
CENTRO TECNOLÓGICO

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA

# Algoritmos para Análise de Similaridade e Mineração de Trajetórias MultiAspecto

Matheus Henrique Schaly

**ORIENTADORA**
Vania Bogorny

Agosto de 2019

# Contents

# 1 Resumo

Este projeto se baseia na utilização de algoritmos de *clustering*, em dados de trajetórias multiaspecto, que permita agrupar trajetórias que possuam características similares. Primeiramente necessitou-se a leitura de artigos científicos para a área de trajetórias brutas, semânticas e multiaspecto. Em seguida realizou-se uma pesquisa sobre algoritmos de *clustering* tradicionais. Alguns desses algoritmos pesquisados foram aplicados para realizar experimentos utilizando tanto dados artificiais como dados reais de trajetórias multiaspecto. Para realizar os experimentos, utilizou-se a linguagem de programação Python e algumas de suas bibliotecas, tais como NumPy e Pandas para a análise dos dados, Matplotlib, Seaborn e Plotly para a visualização dos dados, e Scikit-learn para os algoritmos de aprendizagem de máquina. Posteriormente, realizou-se a leitura e experimentos de algoritmos da área de *co-clustering*, *fuzzy clustering* e *fuzzy co-clustering*. Por fim, optou-se por fazer os experimentos finais utilizando o algoritmo *PaNDa*$^{+}$ (*Patterns in Noisy Datasets*) que possui características similares a algoritmos de *fuzzy co-clustering*.

**Palavras-chave:** Trajetórias Multiaspecto, *Co-Clustering*, *Fuzzy Clustering*, *Data Science*.

# 2 Introdução

A trajetória bruta é a representação de trajetória mais simples, possuindo apenas a informação de espaço e tempo associado com cada ponto da trajetória. Trajetórias brutas podem ser enriquecidas com mais informações, tais como o nome do lugar visitado por um indivíduo, chamado de *Point of Interest* (*POI*), e a quantidade de tempo despendida em cada *POI*. Com mais informação associada às trajetórias brutas, uma nova representação é definida, a trajetória semântica [1]. Uma trajetória multiaspecto vai além de uma trajetória semântica. Em uma trajetória multiaspecto temos o enriquecimento da trajetória com mais aspectos, tais aspectos podem ser locais visitados, condições climáticas, meios de transporte, posts de mídia social e a saúde. Além disso, cada aspecto pode ser descrito por seus próprios atributos, por exemplo, o local visitado pode possuir uma posição espacial, uma categoria, uma nota referente a qualidade, e o preço [2].

*Clustering* é o processo de agrupar um conjunto de objetos físicos ou abstratos em classes de objetos similares. *Clustering* tem sido amplamente utilizado em inúmeras aplicações, como pesquisa de mercado, reconhecimento de padrões, análise de dados e processamento de imagens. Diversos algoritmos de clustering foram relatados na literatura, como *k-means*, *BIRCH*, *DBSCAN*, *OPTICS* e *STING* [3]. Normalmente, os dados que surgem das aplicações para o *clustering* são organizados como uma tabela de contingência ou co-ocorrência [4], tal como uma matriz de palavra por documento, onde as linhas correspondem a palavras e as colunas correspondem a documentos [3]. Tal matriz pode se beneficiar do algoritmo de *co-clustering* que é capaz de simultaneamente agrupar as duas dimensões da tabela de contingência [4]. Por outro lado, o clustering baseado em logica *fuzzy*, como o *Fuzzy C-Means* (*FCM*) considera a incerteza ao permitir que cada elemento de dado pertença a um cluster diferente por um certo grau de pertencimento [5]. O método de *fuzzy co-clustering* estende o método de *co-clustering* ao atribuir funções de pertencimento tanto às linhas quantos às colunas [6]. Já o algoritmo *PaNDa*$^{+}$ possui o objetivo de descobrir um conjunto *k* de padrões que melhor descreve, ou modela, os dados de entrada [7]. O *PaNDa*$^{+}$ possui características semelhantes aos algoritmos de *fuzzy co-clustering*.

Este documento aborda as principais partes do processo de aprendizagem, com o objetivo de demonstrar os passos que foram realizados para chegar ao resultado final. Para isso demonstra-se na seção 3 os *datasets* utilizados, e, na seção 4, aprofunda-se nos algoritmos de *clustering*, *fuzzy clustering*, *co-clustering*, *fuzzy co-clustering* e o algoritmo selecionado *PaNDa+*. Em seguida, na seção 5, aborda-se como os testes dos algoritmos foram realizados. Por fim, na seção 6, analisam-se os resultados os obtidos durante o processo de pesquisa.

# 3 Datasets

Nesta seção será apresentado os *datasets* utilizados nos experimentos de *clustering*. Os três primeiros datasets (Foursquare, Gowalla e Brightkite) são *datasets* reais de trajetórias. O último dataset é um *toy dataset* criado apenas para melhor entender o funcionamento dos algoritmos de *clustering*.

## 3.1 Foursquare

O Forsquare *dataset* possui 66.962 linhas e 14 colunas, as 5 primeiras linhas estão na figura 1. Cada linha representa um *check-in*. Enquanto as colunas definem:

- *checkin_id* possui o id do *check-in*;
- *venue_id* é o id do local;
- *tid* é o id da trajetória, onde cada id representa uma semana distinta;
- *lat_lon* é a latitude e longitude do local onde o *check-in* foi realizado;
- *date_time* é o ano, mês, dia e hora; *time* são os minutos passados dês do começo do daquele dia;
- *day* é o dia da semana;
- *poi* é o *point of interest*, isto é, o nome do local no qual o *check-in* for realizado; *type* é uma generalização do *POI*;
- *root_type* é uma generalização do *type*;
- *price* é o custo referente ao *POI*, onde -1 indica a ausência do atributo *price*; *rating* é a nota dada ao *POI*;
- *weather* é a condição do tempo no local onde o *check-in* foi realizado;
- *label* é o usuário que realizou o *check-in*, isto é, todos os *labels* iguais pertencem ao mesmo usuário.

O *dataset* possui 10809 *POIs* distintos, 3079 *TIDs* distintos e 193 *labels* distintos.

| | checkin_id | venue_id | tid | lat_lon | date_time | time | day | poi | type | root_type | price | rating | weather | label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 283468 | 4eba3331722edc0eaf1762bb | 126 | 40.8331652006224 -73.9418603427692 | 2012-11-12 05:17:18 | 317 | Monday | The Lair Of Modern Strange Cowboy | Home (private) | Residence | -1 | -1.0 | Clear | 6 |
| 1 | 284212 | 4fe9524ce4b0d971aa120f82 | 126 | 40.8340978041072 -73.9452672225881 | 2012-11-12 23:24:55 | 1404 | Monday | Galaxy Gourmet Deli | Deli / Bodega | Food | 1 | 8.2 | Clouds | 6 |
| 2 | 284225 | 4eba3331722edc0eaf1762bb | 126 | 40.8331652006224 -73.9418603427692 | 2012-11-13 00:00:07 | 0 | Tuesday | The Lair Of Modern Strange Cowboy | Home (private) | Residence | -1 | -1.0 | Clouds | 6 |
| 3 | 284771 | 4d5febf914963704def9dd94 | 126 | 40.7646959283254 -73.8851974964414 | 2012-11-15 17:49:01 | 1069 | Thursday | Popeyes Louisiana Kitchen | Fried Chicken Joint | Food | 3 | 6.6 | Clear | 6 |
| 4 | 284821 | 4dd408046365c27b0dbf4239 | 126 | 40.7660790376824 -73.8835287094116 | 2012-11-15 18:40:16 | 1120 | Thursday | MTA Bus Operations Depot - LaGuardia | Bus Station | Travel & Transport | -1 | -1.0 | Clear | 6 |

