# Uma Revisão Sobre Combinação de Agrupamentos

Murilo C. Naldi [1]
Katti Faceli [2], André C. P. L. F. Carvalho [1]

**Resumo:** Vários algoritmos de agrupamentos foram propostos na literatura. O uso de diferentes algoritmos de agrupamento, ou até mesmo de um único algoritmo, pode obter diferentes resultados quando aplicados em um mesmo conjunto de dados. A combinação de resultados, obtidos de uma técnica de classificação ou de técnicas distintas, é utilizada com sucesso para melhorar a estabilidade ou desempenho dessas técnicas. Por isto, nos últimos anos houve um aumento crescente no interesse do uso de combinação de agrupamentos de dados. Neste trabalho, é feita uma revisão sobre os principais métodos de combinação de agrupamentos encontrados na literatura. Para isso, a revisão começa com uma descrição do problema de combinação e uma análise dos objetivos comumente adotados por métodos de combinação. Em seguida, discorre-se sobre a necessidade da diversidade nos agrupamentos a serem combinados e métodos para medi-la. Também é definido um critério para medir a informação mútua entre agrupamentos e são apresentados exemplos de seu uso. O desempenho dos métodos foi comparado por vários autores na literatura e uma análise dessas comparações é realizada neste trabalho.

**Abstract:** Several clustering algorithms have been proposed in the literature. Different algorithms, or even the same algorithm, may obtain different partitions when applied to the same dataset. Ensemble or combination of output, provided by distinct or the same classification technique have been successfully used in the literature to improve the stability and quality of these techniques. For this reason, there has been an increasing interest on the usage of cluster ensembles over the past few years. This work is a review of cluster ensemble methods of importance in literature, beginning with a description of the ensemble problem and the analyses of the most used objectives. Following, arguments about the need of diversity in the clusters to ensemble and methods to measure it are presented. Also, a criterion to measure the mutual information between clusters is defined and examples of its usage are presented. The performance of ensemble methods was compared by various authors and their results are discussed here.

[1]Instituto de Ciências Matemáticas e de Computação, ICMC, USP, Caixa Postal 668
{murilocn, andre@icmc.usp.br}
[2]Universidade Federal de São Carlos, Ufscar, Campus Sorocaba
{katti@ufscar.br}

# 1 Introdução

Agrupamento de dados consiste na classificação não-supervisionada de objetos semelhantes em grupos. Sua utilização é bastante comum em casos onde não há conhecimento prévio sobre classes ou valores associados aos dados, como em vários problemas nas áreas de: segmentação de mercados [1, 2]; bioinformática [3, 4]; mineração de textos [5, 6]; detecção de intrusão [7]; processamento de imagens [8]; taxonomia [9].

Para lidar com esses problemas, diversos algoritmos de agrupamento foram propostos na literatura [10, 11]. Esses algoritmos podem ser classificados em duas categorias principais: particionais e hierárquicos. Algoritmo particionais geram partições de dados, enquanto algoritmos hierárquicos diferem dos particionais por gerar uma seqüência de partições aninhadas hierarquicamente. Considerando uma base de dados $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ cujos elementos podem ser representados por vetores com $n$ características ou atributos, uma partição exclusiva é a coleção $\pi = \{\mathbf{C}_1^\pi, \mathbf{C}_2^\pi, ..., \mathbf{C}_k^\pi\}$ de $k$ grupos ou subconjuntos $\mathbf{C}_i^\pi$ tais que $\mathbf{C}_1^\pi \cup \mathbf{C}_2^\pi \cup ... \cup \mathbf{C}_k^\pi = \mathbf{X}$, $\mathbf{C}_i^\pi \neq \emptyset$ e $\mathbf{C}_i^\pi \cap \mathbf{C}_l^\pi = \emptyset$ para $i \neq l$. Partições não exclusivas permitem a participação ou pertinência dos objetos a grupos distintos. Este trabalho dá ênfase no uso de algoritmos de agrupamento particionais exclusivos.

Cada algoritmo de agrupamento possui suas próprias características, o que gera uma grande variedade de soluções possíveis para um determinado conjunto de dados. Alguns algoritmos possuem parâmetros livres que influenciam na partição obtida, o que gera uma diversidade ainda maior de soluções. Dependendo da aplicação, existe algum conhecimento prévio sobre o conjunto de dados que pode ser utilizado na análise. Também pode não ser possível agrupar toda a base de dados de uma única vez. Nesses casos, é importante que um agrupamento final seja gerado a partir de agrupamentos obtidos separadamente. A combinação de agrupamentos, também conhecida como "*cluster ensemble*", consiste em combinar agrupamentos em comitês para obter um único agrupamento que possa aproveitar as características de cada um [12, 13].

Uma definição simplificada de combinação de algoritmos de agrupamento é apresentada em [14]: dado um conjunto de $n_\pi$ partições base, $\Pi = \{\pi_1, \pi_2, ..., \pi_{n_\pi}\}$, resultantes de várias aplicações de um ou mais algoritmos de agrupamento em um determinado conjunto de dados $\mathbf{X}$, encontrar uma única partição consenso $\pi_f$. A combinação pode ter diferentes objetivos, dentre os quais podem ser citados [15]:

- **Robustez**: obter uma partição consenso cuja qualidade seja melhor do que a maioria das partições que a originaram, segundo algum critério, para diversos domínios e conjuntos de dados, se possível sem a necessidade de ajustes feitos pelo usuário [16, 17, 18, 19, 20]. Também é possível obter uma partição consenso com menor sensibilidade a ruídos, *outliers* ou variações de amostragem [21, 17, 19, 22].

- **Novidade**: gerar uma partição consenso inédita, inspirada nos grupos das partições base, que não poderia ser obtida por nenhum dos algoritmos utilizados individualmente [14, 23].
- **Reaproveitamento de conhecimento**: utilizar conhecimento prévio juntamente com novas partições para construir uma partição final. Em alguns domínios de aplicação, partições do conjuntos de dados já são conhecidas, oriundas da análise prévia dos dados ou resultados obtidos de regras de agrupamento pré-definidas. Esse conhecimento prévio sobre a estrutura do conjunto analisado pode ser utilizado para a construção da partição consenso [24, 16, 25, 17].
- **Computação distribuída**: gerar partições por meio de métodos distribuídos e combinar o resultado em uma única partição consenso. Esse tipo de combinação geralmente é aplicado quando, por algum motivo, não é possível executar o agrupamento de todo o conjunto de dados em uma única máquina ou é desejável a geração paralela dos agrupamentos. Segundo [13], dois subdomínios importantes da área de agrupamento distribuído são o agrupamento de atributos distribuídos e o agrupamento de objetos distribuídos. Em cada subdomínio, diferentes subconjuntos do conjunto de dados são gerados, contendo parte dos atributos ou dos objetos do conjunto de dados. A aplicação de um algoritmo de agrupamento a esses subconjuntos resultam nas partições base. A mesma estratégia pode ser utilizada para gerar paralelização e escalabilidade nos algoritmos [16, 17, 26].
- **Consistência**: obter uma partição consenso tal que, de alguma forma, esteja em concordância com as partições base. Uma forma de medir a consistência da partição consenso é por meio da análise da informação mútua entre as partições base [27, 28, 29, 30, 19].
- **Desempenho e custo computacional**: com o objetivo de se obter um melhor desempenho computacional, algoritmos menos robustos e de baixo custo computacional podem ser executados e seus resultados combinados, de forma a obter uma partição consenso mais robusta que as partições base [14].

A combinação de estimadores é comumente empregada sob o paradigma de aprendizado supervisionado, isto é, em problemas de classificação e regressão [31, 32]. Porém, devido à falta de conhecimento *a priori* sobre como os dados são rotulados, muitas vezes a aplicação direta das técnicas de combinação de estimadores supervisionados não é possível no paradigma não-supervisionado [32]. Além disso, técnicas de agrupamento diferentes podem produzir grupos incompatíveis, o que resulta em problemas de correspondência entre os rótulos de elevada complexidade computacional ou, algumas vezes, intratáveis [14].

Por esses motivos, a combinação de partições requer novas técnicas ou a adaptação de técnicas já utilizadas no paradigma supervisionado. Para isso, é preciso considerar dois aspectos importantes [17, 33, 34]:

- **Diversidade entre os agrupamentos**: para justificar o custo computacional do uso de múltiplos algoritmos de agrupamento em uma combinação, é preciso se certificar que os mesmos apresentem um certo grau de diversidade. Desta forma, caso haja um consenso entre as partições formadas, são maiores as chances que o agrupamento resultante se aproxime da estrutura natural dos dados e resulte em uma combinação mais robusta.

- **Função de consenso**: é preciso estabelecer um método para combinar as diferentes partições, de forma a garantir consenso simétrico e balanceado em relação a todos os componentes da partição e resolver problemas como formas geométricas distintas para representar diferentes grupos.

Devido à diversidade de objetivos, existem métodos distintos de combinação de partições. Nas seções 2 e 3 são apresentadas as abordagens mais freqüentes na literatura para gerar a diversidade necessária nas partições iniciais e encontrar a partição final. Na Seção 3.5, é feita uma análise dos resultados obtidos por trabalhos que comparam as abordagens descritas. As considerações finais desse estudo são feitas na Seção 4.

