

平成 25 年度卒業論文

論文題目

ファイルの意味情報と群知能を用いたファイルシステム

神奈川大学 工学部 電子情報フロンティア学科  
学籍番号 201002671  
内田 涼平

指導担当者 木下宏揚 教授

# 目次

<b>第1章</b>	<b>序章</b>	<b>4</b>
1.1	背景	4
1.2	問題点	4
1.3	関連研究	5
<b>第2章</b>	<b>基礎知識</b>	<b>7</b>
2.1	ファイルシステム	7
2.1.1	Network File System(NFS)	7
2.1.2	Virtual File System(VFS)	8
2.1.3	セマンティックファイルシステム (SFS)	8
2.2	群知能	8
2.2.1	Boid	11
2.2.2	ACO(Ant Colony Optimization)	13
2.2.3	PSO(Particle Swarm Optimization)	14
2.3	家族的類似	14
2.4	メタデータ	15
2.5	インデックス	15
2.6	リレーショナルデータベース	15
2.7	クラスタリング	15
2.7.1	データのクラスタリング	15
2.7.2	階層的手法	18
2.7.3	非階層的手法	20
2.7.4	ソフトクラスタリング	20
2.7.5	ハードクラスタリング	21
<b>第3章</b>	<b>提案</b>	<b>22</b>
3.1	目的	22
3.2	ファイルの属性による分類を表現	23
3.3	視認性向上のための群れの安定化	24
3.3.1	群知能によるクラスタリング	24
3.3.2	カテゴリによるファイルの分割	25
3.3.3	中心にあるファイルに対して類似するファイルを集める	26

---

3.3.4 分類されたファイルの見せ方 . . . . .	27
第4章 結論	28
第5章 謝辞	29
第6章 質疑応答	30
第7章 付録	31

## 目 次

1.1	インデックス構造 . . . . .	6
2.1	集団行動を生み出す主な原理 . . . . .	9
2.2	群れの基本的な振る舞い . . . . .	10
2.3	Separation . . . . .	11
2.4	Alingment . . . . .	12
2.5	Cohesion . . . . .	12
2.6	Ant Colony Optimization . . . . .	13
2.7	Particle Swarm Optimization . . . . .	14
2.8	3種類のオブジェクトを含むデータセット . . . . .	16
2.9	類似のオブジェクトを持つ3つのクラスタ . . . . .	17
2.10	最短距離法 . . . . .	18
2.11	最長距離法 . . . . .	19
2.12	重心法 . . . . .	19
2.13	ソフトクラスタリング . . . . .	20
2.14	ハードクラスタリング . . . . .	21
3.1	研究目的 . . . . .	22
3.2	ファイルの属性による分類 . . . . .	23
3.3	クラスタリング . . . . .	23
3.4	カテゴリによるファイルの分割 . . . . .	25
3.5	隣接するカテゴリの設定 . . . . .	25
3.6	中心にあるファイルに対して類似するファイルを集める . . . . .	26
3.7	ファイルの見せ方 . . . . .	27

# 第1章 序章

## 1.1 背景

インターネットが普及し、近年の社会ではクラウド・コンピューティングが注目されている。中でも世の中はデータセンターとサービスとしてのクラウドソフトウェア（SaaS）を組み合わせたビジネスモデルである。我々はクラウドを多様で不確実にインターネット上に存在する情報リソースと人間の「関係性」、そしてその「振る舞い」とみなしている。そこで、情報リソースをPC内のファイルとみなし、そのファイルの振る舞いを群知能によって実現することによって、他者との連携により人間の創発的活動（学術的活動・ビジネス・地域ボランティアなど）を刺激・支援するシステムを研究の目的としている。キュレーションという言葉がある。キュレーションとは、博物館や図書館の学芸員を「キュレーター（curator）」という。「情報を収集し、選別し、意味付けを与えて、それをみんなと共有すること」という定義。情報の洪水の中で、それぞれのユーザーにとって必要なものをコンテキスト（意味の文脈）に沿ってピックアップし、整理し、新しい付加価値を与えることである。ファイルシステム内の大量にあるファイルの種類を自分なりにキュレーションし、「ファイルの見える化」を行い、ユーザーにとって見やすいファイルシステムを作ることがこの研究の目的である。

## 1.2 問題点

ファイルシステムに格納されているデータ量は年々増加している。近年のハードディスク容量は増加しているが、同時に格納されているデータ量も増加している。コンピュータで利用するアプリケーションの増加やコンピュータに接続する新しいデバイスの登場とともに扱うファイルの数は増加している。それに伴ってコンピュータを利用する人がファイルを目的や用途などに応じて分類するのは手間が増加してしまう。ファイル数が少ない場合には、手作業のファイル整理は可能であるが、ファイル数が多量になると手作業で整理するのは困難である。しかし、現状のファイルシステムにはファイルのコピーや移動といった操作しかなく、ファイルを分類する手間を軽減することは難しい。その原因のひとつとして、現状のファイルシステムでは、ディレクトリが木構造であり、その構造が静的であることが挙げられる。このため、ファイルは利用者が定めたひとつの整理ポリシーに則った静的なディレクトリ構造を作って、ファイルの整理をしなければならず、その整理ポリシーに束縛され多

様なファイル整理が不可能である。また、ファイルへのシンボリックリンクもしくはハードリンクを用いて整理ポリシーを柔軟にする方法は存在するが、必要なファイル数だけ利用者がリンクを行う必要があり、整理する手間の軽減になっていない。現状のファイルシステムでは、その実装の中にファイル名、作成年月日、更新年月日、最終アクセス年月日、所有者、パーミッションといった意味情報を持たせるしくみがある。しかしそれらの情報はファイルの並べ替えには使うが、通常ユーザがファイルの識別に利用するのは、そういった情報よりもむしろどのディレクトリに置かれ、どのファイル名であるかというふたつのパラメータである [6]。

### 1.3 関連研究

セマンティックファイルシステムの研究は以前から行われている。以下に外部の研究者、企業が開発したファイルシステムについて紹介する。

- WinFS

WinFS[7]とは、米 Microsoft の次期 Windows である「Longhorn」が搭載する予定の、新しいファイルシステムの開発コード名である。既存のファイルシステムにデータベース管理システム技術を加えることで、大容量ハードディスク装置の中身を縦横無尽に検索できるようにする。リレーショナルデータベースの技術を利用している。従来のファイルシステムは、ファイルの open,close といった基本的な API しか備えなかった。このため、大量のデータを管理・検索する場合は SQL といった特別な言語とデータベース管理システムを利用するしかなかった。これに対して WinFS はデータベース管理システムを内蔵しているため、これにより、ファイルやフォルダだけではなく、個人の連絡先やイベントカレンダー、電子メールなどファイルではないデータも管理できるようになる。さらに、1つのデータを複数のアプリケーションで共有できるのが特徴で、目的に応じてさまざまな角度からデータを検索したり、関連付けしたり、情報を管理することもできる。アプリケーションは、WinFS に用意された XML、SQL、C++、C などのインタフェースを使い、ファイルの内容を加味した検索が可能になる。