*Figura 1 Primeiras 5 linhas do Foursquare dataset*

### 3.2 Gowalla

O Gowalla *dataset* possui 98.158 linhas e 7 colunas, as 5 primeiras linhas estão na figura 2. Assim como no Foursquare, cada linha representa um *check-in*. As colunas são também semelhantes às encontradas no Foursquare. O *dataset* possui 24374 *POIs* distintos, 5329 *TIDs* distintos e 300 *labels* distintos.

| | tid | label | day | hour | poi | lat | lon |
|---|---|---|---|---|---|---|---|
| **0** | 3044 | 142 | Friday | 19 | 16194 | 39.067175 | -94.581840 |
| **1** | 3044 | 142 | Saturday | 12 | 74164 | 39.294670 | -94.719100 |
| **2** | 3044 | 142 | Saturday | 14 | 261214 | 32.845378 | -96.852121 |
| **3** | 3044 | 142 | Saturday | 15 | 88490 | 32.785507 | -96.801535 |
| **4** | 3044 | 142 | Saturday | 22 | 335003 | 32.785395 | -96.800940 |

*Figura 2 Primeiras 5 linhas do Gowalla dataset*

### 3.3 Brightkite

O Brightkite dataset possui 169982 linhas e 7 colunas, as 5 primeiras linhas estão na figura 3. O formato do dataset é idêntico ao do Gowalla. O dataset possui 4913 *POIs* distintos, 7911 *TIDs* distintos e 300 *labels* distintos.

| | tid | label | day | hour | poi | lat | lon |
|---|---|---|---|---|---|---|---|
| **0** | 3174 | 41 | Wednesday | 16 | 717121 | 39.739154 | -104.984703 |
| **1** | 3174 | 41 | Wednesday | 20 | 717121 | 39.739154 | -104.984703 |
| **2** | 3174 | 41 | Wednesday | 20 | 717121 | 39.739154 | -104.984703 |
| **3** | 3174 | 41 | Thursday | 2 | 717121 | 39.739154 | -104.984703 |
| **4** | 3174 | 41 | Thursday | 3 | 717121 | 39.739154 | -104.984703 |

*Figura 3 Primeiras 5 linhas do Brightkite dataset*

### 3.4 Toy dataset

O *toy dataset* possui apenas 6 linhas e 7 colunas, o *dataset* completo é a figura 4. Neste *dataset*, cada linha representa um *check-in* e cada coluna representa um *POI* distinto. Ou seja, a linha 0 coluna 0 mostra que o *check-in* 0 foi no *POI*, a linha 3 coluna 2 mostra que o *check-in* 3 foi não foi no *POI* 2. Este *dataset* artificial é utilizado apenas para melhor visualizar o resultado do *PaNDa+*.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **1** | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **2** | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **3** | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| **4** | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| **5** | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

*Figura 4 Toy dataset completo*

# 4 Clustering

A análise de *cluster* (também chamado de aprendizagem não supervisionada) consiste em distinguir, no conjunto de dados analisados, os grupos, denominados *clusters*. Esses grupos são subconjuntos disjuntos do conjunto de dados, possuindo tal propriedade que os dados pertencentes a diferentes *clusters* diferem entre si muito mais do que os dados, pertencentes ao mesmo *cluster*. O papel da análise de *cluster* é, portanto, descobrir um certo tipo de estrutura natural no conjunto de dados. Os meios que permitem a realização dessa tarefa são geralmente constituídos por uma certa medida de similaridade ou dissimilaridade. A análise de *cluster* não é apenas uma ferramenta cognitiva importante, mas também um método para reduzir grandes conjuntos de dados, uma vez que permite a substituição de um grupo de dados por sua caracterização compacta, como, por ex. o centro de gravidade do grupo dado [8]. Nos exemples de *clustering* da seção 4, todos os *datasets* possuem *ground truth* (*label*). Isto é, sabe-se os grupos que devem ser formados, portanto, podemos utilizá-los para verificar a qualidade dos *clusters* formados pelos algoritmos de *clustering*.

## 4.1 Fuzzy clustering

Enquanto que no *cluster* regular (ou *hard clustering*) cada objeto é membro de apenas um cluster, no *fuzzy clustering* (ou *soft clustering*) um objeto pode ser parcialmente classificado em mais de um *cluster*. Para isso, indica-se o grau de pertencimento de cada objeto para cada *cluster*. Suponha que temos *K* clusters e definimos um conjunto de variáveis $m_{i1}$, $m_{i2}$, ..., $m_{iK}$, que representam a probabilidade de que o objeto *i* seja classificado no cluster *k*. Nos algoritmos de *clustering* regulares, um desses valores será um e o restante será zero. Isso representa o fato de que esses algoritmos classificam um objeto em um e somente um cluster.

No *fuzzy clustering*, os membros são distribuídos entre todos os *clusters* a partir de seu grau de pertencimento. Portanto, o $m_{ik}$ pode agora estar entre zero e um, com a estipulação de que a soma de seus valores é um. Chama-se isso de *fuzzification* da configuração do *cluster*. A vantagem da *fuzzificantion* é de não forçar que todos os objetos pertençam a um *cluster* específico. Por outro lado, sua desvantagem, é de que há muito mais informação a ser interpretada.

### 4.1.1 Fuzzy clustering exemplo um

Por exemplo, considere o seguinte *dataset* de duas variáveis cujo valores estão plotados na figura 5 abaixo.

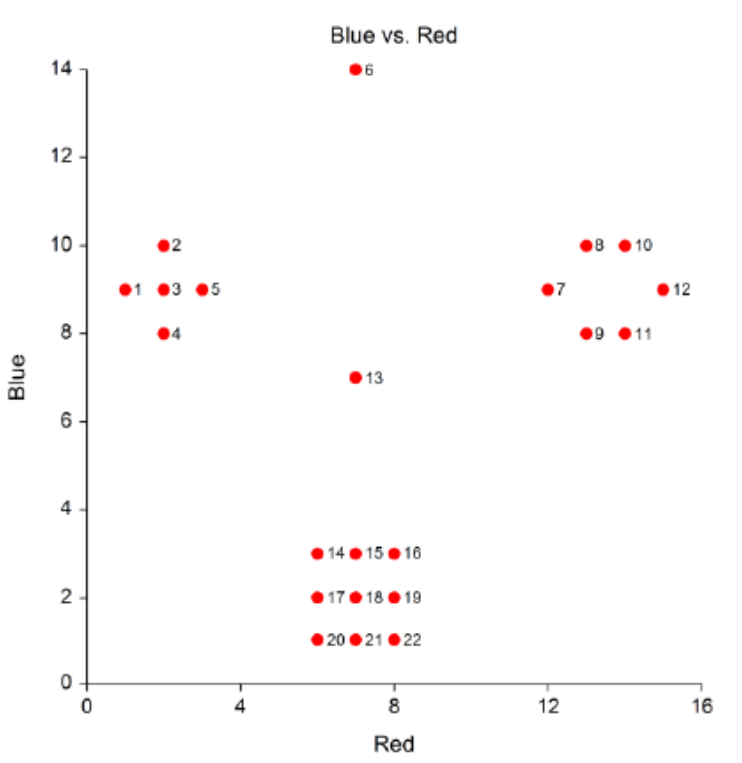


*Figura 5 Conjunto de dados de duas variáveis*

Os dados possuem três *clusters* óbvios e dois pontos discrepantes (6 e 13). Um algoritmo de *clustering* regular procurando por três *clusters* forçará esses dois pontos em *clusters* específicos. Isso pode causar distorção na solução final. O *fuzzy clustering*, no entanto, atribuirá um grau de pertencimento de cerca de 0,33 para cada *cluster*. Esse grau de pertencimento igual indica que esses dois pontos são *outliers* [9].