## 2 Diversidade dos agrupamentos base

Ao buscar uma partição consenso, é preciso ter muito claro o(s) objetivo(s) da combinação e conhecer bem os algoritmos a serem combinados, a fim de garantir que os algoritmos escolhidos possam contribuir para esse(s) objetivo(s). Assim, de acordo com o objetivo da combinação, deve-se escolher algoritmos de agrupamento, ou formas de aplicação de um algoritmo, que forneçam a diversidade necessária.

É preciso enfatizar que a diversidade necessária se refere à forma ou método utilizado para gerar as partições base, não as partições resultantes destas. A geração de partições semelhantes por métodos distintos pode refletir a presença de uma forte estrutura nos dados agrupados, identificada nestas partições.

Os métodos mais utilizadas para se obter essa diversidade são [13, 33, 34, 15]:

- Utilizar diferentes algoritmos convencionais de agrupamento para gerar as partições base [35, 36, 16, 37, 38, 39]. Neste caso, a combinação é dita heterogênea.

- Executar várias vezes um mesmo algoritmo convencional, com diferentes inicializações ou ordem de apresentação dos dados, de forma a se obter partições distintas [28, 29, 30, 40, 38, 17, 41, 34]. Neste caso a combinação é dita homogênea.

- Empregar algoritmos de agrupamento mais simples do que os convencionais, chamados de algoritmos de agrupamento fracos [14], com o objetivo de reduzir o custo computacional do processo.
- Utilizar diferentes conjuntos de dados, que podem ser formados por:
  - seleções de alguns atributos dos objetos do conjunto de dados [16].
  - projeções do conjunto original em um espaço de menor dimensão [18].
  - amostragem, referente a subconjuntos de objetos de um mesmo conjunto de dados [16, 21, 18, 34].
  - inserção de ruído ou perturbação para aumentar a diversidade no conjunto de dados [42, 34, 27, 43].

Foram propostas na literatura algumas funções para medir a diversidade das partições base. Dentre elas, podemos citar a medida não-pareada de entropia [44]. A diversidade quantificada por esta medida é dada pela Equação (1), em que $pp_{ij}$ representa a proporção de vezes em que o par composto pelo $i$-ésimo e o $j$-ésimo objetos do conjunto de dados estão inseridos no mesmo grupo, para todas as partições base geradas a partir de um conjunto de dados de $n_o$ objetos.

$$div_{ent} = \frac{2}{n_o(n_o-1)} \sum_{i=1}^{n_o} \sum_{j=i+1}^{n_o} -(pp_{ij} \log_2 pp_{ij} + (1-pp_{ij}) \log_2 (1-pp_{ij})) \qquad (1)$$

O índice de validação *Rand* ajustado (*AR*) [45] foi utilizado para medir a diversidade pareada entre partições [34] . Dado o par de objetos $\mathbf{x}_i$ e $\mathbf{x}_j$ pertencentes ao conjunto de dados $\mathbf{X}$, tal que $i \neq j$, e os grupos $C^l \in \pi_l$ e $C^m \in \pi_m$, existem quatro possíveis situações, apresentadas a seguir:

- 11: ($\mathbf{x}_i \in C^l$) e ($\mathbf{x}_j \in C^l$), ($\mathbf{x}_i \in C^m$) e ($\mathbf{x}_j \in C^m$)
- 01: ($\mathbf{x}_i \in C^l$) e ($\mathbf{x}_j \notin C^l$), ($\mathbf{x}_i \in C^m$) e ($\mathbf{x}_j \in C^m$)
- 10: ($\mathbf{x}_i \in C^l$) e ($\mathbf{x}_j \in C^l$), ($\mathbf{x}_i \in C^m$) e ($\mathbf{x}_j \notin C^m$)
- 00: ($\mathbf{x}_i \in C^l$) e ($\mathbf{x}_j \notin C^l$), ($\mathbf{x}_i \in C^m$) e ($\mathbf{x}_j \notin C^m$)

Seja $np_t$ o número de todas as possíveis combinações de pares de objetos do conjunto de dados ($np_t = n_o(n_o-1)/2$), $np_{11}$ o número de pares de objetos na situação 11, $np_{01}$ o

número de pares de objetos na situação 01, $np_{10}$ e $np_{00}$ os números de pares de objetos nas situações 10 e 00, respectivamente. O valor do índice *Rand* ajustado entre as partições $\pi_i$ e $\pi_j$, é dado por:

$$AR(\pi_i, \pi_j) = \frac{np_{11} - \frac{(np_{11}+np_{01})(np_{11}+np_{10})}{np_t}}{\frac{(np_{11}+np_{01})+(np_{11}+np_{10})}{2} - \frac{(np_{11}+np_{01})(np_{11}+np_{10})}{np_t}} \quad (2)$$

A partir do estudo desse índice, foram propostas quatro funções de diversidade não-pareadas que quantificam a diversidade individual de cada partição [34]. Seja $1 - AR(\pi_i, \pi_j)$ a diversidade entre $\pi_i$ e $\pi_j$, a primeira função, $D_1$, proposta em [34] calcula a média das diversidades entre todas as $n_\pi$ partições base $\Pi = \{\pi_1, \pi_2, ..., \pi_{n_\pi}\}$ e o agrupamento consenso $\pi_f$. De forma semelhante, a segunda função, $D_2$, mede o desvio padrão dessas diversidades. A terceira e quarta funções são derivadas das duas primeiras, sendo dadas por $D_3 = \frac{1}{2}(1 - D_1 + D_2)$ e $D_4 = \frac{D_2}{D_1}$, respectivamente. Essas medidas foram comparadas com a medida não-pareada de entropia (Equação (1)) e os resultados destacam a função $D_3$, cujos valores médios estão associados à boa qualidade das combinações de partições obtidas.

Em relação ao número de partições base a serem combinadas, grande parte dos pesquisadores concordam que, quanto maior o número, maior a quantidade de evidência gerada e, portanto, mais precisa será a combinação [46, 47, 48]. Segundo esses autores, é esperado que a maioria dos métodos de combinação dêem bons resultados para conjuntos com mais de 1000 partições [46]. Porém, alguns autores afirmam que a precisão da partição consenso aumenta pouco com o uso de milhares de partições, se essa for comparada a combinações obtidas com algumas centenas de partições base, o que não justificaria seu alto custo computacional [48].

Experimentos realizados em [34] concluem que quantidades excessivas de diversidade levam a combinações pobres e que combinações com níveis moderados de diversidade são mais precisas.

## 3 Determinação do agrupamento consenso

A função consenso constitui a essência da combinação, pois é nela que se concentra a metodologia pela qual as partições serão combinadas. Em [44], foi observado que, apesar de necessária, a diversidade entre os membros de uma combinação não é suficiente para gerar uma solução melhorada sem a escolha de um método de integração efetivo.

A escolha de uma função consenso apropriada pode ser uma tarefa árdua, em especial pela ausência de uma classificação intrínseca dos objetos a serem agrupados, o que faz com que não haja uma correspondência explícita entre os grupos das diversas partições base [14].

Partições com números de grupos distintos e o problema da correspondência entre os rótulos dos grupos agravam ainda mais a complexidade desse problema.

Outra característica que deve ser considerada na escolha da função consenso é sua capacidade de estimar ou não o número de grupos da partição a ser gerada. Caso esse número não seja conhecido, alguma função pode ser utilizada para estimá-lo. Por exemplo, o exame dos valores de alguns tipos de função objetivo para partições com diferentes números de grupo pode ser utilizado nessa estimação [46, 47].

Adicionalmente, pode-se utilizar algum critério para estimar a informação compartilhada entre as partições base e a partição consenso formada. Um dos critérios mais utilizados é a informação mútua normalizada ($NMI$, do inglês *Normalized Mutual Information*) que consiste em uma medida estatística simétrica capaz de quantificar a informação comum entre duas distribuições [16]. Esta medida é dada pela Equação (3), estimada entre duas partições $\pi_a$ e $\pi_b$.

$$\phi^{(NMI)}(\pi^a, \pi^b) = \frac{\sum_{h=1}^{k^a}\sum_{l=1}^{k^b}\left|C_h^a \cap C_l^b\right| \log(\frac{n_o\left|C_h^a \cap C_l^b\right|}{\left|C_h^a\right|\left|C_l^b\right|})}{\sqrt{\left(\sum_{h=1}^{k^a}\left|C_h^a\right| \log(\frac{\left|C_h^a\right|}{n_o})\right)\left(\sum_{l=1}^{k^b}\left|C_l^b\right| \log(\frac{\left|C_l^b\right|}{n_o})\right)}} \tag{3}$$

em que $k^a$ e $k^b$ são os números de grupos das partições $\pi_a$ e $\pi_b$, respectivamente, $C_h^a$ é o $h$-ésimo grupo de $\pi_a$, $C_l^b$ é o $l$-ésimo grupo de $\pi_b$ e $n_o$ é o número de objetos do conjunto de dados.

Em um trabalho anterior [49], os autores definiram a *NMI* com uma normalização que assume o balanceamento entre grupos. Esta normalização, que também é usada por [19], utiliza média aritmética e assume entropia máxima causada pelo balanceamento perfeito. Porém, em [16], os autores alteram a *NMI* de forma que não seja mais necessário assumir grupos balanceados, resultando na Equação (3).