- Google Desktop Search

PC 内のファイルから Web サイトまでを一括検索できるツール。ファイル名以外にもファイル内部の文字列やメールの内容から検索することができる [11]。

- Spotlight

検索フィールドに欲しい情報のキーワードを入力するだけで関連する全ての対象を検索できる [12]。

Google Desktop Search と Spotlight はインデックスを利用して高速にキーワード検索を行うのが特徴である。

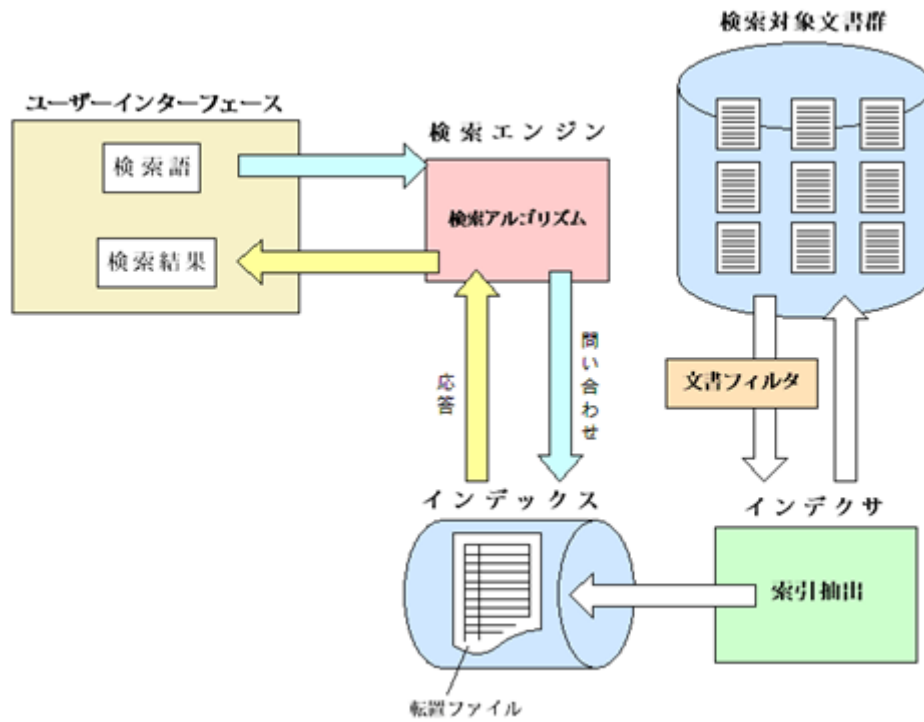


図 1.1: インデックス構造

## 第2章 基礎知識

### 2.1 ファイルシステム

記憶装置に記録されているデータを管理する方式. また, 管理を行うソフトウェアや, 記憶媒体に設けられた管理領域や管理情報のこともファイルシステムと呼ぶこともある. 記憶装置にファイルやフォルダ(ディレクトリ)を作成したり, 移動や削除を行ったりする方法や, データを記録する方式, 管理領域の場所や利用方法などが定められている. 方式によって, ファイル名の文字長や, アクセス権の管理方法などが異なる. 通常, ファイルシステムは OS の持つ機能の一つとして提供され, OS ごとに異なるファイルシステムを用いている. Mac OS では HFS(HFS+) が, Windows では FAT や VFAT, FAT32 などが, Windows NT では NTFS が主に用いられている. また, CD-ROM や DVD などは独自のファイルシステムをもっている. ファイルは画像や音楽, テキスト等のデータやコンピュータプログラムを表現, 管理するための論理的な最小単位になっている. また, 近年のファイルシステムでは, ファイルの一貫性の保持や, ジャーナリングを行っている場合があり, 必要に応じてエラー等の訂正を行っている.

#### 2.1.1 Network File System(NFS)

UNIX システムで利用されるファイル共有システム. Sun Microsystems 社によって開発され, UNIX 系 OS における標準的な分散ファイルシステムとなっている. NFS を使うと離れた場所にあるコンピュータのファイルを, あたかも自分のコンピュータにあるファイルのように操作することができる. ファイルを他のコンピュータに提供するコンピュータを NFS サーバ, 他のコンピュータにあるファイルを利用するコンピュータを NFS クライアントという. UNIX 以外のプラットフォームからも利用することができる.



### 2.1.2 Virtual File System(VFS)

実際のファイルシステムの上位に位置する抽象化層である.VFSの目的はクライアントアプリケーションが様々なファイルシステムに同じ方法でアクセスできるようにすることである.例えばVFSを使うと,クライアントアプリケーションはローカルな記憶装置にもネットワーク上の記憶装置にも透過的にアクセスできるため,その違いを意識する必要がない.あるいは,Windows、Mac OS、UNIXといったオペレーティングシステム(OS)の違いを超えてファイルシステムの違いを意識することなくアクセスすることが可能となる.VFSは,カーネルと実際のファイルシステムとのインタフェースあるいは規約を定義している.従って,その規約に従うことで簡単に新たなファイルシステムをカーネルに追加することができる.規約の条件はリリースの度に非互換な変更を加えられる可能性があり,ファイルシステムは新たなリリースに対応するために修正を施したり,再コンパイルする必要がある.あるいは,OS側で規約変更を下位互換を保つように行っていれば,各ファイルシステムは新たなバージョンのOSでもそのまま使用可能となる.

### 2.1.3 セマンティックファイルシステム(SFS)

SFSでは,ファイルをファイル名とパス情報以外の方法で分類し動的に木構造ディレクトリにマッピングすることができる.これはファイルが持つ意味情報を属性として抽出し,抽出された属性を整理ポリシーに基づいて分類することによって実現されている.このため現在利用されているNTFSやext3といったファイルシステムとは異なりファイルから抽出する属性や整理ポリシーを追加,変更することによってさまざまな分類を提供でき,同時にディレクトリの木構造を動的にマッピングできる.

## 2.2 群知能

群知能(Swarm Intelligence, SI) [2]は,分権化し自己組織化されたシステムの集会的ふるまいの研究に基づいた人工知能技術である.「群知能」という用語は,1989年、BeniおよびWangが提唱したもので,セルラーロボットシステムに関して使ったのが最初である.SIシステムは一般に単純なエージェントやボイドの個体群から構成され,各個体はローカルに互いと,そして彼らの環境と対話する.個々のエージェントがどう行動すべきかを命じている集中的な制御構造は通常存在しないが,そのようなエージェント間の局所相互作用はしばしば全体の行動の創発をもたらす.このようなシステムの自然界の例として,アリの巣,鳥の群れ,動物の群れ,細菌のコロニー,魚の群れなどがある.