### 4.1.2 Fuzzy clustering exemplo dois

Agora considere este segundo exemplo prático executado em um *dataset* artificial (figura 6), no qual um algoritmo de *fuzzy clustering* foi executado.

| | 0 | 1 | 2 | 3 | class |
|---|---|---|---|---|---|
| **0** | 2.1 | 1.9 | 3.2 | 0.9 | 2 |
| **1** | 1.9 | 1.8 | 2.9 | 1.1 | 2 |
| **2** | 1.2 | 0.9 | 1.4 | 1.9 | 0 |
| **3** | 0.2 | 0.3 | 0.1 | 4.9 | 1 |
| **4** | 0.3 | 0.1 | 0.2 | 5.1 | 1 |

*Figura 6 Dataset artificial*

O algoritmo utilizado pode ser encontrado em maior detalhe em [10]. O nome do algoritmo utilizado é *FCM* e seus sete parâmetros são:

- *data:* 2d array, size (S, N)
    - Dados para serem agrupados, N é o número de objetos; S é o número de atributos.
- *c:* int
    - Número de *clusters* desejado.
- *m*: float
    - Exponenciação de matriz aplicada à função de associação u_old em cada iteração, em que U_new = u_old ** m.
- *error:* float
    - Critério de parada; pare cedo se a norma de (u [p] - u [p-1]) < erro.
- *maxiter:* int
    - Máximo número de iterações permitidas.
- *init*: 2d array, size (S, N)
    - Matriz inicial particionada por *c fuzzy* partições.
- *seed:* int
    - Se fornecido, define a semente aleatório do *init*. Não possui efeito se o *init* é fornecido. É utilizado principalmente para teste e depuração.

Para rodar o treinamento do algoritmo, foram utilizados os parâmetros *data* sendo o *dataset* da figura 6 sem o atributo (coluna) *class*; *c* = 3; *m* = 1.5; *error* = 0.005; *maxiter* = 1000 e *seed* = 0. Após treinar o algoritmo é possível fazer predição, para maiores detalhes referencie a documentação em [11]. O FCM possui seis retornos, mas apenas a matriz final particionada por c *fuzzy* partições *u* (figura 7) destaca-se para o melhor entendimento do algoritmo.

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | 0.01 | 0.01 | 0.98 | 0.0 | 0.0 |
| **1** | 0.00 | 0.00 | 0.00 | 1.0 | 1.0 |
| **2** | 0.99 | 0.98 | 0.02 | 0.0 | 0.0 |

*Figura 7 Resultado* u *do FCM*

A figura 7 representa o grau de pertencimento de cada objeto para cada *cluster*. As linhas representam os *clusters* e as colunas representam os objetos. Por exemplo, a linha 0 coluna 1 indica que o objeto 0 possui 0.01 de grau de pertencimento em relação ao *cluster* 0. Outro exemplo, a linha 2 coluna 0 indica que o objeto 0 possui 0.99 de grau de pertencimento ao *cluster* 2.

### 4.2 Co-clustering

Com os dados descritos por uma matriz de dados X, geralmente agrupa-se as linhas representando objetos. Mas, do ponto de vista formal, pode-se transpor tal matriz e, com os mesmos algoritmos de *clustering*, pode-se obter o agrupamento de feições que descrevem objetos. Assim, por exemplo, em vez de agrupar documentos de texto, pode-

se agrupar as palavras desses documentos para revelar algumas frases fixas, ou subáreas específicas de uma linguagem, etc. Caso haja várias maneiras distintas de medir propriedades de objetos, como diferentes métodos radiológicos determinar a idade das rochas em geologia, ou diferentes métodos de determinação das contagens de glóbulos vermelhos em amostras de sangue em medicina, diferentes métodos de medição de velocidade de corrosão em química, etc. Pode-se estar interessado em quais destes métodos retornam resultados correspondentes e quais divergem.

Com os dados descritos por uma matriz de dados X, geralmente agrupa-se as linhas representando objetos. Mas, do ponto de vista formal, pode-se transpor tal matriz e, com os mesmos algoritmos de agrupamento, pode-se obter o agrupamento de feições que descrevem objetos. Assim, por exemplo, em vez de agrupar documentos de texto, pode-se agrupar as palavras desses documentos para revelar algumas frases fixas, ou subáreas específicas de uma linguagem, etc. Caso haja várias maneiras distintas de medir propriedades de objetos, como diferentes métodos radiológicos determinar a idade das rochas em geologia, ou diferentes métodos de determinação das contagens de glóbulos vermelhos em amostras de sangue em medicina, diferentes métodos de medição de velocidade de corrosão em química, etc. Pode-se estar interessado em quais destes métodos retornam resultados correspondentes e quais divergem.

Sendo assim, pode-se estar particularmente interessado em agrupar tanto os objetos quanto os atributos ao mesmo tempo. Comumente, esse *cluster* simultâneo de atributos e objetos é chamado e *co-clustering* (ou *bi-clustering*). Assim, ao realizar-se o co-*clustering* do documento da *Web* pode-se descobrir que os documentos são agrupados por idiomas nos quais são escritos, mesmo que não se conheçam esses idiomas. No caso dos métodos de medição da idade de rochas, pode-se detectar ambos os métodos produzindo resultados semelhantes e tipos de rochas para os quais é esse o caso. Outras aplicações interessantes de *co-clustering* dizem respeito à detecção de grupos de clientes que compram determinados produtos, embora seus padrões gerais de compra sejam muito diferentes. Na análise de redes sociais, podemos querer detectar grupos sociais engajados em alguns tipos de atividades sociais. Em geral, com o *co-clustering*, obtém-se uma percepção mais profunda dos dados do que pelo agrupamento separado de objetos e atributos [8].

#### 4.2.1 Co-clustering exemplo

Agora considere este exemplo prático executado em um o *dataset* [11] (figura 8) de um zoológico, no qual um algoritmo de *co-clustering* foi executado. Esse *dataset* possui 101 linhas (animais) e 18 colunas e 7 *class_type* (*labels*) diferentes.

| | animal_name | hair | feathers | eggs | milk | airborne | ... | fins | legs | tail | domestic | catsize | class_type |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | aardvark | 1 | 0 | 0 | 1 | 0 | ... | 0 | 4 | 0 | 0 | 1 | 1 |
| **1** | antelope | 1 | 0 | 0 | 1 | 0 | ... | 0 | 4 | 1 | 0 | 1 | 1 |
| **2** | bass | 0 | 0 | 1 | 0 | 0 | ... | 1 | 0 | 1 | 0 | 0 | 4 |
| **3** | bear | 1 | 0 | 0 | 1 | 0 | ... | 0 | 4 | 0 | 0 | 1 | 1 |
| **4** | boar | 1 | 0 | 0 | 1 | 0 | ... | 0 | 4 | 1 | 0 | 1 | 1 |

*Figura 8 Dataset do zoológico*

O algoritmo utilizado pode ser encontrado em maior detalhe em [12]. O nome do algoritmo utilizado é *Spectral Coclustering* e ele possui oito parâmetros. Entretanto, para não entrar em muitos detalhes, cita-se apenas dois parâmetros:

- *n_clusters*: integer, optional, default: 3
  - O número de *bi-clusters* para serem encontrados.
- *random_state*: int, RandomState instance or None (default)
  - Usado para randomizar a decomposição do valor singular e a inicialização do *k-means*. Use um *int* para tornar a aleatoriedade determinística.