Com base na medida de informação mútua entre duas partições, é definida uma medida de informação mútua entre uma partição $\pi_i$ e um conjunto $\Pi$ de $n_\pi$ partições, como a informação mútua normalizada média ($ANMI$) [16], dada pela Equação (4).

$$\phi^{(ANMI)}(\pi_i, \Pi) = \frac{1}{n_\pi}\sum_{q=1}^{n_\pi}\phi^{(NMI)}(\pi_i, \pi_q) \tag{4}$$

Muitos autores adotam maximizar a *ANMI* como função consenso da combinação [16, 14, 19, 26]. A partição consenso $\pi_f$ é aquela cuja $\phi^{(ANMI)}$ é máxima em relação às

partições base em $\Pi$. O número de grupos da partição consenso é fixo e deve ser escolhido previamente. A partição consenso é obtida pela Equação (5), em que $\pi_i$ corresponde a todas as possíveis partições com $k$ grupos.

$$\pi_f = \arg\max_{\pi_i} \sum_{q=1}^{n_\pi} \phi^{(NMI)}(\pi_i, \pi_q) \tag{5}$$

Em [30], os autores contestam a afirmação feita em [16] de que maximizar a função dada pela Equação (4) pode ser uma forma adequada para estimar o valor do número de grupos mais próximo da estrutura real dos dados. Contudo, eles verificam que combinações com número de grupos próximos a esse valor tendem a ser mais robustas [19], ou seja, tendem a gerar partições mais semelhantes e, portanto, com menor variância de *NMI*. Baseados nessa afirmação, os autores propõem o estudo da variância de $\phi^{(ANMI)}(\pi_i, \Pi)$, calculado por meio de pequenas perturbações em $\Pi$ em [30]. Estas perturbações são obtidas por amostragens de objetos com reposição feitas a partir do conjunto de dados original (*bootstrap*), para cada valor de número de grupos estudado. Segundo os autores, o valor ótimo consiste no número de grupos da partição consenso de menor variância obtida dentre as combinações analisadas. A variância também tem sido utilizada para identificar partições mais robustas.

Em um estudo mais recente, é mostrado uma possível melhora no desempenho da combinação por meio da seleção de um subconjunto das partições base, ao invés de utilizar todo o conjunto [50]. O método utilizado consiste em selecionar as partições base de maior "qualidade" e "diversidade" possíveis. A "qualidade" de uma partição é medida pela soma de *NMI* em relação as outras partições base e a "diversidade" é medida pelo inverso da soma da *NMI* em relação as outras partições selecionadas para esse subconjunto. Os autores propõe três métodos da seleção para medir essas características e os resultados obtidos para dois deles são significativamente melhores do que o obtido com uso de todo o conjunto de partições base.

Seja qual for o método escolhido para avaliar as partições consenso, ainda é preciso utilizar algum tipo de função consenso para gerar estas partições. Nas próximas subseções são apresentadas algumas funções consenso, divididas pelo tipo de técnica utilizada [17]. Parte dos trabalhos envolve o uso de mais de uma função consenso.

### 3.1 Funções baseadas em co-associação

Este tipo de função calcula a similaridade entre dois objetos por meio do número de grupos compartilhados entre eles em todas as partições base. Esta similaridade é utilizada para representar a força de co-associação entre os objetos e é organizada em uma matriz conhecida como matriz de co-associação [19].

Para ilustrar o processo, considere um conjunto de dados $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_o}\}$. São geradas $n_\pi$ partições base $\Pi = \{\pi_1, \pi_2, ..., \pi_{n_\pi}\}$ utilizando algumas das técnicas descritas na Seção 2. Para cada partição $\pi_z$, uma matriz de co-associação $\mathbf{M}^{(z)}$, de tamanho $n_o \times n_o$, é formada de maneira que o conteúdo da posição $\mathbf{M}^{(z)}(i, j)$ seja 1 caso $\mathbf{x}_i$ e $\mathbf{x}_j$ pertençam ao mesmo grupo na partição $\pi_z$ ou 0 caso contrário. Para ilustrar este processo, considere os agrupamentos apresentados na Tabela 1 [16, 15]. As matrizes de co-associação resultantes destas partições são apresentadas na Tabela 2.

**Tabela 1.** Exemplo de conjunto de partições base.

| Partição | *Clusters* |
|---|---|
| $\pi_1$ | $C_1^1 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, C_2^1 = \{\mathbf{x}_4, \mathbf{x}_5\}, C_3^1 = \{\mathbf{x}_6, \mathbf{x}_7\}$ |
| $\pi_2$ | $C_1^2 = \{\mathbf{x}_6, \mathbf{x}_7\}, C_2^2 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}, C_3^2 = \{\mathbf{x}_4, \mathbf{x}_5\}$ |
| $\pi_3$ | $C_1^3 = \{\mathbf{x}_1, \mathbf{x}_2\}, C_2^3 = \{\mathbf{x}_3, \mathbf{x}_4\}, C_3^3 = \{\mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$ |
| $\pi_4$ | $C_1^4 = \{\mathbf{x}_1, \mathbf{x}_4\}, C_2^4 = \{\mathbf{x}_2, \mathbf{x}_5\}$, objetos $\mathbf{x}_3$, $\mathbf{x}_6$ e $\mathbf{x}_7$ não agrupados |

**Tabela 2.** Exemplo de matrizes de co-associação para as partições apresentadas na Tabela 1.

| $\pi_1$ | $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ | $\mathbf{x}_4$ | $\mathbf{x}_5$ | $\mathbf{x}_6$ | $\mathbf{x}_7$ |
|---|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_2$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_3$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_4$ | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| $\mathbf{x}_5$ | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| $\mathbf{x}_6$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| $\mathbf{x}_7$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| $\pi_2$ | $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ | $\mathbf{x}_4$ | $\mathbf{x}_5$ | $\mathbf{x}_6$ | $\mathbf{x}_7$ |
|---|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_2$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_3$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_4$ | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| $\mathbf{x}_5$ | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| $\mathbf{x}_6$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| $\mathbf{x}_7$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| $\pi_3$ | $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ | $\mathbf{x}_4$ | $\mathbf{x}_5$ | $\mathbf{x}_6$ | $\mathbf{x}_7$ |
|---|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_2$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_3$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| $\mathbf{x}_4$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| $\mathbf{x}_5$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| $\mathbf{x}_6$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| $\mathbf{x}_7$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| $\pi_4$ | $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ | $\mathbf{x}_4$ | $\mathbf{x}_5$ | $\mathbf{x}_6$ | $\mathbf{x}_7$ |
|---|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| $\mathbf{x}_2$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| $\mathbf{x}_3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_4$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| $\mathbf{x}_5$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| $\mathbf{x}_6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\mathbf{x}_7$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A matriz de co-associação consenso $\mathbf{M}$ é gerada a partir das matrizes $\mathbf{M}^{(z)}$, $z = 1, ..., n_\pi$, de forma que $\mathbf{M} = (\mathbf{M}^{(1)} + ... + \mathbf{M}^{(n_\pi)})/n_\pi$ [21], ilustrada na Tabela 3 . A partição consenso é obtida por meio da aplicação de alguma técnica a $\mathbf{M}$. Um exemplo de técnica utilizada para gerar a partição consenso consiste em estabelecer um limiar $\theta$ tal que $0 < \theta < 1$ em que, caso $\mathbf{M}(i, j) \geq \theta$, os objetos $\mathbf{x}_i$ e $\mathbf{x}_j$ são associados e inseridos no mesmo grupo [13]. Após a associação entre objetos, se algum objeto $\mathbf{x}_i$ não estiver em nenhum grupo, ele formará sozinho um grupo (*singleton*). Escolher um valor para o limiar $\theta$ e aplicá-lo a matriz $\mathbf{M}$ é equivalente ao uso do algoritmo hierárquico de ligação simples (*single link*) usando $\mathbf{M}$ como matriz de similaridade [13].

**Tabela 3.** Exemplo de matriz de co-associação consenso.

| $\mathbf{M}$ | $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ | $\mathbf{x}_4$ | $\mathbf{x}_5$ | $\mathbf{x}_6$ | $\mathbf{x}_7$ |
|---|---|---|---|---|---|---|---|
| $\mathbf{x}_1$ | 1.00 | 0.75 | 0.50 | 0.25 | 0.00 | 0.00 | 0.00 |
| $\mathbf{x}_2$ | 0.75 | 1.00 | 0.50 | 0.00 | 0.25 | 0.00 | 0.00 |
| $\mathbf{x}_3$ | 0.50 | 0.50 | 1.00 | 0.25 | 0.00 | 0.00 | 0.00 |
| $\mathbf{x}_4$ | 0.25 | 0.00 | 0.25 | 1.00 | 0.05 | 0.00 | 0.00 |
| $\mathbf{x}_5$ | 0.00 | 0.25 | 0.00 | 0.05 | 1.00 | 0.25 | 0.25 |
| $\mathbf{x}_6$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.25 | 1.00 | 0.75 |
| $\mathbf{x}_7$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.25 | 0.75 | 1.00 |

O uso de co-associação pode se dar em partições geradas por amostras de objetos de $\mathbf{X}$ [21]. Neste caso alguns objetos da base de dados podem não estar presentes na partição, como acontece com a partição $\pi_4$ que possui valores "vazios" para estes objetos nas matrizes de co-associação. A matriz de co-associação consenso $\mathbf{M}$ será calculada pela média dos valores presentes nas matrizes geradas pelas partições base, desconsiderando as entradas vazias.