図 2.1: 集団行動を生み出す主な原理

- 均質性：群れの中のすべての鳥は同じ行動モデルをもつ。一時的にリーダーが現れたかのように見えたとしても、実際はリーダーなしで行動している。
- 局所性：それぞれの鳥の動きは、それぞれ最も近くにいる鳥にのみ影響を受ける。視覚は、集団の組織化にとって最も重要な感覚能力と考えられる。
- 衝突回避：隣接する仲間との衝突を避ける。
- 速度の一致：隣接する仲間同士が互いの速度に合わせようとする。
- 群れの集中化：隣接する仲間同士が互いに近くに留まろうとする。

群れを外から見ると、ある目的をもって振る舞い、その目的を達成するために協力しているエージェントの集団のように見える。この集団的な知性は、多くの場合において比較的単純なエージェントで構成される大きな集団から創発される。個々のエージェントは簡単な局所的ルールに従って自分たちの行動を決定し、そのようなエージェント同士の相互作用が集団全体において行われることで群れは自らの目的を達成する。まさに、グループ内での様々な行動の集合から創発される自己組織化の一つである。

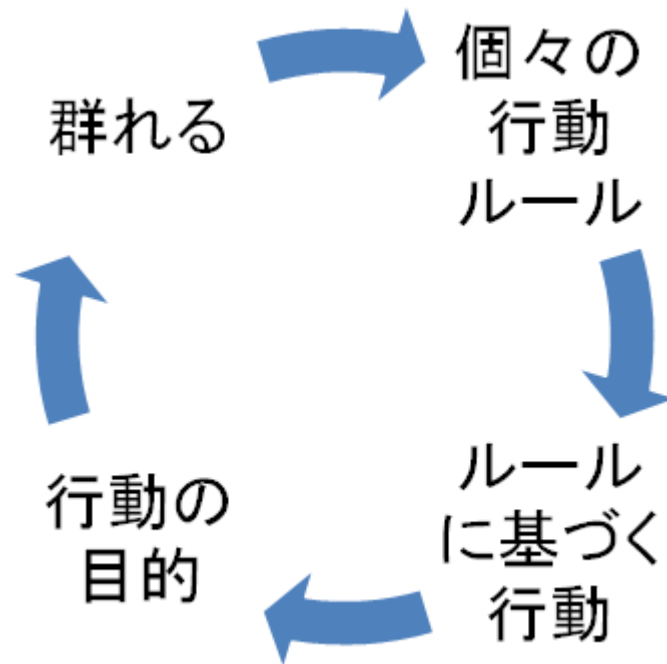


図 2.2: 群れの基本的な振る舞い

自律エージェント [5] は環境と相互作用するサブシステムであり, その環境自体も他のエージェント群によって構成されるものの, 個々のエージェントは自分以外のどのエージェントからも比較的独立している. つまり, 自律エージェントはリーダーからの命令や集団としての計画に基づいて行動するのではない. 例えば集団にある一羽の鳥は集団にいる仲間, 通常は近くにいる仲間の動きに合わせて行動しているだけである. ただし, 近くの鳥のそばにしようとするだけであるが, 衝突は避ける. 集団にはリーダーが存在しないので, それぞれの鳥は誰からの命令を受けることなく, どの鳥も集団の先頭や真ん中, 最後尾になり得る. 鳥たちにとって群れで行動することは, 敵から身を守るときや餌を探するときなどにおいて, 様々な利点がある. 以下に昆虫や鳥の群れに見られる集団行動に着想を得た様々なアルゴリズムを紹介する.

### 2.2.1 Boid

Boid[8]とは1987年にアメリカのアニメーション・プログラマであるクレイグ・レイノルズによって考案・作製された人工生命シミュレーションプログラムである。Boidというモデル名は、鳥もどきという意味の言葉である“bird+android(バード・アンドロイド)”が短くなったことに由来している。Boidの群を実現させるふるまいは、3つの要素からなり、「衝突の回避」,「速度を合わせる」,「群の中心に向かう」といった3つのルールを規定するだけで鳥の群をシミュレーションすることができる。

- Separation(引き離し) Separationは、近くの鳥や物体に近づきすぎたらぶつからないように離れるルールです。もし、ボイド同士が近づきすぎてしまったら、前を飛んでいるボイドはスピードを速くし、後ろを飛んでいるボイドはスピードを遅くするようにします。障害物、例えば柱とか壁とかに対しては、それにぶつからないように方向転換して衝突を避けるようにします。

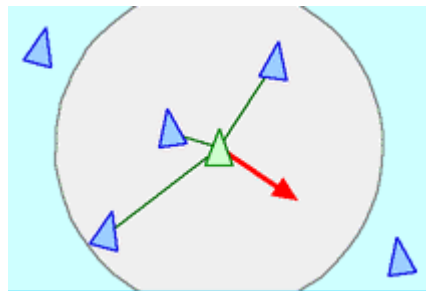


図 2.3: Separation

- Alingment(整列) Alingment は, 近くの鳥たちと飛ぶスピードや方向を合わせようとするルールです. すなわち, 同じ方向にあまり距離を空けないように飛ぶようにします. このルールは, ある一定の距離より遠ざかりすぎてしまったら前を飛んでいるボイドはスピードを遅くし, 後ろを飛んでいるボイドはスピードを速くするようにすることで実現することができます.

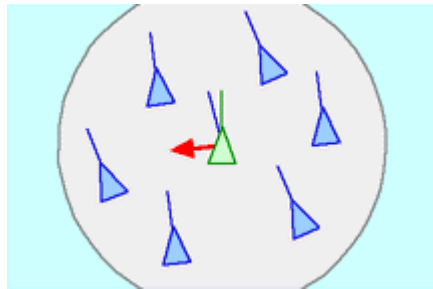


図 2.4: Alingment

- Cohesion(結合) Cohesion は, 鳥たちが多くいる方へ向かって飛ぶルールです. 鳥が多くいる方向というのは, 大ざっぱにいうと群れの中心 (重心) 方向ということになります. ですからこのルールは, ボイドに群れの中心の方向へ飛んでいくことを指示しています. この群れの中心をどう出すかですが, これは全ボイドの位置 (座標) の平均として求めます.

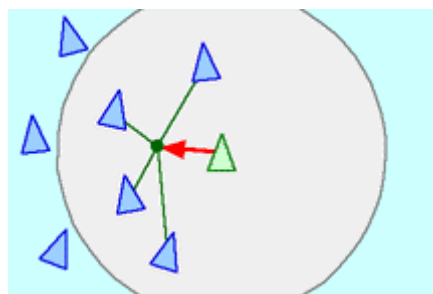


図 2.5: Cohesion

### 2.2.2 ACO(Ant Colony Optimization)

アリの摂食行動から着想を得たアルゴリズム [10]. 実世界では, アリは始めランダムにうろつき, 食物を見つけるとフェロモンの跡を付けながらコロニーへ戻る. 他のアリがその経路を見つけると, アリはランダムな彷徨を止めてその跡を辿り始め, 食物を見つけると経路を補強しながら戻る. しかし, 時間とともにフェロモンの痕跡は蒸発しはじめ, その吸引力がなくなっていく. その経路が長いほどフェロモンは蒸発しやすい. それに対して, 経路が短ければ行進にも時間がかからず, フェロモンが蒸発するよりも早く補強されるため, フェロモン濃度は高いまま保たれる. 従って, あるアリがコロニーから食料源までの良い (すなわち短い) 経路を見つけると, 他のアリもその経路を辿る可能性が高くなり, 正のフィードバック効果によって結局すべてのアリが1つの経路を辿ることになる.

