck[f][k][A[f][k][i]-1][j])*
            (card[i][j] - ck[f][k][A[f][k][i]-1][j]);
    }
    dm[f][k][i] = Math.sqrt(dd);
}
}
}

for (int f=0; f<filename.length; f++)
{
    for (int k=1; k<m; k++)
    {
        for (int i=0; i<N; i++)
        {
            nu[f][k][A[f][k][i]-1] += dm[f][k][i];
        }
        for (int i=0; i<k+1; i++) nu[f][k][i] /= n[f][k][i][0];
    }
}

for (int f=0; f<filename.length; f++)
{
    for (int k=1; k<m; k++)
    {
        for (int i=0; i<=k; i++)
        {
            for (int j=0; j<=k; j++)
            {
                for (int l=0; l<M; l++)
                {
                    eps[f][k][i][j] += (ck[f][k][i][l] - ck[f][k][j][l])*
                        (ck[f][k][i][l] - ck[f][k][j][l]);
                }
                eps[f][k][i][j] = Math.sqrt(eps[f][k][i][j]);
            }
        }
    }
}

for (int f=0; f<filename.length; f++)
{
    for (int k=1; k<m; k++)
    {
        for (int i=0; i<=k; i++)
        {
            for (int j=0; j<=k; j++)
            {

```

```

        if (i!=j) if (eps[f][k][i][j]>0)
            S[f][k][i][j] = (nu[f][k][i]+nu[f][k][j])/eps[f][k][i][j];
    }
}
}

for (int f=0; f<filename.length; f++)
{
    for (int k=1; k<m; k++)
    {
        for (int i=0; i<=k; i++)
        {
            double Smax = 0;
            for (int j=0; j<=k; j++)
            {
                if (S[f][k][i][j]>Smax) Smax = S[f][k][i][j];
            }
            p[f][k] += Smax;
        }
        p[f][k] /= (k+1);
    }
}

for (int f=0; f<filename.length; f++)
{
    for (int k=1; k<m; k++)
    {
        println("f=" + f + ", k=" + k + ", " + p[f][k]);
    }
    print("\n");
}

for (int f=0; f<filename.length; f++)
{
    double p_min = p[f][1];
    k_opt[f] = 1;
    for (int k=2; k<m; k++)
    {
        if (p[f][k]<p_min)
        {
            p_min = p[f][k];
            k_opt[f] = k;
        }
    }
}

for (int f=0; f<filename.length; f++)
{

```

```
    println(filename[f] + ": " + k_opt[f]);  
  }  
}
```