Outra forma de gerar a partição consenso consiste em aplicar algum algoritmo de agrupamento baseado em (dis)similaridade na matriz consenso [36, 16, 29, 30, 21, 33]. Nesse caso, $\mathbf{M}$ pode ser interpretada de duas formas: como matriz de similaridade entre os objetos do conjunto $\mathbf{X}$ ou como os objetos propriamente ditos. No primeiro caso, a informação contida na matriz $\mathbf{M}$ deve ser tratada de acordo com o algoritmo de agrupamento utilizado (por exemplo, uma estratégia consiste em utilizar $1 - \mathbf{M}$ para algoritmos baseados em dissimilaridade [33]). No segundo caso, o algoritmo de agrupamento é aplicado a $\mathbf{M}$ como matriz de dados, utilizando as similaridades presentes como atributos dos objetos. Essa abordagem vem obtendo bons resultados [51].

O número de grupos da partição consenso pode ser pré-especificado ou encontrado por meio da análise de $\mathbf{M}$. No segundo caso é possível derivá-lo. Uma das formas de se fazer isso é por meio do uso do algoritmo hierárquico de ligação simples (como exemplificado anteriormente), a partir das similaridades contidas em $\mathbf{M}$ [30, 13, 44, 34]. Definindo $d(C_j, C_y)^i$ como a dissimilaridade entre o $j$-ésimo e o $y$-ésimo grupo, a serem aglomerados durante a $i$-ésima iteração do algoritmo, a diferença necessária para que esta aglomeração ocorra pode ser calculada por $\Delta_i = d(C_j, C_y)^i - d(C_j, C_y)^{i-1}$. Se $\Delta_i$ for alto, então o sistema pode estar forçando uma estrutura que não ocorre nos dados. Portanto, o processo deve ser parado quando o maior valor de $\Delta_i$ é alcançado e uma partição com $n_\pi - i + 1$ grupos é obtida na iteração $i$. Um procedimento semelhante também é aplicado em [19], em que é feito um corte do dendrograma resultante na posição com maior diferença entre as iterações (*lifetime*).

Outra forma de se derivar o número de grupos $k$ consiste em executar a combinação para determinado intervalo de $k$ e analisar as matrizes de co-associação consenso obtidas. Se existir uma estrutura bem definida no conjunto de dados, os algoritmos de agrupamento utilizados serão capazes de identificá-la separadamente e o consenso entre eles será alto [21] . O valor de $k$ é escolhido pela matriz de co-associação $\mathbf{M}$ de maior consenso (confidência)

[21] em relação as partições base e a partição consenso é obtida por meio da aplicação de um algoritmo de agrupamento nessa matriz.

Em [22], os autores fazem uma avaliação da estabilidade da combinação de partições geradas pelo algoritmo *k*-médias em relação a inicializações aleatórias de grupos. Esta combinação é feita por meio do uso de matriz de co-associação e a estabilidade entre partições é calculada de duas formas: pareada, por meio do cálculo do índice *AR* (Equação 2) entre as partições avaliadas, e não pareada, por meio do cálculo da entropia destas partições. Os resultados indicam que, de forma geral, esse tipo de combinação é mais estável do que as partições base individualmente, em especial para casos onde o número de grupos do conjunto de dados é grande. Também é sugerido o uso da estabilidade das partições na análise do melhor número de grupos para a partição consenso e o conjunto de dados.

Alguns dos problemas da abordagem baseada em co-associação são a falta de uma metodologia para a definição do algoritmo de agrupamento a ser utilizado para a combinação e a baixa confiabilidade da estimativa dos valores de co-associação quando um pequeno número de partições é utilizado.

### 3.2 Funções baseadas em votação

Também conhecido como agrupamento pela maioria de votos [13], as funções baseadas em votação utilizam um mecanismo de votação para atribuir os objetos aos grupos da partição consenso. Entretanto, antes que seja possível votar em qual grupo um determinado objeto pertencerá na partição consenso, é necessário obter a correspondência entre os grupos da partição consenso e os grupos de cada partição base. Ou seja, é necessário encontrar a correspondência entre os rótulos dos objetos de partições distintas. Em [42], os autores apresentam argumentos formais sobre a efetividade do uso da combinação de agrupamentos utilizando função consenso baseada em voto por pluralidade e sobre a perspectiva do uso de uma partição gerada estocasticamente a partir da mudança dos valores dos rótulos. Os autores mostram que, se a probabilidade dos agrupamentos utilizados rotularem corretamente um objeto for maior do que a probabilidade de isso acontecer aleatoriamente, a acurácia da partição consenso em relação a estrutura dos dados aumenta com o aumento do número de partições base, tendendo a 1 quando o número de partições cresce infinitamente.

Um esquema de votação para partições *fuzzy* ou *crisp* com um número de grupos $k$ fixo foi proposto em [52] e utilizado em outros trabalhos [53, 38]. Para esse esquema, foi proposta uma forma de medir a dissimilaridade entre partições, utilizada para procurar por uma partição $\pi_f$ para um dado conjunto de dados $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_o}\}$ que represente de maneira ótima as $n_\pi$ partições iniciais $\Pi = \{\pi_1, \pi_2, ..., \pi_{n_\pi}\}$. Cada partição é representada por uma matriz $\mathbf{U}^{(p)}$, com $p = 1, ..., n_\pi$, em que cada elemento $u_{ij}^{(p)}$ contém o valor de pertinência do objeto $\mathbf{x}_i$ para o $j$-ésimo grupo da partição $\pi_p$. Da mesma forma, a partição

consenso $\pi_f$ é representada por uma matriz $\mathbf{F}$ e $f_{ij}$ corresponde a um elemento da matriz $\mathbf{F}$. A função de dissimilaridade entre $\mathbf{U}^{(p)}$ e $\mathbf{F}$ é dada pela média da distância quadrática, calculada pela Equação (6).

$$d(\mathbf{U}^{(p)}, \mathbf{F}) = \frac{1}{n_o} \sum_{i=1}^{n_o} ||\mathbf{u}_i^{(p)} - \mathbf{f}_i||^2 \tag{6}$$

Deve ser observado que as considerações feitas são válidas apenas para casos em que haja correspondência entre os rótulos dos grupos. Entretanto, encontrar essa correspondência é difícil pois é equivalente a comparar todos os grupos de uma partição com todos os grupos da outra partição e encontrar os grupos correspondentes. Para grandes quantidades de grupos ou partições, esse problema torna-se computacionalmente intratável. Neste caso, resume-se a testar todas as $k!$ permutações de linhas de $\mathbf{F}$ e $\mathbf{U}$. Para se obter uma aproximação heurística de uma rotulação consistente, todas as partições devem ser re-rotuladas com base em sua melhor concordância com uma partição referência [53, 38]. Essa tarefa pode ser computacionalmente dispendiosa para valores altos de $k$. Uma forma de solucionar este problema consiste em aplicar os passos as seguir [53]:

1. Construir a matriz de confusão entre os grupos de $\mathbf{F}$ e $\mathbf{U}$, de forma que o elemento da posição $ij$ indique a quantidade de objetos em comum entre o grupo representado pela $i$-ésima coluna de $\mathbf{F}$ e o grupo representado pela $j$-ésima coluna de $\mathbf{U}$.
2. Encontrar o maior elemento dessa matriz.
3. Associar os dois grupos correspondentes a esse elemento.
4. Remover os dois grupos da matriz.
5. Se ainda houver grupos na matriz, voltar ao passo 2.

A partir da correspondência gerada a partir das associações obtidas entre os grupos, gera-se uma função de permutação $\hat{\varepsilon}$, que é utilizada para reodernar as colunas de $\mathbf{U}$, de modo a maximizar a diagonal da matriz de confusão entre $\hat{\varepsilon}(\mathbf{U})$ e $\mathbf{F}$. Esse processo tem por objetivo obter a função consenso entre as $n_\pi$ partições base, de forma que a partição consenso seja calculada na primeira iteração por:

$$\mathbf{F}^{(1)} = \mathbf{U}, \tag{7}$$

e na $i$-ésima iteração consecutiva por:

$$\mathbf{F}^{(i)} = \frac{1}{i}\sum_{l=1}^{i}\hat{\varepsilon}_l(\mathbf{U}^{(l)}) = \frac{i-1}{i}\mathbf{F}^{(i-1)} + \frac{1}{i}\hat{\varepsilon}_i(\mathbf{U}^{(i)}) \quad (8)$$

até que $i = n_\pi$. A matriz $\mathbf{F}^{(n_\pi)}$ resultante representa o agrupamento consenso final. De forma análoga, outros métodos de renumeração [40], ou até mesmo matrizes de co-associação [28], podem ser utilizadas para estabelecer a correspondência entre grupos. Outra forma de solucionar o problema de correspondência dos rótulos se dá por meio da aplicação do método Húngaro [54, 55], que possui complexidade computacional $O(k^3)$.