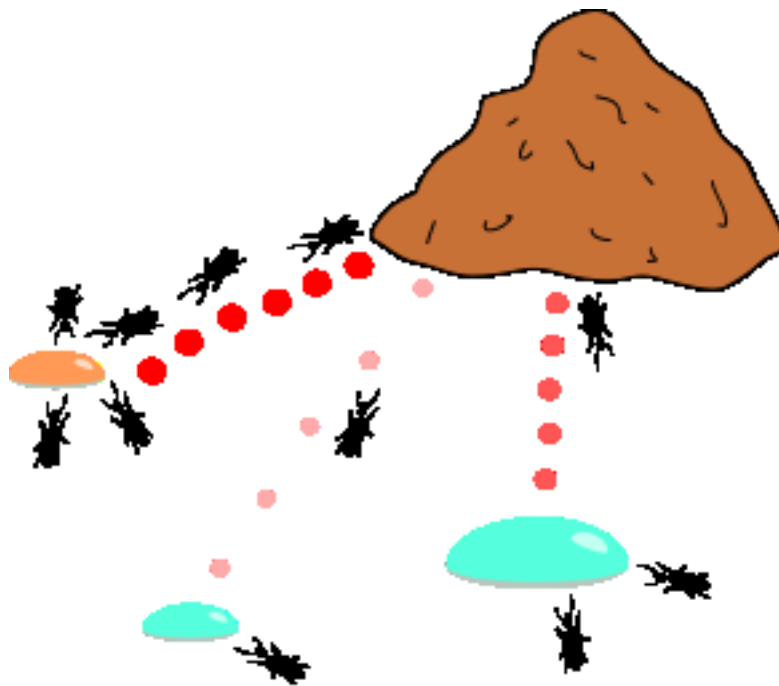


図 2.6: Ant Colony Optimization

### 2.2.3 PSO(Particle Swarm Optimization)

魚や鳥や昆虫などの生物の検索行動に着目し James Kennedy が考案したアルゴリズムである [9]. 昆虫の大群や魚群において, 一匹がよさそうな経路を発見すると, 群れの残りはどこにいても素早くそれに向かうことができる. これは多次元空間において位置と速度を持つ粒子群でモデル化される. これらの粒子はハイパー空間を飛びまわり, 最適な位置を探す. 位置の評価は適応度関数で行う. 群れのメンバーは良い位置について情報交換し, それに基づいて自身の位置と速度を調整する.

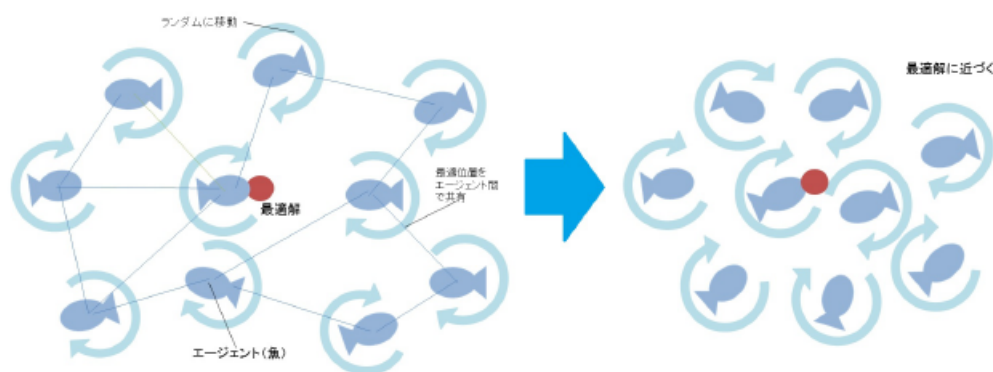


図 2.7: Particle Swarm Optimization

## 2.3 家族的類似

ルートヴィヒ・ウィトゲンシュタインの言語ゲームで振る舞う particle は, 家族的類似性によって群を作ると定義した. 家族的類似性は言語と行為の類似性を表現するものである. 家族的類似性は同値関係でもないし, 等価関係でもない. 常に変動しつつ, 少しずつ似ているエンティティの集まりである. しかし, それは自己から見れば, 同値関係であってもよいし等価関係であってもよい. そのような個人個人の意味論を統合して世界を記述する意味論が無いということである. 家族的類似性は不確実な世界の中の同類の定義である. 家族的類似性は公的言語世界の観察視点の同類の定義である.

群知能において群れるふるまいは家族的類似に基づいている.

## 2.4 メタデータ

データについてのデータ. あるデータそのものではなく, そのデータに関連する情報のこと. データの作成日時や作成者, データ形式, タイトル, 注釈などが考えられる. データを効率的に管理したり検索したりするために重要な情報である. 文書や画像などを保存するファイル形式の多くは, ファイルの先頭にメタデータを格納し, 続いてデータ本体を格納するようにできている.

## 2.5 インデックス

データベースや検索エンジンでは, テーブルなどに格納されたデータをすばやく検索・抽出するための識別データ・索引データなどのことをいう.

## 2.6 リレーショナルデータベース

一件のデータを複数の項目 (フィールド) の集合として表現し, データの集合をテーブルと呼ばれる表で表す方式. ID 番号や名前などのキーとなるデータを利用して, データの結合や抽出を容易に行うことができる.

## 2.7 クラスタリング

自然界において swarm(スウォーム) とは動物の集団 (鳥, 魚などの群れ) のことである. 群れを形成することは, 捕食動物に対する防衛や狩猟採集の効率を向上させる利点がある. 群れの形を維持するために, 群れに属する動物はいくつかの規則に従って行動する. 例えば, 近くの仲間との距離を保つことや, 衝突を避けることなどである. さらに自然界では, 混合種の群れ (すなわち複数種の群れ) も存在する.

### 2.7.1 データのクラスタリング

データのクラスタリングとは, データの集合をデータ間の類似度に従って, いくつかのグループに分けることである. 動物の種類をデータのタイプと見なすと, 複数種の群れは複数のタイプの混ざったデータセットであると考えられる. どのような観点で類似度を設定するかでクラスタリングの結果は異なるので, クラスタリングには厳密な正解はない.