Ao rodar o algoritmo, apenas cria-se uma instancia do tipo *Spectral Coclustering*. Foram utilizados os parâmetros *n_clusters* = 7 e *random_state* = 0. Em seguida, usa-se o método *fit* (*data*) para criar um *bi-clustering* (*co-clustering*) para *data*. O *data* utilizado foi o *dataset* do zoológico sendo que a primeira coluna *animal_name* (objetos) e a última coluna *class_type* (*label*) não são atributos que devem ser utilizados para treinar tais algoritmos. O *Spectral Coclustering* possui vários atributos, mas, para melhor compreendimento, rodou-se outros algoritmos a fim de tornar tais atributos mais fáceis de serem visualizados (figura 9).

```
Therefore examples:
['aardvark', 'antelope', 'bear', 'boar', 'buffalo', 'calf', 'cavy', 'cheetah', 'deer', 'elephant', 'fruitbat', 'giraffe', 'gir
l', 'goat', 'gorilla', 'hamster', 'hare', 'leopard', 'lion', 'lynx', 'mink', 'mole', 'mongoose', 'opossum', 'oryx', 'platypus',
'polecat', 'pony', 'puma', 'pussycat', 'raccoon', 'reindeer', 'squirrel', 'tortoise', 'vampire', 'vole', 'wallaby', 'wolf']
are best described by features:
['hair', 'milk', 'breathes', 'domestic', 'catsize']

Therefore examples:
['bass', 'carp', 'catfish', 'chub', 'dogfish', 'haddock', 'herring', 'pike', 'piranha', 'seahorse', 'sole', 'tuna']
are best described by features:
['fins']
```

*Figura 9 Dois co-clusters gerados ao rodar o algoritmo Spectral Coclustering*

A figura 9 acima mostra o resultado de dois dos sete *co-clusters* gerados. Observe que tanto os objetos foram agrupados quanto seus atributos. Isto é, percebe-se que no primeiro *co-cluster* porco-da-terra, antílope, urso, javali... são caracterizados principalmente pelos atributos cabelo, leite, respiração, domestico, e tamanho de gato. O segundo *co-cluster* afirma que bass, carpa, peixe-gato, caboz... são caracterizados principalmente pelo atributo nadadeira.

## 4.3 Fuzzy co-clustering

Como anteriormente abordado, *co-clustering* é uma técnica para agrupar simultaneamente objetos e atributos. Já o *fuzzy clustering* possui a característica de atribuir funções de pertencimento tanto aos objetos quanto aos atributos, permitindo assim que um mesmo objeto pertença a mais de um grupo. O *fuzzy co-clustering* possui as características tanto do *co-clustering* quanto do *fuzzy-clustering*.

### 4.3.1 Fuzzy co-clustering exemplo

Não se pôde encontrar nenhum algoritmo implementado de *fuzzy co-clustering*. Entretanto, pode-se citar o algoritmo *Fuzzy Clustering for Categorical Multivariate Data (FCCM)* apresentado em [13].

O *dataset* artificial utilizado por [13] é mostrado na figura 10 abaixo. Nele, as linhas representam literaturas e as colunas são as palavras chaves. O *dataset* mostra as relações de co-ocorrência entre as literaturas e as palavras-chave. Cada entrada denota o número de aparições da palavra-chave na literatura correspondente. Por exemplo, a palavra-chave 5 aparece duas vezes na literatura 4.

| | Key1 | Key2 | Key3 | Key4 | Key5 | Key6 | Key7 | Key8 | Key9 | Key10 | Key11 | Key12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lit.1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Lit.2 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| Lit.3 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Lit.4 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Lit.5 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Lit.6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Lit.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Lit.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Lit.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

*Figura 10 Dataset de literaturas e palavras-chave*

O algoritmo pode ser encontrado em maior detalhe em [13]. O *FCCM* possui quatro parâmetros:

- *C:* int
  - Número de *clusters* desejado.
- $T_u$: float
  - Grau de *fuzziness*.
- $T_w$: float

    - Grau de *fuzziness*.
- ε: float
    - Condição de parada.

Ao rodar o algoritmo, apenas cria-se uma instancia do tipo *Spectral Coclustering*. Foram utilizados os parâmetros $C = 2$; $T_u = 0.1$; $T_w = 1.5$ e $\varepsilon = 0.0001$. Os resultados são mostrados nas figuras 11 e 12 abaixo.

| Literature | Cluster 1 | Cluseter 2 |
|---|---|---|
| 1 | 0.338 | 0.662 |
| 2 | 0.011 | 0.989 |
| 3 | 0.011 | 0.989 |
| 4 | 0.002 | 0.998 |
| 5 | 0.141 | 0.859 |
| 6 | 0.894 | 0.106 |
| 7 | 0.988 | 0.012 |
| 8 | 0.996 | 0.004 |
| 9 | 0.973 | 0.027 |

*Figura 11 Grau de pertencimento das literaturas*

| Key word | Cluster 1 | Cluster 2 |
|---|---|---|
| 1 | 0.044 | 0.066 |
| 2 | 0.044 | 0.066 |
| 3 | 0.044 | 0.066 |
| 4 | 0.039 | 0.146 |
| 5 | 0.035 | 0.311 |
| 6 | 0.038 | 0.075 |
| 7 | 0.038 | 0.075 |
| 8 | 0.035 | 0.083 |
| 9 | 0.067 | 0.043 |
| 10 | 0.237 | 0.024 |
| 11 | 0.250 | 0.023 |
| 12 | 0.129 | 0.022 |

*Figura 12 Grau de pertencimento das palavras chave*

Os maiores graus de pertencimento de literaturas e palavras chaves foram sublinhados, pois é assumido que literaturas e palavras chaves são mais prováveis de pertencer ao *cluster* no qual eles possuem maior grau de pertencimento. Ao observar a

figura 11, pode-se ver que as literaturas são dividias em {1, 2, 3, 4, 5} e {6, 7, 8, 9}. Por outro lado, palavras chaves são particionadas em {1, 2, 3, 4, 5, 6, 7, 8} e {9, 10, 11, 12}. Tais resultados estão de acordo com a figura 10 [13].

## 4.4 PaNDa+

O *PaNDa+* é uma estrutura algorítmica capaz de otimizar diferentes funções de custo generalizadas em uma formulação unificadora. Portanto, o *PaNDa+* pode lidar com uma variedade de funções de custo. Além disso, no artigo no qual *PaNDa+* é descrito, também é apresentado uma formulação unificadora de várias funções de custo que são usadas por algoritmos para conduzir suas estratégias de heurística gulosas e avaliar a qualidade dos padrões minerados. O *PaNDa+* também possui parâmetros capazes de lidar com limites de tolerância de ruído, melhorando assim a precisão de cada padrão minerado [7].

O algoritmo do *PaNDa+* foi selecionado para os experimentos finais pois ele possui propriedades similares aos algoritmos de *fuzzy clustering* e *co-clustering*. A sua similaridade em relação aos algoritmos de *fuzzy clustering* é dada pela sua característica de atribuir o mesmo objeto para *clusters* distintos. Por outro lado, a sua similaridade em relação aos algoritmos de *co-clustering* é dada pela sua característica de descrever tantos os objetos que fazem parte de um *cluster* quanto os atributos que caracterizam aquele mesmo *cluster*.

### 4.4.1 PaNDa+ exemplo

Considere este exemplo prático executado em um o *dataset* artificial (figura 13), no qual o *PaNDa+* foi executado. Este *dataset* é o mesmo da figura 4, porém, o da figura 13 é o formato de entrada necessário para o *PaNDa+,* isto é, formato *FIMI* (*Frequent Itemset Mining Implementations Repository*) [14]. Esse *dataset* possui 6 linhas e 7 colunas. Cada linha representa uma transação qualquer e cada coluna representa um item qualquer. Por exemplo, a transação 0 (linha 0) possui os itens (colunas) 0, 1 e 2; a transação 5 possui todos os itens do *dataset*.

```
0, 1, 2
0, 1, 2
0, 1, 2
0, 1, 3, 4, 5, 6
0, 3, 4, 5, 6
0, 1, 2, 3, 4, 5, 6
```

*Figura 13 Dataset artificial*

O algoritmo pode ser encontrado em maior detalhe em [7]. O *PaNDa+* possui dez parâmetros. Entretanto, para não entrar em muitos detalhes, cita-se apenas cinco parâmetros:

- *d*: *dataset* (obrigatório)
  - O formato do *dataset* é o formato usual formato ascii da competição *FIMI*.

- *p*: int
  - Quantidade de padrões gerados
- *c*: cost
  - Pode-se escolher entre cinco funções de custo.
- *y*: float
  - relação de tolerância de linha.
- *t*: float
  - relação de tolerância de coluna.