Apesar do algoritmo proposto em [53] inicialmente não estimar o número de grupos do conjunto de dados, uma variante com esse objetivo é proposta em [56]. Essa variante aplica o algoritmo de votação para gerar combinações para um número crescente de grupos, dentro de um intervalo de interesse, e calcular uma medida de certeza das partições consenso encontradas. Essa medida de certeza consiste na média dos percentuais em que os objetos do conjunto de dados se encontram no grupo da partição consenso correspondente a seus grupos de origem nas partições base. Devido a um espalhamento cada vez maior dos objetos entre mais e mais grupos, existe uma tendência desta medida decrescer com o aumento do número de grupos das partições geradas. Por este motivo, os autores propõem o uso de uma medida baseada na diferença dos valores obtidos entre diferentes números de grupos, dada pela Equação (9):

$$\hat{k} = \max_k\{[certeza(\pi_f^{(k)}) - certeza(\pi_f^{(k-1)})] - [certeza(\pi_f^{(k+1)}) - certeza(\pi_f^{(k)})]\} \quad (9)$$

em que $certeza(\pi_f^{(k)})$ é medida de certeza da partição consenso gerada com $k$ grupos e $\hat{k}$ é o número estimado de grupos para o conjunto de dados.

Uma variante desse algoritmo utiliza perturbação para gerar grupos mais distintos nas partições base, aumentando sua diversidade [27]. Outra variante executa o algoritmo várias vezes para gerar diversas partições consenso e combiná-las em uma única partição final [57], por meio da relação de verossimilhança entre cada par de grupos pertencentes a partições consenso distintas. Nesse caso, pares e seqüências dos grupos com maiores verossimilhança são combinados em um único grupo, de forma que a matriz de pertinência do novo grupo seja a soma das matrizes dos grupos originais. O processo é repetido até que um critério de parada seja satisfeito (ex.: um determinado valor de certeza é alcançado). Também foi publicada uma variante para uso específico em aprendizado semi-supervisionado [24].

Tumer & Agogino [26] propuseram um algoritmo de combinação de agrupamentos que nomearam de grupos de votação ativa (*VAC* - do inglês *Voting Active Clusters*). Nesse algoritmo, cada grupo das partições base é utilizado em um mecanismo de votação para

decidir a qual grupo da partição consenso os objetos devem pertencer. Esse mecanismo utiliza aprendizado por reforço, que permite avaliar a partição consenso por meio de sua medida de informação mútua em relação as partições base e, em seguida, recompensar as ações de voto de acordo com o resultado dessa avaliação.

Recentemente, Ayad & Mohamed [58] utilizam a idéia de voto cumulativo baseada no trabalho feito em [53]. Eles propuseram um esquema de votação cumulativo baseado no uso de uma matriz de probabilidades, no lugar do tradicional voto um-para-um utilizado por outros algoritmos. Cinco algoritmos baseados neste esquema de votação são descritos e os resultados experimentais mostram melhorias na qualidade (medida por *NMI*) das partições encontradas, comparadas com funções baseadas em grafo propostas por [16] (Descritas na Seção 3.3).

### 3.3 Funções baseadas em grafo/hipergrafo

As funções baseadas em grafos representam as partições base por um grafo ou por um hipergrafo. Nesses casos, a partição consenso é encontrada pelo uso de uma técnica de particionamento de grafos ou hipergrafos [18].

Três algoritmos baseados nesta heurística foram propostos em [16]: *CSPA* (*Cluster-based Similarity Partitioning Algorithm*), *HGPA* (*HiperGraph-Partitioning Algorithm*) e *MCLA* (*metaCLustering Algorithm*). Os autores utilizam os três algoritmos, cada um gerando uma partição consenso, e usam uma função supra-consenso baseada em informação mútua compartilhada (Equação (4)) para escolher qual delas é a melhor.

Inicialmente, o conjunto de partições é representado em forma de hipergrafo, representado por uma matriz de adjacências na qual cada grupo das partições $\Pi = (\pi_1, \pi_2, ..., \pi_{n_\pi})$ é representado por uma hiperaresta. Essa matriz é construída pela concatenação das matrizes binárias de pertinência de cada partição $\mathbf{H} = (\mathbf{H}^1, \mathbf{H}^2, ..., \mathbf{H}^{n_\pi})$. As linhas da matriz de pertinência de uma partição correspondem aos objetos e as colunas correspondem aos seus grupos. Cada célula da matriz contém o valor 1 se o objeto pertence ao grupo e 0 caso contrário. A Tabela 4 ilustra o hipergrafo formado a partir dos agrupamentos presentes na Tabela 1.

O algoritmo *CSPA* possui três etapas. A primeira etapa gera a matriz de co-associação consenso entre as partições base. Essa matriz será utilizada na segunda etapa como matriz de similaridade, na construção do grafo de similaridade induzido, em que os pesos das arestas são dados pelos valores dessa matriz. Na terceira etapa, o grafo de similaridade induzido é particionado pelo algoritmo METIS [59]. O objetivo deste particionamento é dividir o grafo em grupos balanceados, cujo quantidade é definida pelo usuário, a partir da eliminação do menor número de arestas ou das arestas com os menores pesos.

**Tabela 4.** Problema ilustrativo de combinação das partições $\pi_1$, $\pi_2$, $\pi_3$ e $\pi_4$: vetores representando as partições originais à esquerda e o hipergrafo equivalente com 11 hiperarestas à direita (exemplo retirado de [16]).

| | | | | | | | $\mathbf{H}^1$ | | | $\mathbf{H}^2$ | | | $\mathbf{H}^3$ | | | $\mathbf{H}^4$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\pi_1$ | $\pi_2$ | $\pi_3$ | $\pi_4$ | | | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $h_5$ | $h_6$ | $h_7$ | $h_8$ | $h_9$ | $h_{10}$ | $h_{11}$ |
| $\mathbf{x}_1$ | 1 | 2 | 1 | 1 | | $v_1$ | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| $\mathbf{x}_2$ | 1 | 2 | 1 | 2 | | $v_2$ | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| $\mathbf{x}_3$ | 1 | 2 | 2 | ? | $\Leftrightarrow$ | $v_3$ | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| $\mathbf{x}_4$ | 2 | 3 | 2 | 1 | | $v_4$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| $\mathbf{x}_5$ | 2 | 3 | 3 | 2 | | $v_5$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| $\mathbf{x}_6$ | 3 | 1 | 3 | ? | | $v_6$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| $\mathbf{x}_7$ | 3 | 1 | 3 | ? | | $v_7$ | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

No algoritmo *HGPA*, a combinação é tratada como um problema de particionamento de um hipergrafo, no qual as hiperarestas representam grupos. Diferentemente do algoritmo *CSPA*, os pesos das hiperarestas e vértices são iguais. Esse particionamento é feito cortando um número mínimo de hiperarestas, ou seja, de grupos representados pelas mesmas, de forma a gerar um número de grupos pré-definido. Para isso, é utilizado o pacote de particionamento de hipergrafos HMETIS [60].

Dentre os três algoritmos propostos em [16], o *MCLA* resultou em partições com bons valores para a medida *NMI* e robustez em relação a ruído. Por esse motivo, ele será apresentado em maiores detalhes nos passos a seguir:

- Construir um metagrafo: cada grupo das partições iniciais (as hiperarestas do hipergrafo descrito) é considerado um vértice de um outro grafo regular não direcionado, o metagrafo. Seja $C_j^i$ o $j$-ésimo grupo da $i$-ésima partição, o peso da aresta entre os vértices correspondentes aos grupos $C_j^i$ e $C_t^s$ é dado pela medida de Jaccard estendida, também conhecida como coeficiente de Tanimoto [61]. Para vetores binários, ela é equivalente a razão da cardinalidade da intersecção pela cardinalidade da união dos objetos pertencentes aos dois grupos, dada conforme a Equação (10).

$$w(C_j^i, C_t^s) = \frac{\left|C_j^i \cap C_t^s\right|}{\left|C_j^i \cup C_t^s\right|} \tag{10}$$

- Agrupar as hiperarestas (grupos): a combinação dos grupos é feita pelo particionamento do metagrafo em $k$ metagrupos balanceados, sendo $k$ definido pelo usuário. Para isso, o pacote de particionamento de grafos utilizado no *CSPA*, METIS [59], é utilizado para particionar o metagrafo. O objetivo desta fase é encontrar os grupos das partições iniciais que são correspondentes. Novos grupos são formados a partir de um conjunto de grupos correspondentes. Esses grupos são chamados metagrupos.

- Unir os conjuntos de cada metagrupo: transforma as hiperarestas em uma única metahiperaresta para cada metagrupo. Cada objeto pode pertencer a mais de um metagrupo. Assim, para cada metahiperaresta, é calculado um vetor de associação descrevendo o nível de associação de cada objeto com o metagrupo. O vetor de associação é obtido calculando-se a média dos vetores que representam as hiperarestas de um metagrupo. Em outras palavras, o nível de associação de um objeto a um metagrupo é dado pela média do número de grupos desse metagrupo, que contêm o objeto.

- Determinar o metagrupo consenso de cada objeto: cada objeto é associado ao metagrupo para o qual ele possui o maior valor de associação. Desempates são decididos aleatoriamente. A partição dos objetos indicada pelos metagrupos é a partição consenso $\pi_f$ resultante do *ensemble*. Deve-se notar que não é garantido que todo metagrupo tenha pelo menos um objeto. Assim, a partição $\pi_f$ tem no máximo (e não exatamente) $k$ grupos.