データのクラスタリングとは、データ集合  $X$  を類似しているデータで構成される部分集合  $C_i \subseteq X$  に分割することである。

クラスタリングの目的は、部分集合  $C_i \subseteq X$  の集合  $C = \{C_i\}$  を得ることである。

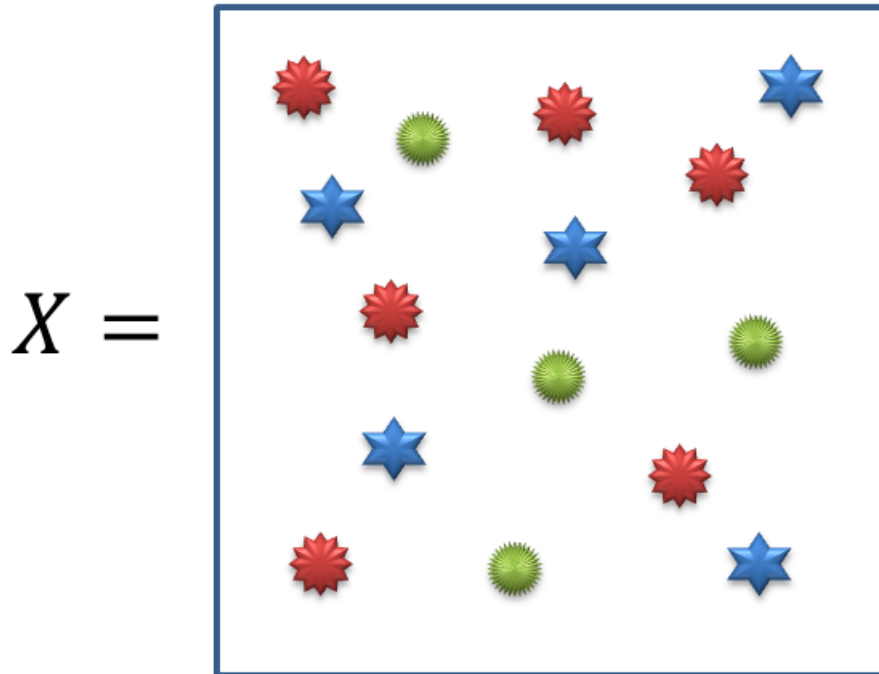


図 2.8: 3種類のオブジェクトを含むデータセット

部分集合  $C_i$  はクラスタと呼ばれ、すべてのクラスタを合わせると全集合  $X$  となる。これは、 $X$  の各要素はいずれかのクラスタに割り当てられることを意味している。通常、クラスタは互いに素である。これは、 $X$  の各要素は必ず一つのクラスタのみに割り当てられることを意味している。あるクラスタ内のデータは類似しており、他のクラスタ  $C_i$  内のデータとは類似していない。

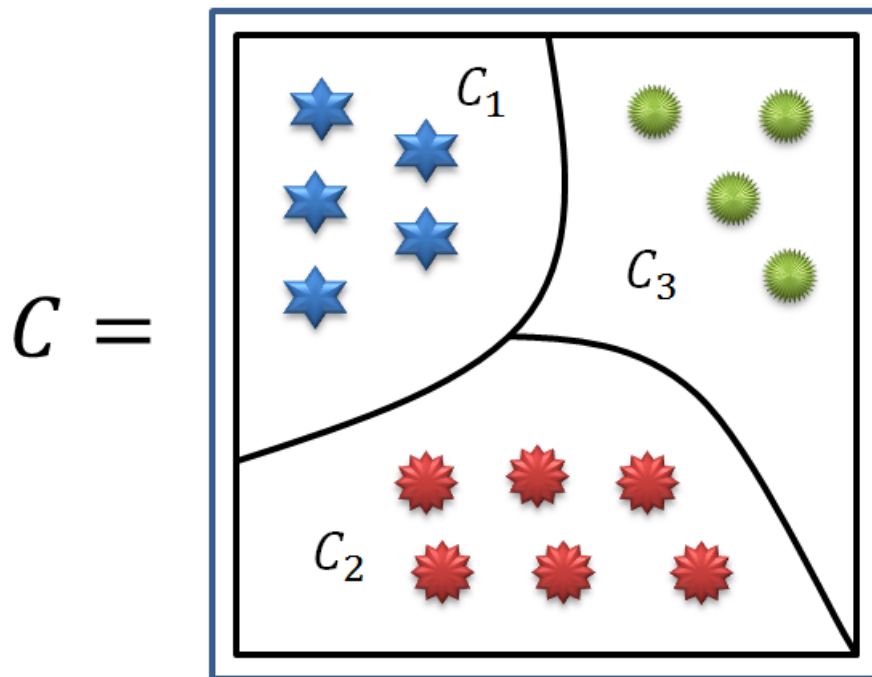


図 2.9: 類似のオブジェクトを持つ 3つのクラス

多くの場合、データは  $n$  次元の属性空間  $A = A_1 \times \dots \times A_n$  内の点  $d_i = (d_{i1}, \dots, d_{in})^T \in A$  である。それぞれの  $d_{iv} \in A_v (1 \leq v \leq n)$  は、ある属性値を表している。

また、データ同士が同じクラスに属するかどうかを決定する場合には、属性空間内での距離が用いられる。

クラスタリングは多くの分野で応用されている。例えば、文字認識、画像分割、画像処理、Web マイニング、データマイニングなどで用いられている。

また、クラスタリングアルゴリズムは階層的クラスタリング、分割型クラスタリング、密度分布に基づく分割、グリッドベースの方法などに分類される。

クラスタリングの手法は,2つの観点から分類される.

1. 階層的手法と非階層的手法
2. ソフトクラスタリングとハードクラスタリング

### 2.7.2 階層的手法

階層的手法は,N個の対象からなるデータが与えられたとき,1個の対象だけを含むN個のクラスタが初期状態を作る.初期状態から2つの対象データ  $x_1, x_2$  間の距離  $d(x_1, x_2)$ (非類似度) からクラスタ間の距  $d(C_1, C_2)$  を計算し,最もこの距離に近い2つのクラスタを逐次的に併合する.そして,この併合を全ての対象が一つのクラスタに併合されるまで繰り返し,階層構造を得る.この階層構造はデンドログラムによって表示される.デンドログラムとは,各終端ノードが各対象を表し,併合されてきたクラスタを非終端ノードで表した二分木である.クラスタ間の類似度を求める方法はいくつかある.

- 最短距離法最も近いデータの距離を2つのクラスタ間の距離と定義する.

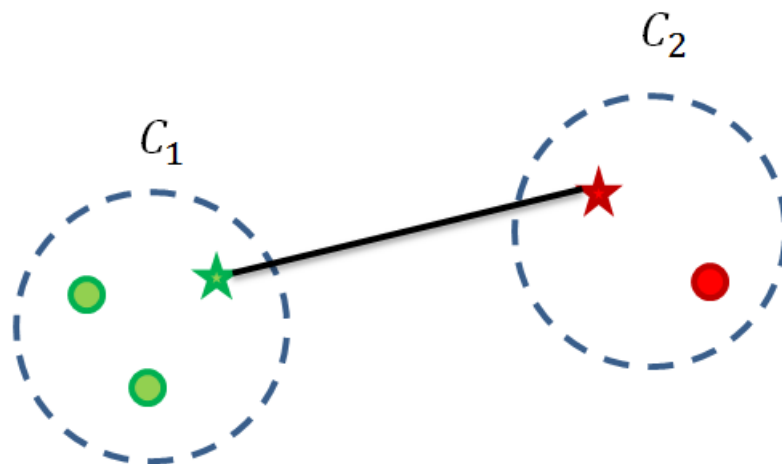


図 2.10: 最短距離法

- 最長距離法最も遠いデータの距離を2つのクラスタの距離と定義する.