Foram utilizados os parâmetros *data* sendo o *dataset* da figura 13; *p* = infinito; *c* = 1 (norm 1); *y* = 0.1 e *t* = 0.1. Ao rodar o algoritmo, gera-se um arquivo com o resultado, que pode ser visualizado na figura 14 abaixo. Ao analisar a figura 14, percebe-se que dois *clusters* foram formados (duas linhas). O primeiro *cluster* é representado pelos itens 0, 1 e 2, e possui 4 transações, sendo elas a 0, 1, 2 e 5. O segundo *cluster* é representado pelos itens 6, 5, 4, 3 e 0, e possui 3 transações, sendo elas a 3, 4 e 5. Saber quais são os itens que distinguem os *clusters* é uma propriedade dos algoritmos de *co-clustering*.

```
0 1 2 (4) [0 1 2 5]
6 5 4 3 0 (3) [3 4 5]
```

*Figura 14 Resultado do PaNDa+*

Observa-se que os *clusters* gerados correspondem com o *dataset*, pois os itens 0, 1 e 2 pertencem às transações 0, 1, 2 e 5. Além disso, os itens 6, 5, 4, 3 e 0 pertencem às transações 3, 4 e 5. Também se nota que a transação 5 pertence a ambos os *clusters*, pois divide características em comum com ambos os *clusters*. Isto é, possui os itens 0, 1 e 2 para pertencer ao *cluster* 1, e possui os itens 3, 4, 5 e 6 para pertencer ao *cluster* 2. Uma transação pertencer a dois *clusters* simultaneamente é uma característica dos algoritmos de *fuzzy clustering*.

# 5 Desenvolvimento

O desenvolvimento deste presente projeto se baseia na utilização de algoritmos de *clustering*, em dados de trajetórias multiaspecto, que possibilite agrupar trajetórias que possuam propriedades semelhantes. Para isso, os algoritmos de *FCM*, *Spectral Coclustering* e *PaNDa+* explicados na seção 2 foram executados em porções dos *datasets* apresentados na seção 3.

## 5.1 Preparação dos datasets

Os *datasets* não foram usados na íntegra apenas para poder-se visualizar os resultados obtidos com maior clareza. Portanto, utilizou-se apenas com o atributo *POI* de cada *dataset*, descartando-se os demais atributos. Além disso, selecionou-se apenas 10 *labels* (usuários) de cada *dataset*. Posteriormente, com a evolução do estudo, os *datasets* voltariam a ser utilizados em sua totalidade. A fim de simplicidade, apenas a alteração realizada no *dataset* do Foursquare será abordada. Ao selecionar-se apenas 10 *labels*, o *dataset* ficou com apenas 142 *TIDs* distintos, 854 *POIs* distintos e 2416 *check-ins* (linhas). As 5 primeiras linhas do Foursquare *dataset* alterado pode ser visto na figura 15 abaixo.

| | tid | poi | label |
|---|---|---|---|
| **0** | 126 | The Lair Of Modern Strange Cowboy | 6 |
| **1** | 126 | Galaxy Gourmet Deli | 6 |
| **2** | 126 | The Lair Of Modern Strange Cowboy | 6 |
| 3 | 126 | Popeyes Louisiana Kitchen | 6 |
| **4** | 126 | MTA Bus Operations Depot - LaGuardia | 6 |

*Figura 15 Primeiras 5 linhas Foursquare modificado*

Em seguida, também para auxiliar na visualização dos resultados, mapeou-se os *TIDs* para começar do 0 e subir gradativamente de um em um. Portanto, por exemplo, o *TID* 126 tornou-se 0, o próximo *TID*, 127, tornou-se 1 e assim por diante (figura 16).

| | tid | poi | label |
|---|---|---|---|
| **0** | 0 | The Lair Of Modern Strange Cowboy | 6 |
| **1** | 0 | Galaxy Gourmet Deli | 6 |
| **2** | 0 | The Lair Of Modern Strange Cowboy | 6 |
| **3** | 0 | Popeyes Louisiana Kitchen | 6 |
| **4** | 0 | MTA Bus Operations Depot - LaGuardia | 6 |

*Figura 16 Primeiras 5 linhas Foursquare dataset após mapeamento dos TIDs*

Após o mapeamento dos *TIDs* foi realizado o mapeamento dos *pois* em ordem alfabética (figura 17). Portanto, por exemplo, o *POI* "#Spottheshuttle - Enterprise NYC" tornou-se 1, o *POI* "AEO & Aerie Store" tornou-se 25, o *POI* "Zero Otto Nove" tornou-se 851 e assim por diante (figura 18).

```
#Spottheshuttle - Enterprise NYC : 1
'Essen : 2
1 Republik : 3
11th Street Bar : 4
135 St Broadway Hamilton Heights : 5
16 Handles : 6
169 Bar : 7
2 Hudson Place : 8
201 Bar and Restaurant : 9
24 Th Street Park : 10
295 Madison Avenue : 11
3.1 Phillip Lim : 12
3315 Pleasant Avenue : 13
417 Fifth Avenue : 14
5 & Diamond : 15
51 Newark Street Offices : 16
61 West 74th Street : 17
7-Eleven : 18
72nd Street Bagel : 19
77 Hudson Condominium : 20
82Mercer : 21
86 Wong Chinese Restaurant : 22
9/11 Tribute Center : 23
9th Street PATH Station : 24
AEO & Aerie Store : 25
AMC Clifton Commons 16 : 26
AMC Loews 34th Street 14 : 27
AMC Loews Lincoln Square 13 : 28
AMC Loews Newport Centre 11 : 29
Abitino's Pizzeria : 30
```

*Figura 17 Primeiros 30 pois mapeados*

| | tid | poi | label |
|---|---|---|---|
| **0** | 0 | 753 | 6 |
| **1** | 0 | 258 | 6 |
| **2** | 0 | 753 | 6 |
| 3 | 0 | 629 | 6 |
| **4** | 0 | 431 | 6 |

*Figura 18 Primeiras 5 linhas Foursquare dataset após mapeamento dos pois*

Posteriormente o mapeamento dos *TIDs* foi realizado o mapeamento dos *pois* em ordem alfabética (figura 17). Portanto, por exemplo, o *POI* "#Spottheshuttle - Enterprise NYC" tornou-se 1, o *POI* "AEO & Aerie Store" tornou-se 25, o *POI* "Zero Otto Nove" tornou-se 851 e assim por diante (figura 18).

Para estimar-se a qualidade dos *clusters* gerados pelos algoritmos de *clustering*, optou-se por usar os *TIDs* como *ground truth*. Pensa-se que uma pessoa possui uma trajetória mais similar com sua própria trajetória do que em relação a outras trajetórias. Ou seja, o algoritmo de *clustering* deveria agrupar todas as trajetórias que possuem a mesma *label* em um único *cluster*. Portanto, a coluna *label* foi removida. Porém ainda precisa-se ter uma correspondência entre quais *TIDs* pertencem à quais *labels*. Para isso, montou-se a figura 19 abaixo, no qual as 10 *labels* aparecem seguidas por seus respectivos *TIDs*.

```
6: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
7: [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]
12: [25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36]
14: [37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52]
19: [53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63]
25: [64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79]
34: [80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103]
50: [104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114]
56: [115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126]
65: [127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141]
```

*Figura 19 Labels e seus respectivos TIDs*

Além disso, para o *dataset* servir como input para os algoritmos de *clustering*, cada *POI* virou uma coluna e cada *check-in* continuou sendo uma linha (figura 20). Por exemplo, se houvesse um 1 na coluna 850, linha 2, isso significaria que o *check-in* 2 possui o *POI* 850. O *dataset* continua possuindo 2416 linhas (*check-ins)* e 855 colunas (854 *POIs* distintos, mais a coluna do *TID*).