Para exemplificar o uso do *MCLA*, a Figura 1 ilustra o metagrafo gerado para as partições da Tabela 4 e seus respectivos hipergrafos. Nessa tabela, cada objeto $\mathbf{x}_i$ corresponde a um vértice e cada hiperaresta $h_j$ representa um dos grupos de uma das partições. No metagrafo, cada hiperaresta dos hipergrafos se torna um vértice. Os pesos das arestas entre os vértices são representados por diferentes tipos de linhas. O particionamento desse grafo em 3 partes resulta nos metagrupos $C_1^M = \{h_3, h_4, h_9\}$, $C_2^M = \{h_2, h_6, h_8, h_{10}\}$ e $C_3^M = \{h_1, h_5, h_7, h_{11}\}$, de forma que as hiperarestas pertencentes ao mesmo metagrupo representem grupos correspondentes.

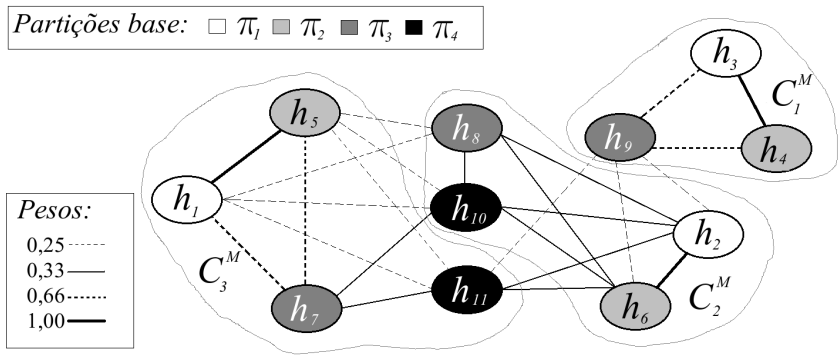


**Figura 1.** Exemplo de metagrafo para as partições $\pi_1$, $\pi_2$, $\pi_3$ e $\pi_4$.

Em seguida, os grupos correspondentes são unidos, formando as metahiperarestas $h_i^M$ representadas na Tabela 5. Nessa tabela também estão representados os respectivos vetores

de associação $a(h_i^M)$. Com base nessas informações, o metagrupo final de cada objeto é determinado. Assim, $\mathbf{x}_1$, por exemplo, vai pertencer ao metagrupo $C_3^M$, pois é o metagrupo com o qual $\mathbf{x}_1$ tem o maior valor de associação (0,75). O mesmo é feito para cada um dos objetos e a partição consenso $\pi^f$ é composta pelos grupos $C_1^f = \{\mathbf{x}_6, \mathbf{x}_7\}$, $C_2^f = \{\mathbf{x}_4, \mathbf{x}_5\}$ e $C_3^f = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$.

**Tabela 5.** Exemplo do *MCLA* - metahiperarestas e vetores de associação

| Vértices | $C_1^M = \{h_3, h_4, h_9\}$ | | $C_2^M = \{h_2, h_6, h_8, h_{10}\}$ | | $C_3^M = \{h_1, h_5, h_7, h_{11}\}$ | |
|---|---|---|---|---|---|---|
| | $h_1^M$ | $a(h_1^M)$ | $h_2^M$ | $a(h_2^M)$ | $h_3^M$ | $a(h_3^M)$ |
| $\mathbf{x}_1$ | 0 | 0 | 1 | 0,25 | 1 | **0,75** |
| $\mathbf{x}_2$ | 0 | 0 | 0 | 0 | 1 | **1** |
| $\mathbf{x}_3$ | 0 | 0 | 1 | 0,25 | 1 | **0,5** |
| $\mathbf{x}_4$ | 0 | 0 | 1 | **1** | 0 | 0 |
| $\mathbf{x}_5$ | 1 | 0,33 | 1 | **0,5** | 1 | 0,25 |
| $\mathbf{x}_6$ | 1 | **1** | 0 | 0 | 0 | 0 |
| $\mathbf{x}_7$ | 1 | **1** | 0 | 0 | 0 | 0 |

Várias outras abordagens com funções consenso baseadas em particionamento de grafos foram publicadas na literatura. Uma delas é o algoritmo proposto em [18], chamado *HBGF* (*Hybrid Bipartite Graph Formulation*), que consiste em construir um grafo bipartido a partir do conjunto de partições a serem combinadas. Tanto objetos quanto grupos são modelados simultaneamente como vértices do grafo que, posteriormente, é particionado com uma técnica tradicional. Os autores utilizam duas técnicas de particionamento de grafo distintas: *Spectral Graph Partitioning* [62] e METIS [59]. O número de grupos da partição consenso é definido pelo usuário.

Outro estudo que utiliza função consenso baseado em grafos foi feito em [63], em que foi proposto o algoritmo GCC (do inglês *Graph-based Consensus Clustering*). Inicialmente, os autores utilizam o algoritmo *k*-médias e um outro algoritmo de agrupamento correlacional [64] para particionar sub-espaços do conjunto de dados formados por amostragens aleatórias dos atributos. Em seguida são geradas as matrizes de freqüências de cada uma das diferentes partições obtidas. Uma matriz de co-associação (Seção 3.1) é gerada a partir do conjunto de matrizes de freqüências e seus valores serão utilizados para compor as arestas de um grafo, cujos nós representam os objetos. Ao final, o algoritmo de corte normalizado proposto em [65] é utilizado para particionar o grafo gerado, de forma que a dissimilaridade entre grupos distintos e a similaridade intra-grupos sejam as máximas possíveis. Os autores também propõem um índice de validação, chamado *Modified Rand Index*, baseado na matriz consensual resultante do algoritmo [63]. Esse índice é utilizado para determinar o número de grupos $k$ da partição consenso.

Uma abordagem semelhante consiste em, dado um número de grupos $k$ para a partição consenso, calcular a intersecção entre $k$ grupos de $k$ componentes [66]. O algoritmo pode

ser divido em três partes: na primeira os grupos de junção são formados, compostos pela intersecção dos grupos de cada componente (partição base). Cada objeto deve pertencer à apenas um dos grupos de junção e objetos que não fizerem parte de nenhum grupo de junção formam grupos individuais (*singletons*). Na segunda parte, um grafo é gerado com cada nó representando um grupo de junção ou um grupo original das partições base, e cada aresta liga um grupo de junção ao respectivo grupo original a que ele pertence. Na terceira parte, o grafo é particionado em $k$ grupos por meio do algoritmo METIS [59], rotulando os objetos segundo seus rótulos originais.

### 3.4 Outras combinações

Além dos tipos já apresentados de função consenso, em [17] é definida uma função consenso cujo objetivo é maximizar a probabilidade de um modelo misto finito em relação a um conjunto de partições base. Esse conjunto de partições é modelado como uma mistura de distribuições multinomiais no espaço formado pelos rótulos dos grupos. Esse modelo é aprimorado com o uso do algoritmo *EM*.

Outra forma de combinar partições consiste em utilizar agentes para agrupar dados distribuídos e votar em qual grupo da partição consenso os objetos devem pertencer [67]. O objetivo da combinação é maximizar a *NMI* entre a partição consenso e os grupos gerenciados pelos agentes, cujo cálculo é feito por meio de troca de informações. Os resultados obtidos mostram tolerância a falhas e desempenho próximo das funções baseadas em grafos propostas em [16], que necessitam de processamento centralizado. O esquema de votação utilizado é detalhado em [26].

### 3.5 Comparações entre funções consenso

Conforme visto nas seções anteriores, existem diversas abordagens para gerar a partição consenso a partir de diferentes tipos de partições base. Na literatura, alguns autores publicaram estudos que comparam algumas dessas abordagens e suas aplicações em diferentes conjuntos de dados. A maioria desses estudos se dá por meio de uma medida que calcula a semelhança entre as partições finais e as partições base ou uma partição conhecida ("ideal").

Strehl & Ghosh [16] comparam seus próprios algoritmos, *CSPA*, *MCLA* e *HGPA*, em termos de complexidade, robustez e estabilidade. Assumindo particionadores de grafos/hipergrafos lineares em relação ao número de grupos $n_o$, o algoritmo *CSPA* é o que apresenta maior complexidade, $O(n_o^2 \cdot k \cdot n_\pi)$, seguido pelo algoritmo *HGPA* , com complexidade $O(n_o \cdot k^2 \cdot n_\pi^2)$, e *MCLA*, com complexidade $O(n_o \cdot k \cdot n_\pi^2)$. Os experimentos feitos pelos autores para obter a $NMI$ (Equação (3)) e $ANMI$ (Equação (4)) de partições consenso oriundas de dados com ruído mostram que o algoritmo *HGPA* tem o pior desempenho. Em relação ao $ANMI$, os algoritmos *CSPA* e *MCLA* apresentaram desempenho semelhante na

presença de pequenas quantidades de ruído, apesar do algoritmo *CSPA* se destacar para grandes quantidades de ruído. O *MCLA* apresenta melhor $NMI$ em boa parte dos conjuntos de dados.