$$d(C_1, C_2) = \max_{x_1 \in C_1, x_2 \in C_2} d(x_1, x_2)$$

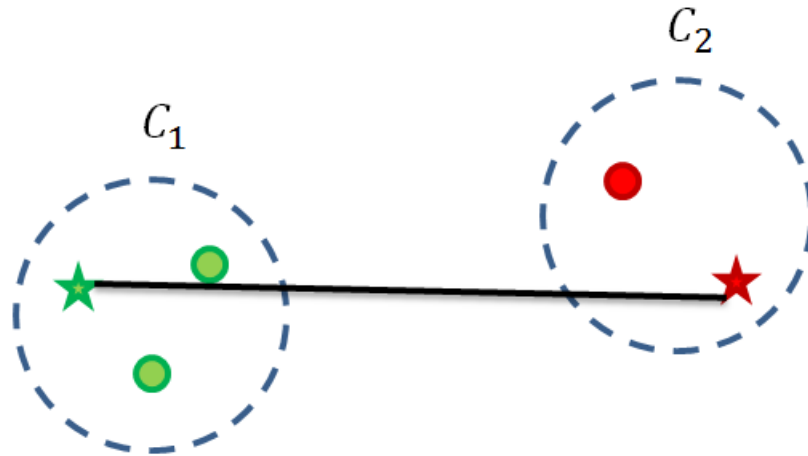


図 2.11: 最長距離法

- 重心法2つのクラスタのそれぞれの重心を求めて重心間の距離をクラスタの距離と定義する.

$$d(C_1, C_2) = \|m_{C_1} - m_{C_2}\|$$

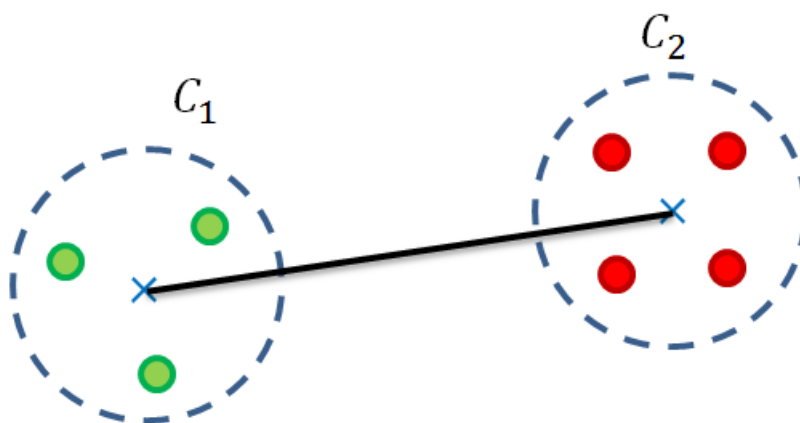


図 2.12: 重心法

### 2.7.3 非階層的手法

データの分割の良さを表すある評価関数によって、最適解を探索する手法。階層的な手法ではデータが多いと階層構造が複雑になってしまうため、非階層的手法のほうが実用的。

### 2.7.4 ソフトクラスタリング

データが複数のクラスターに属することを許すクラスタリング手法。

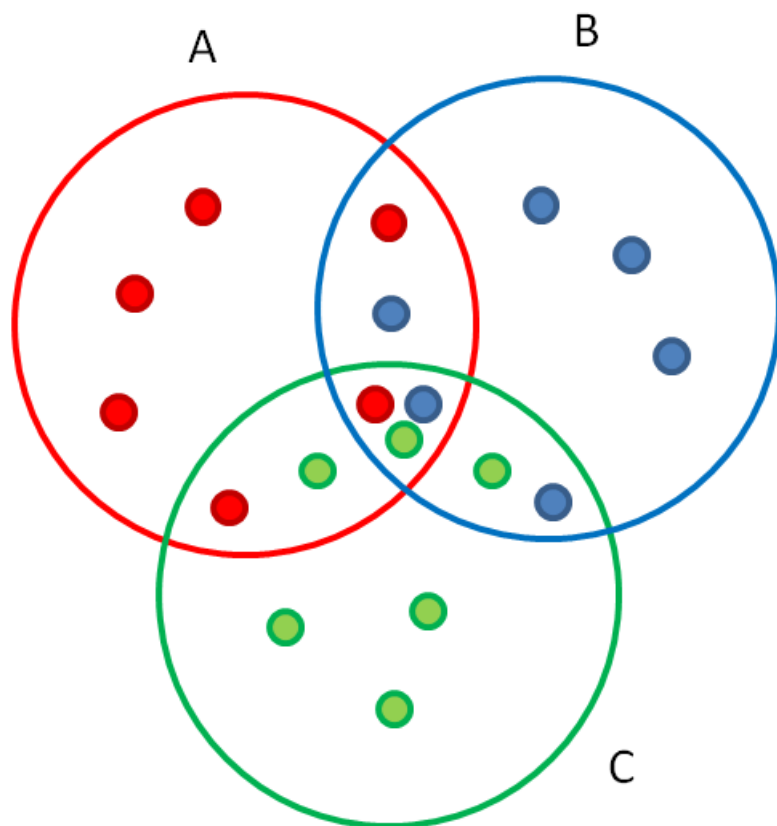


図 2.13: ソフトクラスタリング

### 2.7.5 ハードクラスタリング

すべてのデータが、単一クラスタに属するようなクラスタリング手法.

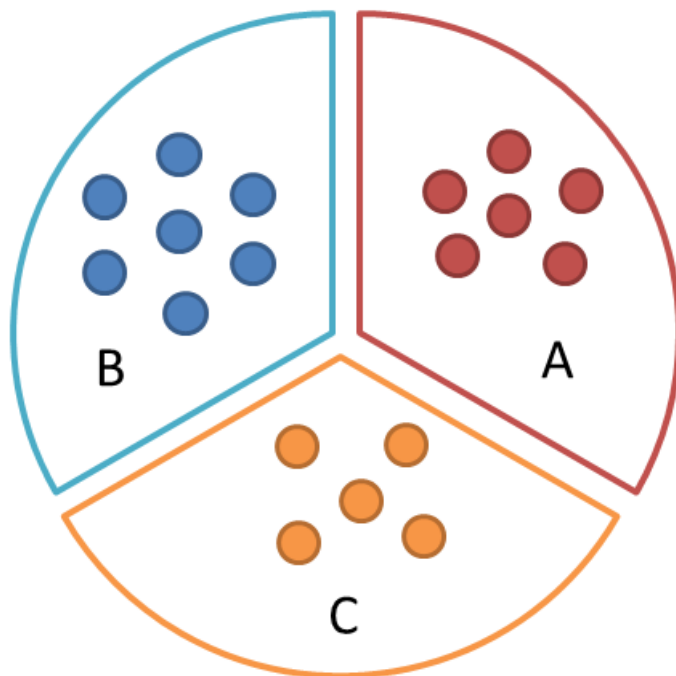


図 2.14: ハードクラスタリング

## 第3章 提案

### 3.1 目的

従来のファイルシステムでは、探しているファイルがどのフォルダに入っているのか忘れてしまうことがある。また、クラウドでは、ファイルが大量なのでコンピュータを利用する人がファイルを目的や用途などに応じて分類するのは手間が増加してしまう。ファイル数が少ない場合には、手作業のファイル整理は可能であるが、ファイル数が多量になると手作業で整理するのは困難である。クラウド内の大量なファイルを自動的に整理・分類できるようにし、重要なファイル・関連するファイルを視覚的に見やすいファイルシステムを開発することが目的である。そこで群知能とファイルのメタデータを用いて、関連のあるファイル同士が集まった群れを複数作り、探しているファイル・重要なファイルがコンピュータを利用するユーザにとって一目でわかる、見やすいファイルシステムを提案する。