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... | 846 | 847 | 848 | 849 | 850 | 851 | 852 | 853 | 854 | tid |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Figura 20 Primeiras 5 linhas, POIs como coluna, linhas como check-ins*

Em seguida, para melhorar a compreensão do resultado, comprime-se todos os *check-ins* que possuem o mesmo *TID* em uma única linha. Além disso, elimina-se a frequência dos *check-ins*. Isto é, por exemplo, se o usuário *label* 6 fez dois *check-ins* no *POI* 50 e um *check-in* no *POI* 100, na mesma semana (no mesmo *TID*) então, ao invés de ter uma linha mostrando os dois *check-ins* no *POI* 50 e outra linha mostrando um *check-in* no *POI* 100 (no mesmo *TID*), haverá agora apenas uma linha representativa do *TID* que comprime essas duas linhas, tendo o valor 1 em ambos os *POIs* 50 e 100. Com isso, passa-se a ter 142 linhas (*TIDs* distintos) e continuamos com 855 colunas (854 *POIs* distintos, mais a coluna do *TID*). Assim, gera-se a figura 21 abaixo, que representa a 5 primeiras linhas do *dataset* modificado. Com isso, tem-se os dados que servirá como input para os algoritmos de *clustering*.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ... | 846 | 847 | 848 | 849 | 850 | 851 | 852 | 853 | 854 | tid |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |

*Figura 21 Primeiras 5 linhas do dataset com os TIDs comprimidos*

## 5.2 Resultados fuzzy clustering

Para rodar o treinamento do algoritmo *FCM*, foram utilizados os parâmetros *data* sendo o *dataset* completo da figura 21 transposta e sem o atributo (coluna) *class*; *c* = 10; *m* = 1.1 (pois foi o que obteve o melhor *fuzzy partition coeficiente*); *error* = 0.005; *maxiter* = 1000 e *seed* = 0. Em relação ao resultado gerado, por não se usar um tipo de *co-clustering*, não sabemos quais são os atributos que caracterizam cada *cluster* formado. Além disso, o resultado gerado possui muita informação, que é a principal desvantagem do *FCM* e acaba prejudicando sua interpretação. Como mostra a figura 7, o resultado principal do *FCM* ao rodar o *dataset* da figura 20 será uma matriz de 10 (número de *clusters*) linhas por 142 colunas (número de *TIDs* distintos). Além disso, cada valor da matriz é um grau de pertencimento daquele *TID* para aquele *cluster*. Para tornar na

interpretação do resultado viável, transforma-se o *soft-clustering* (*FCM*) em um *hard-clustering* (como algoritmos tradicionais de *clustering*). Faz-se isso colocando o *TID* no *cluster* que possuía o maior grau de pertencimento. O resultado final, e interpretável, pode ser visualizado na figura 22 abaixo.

```
cluster 0: ['7', '7', '7', '7', '7', '7', '7', '7', '7', '7', '7', '7']
cluster 1: ['65', '65', '65', '65', '65', '65', '65', '65', '65', '65', '65', '65', '65', '65', '65']
cluster 2: ['34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34', '34',
'34', '34', '34', '34', '34']
cluster 3: ['6', '12', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '14', '19']
cluster 4: ['25', '25', '25', '25', '25', '25', '25', '25', '25', '25', '25', '25', '25', '25', '25', '25']
cluster 5: ['19', '19', '19', '19', '19', '19', '19', '19', '19', '19']
cluster 6: ['6', '6', '6', '6', '6', '6', '6', '6', '6', '6', '6', '6']
cluster 7: ['12', '12', '12', '12', '12', '12', '12', '12', '12', '12', '12', '12']
cluster 8: ['50', '50', '50', '50', '50', '50', '50', '50', '50', '50', '50']
cluster 9: ['56', '56', '56', '56', '56', '56', '56', '56', '56', '56', '56', '56']
```

*Figura 22 Resultado final após hard-clustering do algoritmo FCM*

Na imagem 22 acima pode-se ver os 10 *clusters* gerados e quais foram as *labels* que foram agrupadas. Por exemplo, o *cluster* 0 agrupou apenas *TIDs* que possuíam a *label 7*. Também se observa que que a *label* 7 aparece apenas no *cluster* 0. Portanto, esse *cluster* agrupou perfeitamente a *label 7*. Entretanto, ao analisar o *cluster* 3, percebe-se que ele colocou em seu grupo não apenas a *label* 14, mas também a *label* 6, 12 e 19. Sendo assim, esse *cluster* não conseguiu agrupar perfeitamente a *label* 14.

Após o *hard-clustering* consegue-se facilmente resultados numéricos que representam a qualidade dos *clusters* gerados. A figura 23 abaixo mostra os resultados de homogeneidade, completude e *v measure* dos *clusters* gerados pelo algoritmo *FCM* utilizando os três *dataset* citados, tanto suas versões minimizadas (com 10 *labels*) como o *dataset* completo (com todas as *labels* e apenas com a coluna *POI*). Utilizando o parâmetro *c* sendo igual ao número de *labels* de cada *dataset*. Além disso, pode-se ver qual foi o parâmetro *m* que gerou o melhor *fuzzy partition coeficiente*.

| | Homogeneity | Completeness | V Measure |
|---|---|---|---|
| Fuzzy-Clustering (10) | | | |
| Foursquare ($m$ = 1.1) | 0,968 | 0,964 | 0,966 |
| Brightkite ($m$ = 1.1) | 0,863 | 0,868 | 0,865 |
| Gowalla ($m$ = 1.1) | 0,856 | 0,494 | 0,627 |
| Fuzzy-Clustering (All) | | | |
| Foursquare ($m$ = 1.1) | 0,544 | 0,200 | 0,290 |
| Brightkite ($m$ = 1.1) | 0,625 | 0,028 | 0,053 |
| Gowalla ($m$ = 1.5) | 0,804 | 0,732 | 0,766 |

*Figura 23 Results FCM*

## 5.3 Resultados co-clustering

Para rodar o algoritmo *Spectral Coclustering*, foram usados os parâmetros *c* = 10 e *random state* = 10. Em seguida, foi feito utilizado o método *fit* usando como parâmetro o

*dataset* completo da figura 20 e sem o atributo (coluna) *class*. O resultado gerado pelo *Spectral Coclustering* é facilmente manipulável visualizável. A figura 24 abaixo representa dois *co-clusters* dos dez *clusters* (parâmetro *c*) formados pelo algoritmo. Além disso, por ser um algoritmo de *co-clustering* sabe-se quais *POIs* o seu respectivo conjunto de *TIDs*. Contudo, cada *TID* pode pertencer a apenas um *cluster*.

```
TIDs:
['80', '81', '82', '83', '84', '85', '86', '87', '88', '89', '90', '91', '92', '93', '94', '95', '96', '97', '98', '99', '100',
'101', '102', '103', '122', '126']
São melhores descritos pelos POIs:
['12', '21', '34', '41', '42', '54', '66', '72', '81', '83', '87', '88', '105', '108', '113', '114', '125', '127', '139', '14
3', '150', '155', '160', '164', '167', '175', '177', '203', '213', '235', '259', '265', '277', '278', '284', '288', '298', '29
9', '302', '305', '323', '336', '337', '350', '360', '366', '378', '395', '507', '518', '525', '526', '528', '532', '537', '56
7', '579', '581', '588', '593', '600', '601', '616', '623', '625', '644', '645', '648', '652', '668', '670', '674', '676', '67
9', '691', '698', '700', '701', '702', '703', '707', '710', '718', '726', '728', '729', '733', '741', '758', '761', '773', '78
2', '783', '784', '786', '797', '802', '803', '805', '807', '811', '816', '818', '831', '837', '849']

TIDs:
['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12']
São melhores descritos pelos POIs:
['5', '97', '106', '107', '121', '186', '245', '249', '254', '258', '264', '281', '335', '346', '353', '355', '365', '372', '37
3', '375', '404', '405', '406', '407', '408', '409', '411', '412', '413', '414', '415', '416', '417', '418', '419', '420', '42
1', '422', '424', '425', '427', '428', '429', '430', '431', '432', '433', '434', '435', '436', '437', '438', '439', '440', '44
1', '442', '443', '444', '445', '446', '447', '448', '449', '450', '452', '453', '454', '455', '456', '458', '459', '460', '46
1', '462', '463', '465', '467', '470', '471', '472', '474', '475', '476', '477', '478', '479', '481', '482', '484', '486', '48
7', '488', '490', '494', '506', '517', '550', '552', '568', '571', '578', '628', '629', '636', '639', '641', '658', '678', '70
8', '750', '753', '794', '845', '854']
```