Em [19] , os algoritmos propostos em [16] são comparados com algoritmos baseados em matrizes de co-associação. Com esse objetivo, os autores executam experimentos com conjuntos de dados artificiais e das áreas de bioinformática e segmentação de imagens. Houve grande variação na taxa de erros obtida para cada algoritmo nos diferentes conjunto de dados, em relação à estrutura conhecida. Em alguns casos, as abordagens baseadas em grafos obtiveram menores erros, porém as abordagens baseadas em acúmulo de evidência obtiveram melhores resultados para a maioria dos conjuntos.

Nos experimentos apresentados em [18], os autores observaram que sua abordagem, o HGBF, apresentou um desempenho equivalente ou superior ao obtido pelas abordagens propostas em [16]. Em [68], o desempenho dos mesmos algoritmos são comparados por meio de componentes gerados a partir de agrupamentos hipoteticamente verdadeiros criados por rotulação aleatória. Vários testes foram realizados com esses algoritmos e, em relação ao valor do índice *AR* [45], o *MCLA* obteve resultados ligeiramente melhores. No entanto, os resultados obtidos pelos três algoritmos foram próximos, com casos particulares em que um se destaca mais.

Outro trabalho que merece destaque foi publicado em [46], no qual partições base geradas por *k*-médias e algoritmos hierárquicos são utilizadas para gerar matrizes de co-associação. As partições consenso são geradas a partir de seis metodologias distintas: particionamento por *k*-médias sobre matriz de co-associação; particionamento por ligação simples usando a matriz de co-associação como matriz de similaridade; particionamento por ligação simples sobre matriz de co-associação; particionamento por ligação média sobre a matriz de co-associação; os algoritmos *CSPA* e *HGPA*.

Cem combinações de dez partições base foram executadas para cada caso, utilizando 24 conjuntos de dados artificiais e obtidos de dados médicos. Os resultados foram avaliados por meio da acurácia de classificação e *AR* e posicionados em um *rank* segundo comparações estatísticas feitas com o teste ANOVA. Os melhores resultados foram obtidos por combinações feitas pela aplicação de *k*-médias ou ligação média sobre matriz de co-associação, essa última gerada a partir de partições obtidas pelo *k*-médias. Para esses resultados, o algoritmo de agrupamento é aplicado sobre a matriz de co-associação como se essa fosse composta por objetos. É importante frisar que, dentre os algoritmos propostos por [16], o único algoritmo não comparado por [46] foi o *MCLA*, que é justamente o algoritmo que obteve o melhor desempenho dentre os três propostos em [16].

O trabalho publicado em [69] compara funções consenso baseadas em ligação simples sobre matriz de co-associação, renumeração seguido de votação e particionamento de

grafos (*CSPA*, *HGPA* e *MCLA*). Os experimentos utilizam dados de expressão gênica e a avaliação dos resultados foi feita por meio do índice *AR*, tanto para medir precisão, em relação às partições conhecidas, quanto diversidade, em relação às partições obtidas. Porém, a avaliação difere de outros autores por utilizarem validação cruzada com 10 *folds*, um processo de validação bastante conhecido em classificação supervisionada porém pouco utilizado em classificação não supervisionada. Esse processo consiste em dividir dos dados em 10 *folds*, de forma que agrupamentos base fossem gerados a partir do conteúdo de 9 desses *folds*, considerados como conjunto de treinamento. Partições consenso são obtidas a partir dos agrupamentos base e o *fold* restante, considerado como conjunto de teste, é utilizado para medir a precisão dos agrupamentos encontrados. Os dados do *fold* de teste são classificados a partir do centróide mais próximo e a partição resultante é comparada com a partição conhecida por meio do índice *AR*. O processo é repetido 10 vezes, permutando os papéis de cada *fold*. As partições iniciais foram geradas com os algoritmos *k*-médias, *EM* e agrupamento hierárquico com ligação média. De forma geral, não foi encontrada diferença significativa entre as funções consenso estudadas, com exceção da aplicação de ligação simples sobre a matriz de co-associação que freqüentemente resultou em um pior desempenho, diferentemente do relatado por outros autores anteriormente citados.

O algoritmo *VAC* é comparado com os algoritmos de particionamento de grafos *CSPA*, *HGPA* e *MCLA* em [26]. A comparação levou em consideração a *NMI* das partições consenso resultantes e destacou o desempenho do *VAC* e do *MCLA* em relação aos outros algoritmos. É importante lembrar que, diferentemente dos outros algoritmos comparados, o *VAC* utiliza a *NMI* para guiar o processo de construção da partição consenso, o que pode beneficiá-lo nesse tipo de avaliação.

De forma geral, diferentes autores obtiveram resultados distintos com as mesmas funções consenso. Dentre as causas podem ser citadas o uso de diferentes métodos para gerar as partições iniciais, métodos de experimentação distintos e as características desiguais das estruturas dos diferentes conjuntos de dados utilizados pelos autores. Um ponto comum entre boa parte das análises foi o desempenho relativamente bom do algoritmo MCLA e do particionamento de matrizes de co-associação. Nesses experimentos, algoritmos de votação mais recentes, como o *VAC* e os algoritmos de votação acumulativos propostos em [58], apresentaram resultados comparáveis aos algoritmos baseados em grafos propostos em [16].

## 4 Considerações finais

Neste artigo foram apresentados os principais métodos utilizados para combinar agrupamentos propostos na literatura, incluindo formas de gerar a diversidade inicial necessária para as partições base e diversas maneiras de encontrar uma partição consenso. Em seguida, foi feita uma breve análise sobre comparações entre as metodologias apresentadas nos artigos

encontrados. Diversos novos algoritmos de combinação de agrupamento são propostos com diferentes objetivos e, pelas comparações apresentadas, cada um se sobressai em um tipo de aplicação diferente. Portanto, ao escolher um algoritmo de combinação, faz se necessário avaliar se este é apropriado para a aplicação em questão.

Combinações de agrupamentos também podem ser utilizadas em conjunto com metaheurísticas de otimização. É o caso de algoritmos evolutivos (AEs), que são técnicas de busca e otimização utilizadas para gerar novas soluções por meio da combinação ou alteração de características daquelas que obtiverem melhor desempenho [70]. Dentre esse tipo de algoritmo, podem ser citados o *MOCLE* (do inglês *Multi-Objective Clustering Ensemble*) [15] e o RGFGA (do inglês *Restricted Growth Function Genetic Algorithm*) [71]. Também é possível utilizar um AEs para selecionar o método de combinação mais apropriado para uma determinada aplicação [48]. A análise do uso de AEs em conjunto com combinações é um tema provável para pesquisas futuras.

## Agradecimentos

Os autores agradecem o apoio recebido da FAPESP e do CNPq à esta pesquisa.

## Referências

[1] J. P. Bigus, *Data Mining with Neural Networks: Solving Business Problems from Application Development to Decision Support*. McGraw-Hill, 1996.

[2] K. jae Kim and H. Ahn, "Using a clustering genetic algorithm to support customer segmentation for personalized recommender systems," in *13th International Conference on AI, Simulation, Planning in High Autonomy Systems*, (Jeju Island, Korea), pp. 409–415, Springer Berlin / Heidelberg, 2004.

[3] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*. Adaptative Computation and Machine Learning, MIT Press, 1998.

[4] V. D. Gesú, R. Giancarlo, G. L. Bosco, A. Raimondi, and D. Scaturro, "Genclust: A genetic algorithm for clustering gene expression data," *BMC Bioinformatics*, vol. 6, no. 289, pp. 1–11, 2005.

[5] R. Feldman and J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2006.

[6] R. Saraçoğlu, K. Tütüncü, and N. Allahverdi, "A fuzzy clustering approach for finding similar documents using a novel similarity measure," *Expert Syst. Appl.*, vol. 33, no. 3, pp. 600–605, 2007.

[7] J. B. D. Cabrera, C. Gutiérrez, and R. K. Mehra, "Ensemble methods for anomaly detection and distributed intrusion detection in mobile ad-hoc networks," *Inf. Fusion*, vol. 9, no. 1, pp. 96–119, 2008.

[8] M. Pakhira, S. Bandyopadhyay, and U. Maulik, "A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification.," *Fuzzy Sets Systems*, vol. 155, no. 2, pp. 191–214, 2005.

[9] P. Sneath and R. Sokal, *Numerical Taxonomy*. Freeman, 1973.

[10] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Mathematical Statistics, 1990.

[11] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[12] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning.," in *In Advances in Neural Information Processing Systems*, vol. 7, pp. 231–238, IEEE Press, 1995.

[13] L. I. Kuncheva, *Combining Pattern Classifiers.* John Wiley & Sons, 2004.

[14] A. Topchy, A. Jain, and W. Punch, "Combining multiple weak clusterings," in *Proceedings of the IEEE International Conference on Data Mining (ICDM'2003)*, (Melbourne, Florida, USA), pp. 331–338, 2003.

[15] K. Faceli, *Um framework para análise de agrupamento baseado na combinação multi-objetivo de algoritmos de agrupamento*. PhD thesis, Instituto de Ciências Matemáticas e Computação , ICMC-USP, 2006.

[16] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research - JMLR*, vol. 3, pp. 583–617, 2002.