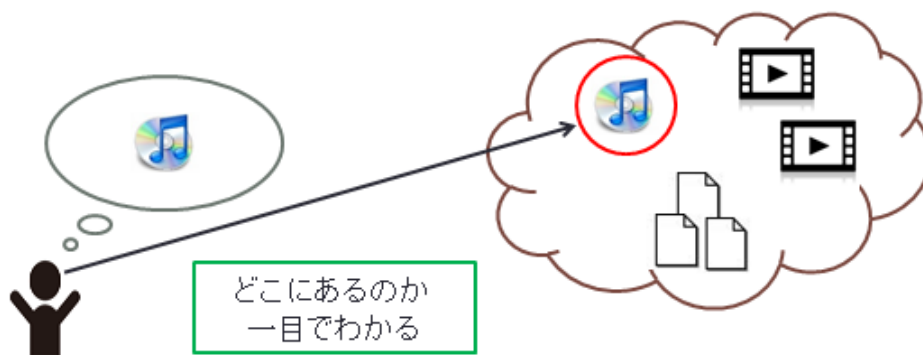


図 3.1: 研究目的

### 3.2 ファイルの属性による分類を表現

Boid モデルでは基本的にある単一の種の鳥の群れを表現する. つまり, エージェントは皆互いに仲間であり, 他どのエージェントとも群れを作ろうとする. これだと最終的に一つの群れにしかならない. しかし, ファイルシステムに格納されているファイルデータは各々異なっており, ファイルが似た属性とのみ群れを形成するような群知能モデルでなければ, ファイル整理・分類することはできない.

そこで, Boid モデル上の各エージェントに異なる属性を記録できるようにし, その属性に応じた群れを形成するような群知能モデルにする必要がある. 今回はファイルのメタデータ (ファイル名, 作成/更新/閲覧日時, サイズ, キーワード, データの形式) で各エージェント进行分类し, 群知能によるクラスタリングを行い, 似たファイル同士のみ群れを形成する.

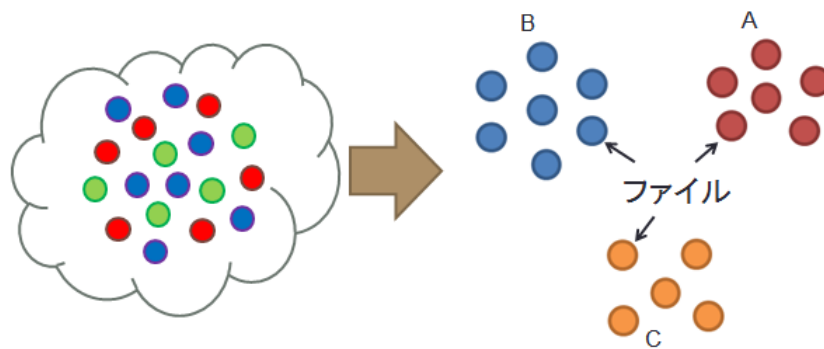


図 3.2: ファイルの属性による分類

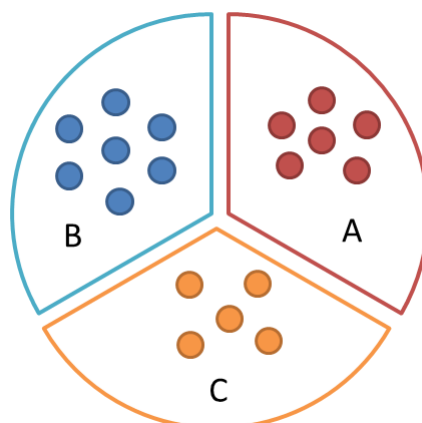


図 3.3: クラスタリング



### 3.3 視認性向上のための群れの安定化

自然界の群れをなす鳥が飛び続けるように, Boid モデルにおいてもエージェントは絶えず動き続けている. しかし, ユーザがクラウドファイル上のデータを群れとして見る際に, 群れ, つまりファイル同士の関係性が常に動き続け, 変化しては, 関係性を把握することが非常に難しい. そこで, 周辺の他エージェントの数により自身の群れの規模を認識して群れの完成度を定義し, エージェントの速度を調整する. 群れの規模がある程度大きければ, その群れは情報の群れとして完成度が高く, ユーザに把握しやすいように移動速度をさげるのがふさわしい. 理想としてはユーザは探しているファイルを見つけた場合, そのファイルをクリックなどで閲覧したり, コピーや削除などの操作を行うと考えられるので, 群れが完成したら, 探索を終了し静止した状態で集まったファイルを見せる. 一方, 群れの規模が小さい, もしくは群れにエージェント自身しかいない場合, そのエージェントは情報の群れを形成しているとは言い難く, どこかの群れに属するために他エージェントを探して活発に動く必要があり, 移動速度を上げるのが望ましい. つまり, 群れに属するエージェントの数によって移動速度を調整する. これによって, 群れの形成を極端に阻害することなく, 使用者がファイルの群れを視覚的に見た際に, ファイルの関係性を留意に素早く把握することが可能となる.

#### 3.3.1 群知能によるクラスタリング

群知能によるクラスタリング (Data Swarm Clustering: DSC) は, 複数種で構成される一つの大きな群れから, それぞれ同種のみを含む部分群への分割を模倣したものである. この DSC を用いて, すべてのファイルをそれぞれ似ているファイル同士の部分群へ分割しクラスタとしてグループ化される. この群知能によるクラスタリングを用いたファイルシステムのモデルを 2 つ提案する.

### 3.3.2 カテゴリによるファイルの分割

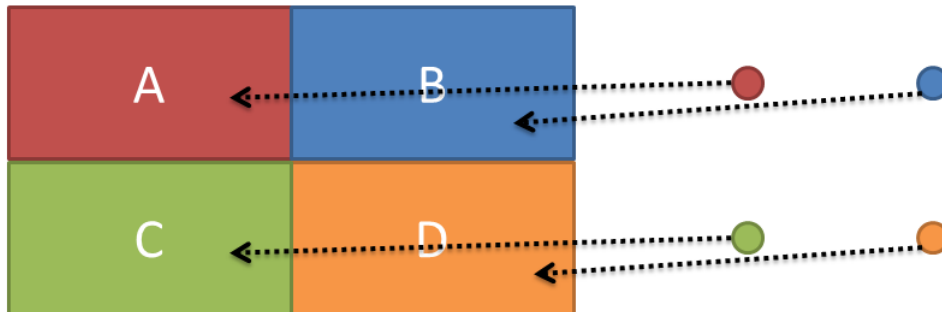


図 3.4: カテゴリによるファイルの分割

ファイルのメタデータをカテゴリとして設定し、ファイル自身が関連するカテゴリの中に集まってくるようにする。複数のカテゴリにあてはまるファイルが存在する場合を考えて隣接するカテゴリは類似した関係なるように設定する。