*Figura 24 Dois dos dez co-clusters formados pelo algoritmo Spectral Coclustering*

Ao analisar o primeiro *co-cluster* da figura 24 acima, percebe-se (com base na figura 19), que ele agrupou quase que perfeitamente apenas os *TIDs* da *label* 6. Entretanto, ele também colocou junto ao *cluster*, duas *TIDs* que pertencem a *label* 56, prejudicando a qualidade do *cluster*. Por outro lado, ao analisar-se o segundo *co-cluster* da figura 24, percebe-se que ele agrupou perfeitamente a *label* 6 em um único *cluster*. Os resultados numéricos gerados pelo *Spectral Coclustering* podem ser visualizados na figura 25 abaixo. Foi utilizado o parâmetro *c* sendo igual ao número de *labels* de cada *dataset*.

| | Homogeneity | Completeness | V Measure |
|---|---|---|---|
| Co-clustering (10) | | | |
| Foursquare | 0,940 | 0,899 | 0,919 |
| Brightkite | 0,942 | 0,920 | 0,931 |
| Gowalla | 0,808 | 0,666 | 0,730 |
| Co-clustering (All) | | | |
| Foursquare | 0,850 | 0,714 | 0,776 |
| Brightkite | 0,927 | 0,838 | 0,880 |
| Gowalla | 0,908 | 0,791 | 0,845 |

*Figura 25 Resultados Spectral Coclustering*

## 5.4 Resultados PaNDa+

Para rodar o algoritmo *PaNDa+*, foram utilizados os parâmetros *dataset* sendo o *dataset* completo da figura 21 sem o atributo *class*; *p* = infinito; *c* = 1 (norm 1); *y* = 0.01 e

*t* = 0.01. O resultado ao rodar o algoritmo pode ser visto na figura 26 abaixo. As cores representam *TIDs* que pertencem a mesma *label*.

| Cluster | POIs | Transactions |
|---|---|---|
| 1 | 305 707 567 702 | [80 83 101 102 103] |
| 2 | 588 497 142 | [124 125] |
| 3 | 588 305 567 | [80 82 83 84 85 87 90 91 92 93 101] |
| 4 | 707 656 341 638 273 | [26 27 28 29 30 31 33 34] |
| 5 | 497 681 | [39 40 41 42 43 45 46 47] |
| 6 | 531 853 630 | [127 128 129 133 134 135 136 137 138 139 140] |
| 7 | 714 229 130 132 683 | [68 69 71 74 75 76 78] |

*Figura 26 Resultado PaNDa+*

Nota-se na figura 26 que 7 *clusters* foram gerados. Pode-se saber quais são os *POIs* representativos de cada *cluster* (característica de *co-clustering*), e há *TIDs* pertencendo a mais de um *cluster* (característica de *fuzzy clustering*). Percebe-se que as *labels* 6, 7, 19 e 50 não geraram *clusters*. Além disso, todos os *clusters* possuem os *TIDs* da mesma *label*, ou seja, são *clusters* puros. Entretanto, nenhum *cluster* possui todas as *TIDs* da sua *label*, ou seja, os *clusters* são incompletos. Nota-se que a *label* 34 está contida tanto no *cluster* 1 quanto no *cluster* 3. Dos 25 *POIs* incluídos no conjunto de todos os *clusters*, 5 se repetem (305, 607, 567, 588, 497). Apenas alguns poucos *POIs* foram selecionados como sendo representativos dos *clusters* gerados, enquanto que no *Spectral Clustering* todos os *POIs* foram colocados em algum *cluster* (sem repetição).

A figura 27 abaixo mostra todas as *labels*, mostrando seus 6 *POIs* mais frequentes e a quantidade total daquele *POI* no *dataset* da figura 21. As células destacadas em azul representam os *POIs* que estão presentes nos *clusters* gerados. Todos os *POIs* que estão incluídos nos *clusters* podem ser visualizados na figura 27, isto é, nenhum *POI* que não seja um dos seis mais frequentes foi incluído em algum *clusters*. Ao se observar a figura 27, nota-se, por exemplo, que a *label* 12 agrupou os 5 *POIs* mais frequentes de seus *TIDs* (341, 638, 273, 656 e 707) que são também os mais frequentes em relação a todo o *dataset*. O *POI* 707 apesar de se repetir 9 vezes nos *TIDs* da *label* 12, ele também se repete 8 vezes nos *TIDs* do *label* 34, o *POI* 707 aparece tanto no *cluster* que caracteriza o *label 12* quanto o *label* 34. Além disso, percebe-se que o *POI* 497 repete-se 16 vezes nos *TIDs* da *label* 14 e apenas 3 vezes nas *TIDs* da *label* 56, mas, mesmo assim, o *POI* 497 foi incluído tanto nos *clusters* representativos da *label* 14 quanto no da *label* 56. Por fim, nota-se, por exemplo, que as *labels* 6, 7, 19 e 50 possuíam *POIs* frequentes unicamente em seus *TIDs*, porém, mesmo assim, acabaram não gerando nenhum *cluster* que as representassem.

| Label | POI | Label POI # | POI # |
|---|---|---|---|
| 6 | 568 | 10 | 10 |
| 6 | 753 | 10 | 10 |
| 6 | 486 | 7 | 9 |
| 6 | 750 | 7 | 7 |
| 6 | 477 | 5 | 5 |
| 6 | 452 | 5 | 5 |
| 7 | 333 | 11 | 11 |
| 7 | 134 | 8 | 8 |
| 7 | 626 | 9 | 9 |
| 7 | 382 | 7 | 7 |
| 7 | 282 | 5 | 5 |
| 7 | 522 | 5 | 5 |
| 12 | 341 | 11 | 11 |
| 12 | 638 | 10 | 10 |
| 12 | 273 | 10 | 10 |
| 12 | 656 | 11 | 15 |
| 12 | 707 | 9 | 23 |
| 12 | 137 | 6 | 6 |
| 14 | 497 | 16 | 20 |
| 14 | 122 | 8 | 8 |
| 14 | 681 | 8 | 12 |
| 14 | 209 | 5 | 5 |
| 14 | 570 | 4 | 4 |
| 14 | 677 | 3 | 3 |
| 19 | 660 | 9 | 9 |
| 19 | 796 | 6 | 7 |
| 19 | 655 | 5 | 5 |
| 19 | 822 | 5 | 5 |
| 19 | 599 | 5 | 5 |
| 19 | 158 | 2 | 2 |
| 25 | 130 | 13 | 13 |
| 25 | 229 | 14 | 14 |
| 25 | 132 | 12 | 12 |
| 25 | 683 | 11 | 11 |
| 25 | 714 | 13 | 14 |
| 25 | 17 | 6 | 6 |
| 34 | 305 | 24 | 24 |
| 34 | 567 | 21 | 21 |
| 34 | 588 | 13 | 23 |
| 34 | 702 | 13 | 13 |
| 34 | 203 | 10 | 14 |
| 34 | 707 | 8 | 23 |
| 50 | 752 | 8 | 8 |
| 50 | 554 | 6 | 6 |
| 50 | 205 | 7 | 7 |
| 50 | 464 | 7 | 7 |
| 50 | 643 | 6 | 8 |
| 50 | 561 | 5 | 5 |
| 56 | 588 | 10 | 23 |
| 56 | 14 | 7 | 7 |
| 56 | 142 | 8 | 8 |
| 56 | 587 | 6 | 6 |
| 56 | 497 | 3 | 20 |
| 56 | 112 | 3 | 3 |
| 65 | 853 | 15 | 15 |
| 65 | 531 | 15 | 16 |
| 65 | 630 | 11 | 11 |
| 65 | 338 | 7 | 7 |
| 65 | 549 | 7 | 7 |
| 65 | 842 | 5 | 5 |