[17] A. Topchy, A. Jain, and W. Punch, "A mixture model for clustering ensembles," in *Proceedings of the SIAM International Conference on Data Mining (SDM'2004)*, (Lake Buena Vista, Florida, USA), pp. 331–338, 2004.

[18] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc. ICML'04*, (New York, NY, USA), p. 36, 2004.

[19] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[20] K. Punera and J. Ghosh, "Consensus-based ensembles of soft clusterings," *Applied Artificial Intelligence*, vol. 22, no. 7, pp. 780–810, 2008.

[21] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, vol. 52, no. 1-2, pp. 91–118, 2003.

[22] L. I. Kuncheva and D. P. Vetrov, "Evaluation of stability of k-means cluster ensembles with respect to random initialization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1798–1808, 2006.

[23] M. Law, A. Topchy, and A. Jain, "Multiobjective data clustering," *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–424–II–430 Vol.2, June-2 July 2004.

[24] E. Dimitriadou, A. Weingessel, and K. Hornik, "A mixed ensemble approach for the semi-supervised problem," in *ICANN '02: Proceedings of the International Conference on Artificial Neural Networks*, (London, UK), pp. 571–576, 2002.

[25] J. Ghosh, A. Strehl, and S. Merugu, "A consensus framework for integrating distributed clusterings under limited knowledge sharing," in *Proceedings of NSF Workshop on Next Generation Data Mining*, pp. 99–108, 2002.

[26] K. Tumer and A. K. Agogino, "Ensemble clustering with voting active clusters," *Pattern Recogn.Lett.*, vol. 29, no. 14, pp. 1947–1953, 2008.

[27] E. Dimitriadou, A. Weingessel, and K. Hornik, "Fuzzy voting in clustering," in *Fuzzy-Neuro Systems* (G. Brewka, R. Der, S. Gottwald, and A. Schierwagen, eds.), pp. 63–74, Leipziger Universitätsverlag, 1999.

[28] A. L. N. Fred, "Finding consistent clusters in data partitions," in *Second International Workshop on Multiple Classifier Systems (MCS'2001)* (J. Kittler and F. Roli, eds.), vol. 2096 of *Lecture Notes in Computer Science*, (Cambridge, UK), pp. 309–318, 2001.

[29] A. Fred and A. Jain, "Evidence accumulation clustering based on the k-means algorithm," in *Proceedings of Structural and Syntactic Pattern Recognition (SSPR'2002)*, (Windsor, Canada), pp. 442–451, 2002.

[30] A. Fred and A. K. Jain, "Robust data clustering," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR'2003)*, vol. II, (Madison - Wisconsin, USA), 2003.

[31] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, vol. 6, no. 1, pp. 63–81, 2004.

[32] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[33] L. Kuncheva, S. Hadjitodorov, and L. Todorova, "Experimental comparison of cluster ensemble methods," *Information Fusion, 2006 9th International Conference on*, pp. 1–7, July 2006.

[34] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova, "Moderate diversity for better cluster ensembles," *Information Fusion*, vol. 7, no. 3, pp. 264–275, 2006.

[35] Y. Qian and C. Suen, "Clustering combination method," *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 2, pp. 732–735 vol.2, 2000.

[36] P. Kellam, X. Liu, N. J. Martin, C. Orengo, S. Swift, and A. Tucker, "Comparing, contrasting and combining clusters in viral gene expression data," in *Proceedings of 6th Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, pp. 56–62, 2001.

[37] Y. Zeng, J. Tang, J. Garcia-Frias, and G. Gao, "An adaptive meta-clustering approach: Combining the information from different clustering results," in *IEEE Computer Society Bioinformatics Conference (CSB'2002)*, (Stanford, California), pp. 276–287, 2002.

[38] A. Weingessel, E. Dimitriadou, and K. Hornik, "An ensemble method for clustering," in *Distributed Statistical Computing (DSC'2003)*, (Wien, Austria), 2003. `http://www.ci.tuwien.ac.at/Conferences/DSC-2003/` [Acessado em 29/08/2008].

[39] S. Asur, S. Parthasarathy, and D. Ucar, "An ensemble approach for clustering scalefree graphs," in *In Proceedings of the Workshop on Link Analysis: Dynamics and Static of Large Networks*, 2006.

[40] D. Frossyniotis, M. Pertselakis, and A. Stafylopatis, "A multi-clustering fusion algorithm," in *Methods and Applications of Artificial Intelligence, Proceedings of the 2nd Hellenic Conference on AI (SETN'2002)* (I. P. Vlahavas and C. D. Spyropoulos, eds.), vol. 2308 of *Lecture Notes in Computer Science*, (Thessaloniki, Greece), pp. 225–236, Springer Verlag, 2002.

[41] J. Kasturi and R. Acharya, "Clustering of diverse genomic data using information fusion," in *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, (New York, NY, USA), pp. 116–120, ACM, 2004.

[42] A. P. Topchy, M. H. C. Law, A. K. Jain, and A. L. Fred, "Analysis of consensus partition in cluster ensemble," vol. 00, (Los Alamitos, CA, USA), pp. 225–232, IEEE Computer Society, 2004.

[43] W. Gablentz, M. Köppen, and E. Dimitriadou, "Robust clustering by evolutionary computation," in *Online World Conf. on Soft Computing in Industrial Applications*, 2000.

[44] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham, "Ensemble clustering in medical diagnostics," in *CBMS '04: Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems*, (Washington, DC, USA), p. 576, IEEE Computer Society, 2004.

[45] L. J. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.

[46] L. Kuncheva, S. Hadjitodorov, and L. Todorova, "Experimental comparison of cluster ensemble methods," *Information Fusion, 2006 9th International Conference on*, pp. 1–7, July 2006.

[47] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. on Evolutionary Computation*, vol. 34, pp. 56–76, 2007.

[48] S. T. Hadjitodorov and L. I. Kuncheva, "Selecting diversifying heuristics for cluster ensembles," in *7th International Workshop*, pp. 200–209, 2007.

[49] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," in *Proceedings of AAAI*, (Edmonton, Canada), pp. 93–98, 2002.

[50] X. Z. Fern and W. Lin, "Cluster ensemble selection," *Stat. Anal. Data Min.*, vol. 1, no. 3, pp. 128–141, 2008.

[51] E. Pekalska and R. P. W. Duin, *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence)*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2005.

[52] E. Dimitriadou, A. Weingessel, and K. Hornik, "A combination scheme for fuzzy clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 7, pp. 901–912, 2002.

[53] E. Dimitriadou, *Exploratory Data Analysis and Applications*. PhD thesis, Technische Universität Wien, 2003.

[54] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[55] A. Frank, "On kuhn's hungarian method - a tribute from hungary," Tech. Rep. 14, Egrerváry Research Group, 2004. `http://www.cs.elte.hu/egres/tr/egres-04-14.pdf`.

[56] E. Dimitriadou, A. Weingessel, and K. Hornik, "Voting in clustering and finding the number of clusters," in *AIDA 99 : Proceedings of the International Symposium on Advances in Intelligent Data Analisys*, pp. 291–296, Academic Press, 1999.

[57] E. Dimitriadou, A. Weingessel, and K. Hornik, "Voting-merging: An ensemble method for clustering," in *ICANN '01: Proceedings of the International Conference on Artificial Neural Networks*, (London, UK), pp. 217–224, 2001.

[58] H. G. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 160–173, 2008.

[59] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1999.

[60] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: applications in vlsi domain," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 7, no. 1, pp. 69–79, 1999.

[61] T. T. Tanimoto, "An elementary mathematical theory of classification and prediction.," Tech. Rep. 13, IBM Report, 1958.

[62] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, vol. 14, 2002.

[63] Z. Yu, H. Wong, and H. Wang, "Graph based consensus clustering for class discovery from gene expression data," *Bioinformatics*, vol. 23, pp. 2888–2896, 2007.

[64] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

[65] J. Shi and J. Malik., "Normalized cuts and image segmentation," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 22, pp. 888–905, 2000.

[66] T. Hu, L. Liu, C. Qu, and S. Y. Sung, "Joint cluster based co-clustering for clustering ensembles," in *Advanced Data Mining and Applications, Lecture Notes in Computer Science* (G. Wang, J. F. Peters, A. Skowron, and Y. Yao, eds.), vol. 4093, pp. 284–295, Springer Berlin - Heidelberg, 2006.

[67] A. Agogino and K. Tumer, "Efficient agent-based cluster ensembles," in *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, (New York, NY, USA), pp. 1079–1086, ACM, 2006.

[68] T. Hu, W. Zhao, X. Wang, and Z. Li, "A comparison of three graph partitioning based methods for consensus clustering," in *Lecture Notes in Computer Science* (G. Wang, J. F. Peters, A. Skowron, and Y. Yao, eds.), vol. 4062, pp. 468–475, Springer, 2006.

[69] M. C. P. Souto, D. S. A. Araujo, and B. L. C. Silva, "Cluster ensemble for gene expression microarray data: Accuracy and diversity," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 4176–4181, IEEE, 2006.

[70] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence.* New York, USA: IEEE Press, 1995.

[71] S. Swift, A. Tucker, J. Crampton, and D. Garway-Heath, "An improved restricted growth function genetic algorithm for the consensus clustering of retinal nerve fibre data," in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 2174–2181, ACM, 2007.