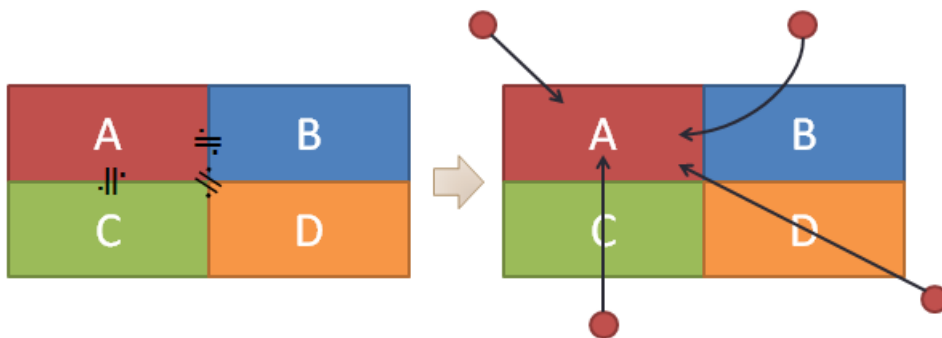


図 3.5: 隣接するカテゴリの設定

### 3.3.3 中心にあるファイルに対して類似するファイルを集める

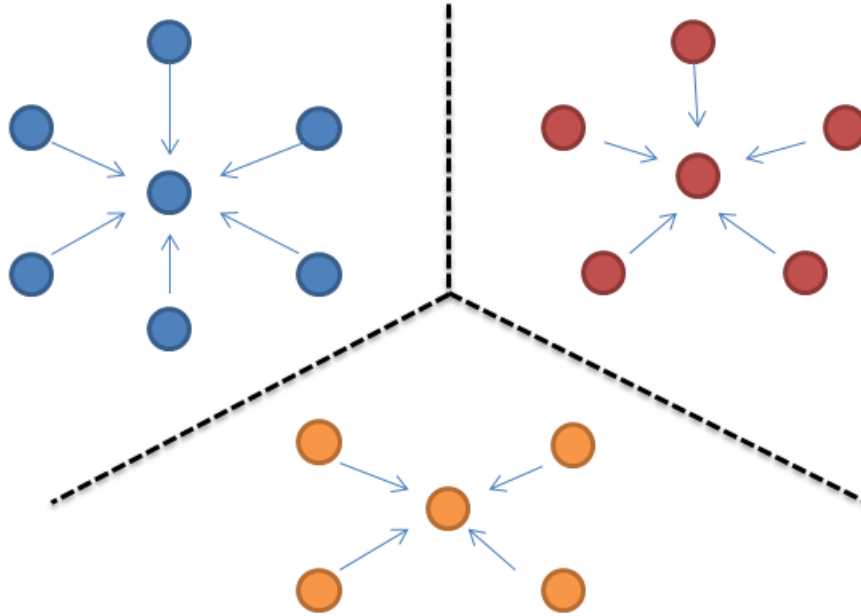


図 3.6: 中心にあるファイルに対して類似するファイルを集める

中心となるファイルを決め, その中心のファイルに類似するファイルが中心に集まってくるようにし, 類似するファイルで構成された部分群にする.

### 3.3.4 分類されたファイルの見せ方

このファイルシステムを用いると、探しているファイルやそれに関連するファイルがコンピュータを利用するユーザにとって一目でわかるようになる。

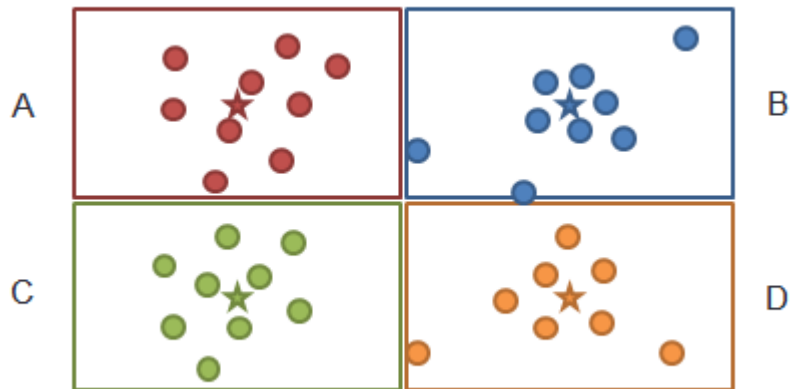


図 3.7: ファイルの見せ方

## 第4章 結論

## 第5章 謝辞

## 第6章 質疑応答

## 第7章 付録



## 参考文献

- [1] 木下宏揚:”情報処理工学”, コロナ社 (2001)
- [2] 石田克憲, 木下宏揚”異なる戦略の群知能を組み合わせたクラウドファイルシステムの実装” 神奈川大学卒業論文 (2012)
- [3] 道下裕介, 木下宏揚”群知能のクラウドファイルシステムへの応用” 神奈川大学卒業論文 (2012)
- [4] 磯村淳, 木下宏揚”クラウドファイルシステム” 神奈川大学卒業論文 (2011)
- [5] (編) アジス・アブラハム, クリナ・グローサン, ヴィクトリーノ・ラモス (役) 栗原聡, 福井健一:”群知能とデータマイニング”, 東京電機大学出版局 (2012)
- [6] 宮元裕樹”セマンティックファイルシステムのフレームワークに関する研究” 高知工科大学大学院 工学研究科 (2011)
- [7] 石川憲一, 森嶋厚行, 田島敬史:”大規模ドキュメント空間管理のための意味ファイルシステムの構築” 社団法人 情報処理学会 (2006)
- [8] 石橋佳明, 吉田典正:”階層的ボイドアルゴリズムによる大規模な魚群のシミュレーション” 日大生産工
- [9] 坊原尚記, 作田健, 香月智:”PSO に対するランダム探索機能付加効果に関する考察” 第9回設計工学に関するシンポジウム 講演論文集 (2002)
- [10] 飯村伊智朗, 松留貴文, 中山茂:”群知能 ACO における女王アリ戦略の検討” 電子情報通信学会 (2005)
- [11] Google Desktop Search ”<http://google-desktop.softonic.jp/>”
- [12] Spotlight ”[http://support.apple.com/kb/HT2531?viewlocale=jaJP&locale=ja\\_JP](http://support.apple.com/kb/HT2531?viewlocale=jaJP&locale=ja_JP)”
- [13] 田中洸一:”クラスタリング入門”クラスタリングの概念, 結果に対する評価基準
- [14] デジタル情報の処理と認識:”階層的クラスタリング”(2012)

- 
- [15] 飯村伊智朗, 中山茂”群知能によるクラスタリングにおける単純 Ant-based Clustering 戦略の検討”システム制御情報学会論文誌 (2006)
  - [16] DW van der Merwe, AP Engelbrecht”Data Clustering using Particle Swarm Optimization”(2003)
  - [17] Lizhong Xiao, Zhiqing Shao and Gang Liu”K-means Algorithm Based on Particle Swarm Optimization Algorithm for Anomaly Intrusion Detection”(2006)