*Figura 27 Quantidade de POIs que cada label possui e quantidade do mesmo POI para o dataset inteiro*

Pelo fato de um *POI* e uma *TID* poderem pertencer a mais de um *cluster* simultaneamente, não se encontrou uma medida numérica para caracterizar a qualidade dos *clusters* gerados. Pode-se ver quantos *clusters* foram gerados pelos diferentes *datasets* utilizando diferentes parâmetros *y* e *t* na figura 28 abaixo.

| | $y = 0.1, t = 0.1$ | $y = 0.5, t = 0.5$ | $y = 1.0, t = 1.0$ |
|---|---|---|---|
| Panda (10) | | | |
| Foursquare | 7 | 2 | 3 |
| Brightkite | 4 | 6 | 6 |
| Gowalla | 3 | 5 | 5 |
| Panda (All) | | | |
| Foursquare | 21 | 25 | 25 |
| Brightkite | 1 | 1 | 1 |
| Gowalla | 4 | 4 | 4 |

*Figura 28 Resultados PaNDa+*

# 6 Conclusão

Na literatura, dificilmente encontra-se algum trabalho que trabalhe tanto com *clustering* e trajetórias multiaspecto. Normalmente, os trabalhos que relacionam *clustering* e trajetórias são específicos para trajetórias brutas. Com isso, torna-se difícil encontrar *datasets* com um *ground truth* para métodos de *clustering* em trajetórias multiaspecto. Além disso, como há poucos trabalhos relacionados, torna-se difícil a comparação com outros algoritmos.

Avaliou-se os algoritmos de *fuzzy clustering*, *co-clustering* e o algoritmo *PaNDa+* em *datasets* resumidos de trajetórias multiaspecto. A princípio, caso seja realmente o caso em que que uma pessoa possui uma trajetória mais similar com sua própria trajetória do que em relação a outras trajetórias, os resultados obtidos pelo *co-clustering* foram os mais positivos (figura 25). Por outro lado, os resultados obtidos com o *fuzzy clustering* não foram satisfatórios (figura 23). Entretanto, ainda pode-se tentar trabalhar melhor com os graus de pertencimento gerados pelo *fuzzy clustering*. No presente relatório, optou-se por tornar o *fuzzy clustering* em um *hard clustering*, devido ao grau de complexidade gerado pelo seu resultado. Além disso, os resultados do *PaNDa+* também não refletem o que se esperava, visto que, por exemplo, o *dataset* completo do *Foursquare* possui 193 *labels* distintas, mas apenas 25 *clusters* foram encontrados ao rodar *PaNDa+*. O resultado torna-se ainda mais discrepante ao analisar os outros *datasets* (figura 28). Porém, por conta da capacidade do *PaNDa+* de lidar com limites de tolerância de ruído, ele gerou resultados mais facilmente analisáveis, com poucos *clusters* e poucos *POIs* que descrevem tais *clusters*. Além disso, o *PaNDa+* é o único algoritmo de *clustering* aqui descrito que não necessita do número de *clusters* como parâmetro.

A continuidade da pesquisa seria em entender melhor o funcionamento interno do algoritmo *PaNDa+* para posteriormente alterá-lo de forma a melhorar a qualidade dos *clusters* gerados. Além disso, realizar-se-ia experimentos com outros parâmetros do *dataset*. O formato do input também poderia ser alterado a fim de considerar-se a frequência de um mesmo *POI* dentro de uma semana. Com isso poder-se-ia realizar um artigo para *cluster* de trajetórias multiaspecto.

## 6.1 Avaliação do aluno em relação ao PIBIT

A participação no Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação (PIBIT) promoveu um vasto aprendizado em relação ao processo de pesquisa acadêmica. Inicialmente realizou-se a pesquisa no campo de trajetórias e de *clustering* através de vários artigos científicos a fim de inteirar-se sobre a área de estudo. Em seguida, precisou-se encontrar algoritmos e aprender a usá-los corretamente para realizar os experimentos. Necessitou-se entender e ser capaz de manipular *datasets* complexos. Por fim, houve a análise mais detalhada dos resultados obtidos pelos algoritmos utilizados. Concomitantemente a todo o processo, teve-se a prática contínua da programação como uma ferramenta para a produção de algoritmos próprios, treinamento de algoritmos de *machine learning*, manipulação, visualização e análise de

dados. Além disso, durante todo o processo, teve-se comunicação contínua não apenas com a orientadora, mas também com outros alunos de PIBIT, mestrado e doutorado que participavam do laboratório. Todo o conhecimento adquirido durante este um ano de PIBIT será profundamente valioso para o meu desenvolvimento acadêmico e profissional.

# 7 Referências


[1] FERRERO, Carlos Andres; ALVARES, Luis Otavio; BOGORNY, Vania. Multiple aspect trajectory data analysis: research challenges and opportunities. In: **GeoInfo**. 2016. p. 56-67.

[2] PETRY, Lucas May et al. Towards semantic-aware multiple-aspect trajectory similarity measuring. **Transactions in GIS**.

[3] LEE, Jae-Gil; HAN, Jiawei; WHANG, Kyu-Young. Trajectory clustering: a partition-and-group framework. In: **Proceedings of the 2007 ACM SIGMOD international conference on Management of data**. ACM, 2007. p. 593-604.

[4] DHILLON, Inderjit S.; MALLELA, Subramanyam; MODHA, Dharmendra S. Information-theoretic co-clustering. In: **Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining**. ACM, 2003. p. 89-98.

[5] PELEKIS, Nikos et al. Clustering uncertain trajectories. **Knowledge and Information Systems**, v. 28, n. 1, p. 117-147, 2011.

[6] LIU, Yongli et al. A fuzzy co-clustering algorithm for biomedical data. **PloS one**, v. 12, n. 4, p. e0176536, 2017.

[7] LUCCHESE, Claudio; ORLANDO, Salvatore; PEREGO, Raffaele. A Unifying Framework for Mining Approximate Top-$ k $ Binary Patterns. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 12, p. 2900-2913, 2013.

[8] WIERZCHOŃ, Sławomir T.; KŁOPOTEK, Mieczysław. **Modern algorithms of cluster analysis**. Springer, 2018.SCIKIT-FUZZY DEVELOPMENT TEAM.

[9] NCSS STATISTICAL SOFTWARE. **Fuzzy Clustering.** Disponível em: <https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Fuzzy_Clustering.pdf>. Acesso em: 17 ago. 2019

[10] **Skfuzzy 0.2 docs.** Disponível em: <https://pythonhosted.org/scikit-fuzzy/>. Acesso em: 17 ago. 2019.

[11] FORSYTH, Richard. **Machine Learning Repository:** Zoo Data Set. Disponível em: <https://archive.ics.uci.edu/ml/datasets/zoo>. Acesso em: 17 ago. 2019.

[12] SCIKIT-LEARN DEVELOPERS. **Scikit Learn:** Spectral Coclustering. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.bicluster.SpectralCoclustering.html>. Acesso em: 17 ago. 2019.

[13] OH, Chi-Hyon; HONDA, Katsuhiro; ICHIHASHI, Hidetomo. Fuzzy clustering for categorical multivariate data. In: **Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)**. IEEE, 2001. p. 2154-2159.

[14] LUCCHESE, Claudio. **PaNDa+:** A unifying framework for mining approximate top-k binary patterns. Disponível em: <https://claudio-lucchese.github.io/archives/20131113/index.html>. Acesso em: 17 ago. 2019.